

**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ЗАДАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ

Направление подготовки	<i>09.03.01 Информатика и вычислительная техника</i>
Направленность (профиль)	<i>Информационные технологии в промышленности</i>
Уровень высшего образования	<i>бакалавриат</i> <i>(бакалавриат, специалитет, магистратура)</i>
Квалификация выпускника	<i>бакалавр</i>



Администрирование сетевых устройств в инфокоммуникационных системах

Методическое пособие по выполнению лабораторных работ и практических занятий

Практическое занятие 1. Установка, настройка и конфигурирование виртуальной машины

Цель работы:

- Приобретение навыков установки и создания виртуальных машин в Oracle VM VirtualBox
- Приобретение навыков установки и начальной настройки операционной системы Ubuntu

Задание:

1. Скачать или получить у преподавателя образ установочного диска ОС Ubuntu
2. Создать виртуальную машину в VirtualBox.
3. Установить ОС Ubuntu на созданную виртуальную машину.
4. В установленной операционной системе:
 - установить дополнения гостевой ОС
 - настроить рабочие столы (эффекты, изображения);
 - изменить раскладку клавиатуры по умолчанию;
 - определить тип сеанса загружаемую по умолчанию.
5. Ознакомиться и описать панель инструментов Ubuntu.

Краткие теоретические сведения

Занятия по дисциплине «Операционные системы проводятся с использованием технологии виртуализации.

Виртуализация — предоставление набора вычислительных ресурсов или их логического объединения, абстрагированное от аппаратной реализации, и обеспечивающее при этом логическую изоляцию друг от друга вычислительных процессов, выполняемых на одном физическом ресурсе.

Примером использования виртуализации является возможность запуска нескольких операционных систем на одном компьютере: при том каждый из экземпляров таких гостевых операционных систем работает со своим набором логических ресурсов (процессорных, оперативной памяти, устройств хранения), предоставлением которых из общего пула, доступного на уровне оборудования, управляет хостовая операционная система — гипервизор.

Благодаря использованию данной технологии у студента появляется возможность безопасно работать с различными гостевыми операционными системами, не подвергая опасности настройку хостовой операционной системы компьютера учебного класса.

В качестве средства виртуализации будем использовать программное средство Oracle VM VirtualBox.

VirtualBox (Oracle VM VirtualBox) — программный продукт виртуализации для операционных систем Microsoft Windows, Linux, FreeBSD, macOS, Solaris/OpenSolaris,



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ReactOS, DOS и других. Использование виртуальной машины (ВМ) на домашнем ПК, прежде всего, позволит одновременно запускать несколько операционных систем (гостевые ОС).

К примеру, в данный момент на компьютере или ноутбуке установлен один из выпусков операционной системы Microsoft Windows (хостовая ОС). Установка же виртуальной машины, в данном случае VirtualBox, позволяет использовать в среде хостовой ОС любые другие системы (гостевые), включая macOS, Linux, Android, Windows и так далее, вариантов здесь может быть очень много.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Программа была создана компанией Innotek с использованием исходного кода Qemu. Первая публично доступная версия VirtualBox появилась 15 января 2007 года. В феврале 2008 года Innotek был приобретён компанией Sun Microsystems, модель распространения VirtualBox при этом не изменилась. В январе 2010 года Sun Microsystems была поглощена корпорацией Oracle, модель распространения осталась прежней.

Ключевые возможности

Кроссплатформенность

Модульность

Поддержка USB 2.0, когда устройства хост-машины становятся доступными для гостевых операционных систем (только в проприетарной версии)

Поддержка 64-битных гостевых систем (начиная с версии 2.0), даже на 32-битных хост-системах (начиная с версии 2.1, для этого обязательна поддержка технологии виртуализации процессором)

Поддержка SMP на стороне гостевой системы (начиная с версии 3.0, для этого обязательна поддержка технологии виртуализации процессором)

Встроенный RDP-сервер, а также поддержка клиентских USB-устройств поверх протокола RDP (только в проприетарной версии)

Экспериментальная поддержка аппаратного 3D-ускорения (OpenGL, DirectX 8/9 (с использованием кода wine) (только в 32-битных Windows XP, Vista, 7 и 8, для гостевых DOS / Windows 3.x / 95 / 98 / ME поддержка аппаратного 3D-ускорения не предусмотрена)

Поддержка образов жёстких дисков VMDK (VMware) и VHD (Microsoft Virtual PC), включая snapshots (начиная с версии 2.1)

Поддержка iSCSI (только в проприетарной версии)

Поддержка виртуализации аудиоустройств (эмуляция AC97 или SoundBlaster 16 или Intel HD Audio на выбор)

Поддержка различных видов сетевого взаимодействия (NAT, Host Networking via Bridged, Internal)

Поддержка цепочки сохранённых состояний виртуальной машины (snapshots), к которым может быть произведён откат из любого состояния гостевой системы

Поддержка Shared Folders для простого обмена файлами между хостовой и гостевой системами (для гостевых систем Windows 2000 и новее, Linux и Solaris)[6]

Поддержка интеграции рабочих столов (seamless mode) хостовой и гостевой операционной системой

Поддержка формата OVF/OVA

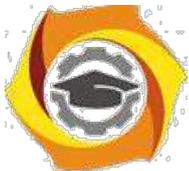
Есть возможность выбора языка интерфейса (поддерживается и русскоязычный интерфейс)

Базовая версия полностью открыта по лицензии GNU GPL, соответственно нет ограничений в использовании

Пакет дополнений

VirtualBox Guest Additions — комплект программного обеспечения, устанавливаемый в гостевую операционную систему и расширяющий её возможности по взаимодействию с системой виртуализации и хост-системой. Например, после установки специального драйвера «виртуальной видекарты» становится возможным изменять разрешение рабочего стола гостевой ОС произвольным образом вслед за размером окна VirtualBox, в котором запущена виртуальная машина.

До версии 4.0.0 существовало две версии, различавшиеся по лицензии и функциональности. Начиная с 4.0.0 закрытые компоненты вынесены в отдельный пакет дополнений (Extension Pack):



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Пакет дополнений содержит закрытые компоненты и распространяется под проприетарной лицензией PUEL (бесплатно только в персональных целях или для ознакомления):

RDP сервер — позволяет подключаться к виртуальной системе удалённо с помощью любого RDP-совместимого клиента;

Поддержка USB — позволяет передавать виртуальной машине USB устройства;

Intel PXE boot ROM — загрузка операционной системы по сети. Используется для создания тонких клиентов/бездисковых рабочих станций.

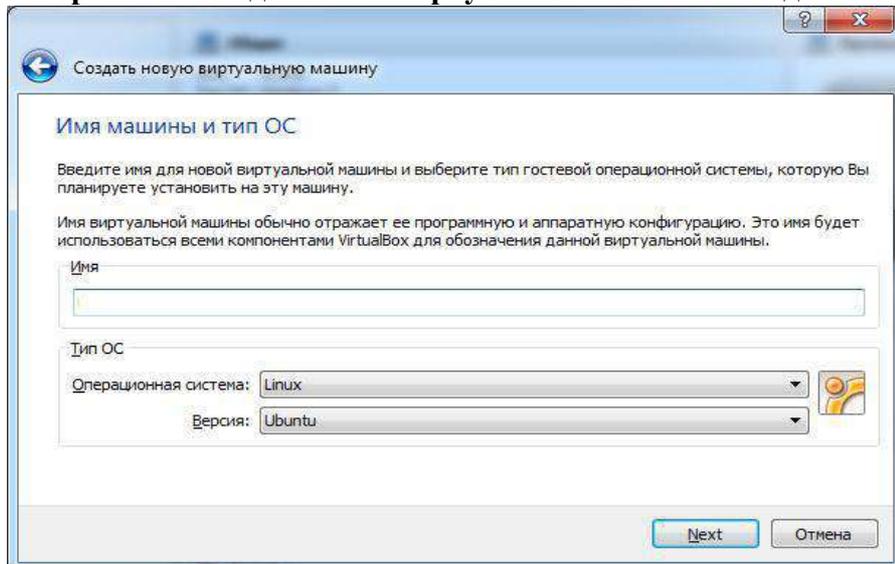
Ход работы:

Установка операционной системы Ubuntu производится на виртуальную машину Oracle VM VirtualBox. Сначала необходимо установить Oracle VM VirtualBox и создать виртуальную машину.

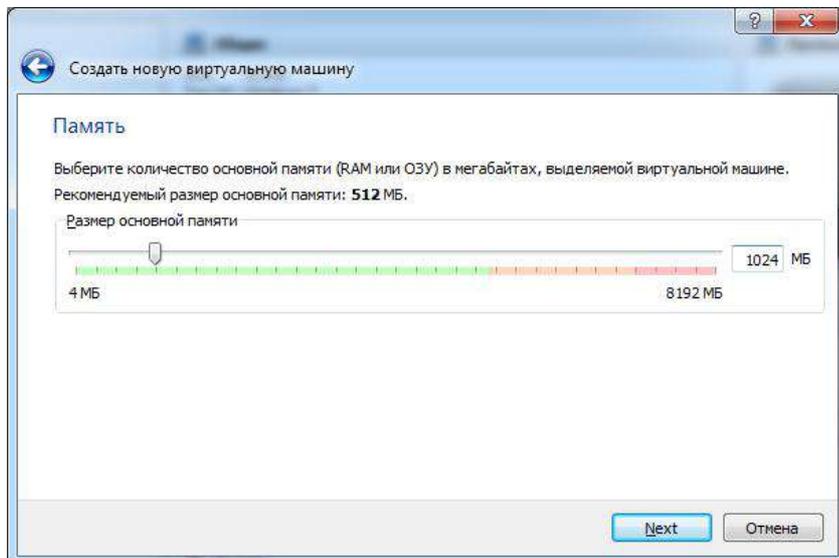
Установка Oracle VM VirtualBox производится с настройками по умолчанию. После установки виртуальной машины установить пакет расширения (Oracle_VM_VirtualBox_Extension_Pack).

1) Создание виртуальной машины:

Выбрать тип создаваемой виртуальной машины и задать имя.



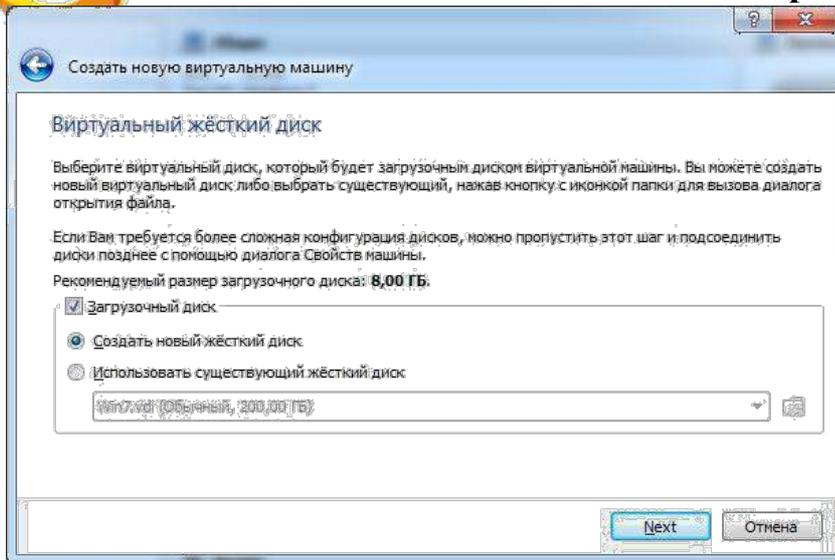
Установить объем памяти 1024 Мб



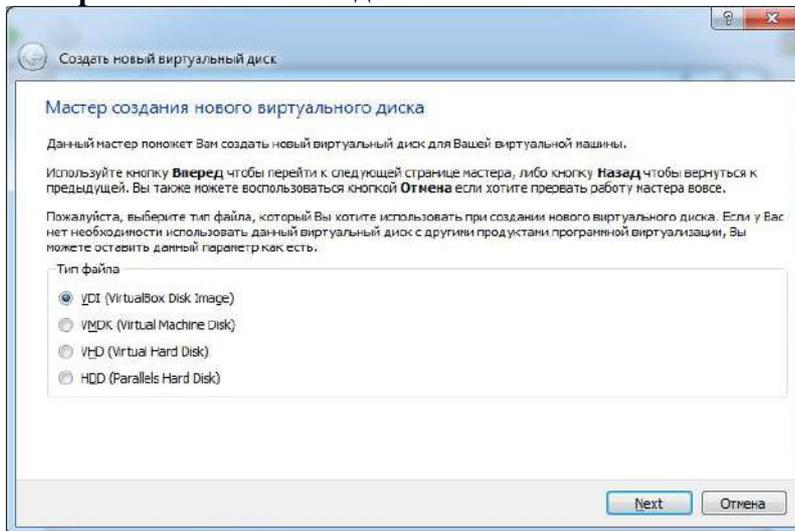
Создать новый жесткий диск



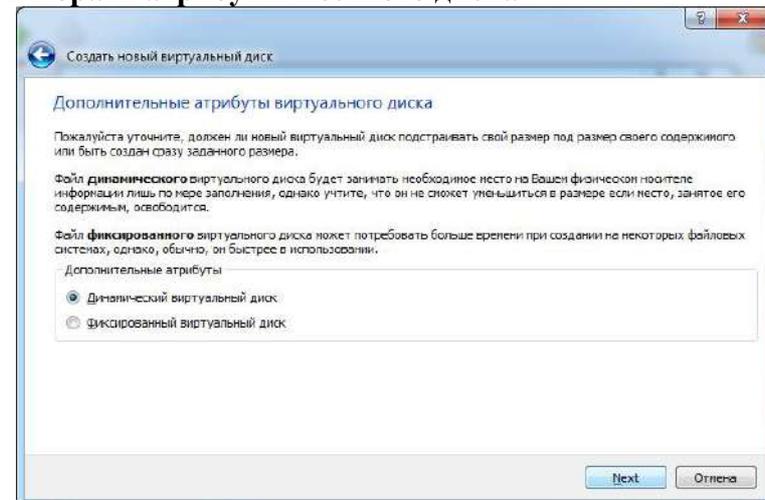
Негосударственное частное образовательное учреждение высшего образования «Университет»



Выбрать тип жесткого диска



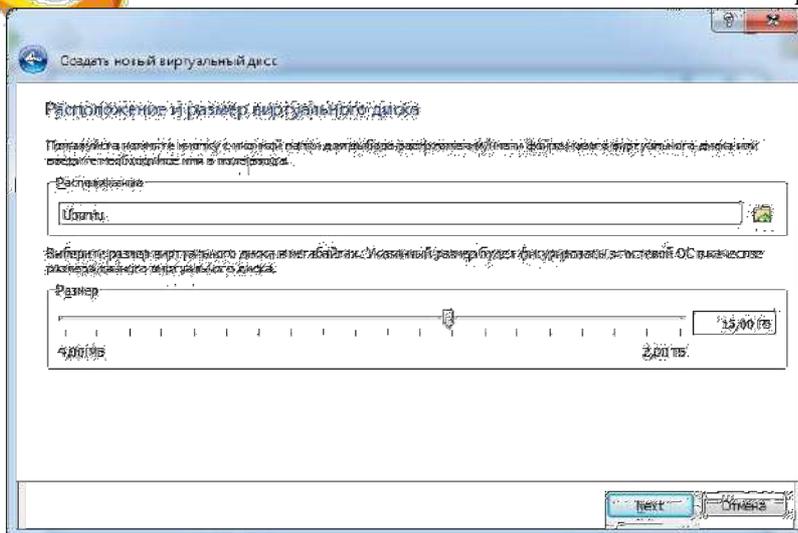
Выбрать атрибуты жесткого диска



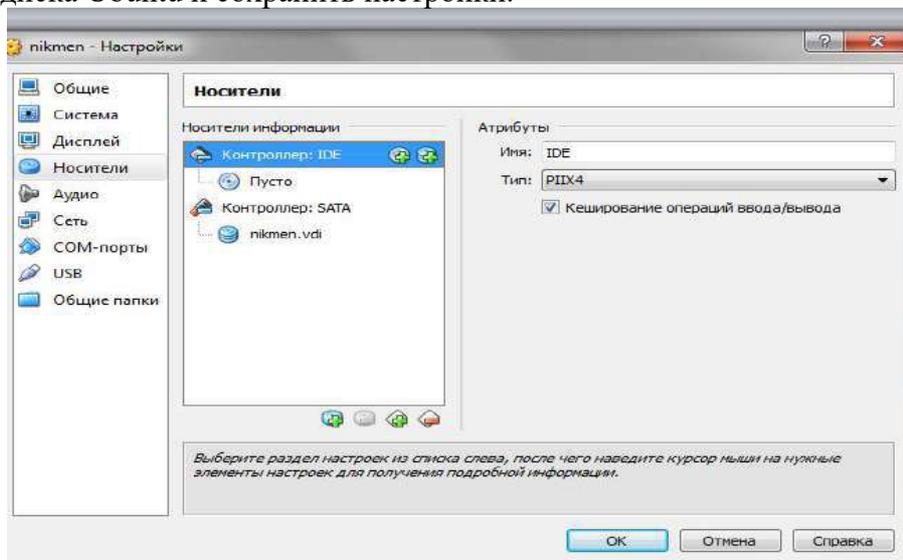
Указать размер жесткого диска (не менее 10Гб)



**Негосударственное частное образовательное
учреждение
высшего образования
«Университет»**



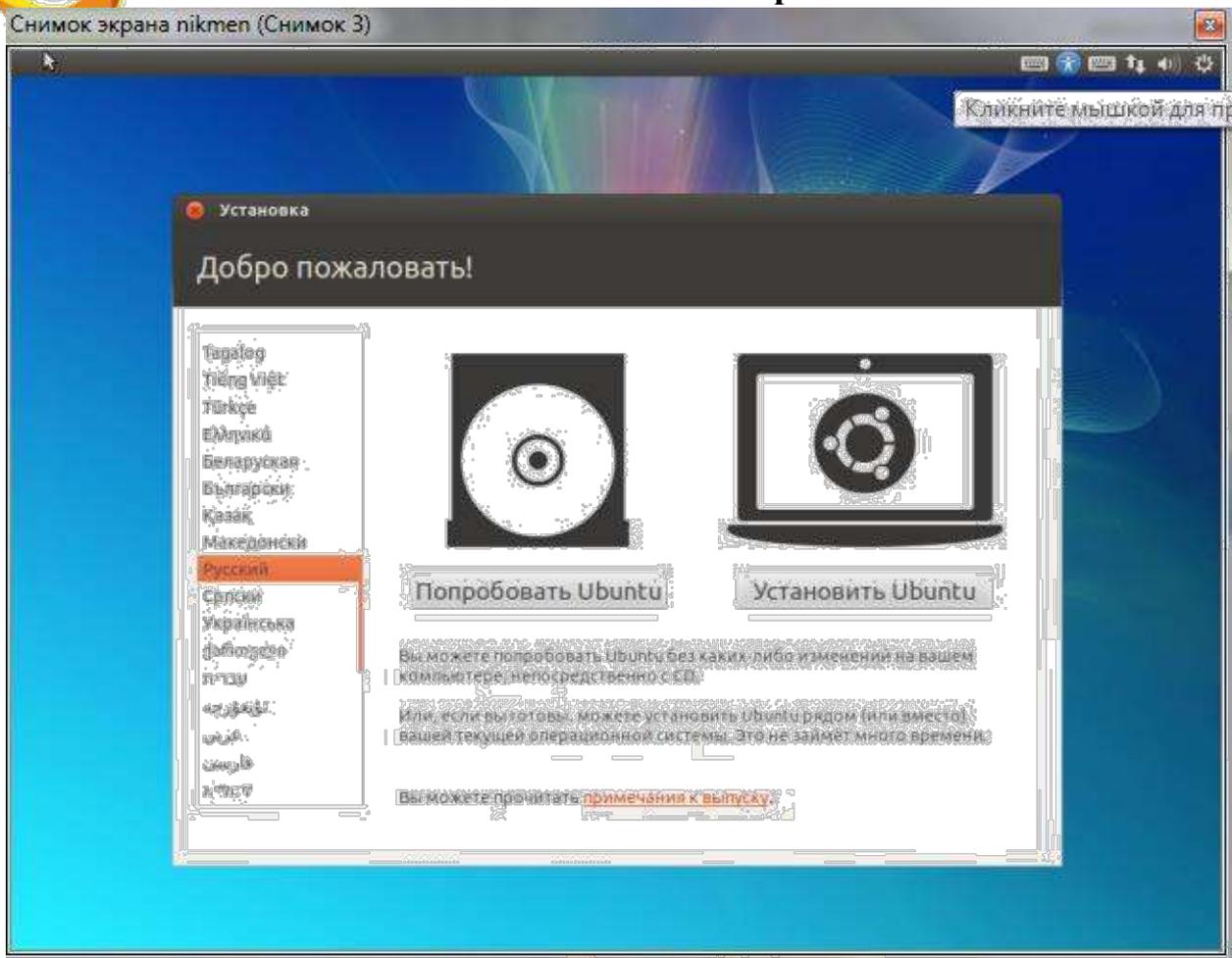
Дождаться установки виртуальной машины. Открыть окно свойств виртуальной машины и перейти в раздел носители. Выбрать IDE контроллер – пусто. Кликнуть по иконке диска в правой панели и выполнить «Выбрать образ загрузочного диска». Указать путь к образу диска Ubuntu и сохранить настройки.



2) Запустить виртуальную машину. Выбрать в списке русский язык.



Негосударственное частное образовательное учреждение высшего образования



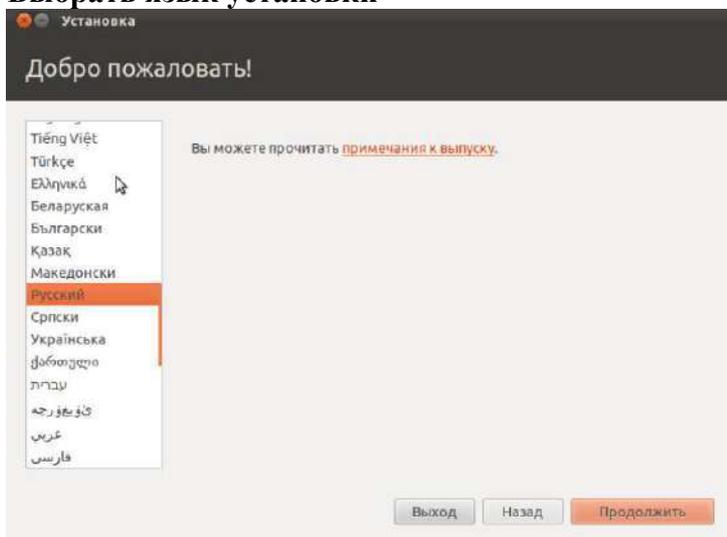
Из предложенных вариантов выбрать «Запустить Ubuntu без установки»
Дождаться загрузки операционной системы. Запустить установка кликнув по ярлыку «Установить Ubuntu»



Негосударственное частное образовательное учреждение высшего образования



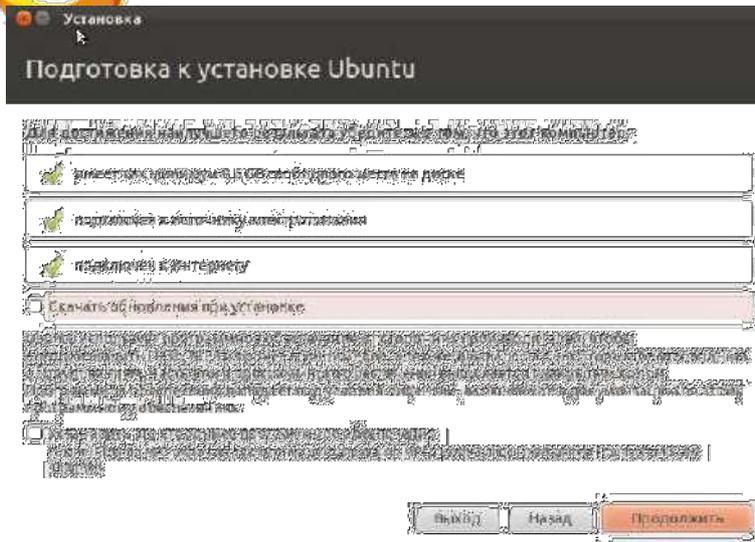
Выбрать язык установки



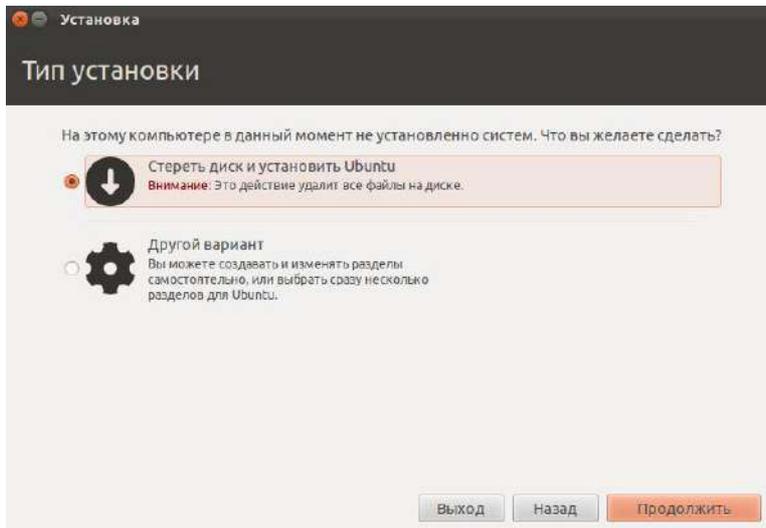
Подтвердить требования к установке



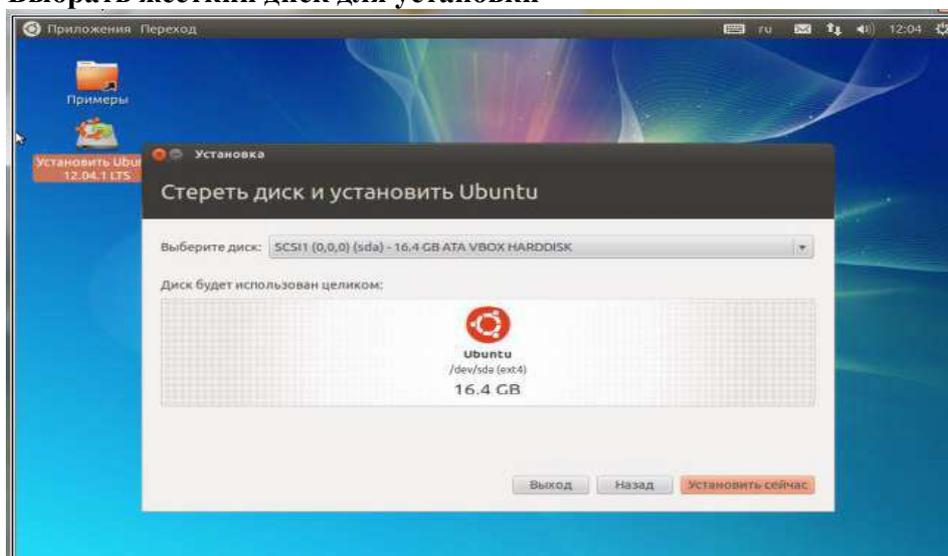
Негосударственное частное образовательное учреждение высшего образования «ИТ-Университет»



Выбрать тип установки



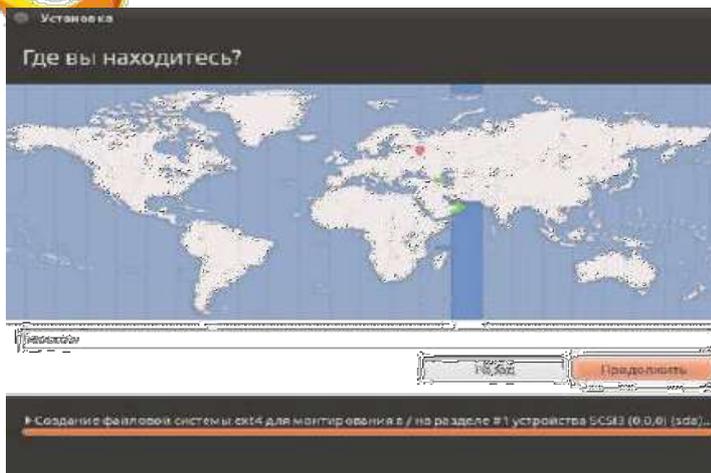
Выбрать жесткий диск для установки



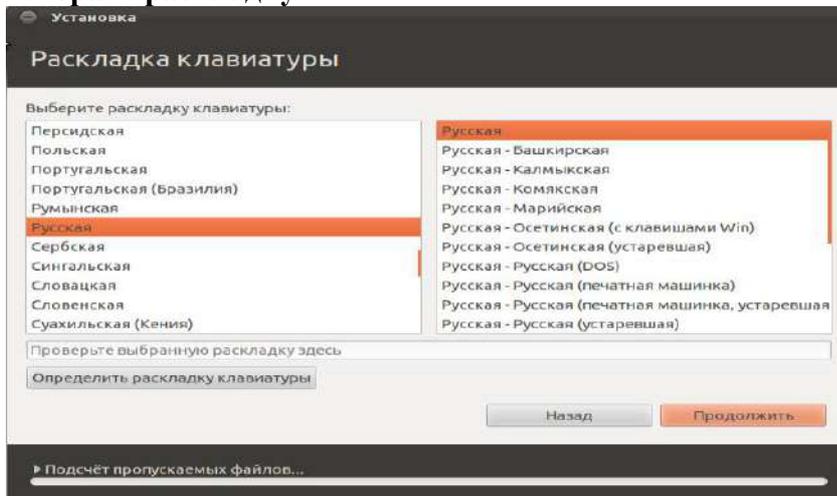
Указать региональные параметры



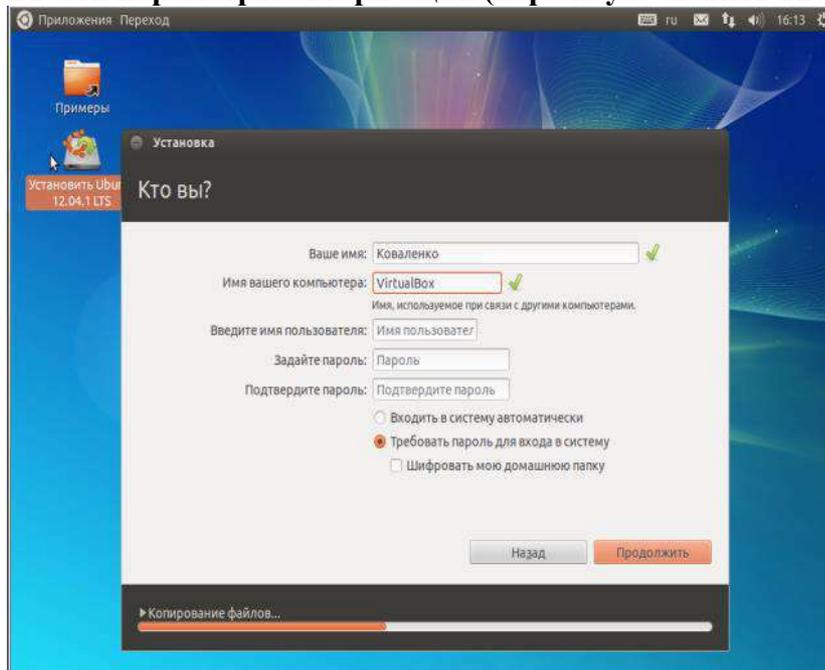
Негосударственное частное образовательное учреждение высшего образования «Юридический университет»



Выбрать раскладку

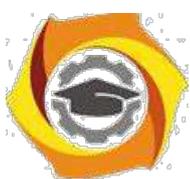


Ввести параметры авторизации (пароль установить 1234)



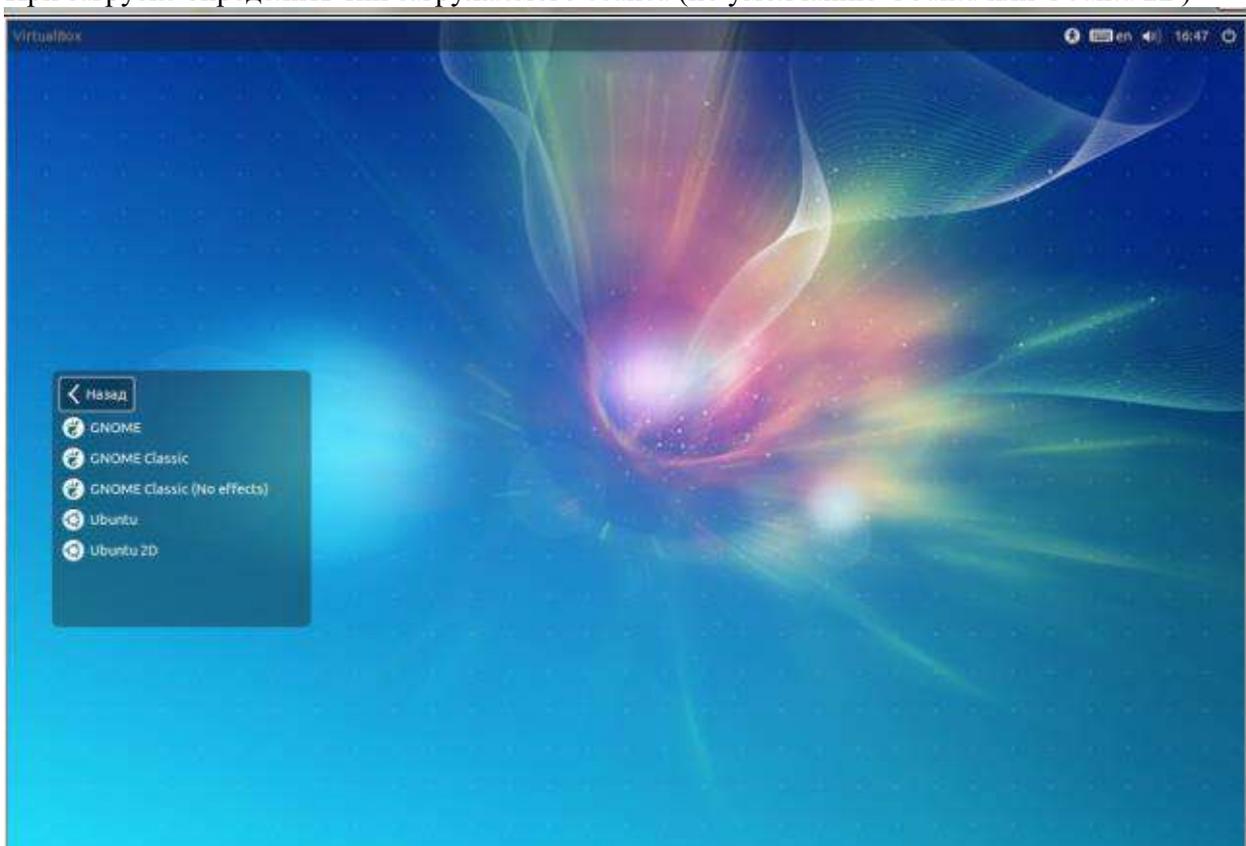
Дождаться установки операционной системы и перезагрузить виртуальную машины для входа в установленную операционную систему.

3) Установить дополнения гостевой операционной системы:



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

выбрав в меню VirtualBox «Устройства» - «Установить дополнения гостевой ОС».
Дождаться установки и перезагрузить операционную систему.
При загрузке определить тип загружаемого сеанса (по умолчанию Ubuntu или Ubuntu 2D)



Выводы:

Получен опыт в создании виртуальных машин в **Oracle VM VirtualBox** знакомство с операционной системой **Ubuntu**.



Лабораторная работа 1. Изучение протокола сетевого уровня модели OSI на примере IP

1.1 Цель работы: знать назначение и классификацию программного обеспечения вычислительных сетей, основные возможности сетевых операционных сред, уметь использовать некоторые сетевые прикладные программные пакеты для решения сетевых задач.

1.2 Указания по оформлению отчета:

Отчет должен содержать: титульный лист, цель работы; ответы на контрольные вопросы; выводы.

Указания по сдаче зачета преподавателю

Для сдачи зачета необходимо:

- 1) предъявить отчет;
- 2) ответить на контрольные вопросы.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Компьютерной сетью называют совокупность узлов (компьютеров, терминалов, периферийных устройств), имеющих возможность информационного взаимодействия друг с другом с помощью специального коммуникационного оборудования и программного обеспечения.

Средства передачи и обработки информации ориентированы в ней на коллективное использование общесетевых ресурсов – информационных, программных, аппаратных.

Компьютерные сети могут работать в различных режимах: обмена данными между абонентами сети, запроса и выдачи информации, сбора информации пакетной обработки данных по запросам пользователей с удаленных терминалов, в диалоговых режимах.

Таким образом, с появлением сетей ЭВМ разрешены две очень важные проблемы:

- 1) обеспечение в принципе неограниченного доступа к ЭВМ пользователей независимо от территориального расположения,
- 2) возможность оперативного перемещений больших массивов информации на любые расстояния, позволяющий своевременно получать данные для принятия тех или иных решений.

Использование вычислительных сетей дает предприятию следующие возможности:

1. Разделение дорогостоящих ресурсов;
2. Улучшение доступа к информации;
3. Быстрое и качественное принятие решений;



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

4. Совершенствование коммуникаций;
5. Свобода в территориальном размещении компьютеров.

Программное обеспечение сетей ЭВМ в расширенном варианте составляют:

- 1) сетевые операционные системы;
- 2) сетевые драйвера, протоколы, службы и другое дополнительное программное обеспечение сетевых интерфейсов;
- 3) прикладное сетевое программное обеспечение.

Под сетевыми операционными системами понимают такие операционные системы, которые обеспечивают пользователям распределенный доступ к сетям ЭВМ.

Во вторую группу входит большой круг всевозможного программного обеспечения в основном изготовителя данного интерфейса (сетевой платы, модема и т.п.) для обеспечения правильной работы сетевого устройства.

При этом под драйвером понимается программа, непосредственно взаимодействующая с интерфейсом - сетевым адаптером и операционной системой (ОС). Драйвер сетевого адаптера взаимодействует с ОС через систему протоколов и служб, которые могут находиться как в самих ОС, так и поставляться вместе с устройством.

При этом под сетевым протоколом понимается набор правил поведения сетевых узлов при передаче-приеме информации.

Под сетевыми службами понимается набор программного обеспечения сетевого обеспечения узкоспециального назначения, например:

- клиенты сетей - позволяют подключаться, обозревать и пользоваться сетевыми ресурсами соответствующих сетей,
- службы контроля трафика сетей,
- службы использования доступа к разделяемым ресурсам,
- доменные службы и др.

Круг прикладного сетевого программного обеспечения составляют всевозможные сетевые приложения.

Каждый компьютер работает под управлением собственной операционной системы, а какая-либо «общая» операционная система, распределяющая работу между компьютерами сети, отсутствует. Взаимодействие между компьютерами сети происходит за счет передачи сообщений через сетевые адаптеры и каналы связи. С помощью этих сообщений один компьютер обычно запрашивает доступ к локальным ресурсам другого компьютера. Такими ресурсами могут быть как данные, хранящиеся на диске, так и разнообразные периферийные устройства — принтеры, модемы, факс-аппараты и т.д. Разделение локальных ресурсов каждого компьютера между всеми пользователями сети — основная цель создания вычислительной сети.

Каким же образом сказывается на пользователе тот факт, что его компьютер подключен к сети? Прежде всего, он может пользоваться не только файлами, дисками, принтерами и другими ресурсами своего компьютера, но и аналогичными ресурсами других компьютеров, подключенных к той же сети. Правда, для этого недостаточно снабдить компьютеры сетевыми адаптерами и соединить их кабельной системой. Необходимы еще некоторые дополнения к операционным системам этих компьютеров. На тех компьютерах, ресурсы которых должны быть доступны всем пользователям сети, необходимо добавить модули, которые постоянно будут находиться в режиме ожидания запросов, поступающих по сети от других компьютеров. Обычно такие модули называются программными серверами



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

(server), так как их главная задача — обслуживать (serve) запросы на доступ к ресурсам своего компьютера. На компьютерах, пользователи которых хотят получать доступ к ресурсам других компьютеров, также нужно добавить к операционной системе некоторые специальные программные модули, которые должны вырабатывать запросы на доступ к удаленным ресурсам и передавать их по сети на нужный компьютер. Такие модули обычно называют программными клиентами (client). Собственно же сетевые адаптеры и каналы связи решают в сети достаточно простую задачу — они передают сообщения с запросами и ответами от одного компьютера к другому, а основную работу по организации совместного использования ресурсов выполняют клиентские и серверные части операционных систем.

Пара модулей «клиент – сервер» обеспечивает совместный доступ пользователей к определенному типу ресурсов, например к файлам. В этом случае говорят, что пользователь имеет дело с файловой службой (service). Обычно сетевая операционная система поддерживает несколько видов сетевых служб для своих пользователей — файловую службу, службу печати, службу электронной почты, службу удаленного доступа и т. п.

Термины «клиент» и «сервер» используются не только для обозначения про-граммных модулей, но и компьютеров, подключенных к сети. Если компьютер предоставляет свои ресурсы другим компьютерам сети, то он называется сервером, а если он их потребляет — клиентом. Иногда один и тот же компьютер может одновременно играть роли и сервера, и клиента.

Сетевые службы всегда представляют собой распределенные программы, состоящие из нескольких взаимодействующих частей, причем каждая часть, как правило, выполняется на отдельном компьютере сети.

До сих пор речь шла о системных распределенных программах. Однако в сети могут выполняться и распределенные пользовательские программы - приложения. Распределенное приложение также состоит из нескольких частей, каждая из которых выполняет какую-то определенную законченную работу по решению прикладной задачи. Например, одна часть приложения, выполняющаяся на компьютере пользователя, может поддерживать специализированный графический интерфейс, вторая - работать на мощном выделенном компьютере и заниматься статистической обработкой введенных пользователем данных, а третья - заносить полученные результаты в базу данных на компьютере с установленной стандартной СУБД. Распределенные приложения в полной мере используют потенциальные возможности распределенной обработки, предоставляемые вычислительной сетью, и поэтому часто называются сетевыми приложениями.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

1. Охарактеризовать сетевые операционные системы по следующей схеме:

- 1) платность,
- 2) доступ к исходному коду,
- 3) многоплатформенность,



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

- 4) мультизадачность,
- 5) количество пользователей,
- 6) функции управления сетью,
- 7) интерфейс работы,
- 8) потребляемые ресурсы.

2. Ответьте на контрольные вопросы

Контрольные вопросы

1. Что понимают под программным обеспечением сетей ЭВМ?
2. Что дает предприятию использование компьютерных сетей?
3. Классификация сетевого программного обеспечения.
4. Что называют операционной системой?
5. Что входит в группу прикладного программного обеспечения?
6. По каким критериям можно охарактеризовать сетевую операционную систему?
7. Что называют сетевым драйвером?
8. Что называют сетевым протоколом?
9. Перечислить сетевые операционные системы.
10. Что такое сетевые службы?
11. Что называют стандартным программным обеспечением ЭВМ?
12. Что такое технология «клиент-сервер»?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

**Практическое занятие 2 Изучение протокола транспортного
уровня модели OSI на примере TCP**

Диагностические утилиты TCP/IP

В состав TCP/IP входят диагностические утилиты, предназначенные для проверки конфигурации стека и тестирования сетевого соединения.

Утилита	Применение
hostname	Выводит имя локального хоста. Используется без параметров.
ipconfig	Выводит значения для текущей конфигурации стека TCP/IP: IP-адрес, маску подсети, адрес шлюза по умолчанию, адреса WINS (Windows Internet Naming Service) и DNS (Domain Name System)
ping	Осуществляет проверку правильности конфигурирования TCP/IP и проверку связи с удаленным хостом.
tracert	Осуществляет проверку маршрута к удаленному компьютеру путем отправки эхо-пакетов протокола ICMP (Internet Control Message Protocol). Выводит маршрут прохождения пакетов на удаленный компьютер.
arp	Выводит для просмотра и изменения таблиц трансляции адресов, используемую протоколом разрешения адресов ARP (Address Resolution Protocol - определяет локальный адрес по IP-адресу)
route	Модифицирует таблицы маршрутизации IP. Отображает содержимое таблицы, добавляет и удаляет маршруты IP.
netstat	Выводит статистику и текущую информацию по соединению TCP/IP.
nslookup	Осуществляет проверку записей и доменных псевдонимов хостов, доменных сервисов хостов, а также информации операционной системы, путем запросов к серверам DNS.

1. Проверка правильности конфигурации TCP/IP с помощью ipconfig.

При устранении неисправностей и проблем в сети TCP/IP следует сначала проверить правильность конфигурации TCP/IP. Для этого используется утилита ipconfig.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Эта команда полезна на компьютерах, работающих с DHCP (Dynamic Host Configuration Protocol), так как дает пользователям возможность определить, какая конфигурация сети TCP/IP и какие величины были установлены с помощью DHCP.

Синтаксис:

```
ipconfig [/all | /renew[adapter] | /release]
```

Параметры:

- all** выдает весь список параметров. Без этого ключа отображается только IP-адрес, маска и шлюз по умолчанию;
- renew[adapter]** обновляет параметры конфигурации DHCP для указанного сетевого адаптера;
- release[adapter]** освобождает выделенный DHCP IP-адрес;
- adapter** – имя сетевого адаптера;
- displaydns** выводит информацию о содержимом локального кэша клиента DNS, используемого для разрешения доменных имен.

Таким образом, утилита `ipconfig` позволяет выяснить, инициализирована ли конфигурация и не дублируются ли IP-адреса:

- если конфигурация инициализирована, то появляется IP-адрес, маска, шлюз;
- если IP-адреса дублируются, то маска сети будет 0.0.0.0;
- если при использовании DHCP компьютер не смог получить IP-адрес, то он будет равен 0.0.0.0 .

2. Тестирование связи с использованием утилиты ping.

Утилита `ping` (Packet Internet Grouper) используется для проверки конфигурирования TCP/IP и диагностики ошибок соединения. Она определяет доступность и функционирование конкретного хоста. Использование `ping` лучший способ проверки того, что между локальным компьютером и сетевым хостом существует маршрут. Хостом называется любое сетевое устройство (компьютер, маршрутизатор), обменивающееся информацией с другими сетевыми устройствами по TCP/IP.

Команда `ping` проверяет соединение с удаленным хостом путем отправки к этому хосту эхо-пакетов ICMP и прослушивания эхо-ответов. `Ping` ожидает каждый посланный пакет и печатает количество переданных и принятых пакетов. Каждый принятый пакет проверяется в соответствии с переданным сообщением. Если связь между хостами плохая, из сообщений `ping` станет ясно, сколько пакетов потеряно.

По умолчанию передается 4 эхо-пакета длиной 32 байта (возможны и другие варианты значения по умолчанию) - периодическая последовательность символов алфавита в верхнем



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

регистре. Ping позволяет изменить размер и количество пакетов, указать, следует ли записывать маршрут, который она использует, какую величину времени жизни (ttl) устанавливать, можно ли фрагментировать пакет и т.д.. При получении ответа в поле time указывается, за какое время (в миллисекундах) отправленный пакет доходит до удаленного хоста и возвращается назад. Так как значение по умолчанию для ожидания отклика равно 1 секунде, то все значения данного поля будут меньше 1000 миллисекунд. Если вы получаете сообщение «Request time out» (Превышен интервал ожидания), то, возможно, если увеличить время ожидания отклика, пакет дойдет до удаленного хоста. Это можно сделать с помощью ключа -w.

Ping можно использовать для тестирования как имени хоста (DNS или NetBIOS), так и его IP-адреса. Если ping с IP-адресом выполнялась успешно, а с именем – неудачно, это значит, что проблема заключается в распознавании соответствия адреса и имени, а не в сетевом соединении.

Утилита ping используется следующими способами:

1) Для проверки того, что TCP/IP установлен и правильно сконфигурирован на локальном компьютере, в команде ping задается адрес петли обратной связи (loopback address):

```
ping 127.0.0.1
```

Если тест успешно пройден, то вы получите следующий

```
ответ: Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128
```

```
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128 Ответ
```

```
от 127.0.0.1: число байт=32 время<1мс TTL=128 Ответ от
```

```
127.0.0.1: число байт=32 время<1мс TTL=128
```

2) Чтобы убедиться в том, что компьютер правильно добавлен в сеть и IP-адрес не дублируется, используется IP-адрес локального компьютера:

```
ping IP-адрес_локального_хоста
```

3) Чтобы проверить, что шлюз по умолчанию функционирует и что можно установить соединение с любым локальным хостом в локальной сети, задается IP-адрес шлюза по умолчанию:

```
ping IP-адрес_шлюза
```

4) Для проверки возможности установления соединения через маршрутизатор в команде ping задается IP-адрес удаленного хоста:

```
ping IP-адрес_удаленного_хоста
```

Синтаксис:



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

```
ping [-t] [-a] [-n count] [-l length] [-f] [-i ttl] [-v tos] [-r count] [-s count] [ [-j host-list]
| [-k host-list] ] [-w timeout] destination-list
```

Параметры:

- t выполняет команду ping до прерывания. Control-Break - посмотреть статистику и продолжить. Control-C - прервать выполнение команды;
 - a позволяет определить доменное имя удаленного компьютера по его IP-адресу; -n count посылает количество пакетов ECHO, указанное параметром count;
 - l length посылает пакеты длиной length байт (максимальная длина 8192 байта);
 - f посылает пакет с установленным флагом «не фрагментировать». Этот пакет не будет фрагментироваться на маршрутизаторах по пути своего следования;
 - i ttl устанавливает время жизни пакета в величину ttl (каждый маршрутизатор уменьшает ttl на единицу);
 - v tos устанавливает тип поля «сервис» в величину tos;
 - r count записывает путь выходящего пакета и возвращающегося пакета в поле записи пути. Count - от 1 до 9 хостов;
 - s count позволяет ограничить количество переходов из одной подсети в другую (хопов). Count задает максимально возможное количество хопов;
 - j host-list направляет пакеты с помощью списка хостов, определенного параметром host-list. Последовательные хосты могут быть отделены промежуточными маршрутизаторами (гибкая статическая маршрутизация). Максимальное количество хостов в списке, дозволенное IP, равно 9;
 - k host-list направляет пакеты через список хостов, определенный в host-list. Последовательные хосты не могут быть разделены промежуточными маршрутизаторами (жесткая статическая маршрутизация). Максимальное количество хостов – 9;
 - w timeout указывает время ожидания (timeout) ответа от удаленного хоста в миллисекундах (по умолчанию – 1сек);
- destination-list указывает удаленный хост, к которому надо направить пакеты ping.

Пример использования утилиты ping:

```
C:\WINDOWS>ping -n 10 www.netscape.com
```

Обмен пакетами с www.netscape.com [205.188.247.65] по 32 байт:

Ответ от 205.188.247.65: число байт=32 время=194мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=240мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=173мс TTL=48



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Ответ от 205.188.247.65: число байт=32 время=250мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=187мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=239мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=263мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=230мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=185мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=406мс TTL=48

Статистика Ping для 205.188.247.65:

Пакетов: послано = 10, получено = 10, потеряно = 0 (0% потерь)

Приблизительное время передачи и приема:

Наименьшее = 173мс, наибольшее = 406мс, среднее =236мс

В случае невозможности проверить доступность хоста утилита выводит информацию об ошибке. Ниже приведен пример ответа утилиты ping при попытке послать запрос на несуществующий хост.

Обмен пакетами с 172.16.6.21 по 32 байт:

Превышен интервал ожидания для запроса.

Статистика Ping для 172.16.6.21:

Пакетов: отправлено = 4, получено = 0, потеряно = 4 (100% потерь),

Приблизительное время передачи и приема:

наименьшее = 0мс, наибольшее = 0мс, среднее = 0мс

Утилита сообщает не об отсутствии хоста, а о том, что за отведенное время не был получен ответ на посланный запрос. Причиной этого не обязательно является отсутствие хоста в сети. Проблема может крыться в сбоях связи, перегрузке или неправильной настройке маршрутизаторов и т. п. Ошибка «сеть недоступна» (network unreachable) прямо указывает на проблемы маршрутизации.



3. Изучение маршрута между сетевыми соединениями с помощью утилиты `tracert`.

Tracert - это утилита трассировки маршрута. Она использует поле TTL (time-to-live, время жизни) пакета IP и сообщения об ошибках ICMP для определения маршрута от одного хоста до другого.

Утилита `tracert` может быть более содержательной и удобной, чем `ping`, особенно в тех случаях, когда удаленный хост недостижим. С помощью нее можно определить район проблем со связью (у Internet-провайдера, в опорной сети, в сети удаленного хоста) по тому, насколько далеко будет отслежен маршрут. Если возникли проблемы, то утилита выводит на экран звездочки (*), либо сообщения типа «Destination net unreachable», «Destination host unreachable», «Request time out», «Time Exceeded».

Утилита `tracert` работает следующим образом: посылаются по 3 пробных эхо-пакета на каждый хост, через который проходит маршрут до удаленного хоста. На экран при этом выводится время ожидания ответа на каждый пакет (Его можно изменить с помощью параметра `-w`). Пакеты посылаются с различными величинами времени жизни. Каждый маршрутизатор, встречающийся по пути, перед перенаправлением пакета уменьшает величину TTL на единицу. Таким образом, время жизни является счетчиком точек промежуточной доставки (хопов). Когда время жизни пакета достигнет нуля, предполагается, что маршрутизатор пошлет в компьютер-источник сообщение ICMP “Time Exceeded” (Время истекло). Маршрут определяется путем отправки первого эхо-пакета с TTL=1. Затем TTL увеличивается на 1 в каждом последующем пакете до тех пор, пока пакет не достигнет удаленного хоста, либо будет достигнута максимально возможная величина TTL (по умолчанию 30, задается с помощью параметра `-h`).

Маршрут определяется путем изучения сообщений ICMP, которые присылаются обратно промежуточными маршрутизаторами.

Примечание: некоторые маршрутизаторы просто уничтожают пакеты с истекшим TTL и не будут видны утилите `tracert`.

Синтаксис:

```
tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] имя_целевого_хоста
```

Параметры:

`-d` указывает, что не нужно распознавать адреса для имен хостов;

`-h maximum_hops` указывает максимальное число хопов для того, чтобы искать цель;



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- j host-list указывает нежесткую статическую маршрутизацию в соответствии с host-list;
- w timeout указывает, что нужно ожидать ответ на каждый эхо-пакет заданное число мсек.

4. Утилита *arp*.

Основная задача протокола ARP – трансляция IP-адресов в соответствующие локальные адреса. Для этого ARP-протокол использует информацию из ARP-таблицы (ARP-кэша). Если необходимая запись в таблице не найдена, то протокол ARP отправляет широковещательный запрос ко всем компьютерам локальной подсети, пытаясь найти владельца данного IP-адреса. В кэше могут содержаться два типа записей: статические и динамические. Статические записи вводятся вручную и хранятся в кэше постоянно. Динамические записи помещаются в кэш в результате выполнения широковещательных запросов. Для них существует понятие времени жизни. Если в течение определенного времени (по умолчанию 2 мин.) запись не была востребована, то она удаляется из кэша.

Синтаксис:

```
arp [-s inet_addr eth_addr] | [-d inet_addr] | [-a]
```

Параметры:

- s занесение в кэш статических записей;
- d удаление из кэша записи для определенного IP-адреса;
- a просмотр содержимого кэша для всех сетевых адаптеров локального компьютера;
- inet_addr - IP-адрес;
- eth_addr - MAC-адрес.

5. Утилита *route*.

Утилита **route** предназначена для работы с локальной таблицей маршрутизации. Она имеет следующий **синтаксис**:

```
route [-f] [-p] [команда [узел] [MASK маска] [шлюз] [METRIC метрика] [IF интерфейс]]
```

Параметры:

- f Очистка таблицы маршрутизации.
- p При указании совместно с командой ADD создает постоянную запись, которая сохраняется после перезагрузки компьютера. По умолчанию записи таблицы маршрутов не сохраняются при перезагрузке.
- команда одна из четырех команд:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

PRINT - вывод информации о маршруте;

ADD - добавление маршрута;

DELETE - удаление маршрута;

CHANGE - изменение маршрута.

<i>узел</i>	адресуемый узел
<i>маска</i>	маска подсети; по умолчанию используется маска 255.255.255.255
<i>шлюз</i>	адрес шлюза
<i>метрика</i>	метрика маршрута;
<i>интерфейс</i>	идентификатор интерфейса, который будет использован для пересылки пакета

Для команд PRINT и DELETE возможно использование символов подстановки при указании адресуемого узла или шлюза. Параметр шлюза для этих команд может быть опущен. При добавлении и изменении маршрутов утилита route осуществляет проверку введенной информации на соответствие условию (УЗЕЛ & МАСКА) == УЗЕЛ. Если это условие не выполняется, то утилита выдает сообщение об ошибке и не добавляет или не изменяет маршрут.

Утилита осуществляет поиск имен сетей в файле networks. Поиск имен шлюзов осуществляется в файле hosts. Оба файла расположены в папке %systemroot%\system32\drivers\etc. Наличие и заполнение этих файлов не обязательно для нормального функционирования утилиты route и работы маршрутизации.

Хотя в большинстве случаев на рабочей станции это не требуется, можно вручную редактировать таблицы маршрутизации.

Пример использования утилиты route:

Добавление статического маршрута:

```
route add 172.16.6.0 MASK 255.255.255.0 172.16.11.1 METRIC 1 IF 0x1000003
```

6. Утилита netstat.

Утилита netstat позволяет получить статическую информацию по некоторым из протоколов стека (TCP, UDP, IP, ICMP), а также выводит сведения о текущих сетевых соединениях. Особенно она полезна на брандмауэрах, с ее помощью можно обнаружить нарушения безопасности периметра сети.

Синтаксис:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

netstat [-a] [-e] [-n] [-s] [-p protocol] [-r]

Параметры:

- a выводит перечень всех сетевых соединений и прослушивающихся портов локального компьютера;
- e выводит статистику для Ethernet-интерфейсов (например, количество полученных и отправленных байт);
- n выводит информацию по всем текущим соединениям (например, TCP) для всех сетевых интерфейсов локального компьютера. Для каждого соединения выводится информация об IP-адресах локального и удаленного интерфейсов вместе с номерами используемых портов;
- s выводит статистическую информацию для протоколов UDP, TCP, ICMP, IP. Ключ «/more» позволяет просмотреть информацию постранично;
- r выводит содержимое таблицы маршрутизации.

7. Утилита nslookup.

Утилита **nslookup** предназначена для диагностики службы DNS, в простейшем случае - для выполнения запросов к DNS-серверам на разрешение имен в IP-адреса. В общем случае утилита позволяет просмотреть любые записи DNS-сервера:

A – каноническое имя узла, устанавливает соответствие доменного имени ip-адресу. *SOA* – начало полномочий, начальная запись, единственная для зоны;

MX – почтовые серверы (хосты, принимающие почту для заданного домена);

NS – серверы имен (содержит авторитетные DNS-серверы для зоны);

PTR – указатель (служит для обратного преобразования ip-адреса в символьное имя хоста)

ит.д.

Утилита nslookup достаточно сложна и содержит свой собственный командный интерпретатор.

В простейшем случае (без входа в командный режим) утилита **nslookup** имеет следующий

Синтаксис:

nslookup хост [сервер]

Параметры:

Хост DNS-имя хоста, которое должно быть преобразовано в IP-адрес.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Сервер Адрес DNS-сервера, который будет использоваться для разрешения имени. Если этот параметр опущен, то будут последовательно использованы адреса DNS-серверов из параметров настройки протокола TCP/IP.

Примеры использования утилиты nslookup:

1. Получение списка серверов имен для домена yandex.ru без входа в командный режим (с использованием ключей).

```
yandex.ru nameserver = ns4.yandex.ru  
yandex.ru nameserver = ns5.yandex.ru  
yandex.ru nameserver = ns2.yandex.ru  
yandex.ru nameserver = ns1.yandex.ru
```

```
ns2.yandex.ru internet address = 213.180.199.34
```

```
ns5.yandex.ru internet address = 213.180.204.1
```

2. Получение записи SOA домена yandex.ru с авторитетного сервера с использованием командного интерпретатора nslookup.

```
> set type=SOA  
> server ns2.yandex.ru  
Default Server: ns2.yandex.ru  
Address: 213.180.199.34  
> yandex.ru  
Server: ns1.yandex.ru  
Address: 213.180.193.1  
>yandex.ru  
primary name server = ns1.yandex.ru  
responsible mail addr = sysadmin.yandex-team.r  
serial = 2009022707
```



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

refresh = 1800 (30 mins)

retry = 900 (15 mins)

expire = 2592000 (30 days)

default TTL = 900 (15 mins)

yandex.ru nameserver = ns5.yandex.ru

yandex.ru nameserver = ns1.yandex.ru

yandex.ru nameserver = ns4.yandex.ru

yandex.ru nameserver = ns2.yandex.ru

ns1.yandex.ru internet address = 213.180.193.1

ns2.yandex.ru internet address = 213.180.199.34

ns4.yandex.ru internet address = 77.88.19.60

ns5.yandex.ru internet address = 213.180.204.1

> exit

3. Получение адреса почтового сервера для домена yandex.ru.

> set q=mx

> yandex.ru

yandex.ru MX preference = 10, mail exchanger = mx2.yandex.ru

yandex.ru MX preference = 10, mail exchanger = mx3.yandex.ru

yandex.ru MX preference = 10, mail exchanger = mx1.yandex.ru

yandex.ru nameserver = ns2.yandex.ru

yandex.ru nameserver = ns1.yandex.ru

yandex.ru nameserver = ns4.yandex.ru

yandex.ru nameserver = ns5.yandex.ru

mx1.yandex.ru internet address = 77.88.21.89

mx2.yandex.ru internet address = 93.158.134.89

mx3.yandex.ru internet address = 213.180.204.89

ns2.yandex.ru internet address = 213.180.199.34



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

ns4.yandex.ru internet address = 77.88.19.60



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ns5.yandex.ru internet address = 213.180.204.1

>

Указав ключ `type=any`, можно получить все записи о узле или домене. Ключи `querytype, t, q` эквивалентны `type`.

Задания на лабораторную работу

1. Изучите методические указания к лабораторной работе.
2. Выполните упражнения.
3. Оформите отчет по лабораторной работе, описав выполнение упражнений и дав краткие ответы на контрольные вопросы.

Упражнение 1. Получение справочной информации по командам.

Выведите на экран справочную информацию по всем рассмотренным утилитам (см. таблицу п.1). Для этого в командной строке введите имя утилиты без параметров и дополните `/?`.

Сохраните справочную информацию в отдельном файле.

Изучите ключи, используемые при запуске утилит.

Упражнение 2. Получение имени хоста.

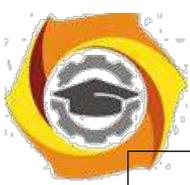
Выведите на экран имя локального хоста с помощью команды `hostname`.

Сохраните результат в отдельном файле.

Упражнение 3. Изучение утилиты ipconfig.

Проверьте конфигурацию TCP/IP с помощью утилиты `ipconfig`. Заполните таблицу:

Имя хоста	
IP-адрес	
Маска подсети	
Основной шлюз	
Используется ли DHCP (адрес DHCP-сервера)	
Описание адаптера	
Физический адрес сетевого адаптера	
Адрес DNS-сервера	



Адрес WINS-сервера

Упражнение 4. Тестирование связи с помощью утилиты ping.

1. Проверьте правильность установки и конфигурирования TCP/IP на локальном компьютере.
2. Проверьте функционирование основного шлюза, послав 5 эхо-пакетов длиной 64 байта.
3. Проверьте возможность установления соединения с удаленным хостом.
4. С помощью команды ping проверьте адреса (взять из списка локальных ресурсов на сайте aspu.ru) и для каждого из них отметьте время отклика. Попробуйте изменить параметры команды ping таким образом, чтобы увеличилось время отклика. Определите IP-адреса узлов.

Упражнение 5. Определение пути IP-пакета.

С помощью команды traceroute проверьте для перечисленных ниже адресов, через какие промежуточные узлы идет сигнал. Изучите ключи команды.

- a) aspu.ru
- b) mathmod.aspu.ru
- c) yarus.aspu.ru

Упражнение 6: Просмотр ARP-кэша.

С помощью утилиты arp просмотрите ARP-таблицу локального компьютера. Внести в кэш локального компьютера любую статическую запись.

Упражнение 7: Просмотр локальной таблицы маршрутизации.

С помощью утилиты route просмотреть локальную таблицу маршрутизации.

Упражнение 8. Получение информации о текущих сетевых соединениях и протоколах стека TCP/IP.

С помощью утилиты netstat выведите перечень сетевых соединений и статистическую информацию для протоколов UDP, TCP, ICMP, IP.

Упражнение 9. Получение DNS-информации с помощью nslookup.

- 1) Узнайте ip-адреса узлов, сайтов ростовских предприятий.
- 2) Узнайте авторитетные (компетентные) сервера для этих узлов.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

3) Получите запись SOA с одного из этих серверов для домена выбранного вами домена.

Контрольные вопросы

1. Раскрыть термины: хост, шлюз, хоп, время жизни пакета, маршрут, маска сети, авторитетный/неавторитетный (компетентный) DNS-сервер, порт TCP, петля обратной связи, время отклика.
2. Какие утилиты можно использовать для проверки правильности конфигурирования TCP/IP?
3. Каким образом команда ping проверяет соединение с удаленным хостом?
4. Каково назначение протокола ARP?
5. Как утилита ping разрешает имена узлов в ip-адреса (и наоборот)?
6. Какие могут быть причины неудачного завершения ping и tracer? (превышен интервал ожидания для запроса, сеть недоступна, превышен срок жизни при передаче пакета).
7. Всегда ли можно узнать символическое имя узла по его ip-адресу?
8. Какой тип записи запрашивает у DNS-сервера простейшая форма nslookup?



Лабораторная работа 2 Изучение протокола прикладного уровня модели OSI на примере HTTP

MAC-адресация

MAC-адрес (от англ. Media Access Control — управление доступом к среде) — это уникальный идентификатор, сопоставляемый с различными типами оборудования для компьютерных сетей. Большинство сетевых протоколов канального уровня используют одно из трёх пространств MAC-адресов, управляемых IEEE: MAC-48, EUI-48 и EUI-64. Адреса в каждом из пространств теоретически должны быть глобально уникальными. Не все протоколы используют MAC-адреса, и не все протоколы, использующие MAC-адреса, нуждаются в подобной уникальности этих адресов.

В широковещательных сетях (таких, как сети на основе Ethernet) MAC-адрес позволяет уникально идентифицировать каждый узел сети и доставлять данные только этому узлу. Таким образом, MAC-адреса формируют основу сетей на канальном уровне, которую используют протоколы более высокого (сетевого) уровня. Для преобразования MAC-адресов в адреса сетевого уровня и обратно применяются специальные протоколы (например, ARP и RARP в сетях TCP/IP).

Адреса типа MAC-48 наиболее распространены; они используются в таких технологиях, как Ethernet, Token ring, FDDI, WiMAX и др. Они состоят из 48 бит, таким образом, адресное пространство MAC-48 насчитывает 248 (или 281 474 976 710 656) адресов. Согласно подсчётам IEEE, этого запаса адресов хватит по меньшей мере до 2100 года.

EUI-48 от MAC-48 отличается лишь семантически: в то время как MAC-48 используется для сетевого оборудования, EUI-48 применяется для других типов аппаратного и программного обеспечения.

Идентификаторы EUI-64 состоят из 64 бит и используются в FireWire, а также в IPv6 в качестве младших 64 бит сетевого адреса узла.

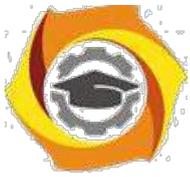
Структура MAC-адреса



Стандарты IEEE определяют 48-разрядный (6 октетов) MAC-адрес, который разделен на четыре части.

Первые 3 октета (в порядке их передачи по сети; старшие 3 октета, если рассматривать их в традиционной бит-реверсной шестнадцатичной записи MAC-адресов) содержат 24-битный уникальный идентификатор организации (OUI)[1], или (Код MFG - Manufacturing, производителя), который производитель получает в IEEE. При этом используются только младшие 22 разряда (бита), 2 старшие имеют специальное назначение:

первый бит указывает, для одиночного (0) или группового (1) адресата предназначен кадр



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

следующий бит указывает, является ли MAC-адрес глобально (0) или локально (1) администрируемым.

Следующие три октета выбираются изготовителем для каждого экземпляра устройства. За исключением сетей системной сетевой архитектуры SNA.

Таким образом, глобально администрируемый MAC-адрес устройства глобально уникален и обычно «защит» в аппаратуру.

Администратор сети имеет возможность, вместо использования «защитого», назначить устройству MAC-адрес по своему усмотрению. Такой локально администрируемый MAC-адрес выбирается произвольно и может не содержать информации об OUI. Признаком локально администрируемого адреса является соответствующий бит первого октета адреса

Среди людей, плохо разбирающихся в сетях, существует распространенное мнение, что MAC-адрес жестко вшит в сетевую карту и сменить его нельзя или можно только с помощью программаторов. На самом деле, это не так. MAC-адрес легко меняется программным путем, так как значение, указанное через драйвер, имеет более высокий приоритет, чем зашитый в плату. Поскольку многие DHCP-серверы, которые раздают динамические IP, обычно делают привязку по MAC-адресу (то есть при неизменном MAC они будут выдавать одинаковый IP), то смена MAC-адреса через драйвер поможет сменить и локальный IP адрес. Постоянно изменяя свой MAC-адрес, пользователь становится почти неуязвимым для администратора сети, поскольку для идентификации компьютера можно использовать только IP, MAC-адрес и сетевое имя. Если эти три параметра будут меняться, то становится невозможно определить, разные это клиенты или один и тот же. При этом администратор может жестко привязать выдаваемые IP адреса к списку определенных MAC-адресов пользователей — таким образом, DHCP может отказать устройству с неизвестным MAC в выдаче адреса.

Типы адресов стека TCP/IP

В стеке TCP/IP используются три типа адресов: локальные (называемые также аппаратными), IP-адреса и символьные доменные имена.

В терминологии TCP/IP под локальным адресом понимается такой тип адреса, который используется средствами базовой технологии для доставки данных в пределах подсети, являющейся элементом составной интерсети. В разных подсетях допустимы разные сетевые технологии, разные стеки протоколов, поэтому при создании стека TCP/IP предполагалось наличие разных типов локальных адресов. Если подсетью интерсети является локальная сеть, то локальный адрес - это MAC-адрес. Однако протокол IP может работать и над протоколами более высокого уровня, например над протоколом IPX или X.25.

В этом случае локальными адресами для протокола IP соответственно будут адреса IPX и X.25. Компьютер в локальной сети может иметь несколько локальных адресов даже при одном сетевом адаптере. Некоторые сетевые устройства не имеют локальных адресов. Например, к таким устройствам относятся глобальные порты маршрутизаторов, предназначенные для соединений типа «точка-точка».

IP-адреса представляют собой основной тип адресов, на основании которых сетевой уровень передает пакеты между сетями. Эти адреса состоят из 4 байт, например 109.26.17.100. IP-адрес назначается администратором во время конфигурирования компьютеров и маршрутизаторов. IP-адрес состоит из двух частей: номера сети и номера



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

узла. Номер сети может быть выбран администратором произвольно, либо назначен по рекомендации специального подразделения Internet (Internet Network Information Center, InterNIC), если сеть должна работать как составная часть Internet. Номер узла в протоколе IP назначается независимо от локального адреса узла. Маршрутизатор по определению входит сразу в несколько сетей. Поэтому каждый порт маршрутизатора имеет собственный IP-адрес. Конечный узел также может входить в несколько IP-сетей. В этом случае компьютер должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

Символьные доменные имена. Символьные имена в IP-сетях называются доменными и строятся по иерархическому признаку. Между доменным именем и IP-адресом узла нет никакого алгоритмического соответствия, поэтому необходимо использовать какие-то дополнительные таблицы или службы, чтобы узел сети однозначно определялся как по доменному имени, так и по IP-адресу. В сетях TCP/IP используется специальная распределенная служба Domain Name System (DNS), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называют также DNS-именами.

1.2. Классы IP-адресов

IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например:

128.10.2.30 - традиционная десятичная форма представления адреса;

10000000 00001010 00000010 00011110 - двоичная форма представления этого же адреса.

Адрес состоит из двух логических частей - номера сети и номера узла в сети. Какая часть адреса относится к номеру сети, а какая - к номеру узла, определяется значениями первых бит адреса. Значения этих бит являются также признаками того, к какому классу относится тот или иной IP-адрес. На рисунке 1 показана структура IP-адресов различных классов.

Класс А. 0ccccccc уuuuuuuу уuuuuuuу уuuuuuuу
Класс В. 10cccccc cccccccc уuuuuuuу уuuuuuuу
Класс С. 110ccccс cccccccc cccccccc уuuuuuuу
Класс D. 1110aaaa aaaaaaaa aaaaaaaa aaaaaaaa
Класс E. 11110zzz zzzzzzzz zzzzzzzz zzzzzzzz

Рисунок 1 – Классы IP-адресов (с - бит, входящий в номер сети; у - бит, входящий в номер узла; а - бит, входящий в адрес группы multicast; з - бит, входящий в зарезервированный адрес)

Если адрес начинается с 0, то сеть относят к классу А и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса А имеют номера в диапазоне от 1 до 126. (Номер 0 не используется, а номер 127 зарезервирован для специальных целей, о чем будет сказано ниже.) Количество узлов в сетях класса А может достигать 2^{24} , то есть 16 777 216 узлов.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Если первые два бита адреса равны 10, то сеть относится к классу В. В сетях класса В под номер сети и под номер узла отводится по 16 бит. Таким образом, сеть класса В является сетью средних размеров с максимальным числом узлов 2^{16} , что составляет 65 536 узлов.

Если адрес начинается с последовательности 110, то это сеть класса С. В этом случае под номер сети отводится 24 бита, а под номер узла - 8 бит. Сети этого класса наиболее распространены, число узлов в них ограничено 2^8 , то есть 256 узлами.

Если адрес начинается с последовательности 1110, то он является адресом класса D и обозначает особый, групповой адрес - multicast. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.

Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к классу E. Адреса этого класса зарезервированы для будущих применений.

1.3. Особые IP-адреса

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов:

- Если весь IP-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет; этот режим используется только в некоторых сообщениях ICMP.

- Если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет.

- Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такая рассылка называется ограниченным широковещательным сообщением (limited broadcast).

- Если в поле номера узла назначения стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным номером сети. Например, пакет с адресом 192.190.21.255 доставляется всем узлам сети 192.190.21.0. Такая рассылка называется широковещательным сообщением (broadcast).

При адресации необходимо учитывать те ограничения, которые вносятся особым назначением некоторых IP-адресов. Так, ни номер сети, ни номер узла не может состоять только из одних двоичных единиц или только из одних двоичных нулей. Отсюда следует, что максимальное количество узлов, приведенное для сетей каждого класса, на практике должно быть уменьшено на 2. Например, в сетях класса С под номер узла отводится 8 бит, которые позволяют задавать 256 номеров: от 0 до 255. Однако на практике максимальное число узлов в сети класса С не может превышать 254, так как адреса 0 и 255 имеют специальное назначение. Из этих же соображений следует, что конечный узел не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса А состоит из одних двоичных единиц.

Особый смысл имеет IP-адрес, первый октет которого равен 127. Он используется для тестирования программ и взаимодействия процессов в пределах одной машины. Когда программа посылает данные по IP-адресу 127.0.0.1, то образуется как бы «петля». Данные не передаются по сети, а возвращаются модулям верхнего уровня как только что принятые. Поэтому в IP-сети запрещается присваивать машинам IP-адреса, начинающиеся со 127. Этот адрес имеет название loopback.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

В протоколе IP нет понятия широковещательности в том смысле, в котором оно используется в протоколах канального уровня локальных сетей, когда данные должны быть доставлены абсолютно всем узлам. Как ограниченный широковещательный IP-адрес, так и широковещательный IP-адрес имеют пределы распространения в интрасети - они ограничены либо сетью, к которой принадлежит узел-источник пакета, либо сетью, номер которой указан в адресе назначения.

Уже упоминавшаяся форма группового IP-адреса - multicast - означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса. Узлы сами идентифицируют себя, то есть определяют, к какой из групп они относятся. Один и тот же узел может входить в несколько групп. Члены какой-либо группы multicast не обязательно должны принадлежать одной сети. Групповой адрес не делится на поля номера сети и узла и обрабатывается маршрутизатором особым образом.

Групповая адресация предназначена для экономичного распространения в Internet или большой корпоративной сети аудио- или видеопрограмм, предназначенных сразу большой аудитории слушателей или зрителей. Если такие средства найдут широкое применение, то Internet сможет создать серьезную конкуренцию радио и телевидению.

1.4. Использование масок в IP-адресации

Важным элементом разбиения адресного пространства Internet являются подсети. Подсеть – это подмножество сети, не пересекающееся с другими подсетями. Это означает, что сеть организации может быть разбита на фрагменты, каждый из которых будет составлять подсеть. Реально, каждая подсеть соответствует физической локальной сети (например, сегменту Ethernet). Подсети используются для того, чтобы обойти ограничения физических сетей на число узлов в них и максимальную длину кабеля в сегменте сети. Например, сегмент тонкого Ethernet имеет максимальную длину 185 м и может включать до 32 узлов. Самая маленькая сеть класса C может состоять из 254 узлов. Для того, чтобы достичь этой цифры надо объединить несколько физических сегментов сети. Сделать это можно либо с помощью физических устройств (например, репитеров), либо при помощи машин-шлюзов. В первом случае разбиение на подсети не требуется, так как логически сеть выглядит как одно целое. При использовании шлюза сеть разбивается на подсети.

Разбиение сети на подсети использует ту часть IP-адреса, которая закреплена за номерами хостов. Администратор сети может замаскировать часть IP-адреса и использовать её для назначения номеров подсетей. Фактически, способ разбиения адреса на две части, теперь будет применяться к адресу хоста из IP-адреса сети, в которой организуется разбиение на подсети.

Маска подсети – это четыре байта, которые накладываются на IP-адрес для получения номера подсети. Например, маска 255.255.255.0 позволяет разбить сеть класса B на 254 подсети по 254 узла в каждой. Подсети не только решают, но и создают ряд проблем. Например, происходит потеря адресов, но уже не по причине физических ограничений, а по причине принципа построения адресов подсети. Так, при выделении трех битов на адрес подсети, приводит к образованию не восьми, а только шести подсетей, так как номера 0 и 7 нельзя использовать в силу специального значения IP-адресов, состоящих из нулей или из единиц.

Для стандартных классов сетей маски имеют следующие значения:

- класс A - 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- класс B - 11111111. 11111111. 00000000. 00000000 (255.255.0.0);



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- класс С- 11111111. 11111111. 11111111. 00000000 (255.255.255.0).

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации. Например, адрес 185.23.44.206 попадает в диапазон 128-191, то есть адрес относится к классу В. Следовательно, номером сети являются первые два байта, дополненные двумя нулевыми байтами - 185.23.0.0, а номером узла - 0.0.44.206. Если этот адрес ассоциировать с маской 255.255.255.0, то номером подсети будет 185.23.44.0, а не 185.23.0.0, как это определено системой классов.

В масках количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8, чтобы повторять деление адреса на байты. Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, то есть в двоичном виде:

IP-адрес 129.64.134.5 -10000001.01000000.10000110.00000101

Маска 255.255.128.0 -11111111.11111111.10000000.00000000

Если использовать для определения границы номера сети маску, то 17 последовательных единиц в маске, «наложенные» на IP-адрес, определяют в качестве номера сети в двоичном выражении число:

10000001. 01000000. 10000000. 00000000 или в десятичной форме записи - номер сети 129.64.128.0, а номер узла 0.0.6.5.

Практическая часть

Определить тип используемых в компьютере расположенного в аудитории MAC и IP адреса, а также маску под сети. Для этого вызываем меню выполнить комбинацией клавиш windows+R, затем вводим в появившемся окошке команду cmd и нажимаем выполнить. В появившейся командной строке вводим команду ipconfig /all.

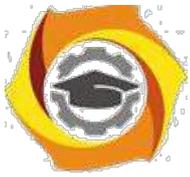
Из вышедшего окошка выписываем данные в лабораторную работу:

1. Физический адрес – mac адрес
2. Ip – адрес
3. Маску подсети.

Определить из каких частей состоят выписанные mac и ip адреса.

Контрольные вопросы

1. Мас адрес и его структура.
2. ip- адрес и его структура.
3. Для чего применяется маска подсети.
4. Какие есть специальные ip – адреса.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

**Практическое занятие 3 Сетевые диагностические утилиты
операционных систем семейства Linux на примере Ubuntu**

ЦЕЛЬ РАБОТЫ:

1. Знать принципы анализа сетевого трафика.
2. Научиться использовать сетевой анализатор (сниффер Wireshark).
3. Научиться анализировать сетевой трафик на примере протоколов ARP, IP и ICMP.

Sniffer (от англ. to sniff – нюхать) – это сетевой анализатор трафика, программа или программно-аппаратное устройство, предназначенное для перехвата и последующего анализа, либо только анализа сетевого трафика, предназначенного для других узлов.

Перехват трафика может осуществляться:

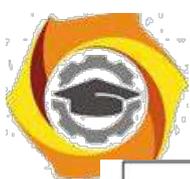
- обычным «прослушиванием» сетевого интерфейса (метод эффективен при использовании в сегменте концентраторов (хабов) вместо коммутаторов (свичей), в противном случае метод малоэффективен, поскольку на сниффер попадают лишь отдельные фреймы);
- подключением сниффера в разрыв канала;
- ответвлением (программным или аппаратным) трафика и направлением его копии на сниффер;
- через анализ побочных электромагнитных излучений и восстановление таким образом прослушиваемого трафика;
- через атаку на канальном (2-й) или сетевом (3-й) уровне, приводящую к перенаправлению трафика жертвы или всего трафика сегмента на сниффер с последующим возвращением трафика в надлежащий адрес.

В начале 1990-х широко применялся хакерами для захвата пользовательских логинов и паролей. Широкое распространение хабов позволяло захватывать трафик без больших усилий в больших сегментах сети.

Снифферы применяются как в благих, так и в деструктивных целях. Анализ прошедшего через сниффер трафика, позволяет:

- Отслеживать сетевую активность приложений.
- Отлаживать протоколы сетевых приложений.
- Локализовать неисправность или ошибку конфигурации.
- Обнаружить паразитный, вирусный и закольцованный трафик, наличие которого увеличивает нагрузку сетевого оборудования и каналов связи.
- Выявить в сети вредоносное и несанкционированное ПО, например, сетевые сканеры, флудеры, троянские программы, клиенты пиринговых сетей и другие.
- Перехватить любой незашифрованный (а порой и зашифрованный) пользовательский трафик с целью узнавания паролей и другой информации.

Постепенно из инструментов, предназначенных только для диагностики, снифферы постепенно превратились в средства для исследований и обучения. Например, они постоянно используются для изучения динамики и взаимодействий в сетях. В частности, они позволяют легко и наглядно изучать тонкости сетевых протоколов. Наблюдая за данными, которые посылает протокол, вы можете глубже понять его функционирование



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

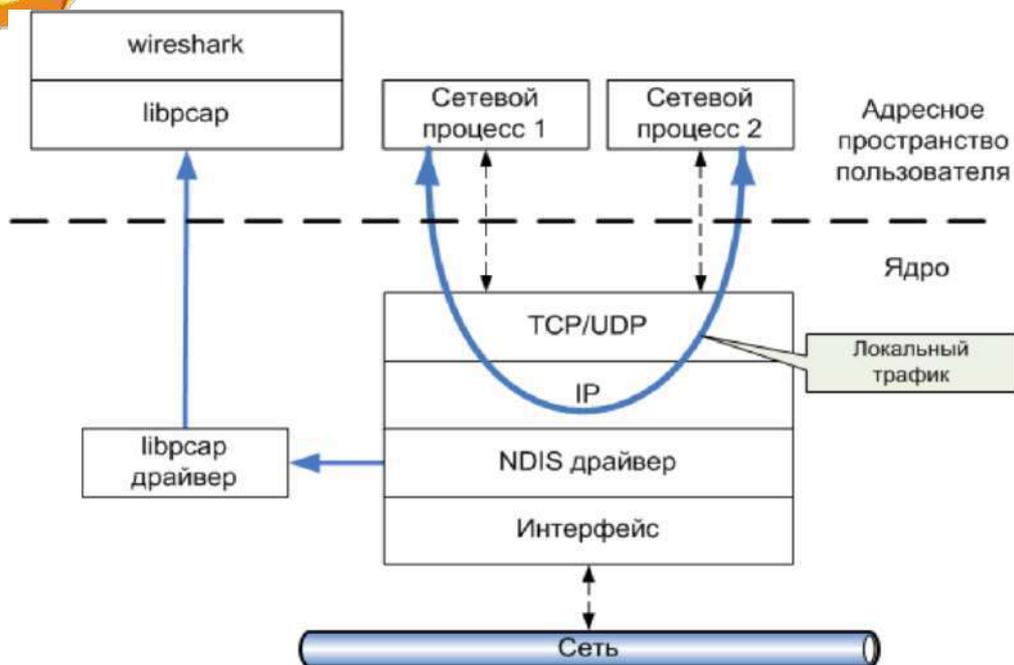


Рис. 1. Принцип «захвата» sniffером сетевого

на практике, а заодно увидеть, когда некоторая конкретная реализация работает не в соответствии со спецификацией.

На сегодняшний момент существует достаточно большое количество хороших реализаций sniffеров. Некоторое из них:

- Tcpdump (<http://www.tcpdump.org/>) – консольный вариант sniffера. Портирован почти подо все наиболее распространенные ОС;
- Wireshark (<http://www.wireshark.org/>) до недавнего момента был известен под названием Ethreal;
- WinDump <http://www.winpcap.org/windump/>;
- IP Sniffer
- и др.

Сниффер Wireshark

Программа Wireshark является одной из самых удобных реализаций sniffеров. Портирована на большое количество платформ. Распространяется абсолютно бесплатно. Имеет смысл использовать данный sniffер для изучения и анализа сетевых протоколов.

Но для начала рассмотрим базовый принцип работы sniffера как раз на примере Wireshark.

Базовый принцип работы sniffеров

Давайте рассмотрим с вами рис. 1. На нем изображена схематично структура сетевой подсистемы ОС. Вся базовая инфраструктура реализована в виде драйверов и работает в режиме ядра. Пользовательские процессы и реализации прикладных протоколов, в частности интерфейс sniffера работают в пользовательском режиме.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

На рисунке отображены 2 пользовательских процесса («сетевой процесс 1» и «сетевой процесс 2»).

Основными компонентами сниффера являются: драйвер для захвата пакетов (libpcap драйвер), интерфейсная библиотека (libpcap) и интерфейс пользователя (Wireshark). Библиотека libpcap (реализация под ОС Windows носит название WinPcap - <http://www.winpcap.org>) – универсальная сетевая библиотека, самостоятельно реализующая большое количество сетевых протоколов и работающая непосредственно с NDIS (Network Driver Interface Specification) драйверами сетевых устройств. На базе данной библиотеки реализовано большое количество сетевых программ, в частности сниффер Wireshark.

Сниффер использует библиотеку в режиме «захвата» пакетов, т.е. может получать копию ВСЕХ данных проходящих через драйвер сетевого интерфейса. Изменения в сами данные не вносятся!

Основной нюанс использования сниффера заключается в том, что он не позволяет производить анализ локального трафика, т.к. он не проходит через драйвер сетевого устройства (см. рис 1.). Т.е., если вы захотите проанализировать сниффером трафик между 2-ми сетевыми процессами на локальной машине (например, ftp-сервер и ftp-клиент), то у вас ждет разочарование. Однако, например при использовании виртуальных машин, сниффер будет работать без проблем, т.к. виртуальные машины эмулируют реальную среду и сетевые адаптеры, поэтому трафик идет через драйвера как и в нормальной ситуации при взаимодействии с другими физическими сетевыми машинами.

Также к недостаткам большинства снифферов стоит отнести и тот факт, что, позволяя анализировать трафик, проходящий через сетевой интерфейс, они не могут указать, какое именно приложение генерирует или получает его. Это объясняется тем, что информация об этом хранится на сетевом (например, IP) уровне сетевого стека, а большинство снифферов использует собственную реализацию стека протоколов (например, библиотеку WinPcap), которая (как уже было показано) работает непосредственно с драйверами устройств.

Также, снифферы вносят дополнительную нагрузку на процессор, т.к. могут обрабатывать достаточно объемный сетевой трафик, в особенности для высокоскоростных соединений (Fast Ethernet, Gigabit Ethernet и др.).

Использование программы Wireshark

Данный сниффер позволяет в режиме реального времени захватывать пакеты из сети, и анализировать их структуру. Также можно анализировать структуру пакетов из файла, содержащего трафик, полученный, например, программой «tcpdump» (unix/linux).

На рис. 2. изображено основное окно программы Wireshark. В стандартном режиме окно сниффера делится на 3 фрейма (панели): список захваченных пакетов, «анализатор» протоколов и исходные данные пакетов. Размер каждого фрейма можно менять по своему усмотрению.

Рассмотрим эти панели подробнее.

Верхняя панель содержит список пакетов, захваченных из сети. Список можно отсортировать по любому полю (в прямом или обратном порядке) – для этого нажать на заголовок соответствующего поля.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

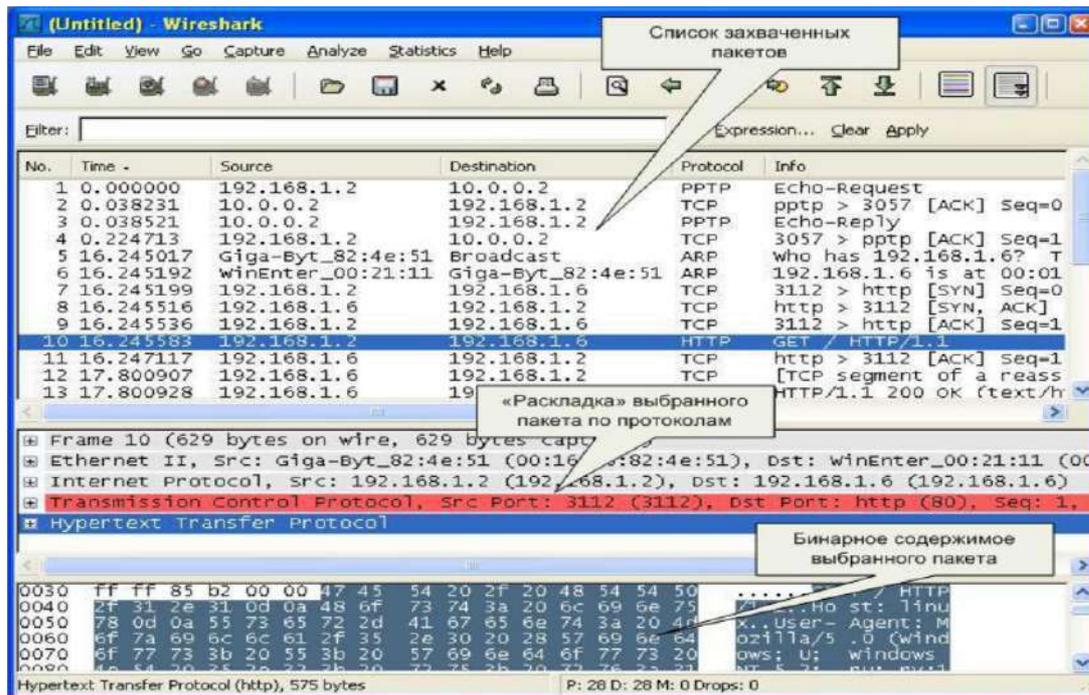


Рис. 2. Основное окно sniffера Wireshark

Каждая строка содержит следующие поля (по умолчанию):

- порядковый номер пакета (No.);
- время поступления пакета (Time);
- источник пакета (Source);
- пункт назначения (Destination);
- протокол (Protocol);
- информационное поле (Info).

Список отображаемых полей настраивается в Edit/Perferencis/Columns. Для того, чтобы изменения возымели эффект необходимо перезапустить программу, предварительно нажав кнопку Save.

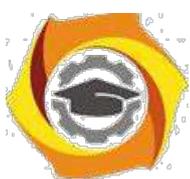
При нажатии правой кнопки мыши на том или ином пакете, появится контекстное меню. Нажатием на среднюю кнопку мыши можно пометить группу интересующих нас пакетов.

Средняя панель содержит т.н. «дерево протоколов» для выбранного в верхнем окне пакета. В этой панели в иерархическом виде для выбранного в верхнем окне захваченного пакета отображается вложенность протоколов в соответствии с моделью взаимодействия открытых систем OSI. По нажатию на правую кнопку мыши вызывается контекстное меню. При «раскрытии» каждого из протокола нажатием на значек «+» слева, выводятся поля данных соответствующих протоколов.

Нижняя панель содержит шестнадцатеричное представление выбранного пакета. При выборе того или иного поля в средней панели автоматически будет подсвечиваться соответствующий участок 16-ого представления.

Захват пакетов

Для начала захвата пакетов необходимо задать параметры захвата. В частности, указать сетевой интерфейс, с которого и будет осуществляться захват пакетов. Это действие



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

доступно через меню как «Capture→Options» или комбинации клавиш CTRL+K (см. рис. 3). Интерфейс, задаваемый в поле «**Interface:**» можно выбрать из соответствующего поля. В примере на рис. 3. Показано, что доступны 3 интерфейса: физический сетевой адаптер («Marvel...»), и интерфейсы для виртуальных каналов, в частности, установленного VPN-соединения («WAN (PPP/SLIP)...»). В большинстве случаев подходит выбор интерфейса сетевого адаптера.

В качестве дополнительных параметров захвата можно указать следующие:

- «**Capture Filter**» – фильтр захвата (будем рассматривать далее). По нажатию на соответствующую кнопку можно применить тот или иной фильтр отбора (из ранее сохраненных). Если таковых не имеется, его можно указать явно в строке редактирования.
- «**Update list of packets in real time**» – обновление списка захваченных пакетов в режиме реального времени.
- «**Stop Capture**» – набор параметров, позволяющих задать то или иное значение при достижении, которого процесс захвата пакетов прекратится.
- «**Name Resolution**» – набор параметров разрешения имен позволяет определить какие из способов разрешения имен должны использоваться.

Для начала мониторинга сетевой активности нужно нажать «Start». После выбора интерфейса, который нас интересует, в дальнейшем можно начинать и останавливать захват пакетов через соответствующие команды в меню «Capture».

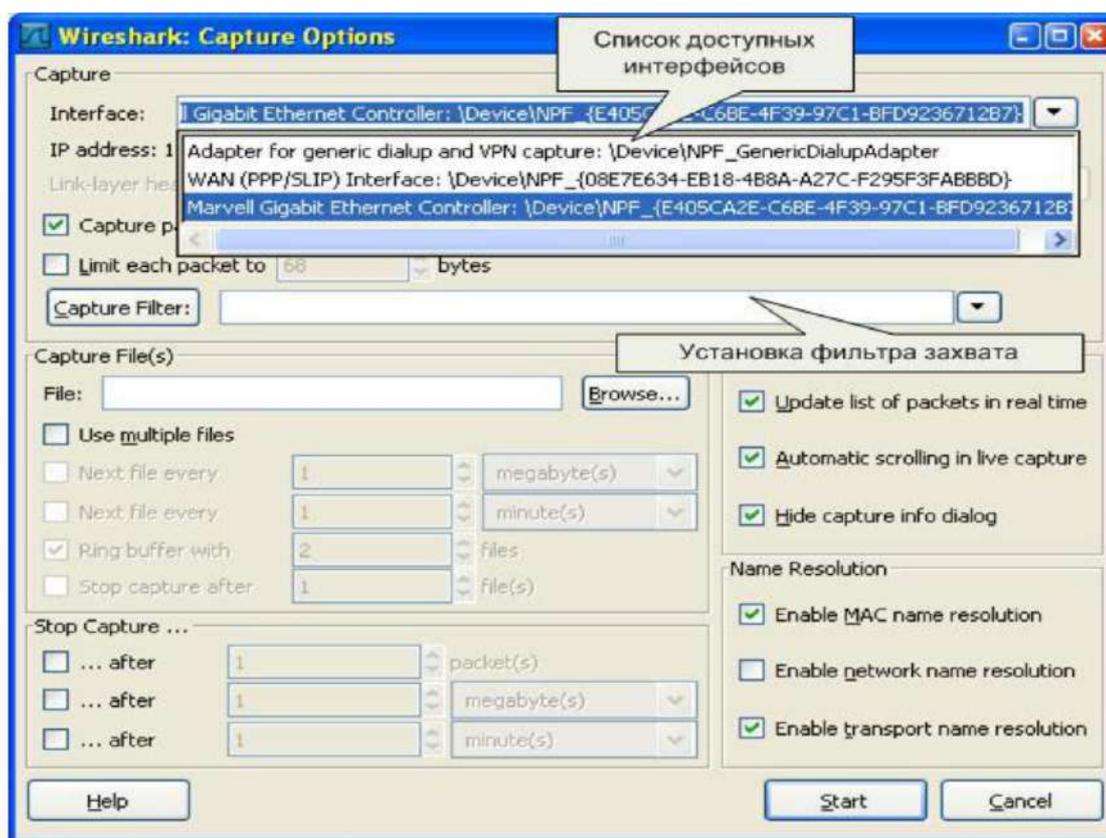


Рис. 3. Выбор интерфейса и параметров захвата пакетов

Фильтрация пакетов

Если запустить сниффер без дополнительных настроек, он будет «захватывать» все пакеты, проходящие через сетевые интерфейсы (см. рис. 1). Вообще, для общего



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

ознакомления с процессами, происходящими в сети, очень полезно пронаблюдать активность сетевых протоколов в реальных условиях работы системы в сети. Пронаблюдать все разнообразие протоколов, запросов, ответов и др. событий.

При целенаправленном использовании сниффера очень часто необходимо выборочно отображать или захватывать пакеты по некоторым заданным критериям. Для этих целей служат фильтры отображения и захвата, соответственно.

Типы фильтрации трафика

Существует два варианта фильтрации пакетов: на этапе захвата и на этапе отображения пользователю. В первом случае эффективность работы сниффера и потребляемые им системные ресурсы значительно ниже, нежели во втором случае. Это объясняется тем, что при достаточно интенсивном сетевом трафике и продолжительном времени захвата все пакеты должны быть захвачены и сохранены либо в память, либо на дисковое устройство. Самые простые подсчеты могут показать, что даже для 100-мегабитной сети системных ресурсов хватит на непродолжительное время. Фильтрация захвата уже на момент получения пакета гораздо эффективнее, однако в таком случае она должна быть реализована на уровне самих драйверов захвата. Данный факт, естественно, усложняет реализацию сниффера. Wireshark поддерживает оба варианта фильтрации. Рассмотрим

Фильтры отображения

Фильтры отображения представляют собой достаточно мощное средство отображения трафика. Фильтры задаются в строке, располагающейся вверху основного экрана («Filter:»). Простейший фильтр отображения, позволяет отобразить пакеты по тому или иному протоколу. Для этого в строке требуется указать название протокола (например HTTP) и нажать кнопку «Apply». После этого в верхнем окне останутся пакеты, принадлежащие этому протоколу. Кнопкой «Reset» действие фильтра отменяется.

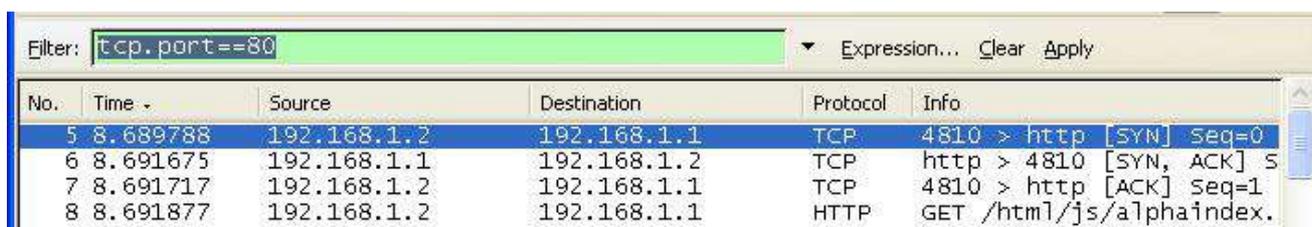
Для работы с фильтрами можно вызвать окно «Analyze/Display Filters». Можно сохранять созданные выражения под определенными именами для последующего использования и т.д.

С помощью логических операций (синтаксис языка Си) можно составлять логические выражения. Логическая истина - 1, ложь - 0.

Список поддерживаемых логических операций:

eq	==	равенство
ne	!=	не равно
gt	>	больше чем
Lt	<	меньше чем
ge	>=	больше равно
Le	<=	меньше равно

Например: tcp.port == 80 (см. рис. 4).





Мастер построения фильтров отображения доступен через кнопку «Expression...» (см. рис. 5).

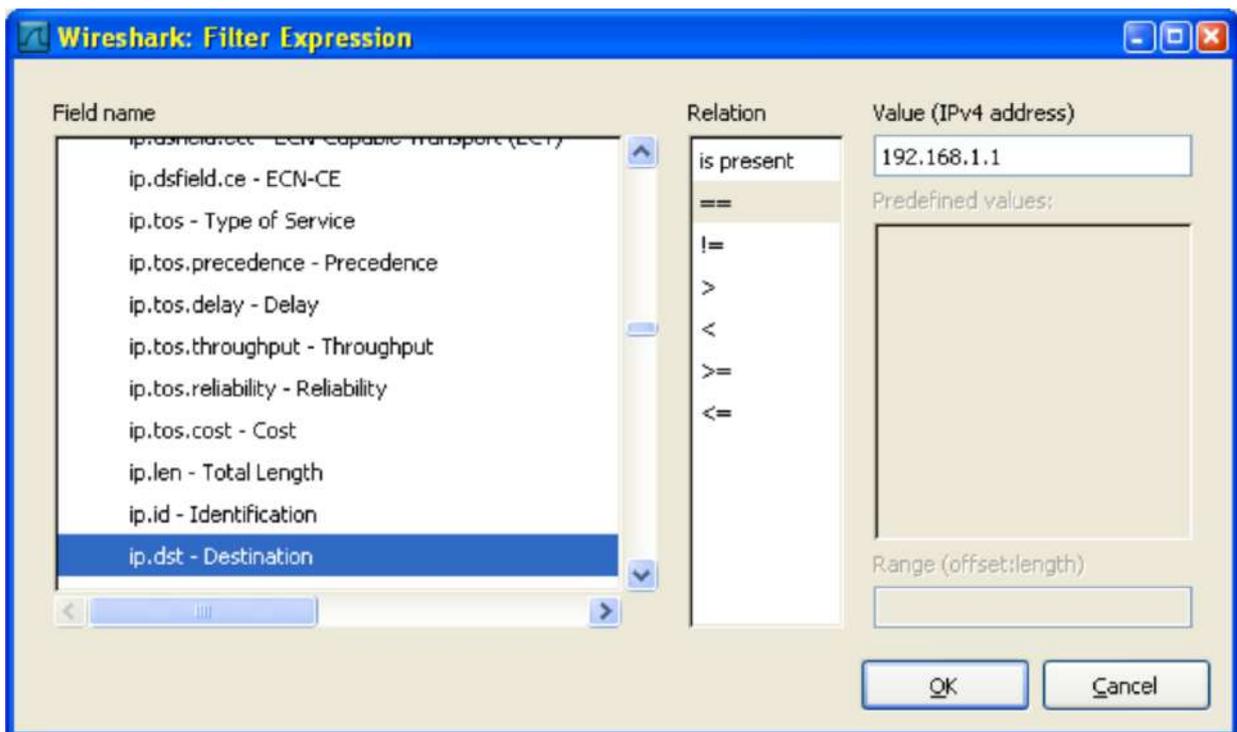


Рис. 5. Построение фильтров отображения

Фильтры захвата

С помощью данных фильтров можно захватывать из сети только те пакеты, которые подходят под критерий отбора. Если не задано никакого фильтра (по умолчанию), то будут захватываться все пакеты. В противном случае только пакеты, для которых указанное выражение будет истинным. Синтаксис фильтров захвата несколько отличается от синтаксиса фильтров отображения. Выражение состоит из одного или более примитивов



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

разделенных пробельными символами. На рис. 6 приведен пример фильтра для захвата пакетов, адресованных на 80-й порт (http) узла с ip-адресом 10.197.0.11.

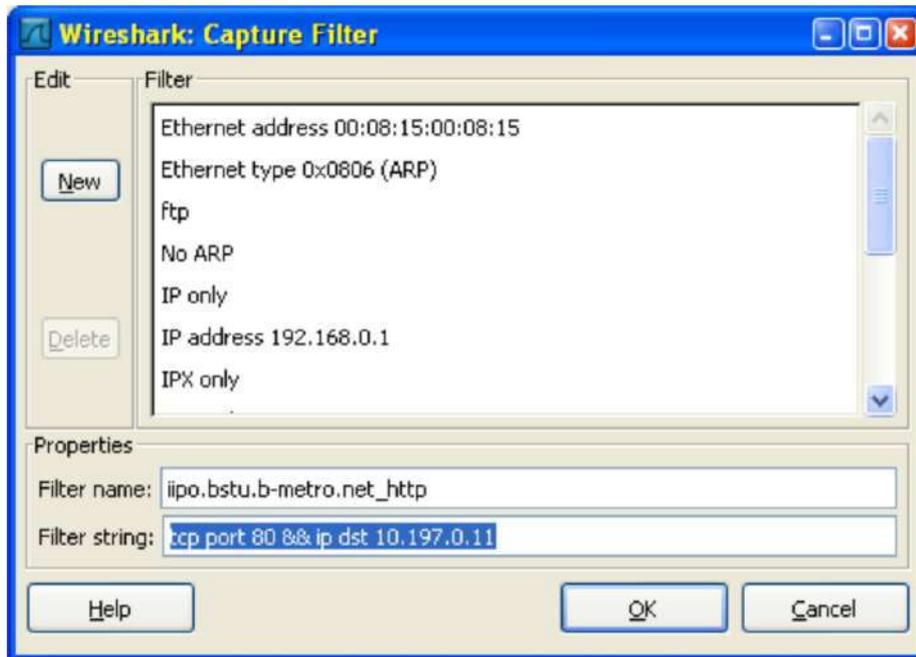


Рис. 6. Пример фильтра захвата

Существует три различных типа примитивов: *type*, *dir*, *proto*.

Спецификатор *type* определяет тип параметра. Возможные параметры: **host**; **net**; **port**.

Например:

- host linux
- net 192.168.128
- port 80

Если не указано никакого типа предполагается что это параметр **host**.

Спецификатор *dir* определяют направление передачи. Возможные направления: **src**; **dst**;

src or dst; **src and dst**.

Например:

- src linux
- dst net 192.168.128
- src or dst port http

Если не определено направление то предполагается направление «**src or dst**». Для протоколов типа point-to-point используются спецификаторы *inbound* и *outbound*.

Спецификатор *proto* определяют тип протокола, которому принадлежит пакет.

Возможные протоколы: **ether**; **fddi**; **tr**; **ip/ipv6**; **arp/rarp**; **decnet**; **tcp**; **udp**.

Например:

- ether src linux
- arp net 192.168.128
- tcp port 80

Если протокол не определен, то будут захватываться пакеты всех протоколов. То есть: «src linux» означает «(ip or arp or rarp) src linux»; «net stam» означает «(ip or arp or rarp) net stam»; «port 53» означает «(tcp or udp) port 53».

Также существует несколько специальных спецификаторов, которые не попадают в описанные выше случаи:

- *gateway*;



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- *broadcast*;
- *less*;
- *greater*;
- *арифметические выражения*.

Сложные фильтры захвата строятся с использованием логических выражений.

Список операций:

not	!	отрицание
and	&&	конкатенация (логическое И)
or		альтернатива (логическое ИЛИ)

Примеры фильтров захвата

Ниже рассмотрены некоторые примеры построения фильтров захвата.

- Захват всех пакетов на сетевом интерфейсе хоста 192.168.1.2:
`host 192.168.1.2`
- Захват трафика между хостом host1 И хостами host2 ИЛИ host3:
`host host1 and (host2 or host3)`
- Захват всех IP-пакетов между хостом host1 и каждым хостом за исключением hostX:
`ip host host1 and not hostX`
- Захват пакетов ни сгенерированных ни адресованных локальными хостами:
`ip and not net localnet`
- Захват IP-пакетов размером больше чем 576 байт, проходящих через шлюз snup:
`gateway snup and ip[2:2] > 576`
- Захват всех ICMP пакетов, за исключением пакетов ping:
`icmp[0] != 8 and icmp[0] != 0`

Статистическая обработка сетевого трафика

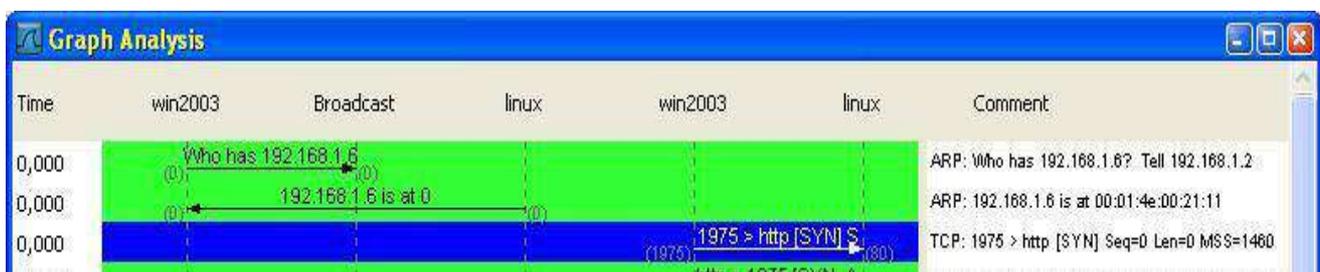
Сниффер Wireshark позволяет выполнять различную статистическую обработку полученных данных. Все доступные операции находятся в меню «Statistics».

Общая статистика – количество полученных/переданных пакетов, средняя скорость передачи и т.д. доступны через пункт «Statistics->Summary».

Получить информацию по статистике обработанных протоколов в полученных пакетах можно через пункт «Statistics->Protocol Hierarchy».

Статистику по типу ip-пакетов, их размеру и порту назначения можно получить выбрав подпункты меню «IP-address...», «Packet length» и «Port type» соответственно.

Одной из наиболее интересных возможностей является генерация диаграммы взаимодействия между узлами, которая доступна пунктом меню «Flow Graph...». В результате можно наблюдать в достаточно наглядной форме процесс взаимодействия на уровне протоколов. Например, на рис. 7 приведена диаграмма взаимодействия при получении узлом win2003 статической web-странички с сервера <http://linux>.





Заключение

Программы-снифферы – это незаменимый инструмент для изучения того, что происходит в сети. Если знать, что в действительности посылается или принимается «по проводам», то трудные, на первый взгляд, ошибки удастся легко найти и исправить. Сниффер представляет собой также важный инструмент для исследований динамики сети, а равно средство обучения.

ПРИМЕР ИССЛЕДОВАНИЯ ПРОТОКОЛА С ИСПОЛЬЗОВАНИЕМ СНИФФЕРА

В качестве примера исследования некоторого протокола с использованием сниффера рассмотрим протокол ARP.

Протокол ARP

ARP (англ. Address Resolution Protocol — протокол разрешения адресов) — сетевой протокол, предназначенный для преобразования IP-адресов (адресов сетевого уровня) в MAC-адреса (адреса канального уровня) в сетях TCP/IP. Он определен в **RFC 826**.

Данный протокол очень распространенный и чрезвычайно важный. Каждый узел сети имеет как минимум два адреса, физический адрес и логический адрес. В сети Ethernet для идентификации источника и получателя информации используются оба адреса. Информация, пересылаемая от одного компьютера другому по сети, содержит в себе физический адрес отправителя, IP-адрес отправителя, физический адрес получателя и IP-адрес получателя. ARP-протокол обеспечивает связь между этими двумя адресами. Существует четыре типа ARP-сообщений: ARP-запрос (ARP request), ARP-ответ (ARP reply), RARP-запрос (RARP-request) и RARP-ответ (RARP-reply). Локальный хост при помощи ARP-запроса запрашивает



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

физический адрес хоста-получателя. Ответ (физический адрес хоста-получателя) приходит в виде ARP-ответа. Хост-получатель, вместе с ответом, шлет также RARP-запрос, адресованный отправителю, для того, чтобы проверить его IP адрес. После проверки IP адреса отправителя, начинается передача пакетов данных.

Перед тем, как создать подключение к какому-либо устройству в сети, IP-протокол проверяет свой ARP-кеш, чтобы выяснить, не зарегистрирована ли в нём уже нужная для подключения информация о хосте-получателе. Если такой записи в ARP-кеше нет, то выполняется широковещательный ARP-запрос. Этот запрос для устройств в сети имеет следующий смысл: «Кто-нибудь знает физический адрес устройства, обладающего следующим IP-адресом?» Когда получатель примет этот пакет, то должен будет ответить: «Да, это мой IP-адрес. Мой физический адрес следующий: ...» После этого отправитель обновит свой ARP-кеш, и будет способен передать информацию получателю.

RARP (англ. Reverse Address Resolution Protocol – обратный протокол преобразования адресов) – выполняет обратное отображение адресов, то есть преобразует аппаратный адрес в IP-адрес.

Протокол применяется во время загрузки узла (например компьютера), когда он посылает групповое сообщение-запрос со своим физическим адресом. Сервер принимает это сообщение и просматривает свои таблицы (либо перенаправляет запрос куда-либо ещё) в поисках соответствующего физическому IP-адреса. После обнаружения найденный адрес отсылается обратно на запросивший его узел. Другие станции также могут «слышать» этот диалог и локально сохранить эту информацию в своих ARP-таблицах.

RARP позволяет разделять IP-адреса между не часто используемыми хост-узлами. После использования каким либо узлом IP-адреса он может быть освобождён и выдан другому узлу. RARP является дополнением к ARP, и описан в **RFC 903**.

Для просмотра ARP-кеша можно использовать одноименную утилиту arp с параметром «-а». Например:

```
D:\>arp -a
Interface: 192.168.1.2 --- 0x10003
Internet Address  Physical Address  Type
192.168.1.1      00-15-e9-b6-67-4f  dynamic
192.168.1.6     00-01-4e-00-21-11  dynamic
```

Из данного результата команды arp видно, что в кеше на данный момент находится 2 записи и видны соответственно ip-адреса машин и MAC-адреса их сетевых адаптеров.

Записи в ARP-кеше могут быть статическими и динамическими. Пример, данный выше, описывает динамическую запись кеша. Хост-отправитель автоматически послал запрос получателю, не уведомляя при этом пользователя. Записи в ARP-кеш можно добавлять вручную, создавая статические (static) записи кеша. Это можно сделать при помощи команды:

```
arp -s <IP адрес> <MAC адрес>
```

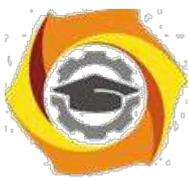
Также можно удалять записи из ARP-кеша. Это осуществляется путем следующего вызова:

```
arp -d <IP адрес>
```

После того, как IP-адрес прошёл процедуру разрешения адреса, он остается в кеше в течение 2-х минут. Если в течение этих двух минут произошла повторная передача данных по этому адресу, то время хранения записи в кеше продлевается ещё на 2 минуты. Эта процедура может повторяться до тех пор, пока запись в кеше просуществует до 10 минут. После этого запись будет удалена из кеша и будет отправлен повторный ARP-запрос.

ARP изначально был разработан не только для IP протокола, но в настоящее время в основном используется для сопоставления IP- и MAC-адресов.

Посмотрим же на практике как работает протокол ARP/RARP. Для этого воспользуемся сниффером для захвата сетевого трафика.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Рассмотрим пример работы протокола ARP при обращении к машине с адресом 192.168.1.5, выполнив запрос с машины с адресом 192.168.1.2. Для успешного эксперимента предварительно очистим arp-кеш командой

```
arp -d 192.168.1.5
```

Для фильтрации ARP/RARP трафика воспользуемся фильтром захвата. В нашем случае это будет простой фильтр

```
arp or rarp
```

Далее запустим захват трафика командой «Start» и выполним обращение к заданной машине, например, «пропинговав» ее:

```
D:\>ping 192.168.1.5
```

```
Pinging 192.168.1.5 with 32 bytes of data:
```

```
Reply from 192.168.1.5: bytes=32 time<1ms TTL=64
```

```
Ping statistics for 192.168.1.5:
```

```
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Так как на момент начала работы утилиты ping в arp-кеше не было информации о MAC-адресе соответствующего узла, то первоначально система должна выполнить определение это самого MAC-адреса, сгенерировав ARP-запрос и отослав его в сеть широковещательным пакетом. После чего она будет ожидать ответа от заданного узла.

Посмотрим же, что мы получим на практике. После остановки сниффера мы должны увидеть результат схожий с тем, что отображен на рис. 8. В нашем случае мы видим 2 захваченных пакета: ARP-запрос и ARP-ответ.

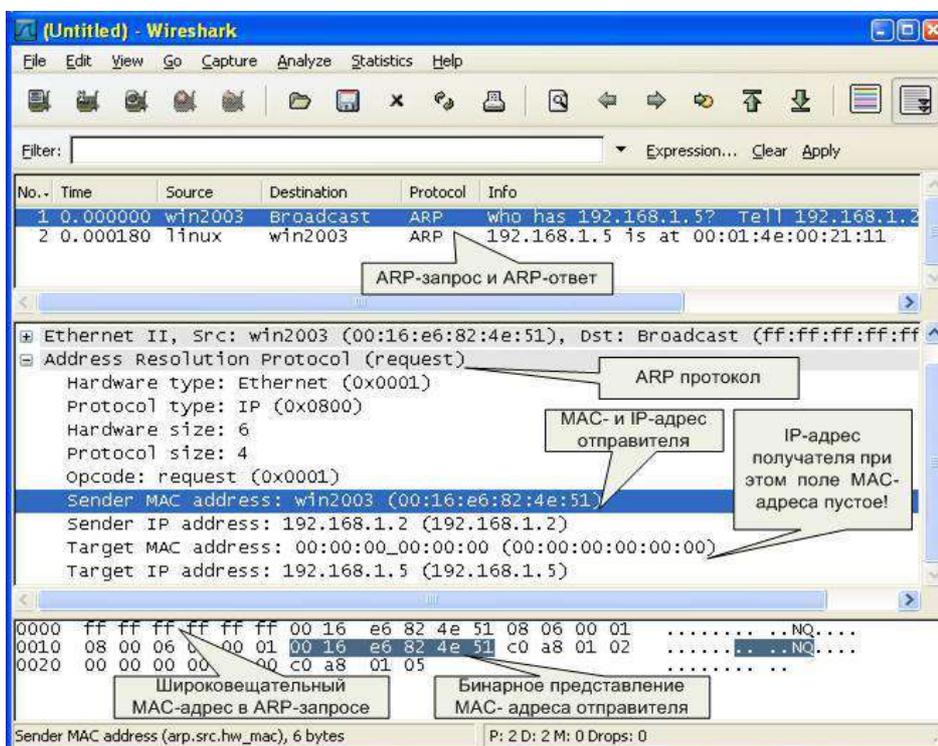


Рис.8. Анализ пакетов. ARP-запрос (пакет №1). Выделив пакет курсором, мы получаем его раскладку по протоколам (Ethernet+ARP) в среднем окне. Wireshark очень наглядно «раскладывает» заголовок протокола по полям.

Проанализируем полученные пакеты. Сначала рассмотрим ARP-запрос (пакет №1). Выделив пакет курсором, мы получаем его раскладку по протоколам (Ethernet+ARP) в среднем окне. Wireshark очень наглядно «раскладывает» заголовок протокола по полям.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Мы можем видеть, что в пакете указаны MAC- и IP-адреса отправителя («Sender MAC address» и «Sender IP address» соответственно). Это параметры машины, с которой выполняется запрос. В данном случае запрос направлен на получения («Opcode: request» – запрос) MAC-адреса машины, у которой IP-адрес («Protocol type: IP») 192.168.1.5 («Target IP address»). При этом поле «Target MAC address» обнулено. Так как получатель ARP-запроса на момент запроса не известен, Ethernet-пакет отправляется всем машинам в данном локальном сегменте, о чем сигнализирует MAC адрес Ethernet-пакета «ff:ff:ff:ff:ff:ff».

Примечание. Обратите внимание, что пакет представляет собой бинарную последовательность и сниффер выполняет большую работу по преобразованию полей из бинарного представления в удобочитаемый вариант.

Все работающие машины в сети получают пакет с ARP-запросом, анализируют его, а ответ отправляет только та машина, чей IP-адрес соответствует IP-адресу в запросе. Таким образом, второй полученный пакет является ARP-ответом (см. рис. 9). Это следует из параметра поля «Opcode: reply». Обратите внимание, что данный пакет был отправлен именно той машиной, чей MAC-адрес нас и интересовал («Sender IP address: 192.168.1.5»). При этом поле «Sender MAC address» заполнено значением «00:01:4E:00:21:11».

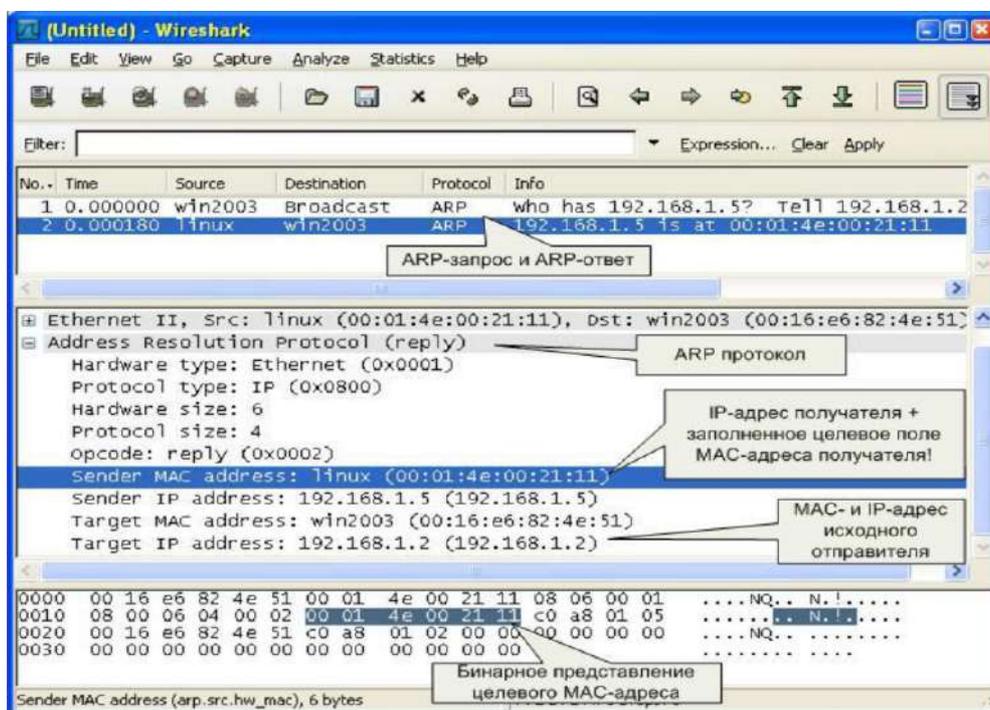


Рис. 9. Анализ ARP-ответа

Примечание. Обратите внимание на поле «Info» в списке захваченных пакетов. Сниффер и тут упрощает анализ сетевого трафика, подсказывая назначение пакетов ☺

Теперь мы можем повторно просмотреть ARP-кеш и сверить данные в нем с данными, которые мы узнали из анализа пакетов ARP-запрос/ответа:

```
D:\>arp -a
Interface: 192.168.1.2 --- 0x10003
Internet Address Physical Address Type
192.168.1.5 00-01-4e-00-21-11 dynamic
```

Стоит также отметить, что в реальных условиях в локальной сети с большим количеством машин arp/rarp трафик бывает гораздо более интенсивным.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Задание на практическую работу

1. Изучить интерфейс программы Wireshark
([\\corp.mgkit.ru/dfs/work/wireshark](http://corp.mgkit.ru/dfs/work/wireshark))
2. Захватить 100 произвольных пакетов. Определить статистические данные:
 - процентное соотношение трафика разных протоколов в сети;
 - среднюю скорость кадров/сек;
 - среднюю скорость байт/сек;
 - минимальный, максимальный и средний размеры пакета;
 - степень использования полосы пропускания канала (загрузку сети).
3. Зафиксировать 20 IP-пакетов. Определить статистические данные:
 - процентное соотношение трафика разных протоколов стека tcp/ip в сети;
 - средний, минимальный, максимальный размеры пакета.
4. Выполнить анализ ARP-протокола по примеру из методических указаний.
5. На примере любого IP-пакета указать структуры протоколов Ethernet и IP, Отметить поля заголовков и описать их.
6. Проанализировать и описать принцип работы утилиты *ping*. При этом описать все протоколы, используемые утилитой. Описать все поля протоколов. Составить диаграмму взаимодействия машин при работе утилиты *ping*.

Контрольные вопросы

1. Каковы основные цели мониторинга сетевого трафика?
2. Чем отличается мониторинг трафика от фильтрации?
3. Каково назначение класса программ-снифферов?
4. Какие основные функции выполняют снифферы?
5. Зачем используются фильтры отображения и фильтры захвата сниффера Wireshark? В чем их отличие?
6. Какие базовые функции статистической обработки захваченных пакетов имеет сниффер Wireshark?
7. Какие задачи рассчитан решать протокол ARP?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Лабораторная работа 3 Анализ и исследование TCP/IP соединений

Понятие и назначение прокси-серверов.
Принцип работы прокси-сервера.
Настройка Nat-прокси на базе Windows Server.

ЦЕЛЬ РАБОТЫ

1. Выяснить назначение принцип работы Proxy-серверов.
2. Изучить механизм работы NAT-прокси.
3. Уметь настраивать NAT-прокси на базе Windows Server.

ВВЕДЕНИЕ

Одной из важных задач при организации работы локальных сетей (и не только локальных) является организация доступа к ресурсам глобальных сетей (в частности, *Internet*). Чаще всего реализовать отдельный канал в глобальную сеть для каждой клиентской машины не представляется возможным. В таких случаях задача решается организацией канала через некоторую машину (в дальнейшем просто *сервер*), которая имеет сетевой интерфейс в глобальную сеть, с установленным специализированным ПО.

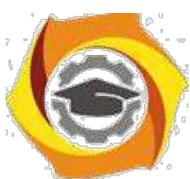
На данной лабораторной работе мы с вами познакомимся с механизмами реализации доступа во внешние сети.

БАЗОВЫЕ ПОНЯТИЯ

Прокси-сервер (от англ. proxy — «представитель, уполномоченный») – служба в компьютерных сетях, позволяющая клиентам выполнять косвенные запросы к другим сетевым службам. Сначала клиент подключается к прокси-серверу и запрашивает какой-либо ресурс (например, файл), расположенный на другом сервере. Затем прокси-сервер либо подключается к указанному серверу и получает ресурс у него, либо возвращает ресурс из собственного кеша (в случаях, если прокси имеет свой кеш). В некоторых случаях запрос клиента или ответ сервера может быть изменён прокси-сервером в определённых целях.

Прокси-сервера используют для того, чтобы обеспечить эффективный и безопасный доступ в Интернет. Их устанавливают в различных организациях для обеспечения взаимодействия локальной сети с глобальной сетью Интернет. Необходимость в такой программе возникает обычно, если с пользовательского компьютера невозможно работать в интернете непосредственно напрямую из-за того, что у него нет прямого подключения к интернету (например, модема), но есть на другом компьютере в его сети. Тогда на этом другом компьютере ставят программу прокси, а все остальные компьютеры в локальной сети настраивают таким образом, чтобы работа велась через прокси. Сейчас через прокси умеют работать практически все популярные интернет-программы.

Важное отличие прокси от маршрутизатора (в IP-сетях) в том, что при использовании маршрутизатора IP-пакеты остаются без изменений – в них сохраняются исходные IP-адреса компьютеров как отправителя так и получателя. А прокси всегда работает от своего адреса, а адреса клиентов не передаются, т.к. чаще всего являются локальными и невалидными в сети Internet.



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

На рис. 1. представлена простейшая схема сети с прокси сервером.

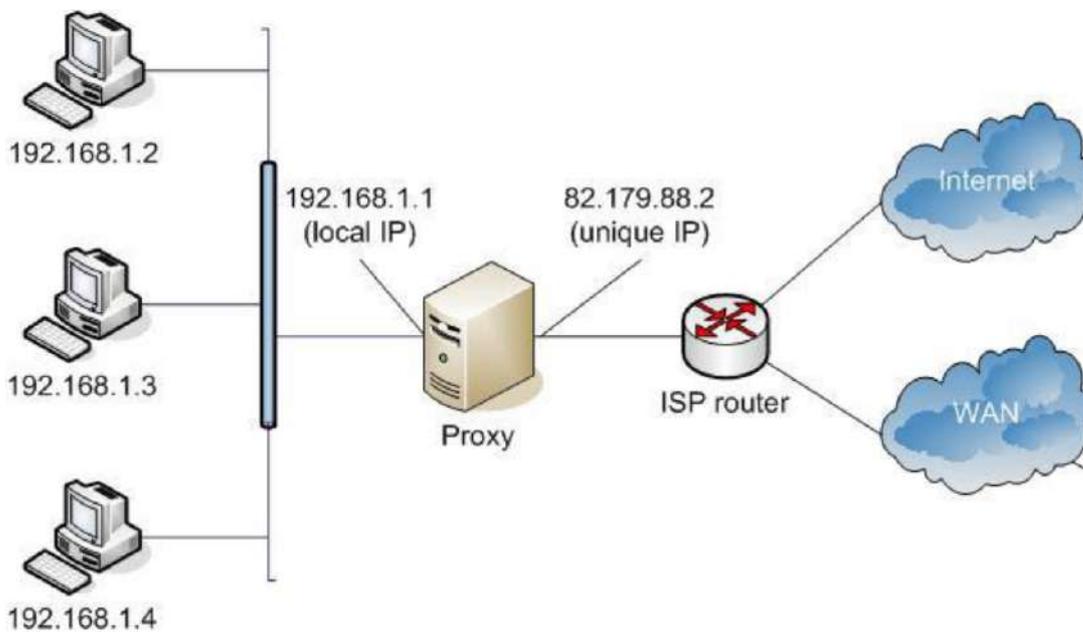


Рис. 1. Схема сети с прокси-сервером



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Виды и функции прокси-серверов

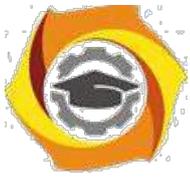
Мы дали обобщенное понятие прокси-серверов. Давайте теперь рассмотрим, какие конкретные реализации прокси существуют и в чем их особенность.

NAT-прокси (Network Address Translation) – самый простой вид прокси. Это, по сути, специализированный маршрутизатор, подменяющий адреса. Простейшая реализация NAT входит в состав Windows 2000+. Она называется «Общий доступ к подключению интернета» (Internet Connection Sharing) и включается галочкой в свойствах модемного соединения. Этот прокси работает прозрачно для пользователя, никаких специальных настроек в программах не требуется. Но он достаточно ограничен в возможностях. Другие реализации NAT-прокси могут быть более гибкими, но их общая проблема – универсальность. Они «не вникают» в тонкости тех прикладных протоколов, которые через себя пропускают, поэтому и не имеют средств управления ими. Далее мы более детально рассмотрим механизм NAT.

Специализированные прокси. В общем случае для каждого их сетевых прикладных протоколов необходимо реализовывать свои прокси. Обычно прокси – это сетевой сервис, который настраивается для работы на некоторый порт. Клиентские программы должны знать этот порт и, естественно, уметь работать с прокси. Далее приведены некоторые специализированные типы прокси и их функции.

HTTP-прокси – самый распространенный. Он предназначен для организации работы браузеров и других программ, использующих протокол HTTP. Браузер передает прокси-серверу URL ресурса, прокси-сервер получает его с запрашиваемого веб-сервера (или с другого прокси-сервера) и отдает браузеру. У HTTP-прокси могут быть реализованы широкие возможности при выполнении запросов:

- Возможно сохранение полученных файлы на диске сервера. Впоследствии, если запрашиваемый файл уже скачивался, то при повторном запросе можно «мгновенно» выдать его с диска без обращения в Интернет, что приводит также к экономии внешней трафик. Эта возможность называется кэшированием. Прокси с поддержкой данной функциональности называют кеширующими. Часто это одна из наиболее приоритетных функций. Обычно на практике экономия трафика достигает порядка 10-15%.
- Можно ограничивать доступ к ресурсам. В частности, производить авторизацию пользователя при запросе ресурсов. Также можно завести «черный список» сайтов, на которые прокси не будет пускать пользователей (или определенную часть пользователей, или в определенное время и т.д.). Ограничения можно реализовать по-разному. Можно просто не выдавать ресурс – например, выдавая вместо него страницу «запрещено администратором» или «не найдено». Можно спрашивать пароль и авторизованных пользователей допускать к просмотру. Можно, не спрашивая пароля, принимать решение на основании адреса или имени компьютера пользователя. Условия и действия в принципе могут быть достаточно сложными.
- Можно выдавать не тот ресурс, который запрашивается браузером. Например, вместо рекламных баннеров и счетчиков показывать пользователям прозрачные картинки, не нарушающие дизайн сайта, но существенно экономящие время и трафик за счет исключения загрузки картинок извне.
- Можно ограничивать скорость работы для отдельных пользователей, групп или ресурсов. Например, установить правило, чтобы файлы *.mp3 качались на скорости не более 1кб/сек. Эта возможность, к сожалению, есть не во всех прокси.
- Ведутся журналы работы прокси – можно подсчитывать трафик за заданный период, по заданному пользователю, выяснять популярность тех или иных ресурсов и т.д.
- Можно маршрутизировать веб-запросы – например, часть направлять напрямую, часть через другие прокси (прокси провайдера, спутниковые прокси и т.д.). Это тоже



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

помогает эффективнее управлять стоимостью трафика и скоростью работы прокси в целом.

FTP-прокси бывает двух основных видов в зависимости от протокола работы самого прокси. С ftp-серверами этот прокси, конечно, всегда работает по протоколу FTP. А вот с клиентскими программами – браузерами и ftp-клиентами (CuteFTP, FAR, и др.) прокси может работать как по FTP, так и по HTTP. Второй способ удобнее для браузеров, т.к. исторически является для них «родным». Браузер запрашивает ресурс у прокси, указывая протокол целевого сервера в URL – http или ftp. В зависимости от этого прокси выбирает протокол работы с целевым сервером, а протокол работы с браузером не меняется – HTTP. Поэтому, как правило, функцию работы с FTP-серверами также вставляют в HTTP-прокси, т.е. HTTP-прокси, описанный выше, обычно с одинаковым успехом работает как с HTTP, так и с FTP-серверами. Но при «конвертации» протоколов FTP \leftrightarrow HTTP теряется часть полезных функций протокола FTP. Поэтому специализированные ftp-клиенты предпочитают и специальный прокси (FTP-gate), работающий с обеими сторонами по FTP. Хотя встречаются и вносящие путаницу названия. Например, в программе CuteFTP и FAR FTP-gate называют firewall, хотя FireWall в общем случае – это вообще не прокси.

HTTPS-прокси – фактически часть HTTP-прокси. S в названии означает «secure», т.е. безопасный. Не смотря на то, что программно это часть HTTP-прокси, обычно HTTPS выделяют в отдельную категорию (и есть отдельное поле для него в настройке браузеров). Обычно этот протокол – безопасный HTTP – применяют, когда требуется передача секретной информации, например, номеров кредитных карт. При использовании обычного HTTP-прокси всю передаваемую информацию можно перехватить средствами самого прокси или на более низком уровне. Поэтому в таких случаях применяют secure HTTP – всё передаваемое при этом шифруется. Прокси-серверу при этом дается только команда «соединится с таким-то сервером», и после соединения прокси передает в обе стороны зашифрованный трафик, не имея возможности узнать подробности (соответственно и многие средства управления доступом – такие как фильтрация картинок – не могут быть реализованы для HTTPS, т.к. прокси в этом случае неизвестно, что именно передается). В частности, в таком случае невозможна и антивирусная фильтрация трафика!!! Собственно в процессе шифрации/дешифрации прокси тоже участия не принимает – это делают клиентская программа и целевой сервер. Наличие команды «соединиться с таким-то сервером» в HTTPS-прокси приводит к интересному и полезному побочному эффекту, которым все чаще пользуются разработчики клиентских программ. Так как после соединения с указанным сервером HTTPS-прокси лишь пассивно передает данные в обе стороны, не производя никакой обработки этого потока вплоть до отключения клиента или сервера, это позволяет использовать прокси для передачи почти любого TCP-протокола, а не только HTTP. То есть HTTPS-прокси одновременно является и простым POP3-прокси, SMTP-прокси, IMAP-прокси, NNTP-прокси и т.д. – при условии, что соответствующая клиентская программа умеет так эксплуатировать HTTPS-прокси (увы, далеко не все еще это умеют, но есть вспомогательные программы, «заворачивающие» трафик обычных клиентов через HTTPS-прокси). Никаких модификаций целевого сервера не требуется.

Mapping-прокси – это способ заставить работать через прокси те программы, которые умеют работать с интернетом только напрямую. При настройке такого прокси создается как бы «копия» целевого сервера, но доступная через один из портов прокси-сервера для всех клиентов локальной сети. Таким образом, устанавливает локальное «отображение» заданного сервера.

Socks-прокси. Большинство прокси-серверов являются узкоспециализированными и рассчитаны на конкретные прикладные протоколы (см. выше), то есть для новых протоколов требуются новые прокси-серверы.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Socks (современная версия Socks5) обладает двумя ключевыми особенностями, выделяющими его из группы прокси-серверов:

- Он не зависит от высокоуровневых протоколов (HTTP, FTP, POP3, SMTP, NNTP и т. д.), так как осуществляет представление клиентов на более низком уровне (TCP и UDP).
- Приложение в локальной сети может попросить Socks-сервер выступить в роли сервера от лица клиента. То есть приложение в локальной сети сможет принимать соединения извне, несмотря на отсутствие реального IP-адреса.

Через Socks можно заставить работать даже приложения, которые и понятия не имеют о прокси. Фактически Socks-сервер является программно-управляемым mapping-прокси, причем с описанным единым интерфейсом. Все mapping-прокси так или иначе программно управляются, но под руководством администратора сети (человека), и отображения статичны. А Socks-сервер управляется прикладными программами, и отображения устанавливаются, только когда они нужны, и на то время, пока они нужны.

Прозрачный прокси-сервер

Данная функциональность поддерживается большинством современных прокси-серверов. До внедрения этой функции пользователи локальной сети должны были указывать адрес и порт прокси в LAN настройках браузера. При использовании прозрачного прокси запросы незаметно для пользовательских приложений транслируются на стороне сервера порт прокси-сервера. В остальном работа ни чем не отличается от работы с обычным прокси-сервером.

Открытый прокси-сервер

В отличие от обычных прокси-серверов, которыми пользуются ограниченное количество лиц (обычно в зоне ответственности владельца прокси-сервера), открытый прокси-сервер позволяет практически любому узлу сети обращаться к другим узлам сети, скрывая свой истинный адрес от узла-получателя.

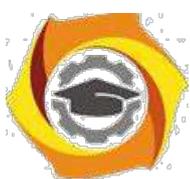
Открытые прокси-сервера могут использоваться для обеспечения (частичной) анонимности в Интернете, так как скрывают IP-адрес пользователя, направляя все пользовательские запросы через себя. Поэтому такие прокси-сервера еще называют анонимными. При этом сам прокси-сервер может вести логи обращений.

Потенциальное использование открытых прокси-серверов для скрытия адреса пользователя приводит к тому, что сайты некоторых интернет-сервисов запрещают доступ к своим ресурсам с открытых прокси-серверов. Например, почтовые службы mail.yandex.ru отказываются работать с пользователями открытых прокси-серверов.

Механизм работы NAT-прокси

NAT – это механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов. Был разработан в 1994 году для использования совместно с протоколом IPv4. Механизм NAT определен в RFC 1631, RFC 2663 и RFC 3022. Преобразование адресов методом NAT может производиться почти любым маршрутизирующим устройством: маршрутизатором, сервером доступа, межсетевым экраном. Например, практически во всех сетевых ADSL-модемах имеется встроенный NAT-прокси.

Суть механизма состоит в замене обратного (source) адреса при прохождении пакета в одну сторону и обратной замене адреса назначения (destination) в ответном пакете. Наряду с адресами source/destination могут также заменяться номера портов source/destination. Информация о произведенной трансляции адресов сохраняется в течении некоторого времени в таблице трансляции. На данный момент существует 3 базовых концепции трансляции адресов: статическая (Static Network Address Translation), динамическая (Dynamic



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Address Translation), маскарадная (NAPT, PAT). Пример статической трансляции представлен на рис. 2.

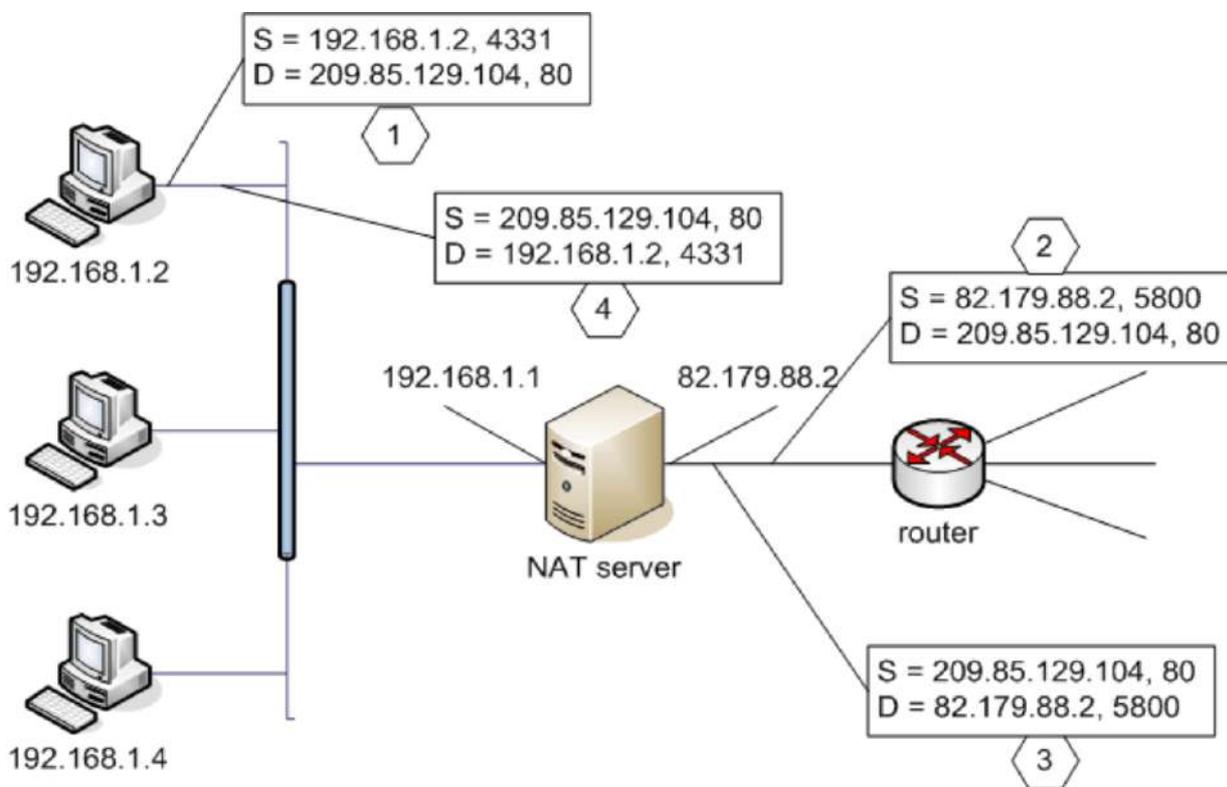
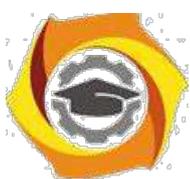


Рис. 2. Принцип работы механизма трансляции адресов

В данном примере у нас имеется локальная сеть, подключенная через NAT-прокси (например, на базе ADSL модема) с локальным адресом 192.168.1.1 и внешним интерфейсом 82.179.88.2 к глобальной сети Internet. Пользователи локальной сети 192.168.1.0/24 имеют прозрачный доступ в *Internet* через данный сервер.

Клиент с адресом 192.168.1.2 обращается к web-серверу по адресу 209.85.129.104 (www.google.com) на 80-м порту. Пакет (1) с порта 4331 отправляется на NAT-прокси (192.168.1.1). Прокси получает пакет, генерирует новый номер порта 5880, которым заменяет оригинальный номер порта отправителя, а IP-адрес клиента заменяет на адрес своего внешнего сетевого интерфейса (82.179.88.2) и отправляет пакет (2) на адрес 209.85.129.104, при этом сохраняя в таблице трансляции (см. таб. 1) данные об IP-адресах и номерах портов. Генерируя новый номер порта, NAT-прокси может выбирать произвольный номер, которого нет в таблице трансляции. Целевой web-сервер, не имея представления о том, что пакет был «проксирован», отправляет на адрес 82.179.88.2 и порт 5800 ответный пакет (3). Получив дейтаграмму, NAT-прокси по указанному в пакете IP-адресу (82.179.88.2) и номеру порта (5800) находит в таблице трансляции исходный адрес клиента (192.168.1.2) и номер порта (4331). После чего заменяет в пришедшем пакете оба поля найденными в таблице и перенаправляет пакет (4) клиенту.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Таблица трансляции сетевых адресов	
Сторона глобальной сети	Сторона локальной сети
82.179.88.2, 5800	192.168.1.2, 4331
...	...

NAT выполняет две важных функции:

1. Позволяет сэкономить IP-адреса (актуально для протокола IPv4), транслировав несколько внутренних IP-адресов в один внешний публичный IP-адрес (или в несколько внешних, но меньше, чем внутренних).
2. Позволяет предотвратить или ограничить обращение снаружи ко внутренним хостам, оставляя возможность обращения изнутри наружу. При инициации соединения изнутри сети создаётся трансляция. Ответные пакеты, поступающие снаружи, соответствуют созданной трансляции и поэтому пропускаются. Если для пакетов, поступающих снаружи, соответствующей трансляции не существует (а она может быть созданной при инициации соединения или статической), они не пропускаются.

Сложности при использовании NAT:

1. Необходимо индивидуально отслеживать работу ряда протоколов, например, ICMP (т.к. в данном протоколе отсутствует понятие портов), фрагментированные варианты IP-пакетов.
2. Не все протоколы прикладного уровня могут «преодолеть» NAT. Некоторые не в состоянии работать, если на пути между взаимодействующими хостами есть трансляция адресов. Но большинство современных реализаций NAT справляются с данной ситуацией, соответствующим образом заменяя IP-адреса не только в заголовках IP, но и на более высоких уровнях (например, в командах протоколов FTP или H.323).
3. Из-за трансляции адресов «много в один» появляются дополнительные сложности с идентификацией пользователей. Необходимо хранить полные логи трансляций.
4. Трансляция адресов достаточно сложна при использовании шифрованного IP-трафика (IPSec).

Несмотря на ряд недостатков, трансляция сетевых адресов является важной составляющей Internet. Более детальную информацию о работе NAT можно найти в соответствующих RFC и дополнительных источниках [2-4].

Маскарадная трансляция

Маскарадная трансляция адресов, известная также как NAT (Network Address Port Translation) или «Virtual Servers» – применяется в случаях, когда необходимо реализовать доступ к локальной машине из глобальной сети. Примерная схема работы NAT приведен на рис. 3.

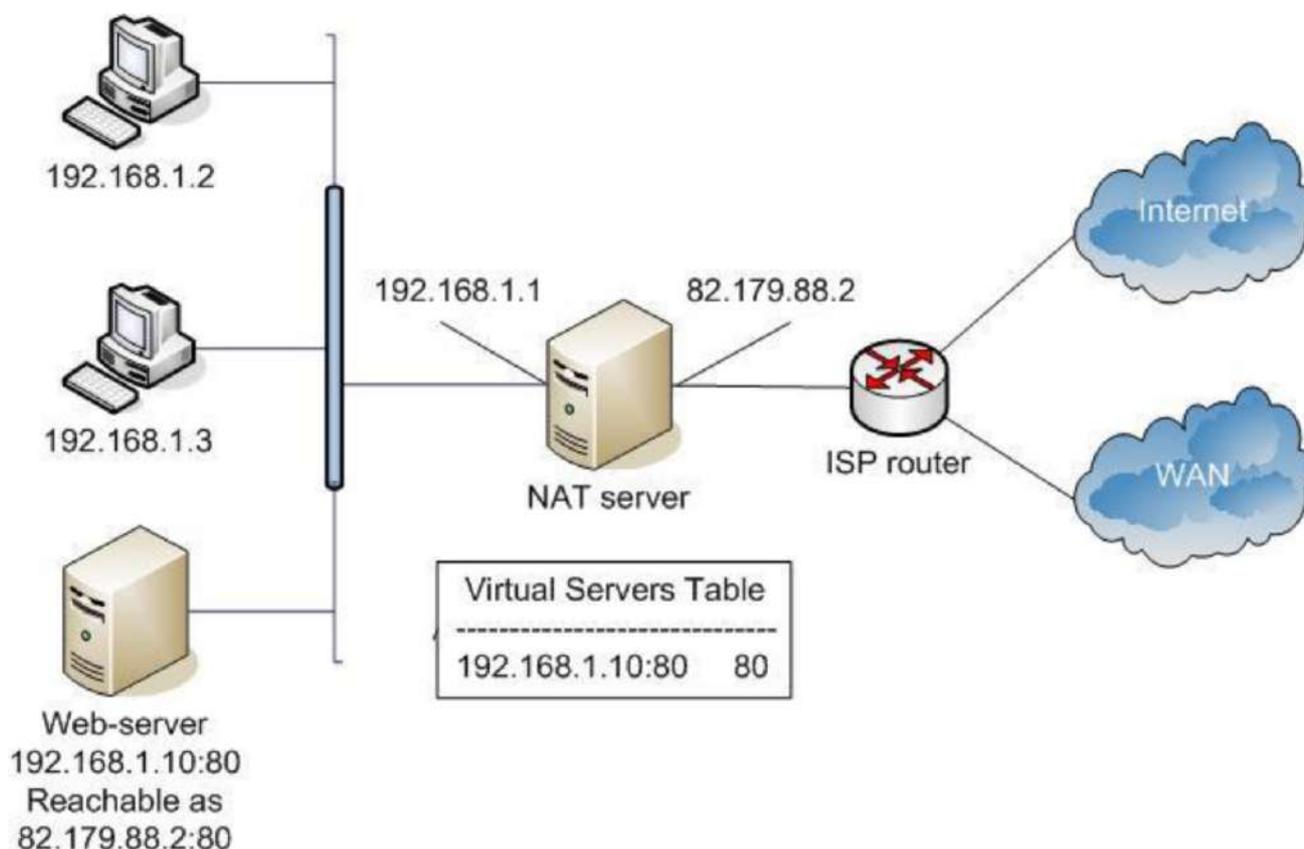
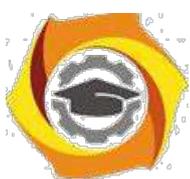


Рис. 3. Схема работы NAT

На схеме показана ситуация «публикации» локального web-сервера с адресом 192.168.1.10:80 на внешний интерфейс 82.179.88.2:80. В данном случае при включении режима маскардинга сервер открывает на интерфейсе 82.179.88.2 порт 80 на ожидание входящих подключений. При поступлении запросов на 82.179.88.2:80 NAT-сервер транслирует адреса пакета по рассмотренному выше алгоритму и перенаправляет пакеты на локальную машину с адресом 192.168.1.10 на 80-й порт. В дальнейшем работа сервера с клиентами ничем не отличается от механизма NAT.

В общем случае внешний порт и порт локального сервера могут не совпадать (port-translation).

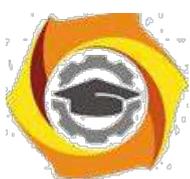
Естественно, данный механизм позволяет одновременно транслировать запросы с одного порта только на одну назначенную машину в локальной сети.

Механизм маскардинга является классическим приемом, используемым на ADSL-модемах (например, для работы в пиринговых сетях).

NAPT и NAT часто используются совместно.

Реализации Proxy-серверов

На данный момент существует большое количество практических реализаций прокси серверов. Наиболее популярными для платформы Window NT являются ISA Server, Usergate,



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

WinGate, WinRoute и др. В серверных версиях Windows также имеется встроенный достаточно функциональный NAT-маршрутизатор.

Для платформы Unix/Linux самыми известным прокси-сервером считается squid (<http://www.squid-cache.org/>).

*Настройка и тестирование NAT-прокси на базе
Windows 2003 server*

В качестве практического примера рассмотрим с вами настройку имеющегося в Windows 2003 Server встроенного NAT-прокси.

Настройку и тестирование прокси-серверов будем выполнять в сети аналогичной рис. 4. (ВНИМАНИЕ: реальные адреса сетевых интерфейсов необходимо выбирать в зависимости от учебной лаборатории! При необходимости нужно проконсультироваться с преподавателем)



Рис. 4. Простейшая сеть для настройки и тестирования NAT

Поставим задачу следующим образом: необходимо настроить на сервере NAT-прокси таким образом, чтобы для машин из подсети 192.168.198.0/4 (рис. 4, левая подсеть) осуществлялась трансляция адресов в некоторую внешнюю «подсеть» 192.168.1.0/24 (рис. 4, правая подсеть). Дополнительно необходимо настроить на машине 192.168.198.5 telnet или web сервер и посредством технологии NAPT «опубликовать» данные сервисы для доступа из вне.

Настройка NAT-прокси осуществляется при помощи оснастки «Routing and Remote Access» (см. рис. 5).

Настройка NAT достаточно проста и не требует особых пояснений. На рис. 5-6 показано, как добавить функциональность NAT-маршрутизатора на сервер. На рис. 7 показана настройка основных параметров NAT: задание внешнего интерфейса и опциональное включение встроенного сетевого фильтра. В дальнейшем для каждого из интерфейсов можно вызвать диалог настроек (рис. 8). Для внешнего интерфейса настроек больше, в частности, имеется настройка NAPT (см. рис. 9). Пример настройки NAPT для протокола telnet приведен на рис. 10.



Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»

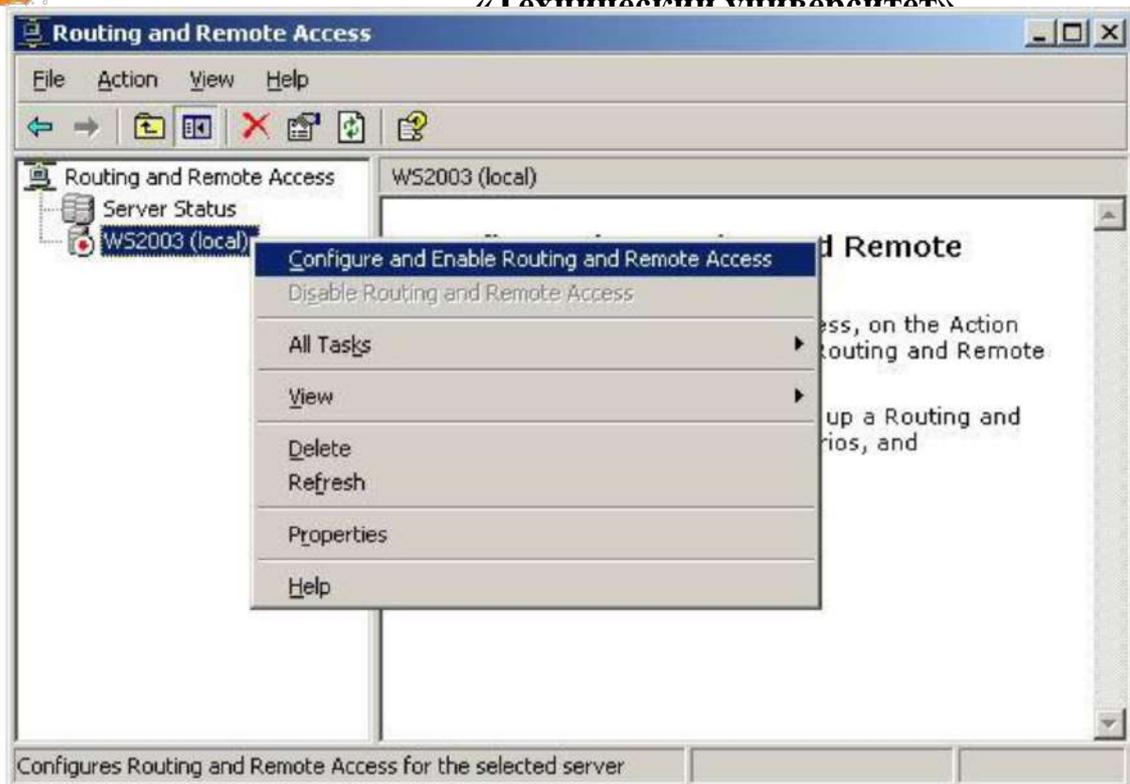


Рис. 5. Добавление NAT-прокси

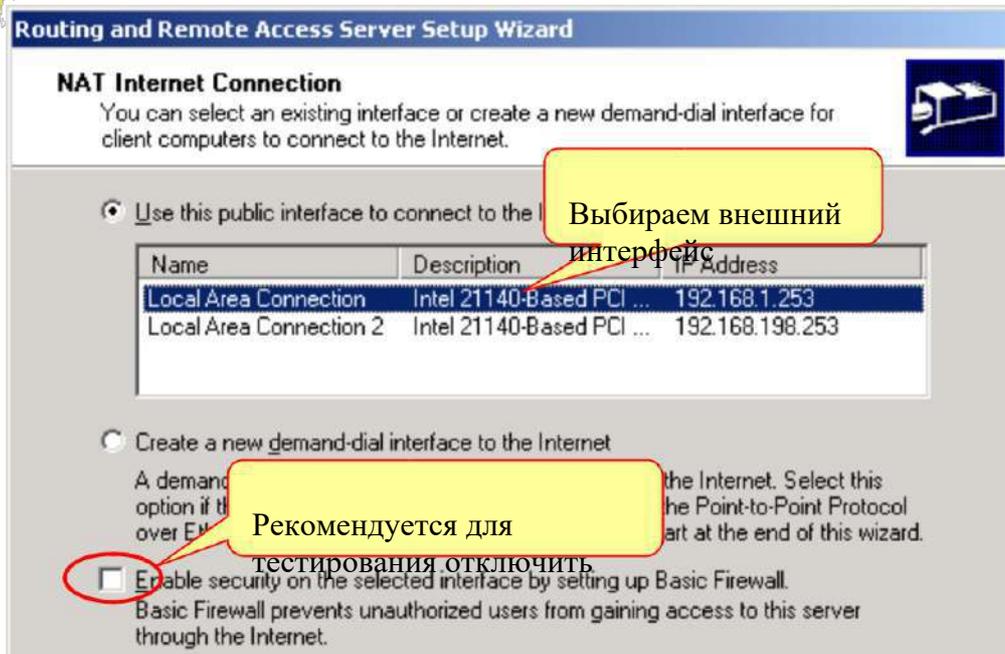


Рис. 7. Задание базовых параметров NAT

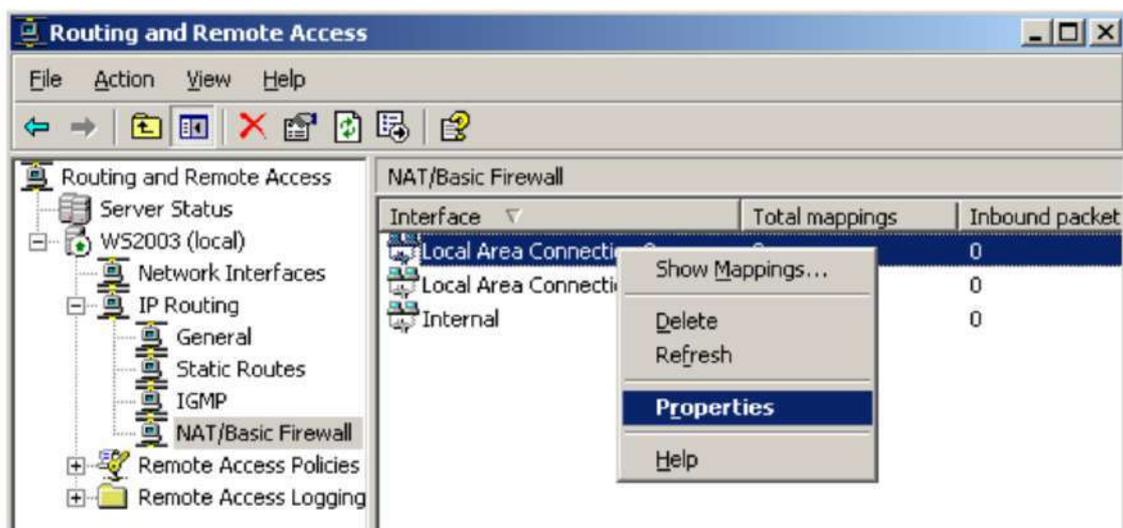


Рис. 8. Доступ к конфигурации NAT



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

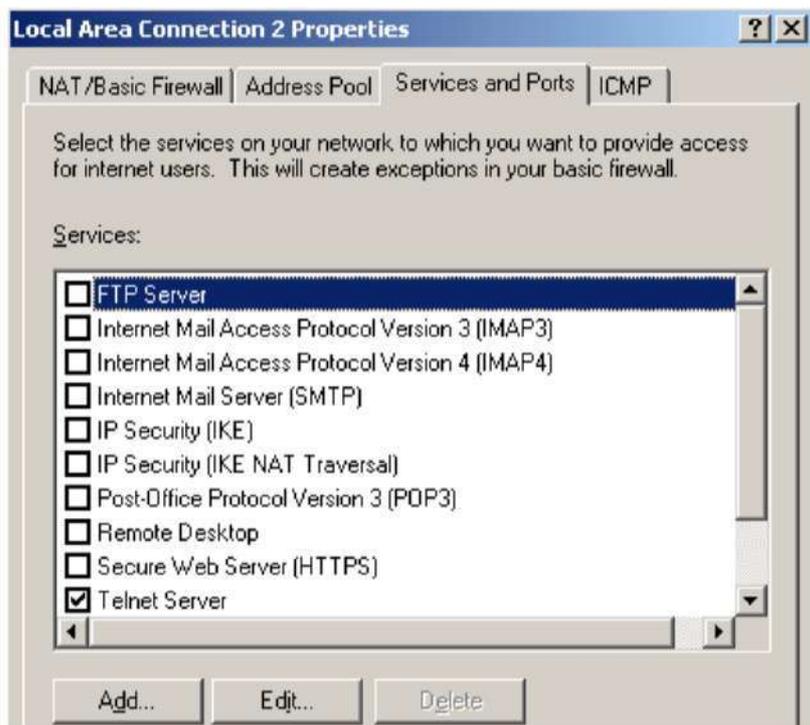


Рис. 9. Настройка NAPT для внешнего интерфейса

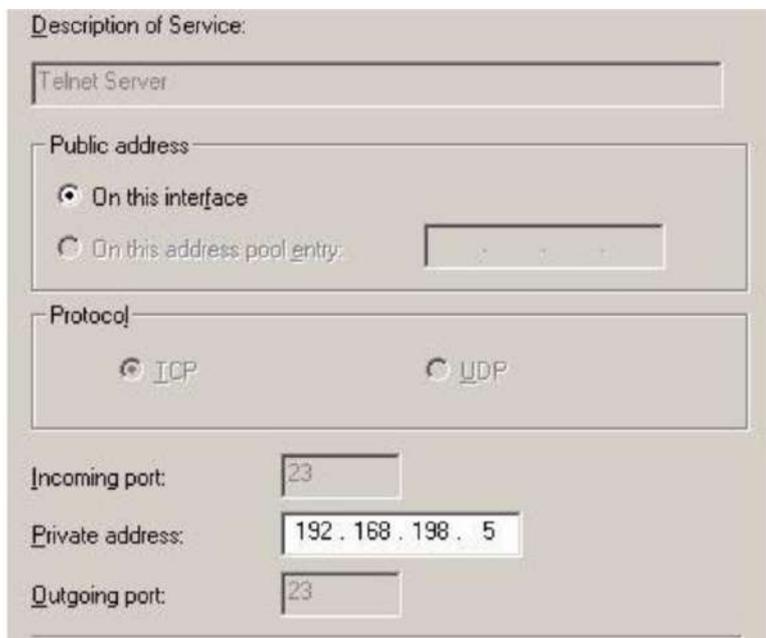
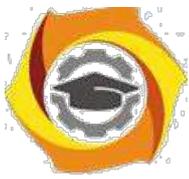


Рис. 10. Конфигурирование NAPT

Обратите внимание, что для прозрачной работы с NAT-прокси необходимо настроить маршрутизацию по умолчанию на сервер с настроенным NAT.



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Контрольные вопросы

1. Что такое прокси-сервер?
2. Перечислите цели и задачи применения прокси серверов.
3. В чем принципиальное отличие специализированного прокси-сервера от неспециализированного?
4. Что такое NAT-маршрутизатор (прокси)?
5. Объясните основной принцип работы NAT-маршрутизатора.
6. Каковы преимущества и недостатки использования NAT-маршрутизатора в сравнении со специализированными прокси-серверами?
7. Для чего применяется механизм NAPT? В чем его отличие от NAT?

Задание на лабораторную работу

1. Настроить NAT-маршрутизатор на базе Windows Server.
2. Настроить NAPT для доступа с HTTP web-серверу в локальной сети.
3. Протестировать работу маршрутизатора с использованием любого браузера.
4. С использованием sniffers, установленных как на сервере, так и на клиентских машинах, зафиксировать трафик при работе через NAT. Сопоставить полученные результаты с приведенным описанием принципа работы NAT-маршрутизатора.
5. Сделать выводы. Подготовить отчет.

Список рекомендованной литературы

1. Что такое прокси-сервер, и зачем он нужен. <http://www.eserv.ru/WhatIsProxyServer>
2. The IP Network Address Translator (NAT). RFC 1631, RFC 2663, RFC 3022
3. Cisco System Inc., «How NAT Works», <http://www.cisco.com/warp/public/556/nat-cisco.shtml>
4. Cisco System Inc., «The Trouble with NAT», http://www.cisco.com/web/about/ac123/ac147/ac174/ac182/about_cisco_ipj_archive_article09186a00800c83ec.html



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие 4 Изучение диагностических утилит для администрирования

Ознакомиться с основными сетевыми сервисами и связанными с ними портами. Научиться использовать команду netstat для контроля за состоянием локальных портов.

Взаимодействие компьютеров между собой, а также с другим активным сетевым оборудованием, в TCP/IP-сетях организовано на основе использования сетевых служб, которые обеспечиваются специальными процессами сетевой операционной системы (ОС) — демонами в UNIX-подобных ОС, службами в ОС семейства Windows и т. п. Примерами сетевых сервисов являются веб-серверы (в т.ч. сайты всемирной паутины), электронная почта, FTP-серверы для обмена файлами, приложения IP-телефонии и многое другое.

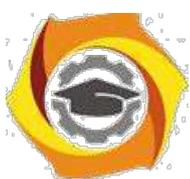
Специальные процессы операционной системы (демоны, службы) создают «слушающий» сокет и «привязывают» его к определённому порту (пассивное открытие соединения), обеспечивая тем самым возможность другим компьютерам обратиться к данной службе. Клиентская программа или процесс создаёт запрос на открытие сокета с указанием IP-адреса и порта сервера, в результате чего устанавливается соединение, позволяющее взаимодействовать двум компьютерам с использованием соответствующего сетевого протокола прикладного уровня.

Порт (англ. port) — натуральное число, записываемое в заголовках протоколов транспортного уровня модели OSI (TCP, UDP, SCTP, DCCP). Используется для определения процесса-получателя пакета в пределах одного хоста.

Номер порта для «привязки» службы выбирается в зависимости от его функционального назначения. За присвоение номеров портов определённым сетевым службам отвечает IANA. IANA (от англ. Internet Assigned Numbers Authority — «Администрация адресного пространства Интернет») — функция управления пространствами IP-адресов, доменов верхнего уровня, а также регистрирующая типы данных MIME и параметры прочих протоколов Интернета. Исполняется компанией Public Technical Identifiers, которая находится под контролем ICANN. IANA отвечает за распределение всех зарезервированных имён и номеров, которые используются в протоколах, определённых в RFC. 1 октября 2016 года официально истёк срок действия договора о выполнении функций администрации адресного пространства интернета (IANA) между ICANN и Национальным управлением по телекоммуникациям и информации (NTIA) Министерства торговли США. При этом координирующая роль в исполнении функций IANA перешла в руки международного интернет-сообщества в связи с завершением срока действия договора с правительством США.

Номера портов находятся в диапазоне 0—65535 и разделены на 3 категории:

Номера портов	Категория	Описание
0—1023	Общеизвестные порты	Номера портов назначены IANA и на большинстве систем могут быть использованы исключительно процессами системы (или пользователя root) или прикладными программами, запущенными привилегированными пользователями. Не должны использоваться без регистрации IANA. Процедура регистрации определена в разделе 19.9 RFC 4340 (англ.).



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

1024—49151	Зарегистрированные порты	Номера портов включены в каталог IANA и на большинстве систем могут быть использованы процессами обычных пользователей или программами, запущенными обычными пользователями. Не должны использоваться без регистрации IANA. Процедура регистрации определена в разделе 19.9 RFC 4340.
49152—65535	Динамические порты	Предназначены для временного использования (например, для тестирования приложений до регистрации IANA), а также в качестве клиентских (используемых для частных служб внутри закрытых сетей). Эти порты не могут быть зарегистрированы .

Локальная копия списка входит в установочный пакет сетевых операционных систем. Файл локальной копии списка обычно называется services и в различных операционных системах «лежит» в разных местах:

Windows 98/ME
C:\Windows\services

Windows NT/XP/7
C:\Windows\system32\drivers\etc\services

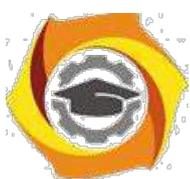
UNIX-подобные ОС
/etc/services

В большинстве операционных систем можно посмотреть состояние сетевых служб при помощи команды ([утилиты](#))

netstat -an

В ОС семейства Windows результат работы этой команды выглядит примерно так:

```
Активные подключения
Имя   Локальный адрес   Внешний адрес   Состояние
TCP   0.0.0.0:135       0.0.0.0:0      LISTENING
TCP   0.0.0.0:445       0.0.0.0:0      LISTENING
TCP   127.0.0.1:1026    0.0.0.0:0      LISTENING
TCP   127.0.0.1:12025   0.0.0.0:0      LISTENING
TCP   127.0.0.1:12080   0.0.0.0:0      LISTENING
TCP   127.0.0.1:12110   0.0.0.0:0      LISTENING
TCP   127.0.0.1:12119   0.0.0.0:0      LISTENING
TCP   127.0.0.1:12143   0.0.0.0:0      LISTENING
TCP   192.168.0.16:139  0.0.0.0:0      LISTENING
TCP   192.168.0.16:1572 213.180.204.20:80 CLOSE_WAIT
TCP   192.168.0.16:1573 213.180.204.35:80 ESTABLISHED
UDP   0.0.0.0:445       *: *
UDP   0.0.0.0:500       *: *
UDP   0.0.0.0:1025     *: *
UDP   0.0.0.0:1056     *: *
UDP   0.0.0.0:1057     *: *
UDP   0.0.0.0:1066     *: *
UDP   0.0.0.0:4500     *: *
UDP   127.0.0.1:123     *: *
UDP   127.0.0.1:1900    *: *
UDP   192.168.0.16:123 *: *
UDP   192.168.0.16:137 *: *
UDP   192.168.0.16:138 *: *
UDP   192.168.0.16:1900 *: *
```



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

В UNIX-подобных ОС результат работы команды netstat -an имеет примерно такой вид:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp    0      0 0.0.0.0:37             0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:199           0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:2601          0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:3306          0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:2604          0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:2605          0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:13            0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:179           0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:21            0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:22            0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:1723          0.0.0.0:*              LISTEN
tcp    0      0 0.0.0.0:254:1723      10.0.0.243:2441        ESTABLISHED
tcp    0      0 192.168.19.34:179     192.168.19.33:33793    ESTABLISHED
tcp    1      0 192.168.18.250:37     192.168.18.243:3723    CLOSE_WAIT
tcp    0      0 10.0.0.254:1723       10.0.0.218:1066        ESTABLISHED
tcp    1      0 192.168.18.250:37     192.168.18.243:2371    CLOSE_WAIT
tcp    0      0 10.0.0.254:1723       10.0.0.201:4346        ESTABLISHED
tcp    0      0 10.0.0.254:1723       10.0.0.30:2965         ESTABLISHED
tcp    0      48 192.168.19.34:22      192.168.18.18:43645    ESTABLISHED
tcp    0      0 10.0.0.254:38562      10.0.0.243:22          ESTABLISHED
tcp    0      0 10.50.1.254:1723      10.50.1.2:57355        ESTABLISHED
tcp    0      0 10.50.0.254:1723      10.50.0.174:1090       ESTABLISHED
tcp    0      0 192.168.10.254:1723   192.168.13.104:65535   ESTABLISHED
tcp    0      0 10.0.0.254:1723       10.0.0.144:65535       ESTABLISHED
tcp    0      0 10.0.0.254:1723       10.0.0.169:2607        ESTABLISHED
tcp    0      0 10.0.0.254:1723       10.0.0.205:1034        ESTABLISHED
udp    0      0 0.0.0.0:1812          0.0.0.0:*
udp    0      0 0.0.0.0:1813          0.0.0.0:*
udp    0      0 0.0.0.0:161           0.0.0.0:*
udp    0      0 0.0.0.0:323           0.0.0.0:*
udp    0      0 0.0.0.0:123           0.0.0.0:*
raw    0      0 192.168.10.254:47     192.168.13.104:*      1
raw    0      0 10.0.0.254:47         10.0.0.120:*           1
raw    0      0 10.10.204.20:47       10.10.16.110:*         1
raw    0      0 192.168.10.254:47     192.168.11.72:*        1
raw    0      0 10.0.0.254:47         10.0.0.144:*           1
raw    0      0 10.0.0.254:47         10.0.0.205:*           1
raw    0      0 10.50.0.254:47        10.50.0.174:*          1
raw    0      0 10.0.0.254:47         10.0.0.170:*           1
raw    0      0 10.0.0.254:47         10.0.0.179:*           1
```

Состояние (State) LISTEN (LISTENING) показывает пассивно открытые соединения («слушающие» сокеты). Именно они и предоставляют сетевые службы. ESTABLISHED — это установленные соединения, то есть сетевые службы в процессе их использования.

В случае обнаружения проблем с той или иной сетевой службой, для проверки её доступности используют различные средства диагностики, в зависимости от их наличия в данной ОС.

Одно из самых удобных средств — команда (утилита) tcptraceroute (разновидность traceroute), которая использует TCP-пакеты открытия соединения (SYN|ACK) с указанным сервисом (по умолчанию — web-сервер, порт 80) интересующего хоста и показывает информацию о времени прохождения данного вида TCP-пакетов через маршрутизаторы, а также информацию о доступности службы на интересующем хосте, либо, в случае проблем с доставкой пакетов — в каком месте пути они возникли.

В качестве альтернативы можно использовать отдельно traceroute для диагностики маршрута доставки пакетов (недостаток — использование UDP-пакетов для диагностики) и telnet или netcat на порт проблемной службы для проверки её отклика.

Traceroute — это служебная компьютерная программа, предназначенная для определения маршрутов следования данных в сетях TCP/IP. Traceroute может использовать



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

разные протоколы передачи данных в зависимости от операционной системы устройства. Такими протоколами могут быть UDP, TCP, ICMP или GRE. Компьютеры с установленной операционной системой Windows используют ICMP-протокол, при этом операционные системы Linux и маршрутизаторы Cisco — протокол UDP.

Traceroute входит в поставку большинства современных сетевых операционных систем. В системах Microsoft Windows эта программа носит название tracert, а в системах GNU/Linux, Cisco IOS и Mac OS — traceroute.

Рассмотрим пример работы программы в операционной системе Windows. Программа tracert выполняет отправку данных указанному узлу сети, при этом отображая сведения о всех промежуточных маршрутизаторах, через которые прошли данные на пути к целевому узлу. В случае проблем при доставке данных до какого-либо узла программа позволяет определить, на каком именно участке сети возникли неполадки. Необходимо отметить, что программа работает только в направлении от источника пакетов и является весьма грубым инструментом для выявления неполадок в сети. В силу особенностей работы протоколов маршрутизации в сети Интернет, обратные маршруты часто не совпадают с прямыми, причём это справедливо для всех промежуточных узлов в трейсе. Поэтому ICMP ответ от каждого промежуточного узла может идти своим собственным маршрутом, затеряться или прийти с большой задержкой, хотя в реальности с пакетами, которые адресованы конечному узлу, этого не происходит. Кроме того, на промежуточных маршрутизаторах часто стоит ограничение числа ответов ICMP в единицу времени, что приводит к появлению ложных потерь.

Для определения промежуточных маршрутизаторов traceroute отправляет целевому узлу серию ICMP-пакетов (по умолчанию 3 пакета), с каждым шагом увеличивая значение поля TTL («время жизни») на 1. Это поле обычно указывает максимальное количество маршрутизаторов, которое может быть пройдено пакетом. Первая серия пакетов отправляется с TTL, равным 1, и поэтому первый же маршрутизатор возвращает обратно ICMP-сообщение «time exceeded in transit», указывающее на невозможность доставки данных. Traceroute фиксирует адрес маршрутизатора, а также время между отправкой пакета и получением ответа (эти сведения выводятся на монитор компьютера). Затем traceroute повторяет отправку серии пакетов, но уже с TTL, равным 2, что заставляет первый маршрутизатор уменьшить TTL пакетов на единицу и направить их ко второму маршрутизатору. Второй маршрутизатор, получив пакеты с TTL=1, так же возвращает «time exceeded in transit».

Процесс повторяется до тех пор, пока пакет не достигнет целевого узла. При получении ответа от этого узла процесс трассировки считается завершённым.

На конечном хосте IP-датаграмма с TTL = 1 не отбрасывается и не вызывает ICMP-сообщения типа срок истёк, а должна быть отдана приложению. Достижение пункта назначения определяется следующим образом: отсылаемые traceroute датаграммы содержат UDP-пакет с заведомо неиспользуемым номером порта на адресуемом хосте. Номер порта будет равен $33434 + (\text{максимальное количество транзитных участков до узла}) - 1$. В пункте назначения UDP-модуль, получая подобные датаграммы, возвращает ICMP-сообщения об ошибке «порт недоступен». Таким образом, чтобы узнать о завершении работы, программе traceroute достаточно обнаружить, что поступило ICMP-сообщение об ошибке этого типа.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Использование утилиты `tracert`

Запуск программы производится из командной строки. Для этого вы должны войти в неё. Для операционных систем семейства Windows существует несколько способов запуска командной строки:

Пуск — Выполнить — В графе «Открыть» написать «`cmd`» и нажать Ок. Сочетание клавиш Win (кнопка с логотипом Windows) + R (должны быть нажаты одновременно) — В графе «Открыть» написать «`cmd`» и нажать Ок.

Пуск — Все программы (или просто «Программы», зависит от версии операционной системы) — Стандартные — Командная строка.

В открывшемся окне написать:

```
tracert example.net
```

Где *tracert* — обращение к программе, а example.net — любой домен или IPv4 адрес.

```
C:\Documents and Settings\Administrator>tracert ru.wikipedia.org

Трассировка маршрута к rr.esams.wikimedia.org
[91.198.174.2] с максимальным числом прыжков 30:

 1  1 ms  <1 ms  <1 ms  vpn4.kras.gldn [10.10.1.14]
 2  2 ms  <1 ms  <1 ms  C7604-BRAS4-FTTB.ranetka.ru [80.255.150.41]
 3  1 ms  1 ms    4 ms  C76-External.ranetka.ru [80.255.128.162]
 4  1 ms  <1 ms  <1 ms  pe-1.Krasnoyarsk.gldn.net [195.239.173.37]
 5  79 ms  79 ms  98 ms  cat01.Stockholm.gldn.net [194.186.157.62]
 6  131 ms  131 ms  132 ms  ams-ix.2ge-2-1.br1-knams.wikimedia.org [195.69.145.176]
 7  131 ms  131 ms  131 ms  te-8-2.csw1-esams.wikimedia.org [91.198.174.254]
 8  133 ms  134 ms  133 ms  rr.esams.wikimedia.org [91.198.174.2]

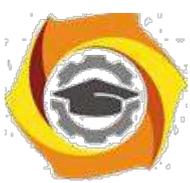
Трассировка завершена.
```

В [UNIX/Linux](#) системах существуют режимы, в которых запуск программы возможен только от имени суперпользователя root (администратора). К числу этих режимов относится важный режим трассировки с помощью [ICMP](#) (ключ `-I`).

Во всех остальных случаях, `traceroute` может работать от имени обычного рядового пользователя. При этом, параметры по умолчанию различаются от дистрибутива к дистрибутиву, хотя в справке традиционно пишется ключ `-U` ([UDP](#)) в качестве такового. В отдельных RedHat-based дистрибутивах фактически в качестве умолчания используется `-I`, поэтому в случае, если команда из следующего примера выдаст сообщение о недостатке прав, попробуйте явно указать ключ `-U`.

```
[user@localhost ~]$ traceroute www.ru
traceroute to www.ru (194.87.0.50), 30 hops max, 38 byte packets
 1  mygateway.ar7                (192.168.1.1)    0.777 ms  0.664 ms  0.506 ms
 2  L0.ghsdr04                    (213.227.224.91) 15.661 ms 15.867 ms 31.426 ms
 3  213.227.224.1                  (213.227.224.1) 16.797 ms 18.221 ms 16.756ms
 4  dg                             (213.186.216.161) 53.068 ms 39.163 ms 38.283 ms
 5  br13                           (213.186.193.43) 40.156 ms 39.768 ms 42.803 ms
 6  aggr                           (62.221.40.169) 37.884 ms 38.712 ms 37.207 ms
 7  edge-3GE-216dot1q.kiev.ucomline.net (213.130.30.182) 39.723 ms 38.039 ms 41.261 ms
 8  ae0-202.RT771-001.kiv.retn.net (81.222.15.1) 40.029 ms 37.088 ms 40.039 ms
 9  ae0-3.RT502-001.msk.retn.net (81.222.15.1) 128.932 ms 122.043 ms 121.612 ms
10  GW-Demos.retn.net             (81.222.8.46) 120.023 ms 121.135 ms 119.493 ms
11  iki-1-vl10.demos.net          (194.87.0.83) 119.074 ms 119.784 ms 123.607 ms
12  www.ru                        (194.87.0.50) 120.358 ms 122.545 ms 119.399 ms
```

Часто встречается заблуждение, что `traceroute`, как и [ping](#), работает только по протоколу ICMP. В связи с этим начинающие администраторы, разрешив в файрволе протокол ICMP, получают рабочий `ping` и нерабочий `traceroute`. Для исправления такой ситуации необходимо дополнительно разрешить в файрволе UDP-пакеты на порты выше



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

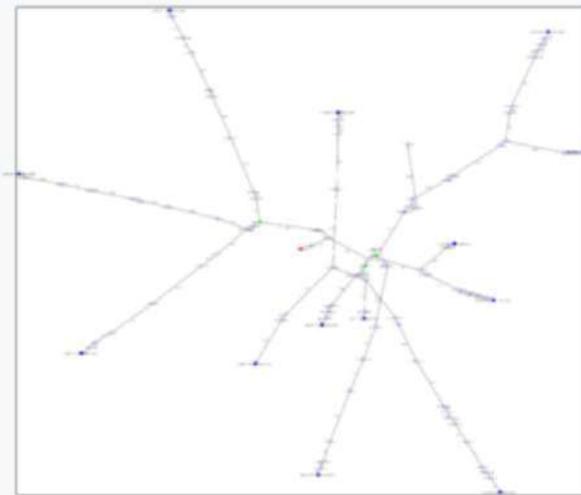
33434 (в некоторых источниках указано, что достаточно указать диапазон портов от 33434 до 33534).

`tracpath` — похожая на `tracroute` программа, но может строить асимметричные трассы и имеет некоторые другие отличия.

`mtr` — Интерактивная программа, способная постоянно выводить обновлённую статистику по трассе.

`tracemip` — программа, позволяющая выполнять трассировку пути на несколько хостов сразу и представить полученные данные в виде графической карты.

`tcptraceroute (tracetcp)` — аналогичная `tracroute` программа, но предназначена для диагностики TCP соединений; вместо UDP-пакетов использует TCP-пакеты открытия соединения (SYN|ACK) с указанным сервисом (по умолчанию — web-сервер, порт 80) интересующего хоста; в результате получаем информацию о времени прохождения данного вида TCP-пакетов через маршрутизаторы и информацию о доступности сервиса на интересующем хосте, либо, в случае проблем с доставкой пакетов — видим, в каком месте трассы они возникли.



Полученная с помощью `tracemip` графическая карта трассировки пути на корневые серверы DNS с хоста

ЗАДАНИЕ

1. Найдите локальную версию списка распределения портов на своём компьютере.
2. При помощи утилиты `netstat` проверьте состояние портов на своём компьютере.
3. При помощи утилиты `tracert` проведите трассировку узлов `google.com`, `yandex.ru`, `rnd.mts.ru` и ещё двум узлам на своё усмотрение. Дайте интерпретацию полученным результатам.
4. Ответьте на 2 вопроса из приведенного ниже списка. Номера вопросов определите остаток от деления вашего номера по журналу на 11 и следующий (предыдущий, если следующего нет) вопрос
5. Оформите отчёт по лабораторной работе с включением в него скриншотов выполняемых операций, пояснений сделанного и ответов на вопросы.

Контрольные вопросы

1. Какую роль выполняет ОС при организации сетевой работы?
2. Что понимают под термином «сетевая оболочка»?
3. Что означает термин «сетевой сервис»?
4. Может ли использоваться сетевая оболочка в сетевых операционных системах?



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

5. В чем сходство и различие сетевой и распределенной ОС?
6. Какие два признака характеризуют термин «сетевая ОС»?
7. Какие основные функциональные компоненты можно выделить в сетевой ОС?
8. Дайте определение сетевой службы.
9. Является ли сетевая служба только клиентской или только серверной?
10. Назовите принципиальное различие между клиентом и сервером.
11. Перечислите основные требования, предъявляемые к современным операционным системам



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

**Лабораторная работа 4 Установка, настройка и
конфигурирование Web-сервера IIS**

Теоретическая часть

Понятие серверного приложения

В практикуме рассматривается служба Интернета WWW (World Wide Web — Всемирная паутина), которая управляет передачей гипертекстовых страниц и регламентируется протоколом HTTP (Hyper Text Transfer Protocol). На компьютере-сервере, поддерживающем эту службу, должна быть установлена программа, которая также называется Web-сервером (кратко — сервером). В специальной папке компьютера-сервера хранятся Web-страницы, которые могут быть запрошены с компьютера-клиента общеизвестными способами (например, набором имени страницы в адресной строке браузера или активизацией гиперссылки).

Web-страницы могут быть двух видов: страницы, написанные на языке HTML (с расширением .htm или .html) и серверные приложения. HTML-страницы отсылаются сервером клиенту без предварительной обработки. Серверные приложения создаются с помощью специальных технологий; расширение файла приложения показывает, какая именно технология использовалась: .asp — ASP-технологии, .aspx — технологии ASP.NET,

.php — PHP-технологии, и т.п. При запросе клиентом серверное приложение преобразуется Web-сервером в HTML-файл, и этот файл отсылается клиенту. Интерпретация HTML-файлов (независимо от того, был он получен из серверного приложения или нет) осуществляется на компьютере-клиенте программой-браузером. Серверное

приложение может быть программой на алгоритмическом языке, или текстом на языке

HTML, в который включены фрагменты (называемые сценариями или скриптами) на алгоритмическом языке. Обработка серверного приложения Web-сервером представляет собой трансляцию серверного приложения в HTML-текст. В процессе трансляции могут быть использованы данные из запроса клиента; например, сведения компьютеру или браузеру клиента, а также данные, посылаемые клиентом в соответствии с решаемой задачей. Эти данные могут существенно повлиять на вид ответной Web-страницы. Таким образом, серверные приложения представляют собой динамические, интерактивные Web-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

страницы, формируемые на сервере. Трансляцию серверных приложений осуществляет специальный программный модуль, входящий в состав Web-сервера или подключенный к нему. Такой модуль, включенный в IIS-сервер и осуществляющий трансляцию asp-приложений, называется Script host.

Заметим, что существуют технологии формирования интерактивных Web-страниц на компьютере-клиенте с помощью браузера. Настоящий практикум посвящен именно серверным приложениям.

Передача данных задачи пользователя на сервер

Основным способом передачи данных от клиента к серверу является использованием

HTML-форм. Формы содержат интерфейсные элементы (элементы управления). Примеров

таких элементов могут быть текстовые окна для ввода данных, списки (селекторы) для выбора

значений, флажки, радиокнопки. Предполагается, что читатель знаком с кодированием этих

элементов на языке HTML. С каждым элементом формы связаны имя (атрибут элемента

NAME) и значение (как правило, атрибут VALUE). Для передачи данных на сервер форма

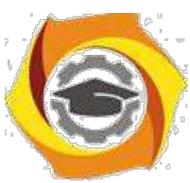
обязательно должна содержать элемент управления submit. Этот элемент представляет собой

кнопку, при нажатии которой данные формы автоматически включаются



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

в запрос и запрос отправляется на сервер. При использовании формы для отправки данных на сервер тэг <FORM> обязательно



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

должен содержать два атрибута — ACTION и METHOD. В атрибуте ACTION записывается URL серверного приложения, формирующего ответную Web-страницу. При нажатии кнопки submit указанный URL включается в стартовую строку отправляемого запроса. Если серверное приложение находится в том же виртуальном каталоге сервера, из которого была вызвана Web-страница, содержащая форму, или в подчиненных ему папках, то вместо полного URL можно указать путь к серверному приложению относительно виртуального каталога.

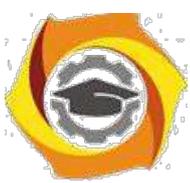
Атрибут METHOD определяет метод передачи данных от клиента к серверу. В нашем практикуме мы будем рассматривать два метода — GET и POST. Метод GET обеспечивает присоединение данных формы к URL серверного приложения через знак вопроса (?), и расширенный таким образом URL, как уже было сказано выше, при нажатии кнопки submit включается в стартовую строку запроса. Метод POST означает, что данные формы включаются в тело запроса. Для обоих методов включаемые в запрос данные имеют вид: имя элемента формы = значение элемента. Такие пары вида имя = значение отделяются друг от друга символом «&». Отметим, что все символы, входящие в имя и значение, кроме латинских букв и пробела, при включении в запрос автоматически заменяются своим шестнадцатеричным кодом, перед которым ставится символ «%». Латинские буквы не подлежат перекодировке, а пробел заменяется символом «+». Сравним два рассмотренных метода передачи данных. Метод POST меньше, чем GET, ограничивает объем передаваемых данных и предпочтительней с точки зрения безопасности (так как данные нельзя прочитать в адресной строке браузера). Однако, используя GET, можно не только передавать данные полей формы, но и «вручную» присоединить данные к URL (после символа «?»), например, при запросе страницы из адресной строки браузера или в гиперссылке.

Структура простейшего asp-приложения

Файл asp-приложения обязательно имеет расширение .asp. Он содержит текст на языке HTML, в который вставлены сценарии на алгоритмическом языке. Сценарий ограничен парами символов «<%» и «%>», первая пара играет роль открывающей, а вторая — закрывающей скобки. Существуют другие способы ограничения сценариев, они будут рассмотрены позже. Алгоритмический язык сценариев указывается в инструкции <%@ Language = язык%>. Эта инструкция располагается в первой строке файла и относится ко всем включенным в него сценариям. Стандартным языком сценариев является VBScript; при его использовании указанная инструкция может быть упущена. Код, полученный в результате интерпретации сценария, вставляется на место сценария в HTML-файле. Конструкции языка VBScript, используемые в данном пособии, приведены в Приложении 1.

ASP-технология предоставляет широкий спектр возможностей для извлечения данных из запроса, поступившего на сервер, и формирования ответной Web-страницы. Основные возможности будут рассмотрены в настоящем пособии. При создании ответной Web-страницы очень часто используется оператор Response.Write, который выводит строку символов в формируемый HTML-текст. Отметим, что Response — это объект, содержащий основные средства формирования динамической Web-страницы, а Write — метод этого объекта. Приведенное ниже простейшее приложение выводит фразу «HELLO!» в окно браузера:

```
<%@ Language = VBScript%>.  
<HTML>  
<HEAD> <TITLE> Первый пример</TITLE>  
</HEAD> <BODY>  
  <% Response.Write "HELLO!" %>  
</BODY>  
</HTML>
```



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Если сценарий состоит из вывода одной строки, то его можно сделать еще короче, заменив оператор Response.Write символом «=>». Так, сценарий в приведенном выше примере можно заменить следующим: <% = "HELLO!" %>. Рассмотрим приложение, которое выводит в окно браузера время формирования ответной

Web-страницы (Time — встроенная функция VBScript):

```
<HTML>
<HEAD> <TITLE> Узнай время</TITLE>
</HEAD> <BODY>
  Точное время на стороне сервера:<%=Time
  %> </BODY>
</HTML>
```

Обратите внимание, что для вызова серверного приложения надо обязательно сформировать запрос от клиента к серверу, даже если вы работаете в отладочном режиме «обратной петли», и Ваш компьютер является одновременно и сервером, и клиентом. Например, можно набрать адрес серверного приложения в окне браузера. Серверные приложения нельзя вызывать, как обычные приложения, двойным кликом на пиктограмме в окне папок Проводник или Мой компьютер.

Рассмотрим простые примеры asp-приложений, обрабатывающих данные HTML-форм. Ниже приведен файл (назовем его concat.htm), который формирует на стороне клиента простейший запрос, обеспечивающий передачу на сервер значений двух строк из полей редактирования формы в окне браузера:

```
<HTML>
<HEAD><TITLE>    Ввод значений    a    и b    для передачи    на сервер</TITLE>
</HEAD>
<BODY>
<FORM ACTION="concat.asp" METHOD=POST NAME="forma">
  Первое значение <input type="text" name="a" value=""> <br> Второе
  значение <input type="text" name="b" value=""> <br>
<input type="submit" name="plus"
  value="результат"> </FORM>
</BODY>
</HTML>
```

Обратите внимание, что этот файл не является asp-приложением, о чем говорит и его расширение. В теге <FORM> указано имя серверного приложения (concat.asp) для обработки этих строк. Это приложение осуществляет конкатенацию (сцепление) полученных строк и формирует ответную Web-страницу, содержащую поля редактирования с исходными значениями строк и результатом сцепления. Содержание файла concat.asp:

```
<HTML>
<HEAD> <TITLE> Результат конкатенации<TITLE > </HEAD >
<BODY>
<%
  a=Request("a") ' в переменную a считывается строка из элемента a
  b=Request("b") ' в переменную b считывается строка из элемента b
  c=a+b ' c принимает значение результата сцепления a и b
  ' ниже в поля редактирования выводятся значения a, b, c
  %>
<FORM>
  Первое слагаемое <input type="text" value=<% Response.Write a %> > <br>
  Второе слагаемое <input type="text" value=<% Response.Write b %> >
  <br> Сумма <input type="text" value=<% Response.Write c %> > <br>
</FORM>
```



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

</BODY>

</HTML>

Заметим, что оператор `имя=Request("имя")` извлекает значение данного из запроса. Имя в правой части оператора (в данной ситуации кавычки обязательны!) — это имя элементы формы, значение которого передано в запросе; имя в левой части — это имя ячейки оперативной памяти сервера. Естественно, имена в левой и правой частях могут не совпадать. При выборе одинаковых имен легче читается программный код.

Операция «+» в языке VBScript (как, например, и в языке Pascal) выполняется в зависимости от контекста: над строками как конкатенация, над числами как сложение. Если в файле `concat.asp` оператор `c=a+b` заменить оператором `c=Cdbl(a)+Cdbl(b)`, то `c` будет не результатом сцепления строк, а суммой чисел `a` и `b`, так как `Cdbl` — это функция преобразования данного в вещественное число. Если, кроме того, в сценарии `<% Response.Write c %>` `c` изменить на `CStr(c)`, то серверное приложение будет выводить сумму двух чисел (`CStr` — функция преобразования в строку).

4. 1.4. Установка IIS-сервера

IIS-сервер – серверный программный комплекс, входящий в состав операционной системы Windows (начиная с Windows 2000). В этот комплекс входит Web-сервер и ASP-технология подготовки серверных приложений.

Для установки IIS-сервера надо выполнить следующие шаги:

Вызовите окно Мастера компонентов Windows (*Пуск/ Настройка/Панель управления /Установка и удаление программ/ Добавление и удаление компонентов Windows*). Окно мастера приведено на рис.1. В списке компонентов выделите *Internet Information Services (IIS)*

Контрольные вопросы

1. Что понимают под программным обеспечением сетей ЭВМ?
2. Что дает предприятию использование компьютерных сетей?
3. Классификация сетевого программного обеспечения.
4. Что называют операционной системой?
5. Что входит в группу прикладного программного обеспечения?
6. По каким критериям можно охарактеризовать сетевую операционную систему?
7. Что называют сетевым драйвером?
8. Что называют сетевым протоколом?
9. Перечислить сетевые операционные системы.
10. Что такое сетевые службы?
11. Что называют стандартным программным обеспечением ЭВМ?
12. Что такое технология «клиент-сервер»?



Практическое занятие 5 Установка, настройка и конфигурирование Web-сервера Apache

Цель работы: настройка и администрирование web-сервера *Apache* под ОС Linux.

Практическое задание выполняется в локальной сети на рабочей станции с операционной системой Linux 7 или более поздней. В лабораториях кафедры операционная система Linux работает на компьютерах под управлением программного пакета VMware. Этот пакет позволяет создавать так называемые «виртуальные машины» – мнимые компьютеры, не зависящие от выполняющейся в текущее время на данном компьютере операционной системы (ОС). Для запуска ОС Linux необходимо запустить VMware на рабочей станции, выбрать из списка требуемую операционную систему и нажать кнопку «Power ON».

*После окончания загрузки для входа в систему необходимо использовать имя пользователя **root**, пароль – **rootuser**.*

Порядок выполнения лабораторной работы

Подготовка и допуск к работе. К выполнению лабораторной работы допускаются студенты, которые подготовились к работе и имеют не более двух невыполненных предыдущих работ.

Перед работой студент должен:

- предъявить преподавателю полностью оформленный отчет о предыдущей работе;
- ответить на вопросы преподавателя.

К работе не допускаются студенты, которые не выполнили одно из вышеперечисленных требований.

Отчёт по работе должен содержать:

- текст задания;
- перечень всех использованных в лабораторной работе команд и инструкций;
- содержимое конфигурационных файлов (без комментариев!!!);
- отрывок из log файлов, демонстрирующий обращения в web-серверу;
- вывод по работе.

Администрирование и установка WEB-СЕРВЕРА

Самый распространенный web-сервер в мире – это **Apache**. По данным компании Netcraft (<http://www.netcraft.com/Survey/>) общее число web-узлов, работающих под его управлением, к концу 1998 г. достигало 2 млн (55 % общего числа узлов) и это число постоянно растет. Для сравнения – на долю серверов Microsoft приходится 25 %, Netscape – 7 %. Будучи бесплатной и открытой программой, предназначенной для бесплатных же Unix-систем (FreeBSD, Linux и др.), **Apache** по функциональным возможностям и надежности не уступает коммерческим серверам, а широкие возможности конфигурирования позволяют настроить его для работы практически с любой конкретной системой. Существуют локализации сервера для различных языков, в том числе и для русского.

Список директив (настроек), используемых для конфигурации



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

web-сервера

Каждая строка конфигурационного файла, кроме строк, начинающихся с #, описывает какую-то директиву конфигурации и ее значение. Строки, начинающиеся с # – это комментарии. Более подробное описание всех директив находится в файле 12.html методического пакета к лабораторной работе, находящегося в методическом разделе сервера кафедры «Автоматика и системотехника». Описание директив, необходимых для выполнения лабораторной работы, приведено в табл 1.

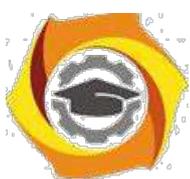
Таблица 1

Описание основных директив конфигурирования web-сервера

Наименование директивы	Описание
ServerType standalone	Показывает тип запуска программы <i>httpd</i> – она может вызываться через <i>inetd (inetd)</i> или как отдельный демон (<i>standalone</i>). По умолчанию вызывается как отдельный демон
Port 80	Порт, по которому http-сервер будет получать запросы. По умолчанию 80
HostnameLookups off	Определяет, будут ли записываться в лог-файлы имена (<i>on</i>) или только <i>ip</i> -адреса (<i>off</i>) хостов, просматривающих сайт. По умолчанию <i>off</i>
User nobody Group nobody	Определяет то, под каким пользователем будет запускаться <i>httpd</i> . Для доступа к 80 порту <i>httpd</i> должен запускаться под <i>root</i> 'ом. Если в директивах User и Group указаны другие пользователи, то <i>httpd</i> все свои действия выполняет с правами этих пользователей, а не <i>root</i> 'а
ServerAdmin root@localhost	Здесь указывается <i>e-mail</i> администратора, отвечающего за работу сервера
ServerRoot /etc/httpd	Каталог, в котором находятся все конфигурационные файлы, лог-файлы и модули. Все пути, которые используются далее, указаны относительно этого каталога
ErrorLog logs/error_log	Местонахождение файла логов об ошибках сервера
LogLevel warn	Уровень сообщений в файлах логов. Возможные значения: <i>debug, info, notice, warn, error, crit, alert, emerg</i>
LoadModule имя_модуля modules/имя_файла_модуля.so	Команда используется для загрузки различных модулей сервера Apache
ClearModuleList	Очистка списка модулей
AddModule имя_файла_модуля.c	Добавления модулей в список модулей

Продолжение табл. 1

Наименование директивы	Описание
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User- Agent}i\" combined	Описание различных форматов записи логов. Имя формата используется в директиве <i>CustomLog</i>
LogFormat "%h %l %u %t \"%r\" %>s %b\" common	

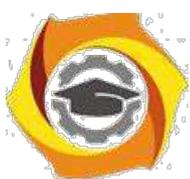


Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

LogFormat "%{Referer}i -> %U" referer LogFormat "%{User-agent}i" agent LogFormat "формат" имя	
CustomLog logs/access_log common	Задаёт формат файла логов запросов. С помощью <i>CustomLog</i> можно задать файл логов любого формата и имени
PidFile /var/run/httpd.pid	Имя файла, в котором хранится <i>PID</i> запущенного <i>httpd</i> . Используется для останова сервера и перезапуска
UseCanonicalName on	Разрешает использование коротких имен в <i>URL</i> 'ах. Удобно при использовании в Intranet-серверах. Если включено (<i>on</i>), преобразует, например строку <i>http://www/stat</i> в строку <i>http://www.мой-домен.ru/stat</i> , если отключено (<i>off</i>), преобразования не происходит
Timeout 300	Время в секундах, по истечении которого при отсутствии запросов сервер прерывает связь с клиентом
KeepAlive On	Разрешать (<i>on</i>) или нет(<i>off</i>) множественные запросы через одно TCP соединение
MaxKeepAliveRequests 100	Максимальное количество запросов через одно TCP соединение
KeepAliveTimeout 15	Время ожидания следующего запроса в секундах

Продолжение табл. 1

Наименование директивы	Описание
MinSpareServers 8 MaxSpareServers 20	Минимальное и максимальное значение загруженных процессов <i>httpd</i> . Если набрать в консоли ps ax grep httpd , то можно просмотреть все процессы <i>httpd</i> , загруженные на машине. Если используется Apache на локальной машине, то необходимо уменьшить значение <i>MinSpareServers</i>
StartServers 10	Количество изначально запускаемых процессов
MaxClients 150	Ограничивает количество одновременно подключенных клиентов к серверу
MaxRequestsPerChild 100	Максимальное количество запросов, которые обрабатывает процесс до завершения работы
DocumentRoot /home/httpd/html	Путь к файлам сайта. Здесь хранятся те файлы, которые пользователь получает при наборе <i>URL</i> 'а <i>http://мой.сайт.ru</i> . После установки там находится документация по Apache . Эти файлы следует заменить на файлы своего сайта
UserDir public_html	Имя каталога в домашней директории пользователя, где должны находиться файлы его веб-страницы. Обращение к ним происходит по <i>URL</i> 'у <i>http://мой.сайт.ru/~имя_пользователя</i> (обрати



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

	внимание на ~ ("тильда"))
DirectoryIndex index.html index.shtml index.cgi	Файлы индекса каталога. Это те файлы, которые выводятся при задании в браузере URL'a, являющемся ссылкой на каталог. Например, <i>http://localhost/manual</i> – выведет файл <i>index.html</i> , находящийся в каталоге <i>/home/httpd/html/manual</i> . Если файл индекса отсутствует в каталоге, выводится все содержимое каталога. Можно задать <i>DirectoryIndex default.htm</i>

Продолжение табл. 1

Наименование директивы	Описание
FancyIndexing on	При включении этой опции список файлов каталога будет выводиться в виде таблицы, в которой над каждой колонкой есть ссылка, при нажатии на которую список сортируется по этой колонке (например, по времени модификации или размеру)
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip AddIconByType (TXT,/icons/text.gif) text/* ... AddIcon /icons/binary.gif .bin .exe ... DefaultIcon /icons/unknown.gif	Эти директивы задают имя файла иконки в зависимости от типа, расширения и имени файла, которая будет выводиться перед именем файла в списке содержимого каталога
ReadmeName README HeaderName HEADER	<i>ReadmeName</i> – имя README файла, который сервер ищет по умолчанию. <i>HeaderName</i> – имя файла, который включается в листинг каталога в качестве заголовка.
IndexIgnore .??* ~*# HEADER* README* RCS	Файлы, не включаемые в список каталога
AccessFileName .htaccess	Имя файла с правилами доступа, содержимое которого проверяется для каждого каталога
TypesConfig /etc/mime.types	Файл с описаниями типов файлов и принадлежащих к ним расширений
DefaultType text/plain	Тип файла по умолчанию для тех файлов, тип которых неизвестен
AddEncoding x-compress Z AddEncoding x-gzip gz	Позволяет некоторым браузерам распаковывать “сжатые” и gzip-овые архивы “на лету”, т. е. при копировании файла типа <i>textcounter.tar.gz</i> получается уже распакованный файл <i>textcounter.tar</i>

Продолжение табл. 1

Наименование директивы	Описание
AddLanguage en .en	Позволяет определить язык документа. Если на

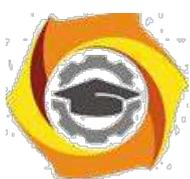


Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

AddLanguage fr .fr AddLanguage de .de AddLanguage da .da AddLanguage el .el AddLanguage it .it	сайте есть английская и русская версии страницы, то можно создать файлы <i>doc.html.ru</i> и <i>doc.html.en</i> и в зависимости от языка, прописанного в браузере клиента, ему будет показываться одна из страниц при обращении к <i>doc.html</i> . Расширение не обязательно должно совпадать с аббревиатурой языка
LanguagePriority en fr de	Определяет приоритет языка документа
Alias /icons/ /home/httpd/icons/	Разрешает создавать <i>алиасы</i> (альтернативные имена) для каталогов, т.е. если html-страница лежит в <i>/home/httpd/html</i> , то при обращении к <i>http://мой.сайт.ru/icons</i> Apache будет искать каталог <i>/home/httpd/html/icons</i> , а после определения такого <i>алиаса</i> Apache будет искать каталог <i>/home/httpd/icons</i>
ScriptAlias /cgi-bin/ /home/httpd/cgi-bin/	То же самое, что и <i>Alias</i> , но для серверных скриптов (CGI)
AddType text/html .shtml AddHandler server-parsed .shtml	Добавляет поддержку SSI для файлов с расширением .shtml
AddHandler imap-file map	Добавляет поддержку <i>imagemap</i> 'ов
BrowserMatch "Mozilla/2" nokeepalive BrowserMatch "MSIE 4.0b2;" nokeepalive downgrade-1.0 force-response-1.0 BrowserMatch "RealPlayer 4.0" force-response-1.0 BrowserMatch "Java/1.0" force-response-1.0 BrowserMatch "JDK/1.0" force-response-1.0	Прописывает различные параметры работы <i>httpd</i> при обращении к нему описанных браузеров

Продолжение табл. 1

Наименование директивы	Описание
<Directory /> Options None AllowOverride None </Directory>	Описывает параметры доступа к корневой директории. Options контролирует то, какие опции доступа к директории доступны. Как видно из примера – никакие (None). Следовательно, доступа никакого. AllowOverride None – запрещает изменять опции доступа посредством содержимого файла <i>.htaccess</i> каталога
<Directory /home/httpd/html> Options Indexes Includes FollowSymLinks AllowOverride None order allow,deny allow from all </Directory>	Устанавливает параметры доступа к каталогу <i>/home/httpd/html</i> . Опции доступа (Options) следующие:
Indexes	если эта опция включена, то при указании URL'а, ссылающегося на каталог (типа <i>http://www.ru/nirvana</i> или <i>http://localhost/manual</i>), при отсутствии в каталоге файла, описанного директивой

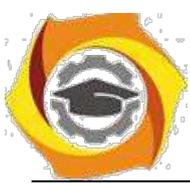


Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

	DirectoryIndex (<i>index.html,index.php3,index.cgi</i> и т. д.), <i>httpd</i> возвращает клиенту листинг этого каталога
Includes	разрешает выполнение SSI директив в файлах, описанных директивой AddHandler server-parsed <i>mun</i> и находящихся в каталоге <i>/home/httpd/html</i>
FollowSymLinks	позволяет использовать символические ссылки на файлы или каталоги, не находящиеся в <i>/home/httpd/html</i>
AllowOverride None	см. выше

Окончание табл. 1

Наименование директивы	Описание
Order allow, deny	определяет последовательность применения правил запрета и правил разрешения доступа к каталогу. Здесь сначала проверяются разрешающие правила, затем запрещающие, т. е. если разрешено, например, по <i>IP</i> , и запрещено, например, по имени <i>хоста</i> , то будет запрещен доступ к этому каталогу, т. к. запрещающие правила сработали последними
Allow from all	разрешен доступ отовсюду.
<Directory /home/httpd/cgi-bin> AllowOverride None Options ExecCGI </Directory>	Прописывает параметры доступа к каталогу <i>/home/httpd/cgi-bin</i> (к тому, где лежат <i>CGI</i> -скрипты). AllowOverride None – см. выше. Options ExecCGI – разрешен запуск <i>CGI</i> и больше ничего



```
Alias /doc /usr/doc
<Directory /usr/doc>
order deny,allow
deny from all
allow from localhost
Options Indexes FollowSymLinks
</Directory>
```

Создает *алиас /doc* для каталога */usr/doc* и описывает параметры доступа к нему. **order deny,allow** – сначала срабатывают запрещающие правила, затем разрешающие. **deny from all** – запрещен доступ всем клиентам, откуда бы то ни было. **allow from localhost** – разрешен доступ с машины, на которой и запущен Apache. **Options Indexes FollowSymLinks** – см. выше. (Если запустить Netscape на той же машине, на которой загружен Apache, и набрать URL – *http://localhost/doc* или *http://127.0.0.1/doc*, то можно увидеть листинг каталога */usr/doc*.)

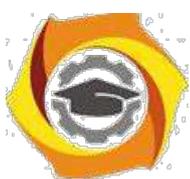
Настройка виртуальных серверов в файле `httpd.conf`

В большинстве случаев один *http*-сервер способен обрабатывать запросы, поступающие на различные, так называемые виртуальные, Web-серверы. Виртуальные серверы могут иметь как один и тот же IP-адрес, но разные доменные имена, так и разные IP-адреса. С точки зрения пользователя второй вариант чуть более предпочтителен, поскольку запрос к серверу, отличающемуся от основного только доменным именем, должен содержать его имя, а некоторые старые браузеры, не поддерживающие протокол HTTP/1.1 (например, Microsoft Internet Explorer 2.0), не включают в запрос эту информацию. Однако такие браузеры выходят из употребления (сейчас их уже менее 0,5 % от общего числа); с другой стороны, выделение собственного

IP-адреса каждому виртуальному серверу может быть неоправданной растратой адресного пространства компании.

Для описания адресов и доменных имен виртуальных серверов служат директивы **ServerName**, **ServerAlias**, **NameVirtualHost** и **VirtualHost**. Они необходимы, только если вам нужно установить более одного виртуального сервера.

Директива **ServerName**, находящаяся вне секций **VirtualHost**, определяет имя основного сервера, т. е. сервера, корневой каталог которого задан директивой **DocumentRoot** в файле *srm.conf*. Виртуальные серверы наследуют настройки основного; при необходимости специальной настройки соответствующие директивы помещаются в секции **VirtualHost**, относящейся к данному серверу. Допустимы любые директивы, которые могут встретиться в файлах *httpd.conf* и *srm.conf*, например **DocumentRoot**, **ErrorLog**, **CustomLog**, **Location**, **ServerAdmin**.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Задание на лабораторную работу

Лабораторная работа выполняется по вариантам. Задания приведены в табл. 2.

Таблица 2

Задания на лабораторную работу

Вариант	Задание
Подгруппа 1, 5	Сконфигурировать сервер на порт 90; для тестового сайта использовать содержимое каталога testserver1; добиться, чтобы выполнялись тесты 1, 3, 4, 5; указать, чтобы в файлах log использовались буквенные имена, а не цифровые; создать дополнительный виртуальный сервер на порт 100, для тестового сайта использовать содержимое каталога testserver2; добиться, чтобы выполнялись тесты 2, 3
Подгруппа 2, 6	Сконфигурировать сервер по умолчанию; для тестового сайта использовать содержимое каталога testserver1; добиться, чтобы выполнялись тесты 4, 5; указать, чтобы в файлах log использовались цифровые имена, а не буквенные; создать дополнительный виртуальный сервер на домен (буквенное имя) <i>lvirn</i> , где <i>n</i> – номер компьютера; для тестового сайта использовать содержимое каталога testserver2; добиться, чтобы выполнялись тесты 1, 3
Подгруппа 3, 7	Сконфигурировать сервер на порт 80; для тестового сайта использовать содержимое каталога testserver1; переименовать файл index.html в этом каталоге в sind.htm; указать в конфигурации, чтобы данный файл загружался по умолчанию; добиться, чтобы выполнялись тесты 4, 5; указать, чтобы в файлах log использовались цифровые имена, а не буквенные; создать дополнительный виртуальный сервер на порт 100; для тестового сайта использовать содержимое каталога testserver2; добиться, чтобы выполнялись тесты 1, 2

Окончание табл. 2

Вариант	Задание
Подгруппа 4, 8	Сконфигурировать сервер по умолчанию; для тестового сайта использовать содержимое каталога testserver1; добиться, чтобы выполнялись тесты 3, 5; указать, чтобы в файлах log использовались буквенные имена, а не цифровые; создать дополнительный виртуальный сервер на домен (буквенное имя) <i>lvirn</i> , где <i>n</i> – номер компьютера, для тестового сайта использовать содержимое каталога testserver2; переименовать файл index.html в этом каталоге в sind.htm; указать в конфигурации, чтобы данный файл загружался по умолчанию; добиться, чтобы выполнялись тесты 3, 4

Для редактирования файлов конфигурации и навигации по файловой системе удобно использовать программу **mc**, для ее загрузки необходимо набрать в командной строке:



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

[root@lis] \$ mc

Интерфейс программы интуитивно понятен и не представляет трудностей при использовании.

Файлы конфигурации web-сервера Apache в данной системе находятся в каталоге: **/etc/httpd/conf/**

Для запуска, остановки, перезапуска web-сервера используются следующие скрипты (программы):

Остановка: **/etc/rc.d/init.d/httpd stop.**

Запуск: **/etc/rc.d/init.d/httpd start.**

Перезапуск: **/etc/rc.d/init.d/httpd restart.**

В ходе лабораторной работы необходимо произвести настройку web-сервера в соответствии с заданием, продемонстрировать его работоспособность преподавателю.

Для демонстрации работоспособности необходимо переписать с

<ftp://ais.khstu.ru/incoming> средствами Linux на локальный компьютер каталоги testserver1 и testserver2, для чего использовать в программе mc пункт меню **left (right) -> ftp link**. В данных каталогах содержатся тестовые сайты:

- проверка исполнения файла */cgi-bin/test1.cgi*
- проверка исполнения файла */cgi-bin/test2.pl*
- проверка исполнения файла */cgi-bin/test3*
- проверка SSI-включения файла */cgi-bin/test1.cgi* в файл *test4.html*
- проверка SSI-включения файла */cgi-bin/test1.cgi* в файл *test4.shtml*

Для создания виртуальных серверов по доменным именам используемые доменные имена необходимо прописать в файле *c:\winnt\system32\driver\etc\hosts* на том компьютере, на котором будет проводиться проверка.

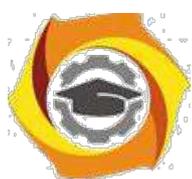
При защите необходимо знать назначение используемых директив, уметь объяснить информацию, содержащуюся в log файлах.

Контрольные вопросы

1. Какие файлы содержат конфигурационную информацию web-сервера?
2. Какова последовательность установки web-сервера?
3. Как проверить работоспособность web-сервера?
4. Где хранятся log файлы?
5. Что такое виртуальный web-сервер?

Библиографический список

1. Бэндел Дэвид. Защита и безопасность в сетях Linux = Linux security toolkit: Пер. с англ. / Бэндел Дэвид. - СПб.: Питер, 2002. – 480 с.
2. Блэк У. Интернет: Протоколы безопасности = Internet Security Protocols.Protecting IP Traffic: Пер. с англ. / У. Блэк. – СПб.: Питер, 2001. – 288 с.
3. Попов В. Б. Основы компьютерных технологий: Учеб. пособие для вузов / В. Б. Попов. – М.: Финансы и статистика, 2002. - 704с.
4. Уэйнгроу К. UNIX. Администрирование. – М.: ДМК Пресс, 2002. – 416 с.
5. Холден Г. Apache Server в комментариях. – М.: DiaSoft, 2000. – 480 с.
6. Рич Б. Apache. Настольная книга администратора. – М.: DiaSoft, 2002. –378 с.
7. Хокинс С. Администрирование Web-сервера Apache и руководство по электронной коммерции. – М.: Вильямс, 2001. – 330 с.
8. Ньюкомер Э. Веб-сервисы. XML, WSDL, SOAP и UDDI. – СПб.: Питер, 2003. – 256 с.



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Лабораторная работа 5 Установка, настройка и администрирование проху-сервера

Цель работы: настройка и администрирование Proху-сервера **squid** под операционной системой Linux.

Лабораторная работа выполняется в локальной сети на рабочей станции с операционной системой Linux 7 или более поздней. В лабораториях кафедры операционная система Linux работает на компьютерах под управлением программного пакета VMware. Этот пакет позволяет создавать так называемые «виртуальные машины» – мнимые компьютеры, не зависящие от выполняющейся в текущее время на данном компьютере операционной системы (ОС). Для запуска ОС Linux необходимо запустить VMware на рабочей станции, выбрать из списка требуемую операционную систему и нажать кнопку «Power ON».

*После окончания загрузки для входа в систему необходимо использовать имя пользователя **root**.*

Порядок выполнения лабораторной работы

Подготовка и допуск к работе. К выполнению лабораторной работы допускаются студенты, которые подготовились к работе и имеют не более двух невыполненных предыдущих работ.

Перед работой студент должен:

- предъявить преподавателю полностью оформленный отчет о предыдущей работе;
- ответить на вопросы преподавателя.

Студенты, которые не выполнили одно из вышеперечисленных требований, к работе не допускаются.

Отчёт по работе должен содержать:

- текст задания;
- перечень всех использованных в лабораторной работе команд и инструкций;
- вывод по работе.

Установка и администрирование Proху-сервера

Proху-сервер, осуществляющий доступ в Internet, предоставляет следующие возможности:

- централизованный выход в Internet через один сервер в сети;
- локальное хранение часто просматриваемых документов для увеличения скорости загрузки страниц (один пользователь загрузил документ с удаленного сервера в Internet, а все остальные после этого берут этот документ с Proху-сервера);
- регулирование пропускной способности канала в зависимости от его нагрузки;
- авторизованный доступ в Internet (пользователь может загружать документы из Internet только при наличии логина и пароля).

Установка Proху-сервера. Прежде чем устанавливать Proху-сервер *squid*, надо убедиться в том, что:



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- машина, на которой будет работать *Proxy*, может соединиться (по WWW, FTP, Telnet – неважно) с другими машинами в сети;
- машины из внутренней сети могут соединиться с машиной, на которую устанавливается *Proxy*, опять же неважно каким клиентом;

Если же связь изнутри к *Proxy*-машине есть и эта *Proxy*-машина может общаться с внешним миром, можно переходить к настройке *squid*. *Squid* надо устанавливать из *packages* или *ports*, тогда есть уверенность, что все его компоненты будут размещены по нужным директориям. После установки должно получиться примерно следующее:

- двоичные файлы должны находиться в каталоге */usr/local/sbin*;
- в каталоге */usr/local/etc* должна появиться директория *squid*, в которой лежит *squid.conf* – это его конфигурационный файл, его надо будет редактировать;
- в каталоге */usr/local/etc/rc.d* появился файл *squid.sh* – это основной стартовый файл (дело в том, что система при старте просматривает этот каталог и все, что найдет там типа **.sh*, запустит автоматически). Но можно его запустить и вручную, просто написав *./squid.sh*. Если такого файла нет, то при перезагрузке системы *squid* не будет запускаться;
- в каталоге */usr/local* образовалась директория *squid*, в которой две поддиректории: *cache* – там будет его кэш и *logs* – логи. Там же должен быть файл (возможно, он появится после первого запуска *Proxy*-сервера) *squid.out* – это основной лог-файл, в котором и будут сообщения об ошибках, если *squid* почему-либо не сможет стартовать нормально.

Для начала *squid.conf* можно редактировать незначительно. Там стоят значения «по умолчанию» и они вполне приемлемы. Единственное, что необходимо сделать – определиться, сколько мегабайт диска следует выделить под кэш. По умолчанию – 100 Мб. Для изменения этого значения найдите в *squid.conf* строчку

```
cache_swap 100
```

и поставьте подходящее значение. Максимальный объем можно оценить исходя из пропускной способности канала Internet за одни сутки.

Скорее всего, понадобится еще одно исправление. Дело в том, что нельзя запускать *Proxy*-сервер от имени *root*. Поскольку при старте машины *squid.sh* будет исполняться от имени *root*, то надо сделать следующее. Надо найти в конфигурационном файле строчку

```
cache_effective_user nobody nogroup
```

и «раскомментировать» ее. Это будет означать, что в процессе работы *squid* будет иметь права «псевдоюзера» *nobody*.

На самом деле это не совсем правильно. Правильнее завести нового «псевдоюзера» *squid* (или еще как-нибудь - *www*, *cache* ...) и в конфиг-файле вписать именно его, а не *nobody*. После этого надо будет поправить владельца для директории */usr/local/squid*.

Для этого выполните команду

```
chown -R nobody /usr/local/squid
```

Здесь **-R** означает, что меняется владелец не только директории, но и рекурсивно меняется владелец всего содержимого поддиректорий. Если Вы завели специального «псевдоюзера», то, естественно, в команде вместо *nobody* укажите его имя. Теперь осталось сформировать «внутреннюю структуру кэша». Кстати, если этого не сделать, то *squid* при запуске сам подскажет вам: «запустите программу *squid -z*». Естественно, это надо сделать. Только учтите, что */usr/local/sbin* обычно не прописан в PATH даже у *root*, поэтому лучше набрать полный путь

```
/usr/local/sbin/squid -z
```

Теперь можно попытаться запустить *squid*. Зайдите в */usr/local/etc/rc.d* и запустите *./squid.sh*. На консоли должно появиться сообщение от *squid*: "Ready to serve requests" («готов обслуживать запросы»).

Настройка Proxy-сервера. Общая настройка *Proxy*-сервера чаще всего не вызывает сложностей. Сложности обычно вызывают три обстоятельства: настройка *ACL* (*access control*



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

list – список прав доступа) и правил для них, настройка дополнительных программ вроде «баннерорезалок» и настройка ограничений использования канала.

Настройка ACL. Обратимся к соответствующему месту файла *squid.conf* и прокомментируем его.

ACL прописываются в виде строки ***acl имя_acl тип_acl параметры ACL или acl имя_acl тип_acl «файл»*** – при этом в файле сохраняется по одному значению на строку. Итак, сначала типы списков.

acl aclname src ip-address/netmask – в этом *acl* описывается *ip*-адрес или сеть, принадлежащая клиентам *squid*. Например:

acl vasya src 192.168.1.1/255.255.255.255 – описывает единственную машину с адресом 192.168.1.1 и назначает ей ACL с именем *vasya*.

acl office src 192.168.1.1/255.255.255.0 – описывает диапазон машин с адресами 192.168.1.1-.254 и назначает этому ACL имя *office*. Если диапазон необходимо сузить, то необходимо либо изменить маску подсети, либо воспользоваться явным указанием: ***acl vip_user src 192.168.1.1-192.168.1.5/255.255.255.0***. Здесь *squid* выбирает тот диапазон адресов, который окажется меньше либо по маске, либо по явному указанию.

acl aclname dst ip-address/netmask – этот тип ACL описывает уже сервер, страницы с которого будут запрашивать клиенты. Следует отметить, что в этом типе ACL задается не символьный адрес сервера, а *ip*.

acl aclname srcdomain.domain.ru – описывает клиентов, но уже не по *ip*-адресам, а по реверсным DNS. Это значит, что нет разницы, какие *ip*-адреса принадлежат клиентам, главное, чтобы они определялись *dns*. Соответственно под это правило попадут все клиенты, стоящие в домене *domain.ru*.

acl aclname dstdomain .domain.ru – описывает сервер. Сравнивается с запросом из URL. Под этот ACL попадут все серверы третьего уровня домена *domain.ru*.

acl aclname srcdom_regex [-i] xxx acl aclname dstdom_regex [-i] xxx – описания аналогичны предыдущим, но теперь для выяснения, подходит ли правило под запрос, используются *regex*-правила. Если символьный адрес не смог определиться из *ip*-адреса, к запросу будет применена строка *none*.

acl aclname time [day-abbrevs] [h1:m1-h2:m2] – ACL, описывающий время. Коды дней недели определяются так: *S* – Sunday – Воскресенье, *M* – Monday – Понедельник, *T* – Tuesday – Вторник, *W* – Wednesday – Среда, *H* – Thursday – Четверг, *F* – Friday – Пятница, *A* – Saturday – Суббота.

Ну а вместо *h1:m1* и *h2:m2* вставляется время. Запомните – *h1:m1* всегда должно быть меньше *h2:m2*.

Итак, ***acl worktime time MTWHF 08:00-17:00*** описывает рабочее время с понедельника по пятницу, с 8 утра до 5 вечера, ***acl weekday time SA*** описывает целиком субботу с воскресеньем, ***acl evening time 17:00-23:59*** описывает время до полуночи. Если необходимо описать всю ночь, то приходится заводить два ACL: первый – с вечера до полуночи, а второй – с полуночи до утра.

acl aclname url_regex [-i] ^http:// – *regex*-правила, применяемые ко всему URL.

acl aclname urlpath_regex [-i] \.gif\$ – аналогичные правила, применяемые к URL.

acl aclname port 80 70 21 – ACL, описывающий порты. Вместо простого перечисления можно указать диапазон, например, 1-1024.

acl aclname proto HTTP FTP – ACL, описывающий протокол, по которому клиент желает сделать запрос на сервер.

acl aclname method GET POST – метод, которым передаются данные клиента серверу. ***acl aclname browser [-i] regexp*** – *regex*-запрос на клиентский браузер.

Вычисления основаны на заголовке *User-Agent*, который пересылает браузер.

acl aclname ident username – ACL описывает имя пользователя, от которого запущена программа на клиентской машине. Имя узнается с помощью *ident*-сервера.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

acl aclname ident_regex [-i] pattern – то же самое, но основанное на *regex*-правилах. **acl aclname proxy_auth username acl aclname proxy_auth_regex [-i] pattern** – ACL, описывающий имя пользователя. Это имя возвращает внешняя авторизирующая программа. **acl aclname maxconn number** – это правило сработает, если клиент сделает больше *number* запросов к кешу.

acl req_mime_type mime-type1 – правило, срабатывающее при *upload* файлов клиентом. Заметьте – *uplode*, а не скачивание.

Здесь представлены не все описания ACL, но большинство, необходимых в повседневной практике. Для более полного знакомства с описанием следует обратиться к исходному тексту файла *squid.conf* – там все описано, правда, по-английски. Итак, создадим правила обычной сети:

acl all src 0.0.0.0/0.0.0.0 acl office src 192.168.1.0/255.255.255.0

all – правило, описывающее все машины, и *office* – описывающее все машины в подсети 192.168.1.0.

http_access allow office http_access deny all

Эти два правила описывают полный доступ машинам, описываемым *acl office*, и запрещает доступ машинам, описываемым *all*. В приведенном примере есть конфликт в описания прав доступа: машины, попадающие под правило *all* (а по этому правилу все запрещено) не могут использовать Проху-сервер. Тут в дело вступает порядок просмотра ACL – они просматриваются в порядке объявления, и если сработало одно правило, то другие уже не просматриваются.

К примеру, если мы введем в дополнение ACL **acl vasya src 192.168.1.100/255.255.255.255** и расположим правила так:

http_access allow office http_access deny vasya http_access deny all ,

то машина с *ip*-адресом 192.168.1.100 по-прежнему будет иметь возможность соединиться через Проху сервер;

а если так:

http_access deny vasya http_access allow office http_access deny all ,

то все будет в порядке. Остальные офисные машины не попадают под действие первого правила.

Если в списке нет ни одного правила, то запрос будет отвергнут. Если ни одно правило не сработало, то за основу берется последнее. Если, к примеру, мы заменим предпоследнее правило на *http_access allow all*, то нашим Проху-сервером смогут пользоваться абсолютно все (кроме *vasya*), кто сможет соединиться с портом *squid*. Так что будьте внимательнее.

Но авторы *squid* постарались: даже если последнее правило будет разрешающим для всех, то запрос будет отвергнут. Это поможет избежать дыр в Проху-сервере.

На основе этих же списков-правил так же управляется и доступ к другим возможностям Проху-сервера (см. файл *squid.conf*, где все расписано).

Правила – правилами, но, предположим, что в сети появились пользователи, которые честно подключаются к серверу и начинают выкачивать гигабайтами запрещенную информацию. При этом занесение этих сайтов в *deny*-список вызывает их возмущение.

На этот счет придумали много вещей, но самым эффективным остается сокращение канала для таких пользователей: доступ есть, но качается плохо, возразить им нечего – такая ситуация в Internet не редкость.

Итак, давайте разберемся с «трафик-шейпингом» – именно так это называется. В *squid* же это называется *delay-pool*. Заметим, что *squid* при сборке должен быть собран с опцией *-- enable-delay-pools*.

Итак, сначала разберемся, какие есть пулы. Пулы делятся на три класса. Первый, и самый простой, это когда всему *acl* ограничивается трафик до определенной величины. Второй – когда отдельно ограничивается трафик для одной машины из подсети и для всей подсети. И



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

третий класс – когда ограничивается трафик для отдельных машин, для подсети класса *C* или меньше и для подсети класса *B*.

Итак, давайте обыграем ситуацию, когда в сети завелся «дискокачалщик».

delay_pools 1 – у нас всего один пул.

delay_class 1 1 – первый пул первого класса.

delay_access 1 allow vasya

delay_access 1 deny all

В первый пул попадают только машины, описываемые ACL *vasya*. Остальные работают, как им положено, ведь им доступ к первому пулу запрещен.

delay_parameters 1 800/64000

Вот и все. Теперь файлы и страницы объемом до 64 Кб будут скачиваться на максимальной скорости, а то, что больше этого – на скорости 800 б/с.

Или совсем уж радикальная мера:

delay_parameters 1 800/800 – и «злой качальщик» все будет качать на скорости 800 б/с.

Но даже в не очень большой сети будут возникать ситуации, когда все хотят что-то качать, в итоге никому ничего не хватает.

Исправляем строчку с *delay_pools* на *delay_pools 2*. Теперь у нас будет два пула.

delay_class 2 2 – второй пул будет второго класса (совпадение номеров чисто случайно) – первый – это *vasya*.

delay_access 2 allow office

delay_access 2 deny all

Во второй пул попадают только машины с ACL

office. *delay_parameters 2 64000/64000 4000/4000*

В итоге вся подсеть, описываемая *office*, будет использовать канал не больше 512 Кбит/с (64 Кб/с), но каждый отдельный хост будет качать не более 4 Кб/с. Этим правилом очень легко разграничить по скорости разные подсети, использующие один канал.

К примеру, у нас есть две подсети, описываемые *office* и *office1*. При этом *office* не должна иметь никаких ограничений на канал (примем канал за 256 Кбит) в целом, но каждый из *office* не должен качать быстрее 6 Кб/с. А *office1* – это пользователи, которым всем и 5 Кб/с хватит.

Создаем два пула второго класса и прописываем для них ACL. Затем определяем этим пулам параметры.

delay_parameters 3 -1/-1 6000/6000 – это определение для *office* (ему отдан номер пула 3).

delay_parameters 4 5000/5000 -1/1 – а это для *office1*.

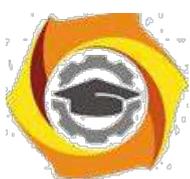
В итоге после применения этих правил получаем все, что заказано – первый офис грузит канал как хочет (-1/-1), но никто из сотрудников больше 6 Кб/с не получает. А второй офис грузит канал не больше 5 Кб/с, но в распределении этих 5 Кб/с между сотрудниками нет никаких правил.

Понятно, что в описание пулов можно заложить и другие параметры, например, время, место доступа и т. д. Остается еще одна маленькая вещь, которую нельзя оставить без внимания. И эта вещь – навязанная реклама через баннеры и другие объекты. Для того, чтобы такую рекламу не пропустить на браузер, каждый URL, который передается *squid*, первоначально передается *редиректору*. И тот либо возвращает прежний URL в случае, если все в порядке, либо возвращает тот, который, по его мнению, более правильный. А кто мешает нам перехватывать обращения к баннерам и счетчикам и вместо них подсовывать свою картинку? В итоге страницы можно заполнить прозрачными окошками.

Итак, в *squid.conf* прописываем строку

redirect_program /squid/bin/redirector

где */squid/bin/redirector* – путь до выполняемой программы, которая как раз и обеспечивает разбор URL. Ее можно написать на чем угодно, но наиболее предпочтительным является Perl



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

– этот язык как раз предназначен для подобного рода работ. Полная версия *редиректора* лежит на <http://linuxnews.ru/redirector>.

Описание директив squid. Описание директив содержится непосредственно в файле конфигурации *squid.conf* и в документации, прилагаемой к данной лабораторной работе.

Задание на лабораторную работу

Лабораторная работа выполняется по вариантам.

Вариант	Задание
Подгруппа 1, 5	Сконфигурировать Проху-сервер на порт 3128; разрешить доступ с <i>ip</i> 10.10.146.150 с 10-00 до 15-00 ч; запретить доступ с <i>ip</i> 10.10.146.176 с 10-00 до 15-00 ч; запретить доступ к доменам <i>*.khhb.ru</i>
Подгруппа 2, 6	Сконфигурировать Проху-сервер на порт 8080; разрешить доступ с <i>ip</i> 10.10.146.176 с 10-00 до 15-00 ч; запретить доступ к файлам <i>*.gif</i> ; запретить доступ к домену <i>*.khhb.ru</i> с <i>ip</i> 10.10.146.150
Подгруппа 3, 7	Сконфигурировать Проху-сервер на порт 12345; разрешить доступ с <i>ip</i> 10.10.146.176; запретить доступ к файлам <i>*.cgi</i> ; запретить доступ к домену <i>*.khhb.ru</i> с <i>ip</i> 10.10.146.150 с 10-00 до 15-00 ч
Подгруппа 4, 8	Сконфигурировать Проху-сервер на порт 1345; запретить доступ с <i>ip</i> 10.10.146.176; запретить метод <i>POST</i> ; запретить доступ к домену <i>*.khstu.ru</i> и к файлам <i>*.gif</i> с <i>ip</i> 10.10.146.150 с 10-00 до 15-00 ч

При защите необходимо знать назначение используемых директив, уметь объяснить информацию, содержащуюся в *log*-файлах

Для редактирования файлов конфигурации и навигации по файловой системе удобно использовать программу *mc*. Для ее загрузки необходимо набрать в командной строке **[root@lis] \$ mc**

Интерфейс программы *mc* интуитивно понятен и не представляет трудностей при использовании.

Содержание отчета

1. Содержимое конфигурационных файлов (без комментариев в тексте).
2. Отрывок из *log*-файлов, демонстрирующий обращения через прокси-сервер.

Контрольные вопросы

1. Назначение сервера прокси.
2. В каком файле содержится информация о конфигурации Проху-сервера?
3. Какие протоколы кэшируются прокси?
4. Какие условия должны быть выполнены перед установкой Проху-сервера?
5. Что такое список прав доступа?
6. Общий синтаксис ACL.

Библиографический список



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

1. *Мартин Майкл Дж.* Введение в сетевые технологии = Understanding the Network: Практическое руководство по организации сетей / Мартин Майкл Дж. – М.: ЛОРИ, 2002. – 660 с.
2. *Бэндел Дэвид.* Защита и безопасность в сетях Linux = Linux security toolkit: Пер. с англ. / Бэндел Дэвид. – СПб.: Питер, 2002. – 480 с. (Для профессионалов. Б-ка Linux).
3. *Мельников Д. А.* Информационные процессы в компьютерных сетях: Протоколы, стандарты, интерфейсы, модели... / Мельников Д. А. – М.: КУДИЦ-ОБРАЗ, 1999. – 256 с. (Б-ка профессионала).



Практическое занятие 6 Создание защиты компьютерной сети с использованием брандмауэра

Упражнение 1. Сетевое сканирование nmap

nmap — свободная утилита, предназначенная для разнообразного настраиваемого сканирования IP-сетей с любым количеством объектов, определения состояния объектов сканируемой сети (портов и соответствующих им служб). Изначально программа была реализована для систем UNIX, но сейчас доступны версии для множества операционных систем [5].

nmap использует различные методы сканирования, таких как UDP, TCP (connect), TCP SYN (полукоткрытое), FTP-proxy (прорыв через ftp), Reverse-ident, ICMP (ping), FIN, ACK, Xmas tree, SYN- и NULL-сканирование. Nmap также поддерживает большой набор дополнительных возможностей, а именно: определение операционной системы удалённого хоста с использованием отпечатков стека TCP/IP, «невидимое» сканирование, динамическое вычисление времени задержки и повтор передачи пакетов, параллельное сканирование, определение неактивных хостов методом параллельного ping-опроса, сканирование с использованием ложных хостов, определение наличия пакетных фильтров, прямое (без использования portmapper) RPC-сканирование, сканирование с использованием IP-фрагментации, а также произвольное указание IP-адресов и номеров портов сканируемых сетей.

1. Создайте виртуальную машину, используя инструкции из упражнения 1 практического занятия №1. Используйте установочный образ **RedHat Enterprise Linux 6**.
2. Ping-сканирование - наиболее распространенным и простым способом сканирования является простое ping-сканирование, которое заключается в отправке ICMP пакетов на исследуемый узел сети. Просканируйте заданный преподавателем узел командой

```
ping 192.168.58.103
```

3. TCP Connect - метод сканирования, при котором сканирующая машина пытается установить соединение со сканируемой. Успешный результат говорит о том, что порт открыт, неудачный — о том, что он закрыт или фильтруется. Просканируйте заданный преподавателем узел командой

```
nmap -sT 192.168.58.103
```

4. Увеличьте информативность вывода результатов сканирования, используя ключ `-v`.
5. TCP-SYN - режим полукоткрытого сканирования. При вызове nmap посылает SYN-пакет. Если в ответе присутствуют флаги SYN или ACK, считается, что порт открыт. Флаг RST говорит об обратном. Сканирование осуществляется только при наличии прав суперпользователя (root). Просканируйте заданный преподавателем узел командой

```
nmap -sS 192.168.58.103
```

6. Межсетевой экран или другие защитные средства могут ожидать входящие SYN-пакеты. Существует еще целая группа возможных способов сканирования, альтернативных TCP SYN. Это FIN, Xmas Tree и NULL-сканирования.

Просканируйте заданный преподавателем узел

```
командами nmap -sF 192.168.58.103
```

```
nmap -sX 192.168.58.103
```

```
nmap -sN 192.168.58.103
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

7. Сканирование протоколов IP - сканируемому узлу передаются IP пакеты без заголовков для каждого протокола сканируемого хоста. Если получено сообщение, говорящее о недоступности протокола, то этот протокол не поддерживается хостом. В противном случае — поддерживается. Просканируйте заданный преподавателем узел командой

```
nmap -sO 192.168.58.103
```

8. АСК сканирование заключается в передаче АСК пакетов на сканируемый порт. Если в ответ приходит RST-пакет, порт классифицируется как нефильтруемый. Просканируйте заданный преподавателем узел командой

```
nmap -sA 192.168.58.103
```

Упражнение 2. Изучение возможностей файервола Windows 7

Первая версия Windows Firewall позволяла очень много сделать в плане защиты от взлома, например настраиваться как с помощью Group Policy Object (GPO) так и из командной строки. Однако возможностей защиты явно не хватало в таких областях, как настройка правил доступа и фильтрация исходящего трафика. Версия Windows Firewall, которая поставляется в составе Windows Vista, имеет 2 существенных нововведения:

Microsoft включила расширенный интерфейс управления для того, чтобы дать администраторам очень подробные правила настройки рабочих станций. Более того, Брандмауэр может быть настроен из оснастки Групповая политика, что означает для корпоративных IT-специалистов более легкий путь навязывания политик организации, например, запрещение специфической активности – обмен мгновенными сообщениями или сети peer-to-peer.

Управление МЭ сосредоточено в MMC-консоли, получившей название Windows Firewall with Advanced Security. Некоторые параметры брандмауэра можно по-прежнему настроить централизованно через Group Policy или же настроить брандмауэр локально при помощи командной утилиты Netsh. Как и другие оснастки, Windows Firewall with Advanced Security поддерживает удаленный доступ, который позволяет управлять настройкой брандмауэра на удаленной станции.

Брандмауэр Windows блокирует входящий трафик по умолчанию, так что необходимо сразу же настроить список Exceptions, если планируется работать с сетевым приложением. Exceptions — это то, что Microsoft называет правилами (rules), а более точно, списки ACL.

Во многих случаях хост-брандмауэры, написанные независимыми разработчиками, выдают предупреждения при попытке установить исходящее соединение и запрашивают разрешение у пользователя. На основе полученного ответа брандмауэр может даже создать правило реагирования на такие события в будущем. Брандмауэр Windows поступает иначе — весь исходящий трафик по умолчанию разрешен. Создание исключений для исходящего трафика — процедура несложная, но она требует использования новой оснастки. Большинство пользователей об этом даже и не вспомнит, но администраторы систем должны как следует разобраться во всех обязательных настройках Windows Firewall with Advanced Security.

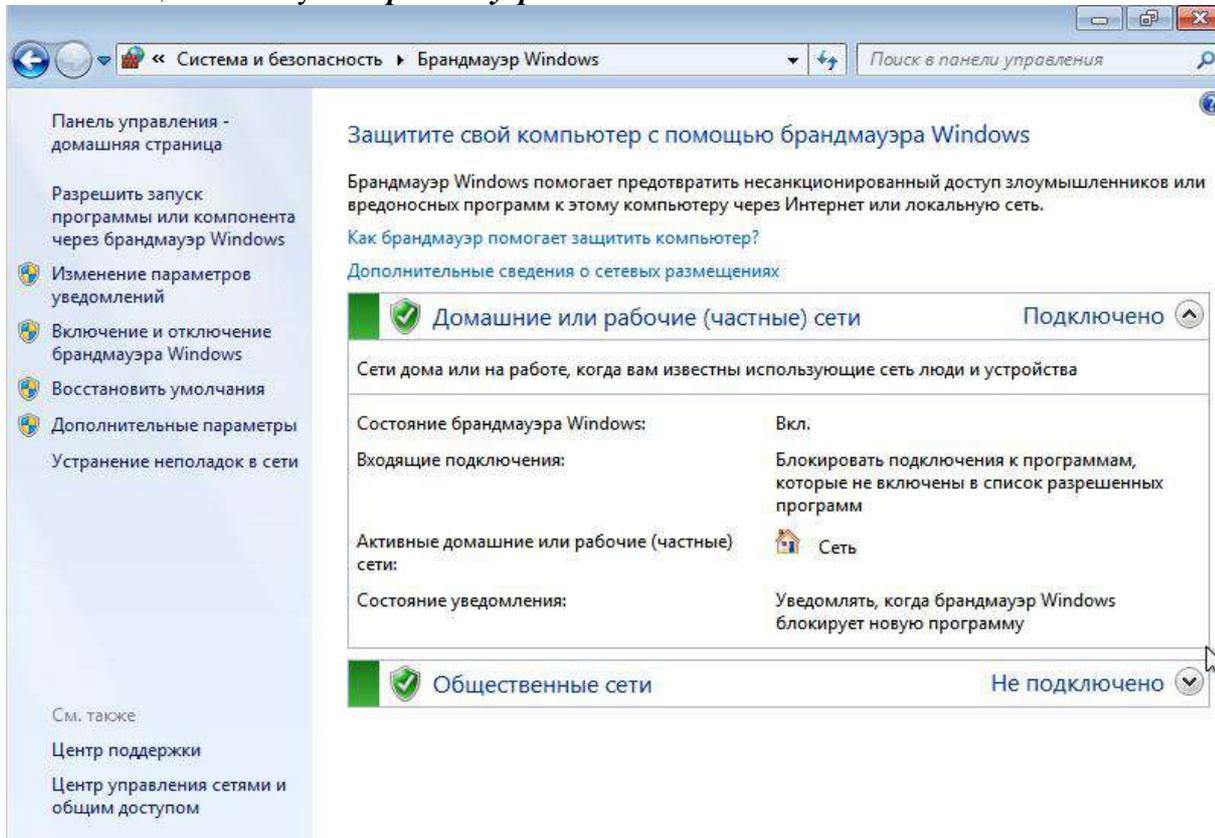
Оснастка позволяет настроить все функции брандмауэра. В левой панели можно выбрать Inbound Exceptions, Outbound Exceptions, Computer Connection Security или Firewall Monitoring, дважды щелкнуть по выбранному элементу — узлу структуры — и увидеть дополнительные настройки в центральной панели. На правой панели располагается список всех доступных действий для выбранного узла. Такое построение консоли делает процесс настройки брандмауэра интуитивно понятной процедурой; например, если нужно включить или отключить правило, достаточно щелкнуть правой кнопкой мыши, вызвав контекстное меню, или же на правой панели выбрать требуемое действие. Большинство действий вступает в силу немедленно, что упрощает отладку при настройке брандмауэра. Чтобы просмотреть и задать свойства брандмауэра, необходимо открыть контекстное меню

Windows Firewall with Advanced Security в левой панели и выбрать Properties.

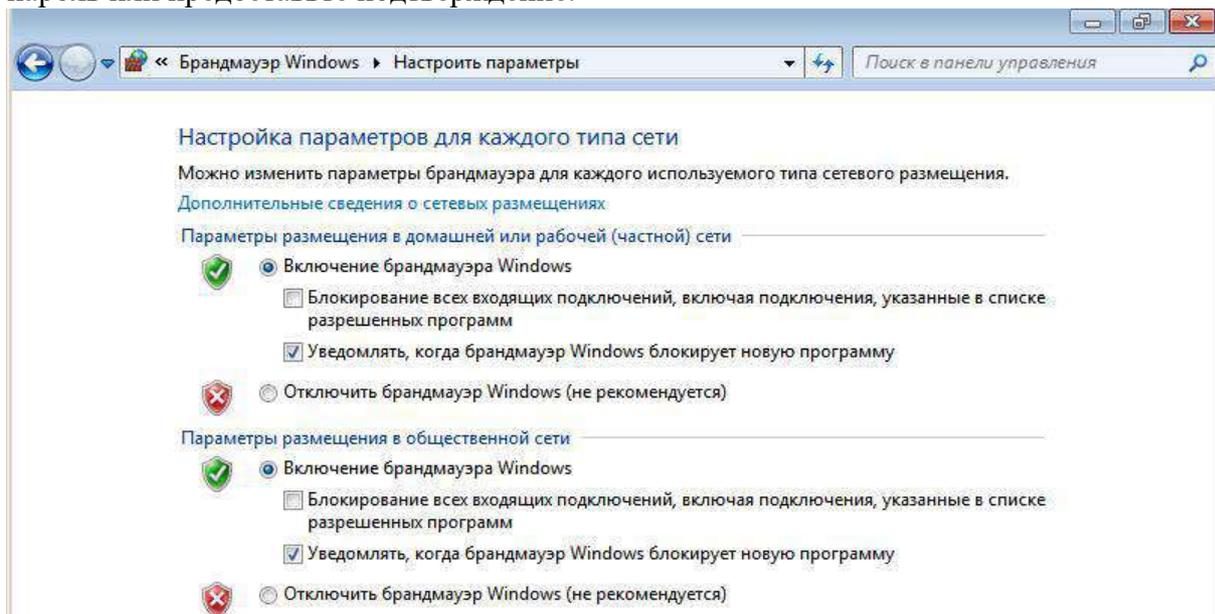


Негосударственное частное образовательное учреждение высшего образования «Технический университет»

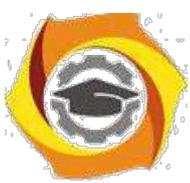
1. Упражнение выполняется на ранее установленной виртуальной машине с операционной системой Windows 7.
2. Откройте компонент **Брандмауэр Windows**. Для этого нажмите кнопку **Пуск** и выберите пункт **Панель управления**. В поле поиска введите брандмауэр и затем щелкните пункт **Брандмауэр Windows**.



В левой области выберите **Включение и отключение брандмауэра Windows**. Если отображается запрос на ввод пароля администратора или его подтверждения, укажите пароль или предоставьте подтверждение.



Блокирование всех входящих подключений, включая подключения, указанные в списке разрешенных программ. Этот параметр блокирует все неожиданные попытки подключения к компьютеру. Этот параметр служит для максимальной защиты компьютера, например, при подключении к общедоступной сети в отеле или в аэропорте или в периоды распространения

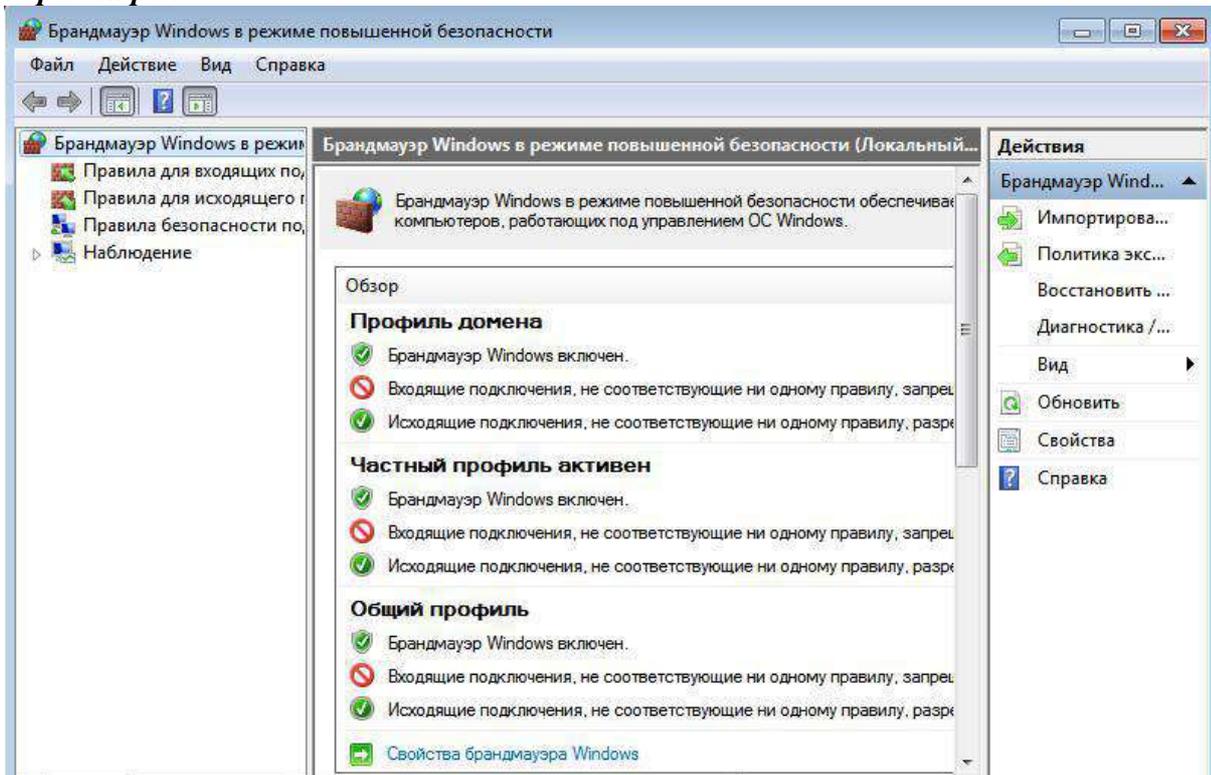


Негосударственное частное образовательное учреждение высшего образования «Технический университет»

через Интернет особо опасных вирусов-червей. При использовании этого параметра вы не будете уведомлены о блокировке программ брандмауэром Windows, и все программы из списка разрешенных программ будут проигнорированы.

Включение брандмауэра Windows. Этот вариант выбран по умолчанию. Если брандмауэр Windows включен, то установка связи большинства программ брандмауэром блокируется. Если следует разрешить программе устанавливать связь через брандмауэр, можно добавить ее в список разрешенных программ.

Убедитесь, что брандмауэр Windows включен, и перейдите во вкладку *Дополнительные параметры*.



Создайте дополнительное правило для *Брандмауэра Windows*. Выберите перечень правил *Правила для входящих\исходящих подключений* и нажмите кнопку *Создать правило*.

Создайте дополнительное правило для входящих подключений для порта 80/TCP.

Объясните назначение порта.

Проведите сканирование АРМ, защищённого настроенным вами межсетевым экраном.

Сделайте выводы о защищённости.

Упражнение 3. Изучение возможностей межсетевого экрана iptables.

iptables — утилита командной строки, является стандартным интерфейсом управления работой межсетевого экрана (брандмауэра) netfilter для ядер Linux версий 2.4, 2.6, 3.x, 4.x. Для использования утилиты iptables требуются привилегии суперпользователя (root). Иногда под словом iptables имеется в виду и сам межсетевой экран netfilter.

Все пакеты пропускаются через определенные для них последовательности цепочек. При прохождении пакетом цепочки, к нему последовательно применяются все правила этой цепочки в порядке их следования. Под применением правила понимается: во-первых, проверка пакета на соответствие критерию, и во-вторых, если пакет этому критерию соответствует, применение к нему указанного действия. Под действием может подразумеваться как элементарная операция (встроенное действие, например, ACCEPT, MARK), так и переход в одну из пользовательских цепочек. В свою очередь, действия могут быть как терминальными, то есть прекращающими обработку пакета в рамках данной базовой цепочки (например, ACCEPT, REJECT), так и нетерминальными, то есть не



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

прерывающими процесса обработки пакета (MARK, TOS). Если пакет прошел через всю базовую цепочку и к нему так и не было применено ни одного терминального действия, к нему применяется действие по умолчанию для данной цепочки (обязательно терминальное).

1. Загрузите виртуальную машину с ОС Linux.

Проверьте наличие в составе ОС межсетевого экрана *iptables*. При необходимости доустановите/переустановите компоненты системы с учётом следующих требований: на виртуальной машине должны использоваться 2 сетевых адаптера, один из которых используется для связи с сетью учебной аудитории, а второй – для связи с виртуальной сетью *VMware vSphere*;

во внутренней сети *VMware vSphere* должно входить минимум 2 виртуальных машины. Используя справочные материалы, настройте следующую конфигурацию МЭ *iptables*: обеспечьте связь через виртуальную машину с МЭ сети класса и виртуальной сети; добавьте правило, запрещающее прохождение ICMP трафика между сетями.

2. Проверьте настройки МЭ, используя компьютеры в сети аудитории и *VMware vSphere*.



Лабораторная работа 6 Установка, настройка маршрутизатора сети

Цель работы

Изучить принципы маршрутизации в IP-сетях, получить практические навыки настройки программных маршрутизаторов на базе операционной системы (ОС) Linux с применением маршрутизации интерфейсов и маршрутизации виртуальных локальных сетей (VLAN).

Теоретические сведения

Маршрутизация включает в себя следующие частные задачи:

1. Обмен информацией о топологии сети. Реализуется протоколами маршрутизации.
2. Определение оптимальных маршрутов и построение таблиц маршрутизации.
3. Продвижение пакета маршрутизаторами на основании таблиц маршрутизации.

В данной лабораторной работе рассматриваются вопросы продвижения пакетов. Информация об оптимальных маршрутах представляется в маршрутизаторе в виде таблицы маршрутизации. В случае адресации без масок таблица маршрутизации имеет вид:

Таблица 1.

Формат таблицы маршрутизации без масок

Адрес назначения	IP след. маршр.	Интерф.	Метрика

В таблице:

- Адрес назначения – это IP-адрес сети или конкретного узла.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- IP следующего маршрутизатора – адрес следующего транзитного маршрутизатора на пути к адресу назначения.
- Интерфейс – это идентификатор сетевого интерфейса маршрутизатора, через который нужно передавать данные следующему маршрутизатору.
- Метрика – абстрактная характеристика качества маршрута. В качестве метрики может выступать количество транзитных узлов («хопов»), пропускная способность и т.п. Меньшее значение метрики всегда соответствует лучшему маршруту.

При поиске маршрута для продвижения IP-пакета предпочтение отдается специфическим маршрутам (то есть маршрутам до конкретного узла, а не сети), даже в том случае, если метрика этого маршрута хуже.

При использовании масок алгоритм работы маршрутизатора при продвижении пакета несколько усложняется. В таблицу маршрутизации добавляется колонка с маской, соответствующей адресу назначения. Рассмотрим пример:

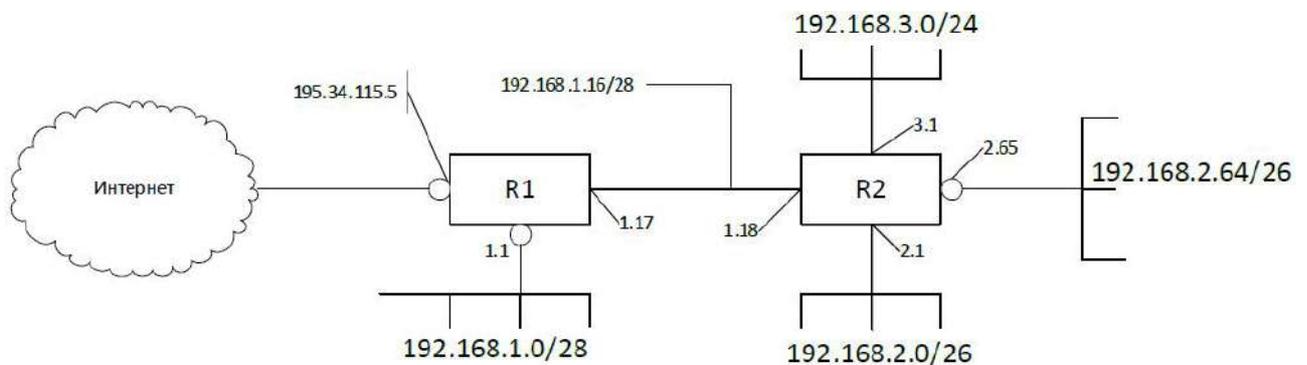


Рис. 1. Пример структуры сети

Для данного примера в таблице маршрутизации R2 могут быть следующие записи:



Пример таблицы маршрутизации с масками

Адрес назначения	Маска	Адрес след. маршр.	Интерфейс	Метрика
192.168.1.16	255.255.255.240	-	192.168.1.18	0
192.168.3.0	255.255.255.0	-	192.168.3.1	0
192.168.2.0	255.255.255.192	-	192.168.2.1	0
192.168.2.64	255.255.255.192	-	192.168.2.65	0
192.168.1.0	255.255.255.240	192.168.1.17	192.168.1.18	1
Default	0.0.0.0	192.168.1.17	192.168.1.18	16

Продвижение пакета основывается на следующем алгоритме:

1. Маршрутизатор извлекает из пакета IP-адрес назначения.
2. Поиск специфического маршрута, то есть записи, в которой адрес назначения равен целевому IP-адресу. Если запись найдена, то используется этот маршрут, иначе на шаг 3.
3. Поиск неспецифического маршрута. Включает следующие шаги:
 - для каждой записи таблицы маршрутизации выполняется операция $IPd \& M$, где IPd – целевой адрес из пакета, M - маска из записи таблицы маршрутизации;
 - если результат равен адресу назначения IP, запись отмечается подходящей.
4. Из всех подходящих маршрутов выбирается наиболее специфический. Такой маршрут содержится в записи с наиболее длинной маской.
5. Среди подходящих маршрутов, обладающих одинаковой "специфичностью", выбирается маршрут с меньшей метрикой.



Задание

Настроить взаимодействие двух IP-сетей между собой и с внешней сетью средствами программного маршрутизатора на базе ОС Linux (рис. 2).
Настроить простейшие правила фильтрации трафика средствами ОС Linux.

Ситуация 1. Сети изолированы друг от друга физически, т.е. построены на различных коммутаторах, не связанных друг с другом непосредственно.

Ситуация 2. Изоляция сетей обеспечивается за счет применения технологии виртуальных локальных сетей.

Проверить настройку маршрутизации и фильтров на примере взаимодействия рабочих станций PC-1 и PC-2, принадлежащих различным сетям.

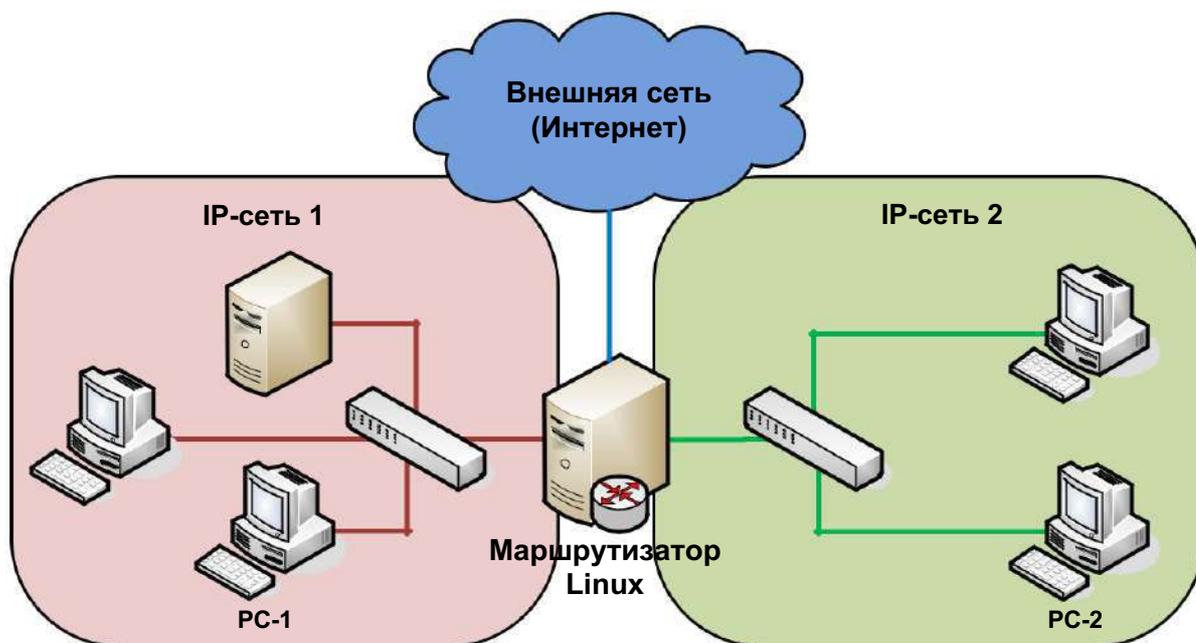
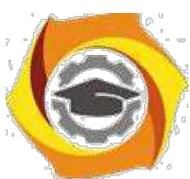


Рис. 2. Принципиальная схема взаимодействия сетей для лабораторной работы №1

Схема ЛВС

В приведённых примерах схем ЛВС для лабораторной работы указана адресация для компьютеров PC151 (ПК-1), PC152 (ПК-2), PC154 (ПК-4) лаборатории. При использовании других компьютеров IP-адреса будут другими (см. прил. 1).



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

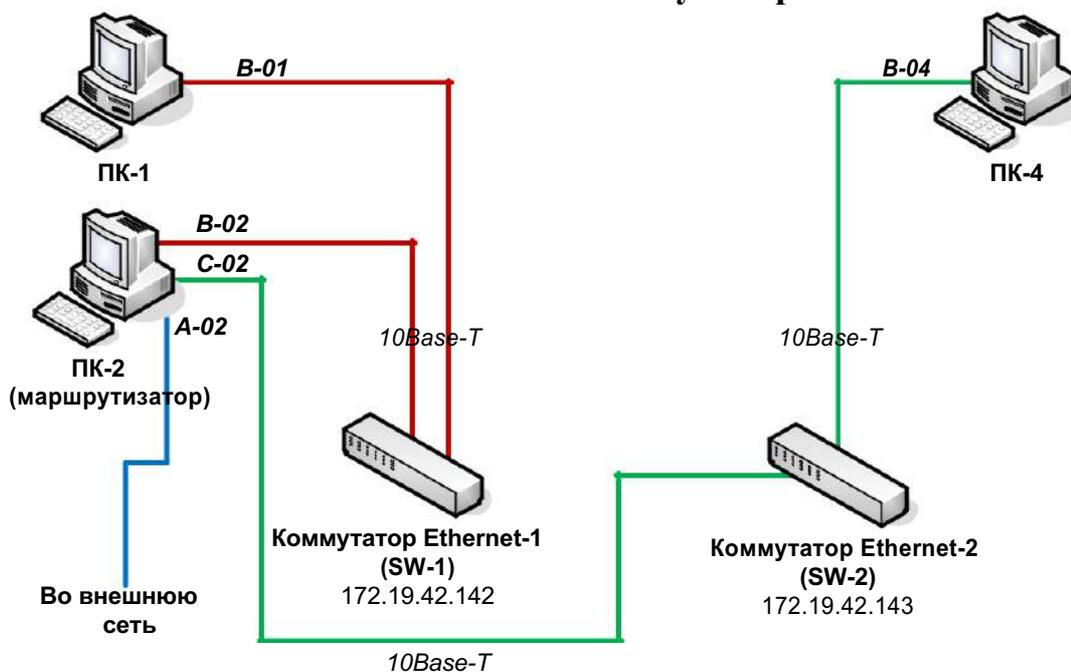


Рис. 3. Физическая схема ЛВС для лабораторной работы №1 (ситуация 1)

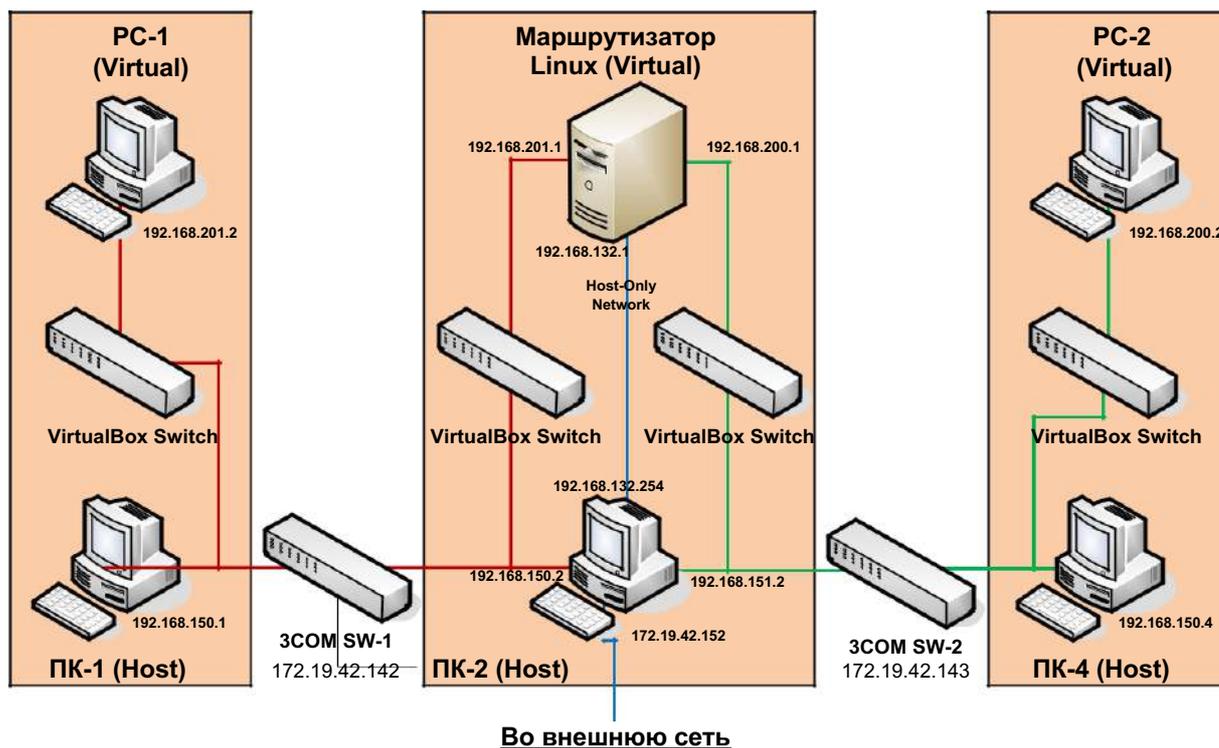
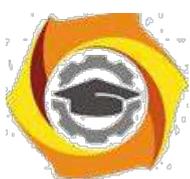


Рис. 4. Логическая схема ЛВС для лабораторной работы №1 (ситуация 1). IP-сеть 1 имеет адрес 192.168.201.0/24, IP-сеть 2 имеет адрес 192.168.200.0/24, внешняя сеть подключена через 192.168.132.0/24



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

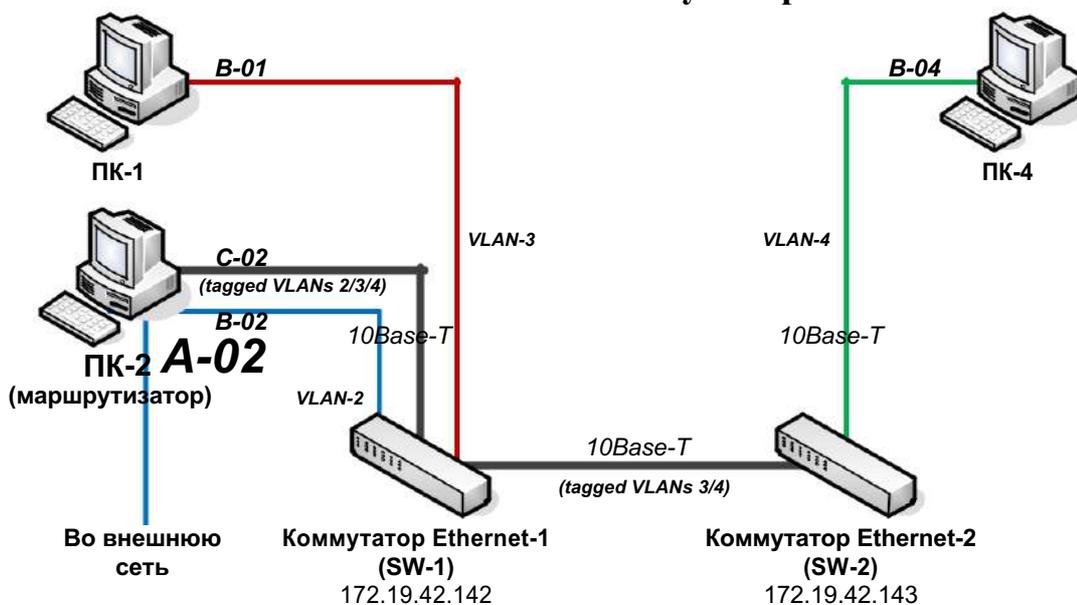


Рис. 5. Физическая схема ЛВС для лабораторной работы №1 (ситуация 2)

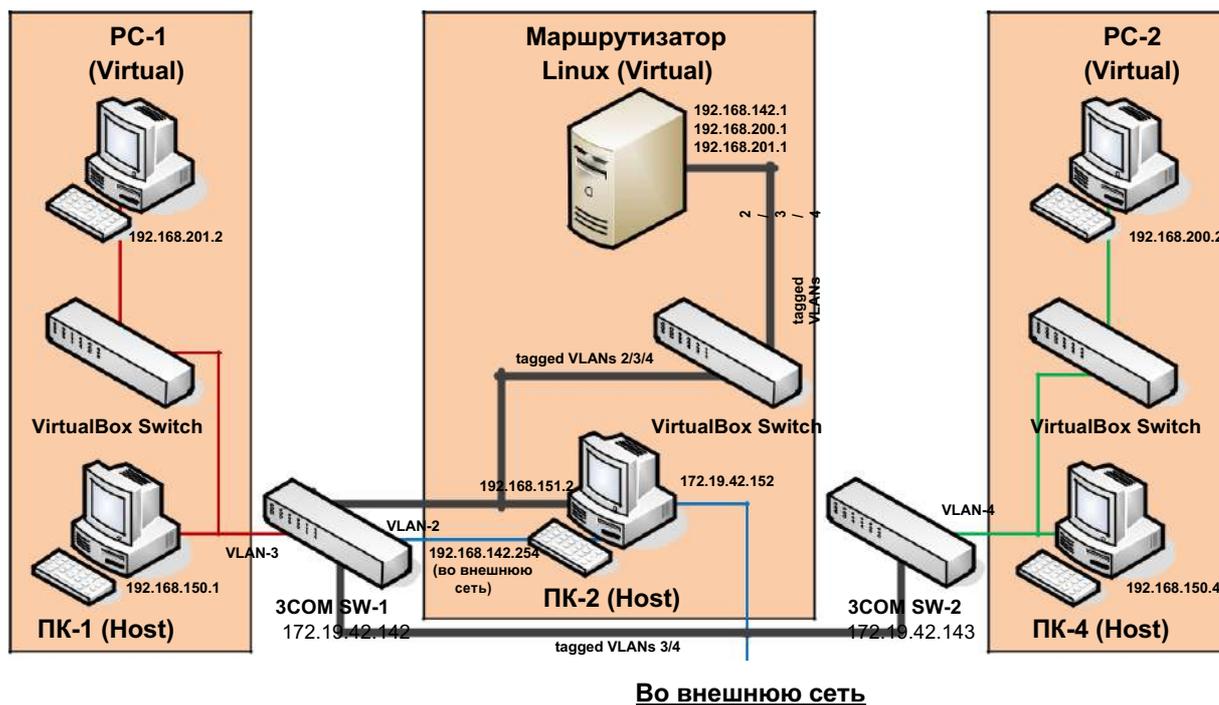


Рис. 6. Логическая схема ЛВС для лабораторной работы №1 (ситуация 2). IP-сеть 1 имеет адрес 192.168.201.0/24, IP-сеть 2 имеет адрес 192.168.200.0/24, внешняя сеть подключена через 192.168.142.0/24



Порядок выполнения работы

Перед выполнением лабораторной работы необходимо:

1. Подготовить схемы сети с указанием ip-адресов в соответствии с вариантом.
2. Подготовить виртуальные машины PC-1, PC-2, на которых может быть установлена любая сетевая операционная система, например Microsoft Windows XP или Linux.
3. Подготовить виртуальную машину, которая будет выполнять роль маршрутизатора. Рекомендуется использовать дистрибутив Debian GNU/Linux. Для данной виртуальной машины использовать сетевые адаптеры «*PCnet*», обеспечивающие передачу тэгированных кадров VLAN.

В установленной ОС Linux должна быть включена поддержка VLAN. В Debian необходимо установить соответствующий пакет командой

```
apt-get install vlan
```

Рекомендуется также установить графическую версию текстового редактора vi. В Debian соответствующий пакет устанавливается командой

```
apt-get install vi-gnome
```

Только для Debian! В случае, если образ ОС Debian Linux копируется/клонировается, необходимо отключить контроль MAC-адресов интерфейсов в файле «`/etc/udev/rules.d/75-persistent-net-generator.rules`» (изменять файл можно только с правами администратора). Для этого внести в приведенный ниже раздел строку с маской сетевых адаптеров VirtualBox и перезагрузить операционную систему.

```
# ignore interfaces with locally administered or null MAC addresses
```

```
# and VMWare virtual interfaces
```

```
ENV{MATCHADDR}=="?[2367abef]:*", ENV{MATCHADDR}=""
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

```
ENV{MATCHADDR}=="00:00:00:00:00:00", ENV{MATCHADDR}=""  
ENV{MATCHADDR}=="00:0c:29:*|00:50:56:*", ENV{MATCHADDR}=""  
ENV{MATCHADDR}=="08:00:27:*", ENV{MATCHADDR}=""
```

4. Подготовить коммутаторы 3COM Switch 1100 (SW-1, SW-2) для моделирования ситуаций 1, 2 задания (при подключении к коммутаторам использовать имя «manager», пароль – «superuser»). Для этого настроить:

- порты 1-2 коммутаторов SW-1, SW-2 для работы в VLAN-2 без тегирования;
- порты 3, 4 коммутатора SW-1 для работы в VLAN-3 без тегирования;
- порты 3, 4 коммутатора SW-2 для работы в VLAN-4 без тегирования;
- порты 5, 6 коммутаторов SW-1, SW-2 для работы в VLAN-2/3/4 с тегированием.

Выполнение лабораторной работы включает следующие этапы:

1. Подключение и запуск рабочих станций PC-1, PC-2

Цель данного этапа состоит в подготовке моделей IP-сетей, взаимодействие между которыми будет настраиваться в лабораторной работе.

1. Подключить образ виртуальной машины в системе VirtualBox на одном из компьютеров лаборатории. В настройках виртуальной машины в разделе «Сеть» включить 1 сетевой адаптер, для которого указать тип подключения «Сетевой мост» и сетевой адаптер «Realtek 8029» или «3COM» (в зависимости от компьютера).
2. Запустить виртуальную машину.
3. Настроить IP-адрес сетевого интерфейса виртуальной машины в соответствии с вариантом.



4. Подключить компьютер через соответствующий разъем патч-панели «B-0x» на порт коммутатора 1 или 2 (для 1-й ситуации из задания).
5. Повторить шаги 1-5 для 2-й виртуальной машины (на другом компьютере).

2. Подключение и настройка маршрутизации. Ситуация 1

Цель данного этапа состоит в настройке маршрутизации средствами ОС Linux для ситуации, когда на маршрутизаторе установлено 3 сетевых интерфейса. Фильтры на данном этапе не настраиваются. Выход во внешнюю сеть обеспечивается через VirtualBox Host-Only Network (адрес сети 192.168.13x.0/24) и маршрутизируется хост-компьютером.

1. Подключить хост-компьютеры в сеть в соответствии с физической схемой для ситуации 1. Использовать порты 1, 2 коммутаторов.
2. Подключить образ виртуальной машины под управлением ОС Linux в системе VirtualBox на одном из компьютеров лаборатории. В настройках виртуальной машины в разделе «Сеть» включить 3 сетевых адаптера и настроить следующим образом:
 - адаптер 1: тип подключения «Сетевой мост» и сетевой адаптер «Realtek 8029» или «3COM» (в зависимости от компьютера);
 - адаптер 2: тип подключения «Сетевой мост» и сетевой адаптер «D-Link DGE-528»;
 - адаптер 3: тип подключения «Виртуальный адаптер хоста» и сетевой адаптер «VirtualBox Host-Only Ethernet Adapter».
3. Запустить ОС Linux на виртуальной машине, запустить консоль с правами администратора.
4. Выключить сетевые интерфейсы командой

```
ifdown --all
```
5. Настроить статические адреса сетевых интерфейсов маршрутизатора. Для этого в файле /etc/network/interfaces для каждого сетевого интерфейса (eth0, eth1, eth2) задать ip-адрес, маску подсети и шлюз:



```
auto eth0
iface eth0 inet static
    address x.x.x.x
    netmask x.x.x.x
    gateway x.x.x.x
```

6. Включить сетевые интерфейсы с заданными настройками:

```
ifup --all
```

7. Проверить с помощью команды `ifconfig`, что требуемые настройки сетевых интерфейсов установлены.

8. Включить перенаправление пакетов с использованием команды:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

9. С использованием утилиты `ping` проверить возможность сетевого взаимодействия виртуальных рабочих станций PC-1, PC-2 с сервером и между собой (проверять по IP-адресам!). Проверить возможность доступа с сервера Linux к серверу кафедры Asuserv (для доступа к Asuserv с PC-1, PC-2 потребуется выполнить настройку NAT, см. п. 4).

3. Подключение и настройка маршрутизации. Ситуация 2

Цель данного этапа состоит в настройке маршрутизации средствами ОС Linux для ситуации, когда на маршрутизаторе установлен 1 сетевой интерфейс, а сегменты сети разделены за счет использования VLAN. Фильтры на данном этапе не настраиваются. Выход во внешнюю сеть обеспечивается через сеть 192.168.14x.0/24 и маршрутизируется хост-компьютером.

1. Подключить хост-компьютеры в сеть в соответствии с физической схемой для ситуации 2:

- ПК-1 (PC-1) включить в VLAN-3;
- ПК-4 (PC-2) включить в VLAN-4;
- ПК-2 (Router Linux): сетевой интерфейс 1 («Realtek 8029» или «3COM») через разъем B-0x включить в VLAN-2; сетевой интерфейс 2 (адаптер «D-Link DGE-528») через разъем C-0x



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

включить в порт коммутатора, настроенный для передачи тегированных кадров 802.1Q (VLAN-2/3/4);

- коммутаторы соединить друг с другом через порты, настроенные для передачи тегированных кадров 802.1Q (VLAN-2/3/4).
2. Подключить образ виртуальной машины под управлением ОС Linux в системе VirtualBox на одном из компьютеров лаборатории. В настройках виртуальной машины в разделе «Сеть» включить 1-й виртуальный сетевой адаптер и настроить его на работу в режиме «Сетевой мост» через физический адаптер «D-Link DGE-528». Другие виртуальные сетевые адаптеры выключить.
 3. Запустить ОС Linux на виртуальной машине, запустить консоль с правами администратора.

4. Создать виртуальные интерфейсы для каждой VLAN с использованием команды:

```
//добавление виртуального интерфейса для VLAN-2 на  
физический интерфейс eth0  
vconfig add eth0 2
```

5. Настроить способ формирования имен виртуальных интерфейсов в виде eth0.x, где x – идентификатор VLAN:

```
vconfig set_name_type DEV_PLUS_VID_NO_PAD
```

6. Выключить сетевые интерфейсы командой

```
ifdown --all
```

7. Настроить статические адреса виртуальных интерфейсов маршрутизатора. Для этого в файле /etc/network/interfaces для каждого виртуального интерфейса (eth0.x) задать ip-адрес, маску подсети и шлюз. При этом на сам физический интерфейс адрес не назначать:

```
auto eth0  
iface eth0 inet static  
    address 0.0.0.0  
    netmask 0.0.0.0  
auto eth0.2
```



```
iface eth0.2 inet static
    address x.x.x.x
    netmask x.x.x.x
```

8. Включить сетевые интерфейсы с заданными настройками:

```
ifup -all
```

9. Проверить с помощью команды `ifconfig`, что требуемые настройки сетевых интерфейсов установлены.

10. Включить перенаправление пакетов с использованием команды:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

11. С использованием утилиты `ping` проверить возможность сетевого взаимодействия виртуальных рабочих станций PC-1, PC-2 с сервером и между собой (проверять по IP-адресам!). Проверить возможность доступа с сервера Linux к серверу кафедры Asuserv (для доступа к Asuserv с PC-1, PC-2 потребуется выполнить настройку NAT, см. п. 4).

4. Настройка правил фильтрации и NAT

Цель данного этапа состоит в настройке простейших правил фильтрации на маршрутизаторе средствами ОС Linux для разграничения доступа между IP-сетями на уровне адресов и сетевых служб (по номерам портов). Средствами стандартного межсетевого экрана Linux настраивается также режим NAT для выхода во внешнюю сеть.

В примерах команд настройки правил фильтрации, приведённых ниже, предполагается, что через интерфейсы `eth0`, `eth1` подключены сегменты локальной сети, а через `eth2` организован выход во внешнюю сеть.

1. Настроить трансляцию адресов для интерфейса, обеспечивающего выход во внешнюю сеть (необходимо для корректной маршрутизации во внешней сети):

```
iptables -t nat -A POSTROUTING -o eth2 -j MASQUERADE
```

2. Проверить возможность доступа с виртуальных рабочих станций к внешнему серверу по адресу 172.19.42.10.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

3. Очистить все цепочки таблицы фильтрации и установить правила по умолчанию для цепочек INPUT, FORWARD, обеспечивающие удаление явно не обрабатываемых пакетов (для всех вариантов):

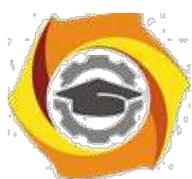
```
iptables -F
iptables -P INPUT DROP //отбрасывание всех входящих
пакетов,
//предназначенных для самого маршрутизатора
iptables -P FORWARD DROP //отбрасывание всех пакетов,
//требующих перенаправления на другой
интерфейс
```

4. Разрешить прохождение ICMP-трафика между сетями для обеспечения работы утилит ping и traceroute (для всех вариантов):

```
//для проверки связи с самим маршрутизатором
iptables -A INPUT -i eth0 -p icmp -j ACCEPT
iptables -A INPUT -i eth1 -p icmp -j ACCEPT
//для проверки связи между подсетями
iptables -A FORWARD -i eth0 -p icmp -j ACCEPT
iptables -A FORWARD -i eth1 -p icmp -j ACCEPT
```

5. Настроить правила доступа к ресурсам сети в соответствии с вариантом, например:

```
//разрешаем подключаться к Asuserv на порт TCP-110 (протокол
//доступа к почтовому ящику POP3) из сети, подключенной на
//интерфейс eth0
iptables -A FORWARD -i eth0 -d 172.19.42.10 -p tcp --dport
110 -j ACCEPT
//разрешаем прохождение трафика в обратном
направлении //для установленных соединений
iptables -A FORWARD -m state --state ESTABLISHED, RELATED -
j ACCEPT
//разрешаем подключаться клиентам из сети 10.1.1.0/24 к
//рабочей станции 10.1.2.1 по любому протоколу
iptables -A FORWARD -s 10.1.1.0/24 -d 10.1.2.1 -p ANY -j
ACCEPT
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**



6. Проверить взаимодействие узлов сети в соответствии с заданием, предусмотренным вариантом. Проверить невозможность других видов взаимодействия между сетями. Проверить невозможность доступа к маршрутизатору из внешней сети.

5. Сохранение сценария настройки маршрутизатора

Цель данного этапа состоит в создании файла сценария, определяющего сетевые настройки ОС Linux для работы в качестве маршрутизатора с правилами доступа, определенными вариантом. Сценарий должен обеспечивать конфигурацию экранирующего маршрутизатора, заданную вариантом, после перезагрузки операционной системы.

Содержание отчета

1. Титульный лист.
2. Цель работы, задание.
3. Схемы ЛВС с указанием IP-адресов по варианту.
4. Маршрутизация интерфейсов.
 - 4.1. Содержание файла «etc/network/interfaces» для ситуации 1.
 - 4.2. Результаты проверки достижимости из сети 1 для сети 2 и внешней сети.
5. Маршрутизация VLAN.
 - 5.1. Содержание файла «etc/network/interfaces» для ситуации 2.
 - 5.2. Результаты проверки достижимости из сети 1 для сети 2 и внешней сети.
6. Правила фильтрации.
 - 6.1. Результаты проверки доступности сетевых служб, предусмотренных вариантом.
 - 6.2. Результаты проверки невозможности сетевого доступа, не предусмотренного вариантом.
7. Сценарий запуска (файл сценария для п.5).



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Контрольные вопросы

1. В чем заключается основной принцип работы устройств: концентратор, хаб, повторитель?
2. Укажите недостатки структурирования сети на основе концентраторов?
3. Используя схему сети, приведите пример передачи кадров от узла отправителя к узлу получателю?
4. Каким из указанных в проекте устройств необходимо наличие физических адресов (MAC)?
5. Какую логическую топологию выстраивает сетевой концентратор? Какую физическую топологию сети используют при применении концентраторов?
6. В каком режиме разделения общей среды передачи данных функционирует концентратор?
7. Опишите сценарий возникновения и обработки коллизии в сети передачи данных, основанной на применении концентраторов?
8. Перечислите наиболее распространенные типы физических сетевых разъемов и номенклатуру соединительных кабелей, используемых на концентраторах?
9. Укажите назначение Uplink-порта на Ethernet-концентраторе и назначение тумблера Normal/Uplink?



Практическое занятие 7 Практическое изучение WIFI роутера

Цель: Изучить основные способы подключения устройств к беспроводной сети

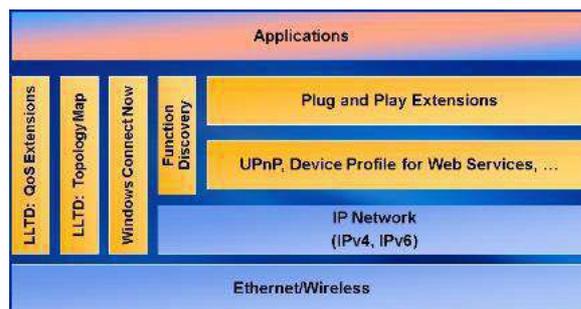
Краткие теоретические сведения

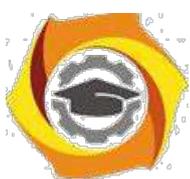
Использование различных устройств с WiFi стремительно врывается в нашу жизнь, сейчас WiFi оснащена не только сложная техника вроде ноутбуков и коммуникаторов, но и даже такие простые гаджеты, как фоторамка. Настраивать их для подключения к беспроводным сетям становится непростым делом, поэтому не удивительно появление технологий, позволяющих значительно упростить процедуру настройки.

Создание новой беспроводной сети начинается непосредственно с конфигурации точки доступа (беспроводного маршрутизатора) подключения к ней компьютеров и другого беспроводного оборудования.

Традиционный способ настройки с точки зрения обычного пользователя выглядит очень сложным: нужно произвести непростые действия с подключением к точке доступа для первой настройки, нужно создать вручную имя беспроводной сети, указать сложный и трудно воспроизводимый ключ безопасности. И весь этот процесс настройки требует, чтобы пользователь имели базовые знания о WiFi. Ему будет куда проще просто нажать на кнопку или ввести ПИН, чтобы все само настроилось и подключилось. **Wireless Protected Setup (WiFi Protected Setup - WPS)** как раз придумана, чтобы быстро, максимально просто и безопасно настраивать сетевые устройства и компьютеры. К примеру, один из способов подключения игровой консоли с WiFi к беспроводной сети: достаточно нажать на кнопки WPS на беспроводном маршрутизаторе и на игровой консоли.

WiFi Protected Setup сейчас является стандартом для простого и безопасного создания беспроводной сети. В терминологии Microsoft это **Windows Connect Now (WCN)**.

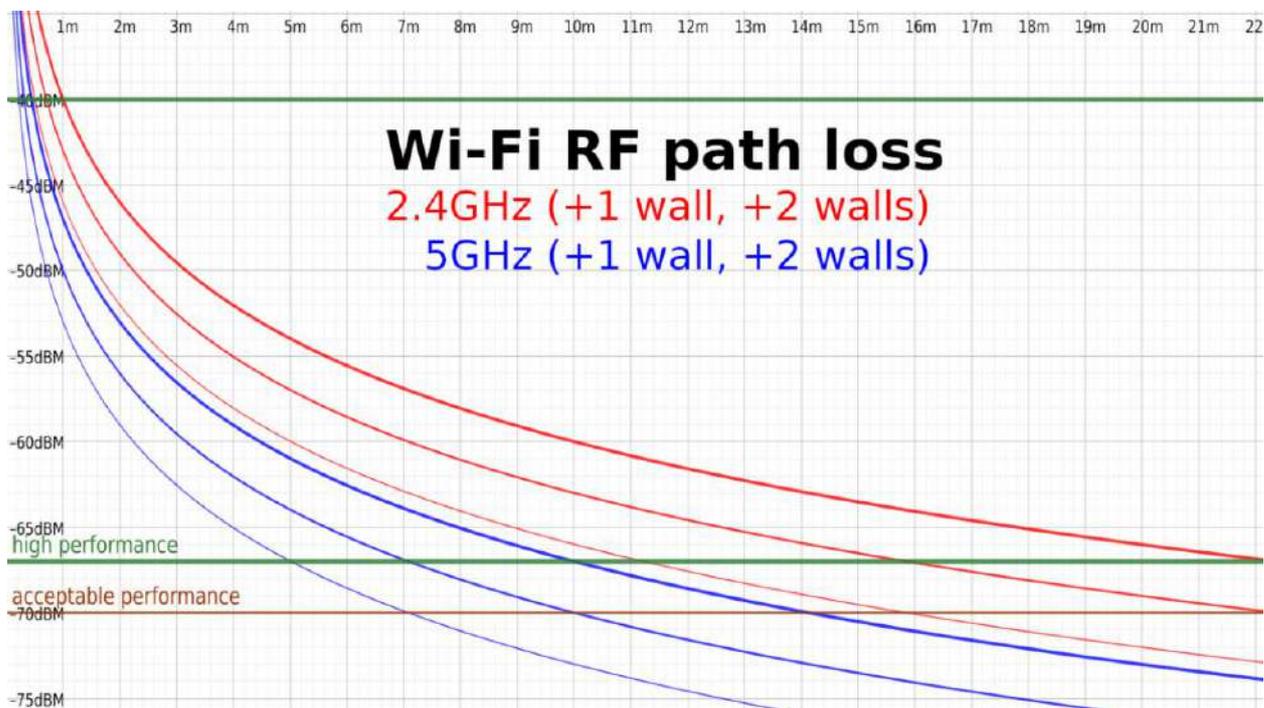




Негосударственное частное образовательное учреждение высшего образования «Технический университет»



Среда конфигурации беспроводных устройств может быть разной: **Ethernet**, **WiFi**, **USB кабель** или **USB флеш диск**. Конфигурация беспроводного устройства может осуществляться через ПИН-код (PIN - Personal Identification Number), через нажатия конфигурационных кнопок (PBC - Push Button Configuration, На качество связи сильное влияние оказывает расстояние и наличие препятствий между роутером и абонентским устройством



Каждое удвоение расстояния приводит к падению сигнала на 6 дБм, и мы чётко видим это, изучая толстую красную кривую для 2,4 ГГц: на 1 м сигнал -40 дБм, на 2 м это -46 дБм, на 4 м это -52 дБм.

Стены и иные препятствия – включая, но не ограничиваясь человеческие тела, шкафы, мебель, бытовую технику – ещё больше ослабят сигнал. Простое практическое правило — -3 дБм для каждой стены или другого значительного препятствия. Более тонкие линии того же цвета на графике показывают падение сигнала на тех же расстояниях при добавлении одной-двух стен (или других препятствий).

В идеале хочется иметь уровень сигнала не менее -67 дБм, однако не нужно беспокоиться о том, чтобы повышать его сильно выше этой отметки – обычно разницы в скорости между мощным -40 дБм и хилым -65 дБм нет, пусть они и находятся далеко друг от друга на графике. На работу WiFi влияет гораздо больше факторов, чем просто



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

мощность сигнала; как только вы превысите минимум, уже неважно, насколько именно вы его превысили.

На самом деле, слишком мощный сигнал может оказаться такой же проблемой, как и слишком хилый — многие пользователи на форумах жалуются на низкую скорость, пока какой-нибудь смыслённый человек не спросит их: вы что, разместили устройство прямо рядом с точкой доступа? Отодвиньте его на метр-два и попробуйте снова. И, конечно, проблема исчезает.

Правило 1: не больше двух комнат и двух стен

Наше первое правило для размещения точки доступа (ТД) — не больше двух комнат и двух стен между ТД и устройствами. Правило довольно расплывчатое, поскольку комнаты бывают разного размера и формы, а у разных домов разный состав стен — но это неплохая точка отсчёта, и она хорошо послужит вам в домах типичного размера и квартирах с достаточно современными межкомнатными стенами из гипсокартона.

Типичный размер, по крайней мере, в большей части США, означает спальни длиной 3-4 метра и гостиные длиной 5-6 метров по одной из стен. Если взять девять метров как среднюю дистанцию, покрывающую «две комнаты», и добавим две внутренних стены, по -3 дБм на каждую, наша кривая потеря радиоволн показывает, что сигналы на 2,4 ГГц будут прекрасно себя чувствовать с показателем -65 дБм. С 5 ГГц ситуация не такая хорошая — если нам потребуются все 9 метров и 2 стены, то мы опустимся до -72 дБм. Этого достаточно для установления связи, но и только. В реальной жизни устройство с сигналом на -72 дБм на 5 ГГц увидит примерно ту же пропускную способность, что и устройство на -65 дБм на 2,4 ГГц — однако формально более медленная связь на 2,4 ГГц окажется более стабильной и покажет меньшие задержки.

Конечно, всё это при условии, что единственными нашими проблемами будут расстояние и ослабление сигнала. Пользователи в сельской местности и в домах с большими участками уже наверняка заметили эту разницу и уяснили себе практическое правило «2,4 ГГц — это круто, а вот 5 ГГц — это полный отстой». У городских жителей или владельцев домов, стоящих на участке размером с почтовую марку, имеется совершенно другой опыт, который мы учтём во 2-м правиле.

Правило 2: слишком большая мощность передачи — это плохо

Плюсом сигнала на 2,4 ГГц служат дальнобойность и эффективное проникновение сквозь препятствия. Минусом сигнала на 2,4 ГГц служат... дальнобойность и эффективное проникновение сквозь препятствия.

Если два WiFi устройства на расстоянии «слышимости» друг от друга передают на одной и той же частоте одновременно, у них ничего не выходит: у устройств, для которых они передают сигнал, нет возможности разобраться в этом и понять, какой из сигналов предназначается для них. Вопреки распространённому мнению, тут совершенно неважно, находится ли устройство в вашей сети или нет — название и пароль WiFi не имеют никакого значения.

Чтобы по большей части избежать такой проблемы, любое WiFi устройство перед передачей должно сначала прослушать эфир — и если любое другое устройство уже передаёт на этом диапазоне частот, то наше должно заткнуться и подождать конца передачи. Это не устраняет проблему полностью; если два устройства решать передавать одновременно, они «столкнутся» — и каждому нужно будет выбрать случайный промежуток времени, которое они проведут в ожидании перед тем, как попытаться снова что-то передавать. Устройство, выбравшее меньшее случайное число, начинает первым —



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

если они не выберут одинаковое случайное число, или какое-то другое устройство не заметит передышку в эфире и не решит передавать сигнал, опередив обоих.

Это называется «затором», и для большинства современных пользователей WiFi это такая же большая проблема, как и ослабление сигнала. Чем больше у вас устройств, тем более загружена сеть. Каждое из ваших устройств может столкнуться с другим, и каждому приходится уважать правила пользования эфиром.

Если ваш роутер или ТД поддерживают такой вариант, то уменьшение мощности исходящего сигнала может наоборот, улучшить быстродействие и роуминг – особенно если у вас меш-набор или другая похожая схема. Сети 5 ГГц обычно не нужно так ослаблять, поскольку сигнал в том спектре и так достаточно быстро ослабляется, однако для 2 ГГц такой вариант может творить чудеса.

Последнее замечание для любителей «дальнобойных» ТД – такая ТД может и правда выдать сигнал сильнее обычной, и добить на большее расстояние. Однако она не может заставить ваш телефон или ноутбук усилить сигнал в ответ. При таком дисбалансе отдельные части веб-страницы могут загружаться быстро, однако в целом соединение будет казаться нестабильным, поскольку ваш ноутбук или телефон будет сначала с трудом загружать десятки или сотни отдельных запросов по HTTP/HTTPS, необходимых для загрузки каждой их веб-страниц.

Правило 3: используйте спектр с умом

Во втором правиле мы упомянули, что все устройства на одном канале соревнуются за эфирное время, вне зависимости от того, к какой сети они принадлежат. У большинства людей отношения с соседями не настолько хорошие, чтобы можно было убедить их понизить мощность передачи – даже если их роутер поддерживает такую функцию – но вы можете понять, какие каналы используют соседние сети, и избегать их.

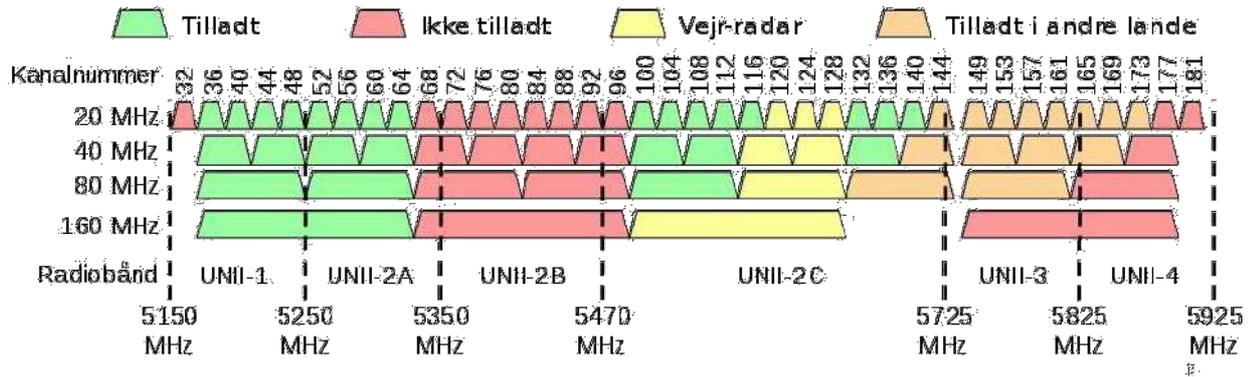
С 5 ГГц такой проблемы обычно не возникает, но на 2,4 ГГц это может довольно сильно влиять. Поэтому мы рекомендуем большинству людей избегать стандарта 2,4 ГГц. А где избегать его не получается, используйте приложение типа inSSIDer, чтобы периодически изучать своё радиоволновое окружение, и пытаться избегать использования самого загруженного спектра в районе вашего дома.

Однако это, к сожалению, может быть сложнее, чем кажется на первый взгляд. Неважно, сколько SSID вы увидите на определённом канале – важно, сколько эфирного времени они реально используют, а это нельзя подсчитать ни исходя из количества SSID, ни исходя из чистой мощности сигнала у видимых SSID. InSSIDer позволяет вам сделать ещё один шаг и изучить реальную утилизацию эфирного времени в каждом канале.

В сетях 5 ГГц загрузка каналов представляет собой гораздо меньшую проблему, поскольку уменьшение дальности действия и проникаемости сигнала означает наличие меньшего количества устройств, с которыми приходится соревноваться. Часто можно услышать заявления о том, что у этого стандарта больше каналов для работы, но на практике это не так, если вы не занимаетесь настройкой WiFi на территории вашего предприятия, где нет конкурирующих сетей. Домашние роутеры на 5 ГГц обычно настраивают на ширину канала 40 или 80 МГц, что означает, что непересекающихся каналов реально всего два – нижний, состоящий из каналов 36-64 шириной 5 МГц, и верхний, на каналах 149-165.



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»



Каждая сеть 5 ГГц шириной 40 МГц занимает чуть больше 8 реальных каналов шириной 5 МГц. Каждый пенёк тут символизирует четыре канала шириной 5 МГц.

В комментариях наверняка стоит ожидать дискуссии по поводу данных утверждений. Технически, можно уместить четыре сети шириной 40 МГц или две сети шириной 80 МГц на нижней части полосы 5 ГГц. На практике же потребительское оборудование работает через пень-колоду с накладывающимися каналами (к примеру, с полосой 80 МГц центрированной на канале 48 или 52), из-за чего такой эффективности спектра в реальных домашних условиях достичь сложно или практически невозможно.

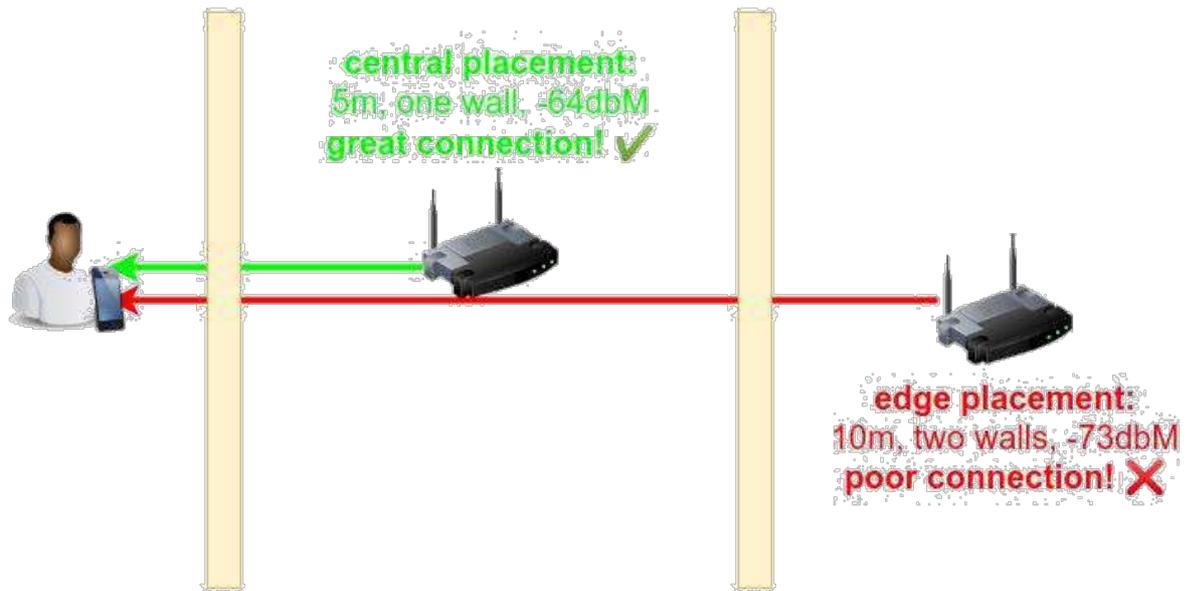
Между двумя стандартными потребительскими полосами (в США) есть ещё два канала с динамической частотой DFS (Dynamic Frequency Spectrum), однако их нужно делить с такими устройствами, как коммерческие и военные радары. Многие потребительские устройства отказываются даже пытаться использовать DFS. И даже если у вас есть роутер или ТД, согласные на использование DFS, они должны подчиняться строжайшим требованиям, чтобы не давать помех никаким радарам. Пользователи «в глуши» могут прекрасно использовать DFS – однако у них и проблем с загрузкой каналов, скорее всего, не будет.

Если вы живёте рядом с аэропортом, военной базой или портом, DFS вам, скорее всего, не подойдёт – а если вы живёте за пределами США, разрешённые у вас частоты могут отличаться от того, что описано тут (как DFS, так и остальные), в зависимости от местных законов.

Правило 4: лучший вариант – центральное размещение



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»



Разница между «роутером с краю дома» и «ТД посередине» может оказаться критической

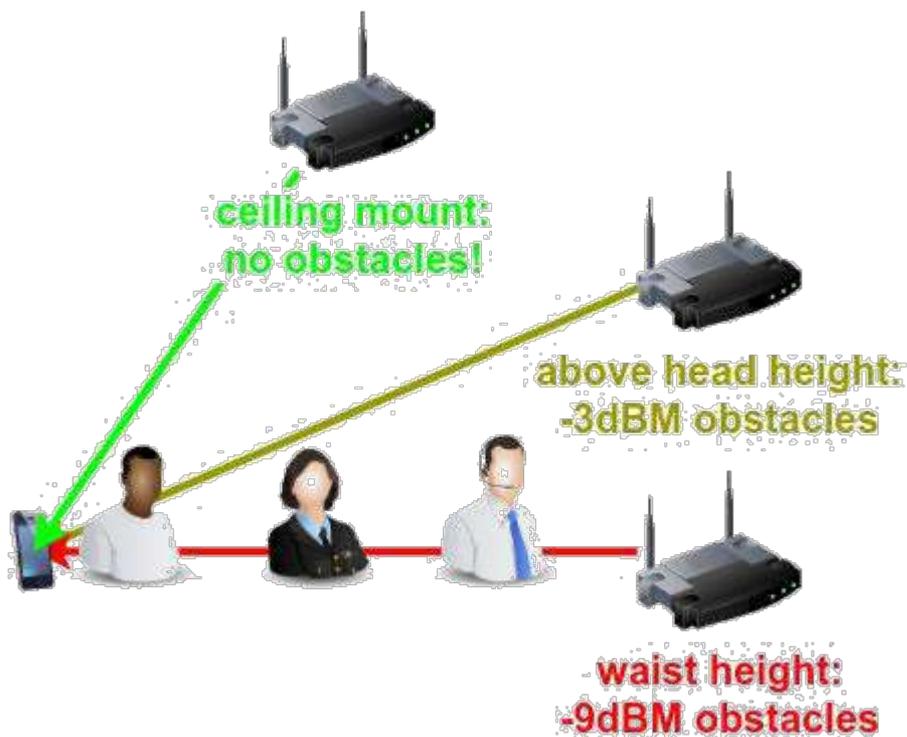
Возвращаясь к ослаблению сигнала, отметим, что идеальное место для расположения ТД WiFi – это центр пространства, которое ему нужно покрывать. Если ваше жилище имеет длину по одной из сторон 30 м, то роутеру, расположенному посередине, нужно будет покрыть только 15 м в каждую сторону, а роутеру с краю (где установщики от провайдера любят заканчивать коаксиальный кабель или линию DSL) придётся покрывать 30 м.

Это же справедливо и для меньших помещений с большим количеством ТД. Помните, сигналы WiFi быстро затухают. Шести метров – длины достаточно большой гостиной – может хватить для того, чтобы сигнал на 5 ГГц, ослабнув, опустился ниже оптимального уровня, если добавить туда пару препятствий типа мебели или людей. Что приводит нас к следующему правилу...

Правило 5: высота – выше человеческого роста



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»



Технически наилучшим расположением будет место у потолка – но если это слишком, то поместите ТД хотя бы наверху книжных полок.

Чем выше вы сможете закрепить ТД, тем лучше. Человеческое тело ослабляет сигнал примерно на столько же, на сколько и внутренняя стена – это одна из причин, по которой WiFi в вашем доме значительно ухудшается, когда на вечеринку пришло много друзей.

Разместив ТД – или роутер – выше человеческого роста, можно избежать необходимости передавать радиоволны сквозь все эти надоедливые и ослабляющие сигнал мешки с мясом. Также сигнал избегает большей части мебели и бытовой техники – диванов, столов, духовок и шкафов.

Самым идеальным вариантом будет размещение ТД на потолке в геометрическом центре комнаты. Если это невозможно, не беспокойтесь – почти так же хорошо будет поставить её наверх шкафа, особенно, если вам нужно, чтобы эта ТД обслуживала как ту комнату, где она стоит, так и комнату с другой стороны стены.

Правило 6: делите расстояния пополам

Допустим, некоторые ваши устройства расположены слишком далеко от ближайшей точки доступа для того, чтобы получить хороший сигнал. Вам повезло купить расширяемую систему, или у вас осталась одна ТД из меш-кита. Где её поместить?

Мы наблюдали замешательство людей в подобной ситуации, размышлявших о том, стоит ли поместить дополнительную ТД поближе к первой (с которой она берёт данные) или поближе к самым дальним устройствам (к которым она должна передавать данные). Ответ обычно такой: ни то, ни другое. Размещайте вашу ТД прямо посередине между ближайшей ТД и самым дальним клиентом, которого она должна обслуживать.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Суть в том, что вы пытаетесь сохранить эфирное время, организовав наилучшее соединение из возможных между дальними устройствами и новой ТД, и между новой ТД и ближайшей к ней. Обычно не стоит отдавать предпочтение одной из сторон. Однако не забывайте правило 1: две стены, две комнаты. Если нельзя разбить расстояние между самыми дальними клиентами и основной ТД, не нарушая первого правила, тогда размещайте новую ТД так далеко, как это позволяет первое правило.

Если вам это кажется слишком простым и логичным, не волнуйтесь: есть ещё один момент «только если не», который необходимо учитывать. У некоторых меш-наборов, например, Netgear's Orbi RBK-50/RBK-53 или Plume's Superpods, связь между ТД имеет очень высокую пропускную способность и работает по схеме 4x4. Поскольку это соединение работает гораздо быстрее 2x2 или 3x3, доступных клиентам, возможно, стоит уменьшить качество сигнала связи между этими ТД, так, чтобы их пропускная способность была ближе к той, которую могут позволить себе лучшие из ваших клиентов.

Если ваш меш-набор предлагает очень быстрое соединение между ТД, и вам никак не удаётся добавить к схеме дополнительных ТД, то вам, возможно, лучше будет поместить последнюю ТД ближе к клиентам, чем к предыдущей ТД. Однако тут придётся поэкспериментировать и изучить результаты.

Прикольная штука — WiFi, не правда ли?

Правило 7: обходите препятствия



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»



Плотно набитый книжный шкаф – серьёзное препятствие для радиоволн. Он стоит пары обычных стен даже при перпендикулярном проникновении. А уж в длину его пересекать вообще бесполезно.

Если вам досталось особо сложное помещение, в нём могут оказаться такие места, куда сигнал просто не сможет пройти. В нашем испытательном доме была бетонная плита и несколько метров плотной земли, закрывавшие линию видимости между роутером и

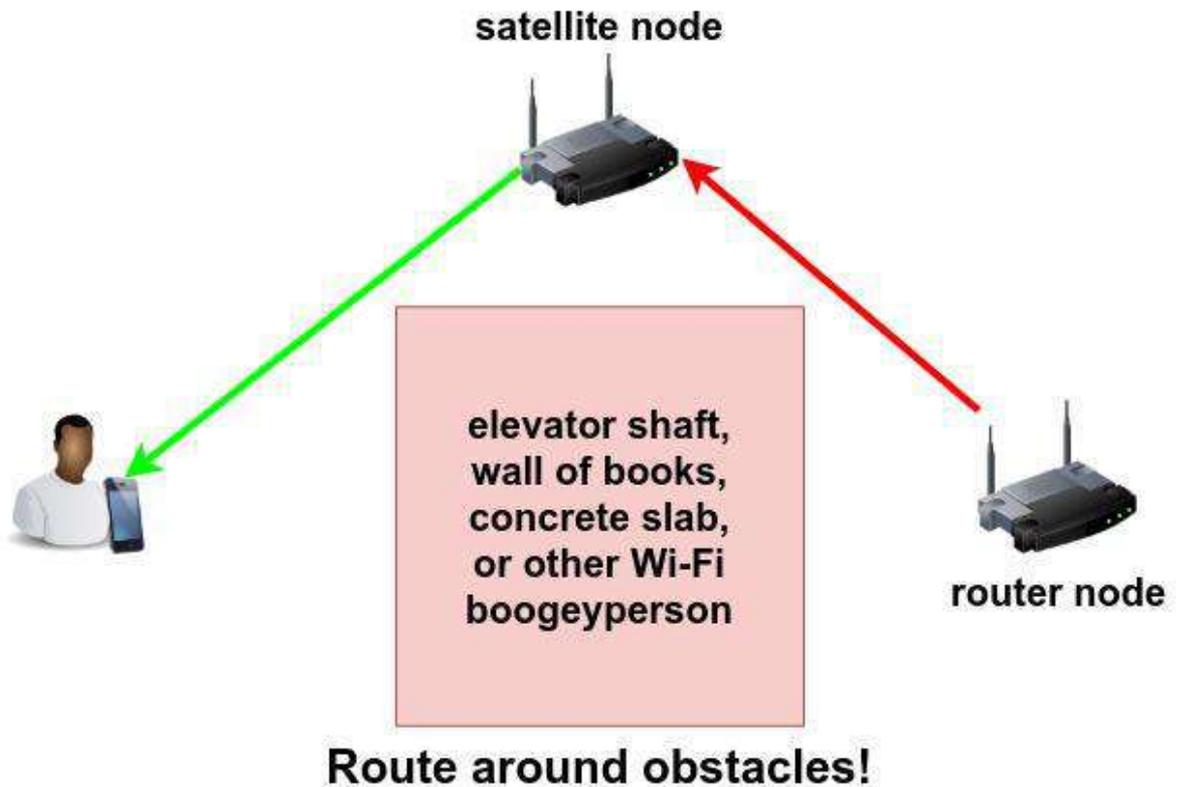


Негосударственное частное образовательное учреждение высшего образования «Технический университет»

подвалом. Мы встречали небольшие предприятия, точно так же обеспокоенные тем, что в одной части помещения WiFi работал хорошо, а в другой его не было – и в итоге оказывалось, что на пути сигнала стоит, например, книжный шкаф, забитый книгами, и расположенный вдоль коридора, из-за чего на пути сигнала оказывались несколько метров ослабляющей его переработанной древесины.

В каждом из случаев решением будет создание обходного пути вокруг препятствия при помощи нескольких точек доступа. Если у вас есть меш-набор WiFi, используйте его так, чтобы сигнал обходил препятствия. С одной стороны препятствия поместите ТД на линии прямой видимости с основной, причём так, чтобы её было видно с другой стороны препятствия, и сигналу не нужно было идти насквозь.

С достаточным количеством ТД и тщательным их размещением вы, возможно, сможете справиться даже со стенами, сделанными из дранки и металлической сетки, как строили в США в начале XX века. Мы видели, как люди успешно размещали ТД в прямой видимости друг друга через дверные проёмы и коридоры, когда для проникновения сквозь стены проще было бы использовать перфоратор.



Если слишком большое количество препятствий не даёт вам обойти их сбоку, сверху или снизу – смотрите правило 8.

Правило 8: всё дело в связи между точками доступа

Большинство потребителей выбирают чистые меш-наборы WiFi, поскольку это удобно – не нужно вести провода, просто подключаете кучу точек доступа, и пусть они там осуществляют свою магию между собой самостоятельно, без шума и пыли.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Звучит удобно, но на самом деле это самое плохое решение. Помните, мы говорили о правилах 2 и 3? Эти проблемы есть и здесь. Если вашему устройству нужно общаться с одной ТД, которой нужно передавать данные в другую ТД, то вы уже занимаете чуть более, чем в два раза больше эфирного времени.

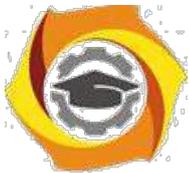
Ладно, на самом деле не так всё плохо – вы удваиваете использование эфирного времени, если ваш клиент находится там же, где и вспомогательная ТД. А поскольку вы последовали правилу 6 – поделили расстояния пополам – это значит, что качество связи у основной ТД с клиентом гораздо лучше той, которую организовал был клиент, подключаясь к основной ТД напрямую. Так что даже в самом худшем случае – когда вспомогательная ТД беседует с клиентом на том же канале, на котором она беседует с основной ТД – у них получится передавать данные, потребляя меньше эфирного времени, чем если бы один клиент работал с гораздо более длинным и менее качественным соединением.

Однако гораздо лучше будет полностью избежать этой проблемы, если ваши ТД будут общаться друг с другом на другой частоте. Двухполосные ТД могут делать это, общаясь с клиентами в диапазоне 2,4 ГГц, а между собой – на 5 ГГц, или наоборот. В реальном мире упрямые клиенты (и пользователи) часто хотят соединиться не так оптимально, в итоге получается, что клиенты есть и на 2,4 ГГц, и на 5 ГГц, поэтому «чистого» канала для внутренней связи не остаётся.

Особо умные наборы, такие, как Eero, могут избежать такой ситуации благодаря динамической маршрутизации внутренней связи, минимизируя зазоры путём передачи в диапазоне, отличной от того, в котором они ведут приём, даже когда диапазоны меняются. Самые продвинутые трёхполосные наборы типа Orbi RBK-50/53 или Plume Superpods могут избежать такой проблемы, используя второй передатчик на 5 ГГц. Это позволяет им соединяться с клиентами либо по 2,4 ГГц, либо по 5 ГГц, оставляя себе незанятый диапазон на 5 ГГц. У Orbi передатчик для внутренней связи фиксированный и выделенный. Plume принимает решения по использованию частот в зависимости от того, какой вариант его облачный оптимизатор считает наилучшим в конкретном окружении).

Лучший вариант – вообще не использовать WiFi для внутренней связи. Если можно проложить Ethernet-кабель, надо так и сделать. Он не только быстрее WiFi, он ещё и не страдает от проблем с загруженностью каналов. При высокой загрузке сети дешёвые проводные ТД типа Ubiquiti UAP-AC-Lites или TP-Link EAP-225v3s удельывают сухую даже самые дорогие меш-наборы, если последние ограничены внутренней связью по WiFi. Проводная внутренняя связь также решает проблему непрозрачных для радиоволн препятствий – если сквозь него нельзя пробить сигнал или обойти его, то протянутый сквозь него кабель творит чудеса!

Пользователям, которым не удалось реализовать ни меш-наборы с WiFi, ни протянуть кабели Ethernet, стоит рассмотреть современное оборудование для передачи сигналов по линиям электропередач. Результаты могут быть совершенно разные, и зависеть от качества проводки в доме и даже от типа подсоединённых бытовых приборов, но в большинстве случаев достаточно надёжным будет оборудование серий AV2 (AV1000 и выше) или g.hn, задержки передачи будут достаточно низкими, сравнимыми с Ethernet. Пропускная способность жёстко ограничена – в реальном мире стоит ожидать не более 40-80 Мб/с для домашних условий. Если вы в интернете занимаетесь только играми или просмотром веб-страниц, тогда передача данных по электропроводке может стать гораздо лучшим решением, чем WiFi.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Пойдя по этому пути, обязательно читайте инструкцию и принимайте меры для шифрования связи. В первый раз мы при проверке такого оборудования случайно построили мост с соседом, и перенастроили его роутер – он был почти такой же модели, что и наш, а пароль на нём стоял по умолчанию. «Здравствуйте, я взломал ваш роутер, прошу прощения» – плохой способ познакомиться, не рекомендуем.

Правило 9: обычно проблемы не в пропускной способности, а в задержках

У пропускной способности хорошо то, что это один красивый яркий номер, который легко получить, соединившись с сайтом для проверки скорости или используя инструмент типа `iperf3` для связи с локальным сервером.

Плохо у пропускной способности то, что это ужасный способ измерять как впечатление пользователя от сети, так и то, как сеть WiFi ведёт себя под реальной нагрузкой. Большинство людей расстраивает их WiFi-сеть либо при просмотре веб-страниц, либо в играх – а не тогда, когда они скачивают большой файл. В обоих случаях проблема не в том, «сколько мегабит в секунду может выдержать эта труба» – а в том, «сколько миллисекунд уходит на завершение конкретного действия».

И хотя можно увидеть ухудшение качества работы загруженной сети по уменьшающимся цифрам «скорости» скачивания, это более сложный, запутанный, и не связанный с реальностью способ, по сравнению с изучением задержек приложений. Задержки являются функцией как от просто скорости, так и от эффективности обработки сетью трафика и эфирного времени.

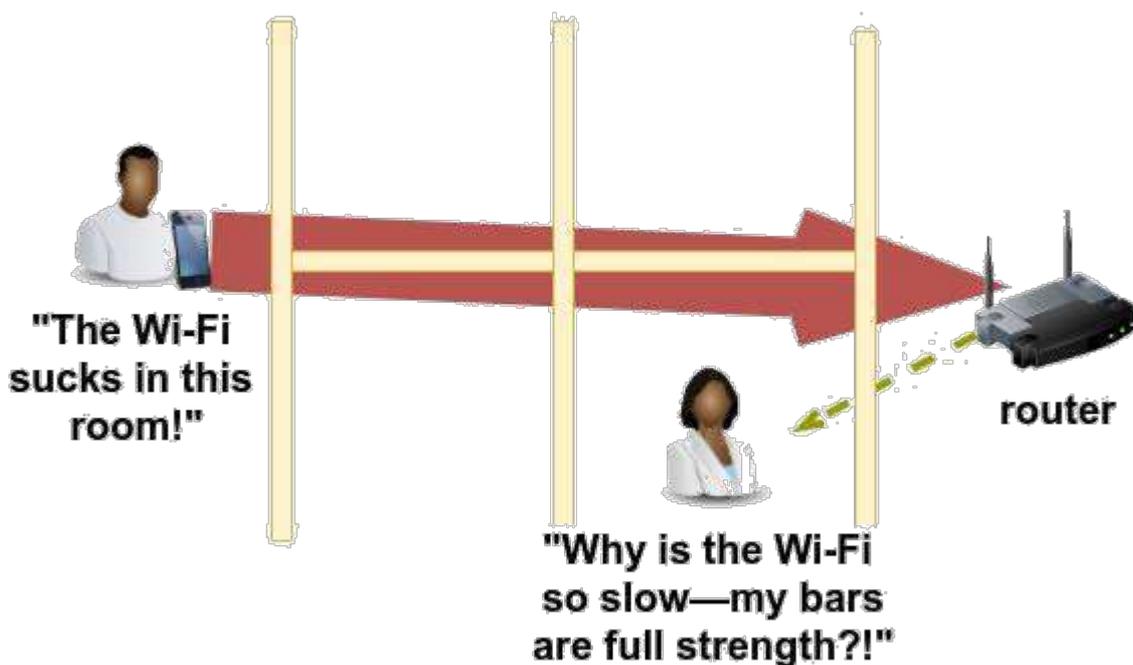
При проверке WiFi-сетей наша любимая метрика – это задержка в приложениях, которую мы симулируем загрузкой достаточно сложной веб-страницы. Что ещё важнее, нужно измерять загрузку страницы параллельно со всей остальной активностью в сети. Помните описание заторов в правилах 2 и 3 – «очень быстрая» сеть с одним активным устройством может превратиться в кошмарного тормоза со многими устройствами, или, во многих случаях, с одним плохо подключённым устройством, что приводит нас к последнему правилу.

Вывод 9-го правила такой, что рекламируемая скорость, идущая после букв AC в модели, — это фигня. Нужно доверять тщательным, технически компетентным обзорщикам, а не рейтингу скорости производителя на коробке.

Правило 10: скорость вашей сети WiFi ограничена скоростью самого медленного из подключённых устройств



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»



Одно устройство с хреновым подключением может убить качество связи для всей сети и всех подсоединённых устройств

К сожалению, один человек, пытающийся посмотреть ролик с YouTube в «спальне с хреновым приёмом», мучается не только сам – его проблемы наступают и других. Телефону, находящемуся в одной комнате с ТД, нужно всего лишь около 2,5% имеющегося эфирного времени для потоковой передачи ролика в качестве 1080P на скорости 5 Мб/с. Но телефон «в плохой спальне», мучающийся из-за буферизации и медленной связи, может забрать себе 100% эфирного времени сети, и не сумеет при этом посмотреть то же самое видео.

Конечно, потоковое видео очень сильно занимает входящий канал, и роутеры или ТД обычно отказываются вести передачу 100% времени. ТД, которой нужно передать большое количество данных, обычно оставит немного эфирного времени для других устройств и запроса собственных данных, а потом оно разобьёт время на скачивание между близлежащим устройством и «плохой спальней», чтобы попытаться выполнить оба запроса. Но это всё равно увеличивает время ожидания окна от этих устройств на сотни миллисекунд, и им всё равно приходится соревноваться друг с другом при открытии этого окна.

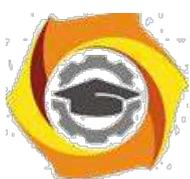
Ситуация ухудшается, если пользователь в «плохой спальне» пытается загрузить видео, отправить емейл или запостить большую фотку в соцсеть. Роутер пытается оставить часть эфирного времени другим устройствам – однако на телефон пользователя эти ограничения не действуют, и он с радостью сожрёт всё доступное эфирное время. Что хуже, телефон не представляет, сколько данных запросили другие пользователи в те краткие моменты, когда у них было окно для запросов. Роутер знает, сколько данных нужно доставить каждому из клиентов, поэтому он может размещать время для скачивания данных сообразно – но всё, что знает телефон, это то, что ему нужно закачать свои данные, поэтому пока он этим занимается, страдают все остальные. Поэтому даже если из всей этой мудрости вам нужно оставить только одно правило, пусть это будет правило 10.



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

1. Контрольные вопросы

1. Перечислите набор стандартов IEEE для организации беспроводной связи WiFi?
2. Укажите рабочие частотные диапазоны стандартов 802.11a, 802.11b, 802.11g и 802.11n?
3. Объясните назначение и функции следующего оборудования: беспроводная сетевая
4. карта (WIC), точка доступа (AP), беспроводной маршрутизатор (WR)?
5. В чем заключается различие между топологиями беспроводной сети Ad-hoc и Hotspot?
6. Укажите способы аутентификации беспроводного клиента? Приведите примеры практического использования указанных способов?
7. Приведите способы защиты доступа к информационной структуре беспроводной сети?
8. Дайте определение понятиям: SSID, WPA, WEP, WLAN, OFDM, DSSS?
9. Укажите методы первоначального конфигурирования класса беспроводного оборудования?
10. Объясните необходимость назначения IP-адреса беспроводной точке доступа?

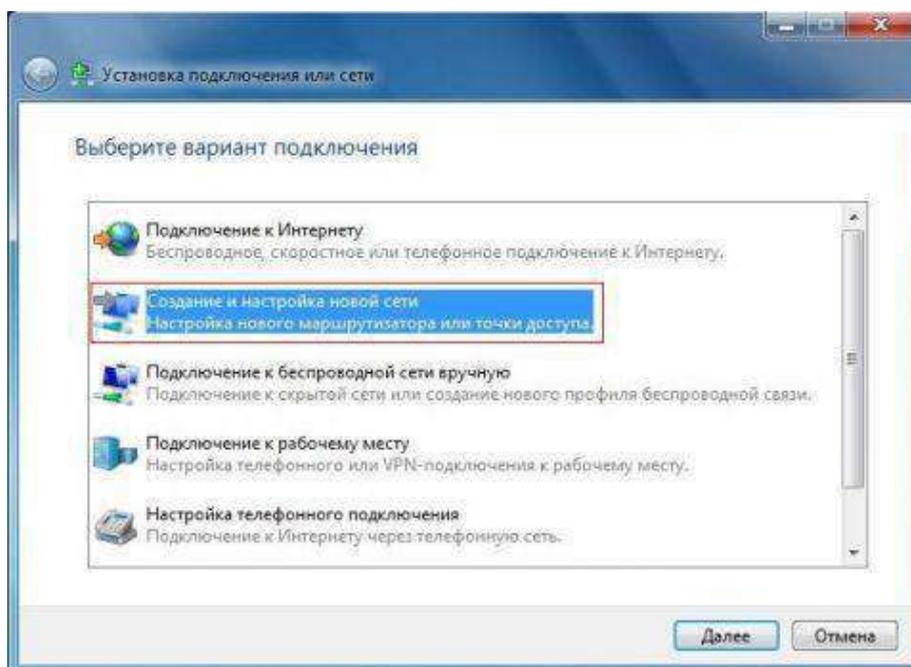


Лабораторная работа 7 Установка и настройка и конфигурирование WiFi роутера

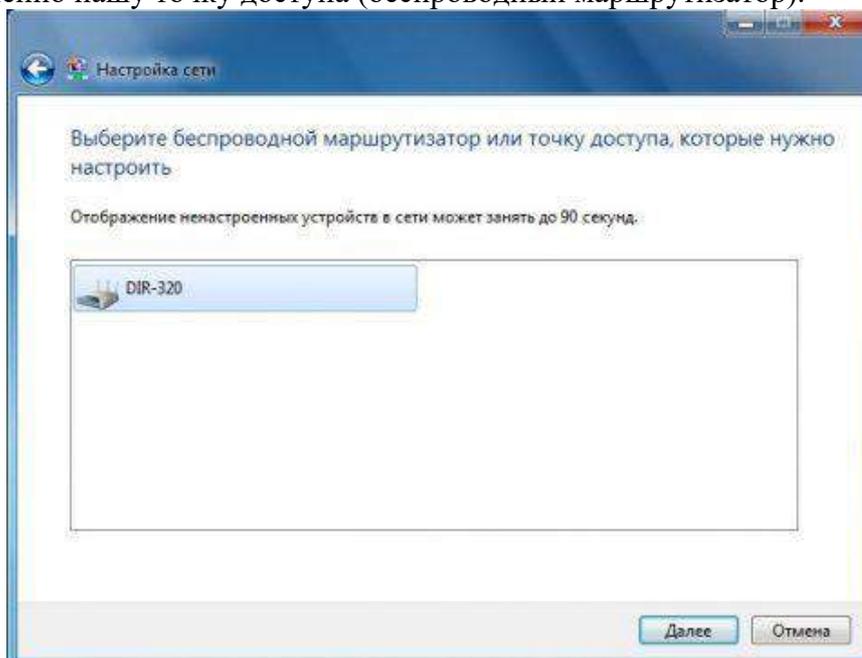
2. Порядок выполнения работы:

1.1 Настройка беспроводного маршрутизатора

1.1.1 Устройство распаковано и подключено к электросети. Настраивать его можно через Ethernet, используя при этом патчкорд (который входит в комплект поставки) или через WiFi, но от этого никак не зависит сам процесс настройки. На ноутбуке или десктопе нужно зайти в **Панель Управления – Центр управления сетями и общим доступом – Настройка нового подключения или сети**, где выбрать **Создание и настройка новой сети**.



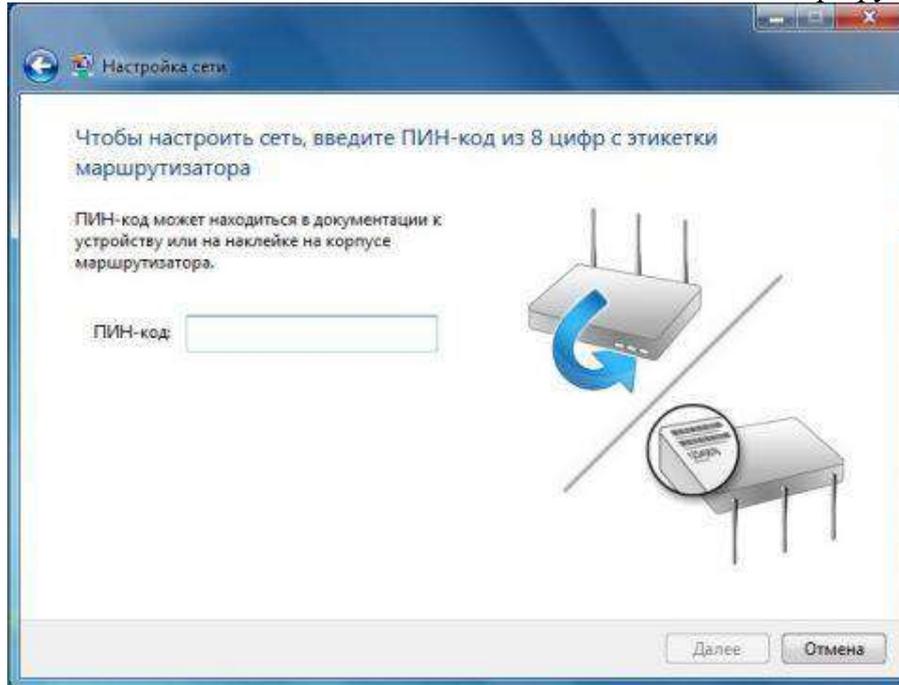
1.1.2 В списке устройств будут видны беспроводные устройства с поддержкой WCN. Выбираем именно нашу точку доступа (беспроводный маршрутизатор).



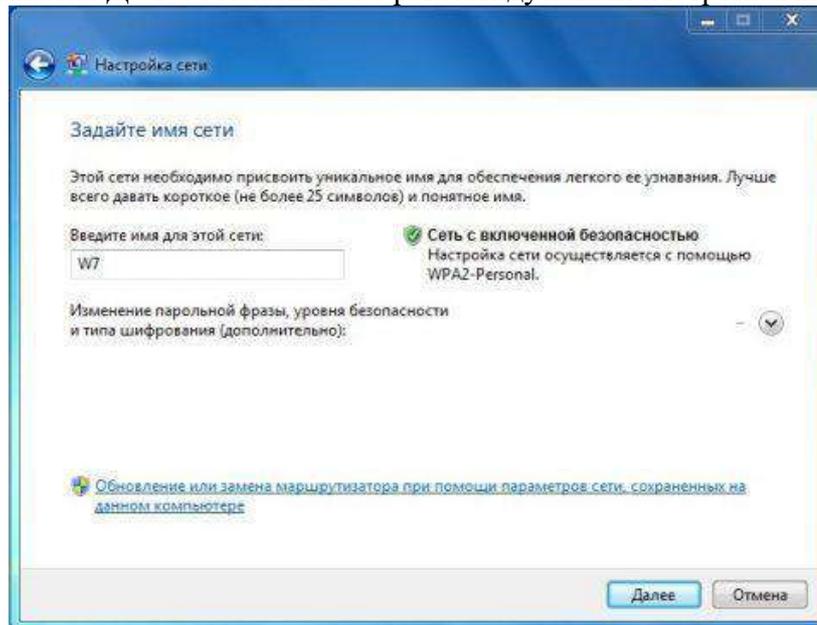


Негосударственное частное образовательное учреждение высшего образования «Технический университет»

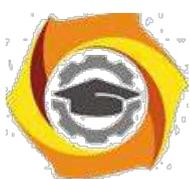
1.1.3 Следующим этапом необходимо ввести ПИН-код с этикетки на маршрутизаторе.



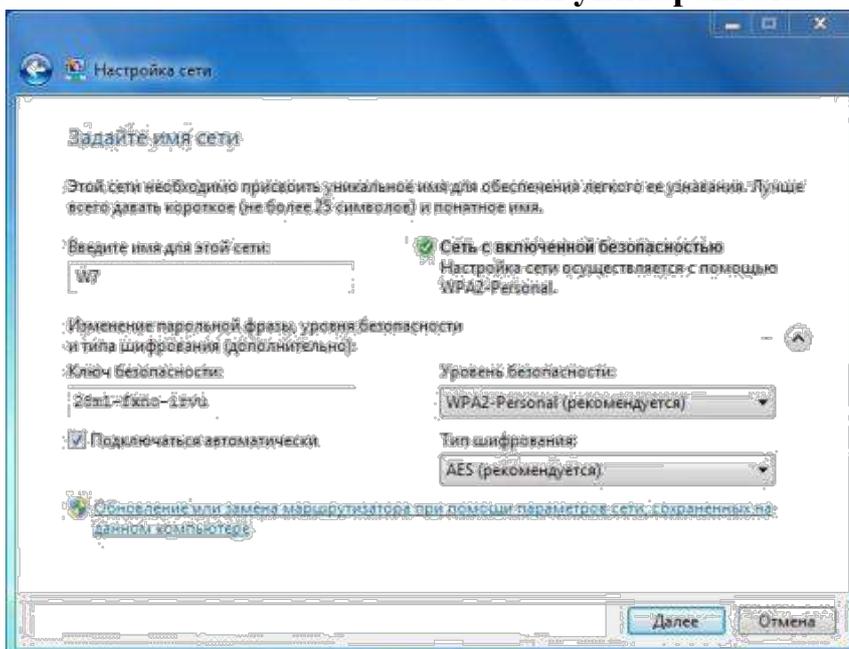
1.1.4 и после нажатия **Далее** согласиться с рекомендуемыми настройками точки доступа



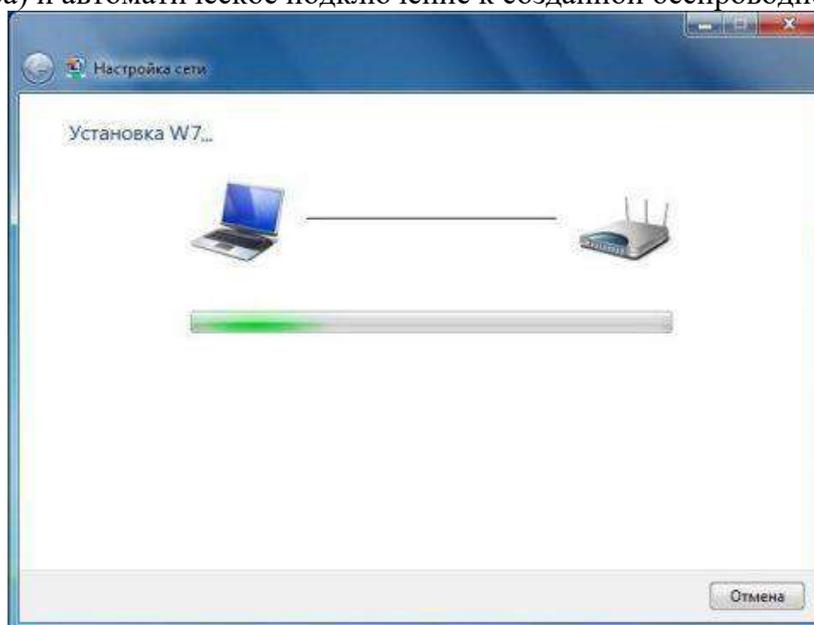
1.1.5 Или задать свои, есть в этом есть необходимость: имя беспроводной сети, пароль для доступа к сети, уровень безопасности и тип шифрования.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

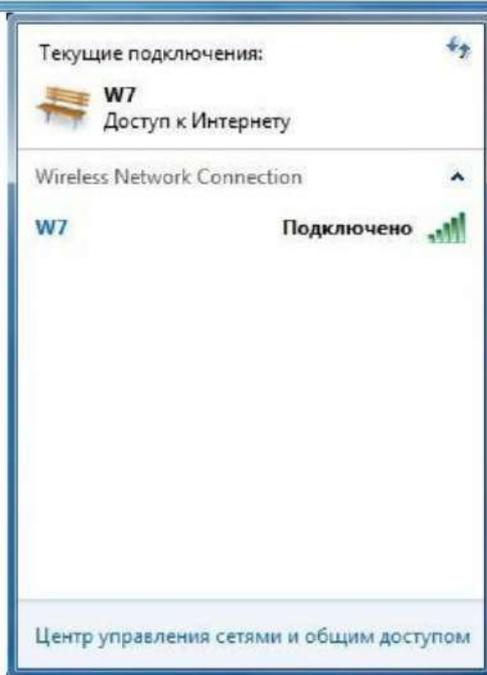
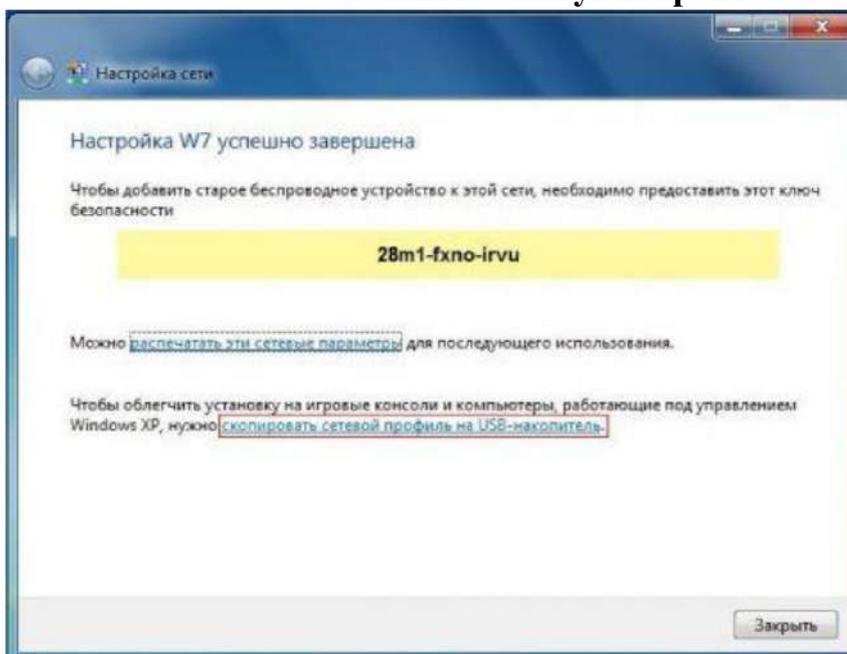


1.1.6 После нажатия кнопки **Далее** произойдет настройка точки доступа (беспроводного маршрутизатора) и автоматическое подключение к созданной беспроводной сети.





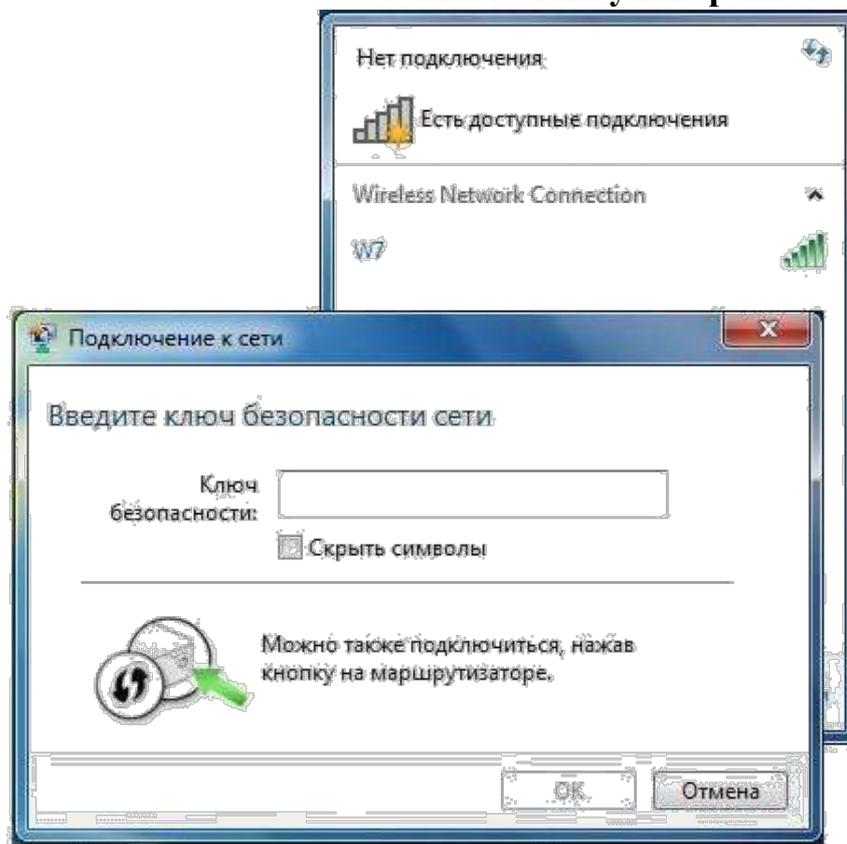
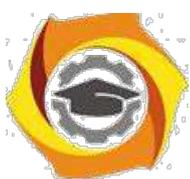
**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**



1.1.7 По завершении настройки можно распечатать подробную инструкцию для подключения остальных компьютеров к точке доступа (беспроводному маршрутизатору), а также подготовить флешку с настройками для импорта сетевого профиля на другие беспроводные устройства. Если в данный момент в этом нет необходимости, то это можно сделать позже, в свойствах беспроводной сети.

1.2 Подключиться к точке доступа через Push Button

1.2.1 При подключении к нашей беспроводной сети с компьютера под управлением [Windows 7](#), можно не вводить ключ безопасности, а нажать кнопку WCN на маршрутизаторе. Подключение к беспроводной сети произойдет автоматически.



1.2.2 На беспроводных устройствах, поддерживающих метод PBC, достаточно нажать кнопку WPS на маршрутизаторе, а потом на беспроводном устройстве, после чего произойдет подключения устройства к беспроводной сети.

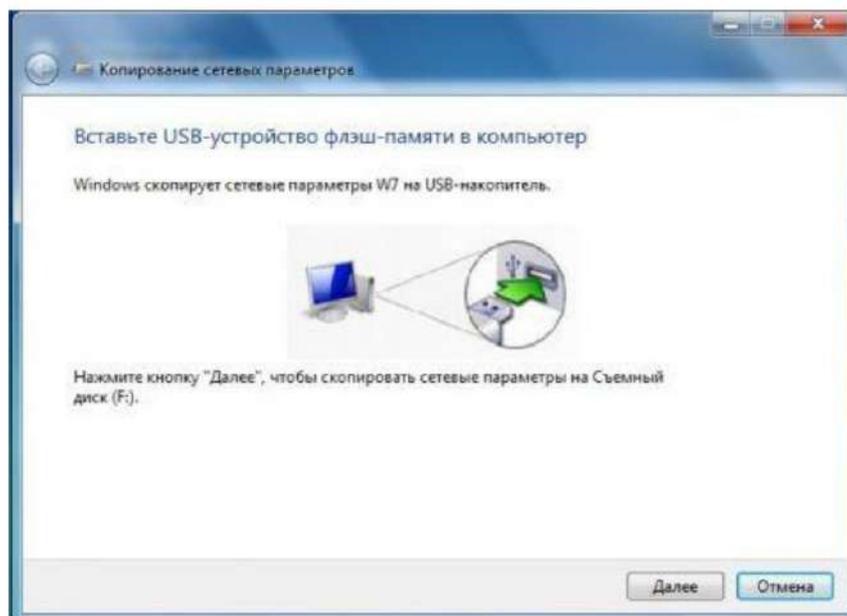
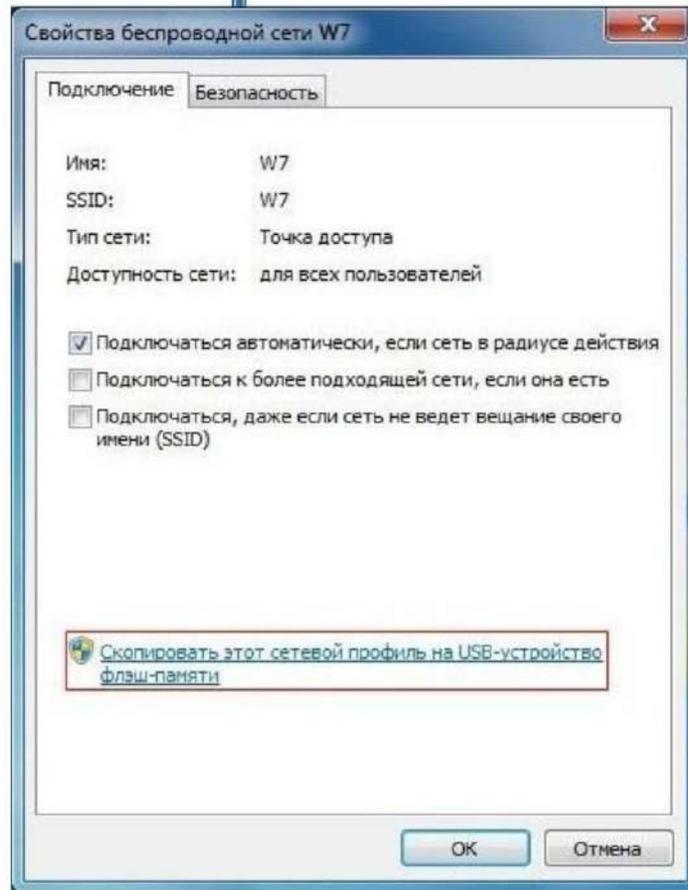
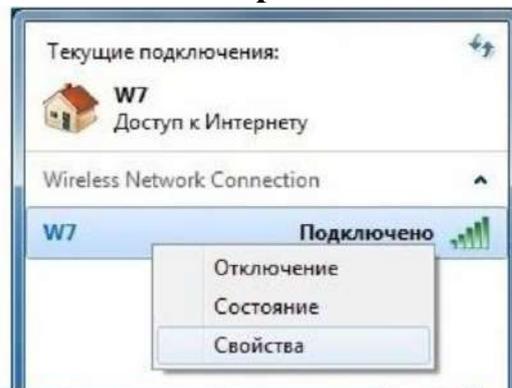
1.2.3 На компьютерах, работающих под управлением более старых операционных систем Windows, а также на беспроводных устройствах, не поддерживающих метод Push Button, необходимо воспользоваться импортом профиля сетевого подключения к беспроводной сети.

1.3 Подключение к точке доступа через импорт профиля сетевого подключения

1.3.1 Если USB флеш диск с настройками сетевого профиля не был создан по завершении настройки точки доступа (беспроводного маршрутизатора), то нужно его создать. Для этого необходимо подключить USB флеш диск, в **центре соединений** вызвать **свойства** беспроводной сети и выбрать **Скопировать этот сетевой профиль на USB Устройство флеш-памяти**. Также можно открыть свойства беспроводной сети через **Панель Управления – Центр управления сетями и общим доступом – Управление беспроводными сетями**.

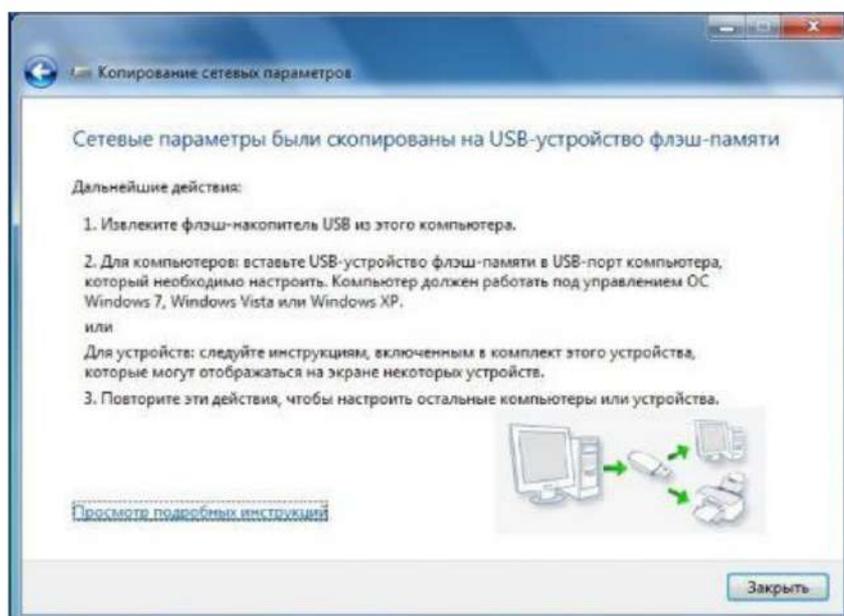
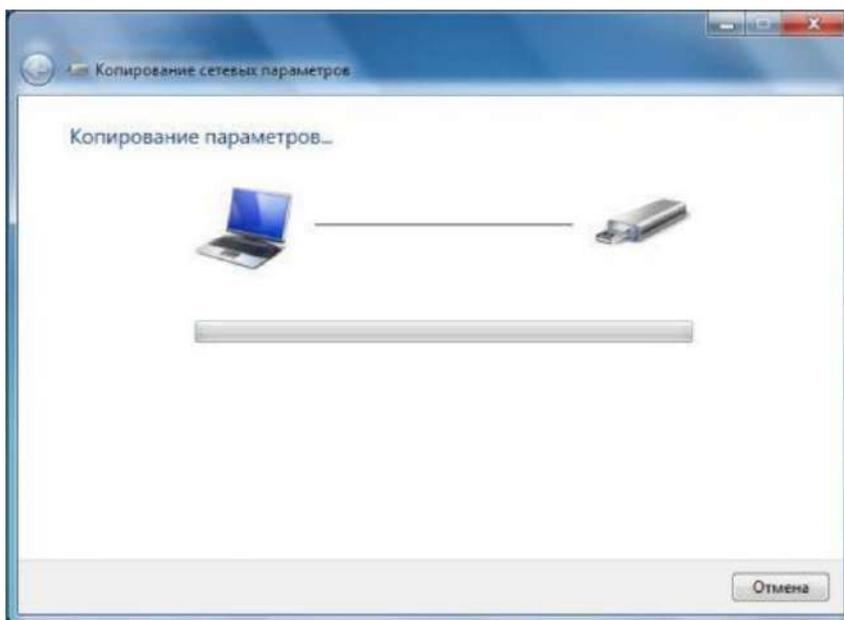


Негосударственное частное образовательное учреждение высшего образования





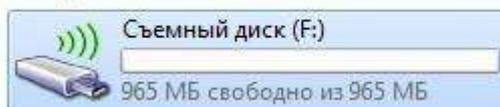
Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»



1.3.2 По завершении работы мастера USB флеш диск может использоваться для подключения различных беспроводных устройств, а также компьютеров, оснащенных беспроводным адаптером и работающих под управлением Windows XP/Vista/Windows 7.

1.3.3 Для устройств с беспроводным адаптером, таких как фоторамки, принтеры, игровые консоли, необходимо подключить к ним USB флеш диск с сетевым профилем и согласиться с импортом настроек. По окончании настройки устройство автоматически подключится к беспроводной сети. Аналогичные действия и для операционных систем Windows: подключить USB флеш диск с сетевым профилем,

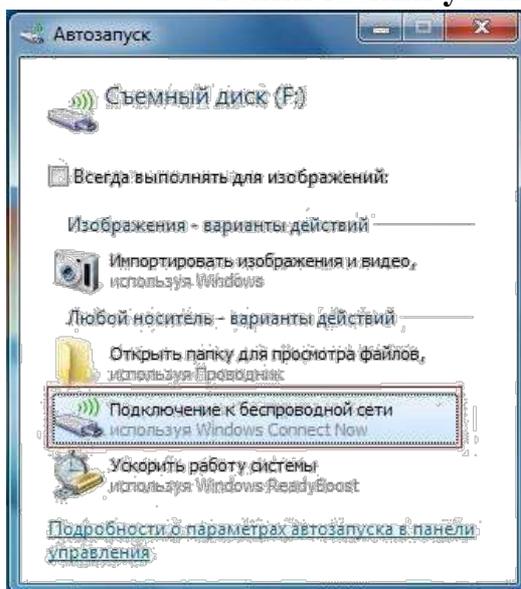
▲ Устройства со съемными носителями (1)



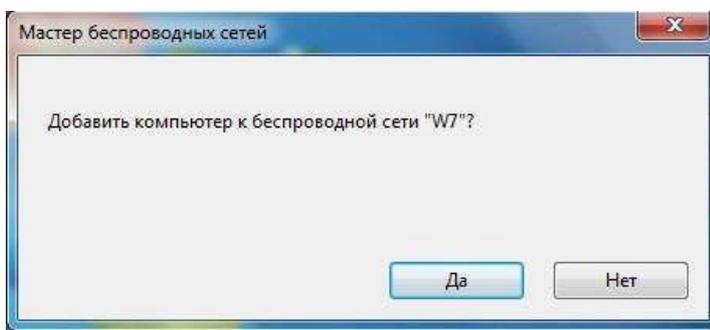
1.3.4 в окне Автозапуска выбрать Подключение к беспроводной сети используя Windows Connect Now,



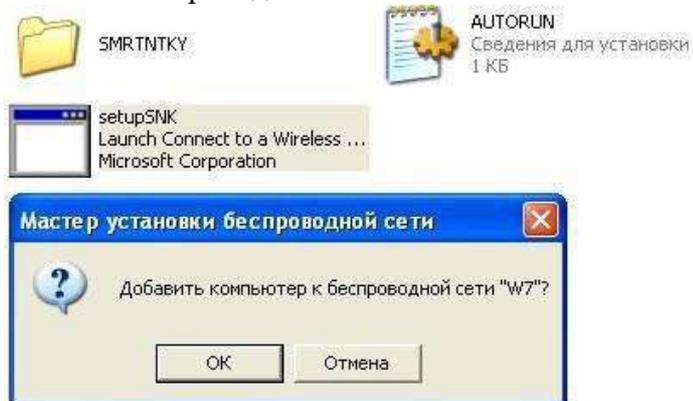
Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»



1.3.5 согласиться с импортом настроек.



1.3.6 Если на компьютере с Windows отключена функция автозапуска, необходимо открыть USB флеш диск, запустить файл **SetupSNK.Exe** и согласиться с импортом настроек для подключения к беспроводной сети.



1.3.7 Windows Connect Now значительно упрощает настройку беспроводной сети и теперь вам не потребуется запоминать учетные данные сети и тратить много времени для подключения новых ПК.

2. Содержание отчета:

- 7.1 Наименование и цель работы
- 7.2 Выполненное задание
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод о проделанной работе



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Контрольные вопросы

1. Какие процедуры и функции уровней эталонной сетевой модели ISO/OSI реализованы в маршрутизаторе?
2. Укажите, в чем заключается основное отличие принципов работы коммутаторов от маршрутизаторов? Приведите пример, основываясь на схеме проекта.
3. Укажите длину IP-адреса протокола IPv4 и протокола IPv6?
4. Объясните назначение следующих адресов:
 - 169.254.0.0/16;
 - 127.0.0.0/8;
 - 255.255.255.255;
 - 224.0.0.0;
 - 240.0.0.0;
5. Объясните необходимость наличия маски подсети, при указании IP-адреса устройства?
6. Перечислите служебные адреса подсети IPv4? Опишите границы частных диапазонов адресов IPv4?
7. Дайте определение понятиям: классовая маршрутизация, VLSM, CIDR?
8. Вычислите адреса и маски сетей по имеющимся IP-адресам:
 - 10.10.12.7 255.255.255.240;
 - 192.168.1.4 255.255.128.1;
 - 172.31.100.15 255.252.0.0;
9. Приведите недостатки и достоинства метода статической маршрутизации в сети ЛВС?
10. Какие среды конфигурации беспроводных устройств вы знаете?
11. Какая технология беспроводного подключения в настоящее время считается наиболее простой и безопасной?



Практическое занятие 8 Администрирование беспроводной сети

Основной целью администрирования является приведение сети в соответствие с целями и задачами, для которых она предназначена. Администрирование заключается в контроле за работой сетевого оборудования и управление функционированием сети в целом. Администрирование выполняет администратор сети – специалист, отвечающий за нормальное функционирование и использование ресурсов сети. Если более детально, то администрирование информационных систем включает следующие цели:

- Установка и настройка сети. Поддержка её дальнейшей работоспособности.
- Мониторинг. Планирование системы.
- Установка и конфигурация аппаратных устройств.
- Установка программного обеспечения.
- Архивирование (резервное копирование) информации.
- Создание и управление пользователями.
- Установка и контроль защиты.

Вот выписка должностных обязанностей администратора сети:

- Устанавливает на серверы и рабочие станции сетевое программное обеспечение.
- Конфигурирует систему на сервере.
- Обеспечивает интегрирование программного обеспечения на файл-серверах, серверах систем управления базами данных и на рабочих станциях.
- Поддерживает рабочее состояние программного обеспечения сервера.
- Регистрирует пользователей, назначает идентификаторы и пароли.
- Обучает пользователей работе в сети, ведению архивов; отвечает на вопросы пользователей, связанные с работой в сети; составляет инструкции по работе с сетевым программным обеспечением и доводит их до сведения пользователей.
- Контролирует использование сетевых ресурсов.
- Организует доступ к локальной и глобальной сетям.
- Устанавливает ограничения для пользователей по:
 - использованию рабочей станции или сервера;
 - времени;
 - степени использования ресурсов.
- Обеспечивает своевременное копирование и резервирование данных.
- Обращается к техническому персоналу при выявлении неисправностей сетевого оборудования.
- Участвует в восстановлении работоспособности системы при сбоях и выходе из строя сетевого оборудования.
- Выявляет ошибки пользователей и сетевого программного обеспечения и восстанавливает работоспособность системы.
- Проводит мониторинг сети, разрабатывает предложения по развитию инфраструктуры сети.
- Обеспечивает:
 - сетевую безопасность (защиту от несанкционированного доступа к информации, просмотра или изменения системных файлов и данных);
 - безопасность межсетевого взаимодействия.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- Готовит предложения по модернизации и приобретению сетевого оборудования.
- Осуществляет контроль за монтажом оборудования специалистами сторонних организаций.
- Сообщает своему непосредственному руководителю о случаях злоупотребления сетью и принятых мерах.
- Ведет журнал системной информации, иную техническую документацию.

Любая беспроводная сеть состоит как минимум из двух базовых компонентов – точки беспроводного доступа и клиента беспроводной сети (режим ad-hoc, при котором клиенты беспроводной сети общаются друг с другом напрямую без участия точки доступа, мы рассматривать не будем). Стандартами беспроводных сетей 802.11a/b/g предусматривается несколько механизмов обеспечения безопасности, к которым относятся различные механизмы аутентификации пользователей и реализация шифрования при передаче данных.

Протокол WEP

Все современные беспроводные устройства (точки доступа, беспроводные адаптеры и маршрутизаторы) поддерживают протокол безопасности WEP (Wired Equivalent Privacy), который был изначально заложен в спецификацию беспроводных сетей IEEE 802.11. Данный протокол является своего рода аналогом проводной безопасности (во всяком случае, расшифровывается он именно так), однако реально никакого эквивалентного проводным сетям уровня безопасности он, конечно же, не предоставляет.

Протокол WEP позволяет шифровать поток передаваемых данных на основе алгоритма RC 4 с ключом размером 64 или 128 бит.

Данные ключи имеют так называемую статическую составляющую длиной от 40 до 104 бит и дополнительную динамическую составляющую размером 24 бита, называемую вектором инициализации (Initialization Vector, IV).

На простейшем уровне процедура WEP-шифрования выглядит следующим образом: первоначально передаваемые в пакете данные проверяются на целостность (алгоритм CRC-32), после чего контрольная сумма (integrity check value, ICV) добавляется в служебное поле заголовка пакета. Далее генерируется 24-битный вектор инициализации, (IV) и к нему добавляется статический (40-или 104-битный) секретный ключ. Полученный таким образом 64-или 128-битный ключ и является исходным ключом для генерации псевдослучайного числа, используемого для шифрования данных. Далее данные смешиваются (шифруются) с помощью логической операции XOR с псевдослучайной ключевой последовательностью, а вектор инициализации добавляется в служебное поле кадра. Вот, собственно, и всё.

Протокол безопасности WEP предусматривает два способа аутентификации пользователей: Open System (открытая) и Shared Key (общая). При использовании открытой аутентификации никакой аутентификации, собственно, и не существует, то есть любой пользователь может получить доступ в беспроводную сеть. Однако даже при использовании открытой системы допускается использование WEP-шифрования данных.

Протокол WPA

Как будет показано чуть позже, протокол WEP имеет ряд серьёзных недостатков и не является для взломщиков труднопреодолимым препятствием. Поэтому в 2003 году был представлен следующий стандарт безопасности — WPA (Wi-Fi Protected Access). Главной особенностью этого стандарта является технология динамической генерации ключей шифрования данных, построенная на базе протокола TKIP (Temporal Key Integrity Protocol), представляющего собой дальнейшее развитие алгоритма шифрования RC 4. По протоколу TKIP сетевые устройства работают с 48-битовым вектором инициализации (в отличие от 24-битового вектора WEP) и реализуют правила изменения последовательности его битов, что исключает повторное использование ключей. В



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

протоколе TKIP предусмотрена генерация нового 128-битного ключа для каждого передаваемого пакета. Кроме того, контрольные криптографические суммы в WPA рассчитываются по новому методу под названием MIC (Message Integrity Code). В каждый кадр здесь помещается специальный восьмибайтный код целостности сообщения, проверка которого позволяет отражать атаки с применением подложных пакетов. В итоге получается, что каждый передаваемый по сети пакет данных имеет собственный уникальный ключ, а каждое устройство беспроводной сети наделяется динамически изменяемым ключом.

Кроме того, протокол WPA поддерживает шифрование по стандарту AES (Advanced Encryption Standard), то есть по усовершенствованному стандарту шифрования, который отличается более стойким криптоалгоритмом, чем это реализовано в протоколах WEP и TKIP.

При развёртывании беспроводных сетей в домашних условиях или небольших офисах обычно используется вариант протокола безопасности WPA на основе общих ключей – WPA-PSK (Pre Shared Key). В дальнейшем мы будем рассматривать только вариант WPA-PSK, не касаясь вариантов протокола WPA, ориентированных на корпоративные сети, где авторизация пользователей проводится на отдельном RADIUS-сервере.

При использовании WPA-PSK в настройках точки доступа и профилях беспроводного соединения клиентов указывается пароль длиной от 8 до 63 символов.

Фильтрация MAC-адресов

Фильтрация MAC-адресов, которая поддерживается всеми современными точками доступа и беспроводными маршрутизаторами, хотя и не является составной частью стандарта 802.11, тем не менее, как считается, позволяет повысить уровень безопасности беспроводной сети. Для реализации данной функции в настройках точки доступа создаётся таблица MAC-адресов беспроводных адаптеров клиентов, авторизованных для работы в данной сети.

Режим скрытого идентификатора сети SSID

Ещё одна мера предосторожности, которую часто используют в беспроводных сетях – это режим скрытого идентификатора сети. Каждой беспроводной сети назначается свой уникальный идентификатор (SSID), который представляет собой название сети. Когда пользователь пытается войти в сеть, то драйвер беспроводного адаптера прежде всего сканирует эфир на наличие в ней беспроводных сетей. При использовании режима скрытого идентификатора (как правило, этот режим называется Hide SSID) сеть не отображается в списке доступных, и подключиться к ней можно только в том случае, если, во-первых, точно известен её SSID, и, во-вторых, заранее создан профиль подключения к этой сети.

Взлом беспроводной сети с протоколом WEP

Чтобы у читателя не сложилось впечатления, что перечисленных средств защиты вполне достаточно, дабы не опасаться непрошенных гостей, поспешим его разочаровать. И начнём мы с инструкции по взлому беспроводных сетей стандарта 802.11 b / g на базе протокола безопасности WEP.

Собственно, утилит, специально разработанных для взлома таких сетей и доступных в Интернете, предостаточно. Правда, есть одно «но». Почти все они «заточены» под Linux-системы. Собственно, с точки зрения продвинутого пользователя – это не только не помеха, но и наоборот. А вот обычными пользователями операционная система Linux используется редко, поэтому мы решили ограничиться рассмотрением утилит, поддерживаемых системой Windows XP.

Итак, для взлома сети нам, кроме ноутбука с беспроводным адаптером, потребуется утилита aircrack 2.4, которую можно найти в свободном доступе в Интернете.

Данная утилита поставляется сразу в двух вариантах: под Linux и под Windows, поэтому нас будут интересовать только те файлы, которые размещены в директории aircrack-2.4\win 32.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

В этой директории имеется три небольших утилиты (исполняемых файлов): `airodump.exe`, `aircrack.exe` и `airdecap.exe`.

Первая утилита предназначена для перехвата сетевых пакетов, вторая – для их анализа и получения пароля доступа и третья – для расшифровки перехваченных сетевых файлов. Конечно же, не всё так просто, как может показаться. Дело в том, что все подобные программы «заточены» под конкретные модели чипов, на базе которых построены сетевые адаптеры. То есть не факт, что выбранный произвольно беспроводной адаптер окажется совместим с программой `aircrack-2.4`. Более того, даже при использовании совместимого адаптера (список совместимых адаптеров, а точнее – чипов беспроводных адаптеров, можно найти в документации к программе) придётся повозиться с драйверами, заменив стандартный драйвер от производителя сетевого адаптера на специализированный драйвер под конкретный чип. К примеру, в ходе тестирования мы выяснили, что стандартный беспроводной адаптер Intel PRO Wireless 2200 BG, который является составной частью многих ноутбуков на базе технологии Intel Centrino, просто не совместим с данной программой при использовании ОС Windows XP (правда, он поддерживается при использовании Linux-версии программы). В итоге мы остановили свой выбор на беспроводном PCMCIA-адаптере Gigabyte GN-WMAG на базе чипа Atheros. При этом сам беспроводной адаптер устанавливался как Atheros Wireless Network Adapter с драйвером 3.0.1.12.

Сама процедура взлома беспроводной сети достаточно проста. Начинаем с запуска утилиты `airodump.exe`, которая представляет собой сетевой сниффер для перехвата пакетов. При запуске программы (рис. 1) откроется диалоговое окно, в котором потребуется указать беспроводной сетевой адаптер (Network interface index number), тип чипа сетевого адаптера (Network interface type (o/a)), номер канала беспроводной связи (Channel (s): 1 to 14, 0= all) (если номер канал неизвестен, то можно сканировать все каналы). Также задаётся имя выходного файла, в котором хранятся перехваченные пакеты (Output filename prefix) и указывается, требуется ли захватывать все пакеты целиком (cap-файлы) или же только часть пакетов с векторами инициализации (ivs-файлы) (Only write WEP IVs (y/n)). При использовании WEP-шифрования для подбора секретного ключа вполне достаточно сформировать только ivs-файл. По умолчанию ivs-или с ар-файлы создаются в той же директории, что и сама программа `airodump`.

После настройки всех опций утилиты `airodump` откроется информационное окно, в котором отображается информация об обнаруженных точках беспроводного доступа, информация о клиентах сети и статистика перехваченных пакетов (рис. 2).

Если точек доступа несколько, статистика будет выдаваться по каждой из них.

Первым делом, запишите MAC-адрес точки доступа, SSID беспроводной сети и MAC-адрес одного из подключённых к ней клиентов (если их несколько). Ну а дальше нужно подождать, пока не будет перехвачено достаточное количество пакетов.

Количество пакетов, которые нужно перехватить для успешного взлома сети, зависит от длины WEP-ключа (64 или 128 бит) ну и, конечно же, от удачи. Если в сети используется 64-битный WEP-ключ, то для успешного взлома вполне достаточно захватить пол миллиона пакетов, а во многих случаях – даже меньше. Время, которое для этого потребуется, зависит от интенсивности трафика между клиентом и точкой доступа, но, как правило, составляет не более нескольких минут. В случае же использования 128-битного ключа для гарантированного взлома потребуется перехватить порядка двух миллионов пакетов. Для останова процесса захвата пакетов (работы утилиты) используется комбинация клавиш Ctrl+C.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

После того, как выходной ivs-файл сформирован, можно приступить к его анализу. В принципе, это можно делать и параллельно вместе с перехватами пакетов, но для простоты мы рассмотрим последовательное выполнение процедур перехвата и анализа.

Для анализа сформированного ivs-файла потребуется утилита `aircrack.exe`, которая запускается из командной строки. В нашем случае мы использовали следующие параметры запуска:

```
aircrack.exe -b 00:13:46:1C:A4:5F -n 64 -i 1 out.ivs
```

В данном случае «-b 00:13:46:1C:A4:5F» – это указание MAC-адреса точки доступа, «-n 64» – указание длины используемого ключа шифрования, «-i 1» – индекс ключа, а «out.ivs» – это файл, который подвергается анализу.

Полный перечень параметров запуска утилиты можно посмотреть, набрав в командной строке команду `aircrack.exe` без параметров.

В принципе, поскольку такая информация, как индекс ключа и длина ключа шифрования, как правило, заранее неизвестна, обычно используется следующий упрощённый вариант запуска команды: `aircrack.exe out.ivs`.

Результат анализа ivs-файла показан на рис. 4. Вряд ли строка **KEY FOUND!** нуждается в комментариях. И обратите внимание, что секретный ключ был вычислен всего за 3 секунды.

Мы проводили множество экспериментов с использованием и 128-битного ключа, и с различными параметрами запуска команды `aircrack.exe`, но во всех случаях время, за которое вычислялся секретный ключ, не превосходило 7 секунд.

Вот так просто и быстро проводится вскрытие беспроводных сетей с WEP-шифрованием, и говорить о «безопасности» сетей в данной случае вообще неуместно. Ну, действительно, можно ли говорить о том, чего на самом деле нет!

В самом начале мы упомянули, что во всех точках доступа имеются ещё такие возможности, как использование режима скрытого идентификатора сети и фильтрации по MAC-адресам, которые призваны повысить безопасность беспроводной сети. Но не будьте оптимистами – это не спасает.

На самом деле, не таким уж и невидимым является идентификатор сети даже при активации этого режима на точке доступа. К примеру, уже упомянутая нами утилита `airodump` всё равно покажет вам SSID сети, который впоследствии можно использовать для создания профиля подключения к сети (причём несанкционированного подключения). Ну а если говорить о такой наивной мере безопасности, как фильтрация по MAC-адресам, то здесь вообще всё очень просто. Существует достаточно много разнообразных утилит и под Linux, и под Windows, которые позволяют подменять MAC-адрес сетевого интерфейса. К примеру, для несанкционированного доступа в сеть мы подменяли MAC-адрес беспроводного адаптера с помощью утилиты `SMAC 1.2` (рис. 5). Естественно, что в качестве нового MAC-адреса используется MAC-адрес авторизованного в сети клиента, который определяется всё той же утилитой `airodump`.

Итак, преодолеть всю систему безопасности беспроводной сети на базе WEP-шифрования не представляет никакого труда. Возможно, многие скажут, что это малоактуально, поскольку WEP-протокол давно умер и его просто не используют. Ведь на смену ему пришёл более стойкий протокол WPA. Однако не будем торопиться с выводами. Отчасти это действительно так, но только отчасти. Дело в том, что в некоторых случаях для увеличения радиуса действия беспроводной сети разворачиваются так называемые распределённые беспроводные сети (WDS) на базе нескольких точек доступа. Но самое интересное заключается в том, что эти самые распределённые сети не поддерживают WPA-протокола, и единственной допустимой мерой безопасности в данном случае



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

является применение WEP-шифрования. Ну а взламываются эти WDS-сети абсолютно так же, как и сети на базе одной точки доступа.

Теперь посмотрим, как обстоят дела с сетями на базе WPA-шифрования.

Взлом беспроводной сети с протоколом WPA

Собственно, сама процедура взлома сетей с протоколом WPA мало чем отличается от уже рассмотренной процедуры взлома сетей с WEP-протоколом.

На первом этапе используется всё тот же сниффер airodump. Однако есть два важных аспекта, которые необходимо учитывать. Во-первых, в качестве выходного файла необходимо использовать именно сар-файл, а не ivs-файл. Для этого в настройке утилиты airodump на последний вопрос «Only write WEP IVs (y/n)» отвечаем «нет».

Во-вторых, в сар-файл необходимо захватить саму процедуру инициализации клиента в сети, то есть придётся посидеть в «засаде» с запущенной программой airodump. Если используется Linux-система, то можно провести атаку, которая заставит произвести процедуру переинициализации клиентов сети, а вот под Windows такая программка не предусмотрена.

После того, как в сар-файл захвачена процедура инициализации клиента сети, можно остановить программу airodump и приступить к процессу расшифровки. Накапливать перехваченные пакеты в данном случае нет необходимости, поскольку для вычисления секретного ключа используется только пакеты, передаваемые между точкой доступа и клиентом в ходе инициализации.

Для анализа полученной информации используется все та же утилита aircrack, но с несколько иными параметрами запуска. Кроме того, в директорию с программой aircrack придётся установить ещё один важный элемент – словарь. Такие специализированные словари можно найти в Интернете, например, по ссылке <http://ftp.se.kde.org/pub/security/tools/net/Openwall/wordlists/>.

После этого запускаем из командной строки программу aircrack (рис 6), указывая в качестве выходного файла сар-файл (например, out. сар) и название словаря (параметр – w all, где all – название словаря).

Программа перебора ключей из словаря даёт очень интенсивную нагрузку на процессор, так что если для этого используется маломощный ПК, то на эту процедуру потребуется много времени. Если же для этого используется мощный многопроцессорный сервер или ПК на базе двухъядерного процессора, то в качестве опции можно указать количество используемых процессоров. К примеру, в нашем случае использовался новейший двухъядерный процессор Intel Pentium Extreme Edition Processor 955 с поддержкой технологии Hyper-Threading (четыре логических ядра процессора), поэтому в параметрах запуска программы мы использовали опцию «-p 4», что позволило утилизировать все четыре логических ядра процессора, причём каждое ядро утилизируется на 100%. В результате после почти полутора часов работы программы секретный ключ был найден! (рис. 7.)

Это, конечно, не несколько секунд, как в случае с WEP-шифрованием, но тоже неплохой результат, который прекрасно демонстрирует, что и WPA-PSK защита не является абсолютно надёжной. Причём результат взлома секретного ключа никак не связан с тем, какой алгоритм шифрования (TKIP или AES) используется в сети.

Настройка соединения (провайдер dhcp):

Шаг 1. Необходимо на вкладку «Дополнительные настройки»-> «WAN»:

Нужно установить тип подключения WAN: «Динамический IP».



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Шаг 2. На этой же вкладке, можно сразу задать адреса DNS:

Если мы не знаем - необходимо оставить «автоматическое» подключение к серверу DNS (как на рисунке).

Шаг 3 (опционально). Если провайдер сети Интернет - осуществляет привязку к MAC-адресу, роутер ASUS RT N12 C1 (как и более «старый») - позволит вам «подменить» его. Снизу вкладки, находится поле MAC-адреса:

Сюда, заносится требуемое вам значение адреса (последние 5 байт, формат HEX). А посмотреть значение аппаратного адреса карты ПК, можно на «Состоянии» соединения («Поддержка» -> «Подробности»).

По завершении этих настроек нажимаем: «Применить». Настройка роутера Асус RT N12 на соединение «DHCP» - завершена. Оно появится сразу же (через 10-15 секунд после нажатия на «Применить»).

Настройка соединения (провайдер рррое):

Шаг 1. На вкладке «Дополнительные настройки» -> «WAN», установить «тип соединения»: PPPoE.

Шаг 2. Подразумевается, что происходит настройка соединения для «динамически» выделяемого IP-адреса. Поэтому, галочки «получить IP WAN автоматически», и «подключиться к DNS-серверу автоматически» - оставляем, как есть. На этом этапе, нужно только заполнить поля имени пользователя и пароля (см. договор).

Шаг 3 (опционально). Мы можем аналогично настройке для случая «DHCP», вписать «клонировемый» адрес «MAC» (в последнее поле на вкладке).

По завершении этих действий, нужно нажать «Применить». Соединение - будет создано. Мы здесь рассмотрели, как настроить роутер RT N12 на соединение «DHCP» и «PPPoE»- типа (с «динамически» выделяемым IP-адресом). Предусмотрены и другие «типы» соединений, в том числе, если роутер - «клиент VPN», то настраивать нужно L2TP или PPTP (выбирается в зависимости от условий провайдера). А если соединение - создано, и подключено, осталось только «запомнить» настройки.

Запоминание настроек: Для выполнения этого действия, заходим на вкладку «Администрирование» -> «Восстановить... загрузить настройки»:

Именно кнопка «Сохранить», позволяет «запомнить» все сделанные изменения в постоянной памяти роутера. Тогда как, нажатие «Восстановить» - сбросит роутер Асус RT N12 к значениям «по умолчанию», причем - немедленно.

Если мы будем смотреть Интернет-телевидение через компьютеры нашей сети, то в самом роутере, нужно выполнить пару настроек. Установить две галочки (вкладка «Дополнительные настройки» -> «ЛВС» -> «Маршрут»).

Нажать «Применить». Теперь, Multicast-пакеты (многоадресные пакеты), будут «пропускаться» в локальную сеть. Пакеты Multicast, используются в цифровом телевидении.

Дополнительно, можно искусственно ограничить максимальное значение Multicast-трафика в беспроводной сети Wi-Fi. Необходимо зайти на вкладку «Дополнительные настройки» -> «Локальная сеть» -> «Профессионально», установить требуемое значение

Также, ASUS строго рекомендует сменить админский пароль. Это можно сделать в пункте «Администрирование». Там же можно обновить прошивку роутера RT-N12 C1 или экспортировать настройки устройства или восстановить заводские.

Мы также можем выполнить обновление микропрограммы с сайта ASUS в случае, когда предыдущая программная оболочка становится неактуальной.

При необходимости в программе возможно восстановление заводских настроек, функция сохранения и отправления настроек:



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

1. Контрольные вопросы

2. Укажите отличительные особенности в принципах работы концентратора и коммутатора? Приведите пример, основываясь на схеме проекта.
3. Каким из указанных в проекте устройств необходимо наличие физических адресов (MAC)?
4. Перечислите режимы коммутации?
5. Приведите разновидности коммутаторов?
6. Объясните, в чем заключается преимущество агрегирования коммутаторов?
7. Что представляет собой логическое объединение коммутаторов в стек?
8. Укажите методы физического подключения для управления современными коммутаторами? Перечислите основные сетевые протоколы управления активным оборудованием компьютерных сетей передачи данных?
9. Выделите отличительные особенности асимметричной и симметричной коммутации?
10. Вычислите пропускную способность внутренней шины коммутации, если коммутатор работает в неблокирующем режиме и имеет 8 FastEthernet-портов?



Лабораторная работа 8 Расширенная диагностика беспроводной WIFI сети

Медленная скорость загрузки, отсутствие доступа к Сети в определенных уголках квартиры — в домашней сети всегда что-то может пойти не так. Потенциальных причин проблем с Wi-Fi множество: как конфигурация, так и расположение роутера.

Кроме того, у маршрутизатора могут быть внутренние основания нарушать соединение с интернет-провайдером или перегружать только те страницы, информацию с которых вы бы хотели скачать.

Для каждой такой проблемы необходимо систематично выявить свой подход к решению, начав с компьютера и закончив посещенной веб-страницей. Воспользуйтесь для этого веб-интерфейсом своего роутера, командной строкой Windows и рекомендуемыми утилитами.

Кроме того, понадобится установить последний драйвер для Wi-Fi-адаптера вашего компьютера. Чтобы узнать разработчика и модель, нажмите комбинацию клавиш «Win+Pause», вызовите Диспетчер устройств и откройте параметр «Сетевые адаптеры».

Проверяем компьютер

WiFi_1 Если совсем не получается установить Wi-Fi-соединение, в первую очередь проверьте, не отключили ли вы случайно Wi-Fi-адаптер компьютера. Для этого вызовите «Параметры» и перейдите к пункту «Сеть и Интернет | Wi-Fi | Настройка параметров адаптера».

Устанавливаем соединение

WiFi_2 Если «Беспроводная сеть» отмечена красным крестиком, необходимо задействовать ее либо переключателем на устройстве, либо функциональной клавишей или же отсоединить и снова подключить USB-адаптер. В случае, когда сеть обозначена серым цветом, правой кнопкой мыши вызовите контекстное меню и выберите команду «Подключение», чтобы включить ее со стороны программного обеспечения

Тестируем роутер

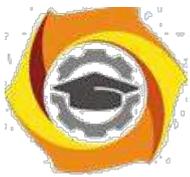
WiFi_3 Чтобы исключить проблемы, связанные с компьютером, убедитесь, что на вашем роутере активен Wi-Fi. Для этого проверьте, что горит соответствующая лампочка. Если нет, активируйте беспроводное соединение подходящей клавишей на маршрутизаторе или с компьютера через веб-интерфейс. Для этого подключите роутер к ПК сетевым кабелем.

Проверяем Wi-Fi-подключение

WiFi_4 Беспроводное соединение установлено, но загрузка тянется мучительно долго? В этом случае следует проверить качество связи между компьютером и маршрутизатором. Для этого с помощью комбинацией клавиш «Win+R» откройте окно «Выполнить», введите команду «cmd» и нажмите клавишу «Enter». В командной строке командой «ipconfig» запустите отображение всех сетевых интерфейсов.

Посылаем ping-запрос

WiFi_5 Зачастую список интерфейсов бывает очень длинным. Найти нужный порт, впрочем, легко: рядом с записью «Основной шлюз» указан IP-адрес — он принадлежит



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

вашему маршрутизатору. В роутерах FritzBox, к примеру, это «192.168.178.1». Введите команду «ping -t 192.168.178.1». После этого Windows начнет посылать запрос к роутеру каждую секунду.

Анализируем время ответа

WiFi_6 Появление сообщения «Ответ от 192.168.178.1; число байтов=32...» означает, что связь между компьютером и роутером налажена. Прервите отправку команды ping комбинацией клавиш «Ctrl+C» и узнайте точное время ответа. Если значение превышает 20 миллисекунд, соединение необходимо оптимизировать.

Размещаем устройства правильно

WiFi_7 Долгие ping-ответы и разрывы соединения являются индикаторами того что, хоть сигнал и проходит, но он очень слабый. Часто для его усиления достаточно поставить маршрутизатор повыше. Постарайтесь не подключать USB-адаптер в порт на задней стороне корпуса. Если разместить его на письменном столе с помощью удлиняющего кабеля, это дополнительно улучшит сигнал.

Меняем радиоканал

WiFi_8 Утилита для анализа под названием InSSIDer (metageek.com) покажет, насколько хорошо и через какой канал функционируют Wi-Fi в вашем окружении. Видите больше 10 беспроводных сетей с той же силой, что и ваша? Подберите один из каналов 1, 5, 9 или 13, в чьем частотном диапазоне меньше всего сетей. Задайте этот канал в маршрутизаторе.

Диагностика Wi-Fi сетей выполняется специализированными утилитами. Среди них есть как платные, так и бесплатные версии, наиболее популярные представлены ниже:

- [WirelessNetView](#);
- [Free Wi-Fi Scanner](#);
- [InSSIDer](#).

Остановимся на утилите **inSSIDer**, так как это мощный инструмент для диагностики беспроводных сетей. Данная программа поможет вам измерить уровень сигнала и оценить производительность вашего Wi-Fi оборудования в различных местах. Вы наглядно сможете посмотреть и проверить, как стены, лестницы, двери, да и в целом планировка и материалы вашего помещения влияют на зону покрытия обслуживаемой беспроводной сети. При этом версия Home — совершенно бесплатна.

В настоящее время в любом современном городе каждый дом или офис буквально кишит обилием Wi-Fi сетей. При этом, когда несколько беспроводных сетей перекрывают канал на котором работают (т.е. несколько AP в здании раздают Wi-Fi сеть по одному каналу) возникает ситуация замедления работы всех сетей Wi-Fi на этом канале.

Сканер **inSSIDer** поможет обнаружить наилучший канал для вашего Wi-Fi.

Особенности inSSIDer

К основным особенностям данного ПО относят:

- **inSSIDer** использует ваше текущее программное обеспечение беспроводной карты и Wi-Fi подключения
- работает с Microsoft Windows Vista, 7 и 8.1 (32 и 64 бит)
- отслеживает силу принимаемых сигналов в dBm с течением времени
- доступна сортировка по MAC-адресу, SSID, номеру канала, RSSI, и времени

Диагностика Wi-Fi сети при помощи inSSIDer

Установка утилиты не вызывает никаких затруднений. Стоит отметить — если вы используете несколько беспроводных адаптеров, то в меню **Сетевое подключение** выберите нужный беспроводной адаптер — при его помощи будет выполняться сканирование. Далее программа автоматически проведет сканирование



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

беспроводных сетей и выведет на экран информацию об эфире. Ниже представлен скриншот рабочего окна inSSIDer:



Рабочее окно программы inSSIDer

Рассмотрим более подробно представленную информацию:

SSID – имя беспроводной сети.

Channel – номер канала, на котором работает беспроводная сеть. Рекомендуется использовать беспроводной канал, на котором работает наименьшее количество других сетей.

RSSI – уровень мощности принимаемого сигнала. Чем выше число RSSI, или чем оно менее отрицательное, тем мощнее сигнал. Старайтесь не делить номер канала (Channel) с точками доступа, которые приближаются к вашей сети по уровню сигнала.

Security – тип безопасности. В некоторых версиях утилиты тип безопасности **WPA2-TKIP** обозначается как **RSNA**, а **WPA2-AES** как **CCMP**.

Max Rate – максимальная скорость работы устройства на физическом уровне (максимальная теоретическая скорость), предоставляемая точкой доступа.

Vendor – производитель точки доступа.

В России разрешены к использованию 13 беспроводных каналов, три из которых являются непересекающимися (это каналы 1, 6 и 11).

Если беспроводной адаптер, установленный на компьютере/ноутбуке/планшетном ПК/смартфоне, предназначен для использования в США, на нем можно будет использовать только каналы с 1 по 11. Поэтому, если установить номер канала 12 или 13 (а также если один из них был выбран алгоритмом автоматического выбора канала), беспроводной клиент не увидит точку доступа. В этом случае необходимо вручную установить номер канала из диапазона с 1 по 11.

Контрольные вопросы

Перечислите преимущества и области использования центральных сетевых контроллеров беспроводных точек доступа?

1. Перечислите ряд распространенных протоколов динамической маршрутизации для локальных и глобальных сетей?
2. Объясните различие терминов «сегментация» и «фрагментация», относительно структуризации данных в эталонной сетевой модели ISO/OSI?
3. Дайте определение понятию «транк» и изобразите подключение, описывающее данный термин?
4. Укажите отличительные черты функционирования сетевых мостов и коммутаторов компьютерных сетей передачи данных?
5. Объясните структуру объединения удаленных сетевых узлов, основанную на принципе микросегментации подключений в компьютерных сетях передачи данных?
6. Опишите основные отличия между способами доступа к разделяемой среде передачи данных по принципу CSMA/CD и CSMA/CA?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

7. Укажите основные реализации компьютерных сетей, использующие методы доступа по принципу CSMA/CD и CSMA/CA?
8. Объясните назначение маркировки MDI/MDI-X на портах Ethernet-концентратора?
9. Чем отличаются симметричные алгоритмы шифрования от асимметричных?

Алгебра и геометрия

Методические указания по практическим занятиям

Практическое занятие № 1

Матрицы и определители

1. Цель занятия:

Выработать умения и навыки по выполнению алгебраических операций над матрицами, нахождению обратной матрицы. Научить студентов находить определители 2-го, 3-го и произвольного порядка; использовать свойства определителей для их вычисления.

2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: **(1)** с. 15 –35, **(2)** с. 39 – 43, 70-76, **(3)** с. 188-191

3. Задание:

Решить примеры: **(2)** №№ 399-405, **(3)** №№ 1211-1234 (выборочно, по рекомендации преподавателя).

4. Порядок выполнения:

- 4.1. Записать формулы вычисления суммы, произведения двух матриц, произведения матрицы на число.
- 4.2. Записать формулу вычисления определителя 2-го порядка.
- 4.3. Записать разложение определителя 3-го порядка по строке или столбцу.
- 4.4. Записать определение определителя произвольного порядка. Перечислить его свойства.
- 4.5. Записать формулу вычисления обратной матрицы.
- 4.6. Решить примеры, см. п.3.

5. Отчет:

- 5.1. Формулы вычисления суммы, произведения двух матриц, произведения матрицы на число.
- 5.2. Формула вычисления определителя 2-го порядка.
- 5.3. Разложение определителя 3-го порядка по строке или столбцу.
- 5.4. Определение определителя произвольного порядка. Его свойства.
- 5.5. Формула вычисления обратной матрицы.
- 5.6. Решенные примеры.

6. Список литературы:

1. Бортаковский А.С., Пантелеев А.В. Линейная алгебра и аналитическая геометрия. Практикум: Учебное пособие. М.: НИЦ ИНФРА-М, 2015
2. Рябушко А. П. Индивидуальные задания по высшей математике. Ч. 1. Линейная и векторная алгебра. Аналитическая геометрия. Дифференциальное исчисление функций одной переменной. Минск: Выш. шк., 2013
3. Д.В. Клетеник. Сборник задач по аналитической геометрии. – М.: Профессия, 2010



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие № 2

Решение систем линейных уравнений по формулам Крамера и матричным методом.

1. Цели занятия:

Научить студентов применять формулы Крамера и матричный метод для решения систем линейных уравнений.

2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: **(1)** с. 35 –40, **(2)** с. 39 – 43, **(3)** с. 194

3.Задание:

Решить примеры: **(2) №№ 225-230, (3) №№ 1236-1241** (выборочно, по рекомендации преподавателя).

4. Порядок выполнения задания:

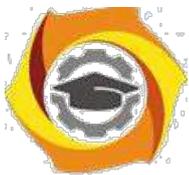
- 4.1. Записать формулы Крамера.
- 4.2. Записать формулу для вычисления обратной матрицы.
- 4.3. Записать общий вид решения системы по матричному методу.
- 4.4. Решить примеры, см. п.3.

5. Отчет:

- 5.1. Формулы Крамера.
- 5.2. Основные формулы матричного метода.
- 5.3. Решенные примеры.

6. Список литературы:

1. Бортаковский А.С., Пантелеев А.В. Линейная алгебра и аналитическая геометрия. Практикум: Учебное пособие. М.: НИЦ ИНФРА-М, 2015
2. Рябушко А. П. Индивидуальные задания по высшей математике. Ч. 1. Линейная и векторная алгебра. Аналитическая геометрия. Дифференциальное исчисление функций одной переменной. Минск: Выш. шк., 2013
3. Д.В. Клетеник. Сборник задач по аналитической геометрии. – М.: Профессия, 2010



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие № 3

Скалярное и векторное произведение

1. Цель занятия:

Выработать умения и навыки по выполнению линейных операций над векторами, вычислению скалярного и векторного произведения. Научить студентов вычислять угол между векторами, площадь треугольника и длину его высоты.

2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: **(1)** с. 65 –80, **(2)** с. 45 – 50, **(3)** с. 116-119, 124, 128

3. Задание:

Решить примеры: **(3)** №№ 748-756, 761-769, 787-794, 795-805, 839-857 (выборочно, по рекомендации преподавателя).

4. Порядок выполнения:

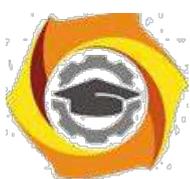
- 4.1. Записать правила выполнения линейных операций над векторами.
- 4.2. Записать формулу для вычисления скалярного произведения.
- 4.3. Записать формулу нахождения угла между векторами.
- 4.4. Записать формулу для вычисления векторного произведения.
- 4.5. Записать формулу вычисления площади треугольника.
- 4.6. Решить примеры, см. п.3.

5. Отчет:

- 5.1. Правила выполнения линейных операций над векторами.
- 5.2. Формула для вычисления скалярного произведения.
- 5.3. Формула нахождения угла между векторами.
- 5.4. Формула для вычисления векторного произведения.
- 5.5. Формула вычисления площади треугольника.
- 5.6. Решенные примеры.

6. Список литературы:

1. Бортакoвский А.С., Пантелеев А.В. Линейная алгебра и аналитическая геометрия. Практикум: Учебное пособие. М.: НИЦ ИНФРА-М, 2015
2. Рябушко А. П. Индивидуальные задания по высшей математике. Ч. 1. Линейная и векторная алгебра. Аналитическая геометрия. Дифференциальное исчисление функций одной переменной. Минск: Выш. шк., 2013
3. Д.В. Клетеник. Сборник задач по аналитической геометрии. – М.: Профессия, 2010



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие № 4

Смешанное произведение

1. Цель занятия:

Выработать умения и навыки по вычислению смешанного произведения. Научить студентов вычислять объем пирамиды и длину ее высоты, проверять компланарность векторов.

2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: **(1)** с. 81 –90, **(2)** с. 49 – 51, **(3)** с. 131

3. Задание:

Решить примеры: **(3)** №№ 865-878 (выборочно, по рекомендации преподавателя).

4. Порядок выполнения:

- 4.1. Записать формулу для вычисления смешанного произведения.
- 4.2. Записать формулу нахождения объема пирамиды.
- 4.3. Записать условие компланарности векторов.
- 4.4. Решить примеры, см. п.3.

5. Отчет:

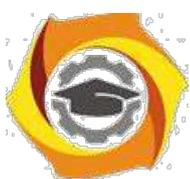
- 5.1. Формула для вычисления смешанного произведения.
- 5.2. Формула нахождения объема пирамиды.
- 5.3. Условие компланарности векторов.
- 5.4. Решенные примеры.

6. Список литературы:

1. Бортакровский А.С., Пантелеев А.В. Линейная алгебра и аналитическая геометрия. Практикум: Учебное пособие. М.: НИЦ ИНФРА-М, 2015
2. Рябушко А. П. Индивидуальные задания по высшей математике. Ч. 1. Линейная и векторная алгебра. Аналитическая геометрия. Дифференциальное исчисление функций одной переменной. Минск: Выш. шк., 2013
3. Д.В. Клетеник. Сборник задач по аналитической геометрии. – М.: Профессия, 2010

Практическое занятие № 5

Контрольная работа №1 по теме «Матричная и векторная алгебра»



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие № 6

Решение задач о прямой на плоскости

1. Цель занятия:

Выработать умения и навыки по нахождению уравнений параллельных и перпендикулярных прямых, расстояния от точки до прямой. Научить студентов определять угловой коэффициент прямой и находить угол между прямыми.

2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: **(1)** с. 47 – 50, **(2)** с. 15 – 19; **(3)** с. 35 - 36.

3. Задание:

Решить примеры: **(3)** №№ 210 – 229, 261 – 266 (выборочно, по рекомендации преподавателя).

4. Порядок выполнения:

- 4.1. Записать общее уравнение прямой.
- 4.2. Записать уравнение прямой с угловым коэффициентом.
- 4.3. Записать условия параллельности и перпендикулярности прямых.
- 4.4. Записать формулу для вычисления угла между прямыми.
- 4.5. Записать формулу нахождения расстояния от точки до прямой.
- 4.6. Решить примеры, см. п.3.

5. Отчет:

- 5.1. Общее уравнение прямой.
- 5.2. Уравнение прямой с угловым коэффициентом.
- 5.3. Условия параллельности и перпендикулярности прямых.
- 5.4. Формула для вычисления угла между прямыми.
- 5.5. Формула нахождения расстояния от точки до прямой.
- 5.6. Решенные примеры.

6. Список литературы:

1. Бортакoвский А.С., Пантелеев А.В. Линейная алгебра и аналитическая геометрия. Практикум: Учебное пособие. М.: НИЦ ИНФРА-М, 2015
2. Рябушко А. П. Индивидуальные задания по высшей математике. Ч. 1. Линейная и векторная алгебра. Аналитическая геометрия. Дифференциальное исчисление функций одной переменной. Минск: Выш. шк., 2013
3. Д.В. Клетеник. Сборник задач по аналитической геометрии. – М.: Профессия, 2010



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие № 7

Решение различных задач, связанных с геометрическими фигурами на плоскости.

1. Цель занятия:

Выработать умения и навыки по нахождению уравнений сторон и диагоналей различных геометрических фигур (треугольников, прямоугольников, квадратов, ромбов и др.). Научить студентов находить уравнения медиан, высот и биссектрис треугольника, вычислять его площадь и координаты вершин.

2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: **(1)** с. 47 – 50, **(2)** с. 20 – 23; **(3)** с. 35 - 36.

3. Задание:

Решить примеры: **(3)** №№ 231 – 245, 267 – 278 (выборочно, по рекомендации преподавателя).

4. Порядок выполнения:

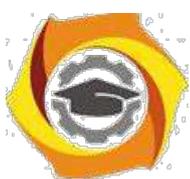
- 4.1. Записать уравнение прямой, проходящей через две точки.
- 4.2. Записать уравнение прямой с угловым коэффициентом, проходящей через заданную точку.
- 4.3. Записать формулы для нахождения координат середины отрезка.
- 4.4. Записать формулу для вычисления площади треугольника.
- 4.5. Решить примеры, см. п.3.

5. Отчет:

- 5.1. Уравнение прямой, проходящей через две точки.
- 5.2. Уравнение прямой с угловым коэффициентом, проходящей через заданную точку.
- 5.3. Формулы для нахождения координат середины отрезка.
- 5.4. Формула для вычисления площади треугольника.
- 5.5. Решенные примеры.

6. Список литературы:

1. Бортакровский А.С., Пантелеев А.В. Линейная алгебра и аналитическая геометрия. Практикум: Учебное пособие. М.: НИЦ ИНФРА-М, 2015
2. Рябушко А. П. Индивидуальные задания по высшей математике. Ч. 1. Линейная и векторная алгебра. Аналитическая геометрия. Дифференциальное исчисление функций одной переменной. Минск: Выш. шк., 2013
3. Д.В. Клетеник. Сборник задач по аналитической геометрии. – М.: Профессия, 2010



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие № 8

Плоскость в пространстве

1. Цель занятия:

Выработать умения и навыки по составлению уравнений плоскости. Научить студентов проверять параллельность и перпендикулярность плоскостей, находить расстояние от точки до плоскости.

2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: **(1)** с. 90 –110, **(2)** с. 53 – 56, **(3)** с. 141

3. Задание:

Решить примеры: **(3)** №№ 913-932 (выборочно, по рекомендации преподавателя).

4. Порядок выполнения:

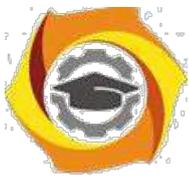
- 4.1. Записать общее уравнение плоскости и координаты вектора нормали.
- 4.2. Записать уравнение пучка плоскостей, проходящих через заданную точку.
- 4.3. Записать уравнение плоскости, проходящей через три точки.
- 4.4. Записать условия параллельности и перпендикулярности плоскостей.
- 4.5. Решить примеры, см. п.3.

5. Отчет:

- 5.1. Общее уравнение плоскости и координаты вектора нормали.
- 5.2. Уравнение пучка плоскостей, проходящих через заданную точку.
- 5.3. Уравнение плоскости, проходящей через три точки.
- 5.4. Условия параллельности и перпендикулярности плоскостей.
- 5.5. Решенные примеры.

6. Список литературы:

1. Бортакровский А.С., Пантелеев А.В. Линейная алгебра и аналитическая геометрия. Практикум: Учебное пособие. М.: НИЦ ИНФРА-М, 2015
2. Рябушко А. П. Индивидуальные задания по высшей математике. Ч. 1. Линейная и векторная алгебра. Аналитическая геометрия. Дифференциальное исчисление функций одной переменной. Минск: Выш. шк., 2013
3. Д.В. Клетеник. Сборник задач по аналитической геометрии. – М.: Профессия, 2010



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие № 9

Прямая в пространстве

1. Цель занятия:

Познакомить студентов с различными способами задания уравнения прямой в пространстве. Выработать умения и навыки по составлению уравнений прямой. Научить студентов решать смешанные задачи на прямую и плоскость.

2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: **(1)** с. 112 –130, **(2)** с. 58 – 62, **(3)** с. 151,154

3. Задание:

Решить примеры: **(3)** №№ 1007-1029, 1051-1074 (выборочно, по рекомендации преподавателя).

4. Порядок выполнения:

- 4.1. Записать каноническое уравнение прямой в пространстве и координаты направляющего вектора.
- 4.2. Записать параметрические уравнения прямой в пространстве.
- 4.3. Записать условия параллельности и перпендикулярности прямых.
- 4.4. Записать условия параллельности и перпендикулярности прямой и плоскости.
- 4.5. Изучить алгоритм решения смешанных задач на прямую и плоскость.
- 4.6. Решить примеры, см. п.3.

5. Отчет:

- 5.1. Каноническое уравнение прямой в пространстве и координаты направляющего вектора.
- 5.2. Параметрические уравнения прямой в пространстве.
- 5.3. Условия параллельности и перпендикулярности прямых.
- 5.4. Условия параллельности и перпендикулярности прямой и плоскости.
- 5.5. Решенные примеры.

6. Список литературы:

1. Бортаковский А.С., Пантелеев А.В. Линейная алгебра и аналитическая геометрия. Практикум: Учебное пособие. М.: НИЦ ИНФРА-М, 2015
2. Рябушко А. П. Индивидуальные задания по высшей математике. Ч. 1. Линейная и векторная алгебра. Аналитическая геометрия. Дифференциальное исчисление функций одной переменной. Минск: Выш. шк., 2013
3. Д.В. Клетеник. Сборник задач по аналитической геометрии. – М.: Профессия, 2010



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие № 10

Кривые второго порядка

1. Цель занятия:

Выработать умения и навыки по нахождению уравнений кривых второго порядка (окружность, эллипс, гипербола и парабола). Научить студентов находить основные параметры кривых второго порядка (вершины, фокусы, директрисы, асимптоты, эксцентриситет, полуоси) по их уравнениям.

2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: **(1)** с. 55 – 65, **(2)** с. 25 – 31; **(3)** с. 64-65, 75-77, 85-86.

3. Задание:

Решить примеры: **(3)** №№ 385-396, 444-449, 515-519, 583-587 (выборочно, по рекомендации преподавателя).

4. Порядок выполнения:

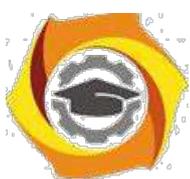
- 4.1. Записать каноническое уравнение окружности.
- 4.2. Записать каноническое уравнение эллипса.
- 4.3. Записать каноническое уравнение гиперболы.
- 4.4. Записать каноническое уравнение параболы.
- 4.5. Записать формулы нахождения основных параметров кривых второго порядка (вершины, фокусы, директрисы, асимптоты, эксцентриситет, полуоси).
- 4.6. Решить примеры, см. п.3.

5. Отчет:

- 5.1. Каноническое уравнение окружности.
- 5.2. Каноническое уравнение эллипса.
- 5.3. Каноническое уравнение гиперболы.
- 5.4. Записать каноническое уравнение параболы.
- 5.5. Формулы вычисления основных параметров кривых второго порядка (вершины, фокусы, директрисы, асимптоты, эксцентриситет, полуоси).
- 5.6. Решенные примеры.

6. Список литературы:

1. Бортакровский А.С., Пантелеев А.В. Линейная алгебра и аналитическая геометрия. Практикум: Учебное пособие. М.: НИЦ ИНФРА-М, 2015
2. Рябушко А. П. Индивидуальные задания по высшей математике. Ч. 1. Линейная и векторная алгебра. Аналитическая геометрия. Дифференциальное исчисление функций одной переменной. Минск: Выш. шк., 2013
3. Д.В. Клетеник. Сборник задач по аналитической геометрии. – М.: Профессия, 2010



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие № 11

Построение кривых второго порядка

1. Цель занятия:

Научить студентов приводить уравнения кривых второго порядка к каноническому виду. Выработать умения и навыки по построению кривых второго порядка.

2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: **(1)** с. 55 – 65, **(2)** с. 32 – 35; **(3)** с. 64-65, 75-77, 85-86.

3. Задание:

Решить примеры: **(2) №№ 187-195** (выборочно, по рекомендации преподавателя).

4. Порядок выполнения:

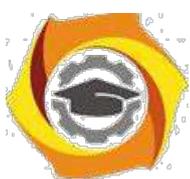
- 4.1. Сгруппировать в заданном уравнении кривой слагаемые по X и по Y .
- 4.2. Выделить полный квадрат по каждой переменной.
- 4.3. Ввести новую систему координат и сделать в уравнении замену переменных.
- 4.4. Определить название кривой и ее параметры (полуоси эллипса, гиперболы, радиус окружно-сти).
- 4.5. Начертить на плоскости старую и новую систему координат, построить кривую.

5. Отчет:

- 5.1. Каноническое уравнение кривой.
- 5.2. Чертеж.

6. Список литературы:

1. Бортаковский А.С., Пантелеев А.В. Линейная алгебра и аналитическая геометрия. Практикум: Учебное пособие. М.: НИЦ ИНФРА-М, 2015
2. Рябушко А. П. Индивидуальные задания по высшей математике. Ч. 1. Линейная и векторная алгебра. Аналитическая геометрия. Дифференциальное исчисление функций одной переменной. Минск: Выш. шк., 2013
3. Д.В. Клетеник. Сборник задач по аналитической геометрии. – М.: Профессия, 2010



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие № 12

Поверхности вращения и цилиндрические поверхности

1. Цель занятия:

Познакомить студентов с понятием поверхности второго порядка. Выработать умения и навыки по составлению уравнений поверхностей вращения. Научить студентов строить цилиндрические поверхности и поверхности вращения.

2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: **(1)** с. 130 –140, **(2)** с. 63 – 67, **(3)** 174-178

3. Задание:

Решить примеры: **(2)** №№ 356-367 (выборочно, по рекомендации преподавателя).

4. Порядок выполнения:

- 4.1. Записать основные типы цилиндрических поверхностей и их уравнения.
- 4.2. Определить вид кривой, образующей цилиндрическую поверхность.
- 4.3. Построить кривую и соответствующую цилиндрическую поверхность.
- 4.4. Записать правило составления уравнений поверхностей вращения.
- 4.5. Решить примеры, см. п.3.

5. Отчет:

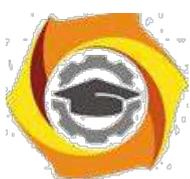
- 5.1. Основные типы цилиндрических поверхностей и их уравнения.
- 5.2. Правило составления уравнений поверхностей вращения.
- 5.3. Решенные примеры.

6. Список литературы:

1. Бортаковский А.С., Пантелеев А.В. Линейная алгебра и аналитическая геометрия. Практикум: Учебное пособие. М.: НИЦ ИНФРА-М, 2015
2. Рябушко А. П. Индивидуальные задания по высшей математике. Ч. 1. Линейная и векторная алгебра. Аналитическая геометрия. Дифференциальное исчисление функций одной переменной. Минск: Выш. шк., 2013
3. Д.В. Клетеник. Сборник задач по аналитической геометрии. – М.: Профессия, 2010

Практическое занятие № 13

Контрольная работа по теме «Аналитическая геометрия на плоскости и в пространстве»



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Заочная форма обучения

Систематическое решение задач — необходимое условие успешного изучения курса дисциплины «Алгебра и геометрия». Решение задач помогает уяснить смысл алгебраических и геометрических понятий, их неразрывную связь с методами передачи и обработки информации, прививает навыки практического применения теоретических знаний.

При решении задач по дисциплине «Алгебра и геометрия» рекомендуется нижеследующее:

1. Изучить теоретический материал по литературным источникам **(1)-(3)**.
2. Ознакомиться с алгоритмом метода решения задачи по методическим указаниям **(4)** по соответствующему разделу дисциплины (см. **табл. 1**).
3. Разобрать модельный пример, приведенный в **(4)**.
4. Решить задачу (см. **табл. 1**).

Табл. 1

ТЕМАТИКА ПРАКТИЧЕСКИХ ЗАНЯТИЙ ДЛЯ СТУДЕНТОВ ЗАОЧНОЙ ФОРМЫ ОБУЧЕНИЯ			
№ п/п	Тема	Стр.	№№ задач
1	Матрицы и определители	7-9	(3) №№ 1211-1234
2	Системы линейных уравнений	5-7, 9-11	(3) №№ 1211-1234
3	Скалярное, векторное и смешанное произведение векторов.	19-28	(3) 748-756, 761-769, 787-794
4	Плоскость и прямая в пространстве.	29-33	(3) №№ 913-932
5	Приведение уравнений поверхностей к каноническому виду	12-15	(3) №№ 1236-1241

**Методические указания
для выполнения лабораторных работ по дисциплине:
«Архитектура информационных систем»
Лабораторная работа №1**

Многофункциональная система обработки и передачи информации REX-400

Цель работы: Изучить и научиться пользоваться системой REX-400.

Содержание работы.

В процессе работы необходимо выполнить следующее:

5. ознакомиться с Руководством оператора по работе системой REX-400;
6. сформировать сообщение;
7. отправить сообщение по электронной почте;
8. отправить сообщение нескольким адресатам одновременно;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

9. включить в состав сообщения один или несколько файлов любого типа;
10. выбрать письма из системных папок для последующего редактирования и отправки по адресам.

Краткие сведения из теории.

Система REX-400 – это распределенная система с многофункционального почтового сервиса, предоставляющая своим пользователям широкие возможности современной электронной почты с максимальным, телексным и телеграфным сервисом, а так-же выход в компьютерные сети. В состав одной станции системы REX-400 входят программно-аппаратные средства, которые образуют единые комплекс, позволяющий своим пользователям осуществлять доступ к системе из разных телематических сред (телефонная сеть общего назначения, сеть АТ-50, сеть X.25 и др.) и пользоваться сервисом системы, используя различные коммуникационное оборудование (модемы, факсимильные аппараты, факс-модемы и др.).

Сеть REX-400 – это совокупность станций REX-400, связанных между собой различными каналами связи.

Компонентами системы REX-400 является:

6. станции электронной почты REX-400 (станция REX-400);
7. агенты пользователей или пользовательские терминалы (АП);

Каналы связи (сеть пакетной коммутации X.25, коммутируемая телефонная сеть, телеграфные каналы, каналы сети ТЕЛКС и др.). Подсистема REX-400.

Система REX-400 включает в себя следующие подсистемы:

- В Информационно – справочная подсистема REX Info;
- В Электронной почты (REX Mail);
- В Факсимильного сервиса (REX Fax);



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Задачей лабораторной работы является изучение работы подсистемы REX Mail / режиме почтовых запросов с использованием специализированной программы Агент пользователя UA.REX [1].

Запуск программы UA.REX.exe.

Через систему DOS:

в найти директорию UA.REX и войти в неё;

в найти программу ua.гех.exe и нажать кл.<ENTER>.

Через WINDOWS XP;

в на рабочем столе;

в найти ua_гех и войти в эту программу.

Создание сообщения.

Ввод темы сообщения.

Для создания сообщения нажать кл. F4 с помощью клавиш навигации выбрать функциональную строку “Тема сообщения” и нажать <ENTER>, после этого появится “окно”, предлагающее ввести тему сообщения ;например; “Моё первое сообщение” и нажать <ENTER>.

Отправка сообщения

Чтобы наглядно продемонстрировать работу системы REX-400, отправим созданное сообщение на телетайп расположенный в лаборатории.

Адрес телетайпа <TGL=123432>.

Чтобы ввести адрес получателя необходимо: в главном меню выбрать функциональную строку “Основные получатели” и нажать <ENTER>; нажать <ESC> и написать имя получателя “TGL=123432”; нажать <ENTER> и 2-а раза <ESC>;

Адрес получателя указан [2].

Ввод текста сообщения.

Чтобы ввести текст сообщения, необходимо выбрать функциональную строку “Текст” и нажать <ENTER>.

Набрать необходимый текст:

- 3) Нажать <ESC> и дать ответ "ДА" на запрос программы о сохранении текста с помощью клавиши <ENTER>.

Отправление созданного сообщения.

Для отправления созданного сообщения ,необходимо перенести созданное сообщение из “Рабочей папки”, в которой оно было создано, в папку



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

“Отправляемые сообщения”. Для этого нажать <F7> и выбрать с помощью клавиш навигации пункт “Отправляемые сообщения”:

3) нажать <ENTER> и дать ответ “ДА” на вопрос программы “Перенести в отправляемые сообщения?” с помощью кп.<ENTER>.

Для отправки сообщения нажать клавиши <Ctrl> + <P6> одновременно. Программа UA.REX отправит сообщение.

Присоединение файлов к сообщению.

Удобство этой функции заключается в том, что можно заранее подготовленный пакет необходимых сообщений отправить за один сеанс СВЯЗИ [3].

Для выполнения этой функции необходимо:

4. с помощью клавиш навигации подвести курсор к строке «СООБЩЕНИЕ» и нажать <ENTER>;
5. выбрать пункт « Читать сообщение из папки « и нажать <Shift>< F7 >;
6. с помощью кл.<TAB> перейти в «окно» <Директории>, выбрать необходимую директорию и нажать <ENTER>. Затем <TAB>, перейти в «окно» <Файлы>, выбрать файл и нажать <ENTER>. После этого, аналогичным образом ,выбрать несколько файлов из разных директорий. В «окне» <Список присоединенных файлов>,появятся выбранные файлы. Число присоединенных файлов не должно превышать 20-ти;
7. с помощью кл.<TAB> перейти в «окно» <Файл>, назвать полученный сборный файл своим именем и нажать <ENTER>;
8. нажать<ESC> и выбрать утвердительный ответ, на вопрос программы. Файл, состоящий из присоединенных файлов находится в графе <Файлы>.

Отчет о работе должен содержать:

6. краткую формулировку цели работы;
7. возможности программы Агент пользователя ЦАВЕХ;
8. распечатки переданных и отправленных сообщений;
9. выводы по выполненной лабораторной работе.

Контрольные вопросы

- 4) Назначение международного стандарта X-400.
- 5) Принцип действия и основные характеристики системы X-400.
- 6) Возможности, предоставляемые пользователю системой REX-400.
- 7) Состав подсистем REX-400.
- 8) Как отправить и принять сообщение?
- 9) Как присоединить файлы к отправляемому сообщению?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ЛАБОРАТОРНАЯ РАБОТА № 2.1

Изучение телеграфного аппарата F-2000

Цель лабораторной работы

Изучение принципов телеграфной связи, основных технических характеристик устройства телетайпа F-2000, овладение практическими навыками по включению, регулировке, вхождению в связь и передаче информации.

Содержание лабораторной работы

Необходимо выполнить следующее;

Изучить назначение, технические характеристики, принцип действия, устройство и конструкцию телетайпа F-2000;

Провести включение и проверку работоспособности аппарата в местном режиме;

Провести передачу и прием заданной информации с помощью телетайпа F-2000;

Произвести оценку качества принятой информации путем сравнения ее с переданной;

Подготовить письменный отчет по лабораторной работе.

Назначение и технические характеристики телетайп F-2000

Автоматизированный рулонный аппарат F-2000 электронного типа предназначен для работы в оконечных пунктах на сетях телеграфной связи с коммутацией каналов общего пользования и абонентского телеграфирования в качестве оконечных пунктов при организации информационной телеграфной связи на крупных АТС.

Аппарат состоит из съемных законченных блоков: реперфоратора (РПФ), трансмиттера (ТРН), запоминающего устройства (ЗУ), клавиатуры (КЛВ), вызывного устройства (ВП), которые будучи подключенными к блоку печатающего устройства (ПУ), образуют максимальный комплект аппарата. Из максимального комплекта при определенных условиях эксплуатации могут изыматься одно, несколько или все блоки, кроме (ПУ).

Так, при работе по некоммутируемым телефонным каналам и физическим линиям изымается блок ВП.

Если нет надобности в приеме сообщений на перфоленту и передачи сообщений с перфоленты, то изымаются блоки (РПФ) и (ТРИ). Минимальный



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

комплект аппарата состоит из блока ПУ. В этом случае он может только принимать телефонные сообщения с распечаткой их на телеграфную рулонную ленту.

Указанный подход к комплектации телетайпа позволяет в каждом конкретном случае выбрать для установки на оконечном пункте только необходимые для работы блоки, чем оптимизируются ряд основных характеристик сетей телефонной связи: стоимость, потребление электроэнергии, надежность и т.д. [4].

Телетайп F-2000 имеет следующие Характеристики:

Используемый телеграфный код.....	МТК-2
Скорость телеграфирования, вод.....	50 и 100
Точность поддержания скорости, %, не более.....	0,3
Производительность, знак-мин.....	400-800
Номинальная длительность, единичные элементы цикла передачи.....	7,5
Цикла приема.....	6,5
Краевые искажения, вносимые передатчиком, %, не более.....	2
Исправляющая способность приемника, %, не менее по краевым искажениям.....	45
по продлениям.....	7
Собственная верность аппарата, не хуже.....	10
Число печатаемых символов.....	52-78
Число знаков в строке.....	69—80
Скорость печати, знак/с.....	25
Ширина бумажного рулона, мм, не более.....	215
Разложение печатаемого знака, точка.....	7 9
Число печатаемых экземпляров при использовании обыкновенной рулонной бумаги.....	1
при использовании специальной рулонной бумаги.....	3
Емкость автоответчика, кодовая комбинация..	20
Емкость электронной памяти, знак.....	1024
Емкость буферной памяти, знак.....	32
Рабочий ток в линейной цепи, на, при работе	
однополосной.....	50
двух полосной.....	20
допустимые отклонения напряжения питающей сети от номинального значения 220В, %.....	—15+10
Потребляемая мощность, ВА	



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

для максимального комплекта	220
для минимального комплекта	140
Масса, кг.	
для максимального комплекта	28,5
для минимального комплекта	18,5

Принцип действия телетайпа F-2000

Режим работы аппарата (передача, прием, местный режим), источники информации (клавиатура, ЗУ, коммутатор автоответа, трансмиттер) и приемники информации (печатающее устройство и (или) реперфоратор) задаются блоком выбора режима на основе информации, введенной оператором с клавиатуры управления или поступившей со входящей линии.

Вызов абонента и установление соединения при передаче информации осуществляются оператором с помощью вызывного устройства - (при работе по коммутируемым тлф каналам набор номера вызываемого абонента осуществляется с помощью клавиатуры (цифровой регистр); при работе по некоммутируемым каналам и физическим цепям номер не набирается). Передаваемая информация в виде кодовых комбинаций поступает от одного из источников информации на вход электронного передатчика. Остальные источники информации в это время блокируются блоком выбора режима. При передаче информации с клавиатуры скорость поступления кодовых комбинаций на вход передатчика в некоторые промежутки времени может превышать скорость телеграфирования. Чтобы не допустить потерь Информации при передаче с клавиатуры, она вводится последовательно, знак за знаком, в буферное устройство (буферный накопитель) емкостью 32 кодовые комбинации (это устройство входит в состав блока клавиатуры). Считывание кодовых комбинаций из буферного накопителя осуществляется с постоянной скоростью, соответствующей скорости телеграфирования. В передатчике каждая кодовая комбинация преобразуется в телеграфный сигнал (последовательную стартстопную комбинацию телеграфных посылок) который через согласующее устройство передается затем в линию связи. При приеме телеграфный сигнал с линии связи поступает на Вход электронного приемника, где проходит через согласующее устройство преобразуется в кодовые комбинации, поступающие с выхода приемника на печатающее устройство, реперфоратора и дешифратора кодовой комбинации запроса автоответа (кто там?). При поступлении такой комбинации дешифратор запрещает реперфоратору перфорацию комбинации запроса автоответа на перфоленту, включает коммутатор автоответа. Коммутатор автоответа подключает ко входу передатчика встроенное в коммутатор устройство автоответа емкостью 20



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

пятиразрядных кодовых комбинаций и посылает комбинации автоответа в передатчик. В устройство автоответа предварительно заносится код (определенная последовательность десятичных цифр и букв русского алфавита, например, 123456-МЕРКУРИЙ), присваиваемый на телеграфной сети только данному оконечному пункту. Во время передачи автоответа работа реперфоратора также блокируется.

При отсутствии сигнала с линии в течение 45с автостоп выключает двигатель приемного аппарата.

Дешифратор служебных сигналов формирует сигналы управления, соответствующие кодовым комбинациям ПЕРЕВОД строки, возврат каретки, ПЕРЕВОД СТРОКИ ВОЗВРАТ КАРЕТКИ, которые управляют работой шаговым двигателями печатающего устройства.

- местном режиме информация с клавиатуры может быть занесена в запоминающее устройство емкостью 1024 пятиразрядные кодовые комбинации или отперфорирована на перфоленте с помощью реперфоратора (с одновременной печатью на рулонную телеграфную ленту печатающим устройством) [5].

Блок питания служит для получения стабилизированных напряжений несколько значений, используемых для питания электронных схем, шаговых двигателей трансмиттера и печатающего устройства, линейных цепей аппарата. Блок опорных частот формирует сетку значений высокостабильных частот, используемых для синхронизации работы отдельных блоков и всего аппарата.

Блок клавиатуры

Клавиатура предназначена для ручного ввода информации в передатчик, запоминающее устройство или реперфоратор. Клавиатура представляет собой конструктивно законченный блок, состоящий из двух отдельных частей: механической контактной и электронной.

Все клавиши блокируются электрически в следующих случаях:

2. при работе трансмиттерной приставки;
3. при работе запоминающего устройства в режиме вывода информации;
4. при работе коммутатора автоответа;
5. при автоматической непрерывной передаче одной и той же кодовой комбинации с клавиатуры.

При передаче с клавиатуры 69 отпечатанных с начала строки знаков электрически блокируются все клавиши, кроме клавиш:



-возврат каретки;



-новая строка.

В этом случае блокировка любой из клавиш снимается.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

При работе на клавиатуре за 13 знаков до конца строки срабатывает звуковая сигнализация, предупреждая об окончании строки (работает до конца строки).

Механическая контактная часть клавиатуры содержит группу из 49 клавиш с нанесенными на них цифрами, служебными знаками, буквами русского и латинского алфавита.

Принцип работы трансмиттера

Трансмиттерная приставка предназначена для считывания информации с перфоленты и выполнена в виде конструктивно и функционально законченного блока. Считывание информации осуществляется автоматически со скоростью, соответствующей выбранной оператором скорости телеграфирования (50 или 2 Бод).

Трансмиттерная приставка обеспечивает:

3. продвижение перфоленты и считывание кодовой комбинации, как в пошаговом, так и в непрерывном режимах по сигналам оператора или автоматически;
4. автоматическое прекращение считывания информации по окончании перфоленты или при открывании ленто прижимной крышки.

Считывание информации осуществляется с помощью оптоэлектронных пар: светоизлучающий диод — фотодиод. Линейка из пяти светоизлучающих диодов (пять - для информационных дорожек и один — для синхро - дорожки) располагается над перфолентой, а линейка из шести фотодиодов - под перфолентой. В момент прохождения отверстия синхро - дорожки мимо считывающего узла свет попадает на фотодиод синхро - дорожки, и электронной схемой узла считывания формируется сигнал (сопротивление-малое) зависит от отсутствия (наличия) отверстия в соответствующей информационной дорожке. В режиме копирования перфолент кодовые комбинации поступают в блок реперфоратора.

Механизм продвижения ленты в трансмиттера собран на базе шагового двигателя, управляемого электронной схемой. Сигнализатором окончания перфоленты является дополнительная оптоэлектронная пара. Реперфораторная приставка предназначена для записи информации на перфоленту в виде системы отверстий и выполнена в виде функционально и конструктивно законченного узла.

Реперфораторная система обеспечивает:

5. перфорацию кодовых отверстий на перфоленте;
6. выпуск перфоленты;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

13. автоматическое выключение двигателя, если информация отсутствует в течение 45с (автостоп);
14. запрет перфорации комбинаций кода МТК-2 (кто там?);
15. механический позначный возврат перфоленты (используется для исправления ошибок).

Принцип работы запоминающего устройства

Полупроводниковое запоминающее устройство (ЗУ) емкостью 1024 пятиэлементных кодовых комбинаций служит для подготовки, корректирования и хранения информации, предназначенной для передачи и автоматической передачи этой информации на вход печатающего устройства, реперфоратора или передатчика со скоростью, соответствующей выбранной скорости телеграфирования.

Ввод и вывод кодовых комбинаций, осуществляется параллельным способом занесение информации в ЗУ, происходит с помощью клавиатуры или трансмиттера в местном режиме работы. При этом обеспечивается "возможность исправления замеченных ошибок в занесенной в ЗУ информации и редактирования текста путем следующих действий:

- В исправление одиночных ошибочных знаков на правильные;
- В уничтожение ошибочных знаков, отдельных слов и строк до 69 знаков в одной строке;
- В уничтожение всей занесенной в ЗУ информации;
- В вписыванием пропущенных или дополнительных знаков, слов и частей текста в произвольном месте ранее занесенной информации.

ЗУ имеет световую индикацию, срабатывающую при:

- В выводе информации;
- В заполнении всей емкости Памяти;
- В записи с 960-го по 1023-й знак (сигнализации окончания емкости ЗУ).

После ввода в ЗУ 1024 знаков работа клавиатуры или трансмиттера автоматически блокируется. Одна и та же информация из ЗУ может выводиться многократно. Предусмотрена возможность блокировки ввода и вывода информации, начиная с произвольного места в ЗУ, с последующей разблокировкой. При выключении питания ЗУ или питания аппарата хранящаяся в запоминающемся устройстве информация полностью уничтожается.

Конструкция и органы управления

Телетайп F-2000 имеет настольное исполнение. Электронная схема аппарата выполнена на дискретных элементах и микросхемах средней и



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

большой степени интеграции, размещенных на стеклотекстолитовых платах с печатным монтажом. Механические перемещения в аппарате осуществляются с помощью шаговых двигателей и электромагнитов. Соединение между блоками и узлами аппарата осуществляется с помощью плоского ленточного кабеля и типовых разъемов для подключения прибора диагностики с целью выявления неисправностей. Основные блоки и узлы аппарата имеют спец разъёмы.

Справа к корпусу ТЛГ аппарата прикреплены реперфоратор и трансмиттер со своими органами управления.

На лицевой панели аппарата закрепляются: клавиатура, органы управления аппаратом, органы управления и контроля ЗУ. С задней стороны аппарата имеются два кабеля. Кабель, для подключения к сети 220 в, и кабель для подключения к линии связи оканчивающейся четырехштырьковой телефонной вилкой и включается в розетку типа РТШ (розетка телефонная штепсельная).

Рулонная телеграфная лента и перфолента устанавливаются в специальных держателях, расположенных сзади телеграфного аппарата и реперфоратора. При отсутствии телеграфной ленты блокируется работа печатающего устройства, а при отсутствии перфоленты-реперфоратора, датчиками отсутствия лент являются специальные металлические рычаги и оптоэлектронные пары. При отсутствии ленты один из концов рычага препятствует прохождению света в оптоэлектронной паре от источника освещения к фотоприемнику.

Доступ к элементам печатающего устройства и к тумблеру включения блока питания обеспечивается при откидывании верхней крышки, закрепленной в исходном состоянии держателями с левой и с правой сторон. Лампа дневного освещения закреплена в специальных держателях на печатающем устройстве.

Назначение органов управления

аппарата Группа клавиш функциональной клавиатуры



Непрерывный пуск (повторение последнего напечатанного знака).

Русский регистр

Латинский регистр



Цифровой регистр



Звонок

Новая строка. Последовательно выполняется возврат каретки



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

и перевод строки передается в линию ВК и ПС.

Группа клавиш функциональной клавиатуры со светоиндикацией

- светоиндикатор не светится
- светоиндикатор светится
- светоиндикатор мигает

Включенный светоиндикатор указывает на нажатую клавишу. Мигание светоиндикатора указывает на неисправность (не запровадена лента, бумага и т. д.)

Группа клавиш включения режимов работы

- Отбой Мигание светоиндикатора указывает на то, что телетайп отключен от линии. Светится в состоянии покоя. Не светится в состоянии покоя.

- Местный режим.

- КТО ТАМ? Запрос автоответа.

- Возврат каретки.

- Перевод строки.

- Местный режим без перебоев.

Если светоиндикатор мигает, то происходит стирание приемобufferного устройства.

- Вызов.

- Сокращенный набор.

Группа клавиш специальных функции
До набор.

- Вызов подготовлен

- Автоматическое разъединение подготовлено.

- Я ЗДЕСЬ Датчик автоответа.

Отключение сигнализации входящего вызова.



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»



Датчик времени, номера, даты.

Группа клавиш управления ЗУ
Режим программирования.



Включается только в режиме покоя.

Запоминающее устройство сокращенных наборов.



Если светоиндикатор мигает, то ЗУ заполнено.

Текстовое запоминающее устройство (ТЗУ)



Если светоиндикатор мигает, то ТЗУ заполнено.



Вывод из текстового накопителя.



Размещение текста.



Поиск



Стирание. Для стирания слова или текста или номера абонента. При этом 3 раза раздается звуковой сигнал.



Возврат.

Группа клавиш управления ПУ



ПУ-вкл/выкл. Если светоиндикатор мигает, то телетайп не подготовлен к работе (ре закрыта крышка кожуха, не заправлена бумага).



Многократный перевод строки.



Видимость. Поднимает вверх печатную бумагу на 4 строки вверх. Группа клавиш управления приставками автоматике. Включение реперфоратора. Если светоиндикатор мигает, то реперфоратор неподготовлен к работе (не заправлена перфоленте).

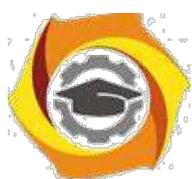


Включение трансмиттера.

Деблокировка клавиатуры.



Деблокировка клавиатуры. Светоиндикатор красный, светится, если клавиатурное буферное запоминание заполнено.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Режим покоя.

В этом режиме светится светоиндикатор "Сеть".

Если какой либо светоиндикатор мигает, то телетайп не готов к работе. Когда телетайп будет подготовлен к работе, мигание светоиндикаторов устранился.

Работа в местном режиме.

Существуют 2 разновидности ЭТОГО режима:

2. Местный режим с перебоем принимаемой телеграммой;
3. Местный режим без перебоев принимаемой телеграммой.

этом режиме телеграмма записывается в приемобуферном ЗУ.

Работа с печатающим устройством (ПУ).

Нажать клавишу   "Местный режим без перебоев".

- Светоиндикатор начнет, СВЕТИТСЯ;
- Печатающая головка уходит в начало строки, одновременно выполняется перевод строки;
- Включается освещение печатного поля;
- Светоиндикатор ПУ светится.

При нажатии одной из клавиш местного режима или клавиши вызова ПУ сразу готово к работе (включено).

Если телетайп находится в местном режиме, то чтобы выключить ПУ надо один раз нажать на клавишу  ПУ вкл. Светоиндикатор погаснет. Печатающая головка останется на последней печатной позиции. Включается ПУ повторным нажатием на клавишу ПУ вкл. 

Многократный перевод строки.

Нажать клавишу  Многократный перевод строки.

На КЛВ нажать цифру от 0 до 9.

Цифра 1 соответствует переводу строки на 1 шаг, 2-2 шагам и т.д.

Цифра 0 соответствует 10-ти кратному переводу строки.

нажать клавишу  Многократный перевод строки.

нажать клавишу  Перевод строки.

произойдет перевод строки на заданное число шагов.

Работа с приставками автоматики.

Работа с реперфоратором.

нажать клавишу  Местный режим.

или   Местный режим без перебоев.

Нажать клавишу  Реперфоратор.

Светоиндикатор начинает светиться. Реперфоратор автоматически перфорирует 24 комбинации РУС (дорожка-ведущей перфорации).



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

Нажать клавишу  Новая строка.

С клавиатуры вводимый текст печатается и перфорируется.

Коррекция перфоленты способом полной перфорации. Нажатием до упора клавиши " возврат перфоленты" перфолента продвигается на 1 знак назад. Максимальное число возвратов на 20 знаков.

Нажать клавишу ЛАТ происходит полная перфорация. Количество нажатий на клавишу ЛАТ равно количеству возвратов перфоленты.

Перфоратор выключается нажатием на клавишу  "перфоратор". При этом автоматически перфорируется 24 комбинации РУС и реперфоратор отключается. Светоиндикатор отключается.

Работа с трансмиттером.

нажать на клавишу  "Местный режим"
или   "Местный режим без перебоев".

Заправить перфоленту.

Возможны два варианта работы трансмиттера:

3. непрерывный пуск;
4. позначный пуск.

При непрерывном пуске надо длительно нажать клавишу "ТРИ", перфоленте продвигается, текст печатается. Для изготовления перфоленты включается перфоратор.

При позначном пуске коротко нажать клавишу "ТРИ" один знак считывается с перфоленты и перфолента продвигается на 1 шаг вперед. Выключение ТРМ-той же клавишей.

Передача телеграмм с КЛВ.

Передача телеграммы.

1. Нажать клавишу  "Вызов". Светоиндикатор светится. Печатающее устройство печатает: GA

2) Набираем номер. ПУ печатает набранный номер.

Если соединение установилось, то ПУ печатает: DF

Если соединение установилось, то аппарат печатает одно из сообщений: ОСС, DER, NP, NC и выключается.

3) Нажать клавиши ЦИФ,  ПУ печатает автоответ противоположного абонента.

4) Нажать клавишу " Я ЗДЕСЬ ". Происходит передача и печать собственного автоответа.

5) Для разъединения нажать клавишу "Отбой".

6) Передача и прием телеграмм через постоянную связь "ТОЧКА- ТОЧКА".



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Нажать клавишу "Вызов".

- 5) Светоиндикатор светится;
- 6) Включается освещение печатного поля;
- 7) Печатающая каретка устанавливается в начале строки.
передать текст.

Телетайп отключается через 1 мин. после последнего отпечатанного знака.

Передача и прием телеграмм с помощью приставок автоматики.

Передача телеграмм с помощью трансмиттера.

4. Заправить в ТРИ перфоленту.
5. Установить соединение.
6. Нажать клавишу « Вкл. ТРН», лента транспортируется. Текст печатается и передается.
7. Нажать клавишу "КТО ТАМ". Печатается автоответ противоположного абонента.
8. Нажать клавишу "отбой". В режим покоя. Кроме этого возможно автоматическое разъединение соединения после окончания перфоленты. Если после включения трансмиттера нажать клавишу "Автоматическое разъединение подготовить", то произойдет автоматическое разъединение соединения с обменом автоответами.

Передача и прием телеграмм с помощью дистанционного вызова считывающего устройства.

Дистанционный вызов считывающего устройства возможен если противоположный абонентский телетайп для этого подготовлен следующим образом:

и В режиме покоя заправить перфоленту в ТРМ.

и Нажать клавишу "Вызов подготовлен".

5. Светоиндикатор светится.

Для дистанционного вызова ТРМ противоположного абонента нужно:

6. Установить связь:

Нажать клавишу "Вызов".

7. Светоиндикатор светится;
8. ПУ отпечатает: GA.

Набрать номер противоположного абонента.

Соединение установится.

8. ПУ отпечатает: DF.

2. Нажать клавишу  "КТО ТАМ?".

2. ПУ отпечатает автоответ противоположного абонента.

3. Нажать клавишу  "Я ЗДЕСЬ". Передача и печать собственного автоответа.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

3. Нажать клавишу  "Новая строка".
4. Дистанционный вызов считывающего устройства. С КЛВ передать комбинацию знаков (ЛАТ) KLKL. Считывающее устройство противоположного абонента включается. Текст передается и печатается на обоих телетайпах (если режим полудуплекс).
5. После окончания передачи текста нажать клавишу "КТО ТАМ?". Печатается автоответ противоположного абонента.
6. Нажать клавишу "Отбой".

Дистанционное включение реперфоратора.

- > Установка соединения.
 - > Обмен автоответами.
 - > Включение РПФ противоположного абонента передачей 4-х комбинаций "ЛАТ" "С" (СССС) Четыре с можно передавать перед обменом автоответами.
4. Текст передается, а у противоположного абонента перфорировается. Выключение дистанционного включения реперфоратора происходит в любое время передачи телеграммы передачей 4-х комбинаций "ЛАТ" "F" (FFFF)
 5. Обмен автоответами.
 6. Окончание передачи.

Виды запоминающих устройств и их использование.

Требования техники безопасности.

К работе с аппаратом допускаются лица, изучившие технические характеристики, структурную схему, принцип работы и конструкцию аппарата, назначение органов управления и следующие правила техники электробезопасности.

При работе с электроаппаратом запрещается:

- >включать аппарат в сеть без разрешения преподавателя;
- >выключать аппарат из сети при включенном тумблере блока питания;
- >эксплуатировать аппарат с открытым кожухом;
- > производить чистку и смазку механизмов, соединение и разъединение разъемов, регулировку и устранения неисправностей в электрической схеме аппарата без отключения его от питающей сети;
- >производить заправку перфоленты при включенном двигателе реперфоратора;
- > касаться голыми руками контактов печатных плат, разъемов, электродвигателя, электромагнитов при включенном питании.

Порядок практической работы с телетайпом F-2000

4. Проверка работоспособности в местном режиме



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

5. Перед началом работы на аппарате необходимо проверить наличие рулонной телеграфной ленты и перфораторной ленты (при отсутствии необходимо произвести их заправку).

С Подключить сетевой кабель к электросети. Поднять верхнюю крышку аппарата. Установить тумблер блока питания в положение СЕТЬ. Опустить крышку. Включить местный режим. Включить реперфораторную приставку. Передать с клавиатуры текст (текст должен печататься на телеграфной ленте и перфорироваться на перфоленте), искусственно внося в него в заданных преподавателем местах ошибки и исправляя их на перфоленте.

d) Проверка работы транзиттерной приставки. Убедиться в отсутствии ошибок в напечатанном тексте.

e) Проверить работу коммутатора автоответа.

f) Проверить работу запоминающего устройства.

Содержание отчета по работе.

С Комплект аппарата F-2000.

С основные технические характеристики.

3. Описание результатов проверки работоспособности аппарата (при этом все напечатанные тексты аккуратно вклеиваются в отчет, а перфоленты вкладываются в наклеенный бумажный карман).

Контрольные вопросы

3) Технические характеристики телетайпа F-2000

4) В чем заключается общий принцип работы аппарата.

5) Как работает блок клавиатуры?

6) Как работает блок печатающего устройства?

7) Как работает транзиттер и реперфоратор?

8) Как работает запоминающее устройство?

9) Какое назначение имеют клавиши управления работой реперфоратора и транзиттера?

10) Какое назначение имеют клавиши управления запоминающего устройства?

Лабораторная работа № 2.2

Изучение телеграфного аппарата F-2500

Цель работы: изучение технических характеристик и устройства телетайпа F-2500, включение, настройка и работа в местном режиме, режиме программирования, передача телеграмм, работа с электронной доской объявлений.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Содержание работы. Изучить назначение, технические характеристики принцип действия аппарата F-2500. Провести включение и проверку работоспособности аппарата в местном режиме. Провести передачу и прием заданной информации с помощью аппарата F-2500. Произвести запись информации на дискету, считать записанную информацию с дискеты. Редактирование текста. Распечатка текста на бумаге. Запрограммировать в телетайпе время, число, провести запрос сигнала об ошибках. Подготовить отчет по лабораторной работе.

Назначение и технические характеристики аппарата F-2500

Аппарат F-2500 представляет собой современный терминал связи, отвечающий самым высоким требованиям, предъявляемым сегодня к средствам коммуникации. Программируемая электроника управления делает возможным легкий доступ в локальный (местный) режим для осуществления подготовки сообщения независимо от того, идет ли передача или прием встречного сообщения [8].

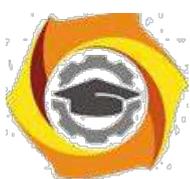
Благодаря наличию накопителя на магнитном диске аппарат обладает практически неограниченной памятью для архивации Ваших сообщений. Аппарат построен из отдельных модулей: основной аппарат, монитор и клавиатура. Такая конструкция позволяет выбрать оптимальное размещение на рабочем месте.

Функциональные возможности аппарата F-2500 могут быть расширены добавлением к нему перфоленточного устройства и платы сопряжения с персональным компьютером MS/DOS.

Технические характеристики аппарата F-2500

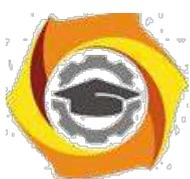
Таблица 2.2.1. – Технические средства аппарата F-2500

Подключение к телеграфной сети	телеграфный алфавит МТК (МККТТ)
Кодирование знаков	и другие 5-элементные коды
Скорость телеграфирования	50/75/100 Бод-переключаемо, возможно перепрограммирование на 200 Бод
Вид модуляции	- высокий уровень, однополюсной ток 40мА; - высокий уровень, двухполюсной ток 20мА при 20, 30, 48, 60 В; - низкий уровень, переменный ток, стык Е.
Подключение к сети питания	
Напряжение сети	220В 15%



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Частота сети	40-60 Гц
Потребляемая мощность	в режиме готовности - 18 Вт в рабочем режиме - 58 Вт
Рабочие параметры	
Уровень шума	вакс. 60 дБ
Модули аппарата(можно устанавливать отдельно)	
Основной аппарат	содержит следующие узлы: - печатающее устройство; - дисковод; - электроника управления; - блок подключения; - электропитание.
Клавиатура	подключается через кабель к основному аппарату
Монитор	подключается через кабель к основному аппарату
Печатающее устройство	
Способ печати	точечная матрица 9 14
Скорость печатания	50 знаков/с
Виды шрифта	латинский шрифт и кириллица, большие буквы
Типы шрифта	- левый курсив - диалог с оператором; - правый курсив-местный режим и передача; - прямой шрифт-прием.
ЗУ на магнитных дисках, дискета	микро дискета 3,5дюйма - DS/DD
емкость памяти	720 кБайт в формате DOS
Электронное ЗУ	емкость 32кБайт
Накопитель сообщений. Приемный накопитель ЗУ комфортного управления, состоит из	
- Накопитель кратких текстов	до 26 кратких текстов вакс. по 256 знаков
- Накопитель для автоматических вызовов	до 35 заданий на передачу макс. по 10 сообщений каждое
- Накопитель для автоматических выводов абонентом	до 35 выводов макс. по 10 сообщений каждый
- Наборный накопитель для сокращенных наборов	макс.35 телексных номеров набора абонентов, макс. по 32 наборных знака и 25 знаков комментария



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- Накопитель повторных наборов	автоматическое запоминание последнего набранного телексного номера
--------------------------------	--

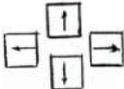
Функциональные клавиши аппарата F-2500

Таблица 2.2.2 – Функциональные клавиши аппарата F-2500

Клавиши управления		
	местный режим	для подготовки сообщения включается "местный" режим" и одновременно экран монитора. Если в это время приходит встречное сообщение, то местный режим покидается, и принимаемое сообщение выводится на печать.
	Невозмущенный местный режим	Для подготовки сообщения без помех входящими вызовами включается "невозмущенный местный режим" и одновременно экран. Заканчивается этот режим работы повторным нажатием этой клавиши, либо в случае наполнения приемного буферного накопителя и если требуется отпечатать принятое сообщение.
	Вызов	После нажатия этой клавиши и получения запроса к набору можно вызвать абонента. Если работаете в режиме прямой связи, то нажатием этой клавиши связь установлена.
	Отбой	Нажатием этой клавиши заканчиваете связь. Во время соединения и на протяжении передачи СИД светится. При нарушении на линии СИД мигает.
	Сокращенный набор	Нажатием этой клавиши и последующим набором сокращенного номера запускается сокращенный либо повторный набор.
	До набор	Для произведения до набора в режиме передачи аппарат переключается в состояние набора. При наборе СИД светится. В режиме ОТХ светоиндикатор мигает.
	Автоматическое разъединение	После нажатия этой клавиши соединение в конце передаваемого сообщения

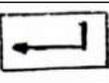


**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

		автоматически разъединяется (в надлежащем порядке, с обменом автоответами).
	Текстовый накопитель	Включение режима обработки текстов. Если СИД мигает, накопитель текстов полон.
	Поиск	Нажатием этой клавиши запускается процедура поиска или вызываются списки содержания.
	Стирание	Клавиша стирания для корректировки текста или для отмены ошибочно введенной команды.
	Режим программирования	Ввод программируемых функций после длительного нажатия клавиши в состоянии режима покоя. СИД светится. Подготовка программирования накопителя наборов накопителя кратких текстов, накопителя вызовов, накопителя заданий на передачу и установки меток останова кратким нажатием клавиши в местном режиме. СИД мигает.
	Клавиши курсора	С помощью этих клавиш курсор позиционируется на экране. Каждое нажатие клавиши соответствует смещению курсора на одну строку вверх/вниз или на один символ влево/вправо.
Алфавитно-цифровая клавиатура		
	Переключение на латинский шрифт.	
	Переключение на русский шрифт.	
	Переключение на цифры/знаки.	
	При одновременном нажатии с клавишами курсора производится перемещение курсора, по словам влево/вправо либо вверх/вниз к началу или к концу текста. При одновременном нажатии с клавишей "стирание" происходит стирание всех знаков одного, начиная с позиции курсора.	



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

	Непрерывное повторение последнего введенного символа; прекращается с отпусканьем клавиши.
	При нажатии производится переход на следующую строку (без возврата каретки к началу строки).
	При нажатии происходит возврат к началу строки (без перевода строки).
	Производится возврат каретки и перевод строки.
	Вызывается код опознавания Вашего встречного абонента.

Порядок выполнения практической работы с F-2500

Редактирование текста в местном режиме.

Для обслуживания диска, нажатием клавиши текстового накопителя, включаем режим обработки текстов. Главное меню дает Вам обзор возможных процедур обработки текста, из которых Вы выбираете нужную. Затем записываем текст (обратите внимание на его номер в левом верхнем углу экрана).

Вернемся в текстовый накопитель и проведем все возможные процедуры обработки текста, посредством нажатия клавиш: 1 2 3 4 5 6. Для обслуживания диска выбираем процедуру 6. Нажимая клавишу 6, вызываем имена для работы с дискетой: обзор диска. Просматриваем содержание диска, нажимая клавишу 4.

Запись текста на магнитный диск.

Вернемся в текстовый накопитель, просмотрим содержание, выберем номер текста который необходимо записать на диск, запоминаем его. Входим в обслуживание диска: 6. После нажатия 1 Вы должны указать номер текста, который Вы желаете переписать и присвоить ему название.

Печать текста на бумагу (вывод).

Указать название текста который Вы хотите отпечатать (см. содержание). Дать подтверждение. Вы можете провести все возможные процедуры со своим текстом [9].

Режим программирования.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Во время режима программирования телетайп не может быть вызван абонентом. Включение режима программирования возможно только из состояния эксплуатационного покоя путем нажатия клавиши режим программирования в течение 1 сек. если ввода не последовало, то через некоторое время режим программирования автоматически отключаются. В режиме программирования нам можно установить любые версии из ниже перечисленных в таблице 2.2.3.

Таблица 2.2.3 – Функция и ее буква.

Функция	Кодовая буква	Функция	Кодовая буква
Тест	А	Число	В
Время	С	Телегр.	D
Авт.ПС	Е	Кол.Знаков	F
Интервал.СТР	G	Сигнал	Н
Текст файла	I	ВСТВ/ПРПС	J
Дисплей	К	DX/НХ	М
БОД	N	ПЕРф	О

(обозначенные звездочкой функции в этом аппарате отсутствуют)

Далее введением соответствующей буквы Вы выбираете подлежащую программированию версию А...0 и подтверждаете, теперь задается требуемое значение и подтверждается.

Передача телеграмм с телетайпа F-2500

Для набора текста телеграммы необходимо включить местный режим работы телетайпа: «Т».

Набор текста производится в следующем порядке:

- 4) нажать клавишу ввода 2 раза;
- 5) набрать с клавиатуры ЗЦЗЦ пробел «П» и без пробела дать набор города в который посылается телеграмма (номер набора искать в справочнике);
- 6) нажать клавишу ввода 2 раза;
- 7) набрать с клавиатуры РОСТОВ-НА-ДОНУ 123432/1 (цифра после дробной черты обозначает порядковый номер телеграммы 1,2,... 123432 - автоответ нашего телетайпа), затем дать пробел и цифрами указать количество слов в



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

телеграмме, затем через пробел указать дату и далее через пробел указать время. В конце, посредством цифрового регистра дать знак =

НАПРИМЕР: РОСТОВ НА ДОНУ 123432/2 20 13/1 1020 = ;

9. нажать клавишу ввода 2 раза;
10. набрать адрес получателя телеграммы и указать его фамилию после чего дать знак = ;
11. нажать клавишу ввода 2 раза;
12. набрать текст телеграммы, соблюдая следующее: после текста перед подписью ставится знак =, после подписи знак - ; Например: ДОЕХАЛ ХОРОШО = СЕРГЕЙ - ;
13. нажать клавишу ввода 2 раза;
14. набрать НННН дать пробел и нажать клавишу ввода.

Таким образом, текст подготовлен к отправлению. Выходим из местного режима последовательно нажав «Т». Включаемся в линию. Выходим на ЦКС набирая с клавиатуры 123912. Запрашиваем автоответ ЦКС нажатием клавиш «ЦИФ». Даем свой автоответ посредством клавиши ♦. Выводим текст телеграммы в линию посредством последовательного нажатия клавиш: «Т» «З» (необходимо указать номер текста телеграммы, под которым она была записана в памяти), нажать клавишу ввода. После окончания вывода текста телеграммы необходимо быстро нажать клавишу «Т» и запросить автоответ ЦКС посредством нажатия клавиш «ЦИФ». После получения автоответа ЦКС можно считать что телеграмма передана.

Работа телетайпа с электронной доской объявлений

При включенном в сеть аппарате F-2500 мы имеем возможность работы с электронной доской объявлений (ЭДО) любого города где существует эта служба.

Для выхода на ЭДО г. Ростова-на-Дону необходимо следующее:

- В Включиться в линию нажатием клавиши;
- В Набрать с клавиатуры номер набора ЭДО г. Ростова-на-Дону 123000;
- В Через некоторое время произвести запрос автоответа посредством последовательного нажатия клавиш;
- В На экране появится сообщение: дайте КТКТНННН пробел.

Необходимо с клавиатуры набрать КТКТНННН и дать пробел посредством нажатия нижней широкой клавиши;

На экране появится меню, которое будет распечатываться на бумаге. Заинтересовавший Вас пункт меню можно просмотреть и одновременно распечатать следующий образом: набрать с клавиатуры КТКТ и номер требуемого пункта, затем сделать пробел посредством нажатия широкой клавиши. Например: КТКТ0001.

Аналогичные ЭДО имеются:

- В в Екатеринбурге - номер набора 721000;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

2. в Хабаровске - номер набора 141000.

Отчет по работе

- В Основные технические характеристики.
- В Описание результатов проверки работоспособности аппарата.
- В Приложить к отчету распечатанную телеграмму.
- В Работа с электронной доской объявлений.

Контрольные вопросы

- 1.Содержание работы.
 - Основные технические характеристики аппарата F-2500.
 - Общий принцип работы.
 - Работа в общем режиме (местном).
 - Передача телеграммы в режиме программирования.
 - Клавиатура и записывающее устройство.
 - Передача телеграмм с помощью дискеты.

Лабораторная работа № 3

Изучение факсимильной связи.

Целью настоящей работы является изучение принципов факсимильной связи, и работы устройства факсимильной аппаратура группы G3, ITU-T на примере факсимильного аппарата типа МХ-2 фирмы 5НА1Р, а так же овладение практическими навыками работы.

Задание:

- Пользуясь настоящим описанием и рекомендованной литературой изучить:
- В принцип факсимильной связи;
 - В требования к факсимильной аппаратуре группы 3 рекомендаций ITU-T;
 - В принцип действия, технические характеристики, устройства и конструкцию факсимильного аппарата ТХ-2;
 - В овладеть практическими навыками работы с факсимильным аппаратом, подготовить письменный отчет по лабораторной работе.

Пояснения к работе.

Область электросвязи, которая занимается передачей неподвижных изображений по каналам электрической связи, называется факсимильной связью. Факсимильной способ передачи информации является универсальным.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Им одинаково удобно передавать любые изображения: цветные и чёрно-белые фотографии, чертежи, графики, корреспонденцию. В случае передачи полутоновых чёрно-белых изображений факсимильные системы, называемые фото факсимильными, поскольку носителем записи чаще всего является фотоплёнка или фотобумага. Системы, обеспечивающие передачу цветных изображений, называются цветными факсимильными. Принцип осуществления факсимильной связи состоит в следующем:

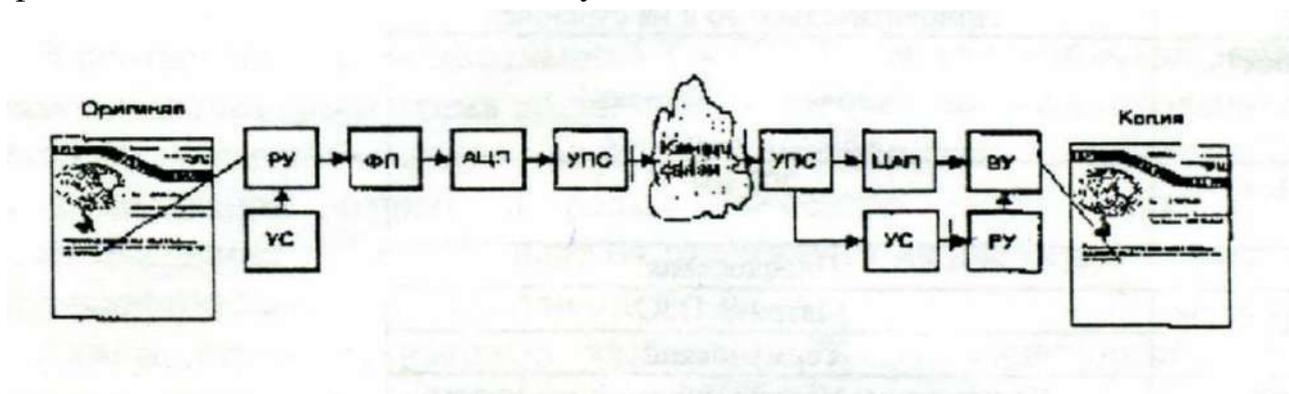


Рисунок 3.1- Схема факсимильной связи

Изображение (оригинал), подлежащее передаче, подвергается сканированию световым пятном требуемых размеров. Пятно формируется светоптической системой, содержащей источник света и оптическое устройство. Перемещение пятна по поверхности оригинала осуществляется развертывающим устройством (РУ). Часть светового потока, падающего на элементарную площадку оригинала, отражается и поступает на фотоэлектрический преобразователь (ФП), в котором происходит его преобразование в электрический видеосигнал. Амплитуда видеосигнала на выходе фотопреобразователя пропорциональна величине отраженного светового потока. Далее видеосигнал поступает на вход аналого-цифрового преобразователя (АЦП), где преобразуется в цифровой код.

4. выхода АЦП цифровой код поступает на вход устройства преобразования сигналов (УПС), в соответствии с одним из протоколов модуляции спектр цифрового видеосигнала переносится в область частот используемого канала связи [6].

На приемной стороне, входящий из канала связи модулированный сигнал последовательно поступает в УПС и ЦАП для демодуляции и цифроаналогового преобразования. Далее видеосигнал поступает в воспроизводящее устройство (ВУ), где в результате действия развертывающего устройства на бланке воспроизводится копия переданного изображения.

Процесс получения конечной факсимильной копии, обратный процессу сканирования, носит название репликации. Для обеспечения синхронности и



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

синфазности разверток на передающей и приемной сторонах используются устройства синхронизации (УС).

Согласно рекомендациям ТТУ-Т в зависимости от используемого вида модуляции различают факсы четырех групп.

Отличие факс аппаратов группы 3 от более ранних заключается в полностью цифровом методе передачи со скоростями до 14000 бит/с. В результате, применения сжатие, факс группы 3 передает страницу за 30 - 60 секунд. При ухудшении качества связи факсы группы 3 переходят в аварийный режим, замедляя скорость передачи.

Согласно стандарту группы 3 возможны две степени разрешения: стандартная, обеспечивающее 1728 точек/дюйм по горизонтали и 100 точек/дюйм по вертикали и высокая, удваивающая количество точек по вертикали, что даёт разрешение 200 x 200 точек/дюйм и в двое уменьшает скорость.

Факсимильные аппараты третьей группы ориентированы на использование телефонных каналов КТСОП. Практически все продаваемые в настоящее время факсы основаны на стандарте группы 3.

Факсимильный аппарат типа NX-2 фирмы SHARP.

Основные технические характеристики факсимильного аппарата представлены в таблице 3.1.

Таблица 3.1 - Характеристики факсимильного аппарата

Тип используемых линий	Коммутируемая телефонная сеть общего пользования
Совместимость	Группа 3 МККГТ
Размер документа (максимальные размеры)	216 мм (ширина) 600 мм (длина)
Время передачи	Приблизительно 30 с на страницу
Разрешающая способность: Горизонтальная Вертикальная	8 точек/мм в строке 3,85; 7,7 линий/мм
Эффективная ширина сканирования	203 мм
Тип развертки	Плоскостная
Тип сканирующего устройства	Матрица ПЗС
Способ записи	Термический
Способ сжатия полосы	Кодирование модифицированным кодом Хаффмена
Скорость работы модема	9600/7200/4800/2400 бит/с
Модуляция	4- и 8- кратная АМ-ФМ
Потребляемая мощность	Приблизительно 65 Вт
Габаритные размеры	326x245x124 мм
Масса	3,0 кг

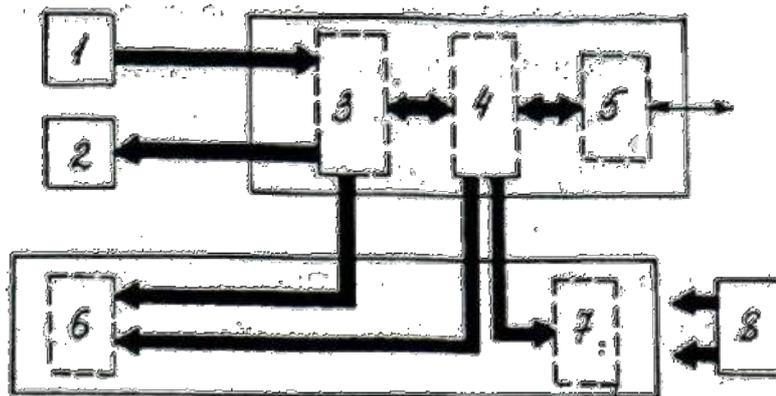


Рисунок 3.2 - Структурная схема цифрового ФАКСА 3-й группы.

На структурной схеме цифрового ФАКСА 3-й группы:

1 - считывающее устройство; 2 - устройство записи; 3 - кодек; 4 - управляющее устройство; 5 - модем; 6 - устройство подачи бумаги; 7 - панель обслуживания; 8 - блок электропитания.

Развертывающее устройство ФА, построенное с использованием приборов с зарядовой связью (ПЗС), не содержит каких-либо механических систем, кроме системы протяжки бланка с изображением, подлежащим передаче.

На рисунке 3.3 изображена структура прибора с зарядовой связью.

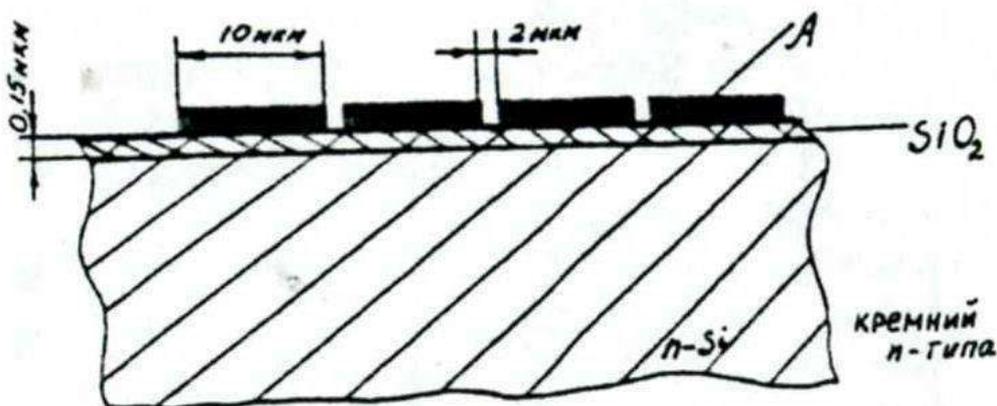


Рисунок 3.3 - Структура прибора с зарядовой связью.

Затворы имеют длину $h = 10 \text{ мкм}$, а расстояние между затворами составляет $d = 2 \text{ мкм}$.

5. приборах с зарядовой связью зарядовый пакет может быть введен путем локального освещения световым потоком, отраженным через оптическую систему факсимильного аппарата от бланка с изображением, подлежащим передаче. Фоточувствительная область выполняется на основе фотодиодов. Линейная фоточувствительная схема с зарядовой связью



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

выполнена на одном кристалле и представляет собой комбинацию области накопления с фотодиодной чувствительной структурой и двух сдвиговых транспортных ПЗС-регистров, каждый из которых отделён от неё разрешающим затвором и имеет собственные входные и выходные устройства. Кодирование по МКХ [7].

4. соответствии с рекомендацией Т.4 ИТУ-Т для факсимильной аппаратуры третьей группы, используется одномерная схема кодирования по МКХ (модифицированный код Хаффмана). МКХ является неравномерным кодом, обеспечивающим сжатие дискретных факсимильных сигналов путем кодирования черных и белых элементов изображения. Каждая серия элементов изображения, длина которой больше 64 разбивается на две серии - основную, длиной $N \times 64$ (где N -целое число) и завершающую длиной $0...63$.

Длины серий одинаковых элементов ($0...63$) кодируются кодовой комбинацией так называемых оконечных кодовых слов (ОКС). Длины серий, содержащих более 64 элементов, кодируются комбинацией начального кодового слова (НКС), которая соответствует требуемой длине серии, и комбинацией ОКС, которая определяет разницу между истинной длиной серии и длиной серии, закодированной ИКС. За каждой кодируемой строкой должна следовать специальная кодовая комбинация конца строки (КС) 000 ... 01 (12 бит), которая не встречается в кодах длин серий.

Характеристика модема.

Модем факса может вести передачу на скоростях 9600/7200/4800/ 2400 бит/с. Для работы с такой скоростью по каналу ТЧ используются многократные методы модуляции. В рекомендации V.29 МККТТ предусмотрен модем на 9600 бит/с, работающий с квадратурной амплитудной модуляцией (КАМ). Модем использует систему сигналов АМ-ФМ, т.е. сочетание многоуровневой АМ с многопозиционной ФМ.

Соответствие между мгновенными значениями параметров сигналов и четырехэлементными комбинациями битов указано на диаграмме сигналов и в таблице 3.2.



Таблица 3.2 - Соответствие между мгновенными значениями параметров.

Абсолютная фаза	Амплитуда	Q1	Q2	Q3	Q4
0°	3	0	0	0	1
	5	1	0	0	1
45°	$\sqrt{2}$	0	0	0	0
45°	$3\sqrt{2}$	1	0	0	0
90°	3	0	0	1	0
	5	1	0	1	0
135°	$\sqrt{2}$	0	0	1	1
	$3\sqrt{2}$	1	0	1	1
180°	-3	0	1	1	1
	-5	1	1	1	1
225°	$-\sqrt{2}$	0	1	1	0
	$-3\sqrt{2}$	1	1	1	0
270°	-3	0	1	0	0
	-5	1	1	0	0
315°	$-\sqrt{2}$	0	1	0	1
	$-3\sqrt{2}$	1	1	0	1

Характеристика устройства записи.

- факсе применяется устройство электротермической записи на специальную термочувствительную бумагу. Разрешающая способность электротермической линейки составляет восемь элементов/мм.

Электротермическая запись используется в среднескоростных аппаратах, обеспечивает хорошее качество воспроизведения и имеет низкую стоимость.

Инструкция по работе с факсимильный аппаратом.

Отправление документов:

R4X-2 может использоваться для получения как факсимильных сообщений, так и голосовых звонков по одной и той же телефонной линии, а также обеспечивает интегрированным прием сообщений любым автоответчиком или параллельными телефонами.

NX-2 имеет два режима приема:

- режим IN, используется когда абонент может отвечать на телефонные вызовы лично;
- режим OUT, используется, когда необходимо чтобы принимать телефонные вызовы автоматически.

режиме OUT, факс автоматически отвечает на все телефонные вызовы после нескольких контрольных посылок вызова, а затем принимает переданный документ. Число звонков исходящих вызовов, после которых отвечает факс



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

можно менять, путем смены опций. Может быть установлено любое число от 1 до 6.

Панель управления аппарата НАКР:

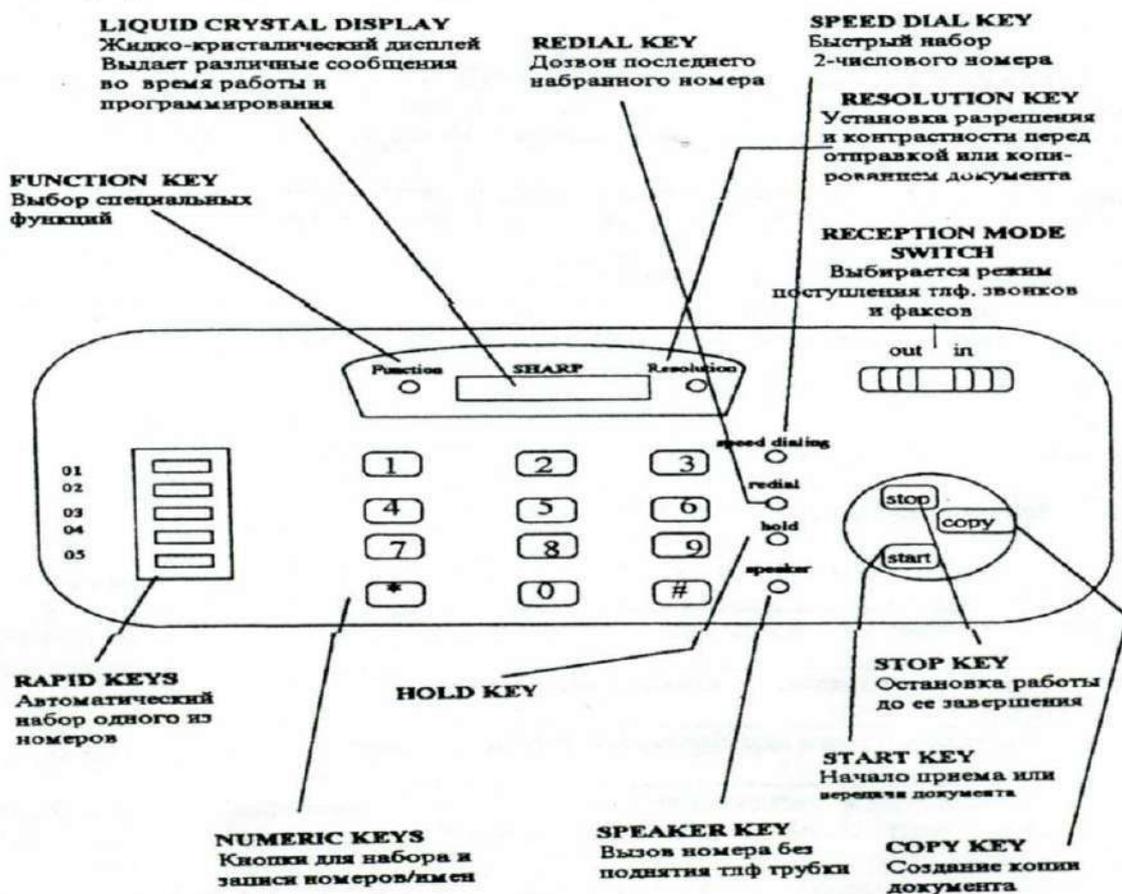


Рисунок 3.4 - Панель управления аппарата NX - 2.

Получение факсимильных сообщений на Г4Х-2.

Для получения факсимильных сообщений необходимо поднять трубку, после сигнала факса выждать до появления сообщения «RECEIVING», на экране дисплея факса, за тем положить трубку.

Если вызывающий абонент вначале передает голосовое сообщение, а затем хочет передать факс, то после разговора необходимо нажать кнопку «START», а затем положить трубку.

Функция Anti Junk Fax.

Функция Anti Junk Fax предотвращает прием факсов от нежелательных абонентов. Чтобы использовать эту функцию, необходимо ввести номер факса, от которого получать сообщения нежелательно.

Комментарии:

- и Функция Anti Junk Fax будет работать только тогда, когда факс находится в режиме OUT;
- и В список Anti Junk Fax может быть введено до 5 номеров.

Программирование функции Anti Junk fax приведено в таблице 3.3.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Таблица 3.3 - Функции Anti Junk

Шаг	Нажимаемые кнопки:	Комментарии
1	Function  	На дисплее появится <i>Data Entry</i> (ввод данных).
2		На дисплее появится <i>Store Junk Fax</i> .
3	 or 	Нажимаем «1» записывается номер в список <i>Anti Junk Fax</i> , «2» удаляет номер из списка.
4	 (пример)	Запись: Введите число от 1 до 5. Это число будет указывать на номер факса который Вы будете записывать в следующем шаге. Удаление: Введите число которое указывает на номер факса который Вы хотите удалить, а затем переходите к шагу #6.
5	  (пример)	Введите номер факса (макс. 20 цифр).
6	«START»	
7	Шаг 4 или «STOP»	Шаг 4 чтобы ввести (удалить) другой номер или нажмите «STOP» для выхода.

Установка опций.

Опций, которые можно использовать при настройке факса, производятся путем нажатия панельных кнопок так, как показано в таблице 3.4.

Таблица 3.4 – Опции настройки факса.

Шаг	Нажимаемые кнопки:	Комментарии
1	Function  	На дисплее появится <i>Optional Setting</i>
2	 or 	Для передвижения по списку установок вперед нажимать «#» или «*» для передвижения назад.
3	 (пример)	Когда необходимая установка появится на дисплее, изменение ее следует производить так, как описано ниже.
4	«STOP»	Выход

Доступ к установкам.

Установки:

Установка 1: NUMBERTO RINGS TO ANSWER (Число звонков для ответа).

Устанавливает число звонков, после которых включается принимающий факсимильный аппарат (в режиме OUT). Вводится любое число от «1» до «6». По умолчанию: «4»



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Установка 2: ввод ИМЕНИ И НОМЕРА ФАКСА

Чтобы запрограммировать ваше имя и факс / телефон номер который печатается наверху каждой страницы, используя инструкцию приведенную ниже.

Если Вы вводите, неправильный номер или символ, нажимает клавишу НОЕГ, чтобы переместить курсор назад в ошибку, затем вводите правильный номер или символ. (Чтобы перемещать курсор вперед, нажмите клавишу SPEAKER.)

Press these keys:	Шаги	Комментарии
 	1.	"ВВОД ДАННЫХ " Будет отображаться на дисплее.
 	2.	"ввод Собственного НОМЕРА " отображается дисплее.
	3.	Запоминание клавишей "start"
       (Example)		Ввод номера факсимильного аппарата (максимум из 20 цифр). Пробел между цифрами, устанавливается клавишей "#". Знак "+" устанавливается клавишей "*"
	5.	Запоминание клавишей "start"
              SPEAKER    (Example: "SHARP")	6.	Ввод имени (Максимум из 20 символов) осуществляется аналогично вводу имени письма
	7.	Запоминание клавишей "start"
		Exit.

Установка 2: УСТАНОВКА ДАТЫ И ВРЕМЕНИ

Дата и время появляется на дисплее и в сообщениях, печатаются наверху каждой страницы, которая передается, и используется для операций таймера. Установите дату и время, нажимая клавиши на панели операции, как показано на таблице 3.5.

- чтобы исправить ошибку, нажмите клавишу SPEAKER, чтобы переместить курсор назад в ошибку, и затем введите правильный номер.
- день недели устанавливается, с помощью ввода номера кода. Номера кода следующие:

Таблица 3.5 – День, код.

День	Воскресенье	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота
Код	0	1	2	3	4	5	6



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

Press these keys:	Шаги	КОМЕНТАРИИ
 	1.	" ВВОД ДАННЫХ " Будет отображаться на дисплее.
 	2.	Установка "ДАТЫ И ВРЕМЕНИ " будет отображаться на дисплее.
	3.	Запоминание клавишей "start"
  (Example: January)	4.	Ввод месяца (2 цифры "01 до 12")
  (Example)	5.	Ввод числа (2 цифры "01 до 31")
  (Example: 1994)	6.	Ввод года (2 цифры) .
 (Example: Thursday)	7.	Введите день недели, используя коды приведённые в таблице
  (Example)	8.	Введите час (2 цифры "01 до 12")
  (Example)	9.	Введите минуту (2 цифры 01 до 59)
 or 	10.	Нажмите "*" для выбора А.М., или "#" для выбора P.M.
	11.	Теперь часы начнут идти
	12.	Выход клавишей STOP

Порядок выполнения практической работы с факсом

- Проверить работу аппарата в телефонной линии, а также наличие рулонной факсимильной бумаги аппарате.
- 2. Осуществить сеанс связи с другим факсимильным аппаратом, установленным в лаборатории (номер факса указывает преподаватель).
- 3. Осуществить работу факсимильного аппарата в режимах IN OUT.

Отчёт по работе должен содержать:

- С Наименование лабораторной работы;
- С Цель работы;
- С Описание лабораторной установки;
- С Результаты сеанса связи.

Контрольные вопросы:

- С Сущность факсимильной передачи сообщения
- С Принцип работы факсимильного аппарата
- С Чем определяется качество воспроизведения копии изображения?
- С В чем состоит основной недостаток факсимильного способа передачи и в чем его достоинство?
- С Как устраняется избыточность факсимильного сообщения?
- С От каких параметров факсимильного аппарата зависит ширина спектра факсимильного сигнала?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Лабораторная работа № 4

Телеграфный терминал на базе ПЭВМ с модемом «Альфа-Телекс»

Введение

На сети общего пользования с 50-х годов применяют как ленточные, так и рулонные стартстопные телеграфные аппараты, на сети абонентского телеграфа

- рулонные. Наличие широкой сети каналов и дешевой оконечной аппаратуры позволило достигнуть не только большой разветвленности телеграфной связи, но и высокой степени ее автоматизации (по тем меркам).

60-х годах 20-го века получил широкое распространение полуавтоматический способ обработки телеграмм методом реперфораторного переприема; в местах переприема ручная обработка телеграмм сводится к переносу от аппарата приема к аппарату передачи перфорированной ленты с напуншированной (нанесенной в виде отверстий) на ней телеграммой. Прием же телеграмм на ленту и дальнейшая передача осуществляются автоматически. Такой метод получил название автоматизированного транзита с отрывом ленты (АТОЛ), он повысил производительность труда телеграфиста в 1,5 раза.

Полную автоматизацию процесса обработки телеграмм дала система прямых соединений (ПС). Телеграмма обрабатывается не поэтапно, от одного телеграфного пункта к другому, а передается непосредственно из отделения связи, где телеграмма подана, в отделение связи, из которого она будет доставлена адресату. Это достигается прямым соединением указанных отделений связи через автоматические коммутационные станции телеграфной сети. Такой способ автоматизации дает наибольший экономический эффект, поскольку производительность труда телеграфиста по сравнению с полуавтоматическим способом повышается не менее чем в 3 раза. Способ автоматизации по системе прямых соединений начал внедряться с конца 60-х годов и в настоящее время является преобладающим на сети общего пользования. Дальнейшее развитие телеграфной связи было направлено не только на совершенствование процессов автоматизации, но и на увеличение разветвленности сети [10].

восьмидесятые годы прошлого столетия была поставлена задача замены всех ленточных телеграфных аппаратов рулонными. Однако, в связи с экономической (и технической) целесообразностью, эта задача не была выполнена в полном объеме - рулонные аппараты используются до сих пор в некоторых отделениях. В 90-х годах, в связи с появлением автоматических систем управления (АСУ), получает развитие орга-связь, представляющая собой комплекс средств связи для передачи данных, т. е. передачи такой информации, которая подлежит обработке электронно-вычислительными



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

машинами (ЭВМ). Эта информация включает производственные, бухгалтерские, финансовые и другие данные, которые после обработки ЭВМ используются при управлении производством в целом. Для повышения оперативности управления обычная скорость телеграфирования 50 Бод считалась уже недостаточной, и были разработаны аппараты для передачи данных со скоростью более 200 Бод.

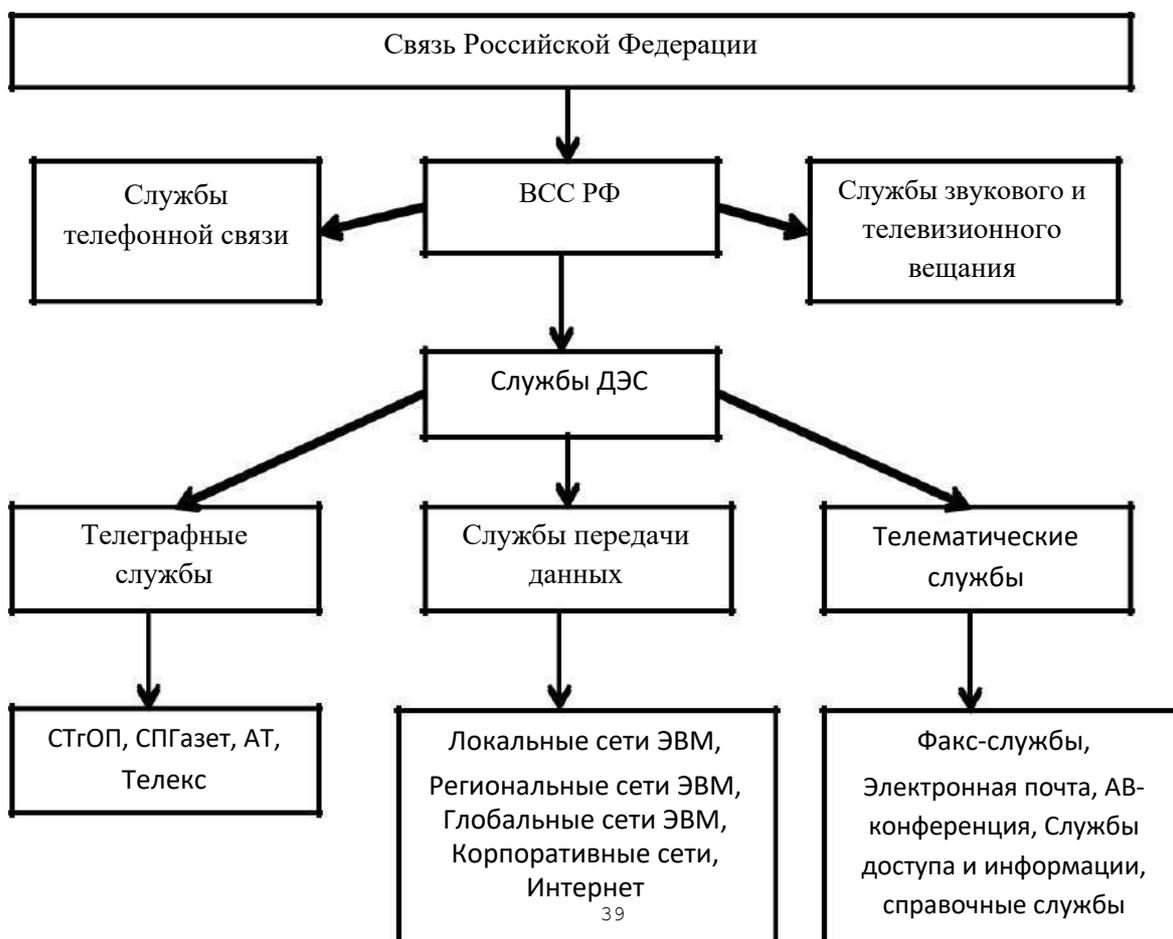
Основой электросвязи Российской Федерации является Взаимовязанная сеть связи (ВСС), обеспечивающая предоставление пользователям услуг электросвязи на всей территории страны [11].

Взаимовязанная сеть связи - это комплекс технологически сопряжённых сетей электросвязи общего пользования и ведомственных сетей с общим централизованным управлением [Федеральный закон от 17 июля 1999 года].

структуры ВСС РФ традиционно входят следующие системы общего пользования таблица 4.1:

- телефонная связь;
- телеграфная связь (ТС);
- факсимильная связь;
- передача газетных полос;
- передача данных;
- распространение программ телевизионного вещания;
- распространение программ звукового вещания.

Таблица 4.1 – структура ВСС РФ





**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

процессе развития ВСС происходит интеграция и образование новых различных служб, в результате чего перечень систем ВСС РФ может измениться.

настоящее время важное место в ВСС занимают службы документальной электросвязи, которые включают в себя системы телеграфной связи, службы передачи данных и телематические службы. Документальной электросвязью называют вид электрической связи, предназначенной для доставки сообщений, представленных в виде документов. На таблице 4.1 схематично изображена классификации систем документальной электросвязи в рамках ВСС РФ [12].

Телеграфная служба - служба доставки сообщений в буквенно-цифровом виде, а также комплекс услуг, предоставляемых населению и предприятиям при использовании системы телеграфной связи.

Предлагаемая лабораторная работа позволит ознакомиться с устройством одного из элементов ТС - Телеграфным терминалом на базе ПЭВМ с модемом «Альфа - Телекс», представляющего собой программно-аппаратный комплекс,

- также получить первичные навыки работы на нём.

Устройство терминального комплекса

Развитие Системы документальной Электросвязи предполагает замену телеграфных аппаратов в отделениях связи на комплект современных терминальных устройств, клиентских служб. Комплекс средств передачи данных по телеграфным каналам связи "Альфа-Телекс", ТУ 6656-042-40713833-98, предназначен для обеспечения обмена информацией в национальных сетях абонентского телеграфирования АТ-50, международного Телекса, в объединенной сети АТ-Телекс и в телеграфной сети общего пользования с коммутацией сообщений (ЦКС). Комплекс может использоваться для работы по четырехпроводным или по двухпроводным линиям связи.

состав комплекса входит:

- модем (адаптер) "Альфа-Телекс", АПК3.080.042;
- блок питания;
- кабель подключения к телеграфной линии;
- кабель подключения к последовательному порту (РС-232С) ПЭВМ;
- программное обеспечение для ПЭВМ "Драйвер АЛЬФА-ТЕЛЕКС комплекса средств передачи данных по телеграфным каналам связи для персональных компьютеров для работы в среде ОО", инструкции по работе с ПЭВМ и модемом Альфа-Телекс - telex.doc, network.doc, (расположена в каталоге Telex3, ПЭВМ);
- программный комплекс WINTELEX.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Адаптер "Альфа-Телекс" эксплуатируется совместно с ПЭВМ типа IBM PC/AT, подключается к ней по последовательному порту H.-232C. Комплекс имеет два режима работы - ручной и автоматический. Прием информации осуществляется в автоматическом режиме. Принятые сообщения накапливаются в памяти адаптера (64 Кбайт) и могут быть считаны ПЭВМ в любое время и распечатаны на принтере. Установленная Flash-память 64 КБайта обеспечивает протоколирование всего обмена информации и процедур вызова и установления соединения. По мере заполнения, информация из Flash-памяти переписывается в отдельный файл на ПК.

Для передачи сообщений в автоматическом режиме, подготовленные в ПЭВМ сообщения загружаются в память адаптера, в очередь для передачи. Далее все процедуры: выборки из очереди в соответствии со временем отправления, установления соединения, контроля автоответа и передачи, адаптер выполняет без участия ПЭВМ. При этом ПЭВМ может быть выключена.

При работе в ручном режиме комплекс "Альфа-Телекс" совместно с ПЭВМ и принтером представляет собой обычный телеграфный аппарат.

Программный комплекс (ПК) WINTELEX предназначен для использования в качестве оконечной установки сетей АТ, ТЕЛЕКС ОП, а также

- качестве автоматизированных рабочих мест узлов связи (УС) и центров коммутации сообщений (ЦКС).

ПК может работать как с модемами семейства Альфа-Телекс (модели 5F, б) по телеграфным каналам передачи данных, так и с виртуальным модемным аулом (ВМП) станции ТК АТ-600 по цифровым каналам передачи данных по протоколу ТСР/ІР.

ПК является многофункциональным рабочим местом. Единый пользовательский интерфейс при работе в различных телеграфных сетях, независимость от аппаратной платформы являются его несомненными достоинствами при переводе телеграфного трафика в цифровые каналы передачи данных. ПК предоставляет удобный текстовый редактор для набора текстов телеграмм, систему архивации и длительного хранения телеграмм, поиск в архивах и много других функций.

Хранение данных в стандартном формате базы данных ACCESS позволяет пользователю писать свои собственные приложения для обработки поступившей информации, формировать статистику и прочее.

При работе в сети АТ/ТЕЛЕКС ПК предоставляет удобный справочник абонентов, многоадресную рассылку, контроль правильности соединения, набор стандартных шаблонов для подготовки телеграмм. Наиболее обширен перечень автоматизировано выполняемых функций при работе в сети ОП. Так



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ПК автоматически обрабатывает кодограммы ЦКС, открывает и закрывает связь, следит за нумерацией, выдает голосовые оповещения о прохождении телеграмм, обрабатывает внекатегорийные телеграммы.

При подготовке телеграмм предоставляется специальный бланк подготовки, и программа автоматически сформирует справочные данные отправителя, подсчитает количество слов, по адресу получателя определит его маршрутный индекс, обеспечит правильный формат телеграммы для сети ОП. ПК обеспечивает и работу в ручном режиме. В таком режиме ПК превращается

- обычный телеграфный терминал и позволяет вести переговоры в режиме On Line.

Описание модема

Органы управления и индикации модема

Модем АЛЬФА-ТЕЛЕКС поставляется заказчику полностью укомплектованным и настроенным. На рисунках 2.1 и 2.2 представлен внешний вид модема.

На задней панели расположены:

- 1 - разъем для подключения блока управления включением ПЭВМ;
- 2- разъем подключения к ПЭВМ;
- 3 - разъем для подключения модема к телеграфной линии;
- В- шнур сетевой.

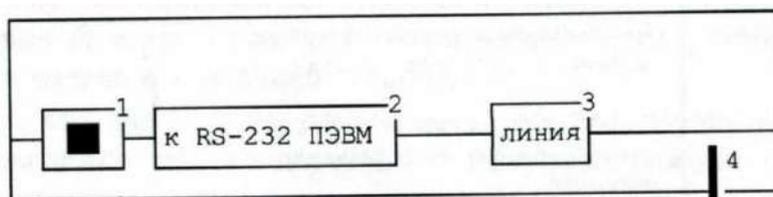


Рисунок 4.1.1- Вид модема со стороны задней панели

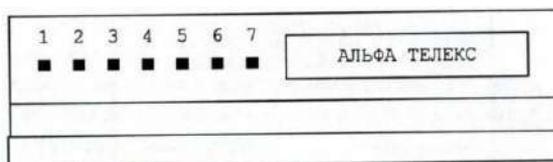


Рисунок 4.1.2 - Вид модема со стороны передней панели

На передней панели модема расположены семь индикаторов 1...7 - светодиоды, свечение которых указывает соответственно на:

- 1 - (СЕТЬ) - наличие питания модема;
- 2 - (ВХ) - наличие в модеме принятых телеграмм;
- 3 - (ИСХ) - наличие в модеме телеграмм, предназначенных для передачи, но еще не переданных;
- 4 - (+ПРМ) - наличие тока положительной полярности приемника (для четырехпроводного режима) или наличие положительной полярности на проводе "б" (для двухпроводного режима);



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

5 - (-ПРМ) - наличие тока отрицательной полярности приемника (для четырехпроводного режима) или наличие положительной полярности на проводе "а" (для двухпроводного режима);

6 - (+ПРД) - наличие тока положительной полярности передатчика (для четырехпроводного режима) или включение передающего "реле" (для двухпроводного режима);

7 - (-ПРД) - наличие тока отрицательной полярности передатчика (для четырехпроводного режима).

Подготовка к включению

6. работе используется ПЭВМ типа IBM/PS/XT/AT под управлением MS DOS версий 6.2 и модем Альфа-Телекс. Запуск программы производится командой TELEX3.

При включении модем должен подать звуковой сигнал запуска - сигнал M2. Перечень сигналов приведен в таблице 4.2.

Таблица 4.2 - Перечень сигналов

СИГНАЛ	Причина выдачи сигнала
1. Серия из 10 коротких гудков	Произведен полный перезапуск модема - при инсталляции
2. До Ре Ми Фа Соль Ля Си До	Горячий рестарт с сохранением данных после восстановления питания
3. Короткий сигнал	Каждые 15 минут- "я жив".
4. Мелодия	Каждые 15 минут при наличии в буфере принятых сообщений - "забери почту"
5. Непрерывная серия сигналов №1	Отказ ОЗУ
6. Звонок	Вас приглашают к диалогу - принята серия ЮЮЮ

Исправный модем подает сигнал №1 только при первом включении после сборки или ремонта. Если он подал сигнал №1 – значит, батарея резервного питания неисправна.

Если модем "молчит", проверьте наличие питания модема: первый слева светодиод должен светиться при нормальной работе блока питания.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Если, кроме первого, горят еще второй и третий светодиоды, то это свидетельствует о не запуске процессора модема.

Режимы работы модема

Модем может находиться в одном из следующих режимов работы:

8. дежурный режим;
9. передача;
10. прием;
11. ручной режим работы.

На любой режим может быть "наложен" режим трассировки. В режиме трассировки модем информирует ЭВМ обо всех происходящих событиях. Используется для контроля над работой модема. Ручной режим работы всегда совмещен с режимом трассировки.

Дежурный режим. В этом режиме модем находится в состоянии ожидания. Вывести модем из этого режима могут следующие события:

- входящий вызов. Модем переходит в режим ПРИЕМ;
- модем получил команду перехода в ручной режим работы;
- просмотрев очередь передачи (периодически) и обнаружив готовую к отправке телеграмму, модем переходит в режим ПЕРЕДАЧА.

Режим приема. Модем ведет прием. Вывести модем из этого режима могут следующие события:

- модем получил сигнал отбой от станции;
- модем получил команду отбой от ЭВМ;
- исчерпана память модема для записи принимаемой телеграммы;
- в течение 20 секунд с начала приема модем не получил запроса «КТО-ТАМ?» (если это задано в настройках модема).

По любому из перечисленных событий модем выдает сигнал отбой (подтверждение отбоя) и переходит в дежурный режим.

Режим передача. Модем ведет передачу. Вывести модем из этого режима могут следующие события:

- модем получил сигнал отбой от станции;
- модем получил команду отбой от ЭВМ;
- несовпадение автоответа или позывного абонента (если задан их контроль);
- встречная передача (если это задано в настройках модема);
- передача успешно завершена.

По любому из перечисленных событий модем выдает сигнал отбой (подтверждение отбоя) и переходит в дежурный режим.

Ручной режим работы. Модем работает под управлением ЭВМ (оператора). Вывести модем из этого режима могут следующие события:
и модем получил сигнал отбой от станции;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

омодем получил команду отбой от ЭВМ;

одлительное время (задано в настройках) нет активности ни одной из сторон.

По любому из перечисленных событий модем вещает сигнал отбой (подтверждение отбоя) и переходит в дежурный режим, оставаясь в режиме трассировки.

Режимы работы терминала

Режим "Телеграмма"

Режим "Телеграмма" предназначен для:

В подготовки текстов;

В ввода текстов с клавиатуры.

Для подготовки текстов предлагается встроенный редактор текстов. Размер редактируемого текста не может превышать 600 строк по 72 символа в строке. Если памяти на ПЭВМ недостаточно, максимальное число строк будет уменьшено.

Переключения режимов ВСТАВКА/ЗАМЕНА производится клавишей INSERT, а вид текущего режима индицируется формой курсора:

В - режим вставки;

_ - режим замены.

Режим отправки присоединенных файлов

Редактор текстов ограничивает размеры отправления. Можно воспользоваться любым другим редактором для подготовки текста. В режиме отправки файла потребуется только задать адрес отправления. Вся информация о приеме/передаче модем хранит в архивах (файлах). Архивы подразделяются на два типа: Личные и Системные.

Существуют четыре системных архива:

файл - QUEUE.TLA очередь передачи. В этот архив заносятся подготовленные телеграммы. Из него производится их загрузка в модем;

файл - SEND.TLA переданные телеграммы. Когда телеграмма будет передана, из архива очереди она будет перенесена в этот архив;

файл - RECEIV.TLA принятые телеграммы. Сюда считывается входящая почта;

файл – HANDLE.TLA в нем хранятся протоколы работы в режиме "Телетайп".

Подготовленная почта заносится в архив очереди на передачу (-QUEUE.TLA) со статусом ПОДГОТОВЛЕНО (READE - 'RD'). После загрузки в память модема телеграмма получает статус ОЧЕРЕДЬ (QUEUE -'QU').

После передачи (отказа) телеграмма удаляется из архива ОЧЕРЕДЬ и переносится в архив ПЕРЕДАНО (SEND.TLA) со статусом ПЕРЕДАНО (SEND



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

-'SN') или ОШИБКА (ERROR – 'ER'). Архивы имеют свою структуру записей, но их можно просмотреть и как текстовые файлы.

Режим "Телетайп"

Режим "Телетайп" предназначен для:

с наблюдения за работой модема в автоматическом режиме;

с ведения диалога с абонентом;

с установления соединения и передачи информации с клавиатуры или передачи заранее подготовленного файла.

Выбрав режим ТЕЛЕТАЙП в основном меню, Вы входите в режим трассировки. Если в это время по линии нет приема/передачи, на экран будет выдано следующее изображение: в верхнем правом углу экрана - текущее время, в верхнем левом углу - Ваш автоответ.

нижней части экрана схематично изображены два телефонных аппарата рисунок 4.2.

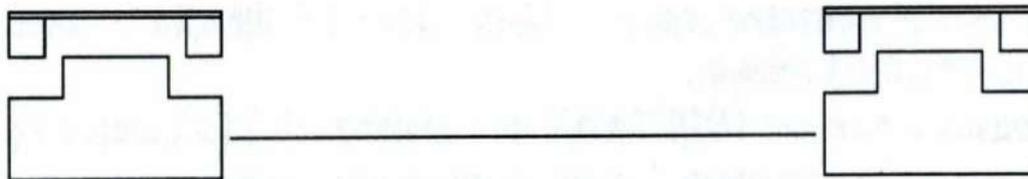


Рисунок 4.2 - Визуализация процесса соединения на мониторе

Слева - это Ваш аппарат, справа Ваш абонент. Пока линия находится в исходном состоянии трубки аппаратов лежат. Если, войдя в режим ТЕЛЕТАЙП, Вы ничего не будете предпринимать, Вы сможете проследить процессы приема/передачи. Подготовьте сообщение к отправке и войдите в режим ТЕЛЕТАЙП. Вы увидите, как модем будет набирать номер, устанавливая связь и передавать сообщение. В режим ТЕЛЕТАЙП можно войти по клавише A11/G1, если Вы находитесь в режимах просмотра архивов или в редакторе текстов. Если Вы вошли в режим, таким образом, то по выходу из него, Вы вернетесь в прерванную функцию. Это очень удобно для входа в режим ДИАЛОГА при вызове Вас абонентом. Если Вы установили в настройках режим "Трассировка на весь экран" режим работы программы изменяется. На экран выводятся только принятые символы, причем используется все поле экрана. Так Вы можете перевести терминал в режим 50 строк или иной. В этом режиме можно использовать либо скроллинг экрана, либо обновление. В режиме 'обновление' курсор перемещается на начало экрана при паузе в приеме, заданной в настройках режима ЦКС - "Пауза в приеме ... считается разделителем сообщений". Такой режим может использоваться для отображения на экран регулярно обновляемой информации, например, на рабочих местах контроля ЭТК-КС. Выход из режима по Escape.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Управление очередью

Загруженное в модем сообщение заносится в очередь передачи. С периодом просмотра очереди (30 сек.), модем просматривает очередь и выбирает из нее сообщение для передачи.

Телеграмма, может быть, простой или срочной, это задается при вводе адреса: время, категория срочности и адрес.

Введем понятие - ЦИКЛ обслуживания очереди. В каждом ЦИКЛЕ будет совершено по одной попытке передачи для каждого срочного отправления и одна попытка отправления одной из несрочных телеграмм. Попытки совершаются с периодом просмотра очереди.

После этого выполняется следующий ЦИКЛ. При этом из простых отправок будет выбрано уже следующее.

Например, в очереди есть следующие сообщения: срочные - 1с 2с 3с и простые - 1п 2п 3п. Если все абоненты заняты - т.е. все запросы остаются необслуженными. Тогда выбор из очереди будет производиться так (тире обозначает начало цикла):

- 1с2с3с 1п- 1с2с3с2п- 1с2с3с3п- 1с2с3с 1п-

Срочность может потребоваться лишь в том случае, когда в модем загружено много телеграмм и вдруг потребовалось действительно срочно что-то отправить. Приведенный пример - 3 срочных и 3 простых - вряд ли может быть взят из реальной жизни.

Находясь в режиме ТЕЛТАЙП по клавише Alt/F1, Вы входите в режим ручного управления очередью. По каждому отправлению, стоящему в очереди, на экран будет выдано сообщение:

- В дата и время последней (очередной для отложенных) попытки передачи;
- В число выполненных попыток передачи;
- В категория срочности;
- В номер абонента;
- В причина неудачи последней попытки передачи.

Клавишами управления курсором Вы можете выбрать интересующее Вас отправление и изменить категорию срочности или снять телеграмму с очереди.

Диалог на автоматическом исходящем соединении

Когда модем ведет передачу в автоматическом режиме, он реагирует только на отбой - PgDown. Нельзя вклиниться в его передачу. Но если надо добавить пару слов, которые забыли записать в текст, нажмите PgUp . Это команда - "передав все, не давай отбой, а перейди в режим диалога". При входе 5.режим на левом, аппарате появится транспарант "Ручной". Далее работать как описано в разделе "Ручное соединение" [18].



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

СПРАВОЧНИК Предназначен для внесения абонентов с которыми часто приходилось обмениваться информацией при помощи терминала Альфа-Телекс.

Режим ТЕЛЭТАЙП

С этом разделе, мы переходим в режим трассировки(связь между двумя и более абонентами).

С данным режиме, происходит обмен информацией между абонентами.

Например: между ПЭВМ и телетайпом производится обмен информацией (телеграмма, сообщение и т.д).

ПЭВМ, нажимаем клавишу F7, для принятия телетайпом кода распознавания адресата. Если посылаем сообщение с телетайпа, мы нажимаем клавишу опознавания (см. описание телетайпа).

Установление соединения: трубки подняты (связь с абонентом). Составление текста телеграммы происходит, в режиме "ТЕЛЕГРАММА". Для установления соединения в ручном режиме,(раздел РУЧНОЕ СОЕДИНЕНИЕ настоящего Руководства). Для завершения работы выходим, нажав клавишу F10.

Меню АРХИВЫ

В режиме АРХИВЫ открыть "окно" и произвести одно из перечисленных действий:

- В просмотреть архивы;
- В создать новый архив;
- В просмотреть статистику принятых сообщений.

Программа ведет статистику работы каждого канала модема.

Фиксируются следующие параметры:

- В общее число принятых\переданных телеграмм и их суммарный объем;
- В максимальный объем принятой\переданной телеграммы;
- В среднесуточные объемы;
- В среднечасовые объемы;
- В установить условие выборки сообщений (происходит последовательная выборка сообщений).

Меню СПРАВОЧНИКИ

справочник заносятся адреса, индексы абонентов с которыми часто надо устанавливать связь.

Порядок проведения работы

Цель работы

Изучение работы ПЭВМ с телеграфным модемом в качестве телеграфного аппарата.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Содержание работы

Изучить состав, назначение, технические характеристики, принцип действия ПЭВМ с модемом Альфа-Телекс в качестве телеграфного терминала.

Произвести включение и проверку работоспособности телеграфного терминала.

Выполнить передачу и прием информации в режимах "Телетайп" и "Телеграмма". Произвести оценку качества принятой информации.

Подготовить отчет о лабораторной работе.

Подготовка терминала к работе и проверка его работоспособности.

Проверить правильность подключения рабочего места.

Порядок включения лабораторной установки:

Включить лабораторную установку;

Войти в программу Альфа-Телекс в каталог <TELEX 3> где найти директорию telex.exe и нажать клавишу <ENTER>;

Восле запуска программы откроется меню:

-Телеграмма;

-Телетайп;

-Архивы;

-Справочники.

Установить режим "ТЕЛЕГРАММА". В режиме "подготовка текста" открывается "окно" редактора для набора любого текста. Заданием адреса заканчивается подготовка телеграммы. Ока заносится в рабочий архив, и программа выходит на уровень основного меню. Для того чтобы убедиться, что почта загружена в модем нужно обратиться к разделу "Режим телетайп-Управление очередью". Перед тем как занести текст телеграммы необходимо заполнить следующие поля:

000 Номер канала отправления телеграммы. Если к ЭВМ подключено несколько модемов, то это порядковый номер модема.

Дата отправления задается в формате ДД.ММ.ГГГГ. Первая попытка соединения будет сделана не ранее указанного срока.

Время отправления в формате ЧЧ:ММ Первая попытка соединения будет сделана не ранее указанного срока.

0- простая 1- срочная

Архивы Набрав текст телеграммы, указывается адрес, например-
64623237

МОДЕМ Ростов-на-Дону, СКФ МТУСИ (телетайп) Модем Ростов-на-Дону, "Альфа" (телекс) В режиме "отправка файла" можно отправить текстовый файл и кодах МТК (код МТК-2 устанавливается при инсталляции программы Альфа-Телекс). Одним из способов является заготовка файлов заранее с именами



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

BLOK_?. NGL, где ?-цифра от 1 до 10. Вставить в текст такой файл можно нажав кл. ALT/1, ALT/2...ALT/0 Число вставок можно увеличить до 20 нажимая кл. SHIFT/ALT/1, SHIFT/ALT/2...SHIFT/ALT/0.

Выход из программы Альфа-Телекс, осуществляется нажатием клавиши "Alt+X" или подведя курсор к "окну" ВЫХОД.

Содержание отчёта

- 3 Описание лабораторной установки.
- 4 Записанные в тетради тексты составленных сообщений.
- 5 Записанный в тетради алгоритм взаимодействия оператора с терминалом (порядок набора, отзывы модема "Альфа-Телекс", используемые функциональные клавиши ПЭВМ; в качестве образца использовать Приложение 1, 2).
- 6 Отчет должен содержать по 4 распечатки переданных и полученных сообщений в режимах "Телетайп" и "Телеграмма", отправленные в адреса СКФ МТУСИ и ЦКС, а также имеющие категории «срочная» и «простая».
- 7 Выводы по работе (среднее время подготовки и набора телеграмм, среднее время прибытия подтверждений на отправления). Мнение обучаемого, действительно-ли целесообразен переход предприятий связи на терминальное оборудование.

Контрольные вопросы

- 9 Назначение и принцип работы модема Альфа-Телекс?
- 10 Перечислите режимы работы терминала.
- 11 Перечислите органы индикации модема и их назначение.
- 12 Проведите операции по переводу комплекса в режим "Телеграмма".
- 13 Проведите операции по переводу комплекса режим отправки присоединенных файлов.
- 14 Проведите операции по переводу комплекса Режим "Телетайп".
- 15 Каким образом осуществляется управление очередью передачи?
- 16 Каким образом осуществляется диалог на автоматическом исходящем соединении?
- 17 Объясните назначение меню АРХИВЫ и СПРАВОЧНИКИ.
- 18 Произведите сравнительный анализ сервиса телетайпов АТ-2000 и АТ-2500 с рабочей станцией оснащенной модемом Альфа-Телекс.

Приложение 1

Для отправки телеграммы необходимо:

- В Включить модем «Альфа-Телекс» в сеть;
- В Включить в сеть аппарат F2000;
- В Включить компьютер нажатием клавиши POWER.



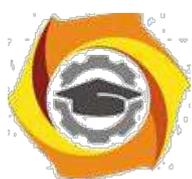
**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

6. На рабочем столе запустить ярлык TELEX.
7. Выбрать меню телеграмма, нажать клавишу ENTER.
8. Выбрать пункт "подготовить текст", ввести текст данный преподавателем.
9. Нажать клавишу F2 - выбрать пункт "Задать адрес и отправить" нажать клавишу ET4ТЕК.
10. Нажать клавишу ESC выбрать пункт меню "Телетайп" нажать клавишу ENTER, нажать клавишу PageUP для поднятия трубки, дождаться поднятия трубки на принимающей стороне, нажать комбинацию A1+17, в появившемся меню архивов выбрать "В ОЧЕРЕДИ" нажать клавишу E14-ТЕК выбрать необходимую телеграмму и нажать ENTER. После успешной передачи нажать клавишу PageDown.

Приложение 2

Передача телеграмм с Телетайпа F-2500

1. Нажимаем кн. «»
2. Нажимаем кн. «» (меню услуг);
В Выбрать цифру 1 (новый текст); [появляется окно с порядковым номером телеграммы]
\quote Нажимаем кн. Рус. Набираем «ЗЦЗЦ», нажимаем пробел, ставим символ «П02» пробел 123432 затем ждем 2 E1ТЕН;
\quote Нажимаем кн. Рус. и набираем: «РОСТОВ на ДОНУ 123432/110 пробел 25 пробел .../... (число и месяц) пробел ... (время без разделительного знака)= 2 знака ENTER;
\quote Нажимаем кн. Рус. и набираем: «РОСТОВ на ДОНУ СКФ МТУСИ=два знака ENTER»;
\quote Нажимаем кн. Рус. ;
\quote «группа №..... проверка выхода=фамилия-» 5 знаков ENTER. НННН 2 знака пробел;
\quote Нажимаем кн. «» ожидаем ответа станции ГА, передаем номер нашей станции «123912» ждем технологическую строку с ЦКС;
\quote Нажимаем кн. «» в меню выбираем п.3 вводим порядковый номер телеграммы ...;
\quote Ожидаем технологическую строку ЦКС;
- 11 .Быстро нажимаем кл. Рус. «▲» (запрос автоответа) получив подтверждение от ЦКС в виде НННН передаем свой символ «» появится 123432 VOLT RU в сеанс закончен.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Лабораторная работа № 5

Эффективное кодирование на примере кода Хаффмана

Цель работы

Изучение принципов эффективного кодирования источников дискретных сообщений.

Домашнее задание

13. Изучить принцип эффективного кодирования алфавита источника дискретных сообщений (ИДС) по методу Хаффмана.

14. Сформировать кодовые комбинации для передачи заданной последовательности знаков алфавита (см. табл.1):

- а) при кодировании алфавита ИДС равномерным кодом;
- б) при кодировании алфавита ИДС кодом Хаффмана.

Студенты заочники выбирают вариант домашнего задания в соответствии с последней цифрой номера студенческого билета.

Таблица 5.1 - таблица вероятностей знаков алфавита

Вариант знак алфавита	1	2	3	4	5	6	7	8	9	0
Z ₁	0,10	0,13	0,04	0,25	0,37	0,12	0,01	0,18	0,06	0,3
Z ₂	0,05	0,17	0,17	0,34	0,13	0,14	0,16	0,14	0,12	0,06
Z ₃	0,17	0,12	0,14	0,16	0,17	0,01	0,13	0,03	0,2	0,18
Z ₄	0,24	0,36	0,26	0,05	0,07	0,23	0,37	0,24	0,09	0,07
Z ₅	0,28	0,15	0,10	0,13	0,10	0,27	0,20	0,05	0,28	0,16
Z ₆	0,02	0,07	0,11	0,07	0,07	0,23	0,10	0,36	0,06	0,23
Z ₇	0,14	-	0,18	-	0,09	-	0,03	-	0,19	-

- Определить значения N_{\max} , $N_{\text{реал}}$ и $n_{\text{сред}}$ для анализируемого варианта.
- Оценить значение $K_{\text{оэ}}$ и $K_{\text{сж}}$.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Краткая теория

Источник дискретных сообщений (ИДС) формирует на выходе счетное число сообщений о состоянии объекта. Такое сообщение в литературе часто называют "знаком", а всю совокупность передаваемых знаков - алфавитом источника.

Например, если в качестве источника сообщения рассматривать текст на русском языке, то в качестве знаков будут выступать буквы, а их совокупность является алфавитом. С учетом пробела алфавит русского языка содержит 33 знака. Английский алфавит содержит 27 знаков [19].

Для передачи каждого знака алфавита по каналу связи необходимо поставить ему в соответствие некоторый определенный сигнал. С этой целью каждому знаку алфавита присваивают определенную комбинацию двоичных символов, то есть осуществляют кодирование источника. Присвоенные комбинации называются кодовыми комбинациями или, по терминологии теории кодирования, кодовыми словами. Если кодовая комбинация состоит из n двоичных символов, то есть является n -разрядной, то n называют длиной кодовой комбинации.

В данной лабораторной работе Вы изучаете кодирование источника, задачей которого является преобразование целесообразным способом заданного набора знаков алфавита в двоичные символы. Целесообразность при этом может означать уменьшение избыточности источника и/или представление сообщений источника в форме пригодной для обработки и передачи.

В технике связи широкое распространение получил метод передачи равномерным кодом, когда количество разрядов в кодовой комбинации постоянно и не зависит от конкретного передаваемого знака. Выбор необходимой длины кодовой комбинации производится в соответствии с объемом алфавита источника. Обозначим N – объем алфавита. Тогда

$$\bullet 2^n$$



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Нетрудно заметить, что равномерный код обладает следующими особенностями:

6. n - разрядная кодовая комбинация может передавать информацию о 2^n знаках исходного алфавита, но по условиям работы источника обычно необходимо передавать меньшее число знаков, а, поэтому, часть кодовых комбинаций оказывается вообще незадействованной;
7. в зависимости от конкретной природы источника вероятность появления каждого из знаков алфавита различна, но при кодировании источника равномерным кодом статистические характеристики источника не учитываются.

Для учета статистических характеристик источника выделяют отдельный класс эффективных методов кодирования.

Если на выходе ИДС, обладающего алфавитом A , знаки a_j появляются независимо с вероятностями p_j , при этом

$$9. \overline{I, K}$$

где K -объем алфавита ИДС, то количество информации, приходящееся на один знак a_j , равно

$$I(a_j) = -\log_2 p_j, \text{ (бит)}$$

Среднее количество информации в битах, приходящееся на один знак на выходе источника, называется энтропией источника и определяется следующим образом

$$H(A) = -\sum_{j=1}^K p_j \log_2 p_j, \text{ (бит)}$$

Известно, что максимальное значение энтропии H_{\max} соответствует случаю равной вероятности появления всех знаков, но в реальных условиях из-за неравновероятности появления знаков всегда выполняется соотношение H_{\max}

• $H_{\text{реал.}}$

теории информации вводится количественная мера избыточности источника, которая показывает, насколько эффективно используются знаки алфавита данного источника. Если избыточность источника D равна нулю, то



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

это означает, что формируемые им сообщения оптимальны с точки зрения количества переносимой ими информации.

$$D = \frac{H_{\max}}{H_{\text{реал}}} - 1 = \frac{H_{\text{реал}}}{H_{\max}}$$

Необходимое число знаков алфавита для передачи заданного количества информации I при отсутствии помех можно определить из соотношения

Так как для реальных источников $H_{\text{реал}} < H_{\max}$, то требуемое число знаков алфавита $K_2 > K_1$, а поэтому избыточность сообщения характеризуется той же мерой избыточности D , что и источник.

$$K_1 = \frac{I}{H_{\max}}$$

Например, известно, что для текста на русском языке при условии равновероятного появления знаков (букв) энтропия составит $H=5$ бит/букву. Исследования показали, что учет всех возможных статистических связей (различная вероятность появления каждой из букв, различная вероятность появления сочетаний букв по две, по три и т.д., вплоть до связи между отдельными словами) снижают реальное значение энтропии до величины $H_{\text{реал}}=1,5$ бит/знак. В этом случае избыточность.

$$D = 1 - \frac{H_{\text{реал}}}{H_{\max}} = 1 - \frac{1,5}{5} = 0,7 \text{ бит/знак.}$$

В ряде случаев сообщение формируется по заранее известным правилам. Так, текст, выполненный на русском языке, подчиняется правилам русской грамматики, а на английском – английской. Знание этих правил позволяет обнаруживать и исправлять ошибки в таких сообщениях, как "наличие избыточности способствует повышению помехоустойчивости сообщений. В этом фразе допущено много ошибок".

Самостоятельно проанализировав текст, Вы можете обнаружить и исправить допущенные грамматические ошибки. Обнаружить ошибки при передаче цифр, используя те же правила, не удастся.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

• технике связи весьма актуальной является задача согласования скорости передачи информации от источника сообщений с пропускной способностью канала связи. Скорость передачи информации определяется энтропией алфавита ИДС и количеством символов, вырабатываемых источником за единицу времени m :

$$C_{ист} = \frac{H}{m}, \text{ (бит / с)}$$

где m - время передачи одного двоичного символа.

Пропускная способность канала характеризуется максимальной энтропией (при отсутствии помех).

$$C_{кан} = H_{max}, \text{ (бит / с)}$$

• том случае, когда $C_{ист} > C_{кан}$, устранение избыточности может обеспечить возможность передачи информации по данному каналу без увеличения его полосы пропускания.

• соответствии с теоремой Шеннона для канала без помех можно увеличить скорость передачи информации посредством преобразования сообщений в статистически независимые и равновероятные символы. Такое преобразование осуществляется кодером источника за счет уменьшения числа символов, которые требуются для передачи одного знака алфавита.

Для этого необходимо наиболее вероятные сообщения передавать наименьшим числом кодовых символов и увеличивать длину кодовой комбинации соответственно с уменьшением вероятности появления кодируемого знака.

Итак, обязательным условием эффективного кодирования является использование неравномерных кодов.

$$n_{сред} = \sum_{i=1}^K p_i n_i$$

Для неравномерных кодовых последовательностей вводят понятие среднего числа двоичных символов на один знак алфавита объемом K : где p_i -



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

вероятность появления i -го знака алфавита, n_i - длина кодовой комбинации, соответствующая передаче i -го знака.

Максимальной эффективностью обладают те неравномерные коды, для которых выполняется равенство средней длины кодовой комбинации и энтропии, т.е.

$$n_{\text{сред}} = H$$

Эффективность неравномерных кодов оценивается коэффициентом относительной эффективности который показывает степень использования статистической избыточности. Для оптимальных кодов $K_{\text{ОЭ}}=1$.

$$K_{\text{ОЭ}} = \frac{H}{n_{\text{сред}}}$$

Отношение среднего числа двоичных символов, приходящихся на один знак алфавита, при кодировании заданного источника неравномерным кодом к длине кодовой комбинации в случае кодирования источника равномерным кодом называется коэффициентом сжатия $K_{\text{сж}}$.

$$K_{\text{сж}} = \frac{n_{\text{сред}}}{n}$$

Для построения оптимального кода необходимо не только учитывать неравно вероятность появления знаков, но и статистические связи между знаками первичного алфавита более высоких порядков [20].

Например, для русского алфавита (с учетом пробела) неравно вероятность появления букв снижает энтропию до значения $H_1=4,35$ бит/знак; дополнительный учет статистики двухбуквенных сочетаний доводит значение энтропии до $H_2=3,52$ бит/знак, а трехбуквенных - до $H_3=3,01$ бит/знак.

Рассмотрим принцип построения эффективного кода, предложенный Хаффманом. Пусть, например, известно, что передаваемый алфавит содержит пять знаков с вероятностями появления:

$$P(a)=0,22 ; P(б)=0,20 ; P(в)=0,26 ; P(г)=0,15 ; P(д)=0,17$$



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Построение кода Хаффмана начинается с упорядочивания указанных знаков по убыванию значений P_i .

Таблица 5.2 – Построение кода Хаффмана

Знаки	Номер вспомогательной операции				
	1	2	3	4	5
	Вероятности				
В	0,26	0,32	0,42	0,58	1,0
А	0,22	0,26	0,32	0,42	
Б	0,20	0,22	0,26	0,42	
Д	0,17	0,20			
Г	0,15				

Определяются два наименее вероятных знака алфавита, которые переходят и новый сжатый алфавит (2), как один знак с вероятностью появления равной сумме вероятностей появления этих двух знаков.

$$P(Д,Г)=P(Д)+P(Г) = 0,15 + 0,17 = 0,32.$$

Вновь проводим систематизацию знаков алфавита по признаку убывания вероятностей. В данном случае первый вспомогательный знак попадает в начало списка. Повторяем процедуру объединения наименее вероятных знаков по новому сжатому алфавиту (2). Теперь объединению подлежат знаки А и Б.

Определяем вероятность второго вспомогательного знака.

$$P(А,Б)=P(А)+P(Б)= 0,22 + 0,20 = 0,42.$$

Вновь производим упорядочивание нового сжатого алфавита (3). Снова выбираем два знака с наименьшими вероятностями, объединяем их и определяем суммарную вероятность.

$$P(В,Д,Г)= 0,26 + 0,32 = 0,58$$

опять повторяем процедуру упорядочивания полученного сжатого алфавита, определения наименее вероятных двух знаков, их объединения и т.д.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Процесс продолжается до тех пор, пока мы не дойдем до единственного вспомогательного знака с вероятностью $P_i = 1,0$.

Для построения кодовой таблицы необходимо проследить путь перехода знака по строкам и столбцам таблицы. Наиболее удобный способ - это построение кодового дерева (рис. 5.1).

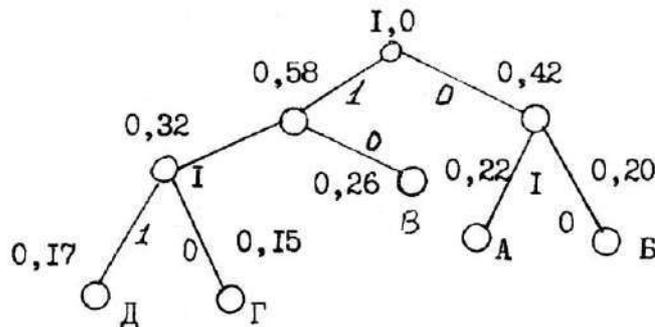


Рис. 1

Рисунок 5.1 - Дерево кодовых слов

Из точки $P_i = 1,0$ направляем две ветви, и той, у которой в соответствии с колонкой 5 таблицы 1 вероятность больше ($P_i = 0,58$), присваиваем символ 1, другой - 0.

Рассматриваем ветвь с символом 0. В соответствии с колонкой 2 из этого узла так же могут быть направлены две ветви. Одной из них снова присваиваем символ 1, а другой - 0. В этой части ствола ветвление закончено.

Возвращаемся к первой ветви с символом 1 (для $P_i = 0,58$) и строим из этого узла ветвление со значением $P_i = 0,32$ ("1") и $P_i = 0,26$ ("0"), и так далее. В результате получено кодовое дерево, двигаясь по ветвям которого легко можно записать кодовую комбинацию для каждого знака.

Составлять кодовую комбинацию начинают от исходной точки ($P_i = 1$) к концу каждой ветви. Для рассматриваемого случая получим:

Б-00, А-01, В-10, Г-110, Д-111.

При таком построении выполняется обязательное требование однозначности декодирования, которое состоит в том, чтобы ни одна комбинация данного кода не совпадала с началом другой, более длинной



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

комбинации. Коды, удовлетворяющие этому условию, называются префиксными кодами. При декодировании последовательности комбинаций префиксного кода определение кода каждого знака производится однозначно. В противном случае, т.е. для комбинаций не префиксного кода характерна неоднозначность декодирования [21].

Пусть, например, некоторый код удовлетворяет требованию префиксности, т.е. знакам алфавита соответствуют кодовые комбинации вида: А-00 В-01 В-101 Г-100.

Составим произвольно комбинацию передаваемых знаков алфавита и соответствующую ей кодовую последовательность:

БАБВГВГЕГААБ.

01000110110010101100000001.()

Эта последовательность декодируется
однозначно: 01 00 01 101 100 101 01 100 00 00 01

БАБВГВБГААБ

Рассмотрим другой случай, когда кодирование ансамбля знаков проведено по кодовой таблице вида
А-00; В-01; В-001; Г-010.

Тогда последовательность кодовых комбинаций того же сообщения будет иметь вид

01000100101000101010000001.

К этому случаю возможны различные варианты декодирования:

01 00 01 001 01 00 01 01 010 00 00 01

Б А БВ Б А БВ Г А А Б

или

010 001 001 010 001 01 010 00 00 01

ГВВГВБГААБ или

01 00 01 001 010 001 01 010 00 00 01

Б А Б В Г В Б Г А АБ



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Возможны и другие варианты. Попробуйте самостоятельно декодировать приведенную последовательность.

Рассмотрим воздействие одиночной, ошибки в принятой кодовой последовательности символов, отмеченной (), удовлетворяющей условию префиксности, т.е. пусть принятая последовательность имеет вид:

0100011010001010110000001.

В этом случае при декодировании получим

01 00 01 101 00 01 01 01 100 00 00 01

БАБВАБББГААБ

Итак, одна ошибка в кодовой комбинации, соответствующей знаку Г, привела к неправильному декодированию нескольких знаков. Такой характер ошибок называют треком ошибок.

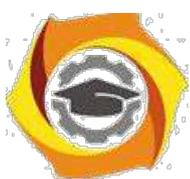
Характерной особенностью систем, использующих методы эффективного кодирования, является обязательное наличие как в передающем, так и в приемном устройстве буферных запоминающих устройств. Эти устройства обеспечивают возможность передачи символов с постоянной скоростью в канал связи при условии их неравномерного поступления на выход кодера.

Описание лабораторной установки

Условная функциональная схема лабораторной установки приведена на рис. 5.2



Рисунок 5.2 -Условная функциональная схема лабораторной установки
Источник сообщения (ИДС) формирует на выходе последовательность знаков из заданного алфавита.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Блок формирования равномерного кода позволяет присвоить каждому знаку алфавита ИДС комбинацию равномерного первичного кода.

Кодер Хаффмана осуществляет формирование кодовых комбинаций методом эффективного кодирования.

Декодеры осуществляют соответствующее декодирование принятых кодовых комбинаций.

Датчик ошибок осуществляет ввод ошибки в последовательность кодовых комбинаций при эффективном кодировании по методу Хаффмана и равномерном кодировании.

Лабораторная работа выполняется на IBM PC. После запуска программы (файл LR 36_108.exe) введите на первой странице свои Ф.И.О. и номер группы. Затем перейдите на страницу 3. Статистика и кодирование - декодирование.

Внешний вид этой страницы показан на рисунке 5.3.

2

Начало | Блок-схема | 3) Статистика и кодирование - декодирование | 4) Декодирование | Задание | Литература | формулы | Конец | Кодовое дерево

Источники
 КОТ МОЛОКО МОЛОКОСОС
 ЕНОТ МОЛОКЕ МОЛОКОСОСОМ
 КОШКА МОЛОКОМ МОЛОКООТСОСОМ

3 Марков_Игорь

длина сообщения L = 12 кол-во знаков N = 10

среднее число бит на один знак для кода
Хаффмена = 3.333 равномерного = 4
коэф. стат. сжатия Kcc = 1.200

Энтропия H = 3.252 Hmax = 3.322 избыточность D = 0.021
Козф. относит. эффективности Koz = 0.975

Кодирование - декодирование равномерным кодом требуется: 48 бит

№	знак	кол-во	вероятн.	равн.код	Хаффмена
1	о	2	0.167	0000	01
2	р	2	0.167	0001	00
3		1	0.083	0010	1111
4	И	1	0.083	0011	1110
5	М	1	0.083	0100	1101
6	а	1	0.083	0101	1100
7	в	1	0.083	0110	1011
8	г	1	0.083	0111	1010
9	к	1	0.083	1000	1001
10	ь	1	0.083	1001	1000

5

Кодирование - декодирование неравномерным кодом требуется: 40 бит

6

ПЕЧАТЬ | Л/Р № 36 16.04.01 15:17:43 Марков Игорь Александрович 028109

Рисунок 5.3 - Статистика и кодирование - декодирование

Ввод сообщения производится в строке, обозначенной цифрой 1 в кружке со стрелкой. После ввода сообщения справа в таблице (2) выводится статистика



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

данного сообщения и кодовые комбинации для равномерного кода и кода Хаффмана. Эта таблица имеет следующую структуру:

- 1 столбец – порядковый номер знака алфавита;
- 2 столбец – знак алфавита;
- 3 столбец – количество данных знаков в сформированном сообщении;
- 4 столбец – вероятность появления данного знака в сообщении;
- 5 столбец – кодовая комбинация для данного знака при равномерном кодировании;
- 6 столбец – кодовая комбинация для данного знака при кодировании кодом Хаффмана.

В окне 3 данного экрана показан результат кодирования - декодирования равномерным кодом. Окно имеет следующую структуру:

- 1 строка – передаваемое сообщение (вход кодера);
- 2 строка – последовательность двоичных символов на выходе кодера;
- 3 строка – позиция искажаемого бита (обозначается □);
- 4 строка – последовательность двоичных символов на выходе канала (вход декодера);
- 5 строка – принятое сообщение (выход декодера).

3. окне 4 данного экрана показан результат кодирования - декодирования кодом Хаффмана. Окно имеет структуру аналогичную окну 3.

Слева от окон 3 и 4 расположены органы управления датчиками ошибок:

Вкл/Откл – кнопка включения/отключения датчика ошибки.

- изменение положение искажённого бита

Если датчик ошибки включен, то позиция, на которой находится искаженный бит, помечается звездочкой □, а если датчик ошибки отключен, то вместо звёздочки появляется точка.

Задание

4. Сформируйте передаваемое сообщение. Сообщение должно отвечать следующим требованиям:
3. включать Вашу фамилию;
4. содержать не более 12 различных символов (объем алфавита $N \leq 12$);
5. иметь длину от 8 до 15 символов;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

4. коэффициент сжатия сформированного сообщения при кодировании полученного алфавита кодом Хаффмана должен получиться не менее 1,2.
6. Выведите на печать содержимое экрана (закладка 3) со статистикой сообщения и процессами кодирования равномерным кодом и кодом Хаффмана и декодирования на приеме в случае отсутствия ошибок, вызванных помехами в канале.
7. Выведите на печать содержимое экрана (закладка 4), иллюстрирующего процесс декодирования при наличии ошибок, вводимых в различные символы, передаваемой двоичной последовательности при использовании равномерного кода и кода Хаффмана.
8. Постройте дерево кодовых слов для обоих случаев кодирования алфавита.

Контрольные вопросы

Принцип формирования кодовых комбинаций при кодировании методом Хаффмана.

Метод определения средней длины кодовой комбинации на знак алфавита и ее оптимальное значение.

3. Сущность требования префиксности эффективных кодов.

Принцип согласования источника сообщения с пропускной способностью канала связи.

9. Количественная оценка эффективности неравномерного кодирования.

Механизм возникновения трека ошибок при декодировании последовательности кодовых комбинаций.

у Принцип декодирования последовательности префиксного кода.

Особенности декодирования последовательности для непrefixного кода.

Содержание отчета

3. Результаты выполнения домашнего задания.
4. Распечатки результатов выполнения работы.
5. Расчетные формулы с записью исходных значений для определения величин N_{\max} , $N_{\text{реал}}$, $n_{\text{сред}}$, $K_{\text{оэ}}$ и $K_{\text{сж}}$ для сообщения, сформированного в результате эксперимента.
6. Дерево кодовых слов для равномерного кода и кода Хаффмана, полученных в результате эксперимента.
7. Выводы.

**Лабораторная работа №6
Исследование линейных искажений**

Цель работы

Изучение методов измерения и оценки характеристик линейных искажений в каналах передачи сетей документальной электросвязи.



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

Домашнее задание

Для интегрирующей RC цепи постройте переходную характеристику $h(t)$ в реакцию на прямоугольный импульс $G(t)$. Длительность импульса возьмите равной ($= RC$ - постоянная времени RC- цепи).

R в кОм возьмите равным количеству букв в Вашем имени,

C в нано фарадах - равным количеству букв в Вашей фамилии.

Описание программы

Лабораторная работа выполняется на компьютере.

Программное обеспечение лабораторной работы можно записать на гибкий диск на кафедре или переписать с сайта кафедры pds-01.boom.ru.

Интерфейс программы содержит несколько страниц:

“Начало” – содержит общие сведения: название работы и т.п.

Содержит кнопки:

‘о программе’ - можно посмотреть номер версии программы, адрес сайта МТУСИ и кафедры.

‘переписать’ - активизирует диалоговое окно, используемое для перезаписи программы, образца отчёта и т.п. на гибкий диск.

‘выход’ - используется для выхода из программы.

“Установки” - содержит различные установки и панель формирования индивидуального задания.

“Читайте” - содержит разделы: цель работы, задание, содержание отчёта и контрольные вопросы.

Раздел ‘задание’ содержит лабораторное задание и методику выполнения каждого пункта [26].



“ $h(t)$ и $G(t)$ ” - содержит схему для наблюдения переходной характеристики и реакции цепи на прямоугольный импульс рисунок 6.1.

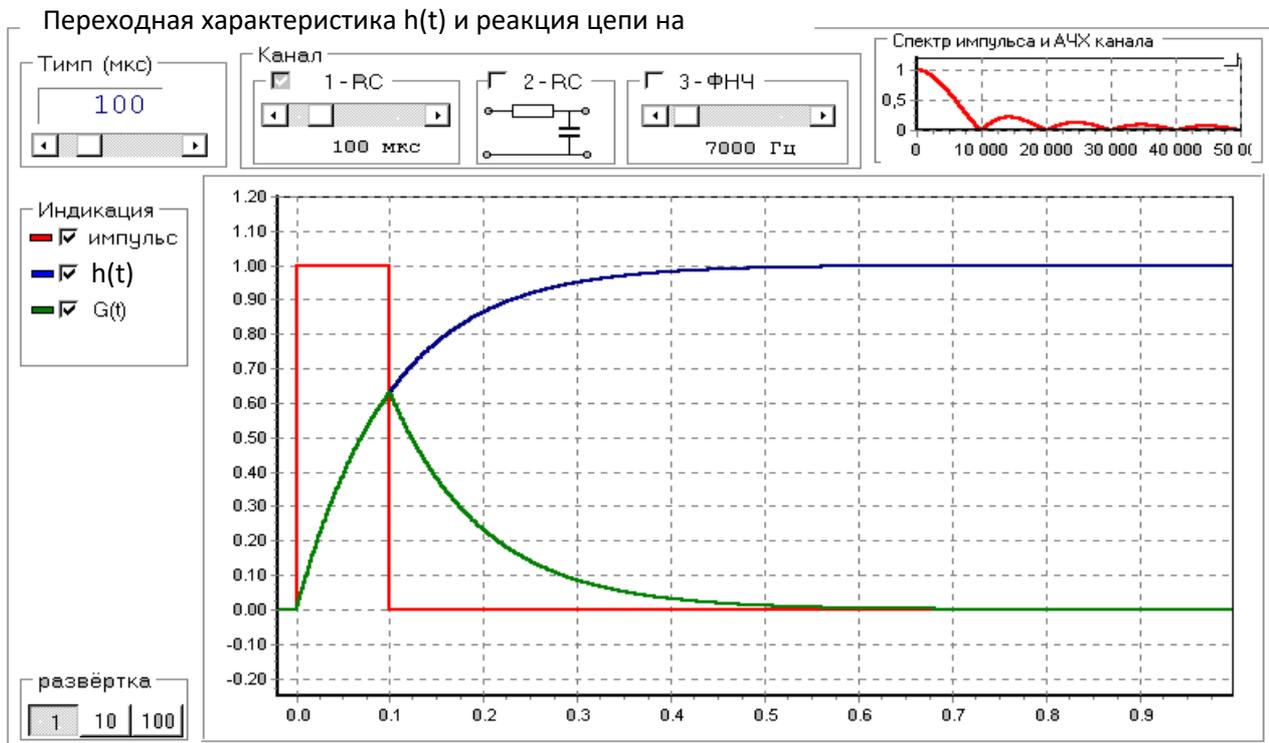


Рисунок 6.1 - Вид страницы “ $h(t)$ и $G(t)$ ”.

2. качестве канала 1 и 2 используются интегрирующие RC цепи .

3. качестве канала 3 используется фильтр нижних частот (ФНЧ).

У каналов 1 и 2 можно менять постоянную RC цепи, а у ФНЧ частоту среза.

Переходная характеристика для интегрирующей RC цепи рассчитывается по формуле:

$$h(t) = 1 - e^{-t/\tau}; RC \quad (1)$$

“Глазковая диаграмма” – содержит схему для наблюдения глазковой диаграммы.

Лабораторное задание

3. Пронаблюдайте и зарисуйте переходную характеристику $h(t)$ и реакцию на импульс $G(t)$ для двух каналов.

2. Исследуйте изменение длительности импульса и постоянной RC цепи



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

на переходную характеристику $h(t)$ и реакцию на импульс $G(t)$.

4. Пронаблюдайте изменение глазковой диаграммы в зависимости от скорости передачи и параметров канала.
5. Оцените межсимвольную интерференцию по глазковой диаграмме.

Методические указания по выполнению работы

Подробную методику по выполнению каждого пункта смотрите в программе на странице: 'Читайте' -> 'Задания' -> 'Методика выполнения'.

Содержание отчета

стр.	содержание
1	Титульный лист.
2	Цель работы и функциональную схему лабораторного макета.
3	Домашнее задание.
4	Переходную характеристику, реакцию цепи на импульс и глазковые диаграммы для первого канала
5	Переходную характеристику, реакцию цепи на импульс и глазковые диаграммы для составного канала (включены первый и второй каналы).
6	Зависимость D-критерия от скорости передачи для двух каналов. Выводы по работе.
7	Краткий ответ на 5 контрольных вопросов (номера вопросов выберите самостоятельно).

- Образец оформления отчёта смотрите в лабораторной работе на странице 'Читайте' -> 'Содержание отчёта'.

*** При выполнении этой лаб. работы распечатки не делаются ***

Контрольные вопросы

- Какие виды помех и искажений действуют в каналах связи?
- Что понимают под линейными искажениями?
- Что такое ФНЧ, ФВЧ, полосовой и режекторный фильтры?
- Как для простейших РС цепей построить АЧХ (качественно)?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

6. Что означают понятия постоянной цепи и частоты среза для RC- цепи.
 7. Как по известному отклику канала на одиночный импульс определить мощность межсимвольной помехи?
 8. Что понимают под условием Найквиста во временной области?
 9. Что понимают под импульсной и переходной характеристиками канала?
 10. Как по известной переходной характеристике получить реакцию на прямоугольный импульс?
 11. Как получается глазковая диаграмма?
 12. Что такое D-критерий?
 13. Из-за чего возникают линейные искажения (межсимвольная интерференция)?
 13. Как по известной схеме RC цепи построить её переходную характеристику?
- В** Поясните разницу между линейными и нелинейными искажениями?
15. Поясните смысл условия отсутствия межсимвольной интерференции в частотной области.
4. Какой эксперимент надо провести чтобы отличить линейную схему от нелинейной?
 5. К какому значению стремиться D-критерий при $B \rightarrow \infty$?
 6. Что такое годограф?
 7. Раскройте понятия: НСНВ, НСДВ, ДСНВ, ДСДВ?
 8. Приведите примеры схем в которых отсутствуют линейные искажения?

Приложение

Краткие сведения из теории линейных искажений

Классификация помех и искажений

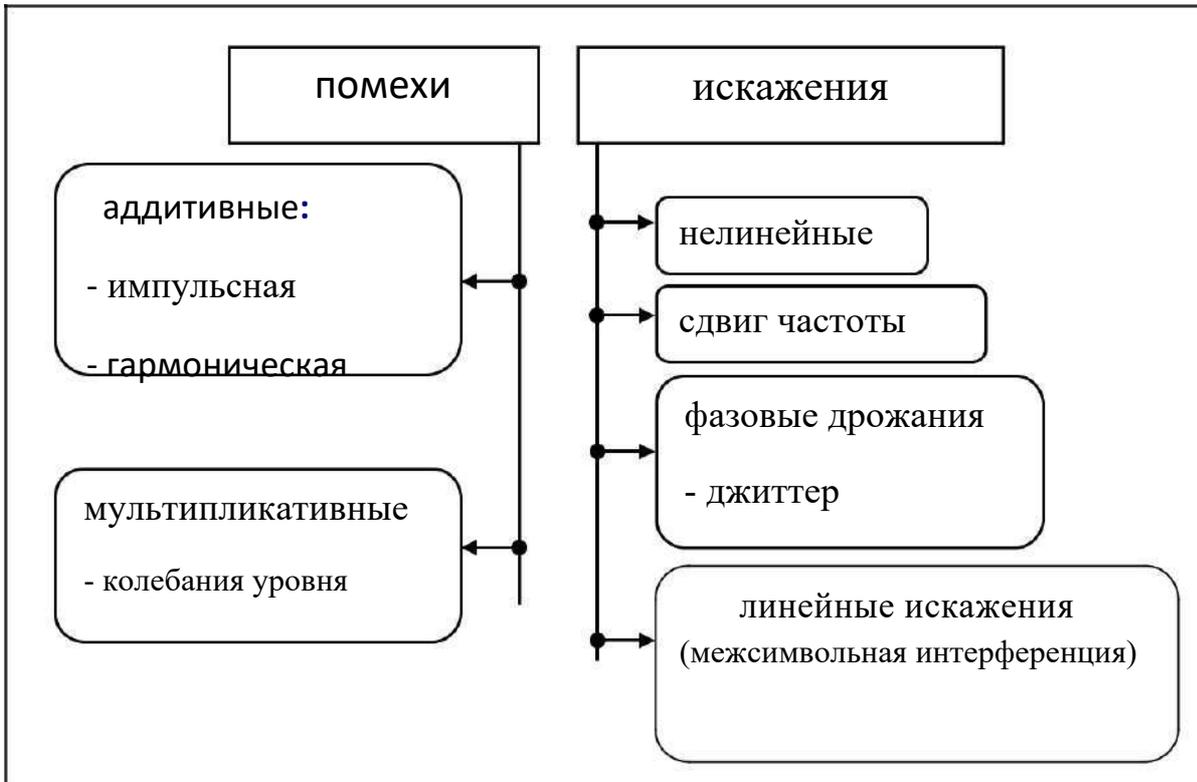
При передаче сигналов по каналам связи они подвергаются действию помех и искажений, что препятствует правильному приёму сигналов.

Под помехами, как правило, понимают дестабилизирующие факторы, имеющие внешнюю природу по отношению к каналу, а под искажением факторы имеющие внутреннюю природу. Наиболее типичные помехи и



искажения можно представить в виде таблицы 6.1.

Таблица 6.1 - Типичные помехи и искажения.



Рассмотрим различные виды помех на примере прослушивания информации записанной на магнитофонную кассету в виде групп цифр (кодограмм). В качестве источника будет выступать запись на кассете, а в качестве канала магнитофон и акустический канал от динамика магнитофона до уха слушателя (приёмника).

Если помеха складывается с сигналом, то её называют аддитивной (*addition* - сложение). Например, во время прослушивания записи кто-то начал стучать молотком по стене – это импульсная помеха, если во дворе монотонно (с одной частотой) завывала сирена – это гармоническая помеха, а если включили пылесос или над домом пролетел реактивный самолёт - то это типичная флуктуационная (шумовая) помеха.

Если помеха перемножается с сигналом, то её называют мультипликативной (*multiplication* - перемножение). Например, в магнитофоне плохо работает регулятор громкости, что приводит к случайным изменениям громкости сигнала. Если громкость сигнала изменяется в небольших пределах, это



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

называют колебаниями уровня.

Если сигнал исчезает полностью, это перерыв связи.

Если в динамике вместо естественного голоса слышен хрипящий (дребезжащий) голос – значит имеют место сильные нелинейные искажения.

Если скорость протяжки ленты отличается от скорости при записи, такие искажения являются сдвигом частоты. При сдвиге частоты вверх женский голос будет восприниматься как голос ребёнка, а при сдвиге вниз - как мужской.

Если скорость протяжки ленты отклоняется от номинального значения в обе стороны, оставаясь в среднем равной номинальному значению, то имеют место фазовые дрожания.

Если фазовые дрожания происходят быстро, то их называют фазовым джиттером (*jitter*- дрожание), а если медленно - вандером (*wander*- блуждать, бродить).

При джиттере голос человека будет похож на голос Буратино, а при вандере женский голос периодически будет переходить, то в голос ребёнка, то в мужской голос.

При помощи ручек регулировки тембра можно изменить АЧХ (Амплитудно Частотную Характеристику - соотношение уровней различных частот), то есть вносить линейные искажения или уменьшать (компенсировать) их.

Линейные искажения, которые вызывает процесс записи и воспроизведения на магнитофоне слишком малы, чтобы вызвать ошибки при приёме (прослушивании) кодограмм.

Разместим магнитофон на лестничной клетке многоэтажного дома, а слушать будем несколькими этажами выше (ниже). При таком размещении источника (магнитофона) и приёмника (слушателя) за счёт сильных переотражений звука от стен и различных полостей (бетонных стен и лестничных клеток) возникнет очень сильное искажение АЧХ.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

‘На слух’ это проявится в том, что слоги речи начнут ‘наползать’ друг на друга, что затруднит безошибочное восприятие кодограмм.

Если скорость чтения кодограмм повысить или увеличить количество этажей между источником и приёмником, то вероятность ошибки при приёме увеличится, а начиная с некоторого значения правильный приём станет невозможным.

Вместо отдельных цифр слушатель будет слышать сплошное бубнение, из которого невозможно выделить отдельные цифры.

Явление, при котором соседние слоги, цифры, сигналы и т.п. накладываются, друг на друга получило название межсимвольной интерференции (МСИ).

Межсимвольная интерференция, как правило, затрудняет правильный приём сигнала.

Линейные искажения в непрерывных каналах связи.

Под линейными искажениями в непрерывных каналах связи (НКС) понимают искажение формы сигнала, вызванное изменением соотношения амплитуд и фаз спектральных составляющих при передаче сигнала через канал связи.

Связь сигнала на выходе канала $s(t)$ с сигналом на его входе $x(t)$ может быть

$$s(t) = x(t) * g(t) = \int_{-\infty}^{\infty} x(\tau) g(t - \tau) d\tau \quad (2)$$

представлена операцией свёртки:

где $g(t)$ – импульсная характеристика канала – отклик канала на дельта-функцию $\delta(t)$:

$$g(t) = \int_{-\infty}^{\infty} \delta(\tau) g(t - \tau) d\tau = g(t) \quad (3)$$

Спектр сигнала на выходе канала $S(j\omega)$ может быть получен умножением спектра входного сигнала $X(j\omega)$ на коэффициент передачи канала $K(j\omega)$:

$$S(j\omega) = X(j\omega) K(j\omega) \quad (4)$$



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Коэффициент передачи канала $K(j\omega)$ однозначно связан с импульсной характеристикой $g(t)$ преобразованием Фурье:

Если представить комплексную функцию $K(j\omega)$ в показательной форме

$$K(j\omega) = \int_0^{\infty} g(t)e^{-j\omega t} dt \quad (5)$$

$$K(j\omega) = |K(j\omega)|e^{j\phi(\omega)},$$

то $|K(j\omega)|$ будет амплитудно-частотной характеристикой (АЧХ) канала, а $\phi(\omega)$ – фазочастотной характеристикой (ФЧХ) канала.

Таким образом, линейные искажения в НКС могут быть полностью описаны и заданы его АЧХ и ФЧХ или импульсной характеристикой.

На практике линейные искажения в НКС чаще всего нормируются требованиями к их частотным характеристикам, что обусловлено сравнительной простотой их описания и измерения.

Однако в зависимости от конкретной решаемой задачи могут существовать и другие методы представления и оценки линейных искажений.

Например, при передаче по каналам сигналов в виде прямоугольных импульсов полезным оказывается понятие переходной характеристики канала $h(t)$, которая является откликом канала на ступенчатую функцию $p(t)$ (сигма-функция):

Прямоугольный импульс $p(t)$ длительностью τ и можно представить в виде разности двух ступенчатых функций, отстоящие на время τ

$$p(t) = \begin{cases} 1 & \text{при } t \geq 0; \\ 0 & \text{при } t < 0; \end{cases} \quad (6)$$

$$p(t) = p(t) - p(t - \tau) \quad (7)$$

с соответствии с правилом суперпозиции для линейных цепей отклик канала на такой импульс будет равен разности переходных характеристик

$$G(t) = h(t) - h(t - \tau) \quad (8)$$

На рисунке 6.2 иллюстрируется связь между $h(t)$ и $G(t)$

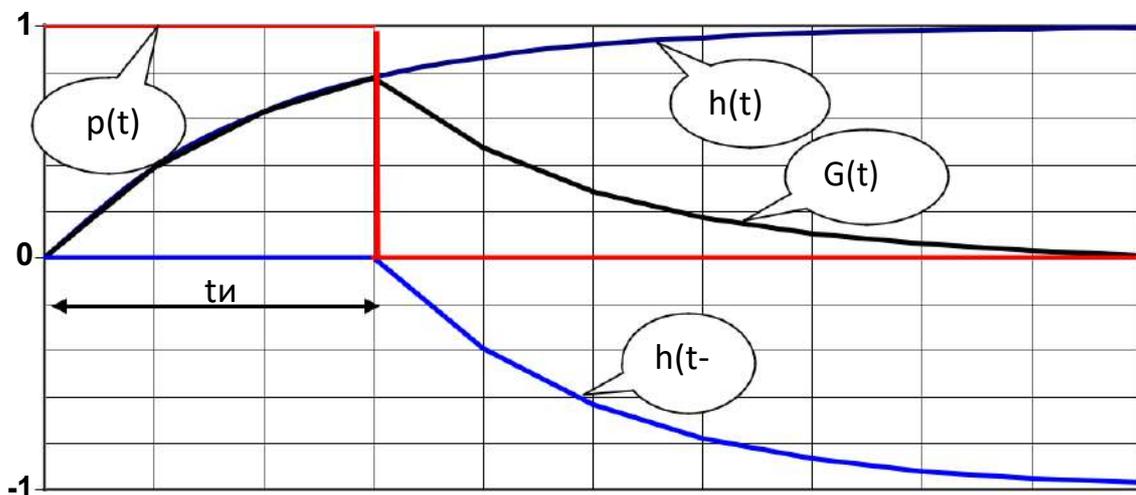


Рисунок 6.2 - Связь между $h(t)$ и $G(t)$.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Из рисунка можно сделать три вывода:

- # за время действия импульса на входе канала переходной процесс не успевает достичь максимального значения;
- # чем меньше длительность импульса на входе канала, тем до меньшего значения успевает дорасти напряжение на выходе канала;
- # теоретически переходной процесс длится бесконечно долго, следовательно, будет влиять на форму отклика канала на последующие импульсы.

Для количественной оценки линейных искажений сигналов в системах документальной электросвязи вводится понятие “межсимвольная помеха” (МСП).

Если длительность отклика превышает период следования импульсов, то соседние отклики накладываются друг на друга – возникает межсимвольная интерференция (МСИ).

Поэлементный приёмник по результатам наблюдения реализации сигнала на выходе канала на каждом единичном интервале выносит решение о переданном единичном элементе сигнала (ЕЭС). Очевидно, что в условиях МСИ реализация отклика, по которому принимается решение, будет суммой переходных процессов нескольких интерферирующих сигналов. При принятии решения об i -м элементе сигнала переходные процессы от всех других ЕЭС будут выступать как помеха, искажающая отклик. Такая помеха называется межсимвольной помехой (МСП) [22].

МСП – это случайный процесс, действующий на i -м интервале обработки сигнала в приёмнике и образованный суммой переходных процессов от тех ЕЭС, решение по которым принимается в другие интервалы времени.

Условие Найквиста

При использовании для передачи в каналах с межсимвольной интерференцией сигналов с определёнными свойствами можно реализовать простой метод поэлементного приёма с однократным отсчётом .

Форма отклика таких сигналов должна отвечать условию Найквиста во временной области:

$$S_0 \quad \text{при} \quad k = 0;$$



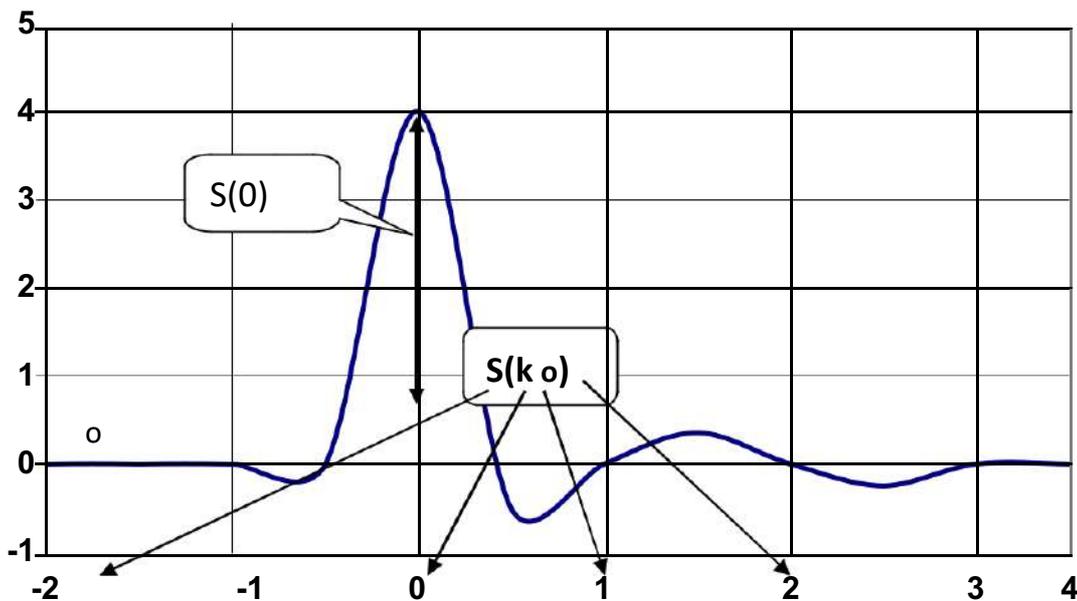
**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

$$S(k\omega) = 0 \text{ при } k \neq 0; \quad (9)$$

$$K(j\omega) = \frac{2m}{\omega} \quad (10)$$

Условие требует, чтобы все отклики обращались в ноль во всех точках, отстоящих от момента взятия отсчёта на интервалы времени кратные единичному интервалу ω . Возможная форма отклика, отвечающая условию Найквиста показана на рисунке 6.3.

Рисунок 6.3 - Отклик канала, удовлетворяющий условию Найквиста.



Требование для выполнения условия Найквиста в частотной области можно записать в виде:

Полученное равенство является необходимым и достаточным условием отсутствия межсимвольной интерференции в канале с частотной характеристикой $K(j\omega)$.

Данное равенство можно интерпретировать следующим образом:

сумма частотных сдвигов частотной характеристики канала вверх и вниз по оси частот на величины $2mB$, где $B = 1/\omega$ – скорость модуляции, равна постоянной величине.

Выбор частотной характеристики канала, удовлетворяющий этому условию



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

не является единственным. Нетрудно проверить, что прямоугольная, треугольная и т.п. частотные характеристики каналов удовлетворяют условию отсутствия межсимвольной интерференции.

Следует иметь в виду, что условие отсутствия межсимвольной интерференции в частотной области жёстко связано со скоростью модуляции B и шириной полосы пропускания низкочастотного канала $F_{нк} = F_{гп}$. Здесь $F_{гп}$ – граничная частота полосы пропускания канала (частота среза). Эта связь определяется соотношением:

$$B \leq 2 F_{нк}. \quad (11)$$

Другими словами равенство (10) может быть удовлетворено, если скорость модуляции B численно не превышает удвоенной граничной частоты пропускания низкочастотного канала.

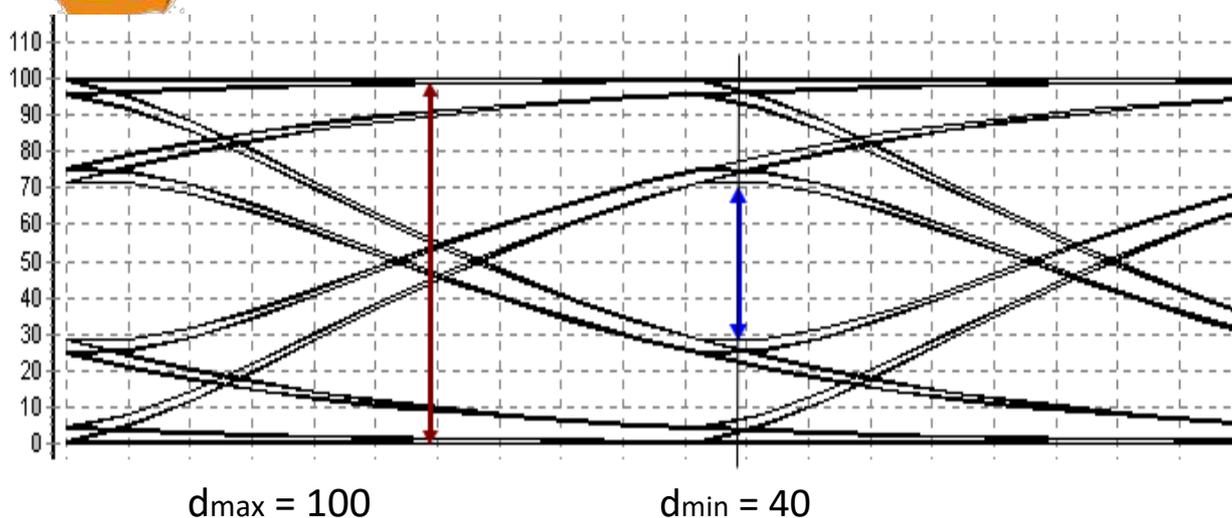
Глазковая диаграмма

Глазковую диаграмму наблюдают при передаче по каналу случайной последовательности импульсов. Если совместить изображение всех возможных реализаций (траекторий) откликов на одном и том же графике, то получим глазковую диаграмму (ГД).

Изображение ГД можно получить с помощью осциллографа, если установить развёртку длительностью примерно равной двум единичным элементам и наблюдать сигнал на выходе канала. Развёртку осциллографа надо засинхронизировать от тактовой частоты передачи. В этом случае на экране будет высвечиваться сразу несколько траекторий сигнала, которые образуют глазковую диаграмму. По глазковой диаграмме можно судить об искажениях сигнала из-за воздействия МСП.

Для оценки величины МСП по глазковой диаграмме определяют d_{\max} и d_{\min} и вычисляют D-критерий (d_{\max} – максимальный раскрыв глаза, d_{\min} – минимальный раскрыв глаза, смотрите рисунок 7.4)

$$D = \frac{d_{\max} - d_{\min}}{d_{\max} + d_{\min}} \quad (12)$$



t_0 - момент принятия решения

Рисунок 6.4 - Возможная реализация глазковой диаграммы.

Минимальный раскрыв глаза определяет границу, внутри которой не может существовать траекторий сигнала при любой входной последовательности.

Если $d_{min} > 0$, то траектории, которые соответствуют передаче единицы, в момент времени t_0 (момент принятия решения) будут проходить выше траекторий, которые соответствуют передаче нуля и на приёме можно производить безошибочный приём.

Если $d_{min} = 0$ (глаз будет закрыт), то приём некоторых элементов сигнала (ЕЭС) будет происходить с ошибками.

Глазковая диаграмма получила своё название из-за того, что при изменении скорости передачи или параметров канала d_{min} начинает изменять своё значение и это напоминает открывание-закрывание глаза.

При изображении глазковой диаграммы показывают лишь внешние и внутренние траектории. Желательно показать два глаза, чтобы можно было оценить запас сигнала по фазе.

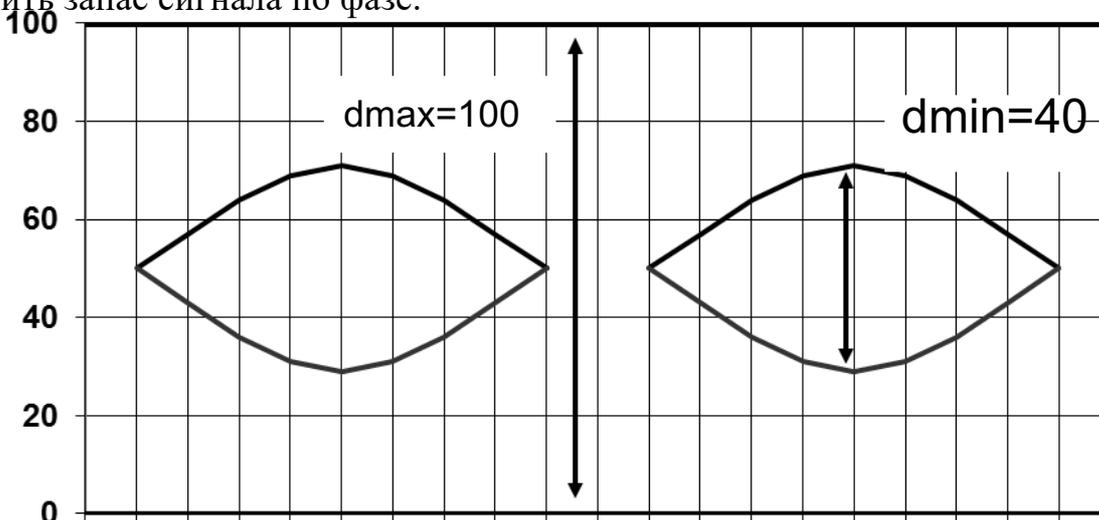


Рисунок 6.5 - Пример изображения глазковой диаграммы.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

По глазковой диаграмме измеряют значение d_{max} , d_{min} и вычисляют величину D-критерия. На глазковой диаграмме надо указать скорость модуляции, на которой она была зарисована.

Проведя аналогичные измерения на других скоростях модуляции можно построить график зависимости D-критерия от скорости модуляции

$$D = f(B). \quad B = 10 \text{ кБод}$$

Различие между линейными и нелинейными искажениями

Линейные искажения возникают в линейных цепях. Линейными называют такие цепи, у которых реакция пропорциональна воздействию. К числу линейных электрических цепей относятся цепи составленные из резисторов, конденсаторов и катушек индуктивностей.

Для линейных цепей справедлив принцип наложения (суперпозиции).

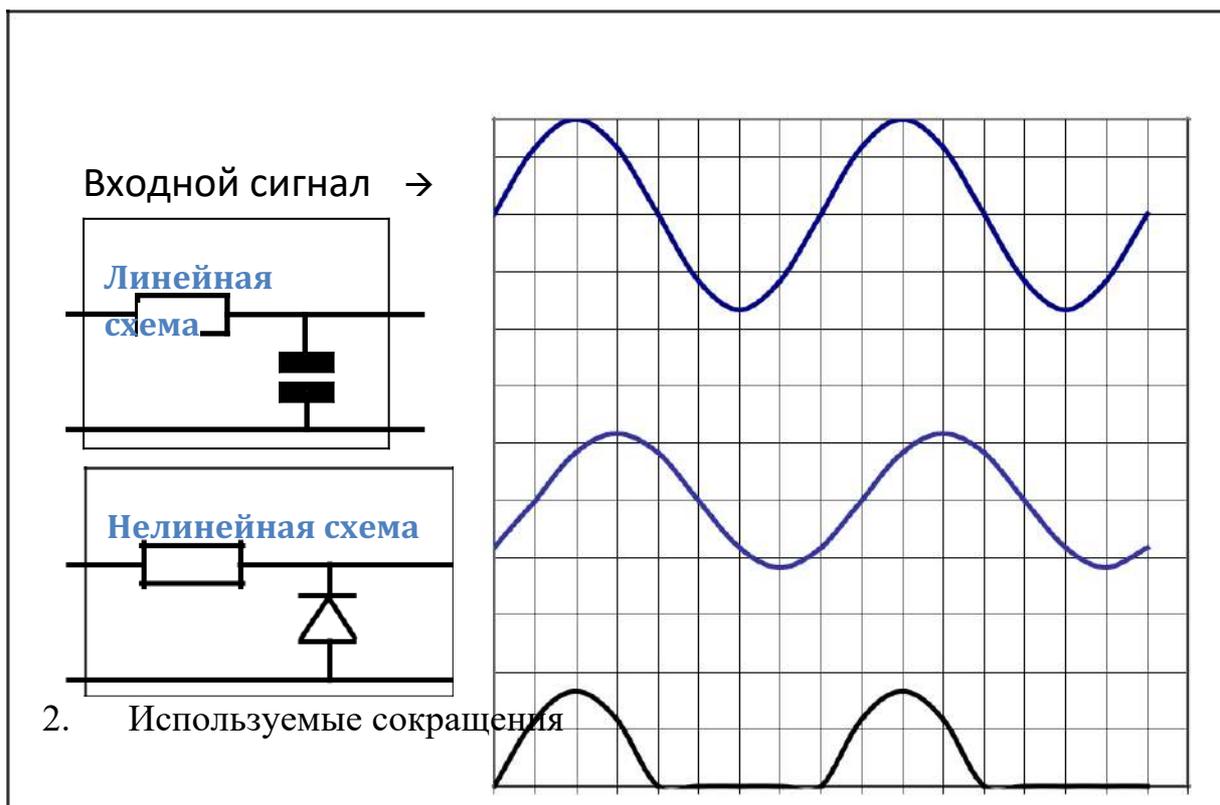
Он формулируется так:

5. Реакция линейной цепи на сумму воздействий равна сумме реакций на каждое воздействие в отдельности.

- нелинейных цепях принцип наложения не действует.

Отличить линейную цепь (схему) от нелинейной можно следующим образом. Надо подать на вход цепи синусоидальное напряжение.

Если цепь линейная, то на выходе будет действовать тоже синусоидальное напряжение, но с другой фазой и амплитудой. Если цепь нелинейная (в ней возникают нелинейные искажения), то форма сигнала на выходе будет отлична от синусоидальной (в спектре сигнала появятся гармоники с частотами кратными исходной частоте).





**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- НКС - непрерывный канал связи;
- МСИ - межсимвольная интерференция;
- МСП - межсимвольная помеха;
- $g(t)$ - импульсная характеристика;
- $\delta(t)$ - дельта-функция;
- $h(t)$ - переходная характеристика;
- $p(t)$ - прямоугольный импульс;
- $G(t)$ - реакция цепи на прямоугольный импульс;
- $K(j\omega)$ - комплексный коэффициент передачи канала;
- $K(\omega)$ - амплитудно-частотная характеристика (АЧХ);
- $\phi(\omega)$ - фазочастотная характеристика (ФЧХ) канала;
- V - скорость модуляции;
- D - D-критерий;
- ЕЭС - единичный элемент сигнала;
- ГД- глазковая диаграмма;
- d_{\min} - минимальный раскрыв глаза;
- d_{\max} - максимальный раскрыв глаза.

Лабораторная работа №7

Исследование свойств циклических кодов

Цель работы

1. Изучить основные принципы помехоустойчивого кодирования.
2. Изучить правила построения циклических кодов.
3. Исследовать обнаруживающие и исправляющие свойства циклических кодов.
4. Познакомиться с принципом построения кодирующих и декодирующих устройств циклических кодов.

Индивидуальное задание

6. лабораторной работе изучаются свойства циклических кодов на примере делимого линейного циклического кода $(7,4)$ с образующим полиномом $g(x) = x^3 + x + 1$. Кодовое расстояние этого кода $d = 3$.

Код может быть использован в режиме исправления ошибок или в режиме обнаружения ошибок.

1. Найдите все кодовые слова заданного кода.
2. Определите характеристики заданного кода в режиме исправления ошибок.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

3. Определите кратность t гарантированно исправляемых кодом ошибок.
4. Найдите число различных векторов ошибок, которые код может исправить.
5. При условии, что кодом в первую очередь исправляются ошибки наименьшей кратности, приведите вектора ошибок, которые могут быть исправлены заданным кодом [27].
6. Для одного из векторов ошибок, исправляемых кодом, найдите соответствующий этому вектору синдром. Найдите несколько из возможных векторов ошибок, при декодировании которых получается тот же синдром, и, следовательно, происходит ошибочное декодирование [23].
7. При условии, что кодом в первую очередь исправляются ошибки наименьшей кратности, рассчитайте для заданного кода вероятность РОШ ошибочного декодирования, если канал является дискретным симметричным каналом без памяти (ДСК) с вероятностью ошибки в двоичном символе p_0 .

Численное значение p_0 возьмите из программы лабораторной работы в разделе 'Индивидуальное задание'.

Определите возможности заданного кода в режиме обнаружения ошибок:

3. Определите кратность гарантированно обнаруживаемых кодом ошибок.
4. Найдите вектора ошибок, которые не могут быть обнаружены заданным кодом.
5. Рассчитайте вероятность $P_{н.о.}$ не обнаружения ошибок заданным кодом в ДСК с вероятностью ошибки в символе p_0 .

Описание программы

Лабораторная работа выполняется на IBM PC совместимом компьютере.

Программное обеспечение лабораторной работы можно записать на кафедре или с сайта кафедры www.pds-01.boom.ru.

Программа содержит несколько страниц:

“Начало” – содержит общие сведения: название работы и т.п.

Содержит кнопки:

‘о программе’ - можно посмотреть номер версии программы, адрес сайта МТУСИ и кафедры.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

‘переписать’ - активизирует диалоговое окно, используемое для перезаписи программы, образца отчёта и т.п. на гибкий диск.

‘выход’ - используется для выхода из программы.

“Установки” - содержит различные установки и панель формирования индивидуального задания.

!!! На распечатках, помещаемых в отчёт должны быть Фамилия И.О. и номер группы исполнителя. Для этого необходимо в начале выполнения работы ввести их в требуемые строки на панели ‘Формирование индивидуального задания’.

“Читайте” - содержит разделы: цель работы, задание, содержание отчёта и контрольные вопросы. Раздел ‘задание’ содержит лабораторное задание и методику выполнения каждого пункта.

“Схема” - содержит схему, иллюстрирующую процесс кодирования и декодирования. На этой странице имеется возможность вводить различные кодовые комбинации, вектора ошибок и наблюдать процесс кодирования и декодирования. Также эта страница содержит таблицу соответствия векторов ошибок и соответствующим им синдромов.

“Таблица” – содержит таблицу разрешённых кодовых комбинаций используемого кода, таблицу расстояний Хемминга для всех пар кодовых комбинаций и таблицу весов кодовых комбинаций.

Имеется возможность изменять образующий полином, используемый для кодирования. Обратите внимание, что только полиномы 1011 и 1101 обеспечивают коду (7,4) требуемые помехоустойчивые свойства ($d_{\min}=3$).

Выполнять работу рекомендуется при полиноме 1011 (установлен по умолчанию). Это связано с тем, что тестовые вопросы для защиты этой лабораторной работы составлены для полинома 1011.

“Формулы” – содержит некоторые формулы, используемые в лабораторий работе.

“Статистика” – демонстрирует использование различных способов помехоустойчивого кодирования для обнаружения и исправления ошибок.

Для распечатки требуемой страницы необходимо нажать клавишу



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

‘Печать’ расположенную в левом нижнем углу.

Если распечатка не произойдёт (замечен такой дефект на некоторых типах принтеров), то можно распечатать файл PRINT_13_XX.BMP , который создаётся при каждом нажатии на кнопку ‘Печать’.

Распечатку этого файла можно произвести, например, при помощи редактора Word.

Описание функциональной схемы

Информационное слово длиной четыре разряда ($k=4$) задается нажатием кнопок на панели Источник.

Кодер формирует кодовую комбинацию делимого циклического кода (7,4) путём добавления к четырём информационным разрядам трёх проверочных разрядов. Проверочные разряды образуются в результате умножения на X^3 , и деления получившейся комбинации на образующий полином 1011 ($g(x)=X^3+X+1$).

Остаток от деления будет представлять проверочные разряды, которые приписываются справа от информационных [24].

Кодер выдает кодовое слово V длиной $n = 7$, соответствующее информационному слову M длиной $k = 4$, поданному на вход кодера.

Семиразрядная кодовая комбинация (вектор V) с выхода кодера подаётся на вход канала. В канале передаваемая кодовая комбинация складывается с семиразрядным вектором ошибки E . Сложение производится поразрядно по модулю 2 ($0 \oplus 0=0, 0 \oplus 1=1, 1 \oplus 0=1, 1 \oplus 1=0$).

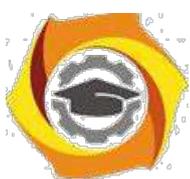
Например, если комбинация на входе канала равна $\overline{1011000}$, а вектор ошибки равен 0010001 , то комбинация на выходе канала будет равна 1001001 .

Таким образом, единица в i -том разряде в векторе ошибки приводит к ошибке (инверсии) i -того разряда в кодовой комбинации на выходе канала.

Вектор ошибки задаётся на панели Источник ошибок. Там же имеются кнопки для циклического сдвига вектора ошибки и для его инверсии.

5. выхода канала принятая кодовая комбинация подаются на входы декодеров.

Декодер 1 работает в режиме исправления ошибок. Если ошибок нет или вектор ошибок принадлежит к векторам ошибок, исправляемых кодом, то



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

сигнал на выходе декодера 1, совпадает с сигналом на входе кодера (выходе источника).

Если сигнал на выходе декодера 1 не совпадает с входным сигналом, поступающим на вход кодера, значит, произошло ошибочное декодирование.

Это соответствует появлению вектора ошибок, который код, используемый системе, исправить не может. Код (7,4) исправляет все одиночные ошибки.

Исправление одиночной ошибки производится следующим образом:

6. принятая 7-ми разрядная комбинация делится на образующий полином;
 7. если остаток от деления (синдром ошибки) не равен нулю, то по таблице соответствия 'вектор ошибки – синдром' определяется вектор ошибки, которому соответствует данный синдром;
- принятая 7-ми разрядная комбинация складывается по модулю 2 с найденным вектором ошибки.

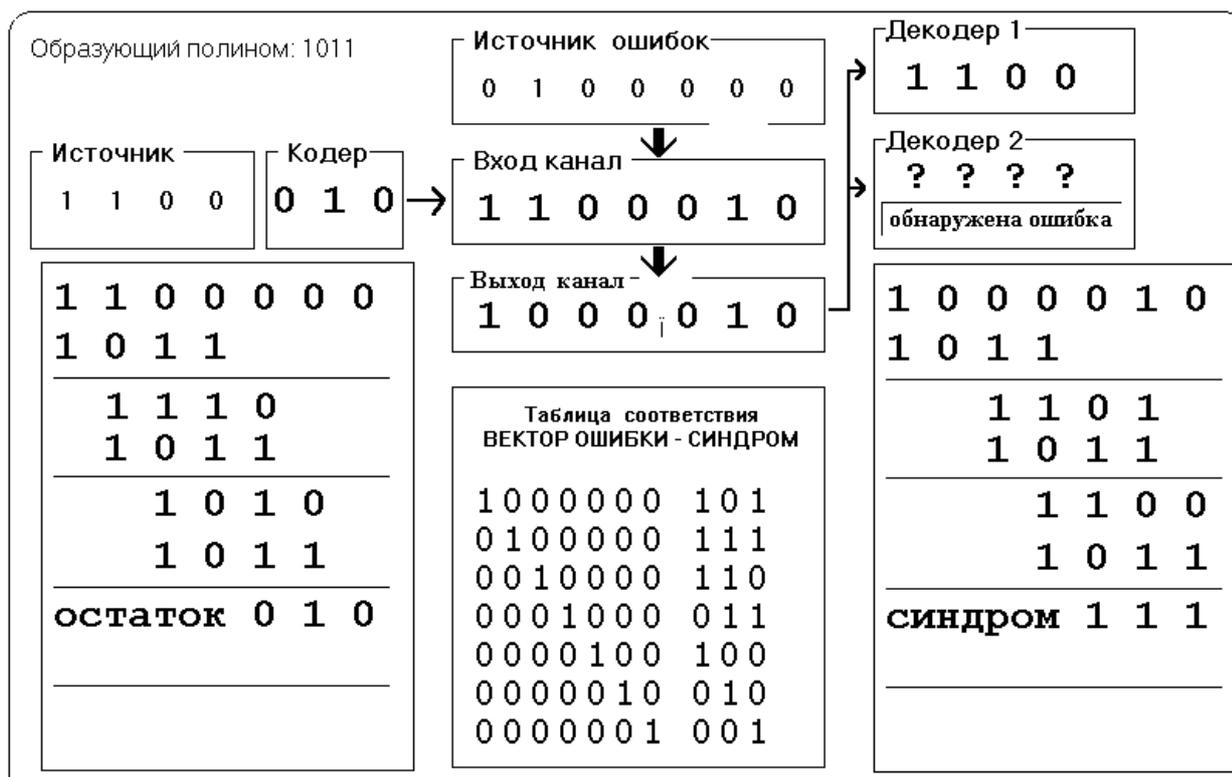


Рисунок 7.1 - Функциональная схема лабораторной установки

Декодер 2 работает в режиме обнаружения ошибок.

Если декодером 2 обнаружена ошибка, то декодер формирует сигнал 'обнаружена ошибка' кодовая комбинация при этом не выдаётся (высвечивается ? ? ? ?). Если вектор ошибок принадлежит к не



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

обнаруживаемым заданным кодом векторам ошибок, сигнал на выходе декодера 2 будет отличным от сигнала, поданного на вход кодера.

Декодер 2 не может обнаружить вектора ошибок, которые совпадают с разрешенными кодовыми комбинациями. Это следует из свойства линейного кода: ‘сумма двух кодовых слов является тоже кодовым словом’.

Лабораторное задание

1. Проверьте работу кодера.
2. Проверьте правильность декодирования при отсутствии ошибок.
3. Исследуйте способность кода исправлять ошибки.
4. Проверьте факт ошибочного декодирования в режиме исправления ошибок.
5. Исследуйте способность кода обнаруживать ошибки.
6. Проверьте факт не обнаружения ошибок кодом.
7. Распечатайте таблицу с разрешенными кодовыми комбинациями и таблицу с расстояниями Хемминга для разрешенных комбинаций.

Методические указания по выполнению работы

Проверка работы кодера

Задайте в ‘источнике’ различные информационные последовательности.

Пронаблюдайте последовательности, полученные на выходе кодера.

Сравните полученные кодовые слова с результатом индивидуального задания.

Проверка правильности декодирования при отсутствии ошибок

Установите в ‘датчике ошибок’ все разряды в ноль, т.е. нет ошибок.

Задайте в источнике различные информационные последовательности.

Пронаблюдайте последовательности, полученные на выходе декодеров (Декодер 1, Декодер 2).

Убедитесь, что при отсутствии ошибок, последовательности на выходах декодеров совпадают с последовательностью на входе кодера.

При выполнении этого пункта обратите внимание, что при отсутствии ошибок синдром равен нулю.

По результатам этого пункта сделайте распечатку (Распечатка №1).



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Методика исследования способности кода исправлять ошибки

Задайте в 'источнике ошибок' однократную ошибку (например 1000000). Убедитесь, что на выходе 'Декодера 1' будет та же последовательность, что и в 'Источнике'.

Сместите циклически 6 раз вектор ошибки и убедитесь, что при любой однократной ошибке 'Декодер 1' правильно её исправит.

При выполнении этого пункта обратите внимание, что синдром ошибки зависит от вектора ошибки и не зависит от передаваемой комбинации. По результатам этого пункта сделайте распечатку (Распечатка №2).

Методика проверки ошибочного декодирования в режиме исправления ошибок.

Задайте в 'Источнике ошибок' несколько векторов ошибок с кратностью от 2 до 7. Убедитесь, что при таких ошибках последовательность на выходе 'Декодера 1' не будет совпадать с последовательностью, которую Вы задали в 'Источнике' [25].

По результатам этого пункта сделайте распечатку (Распечатка №3).

Методика исследования способности кода обнаруживать ошибки

Задайте в 'Источнике ошибок' различные ошибки кратностью от 1 до 7.

Убедитесь, что подавляющее большинство векторов ошибок будет обнаружено (на панели 'Декодер 2' будет гореть индикатор 'Обнаружена ошибка').

По результатам этого пункта сделайте распечатку (Распечатка №4).

Методика проверки факта не обнаружения ошибки кодом

Задайте в 'Источнике ошибок' различные ошибки кратностью от 3 до 7.

Убедитесь, что имеются вектора ошибок, которые кодом не обнаруживаются (при наличии ошибки на панели 'Декодер 2' не будет гореть индикатор 'Обнаружена ошибка').

По результатам этого пункта сделайте распечатку (Распечатка №5).

Распечатка таблицы с разрешенными кодовыми комбинациями.

Перейдите на закладку 'таблицы' и нажмите клавишу 'Печать'

Содержание отчета

Отчёт должен содержать:

стр 1. Титульный лист.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

стр 2. Цель работы и функциональную схему лабораторного макета.

7. Функциональную схему нарисуйте используя карандаш и линейку.

стр 3. Индивидуальное задание.

стр 4. Распечатку №1, показывающую работу кодера и правильности декодирования при отсутствии ошибок.

Распечатку №2, показывающую способности кода исправлять однократную ошибку.

стр 5. Распечатку №3, показывающую факт ошибочного декодирования в режиме исправления ошибок. Распечатку №4, показывающую способности кода обнаруживать ошибки.

стр 6. Распечатку №5, показывающую факт не обнаружения ошибки кодом.

Распечатку №6, показывающую разрешенные кодовые комбинации и расстояния Хемминга.

стр 7. Краткий ответ на 5 контрольных вопросов (номера вопросов выберите самостоятельно) и выводы по работе.

*** Образец оформления отчёта смотрите в файле: ЛР_13_Образец_отчёта.doc

Контрольные вопросы

9. Поясните понятия: блочные, непрерывные, делимые, неделимые, итеративные, линейные, циклические коды?

10. Что такое расстояние Хемминга и кодовое расстояние?

11. Определение и основные свойства циклического кода.

12. Какое правило кодирования циклическим кодом принято в лабораторной работе?

13. Какое правило декодирования принято в декодере в режиме исправления ошибок?

14. Какое правило декодирования принято в декодере в режима обнаружения ошибок?

15. Как связаны кратности гарантированно исправляемых кодом ошибок t и гарантированно обнаруживаемых кодом ошибок с кодовым расстоянием d ?

16. Какие векторы ошибок не могут быть обнаружены линейным циклическим



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

кодом?

10. Сколько различных векторов ошибок может быть исправлено, не исправлено, обнаружено, не обнаружено кодом (7,4)?
4. Как рассчитать вероятность не обнаружения ошибки при заданном канале?
5. Как рассчитать вероятность ошибочного декодирования при заданном канале?
6. Как по одной известной разрешённой комбинации циклического кода определить все остальные кодовые комбинации этого кода?
7. Известна комбинация на входе кодера v_1 и выходе декодера v_2 .
Как определить вектор ошибки E ? ($v_1 \neq v_2$)
2. Дана длина кодовой комбинации n , вероятность ошибки в канале РОШ.
Как определить вероятность появления в кодовой комбинации ошибки кратностью t ?
5. Как производится кодирование - декодирование при использовании кода с проверкой на чётность (на нечётность)?

Приложение

Краткие сведения из теории линейных циклических кодов
Линейные коды являются кодами блочными, регулярными. Для регулярных кодов задаются правила преобразования информационного слова длины k в кодовую последовательность длины n ($n > k$), а также правила декодирования. Наибольшее распространение получили линейные разделимые коды. Разделимым кодом называется код, в кодовых словах которого можно указать места информационных и проверочных символов.

Линейным кодом называют блочный (n, k) код, символы кодовых слов которого являются линейными комбинациями информационных символов.

Если $M=(m_{k-1}, m_{k-2}, \dots, m_0)$ обозначает последовательность k двоичных информационных символов, то правило образования кодового слова V разделимого линейного кода с длиной блока $n > k$ из последовательности M можно записать системой линейных уравнений:



$$v_i = m_i \oplus d_{l,i} \quad 0 \leq i \leq k-1 \quad (I)$$

$$v_i = \sum_{l=0}^{k-1} m_l * d_{l,i} \quad 0 \leq i \leq n-1$$

где \oplus - знак суммирования по модулю 2, коэффициенты $d_{l,i}$ являются символами двоичного алфавита, они могут быть выбраны произвольно, но должны быть фиксированными для данного кода.

Приведем ряд свойств линейных кодов [14, с.283-286].

6. Сумма по модулю 2 двух кодовых слов также является кодовым словом (свойство замкнутости по отношению к операции сложения).

7. Кодовое расстояние d , определяемое как наименьшее расстояние Хемминга между всеми возможными парами кодовых слов, в линейном коде равно минимальному весу ненулевого кодового слова. (Весом кодового слова называется число содержащихся в нем единиц).

Расстояние Хемминга между двумя кодовыми словами равно числу единиц в сумме этих слов по модулю 2, т.е. количеству разрядов, в которых различаются эти два кодовых слова. Например:

первое кодовое слово: 1000110,
 второе кодовое слово: 0100010,
 сумма по модулю два: $\begin{array}{r} 1000110 \\ 0100010 \\ \hline 1100100 \end{array}$ -> расстояние Хемминга равно 3.

Кодовое расстояние d , связано с кратностью t гарантированно исправляемых кодом ошибок соотношением,

$$d \geq 2t + 1$$

9. с кратностью t гарантированно обнаруживаемых кодом ошибок соотношением

$$d \geq t + 1$$

Циклические коды относятся к классу линейных кодов и обладают всеми их свойствами. Дополнительным условием по отношению к циклическому линейному коду является условие замкнутости по отношению к операции



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

циклического сдвига кодовых слов.

Циклическим кодом называется такой линейный код, у которого при любом циклическом сдвиге какого-либо кодового слова получается другое кодовое слово.

Циклическим сдвигом называется операция, превращающая вектор $V_1 = (v_1, v_2, v_3, \dots, v_n)$ в вектор $V_2 = (v_n, v_1, v_2, \dots, v_{n-1})$ – сдвиг вправо, или в вектор $V_3 = (v_2, v_3, \dots, v_n, v_1)$ – сдвиг влево.

Например, дан вектор 0001011. При циклическом сдвиге вправо получится вектор 1000101.

Для описания циклических кодов используют математический аппарат, основанный на сопоставлении множества кодовых слов с множеством степенных полиномов V , с.296, 2, с.110, 3, с.263. Этот аппарат позволяет выявить для циклического кода более простые операции кодирования и декодирования.

Установим формальное соответствие между кодовыми словами и степенными полиномами. Кодовому слову

$$V = (v_{n-1}, v_{n-2}, v_{n-3}, \dots, v_1, v_0)$$

где $v_{n-1}, v_{n-2}, v_{n-3}, \dots, v_1, v_0$ двоичные числа, можно поставить в соответствие полином $(n-1)$ -й степени

$$V(x) = v_{n-1}x^{n-1} + v_{n-2}x^{n-2} + \dots + v_1x + v_0.$$

Такая запись соответствует записи числа в виде многочлена в двоичной системе счисления, где x - основание системы счисления.

Например, кодовому слову 0001011 соответствует полином

$$x^3 + x + 1, \text{ кодовому слову } 0100110 - \text{ полином } x^5 + x^2 + x.$$

Операцию сложения кодовых слов можно заменить сложением соответствующих полиномов. Например, сложив по модулю 2 полиномы из вышеприведенного примера, получим:

$$(x^3 + x + 1) + (x^5 + x^2 + x) = (x^5 + x^3 + x^2 + 1).$$

Этому полиному соответствует слово 0101101, т.е. результат поразрядного сложения по модулю 2 слов 0001011 и 0100110.

Умножение полинома $v(x)$ на x по модулю $(x^n + 1)$ соответствует



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

циклическому сдвигу кодового слова влево на один разряд.

Числом A по модулю B называется остаток от деления A на B и обозначается $A \bmod B$.

12. в вашем случае все полиномы степени выше $(n-1)$ заменяются остатком от их деления на полином $x^n - 1$.

Среди всех полиномов, соответствующих кодовым словам циклического кода, имеется ненулевой полином $g(x)$ наименьшей степени. Этот полином полностью определяет соответствующий код и поэтому называется образующим [15, с. 297-293].

Степень образующего полинома $g(x)$ равна $n-k$, свободный член всегда равен единице.

Образующий полином является делителем всех полиномов, соответствующих кодовым словам циклического кода.

Нулевая комбинация обязательно принадлежит любому линейному циклическому коду и может быть записана как $(x^n - 1) \bmod (x^n - 1) = 0$. Следовательно, образующий полином $g(x)$ должен быть делителем биннома $x^n - 1$.

Это даёт конструктивные возможности построения циклического кода заданной длины n : любой полином, являющийся делителем биннома $x^n - 1$, можно использовать в качестве образующего.

При построении циклических кодов, пользуются таблицами разложения бинномов $x^n - 1$ на неприводимые полиномы, т.е. полиномы, которые нельзя представить в виде произведения двух других полиномов.

Любой неприводимый полином, входящий в разложение биннома $x^n - 1$, а также любое произведение неприводимых полиномов может быть выбрано в качестве образующего полинома, что даёт соответствующий циклический код.

Для построения разделимого циклического кода используется следующее правило построения кодовых слов [14, с. 300] [15, с. 110-113].

$$V(x) = m(x) * x^{n-k} \bmod g(x),$$

где $r(x)$ - остаток от деления $m(x) * x^{n-k}$ на $g(x)$.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Степень $r(x)$, очевидно, меньше $(n-k)$, а потому в кодовом слове первые k , символов будут совпадать с информационными, а последние $n - k$ символов будут проверочными.

Получить все разрешённые комбинации циклического кода можно по одной известной кодовой комбинации (не нулевой). Для этого сначала надо циклически сдвинуть известную комбинацию $k-1$ раз. В результате получится

5. комбинаций. Затем сложить попарно по модулю 2 получившиеся комбинации в различных сочетаниях (по две, по три и т.д.). Нулевую комбинацию (все нули) можно получить, сложив по модулю 2 любую комбинацию саму с собой. Всего получится 2^k кодовых слов.

основу процедуры декодирования циклических кодов может быть положено свойство их делимости без остатка на образующий полином $g(x)$.

режиме обнаружения ошибок, если принятая последовательность делится без остатка на $g(x)$, делается вывод, что ошибки нет или она не обнаруживается. В противном случае комбинация бракуется.

Слова любого линейного кода обладают свойством замкнутости по отношению к операции сложения, т.е. сумма двух и более кодовых слов тоже является кодовым словом.

Из этого свойства, видно, что векторы ошибок, совпадающие с кодовыми словами, не могут быть обнаружены декодером циклического кода.

Вектором ошибок называется двоичная последовательность длины n , в которой единицы стоят на позициях ошибочных символов. Отсюда вероятность не обнаружения ошибки заданным кодом равна вероятности появления в заданном дискретном канале векторов ошибок, совпадающих с кодовыми словами.

Относительно просто эти вероятности могут быть рассчитаны для двоичного симметричного канала (ДСК) без памяти.

таком канале каждый двоичный символ с некоторой фиксированной вероятностью $(1 - p_0)$ принимается правильно и с вероятностью p_0 изменяется помехой на обратный. Передача - приём каждого символа полагается событием независимым (канал без памяти).



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Если по такому каналу передается кодовое слово длины n , то вероятность $P_n(0)$ того, что не произойдет ни одной ошибки, равна $(1 - p_0)^n$.

Вероятность $P_n(1)$ того, что будет одна ошибка в заданном символе, равна $p_0 \cdot (1 - p_0)^{n-1}$.

Вероятность того, что слово на выходе канала будет отличаться от переданного слова в заданных t разрядах, т.е. вектор, ошибок содержит t единиц, равна $P^* = p_0^t (1 - p_0)^{n-t}$.

Вероятность того, что слово на выходе канала будет отличаться от переданного слова в любых t разрядах, равна

$$P_n(t) = C_n^t p_0^t (1 - p_0)^{n-t}, \text{ где } C_n^t \text{ – число сочетаний из } n \text{ по } t.$$

Число единиц в векторе ошибок часто называют его весом.

Положим некоторый линейный код (5,3) содержит одно нулевое слово (как всякий линейный код), два слова веса 2, четыре слова веса 3, одно слово веса 4 (всего $2^k 2^3 = 8$ слов).

Вероятность не обнаруживаемой ошибки этим кодом равна вероятности появления в ДСК векторов ошибок, совпадающих с кодовыми словами, т.е.:

$$P_{н.о} = 2P_5(2) + 4P_5(3) + P_5(4) = 2p_0^2(1 - p_0)^3 + 4p_0^3(1 - p_0)^2 + p_0^4(1 - p_0).$$

2. режиме исправления ошибок декодер вычисляет остаток $S(x)$ от деления принятой последовательности $P(x)$ на $g(x)$. Этот остаток называют синдромом. Принятый полином $P(x)$ представляет собой сумму по модулю два переданного слова $V(x)$ и вектора ошибок $E(x)$:

$$P(x) = V(x) \oplus E(x)$$

Тогда синдром $S(x) = P(x) \bmod g(x)$, так как по определению циклического кода $V(x) \bmod g(x) = 0$.

Определенному синдрому $S(x)$ может быть поставлен в соответствие определенный вектор ошибок $E(x)$. Тогда переданное слово $V(x)$ находят, складывая $P(x) \ominus E(x)$.

Однако один и тот же синдром может соответствовать 2^k различным векторам ошибок. Положим, синдром $S(x)$ соответствует вектору ошибок $E_1(x)$. Но и все векторы ошибок, равные сумме $E_1(x) \oplus V(x)$, где $V(x)$ любое кодовое слово, будут давать тот же синдром. Поэтому, поставив в соответствие синдрому $S_1(x)$ вектор ошибок $E_1(x)$, мы будем осуществлять правильное декодирование в случае, когда действительно вектор ошибок равен $E_1(x)$, во всех остальных $2^k - 1$ случаях декодирование будет ошибочным.

Для уменьшения вероятности ошибочного декодирования из всех возможных векторов ошибок, дающих один и тот же синдром, следует



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

выбирать в качестве исправляемого наиболее вероятный в заданном канале.

Например, в ДСК, в котором вероятность p_0 ошибочного приёма двоичного символа много меньше вероятности $(1 - p_0)$ правильного приема, вероятность появления векторов ошибок уменьшается с увеличением их веса t . В этом случае следует исправлять в первую очередь вектор ошибок меньшего веса.

Если кодом могут быть исправлены только все векторы ошибок веса t и меньше, то любой вектор ошибки веса от $t + 1$ до n , будет приводить к ошибочному декодированию [25].

Вероятность ошибочного декодирования будет равна вероятности $P_n(>t)$ появления векторов ошибок веса $t + 1$ и больше в заданном канале. Для ДСК эта вероятность будет равна

$$P_n(t) = \sum_{i=t+1}^n C_n^i p^i (1-p)^{n-i}.$$

Общее число различных векторов ошибок, которые может исправлять циклический код, равно числу ненулевых синдромов $- 2^{n-k} - 1$.

Лабораторная работа №8

Исследование работы устройства синхронизации с дискретным управлением

Цель работы

Ознакомиться с принципом действия системы синхронизации с дискретным управлением.

Научиться производить расчет основных параметров устройств синхронизации.

Получить практические навыки по моделированию системы синхронизации

3. дискретным управлением.

Домашнее задание

4. Пользуясь настоящим описанием, рекомендуемой литературой и приложением изучите методы и устройства синхронизации в системах передачи дискретных сообщений (ПДС).

5. Для заданной скорости передачи и коэффициента деления (Кд) управляемого делителя частоты (УДЧ) определите номинальную частоту задающего генератора приёма [16. с. 355].



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

5. Рассчитайте расхождение фазы за длительность 1 посылки при заданной нестабильности скорости передачи (при отключенной системе синхронизации).
6. Рассчитайте время поддержания синфазности (для $\epsilon_{доп}=0.5$).

Вариант задания возьмите из таблицы 8.1.

Номинальная скорость передачи задается первой буквой, Кд УДЧ второй буквой, а отклонение реальной скорости передачи от номинальной скорости передачи третьей буквой Вашей фамилии.

Например, для фамилии **МАРКОВ** задание будет таким:

номинальная скорость передачи (буква М): 1600 Бод;

Кд УДЧ (буква А): 8;

отклонение скорости передачи (буква Р): + 30 Бод;

Таблица 8.1.

Скорость передачи	1,2,3 буквы Вашей фамилии					отклонение скорости передачи
1000	А	Ж	Н	Ф	Ы	+60
1100	Б	З	О	Х	Ь	+50
1200	В	И	П	Ц	Э	+40
1300	Г	Й	Р	Ч	Ю	+30
1400	Д	К	С	Ш	Я	-40
1500	Е	Л	Т	Щ		-50
1600	Ё	М	У	Ъ		-60
Кд	8	9	10	12	14	

Описание лабораторной установки

Процесс моделирования производится на компьютере IBM PC. После запуска программы на экране появится структурная схема компьютерной модели лабораторной установки. Она представляет собой систему передачи



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

дискретных сообщений. Для получения на приёме тактовой частоты используется система синхронизации с дискретным управлением.

Назначение системы синхронизации – сформировать на приёме тактовую последовательность, которая будет отмечать границы принимаемых посылок. Это необходимо, чтобы приёмник мог определить, где находятся границы посылок для принятия решения о знаке принимаемой посылки (ноль или единица) [28].

6. блоке 1 (задающий генератор передачи) формируется тактовая последовательность, которая определяет скорость передачи.

7. блоке 2 (датчик) на основе тактовой частоты и датчика формируется передаваемый сигнал.

8. блоке 3 (канал) можно внести различные искажения в передаваемый сигнал.

9. блоке 4 (фазовый детектор) происходит сравнение временного положения фронтов сигнала на выходе канала (моментов смены полярности) с состоянием управляемого делителя частоты (УДЧ).

10. блоке 5 (устройство управления) происходит формирование управляющего сигнала для коррекции фазы УДЧ приёма.

11. блоке 6 (управляемый делитель частоты) происходит деление частоты задающего генератора приёма и формирование частоты приёма.

12. блоке 7 (задающий генератор приёмника) происходит формирование задающей частоты приёмника.

Опишем эти блоки подробнее.

7. блоке задающего генератора передачи (блок 1) задаётся номинальная скорость передачи и отклонение реальной скорости от номинальной скорости. При помощи клавиш '+' и '-' можно задать знак отклонения. Если обе клавиши отжаты, то реальная скорость передачи равна номинальной скорости передачи.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

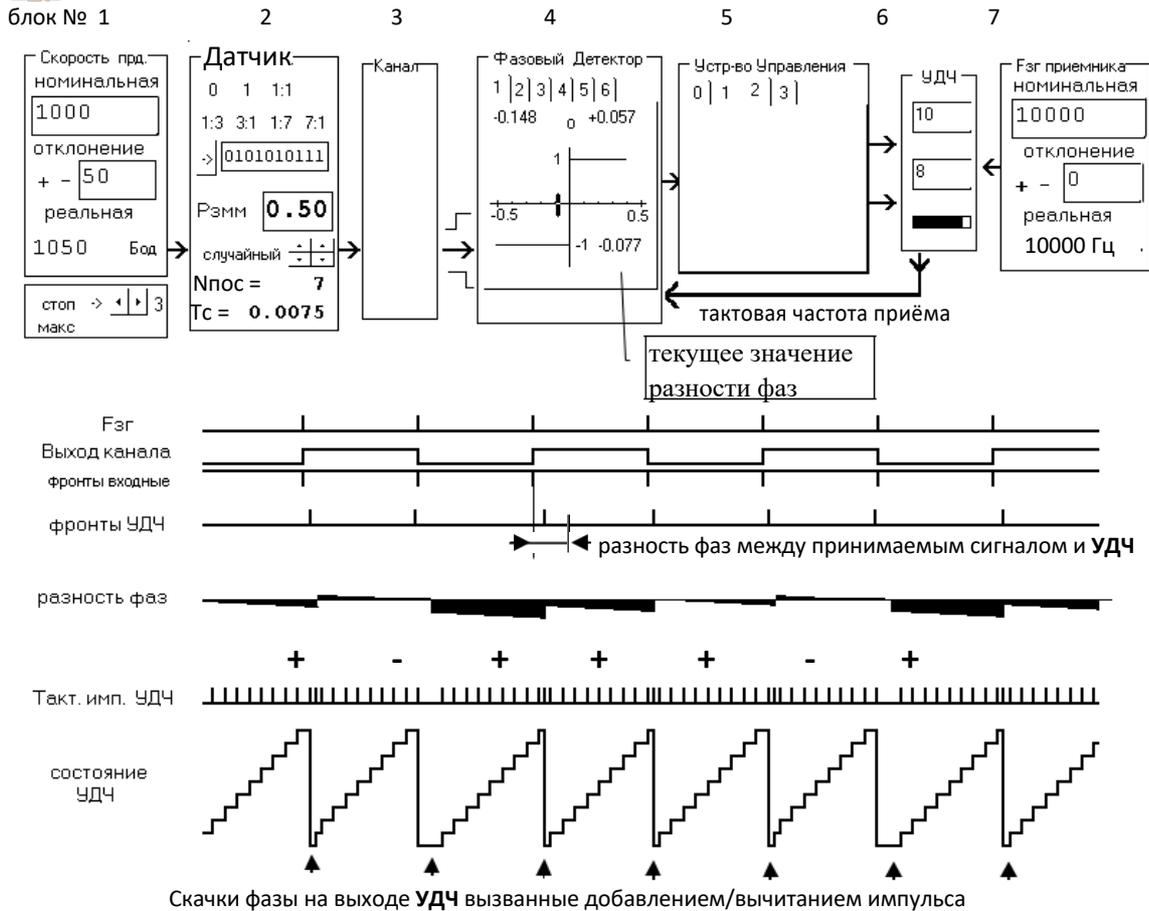


Рисунок 8.1 - Структурная схема лабораторной установки и диаграммы работы

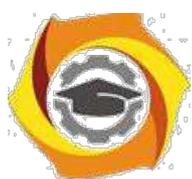
Под этим блоком расположена панель, на которой имеются кнопки управления процессом моделирования:

- Стоп** остановить процесс моделирования;
- ?** продолжить процесс моделирования;
- ?** - 6. уменьшить/увеличить скорость моделирования;
- Макс** - сделать скорость моделирования максимальной (при этом отключается отображение графиков на экране).

8. датчике сигнала (блок 2) имеется возможность установить вид передаваемого сигнала:

- 0 – постоянно передаётся нулевое значение;
- 1 – постоянно передаётся единичное значение;
- 1:1 – передаётся сигнал ‘ТОЧКИ’, постоянно чередование нуля и единицы (0101010101 ...);
- 1:3 – периодический сигнал состоящий из одной единицы и трёх нулей (10001000100...);
- 1:7 – периодический сигнал состоящий из одной единицы и семи нулей (10000000100...);

511 – псевдослучайный сигнал с периодом повторения 511 бит;
Имеется возможность задать собственную структуру сигнала.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Также имеется возможность получения случайного сигнала и задать вероятность появления смены знака посылки в этом сигнале.

В нижней части датчика расположены индикаторы количества переданных посылок и времени сеанса. Если щёлкнуть по ним мышью, то они примут исходные (нулевые) значения.

канале (блок 3) можно вносить краевые искажения (отклонения реальных моментов смены полярности от номинальных) и дробления (появление дополнительных фронтов в пределах одной посылки) в передаваемый сигнал. С выхода канала сигнал поступает на вход фазового детектора.

фазовом детекторе (блок 4) происходит выделение сигнала рассогласования, указывающего величину и знак фазового сдвига между сигналами синхронизации, поступающими из канала (фронты входного сигнала), и сигнала на выходе управляемого делителя частоты [26].

Имеется возможность выбирать тип фазового детектора и его параметры.

10. верхней части фазового детектора имеются индикаторы пиковых значений расхождения фазы. Их значения можно сбросить кнопкой, расположенной между индикаторами. В нижнем правом углу расположен индикатор текущего значения расхождения фазы. Сигнал с выхода фазового детектора поступает на устройство управления.

устройстве управления (УУ) (блок 5) производится анализ сигнала расхождения фазы и формируются управляющие сигналы, которые поступают на управляемый делитель частоты и корректируют его фазу.

Имеется возможность менять алгоритм работы устройства управления.

Первое УУ формирует импульс сброса управляемого делителя частоты, если в принимаемом сигнале произойдёт смена полярности.

Второе УУ в зависимости от сигнала на выходе фазового детектора формирует дополнительный импульс на входе счетчика или запрещает прохождение одного тактового импульса на вход счетчика от задающего генератора приёма.

Если в момент сравнения фаз (в момент появления фронта на выходе канала) окажется, что фаза УДЧ отстаёт от фазы сигнала на выходе канала (приёмник 'отстаёт'), то на вход счётчика поступит дополнительный импульс.

11. результате тактовый сигнал на выходе делителя сдвинется на величину Δt (Δt - период следования импульсов задающего генератора приёма).



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Если фаза счётчика опережает фазу сигнала на выходе канала (приёмник ‘спешит’), то УУ сформирует сигнал на запрет прохождения одного импульса от задающего генератора приёмника, в результате чего тактовая частота на выходе делителя частоты сдвинется в сторону отставания на Δt .

Третье УУ работает аналогично второму, только содержит усредняющее устройство (интегратор).

Если в момент сравнения фаз фаза УДЧ отстает от фазы принимаемого сигнала, то счетчик интегратора увеличивает своё значение на единицу. Если в момент сравнения фаз фаза делителя опережает фазу принимаемого сигнала, то счетчик интегратора уменьшает своё значение на единицу.

Тем самым коррекция фазы управляемого делителя частоты (добавление/вычитание импульса) происходит не после каждого момента сравнения фаз, а после того как интегратор достигнет некоторого порогового значения (его можно изменять). После коррекции фазы УДЧ счетчик интегратора сбрасывается в ноль.

Интегратор позволяет исключить мгновенные рассогласования фазы, вызванные действием помех.

и управляемом делителе частоты (УДЧ) происходит деление частоты задающего генератора приёма на коэффициент Кд. В результате этого на выходе УДЧ появляется сигнал с частотой, численно ‘почти’ равной номинальной скорости передачи (если не производить коррекцию частоты УДЧ).

Эта разность частот передачи и приёма определяется выражением $f = \square$
 $f_{зг.пер} - f_{зг.прм}/Кд$.

Где $f_{зг.пер}$ – реальная частота задающего генератора передатчика,

$f_{зг.прм}$ – реальная частота задающего генератора приёмника,

Кд – коэффициент деления управляемого делителя частоты.

Под воздействием управляющих сигналов (добавить/вычесть импульс или обнулить УДЧ) поступающих из устройства управления можно не только свести f к нулю, но и минимизировать разность фаз между тактовой частотой приёма формируемой на выходе УДЧ и тактовой частотой передачи.

В блоке УДЧ имеется возможность менять коэффициент деления. Также имеется индикатор текущего значения счетчика УДЧ.

В блоке задающего генератора приёма (блок 7) задаётся номинальная частота тактового генератора приёмника и отклонение реальной частоты



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

приёмника от номинальной частоты. При помощи клавиш '+' и '-' можно задать знак отклонения. Если обе клавиши отжаты, то реальная частота приёмника равна номинальной частоте.

Лабораторное задание

- и Запустите программу лабораторной работы на выполнение.
- и Установите требуемые (как в домашнем задании) значения номинальной скорости передачи, нестабильность скорости передачи, коэффициент деления (Кд) управляемого делителя частоты (УДЧ), номинальную частоту задающего генератора приёма .
- и Проверьте правильность сделанных Вами расчетов.
- и Для установленных параметров определите полосу удержания системы синхронизации для различных видов сигналов 1:1, 1:3, 1:7, 511, 3:1,7:1.
- и Определите полосу удержания, если Кд УДЧ будет в 2 раза больше чем в предыдущем пункте (домашнем задании).
- и Определите погрешность синхронизации в зависимости от отклонения реальной скорости передачи от номинальной для двух видов сигналов 1:1 и 1:3. Номинальную скорость передачи и Кд УДЧ возьмите из домашнего задания.
- и В процессе выполнения пунктов 4,5,6 сделайте 4 распечатки показывающие работу системы с дискретной ФАПЧ.

Методические указания по выполнению работы

Подробные методические указание по выполнению лабораторной работы смотрите в программе лабораторной работы в меню “Методические указания”.

Содержание отчета

Отчет оформите в тетради (листочки и коллективные работы не предлагать).

Отчет должен содержать (минимально):

1 страница (титульная): Сокращённое названия университета, название кафедры, номер и название лабораторной работы, Ф.И.О выполняющего работу, Ф.И.О преподавателя, дату выполнения лабораторной работы;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

2 страница: Цель работы, структурную схему устройства синхронизации с дискретным управлением (рис 2). Выполненное домашнее задание (смотрите пункт 2.1-2.3).

3 страница результаты работы по пунктам 5.4, 5.5 и 5.6.

4,5,6 листы. Выводы по работе и письменные ответы на контрольные вопросы. Минимально надо ответить на 5 вопросов, рекомендуется – 8. Для отличников – 10. Ответы снабдите соответствующими рисунками.

Также вклейте в отчёт результаты полученные в пункте 7 лабораторного задания.

Контрольные вопросы

- С Поясните принцип работы устройства синхронизации с непосредственным воздействием на задающий генератор?
2. Поясните принцип работы устройства синхронизации без непосредственного воздействия на задающий генератор?
- и Раскройте следующие понятия: абсолютная и относительная нестабильность частоты, погрешность синхронизации, время синхронизации, время поддержания синхронизма, вероятность срыва синхронизма, шаг коррекции.
- и Почему с увеличением шага коррекции (уменьшении Кд УДЧ) время вхождения в синхронизм уменьшается?
- и Как влияет точность синхронизации на верность передачи сообщений?
- и За сутки часы уходят вперёд на 5 секунд. Чему равна абсолютная и относительная нестабильность генератора в часах ? За какое время часы уйдут вперёд на один час? (задающая частота кварцевого генератора в часах $F_{зг} = 32768 \text{ кГц}$)
- и Почему на сигнале 1:1 время вхождения в синхронизм будет меньше, чем на сигнале 1:7?
- и Поясните следующие понятия : полоса захвата, полоса удержания ?
- и Почему на сигнале 1:1 полоса удержания системы ФАПЧ будет шире, чем на сигнале 1:7?
- и На какую величину (в долях от длительности посылки) сместится стробирующий импульс за время перерыва в канале $t=1 \text{ мин}$, если взаимная относительная нестабильность генераторов передачи и приёма равна 10^{-5} ?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

В Какой алгоритм подстройки фазы Вы используете при коррекции времени на своих часах

В Вы решили на своих часах попробовать алгоритм подстройки частоты (времени) с непосредственным воздействием на генератор. Как это будет выглядеть?

приложение 1.

Основные теоретические сведения

Для правильной работы оконечных устройств в синхронных системах необходимо, чтобы распределители передающего и приёмного устройств работали синхронно и синфазно. Две последовательности событий называются синхронными, если соответствующие события в них происходят одновременно. у системах связи одна последовательность событий происходит в передатчике, а другая в приёмнике. Хотя эти последовательности и сдвинуты относительно друг друга на некоторое время (время прохождения сигнала), их принято считать синхронными.

Для систем передачи данных проблема синхронизации состоит в образовании во всех элементах сети единой шкалы времени с равными интервалами и легко определяемыми границами между ними. Условие синхронизма является необходимым при взаимодействии двух оконечных установок через сеть связи для правильного приема, как отдельных бит принимаемого сигнала, так и комбинации этих бит, отображающие символы передаваемых данных [27].

При взаимодействии различных цифровых станций сохраняется задача синхронизации по битам, называемая обычно тактовой синхронизацией. Необходимость точной идентификации границ между группами бит выдвигает задачу так называемой цикловой синхронизации элементов сети. Среди возможных способов синхронизации наиболее широко распространён способ подстройки скорости работы приёмника под скорость работы передатчика. Для электронных устройств это означает подстройка частоты колебаний генератора приёмной оконечной аппаратуры под частоту поступающих из канала связи сигналов

лабораторной работе исследуется устройство синхронизации без непосредственного воздействия на генератор.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Местный генератор вырабатывает сигналы с частотой $F_{зг}$ в K_d раз большей требуемой тактовой частоты F_t . Делитель частоты делит частоту $F_{зг}$ в K_d раз. На выходе делителя действует сигнал с частотой F_t . Этот сигнал направляется в устройство регистрации приёмника и одновременно на один из входов фазового детектора. На другой вход фазового детектора поступает сигнал $U_{прм}$ из канала связи (от передатчика). Задачей фазового детектора является сравнение фазы поступающего сигнала $U_{прм}$ (моментов смены знака посылки) с фазой тактового сигнала F_t .

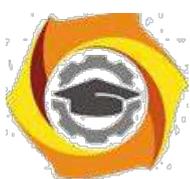
При расхождении фаз этих сигналов фазовый детектор вырабатывает сигнал, соответствующий величине и знаку расхождения фаз анализируемых сигналов. Сигнал с выхода фазового детектора воздействует на управляющее устройство (УУ). На выходе УУ появляется сигнал, который изменяет коэффициент деления делителя, регулируя тем самым фазу частоты F_t так, чтобы фаза частоты F_t совпала (с некоторой погрешностью) с фазой принимаемого сигнала.

Можно коэффициент деления делителя сделать постоянным, а изменять количество импульсов поступающих на вход делителя от местного генератора. Принцип регулирования частоты на выходе делителя частоты путём добавления/вычитания тактового импульса на входе делителя показан на рис. 3.

Если сигнал F_t будет отставать по фазе, то в последовательность импульсов поступающих на вход счетчика будут добавляться дополнительные импульсы. и результате этого частота на выходе счетчика несколько повысится и фаза сигнала F_t начнет 'догонять' фазу принимаемого сигнала.

Если сигнал F_t будет опережать по фазе принимаемый сигнал, то из последовательности импульсов поступающих на вход счетчика будут удаляться (исключаться) импульсы. В результате этого частота на выходе счетчика несколько понизится и разность фаз между F_t и принимаемым сигналом уменьшится.

Достоинством устройств синхронизации без непосредственного воздействия на генератор является то, что частота F_t , остаётся неизменной, что позволяет использовать в качестве местного генератора высокостабильные генераторы, например, генераторы стабилизированные кварцем и делать этот генератор общим для всей аппаратуры (для всех принимаемых потоков).



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

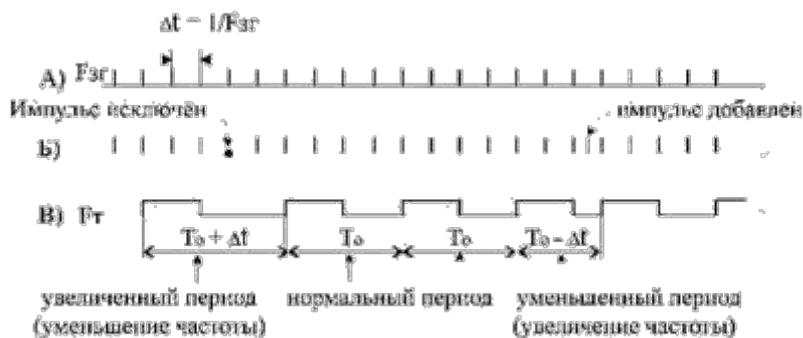


Рисунок 8.2 – Изменение частоты(фазы) при добавлении/исключении импульса (Кдел-4) А-выход местного генератора, Б- вход УДЧ, В- выход УДЧ

Фазирование по элементам (синхронизация) передатчика и приёмника не может постоянно поддерживаться только за счёт стабильности генераторов приёма и передачи. Покажем это на примерах. Обозначим коэффициент относительной нестабильности генератора $k = \Delta f / f_n$, где f_n – номинальная частота генератора, Δf – отклонение реальной частоты от номинального значения. Колебания такого генератора сместятся на один период по отношению к колебаниям с номинальной частотой за время $t_1 = 1/\Delta f = 1/(k \cdot f_n)$.

Время за которое поток на выходе передатчик сместиться на 1 посылку (по сравнению с номинальной скоростью работы) будет равно:

$$t_1 = 1/(k \cdot V), \quad \text{где } V \text{ – номинальная скорость работы.}$$

В самом неблагоприятном случае частоты генераторов передатчика и приёмника могут отклониться от номинала в противоположенные разные стороны, тогда время за которое распределители передачи и приёма разойдутся на 1 посылку равно $t_2 = 1/(|2 \cdot k| \cdot V)$.

Расхождение фазы (на сколько фронт посылки на выходе реального генератора сместится от фронта посылки на выходе идеального генератора) за длительность одной посылки будет равно k .

Время за которое уход по фазе передатчика превысит допустимое значение $\epsilon_{\text{доп}}$ равно: $t_{\epsilon_{\text{доп}}} = \epsilon_{\text{доп}} / (k \cdot V)$.

Нестабильность обычных RC или LC генераторов превышает 10^{-3} . Простые кварцевые генераторы дают нестабильность порядка 10^{-4} . Если использовать специальные схемные решения и поместить кварцевый резонатор с схему генератора в термостат, то можно получить нестабильность 10^{-9} . Кварцевым генераторам свойственно самопроизвольное изменение частоты колебаний с течением времени из-за ‘старения кварца’.

Стремление обеспечить взаимодействие синхронной сети с другими сетями, диктует необходимость руководствоваться рекомендациями МККТТ, касающимися вопросов синхронизации. В частности, рекомендация G.811 гласит, что относительная нестабильность задающих генераторов в узлах связи цифровых сетей должна быть не хуже $1 \cdot 10^{-11}$ за сутки. Современные генераторы обеспечивают суточную относительную нестабильность частоты $1 \cdot 10^{-11}$, например у рубидиевого атомного стандарта Ч1-50.



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

Лабораторная работа № 9

Исследование методов регистрации двоичных сигналов

Цель работы

- и Изучение методов регистрации двоичных сигналов.
- и Изучение принципов работы устройств регистрации.
- и Экспериментальное определение зависимости вероятности ошибки от характеристик устройств регистрации при действии краевых искажений и дроблений.

Индивидуальное задание

Изучите теорию вопроса, программу лабораторной работы, методику проведения эксперимента.

Постройте реализацию сигнала, состоящую из 7 единичных элементов сигнала на входе дискретного канала на его выходе и в контрольных точках приёмников.

Вариант задания формируется на основе Ваших Ф.И.О, номера группы на странице 'Установки' [16].

Дробление в 6 ЕЭ (Единичном Элементе) расположите так, чтобы оно вызвало ошибку у стробирующего приёмника.

Описание программы

Программу лабораторной работы можно записать на кафедре или с сайта кафедры www.pds-01.boom.ru.

Программа содержит несколько страниц:

Страница “Начало” - содержит название лабораторной работы.

На этой странице имеются кнопки:



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

‘о программе’ - можно посмотреть номер версии программы, адрес сайта МТУСИ и кафедры.

‘переписать’ - активизирует диалоговое окно, используемое для перезаписи программы, образца отчёта и т.п. на гибкий диск.

‘выход’ - используется для выхода из программы.

Страница “Установки” - содержит различные установки и панель формирования индивидуального задания [25].

!!! На распечатках, помещаемых в отчёт должны быть Фамилия И.О. и номер группы исполнителя. Для этого необходимо в начале выполнения работы ввести их в требуемые строки на панели ‘Формирование индивидуального задания’.

Страница “Читайте” - содержит разделы: цель работы, задание и т.п.

Раздел ‘задание’ содержит лабораторное задание и методику выполнения каждого пункта.

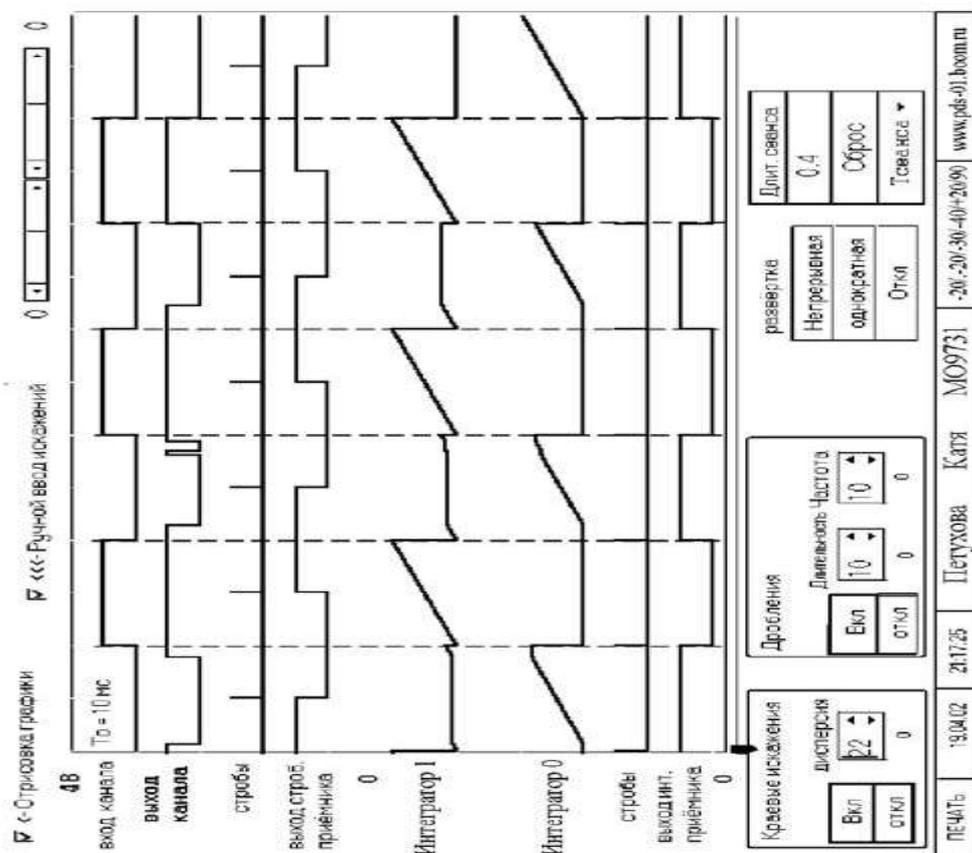


Рисунок 9.1 – ручной вывод искажений

Страница “Диаграммы” - содержит диаграммы, показывающие работу устройств регистрации.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Страница “Схема” - содержит схему устройств регистрации.

Страница “Формулы” содержит различные формулы.

Описание страницы “Диаграммы“

Страница “Диаграммы“ приведена на рисунке 9.1.

На этой странице можно наблюдать следующие сигналы:

вход канала, выход канала, стробы стробирующего приёмника, стробы интегрального приёмника, выход стробирующего приёмника, напряжение на выходе интеграторов нуля и единицы, выход интегрального приёмника. В левой части страницы имеются счетчик переданных единичных элементов и счётчики количества ошибок стробирующего и интегрального приёмников. При возникновении ошибки фон панели счетчика становится красным. Также страница содержит следующие переключатели:

‘Отрисовка графики’ – позволяет включить/отключить отрисовку графики.

‘Ручной ввод искажений’ – включает панель при помощи которой можно вводит искажения в шестую посылку.

Под диаграммами расположен движок, при помощи которого можно перемещать положение дробления в режиме ‘Ручной ввод искажений’

Панель “Краевые искажения” содержит кнопки:

‘Вкл’ - включить краевые искажения; ‘откл’ -
отключить краевые искажения;

Также на этой панели имеется переключатель при помощи которого можно изменять величину дисперсии краевых искажений.

Панель “Дробления” содержит кнопки:

‘Вкл’ - включить дробления
‘откл’ - отключить дробления

Также на этой панели имеется переключатели, при помощи которых можно изменять длительность и частоту дроблений.

Панель “развёртка” содержит кнопки:

Непрерывная - включается непрерывный режим моделирования,



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Однократная - производится отрисовка приёма 8 посылок,

Откл - отключение процесса моделирования.

Панель “Длительность сеанса” содержит кнопку:

‘Сброс’ – обнуление счётчика времени измерения, счётчика переданных единичных элементов и счётчиков ошибок стробирующего и интегрального приёмников. Также на этой панели имеется индикатор времени измерения и переключатель для задания длительности сеанса.

Для распечатки требуемой страницы необходимо нажать клавишу ‘Печать’ расположенную в левом нижнем углу. Если распечатка не произойдёт (замечен такой дефект на некоторых типах принтеров), то можно распечатать файл PRINT_07_хх.BMP, который создаётся при каждом нажатии на кнопку ‘Печать’. Распечатку этого файла можно произвести например при помощи редактора Word [17].

Лабораторное задание

1. Пользуясь методическим описанием, рекомендуемой литературой и программой лабораторной работы изучите методы регистрации двоичных сигналов.

У Определите исправляющую способность приёмников по краевым искажениям.

У Сделайте распечатку, показывающую работу приёмников при действии краевых искажений.

У Определите исправляющую способность приёмников по дроблениям.

У Сделайте распечатку, показывающую работу приёмников при действии дроблений

У Измерьте зависимость количества ошибок на выходе приёмников от длительности дроблений.

Методические указания по выполнению работы

И Методика определения исправляющей способности приёмников по краевым искажениям.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Запустите программу лабораторной работы, на странице 'Установки' введите свои Ф.И.О и номер группы. Перейдите на страницу 'Диаграммы'.

К Отключите датчик краевых искажений. Для этого на панели 'Краевые искажения' нажмите кнопку 'откл'.

К Отключите датчик дроблений. Для этого на панели 'Дробления' нажмите кнопку 'откл'.

3. Включите режим ручного ввода искажений. Для этого включите переключатель 'Ручной ввод искажений'.

2. Смещая движок 'одностороннее искажение' вправо от среднего положения, определите при какой величине краевого искажения начнут возникать ошибки на выходе стробирующего приёмника и интегрального приёмника. Максимальное значение искажения, при котором нет ошибок, и будет исправляющей способностью приёмников по односторонним краевым искажениям.

3. Для определения исправляющей способности по двусторонним краевым искажениям используйте аналогичную методику, только вносите краевые искажения при помощи движка 'двусторонние краевые искажения'. Движок смещайте влево от среднего положения.

Результаты измерения исправляющей способности занесите в табл. 1. (смотрите образец отчета).

Для получения распечатки, показывающей работу приёмников при действии краевых искажений сделайте следующее:

а Выключите переключатель 'Ручной ввод искажений',

а Включите краевые искажения. Для этого на панели 'Краевые искажения' нажмите кнопку 'Вкл'.

3. Установите переключателем 'диспер' (дисперсия) величину краевых искажений, при которой начнут появляться ошибки у обоих приёмников.

Нажимая кнопку 'ОДНОКРАТНАЯ' на панели 'РАЗВЁРТКА' и получите реализацию, на которой ошибки произойдут сразу в двух приёмниках.

Для получения распечатки нажмите кнопку 'ПЕЧАТЬ'.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Методика определения исправляющей способности приёмников по дроблениям.

12. Включите непрерывную развёртку.
13. Выключите датчик краевых искажений.
14. Включите режим 'Ручной ввод искажений'.
15. Установите движком 'Положение маркера и дробления' маркер точно в середину второго единичного элемента.
16. Изменяя величину длительности дробления, определите значение, при котором начнут ошибаться приёмники.

Максимальное значение дробления, при котором ещё нет ошибок, и будет исправляющей способностью приёмников по дроблениям. Результаты измерения исправляющей способности занесите в табл. 2.

Для получения распечатки, показывающей работу приёмников при действии дроблений сделайте следующее:

- Отключите ручной ввод искажений;
- Включите датчик дроблений;
- Подберите длительность дроблений, при которой начнутся ошибки в обоих приёмниках;
- Нажимая кнопку 'ОДНОКРАТНАЯ' на панели 'РАЗВЁРТКА' получите реализацию, на которой ошибки произойдут сразу в двух приёмниках.
- Для получения распечатки нажмите кнопку 'ПЕЧАТЬ'.

Методика измерения количества ошибок на выходе приёмников при действии дроблений.

- p Сделайте установки как в пункте 6.4.
- q Установите длительность сеанса равную 100 сек.
- r Установите длительность дробления равную нулю.
- s Установите частоту дроблений в соответствии с Вашим индивидуальным заданием (смотрите на странице 'Установки').



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

– Нажмите кнопку 'Непрерывная' на панели развёртки. При этом начнётся сеанс измерения. После окончания сеанса измерения кнопка 'Непрерывная' отождётся.

– Изменяя длительность дробления в диапазоне от 0 до 1.2То с шагом 10% определите число ошибок на выходе приёмников. Результаты занесите в таблицу №3 (смотрите образец отчёта в файле ЛР_07_Образец_отчёта.doc).

примечание Для ускорения процесса измерения выключите переключатель 'Отрисовка графики'. При этом процесс моделирования значительно ускорится.

Содержание отчета

Отчёт должен содержать:

стр 1. Титульный лист.

стр 2. Цель работы и схему лабораторного макета.

стр 3. Результаты измерения исправляющей способности приёмников по краевым искажениям;

распечатку, показывающую работу приёмников при действии краевых искажений.

результаты измерения исправляющей способности по дроблениям.

В распечатку, показывающую работу приёмников, при действии дроблений.

стр 4. Результаты измерения и график зависимости вероятности ошибки от длительности дроблений.

стр 5. Индивидуальное задание

стр 6. Краткий ответ на пять контрольных вопроса и выводы по работе

Образец оформления отчёта смотрите в файле ЛР_07_Образец_отчёта.doc

Приложение

Краткие сведения по методам регистрации

Передача дискретных сообщений от передатчика к приёмнику осуществляется по дискретным каналам (ДК), которые нормируются либо номинальной скоростью передачи $V_{\text{НОМ}} = 1/T_0$ и искажениями двоичных



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

сигналов либо скорость передачи $V_{ном}$ и вероятностью ошибки по битам $P_{ош}$ (T_0 – длительность единичного элемента).

Каналы первого типа называют дискретными каналами непрерывного времени (ДКНВ), второго – дискретными каналами дискретного времени (ДКДВ) или цифровыми каналами. Структурная схема передачи - приёма показана на рисунок 9.2.

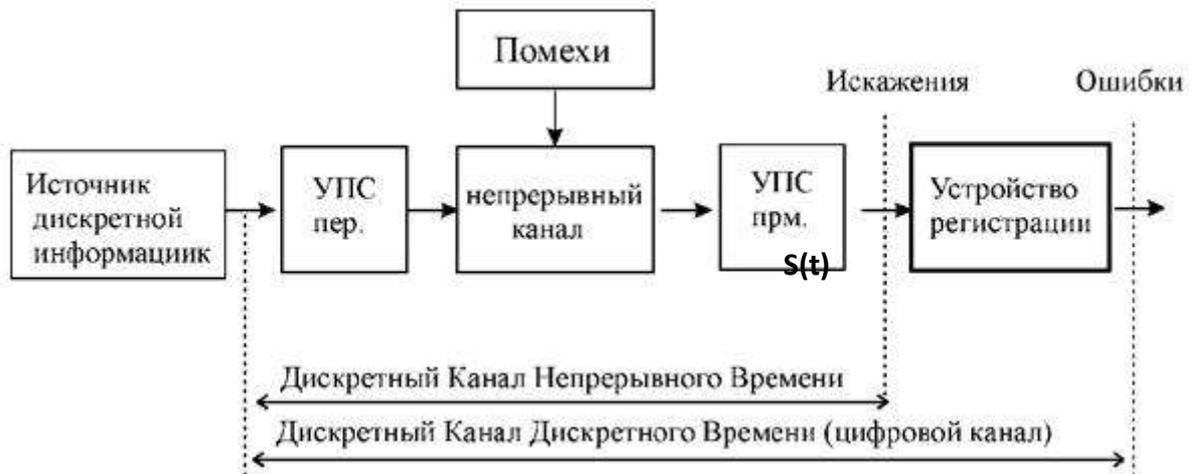


Рисунок 9.2 – структурная схема передачи - приёма

Дискретный сигнал поступает на вход дискретного канала. Моменты времени, в которых происходит смена сигнала (переход от нуля к единице или наоборот) называются значащими моментами (ЗМ).

Под действием помех и других факторов на выходе канала имеется дискретный сигнал $S(t)$ непрерывного времени, который может иметь краевые искажения и дробления.

Краевым искажением называется смещение ZM от его идеального положения.

Дроблением **Схема называется передачи искажение-приёма**, состоящее в появлении элементов обратной полярности во время передачи одного или нескольких подряд идущих единичных элементов одного знака.

Смещение ZM относительно идеального значения вправо принято считать положительным, а в лево – отрицательным.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Сигнал $S(t)$ поступает на вход устройства регистрации, которое на основе принятого искажённого сигнала принимает решение о знаке передаваемого сигнала (0 или 1).

Для решения поставленной задачи используются различные методы регистрации двоичных сигналов, основными показателями которых являются исправляющая способность по краевым искажениям и дроблениям.

Исправляющая способность характеризует свойство устройства правильно регистрировать единичные элементы двоичных сигналов при наличии искажений.

Численно исправляющая способность равна максимальной величине искажений, не способных привести к ошибочной регистрации знака единичного элемента.

Различают исправляющую способность по краевым искажениям и дроблениям.

Устройство регистрации сигналов, обеспечивающее минимальную вероятность неправильного приёма называют оптимальным.

Реализация оптимального устройства регистрации вызывает определённые трудности, и поэтому на практике применяют упрощённые методы регистрации, которые хотя и проигрывают в помехоустойчивости оптимальному методу, однако проще в реализации. К числу наиболее распространённых методов регистрации относятся метод стробирования и метод интегрирования.

Метод стробирования

Метод регистрации единичного элемента по значению его средней части нашёл широкое применение благодаря простоте реализации, а также большой эффективности при действии краевых искажений.

Структурная схема стробирующего устройства, показана на рис. 3.

Устройство состоит из D-триггера, на D вход (информационный вход) которого поступает принимаемый сигнал, а на C вход (тактовый вход) регистрирующие импульсы (стробы) от устройства синхронизации, которые



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

своим передним фронтом (переход от 0->1) устанавливают триггер в состояние, действующее к этому моменту на входе D.

На выходе триггера Q формируются единичные элементы, длительность которых определяется частотой следования регистрирующих импульсов и составляет T_0 .

Другими словами, на выходе регистрирующего устройства формируется восстановленные по длительности единичные элементы (без краевых искажений и дроблений). Стробирующий приёмник ошибается, когда исказится середина посылки. Исправляющая способность стробирующего метода по краевым искажениям равна 50% , а по дроблениям нулю (даже очень короткое дробление попавшее на середину единичного элемента вызывает ошибку).

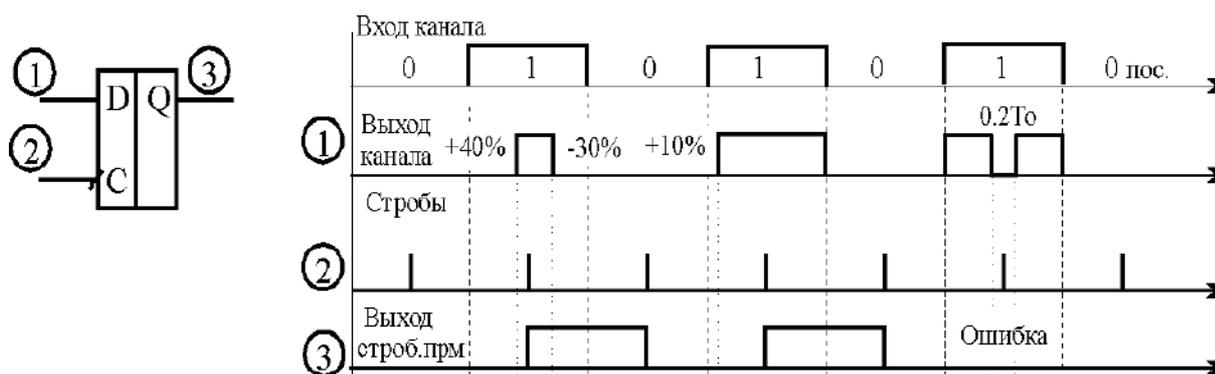


Рисунок 9.3 - Структурная схема стробирующего приёмника и диаграммы его работы

Метод интегрирования

Схема регистрации двоичных сигналов интегральным методом приведена на рисунке 9.3. Устройство состоит из двух интеграторов, инвертора, схемы сравнения и D-триггера.

Работает интегральный приёмник следующим образом.

Принимаемый сигнал поступает на входы двух интеграторов, причём на интегратор_1 непосредственно, а на интегратор_0 через инвертор.

На выходе интеграторов напряжение линейно возрастает когда на входе интегратора логическая 1. Если на входе интегратора логический ноль, то напряжение на выходе интегратора не изменяется.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Напряжения с выходов интеграторов поступают на схему сравнения.

Сигнал с выхода схемы сравнения (0 или 1) поступает на информационный вход D-триггера.

Тактовая последовательность вырабатываемая устройством синхронизации В соответствующая границам единичных элементов исходного сигнала подается на тактовый вход D-триггера.

момент действия тактового импульса в триггер записывается результат сравнения напряжений на выходе интеграторов.

Если за время T_0 сигнал на выходе канал принимал значение 1 большую часть времени чем значение 0, то напряжение на выходе интегратора_1 будет больше чем напряжение на выходе интегратора_0 и в триггер будет записана 1.

и если за это время сигнал на выходе канал принимал значение 1 меньшую часть времени чем значение 0, то напряжение на выходе интегратора_1 будет меньше чем напряжение на выходе интегратора_0 и в триггер будет записан 0.

Также под воздействием тактового импульса происходит обнуление интеграторов (интеграторы подготавливаются к новому циклу интегрирования).

Результат регистрации будет верным, если сумма краевых искажений и дроблений не превысит 50% от длительности единичного элемента T_0 . Рисунок 9.3 - Интегральный приёмник и диаграммы, поясняющие его работу.

Таким образом, исправляющая способность интегрального метода по односторонним краевым искажениям равна 50%. по двухсторонним 25%, а по дроблениям $0.5T_0$.

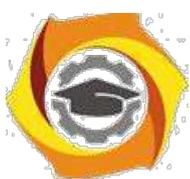
Поскольку метод стробирования имеет большую помехоустойчивость при краевых искажениях, а интегральный метод устойчивее к действию дроблений, то иногда применяют комбинированные способы регистрации двоичных сигналов, сочетая преимущества обоих методов.

Комбинированный метод

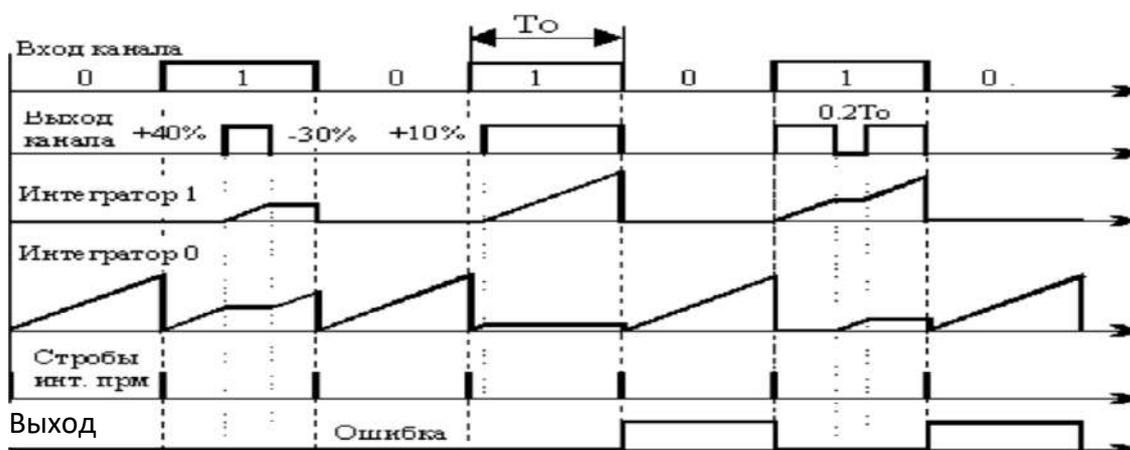
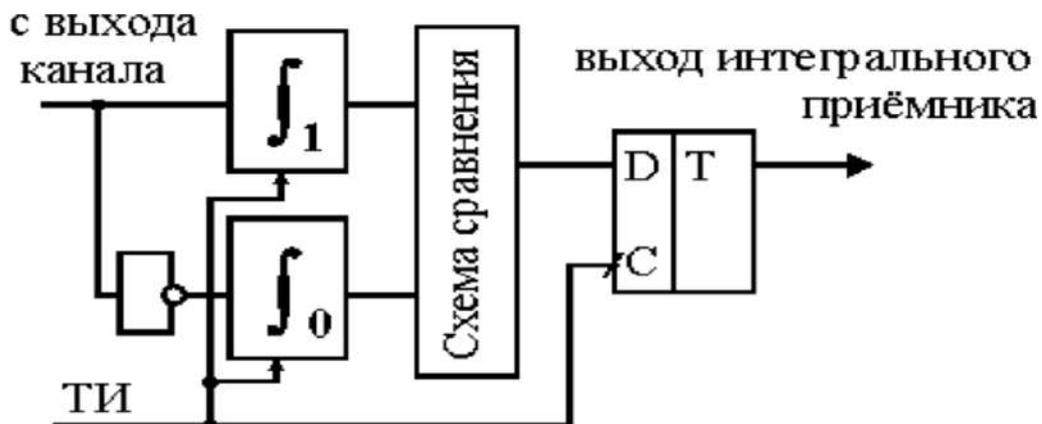
Структурная схема устройства регистрации показана на рис. 5.

Сущность комбинированного способа регистрации состоит в том, что входящий элемент дискретного сигнала стробируется (анализируется) в нескольких точках, например в трёх.

Устройство регистрации включает в себя схему совпадения (логический элемент И), на один вход которой поступает принимаемый сигнал (с краевыми



искажениями и дроблением), а на второй – тактовые импульсы ТИ1 с устройства синхронизации, соответствующие точкам стробирования.



2 выхода схемы совпадения тактовые импульсы, соответствующие элементам сигнала положительной полярности, поступают на тактовый вход счетчика. С выхода Q1 сигнал поступает на информационный вход D-триггера.

Работает схема следующим способом.

После прихода третьего стробирующего импульса ТИ1 тактовый ТИ2 установит триггер в состояние старшего разряда двоичной комбинации на выходе счётчика.

Старший разряд счётчика есть не что иное, как наиболее вероятная полярность принятого сигнала, определяемого мажоритарным способом, т.е. по большинству.

Если в двух или трёх точках регистрации зафиксирована 1, то принимается решение что передавалась 1.

Если в двух или трёх точках регистрации зафиксирован 0, то принимается решение что передавался 0.



Рисунок 9.4 - Комбинированный приёмник и диаграммы, поясняющие работу

Методические указания
к лабораторным работам
по дисциплине
«Безопасность жизнедеятельности»

Лабораторная работа 1
Микроклимат рабочей зоны

Цель работы: 1. Изучить принципы гигиенического нормирования метеорологических условий (микроклимата) в производственных помещениях.

4. Изучить методики и средства контроля параметров микроклимата в производственных помещениях.

5. Научиться оценивать состояние микроклимата на основании проведенных измерений.

1.1. Общие сведения

5. соответствии с СанПиН 2.2.4.548-96 [3], микроклимат - это климат внутренней среды помещения - замкнутого пространства в специально предназначенных зданиях и сооружениях, в которых постоянно (по сменам) или периодически (в течение рабочего дня) осуществляется трудовая деятельность людей. Микроклимат определяется действующими на организм человека сочетаниями его показателей: температуры воздуха, температуры поверхностей*, относительной влажности воздуха, скорости движения воздуха, интенсивности теплового облучения. В гигиеническом отношении эти сочетания оказывают влияние на теплообмен организма работающего с



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

окружающей средой и его тепловое состояние. На формирование производственного микроклимата существенное влияние оказывают технологический процесс и климат местности.

Различные сочетания показателей микроклимата при воздействии их на человека могут быть условно сведены к трем состояниям: комфортное состояние микроклимата, нагревающий микроклимат, охлаждающий микроклимат.

*



При комфортном микроклимате жизнедеятельность человека протекает при температурном балансе, который достигается за счет деятельности различных систем организма (сердечно-сосудистой, дыхательной, выделительной, эндокринной), энергетического, водно-солевого и белкового обменов. При этом количество образующегося тепла равно количеству тепла, отдаваемого организмом в окружающую среду за один и тот же промежуток времени. Создаются оптимальные условия для работы всех функциональных систем организма, обеспечивается высокий уровень работоспособности.

Нагревающий микроклимат – сочетание параметров микроклимата, при котором имеет место нарушение теплообмена человека с окружающей средой, проявляющееся в накоплении тепла в организме и (или) в увеличении доли потерь тепла испарением пота (>30%) в общей структуре теплового баланса. В условиях нагревающего микроклимата при повышении температуры воздуха и окружающих поверхностей происходит перегревание организма, механизм терморегуляции способствует увеличению теплоотдачи, которая осуществляется через систему кровообращения и путем потоотделения. Происходит усиление тока крови через кожу вследствие расширения кожных сосудов, в результате увеличивается теплопроводность тканей и температура кожи, что способствует большему рассеиванию тепла в окружающую среду.

Тепловое воздействие на организм вызывает рефлекторное повышение секреции потовых желез, что обеспечивает значительное увеличение теплоотдачи и, в свою очередь, приводит к обеднению организма водой, с потом теряются соли натрия, калия, кальция, фосфора и др. Нарушается водно-солевой баланс, снижаются резервные возможности организма, характерен высокий уровень простудных заболеваний (ангины, бронхиты, ОРЗ, невралгии), наблюдаются стойкие изменения в нервной, эндокринной, сердечно – сосудистой системах.

Охлаждающий микроклимат – сочетание параметров микроклимата, при котором имеет место изменение теплообмена организма (превышение общей



теплоотдачи организма в окружающую среду), приводящее к образованию общего или локального дефицита тепла в организме за счет снижения температуры глубоких и (или) поверхностных слоев тканей организма. В условиях охлаждающего микроклимата, возникающих при понижении температуры окружающего воздуха, повышении его подвижности и относительной влажности, реакции организма направлены на уменьшение теплоотдачи и увеличение количества тепла, вырабатываемого организмом. Уменьшение теплоотдачи происходит в результате спазм кровеносных сосудов поверхностных тканей и снижения их температуры. Под влиянием низких и пониженных температур воздуха могут развиваться ознобления, обморожения, невриты, радикулиты и др. При длительном охлаждении развиваются заболевания периферийной нервной, мышечной систем, суставов. Организм становится более восприимчивым к гриппу, ангине, пневмонии, катару верхних дыхательных путей и др.

Для организма чрезвычайно опасно излучение лучистой энергии (инфракрасное излучение), возникающее от различных сильно нагретых материалов, оборудования. Такое излучение легко поглощается и проникает в ткани человеческого тела, вызывая повышение температуры тела и внутренних органов, нарушение функционального состояния центральной нервной системы, усиление секреторной деятельности желудка, поджелудочной и слюнных желез, уменьшение нервно-мышечной возбудимости.

Для исключения вредного влияния микроклимата на организм человека и создания нормальных условий в рабочей зоне производственных помещений параметры воздушной среды должны соответствовать гигиеническим нормативам. СанПиН 2.2.4.548-96 регламентирует температуру воздуха, его относительную влажность, скорость движения воздуха, интенсивность теплового излучения для рабочей зоны в виде оптимальных и допустимых величин с учетом сезонов года и тяжести выполняемых работ. При этом под гигиеническими нормативами условий



труда понимаются такие уровни вредных производственных факторов (температуры, относительной влажности, скорости движения воздуха и избытка явного тепла), которые при ежедневной (кроме выходных дней) работе, но не более 40 часов в неделю в течение всего рабочего стажа, не должны вызывать заболеваний или отклонений в состоянии здоровья, обнаруживаемых современными методами исследований, в процессе работы или отдаленные сроки жизни настоящего и последующих поколений.

Выполняемые работы по уровню энергозатрат делятся на легкие (категории Ia и Ib), средней тяжести (категории IIa и IIб) и тяжелые (категории III);

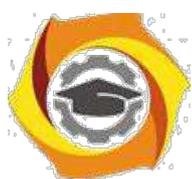
К категории Ia относятся работы с интенсивностью энергозатрат организма до 120 ккал/ч (до 139 Вт), производимые сидя и сопровождающиеся незначительным физическим напряжением.

К категории Ib относятся работы с интенсивностью энергозатрат 121-150 ккал/ч (140-174 Вт), производимые сидя, стоя или связанные с ходьбой и сопровождающиеся некоторым физическим напряжением.

К категории IIa относятся работы с интенсивностью энергозатрат организма 151-200 ккал/ч (175-232 Вт), связанные с постоянной ходьбой, перемещением мелких (до 1 кг) изделий или предметов в положении стоя или сидя и требующие определенного физического напряжения.

К категории IIб относятся работы с интенсивностью энергозатрат организма 201-250 ккал/ч (233-290 Вт), связанные с ходьбой, перемещением и переноской тяжестей до 10 кг и сопровождающиеся умеренным физическим напряжением.

К категории III относятся работы с интенсивностью энергозатрат организма более 250 ккал/ч (более 290 Вт), связанные с постоянными передвижениями, перемещением и переноской значительных (свыше 10 кг) тяжестей и требующие больших физических усилий.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

1.2. Оптимальные и допустимые величины параметров микроклимата



Оптимальные микроклиматические условия установлены по критериям оптимального теплового и функционального состояния человека, которые обеспечивают общее и локальное ощущение теплового комфорта в течение 8-часового рабочего дня при минимальном напряжении механизмов терморегуляции, не вызывают отклонений в состоянии здоровья, создают предпосылки для высокого уровня работоспособности и являются предпочтительными на рабочих местах. Их необходимо соблюдать на рабочих местах производственных помещений, на которых выполняются работы операторского типа, связанные с нервно-эмоциональным напряжением.

Оптимальные параметры микроклимата на рабочих местах должны соответствовать значениям, приведенным в табл. 1.1, применительно к выполнению работ различных категорий в холодный и теплый периоды года. При этом холодным периодом года считается период, характеризуемый среднесуточной температурой наружного воздуха, равной $+10^{\circ}\text{C}$ и ниже, а теплым периодом года – период года, характеризуемый среднесуточной температурой наружного воздуха выше $+10^{\circ}\text{C}$.

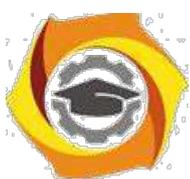


Таблица 1.1

Оптимальные величины параметров микроклимата на рабочих
местах производственных помещений

Период года	Категория работ по уровню энергозатрат	Температура воздуха, °С	Температура поверхностей, °С	Скорость движения воздуха, м/с (не более)
Холодный	Легкая – Ia	22-24	21-25	0,1
	Легкая – Ib	21-23	20-24	0,1
	Средней тяжести – IIa	19-21	18-22	0,2
	Средней тяжести – IIб	17-19	16-20	0,2
	Тяжелая – III	16-18	15-19	0,3
Теплый	Легкая – Ia	23-25	22-26	0,1
	Легкая – Ib	22-24	21-25	0,1
	Средней тяжести – IIa	20-22	19-23	0,2
	Средней тяжести – IIб	19-21	18-22	0,2
	Тяжелая – III	18-20	17-21	0,3

Примечание: относительная влажность воздуха во всех случаях составляет 40-60%.

Допустимые микроклиматические условия установлены по критериям допустимого теплового и функционального состояния человека в течение 8-часового рабочего дня. Они не вызывают повреждений или нарушений состояния здоровья, но могут приводить к возникновению общих и локальных ощущений теплового дискомфорта, напряжению механизмов терморегуляции, ухудшению самочувствия и понижению работоспособности. Допустимые показатели устанавливаются в случаях, когда по технологическим требованиям, технически и экономически обоснованным причинам не могут быть обеспечены оптимальные величины.

Допустимые величины параметров микроклимата на рабочих местах должны соответствовать значениям, приведенным в табл. 1.2, применительно к выполнению работ различных категорий в холодный и теплый периоды



года, при этом перепад температуры воздуха по высоте должен быть не более 3 °С, а перепад температуры воздуха по горизонтали и изменения ее в течение рабочей смены не должны превышать 4–6°С в зависимости от категории работ.

Оптимальные и допустимые показатели микроклимата соответствуют безопасным классам условий труда (1–й и 2–й классы). При отклонении показателей от безопасных они могут соответствовать вредным и опасным классам условий труда (3-й и 4-й классы).



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

14

Таблица 1.2

Допустимые величины показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работ по уровню энергозатрат	Температура воздуха, °С		Температура поверхности, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с		Скорость движения воздуха при температуре 26-28°С, м/с
		диапазон ниже оптимальных величин	диапазон выше оптимальных величин			для диапазонов температур ниже оптимальных величин, не более	для диапазонов температур выше оптимальных величин, не более	
Холодный	Легкая – Ia	20,0-21,9	24,1-25,0	19,0-26,0	15-75*	0,1	0,1	0,1-0,2
	Легкая – Ib	19,0-20,9	23,1-24,0	18,0-25,0	15-75	0,1	0,2	0,1-0,3
	Ср. тяжести – Pa	17,0-18,9	21,1-23,0	16,0-24,0	15-75	0,1	0,3	0,2-0,4
	Ср. тяжести – Pb	15,0-16,9	19,1-22,0	14,0-23,0	15-75	0,2	0,4	0,2-0,5
	Тяжелая – III	13,0-15,9	18,1-21,0	12,0-22,0	15-75	0,2	0,4	0,2-0,5
Теплый	Легкая – Ia	21,0-22,9	25,1-28,0	20,0-29,0	15-75*	0,1	0,2	0,1-0,2
	Легкая – Ib	20,0-21,9	24,1-28,0	19,0-29,0	15-75*	0,1	0,3	0,1-0,3
	Ср. тяжести – Pa	18,0-19,9	22,1-27,0	17,0-28,0	15-75*	0,1	0,4	0,2-0,4
	Ср. тяжести – Pb	16,0-18,9	21,1-27,0	15,0-28,0	15-75*	0,2	0,5	0,2-0,5
	Тяжелая – III	15,0-17,9	20,1-26,0	14,0-27,0	15-75*	0,2	0,5	0,2-0,5



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

* Примечание. Максимально допустимые величины относительной влажности воздуха при температуре воздуха на рабочих местах 25 °С и выше находятся в пределах:
при 25°С-70%, 26°С-65%, 27°С-60%, 28°С-55%.



Допустимые величины интенсивности теплового облучения работающих на рабочих местах от производственных источников, нагретых до темного свечения, в зависимости от площади облучаемой поверхности тела, должны соответствовать значениям:

50 % и более – 35 Вт/м^2 ,

25% и менее – 100 Вт/м^2 .

При наличии теплового облучения работающих температура воздуха на рабочих местах не должна превышать, в зависимости от категории работ, следующих величин:

25°C – при легкой Ia,

24°C – при легкой Ib,

22°C – при средней тяжести IIa,

21°C – при средней тяжести IIб,

20°C – при тяжелой III

В производственных помещениях, в которых допустимые величины показателей микроклимата невозможно установить из-за технологических требований к производственному процессу или экономически обоснованной нецелесообразности, условия микроклимата следует рассматривать как вредные и опасные, в этом случае должны быть использованы защитные мероприятия.

Для регламентации времени работы (непрерывного или суммарного) в пределах рабочей смены в условиях микроклимата с температурой воздуха на рабочих местах выше или ниже допустимых величин рекомендуется руководствоваться данными, приведенными в табл. 1.3.

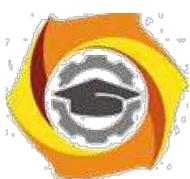


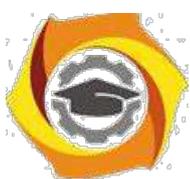
Таблица 1.3

Время пребывания на рабочих местах

Температура воздуха на рабочем месте, °С	Время пребывания при выполнении работ, час				
	Ia	Iб	IIa	IIб	III
32,5	1	1	–	–	–
32,0	2	2	–	–	–
31,5	2,5	2,5	1	1	–
31,0	3	3	2	2	–
30,5	4	4	2,5	2,5	1
30,0	5	5	3	3	2
29,5	5,5	5,5	4	4	2,5
29,0	6	6	5	5	3
28,5	7	7	5,5	5,5	4
28,0	8	8	6	6	5
27,5	–	–	7	7	5,5
27,0	–	–	8	8	6
26,5	–	–	–	–	7
26,0	–	–	–	–	8
При температуре воздуха ниже допустимых величин					
6	–	–	–	–	1
7	–	–	–	–	2
8	–	–	–	1	3
9	–	–	–	2	4
10	–	–	1	3	5
11	–	–	2	4	6
12	–	1	3	5	7
13	1	2	4	6	8
14	2	3	5	7	–
15	3	4	6	8	–
16	4	5	7	–	–
17	5	6	8	–	–
18	6	7	–	–	–
19	7	8	–	–	–
20	8	–	–	–	–

1.3. Классы условий труда по параметрам микроклимата

Отнесение условий труда к классу (подклассу) условий труда при воздействии параметров микроклимата осуществляется с учётом используемого на рабочих местах технологического оборудования, являющегося искусственным источником тепла и (или) холода, и на основе



измерений температуры воздуха, влажности воздуха, скорости движения воздуха и (или) теплового излучения в производственных помещениях на всех местах пребывания работника в течение рабочего дня (смены) с учетом характеристики микроклимата (нагревающий, охлаждающий) путём сопоставления фактических значений параметров микроклимата с нормативными.

Отнесение условий труда к классу (подклассу) осуществляется в следующей последовательности:

на первом этапе класс условий труда определяется по температуре воздуха;

на втором этапе класс условий труда корректируется в зависимости от влажности воздуха, скорости движения воздуха и (или) интенсивности теплового излучения.

Если температура воздуха или влажность, или скорость движения воздуха в помещении с нагревающим микроклиматом не соответствует допустимым величинам, отнесение условий труда к тому или иному классу осуществляется по индексу тепловой нагрузки среды (ТНС – индекс). ТНС – индекс определяется на основе величин температуры смоченного термометра аспирационного психрометра и температуры внутри зачерненного шара. В настоящей работе эти измерения не будут производиться.

При воздействии охлаждающего микроклимата отнесение к тому или иному классу осуществляется отдельно по температуре воздуха, скорости движения воздуха, влажности воздуха, тепловому излучению в соответствии с табл. 1.4.

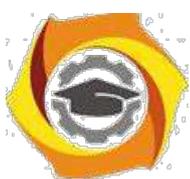
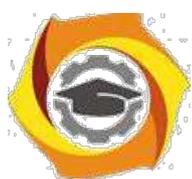


Таблица 1.4.

Отнесение условий труда по классу (подклассу) условий труда при
воздействии параметров микроклимата при работе в помещении с
охлаждающим микроклиматом

Показатель	Категория работ	Класс условий труда						
		оптимальный 1	допустимый 2	вредный 3				опасный 4
				3				
				3.1	3.2	3.3	3.4	
Температура воздуха, °С	Ia	22,0-24,0	21,9-20,0	19,9-18,0	17,9-16,0	15,9-14,0	13,9-12,0	<12,0
	Iб	21,0-23,0	20,9-19,0	18,9-17,0	16,9-15,0	14,9-13,0	12,9-11,0	<11,0
	IIa	19,0-21,0	18,9-17,0	16,9-14,0	13,9-12,0	11,9-10,0	9,9-8,0	<8,0
	IIб	17,0-19,0	16,9-15,0	14,9-13,0	12,9-11,0	10,9-9,0	8,9-7,0	<7,0
	III	16,0-18,0	15,9-13,0	12,9-12,0	11,9-10,0	9,9-8,0	7,9-6,0	<6,0
Скорость движения воздуха, м/с	Ia	≤0,1	≤0,1	Учитывается в температурной поправке на охлаждающее действие ветра. При скорости движения воздуха, большей или равной 0,6 м/с, с условия труда признаются вредными для всех категорий работ				
	Iб	≤0,1	≤0,1					
	IIa	≤0,2	≤0,1					
	IIб	≤0,2	≤0,2					
	III	≤0,3	≤0,2					
Влажность воздуха, %	I–III	60-40	15–<40; >60–75	<15–10	<10	–	–	–
Интенсивность теплового излучения, Вт/м ²	I–III	–	≤140	141–1500	1501–2000	2001–2500	2501–2800	>2800
Экспозиционная доза теплового облучения, Вт·ч	I–III	–	500	1500	2600	3800	4800	>4800

² В таблице приведена температура воздуха применительно к оптимальным величинам скорости его



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

движения. При увеличении скорости движения воздуха на рабочем месте на 0,1 м/с оптимальную температуру воздуха, приведенную в настоящей таблице, следует повысить на 0,2 °С.



Класс (подкласс) условий труда устанавливается по параметру микроклимата, имеющему наиболее высокий класс (подкласс) условий труда.

1.4. Экспериментальная часть.

1.4.1. Требования к организации гигиенического контроля параметров микроклимата

Измерения показателей микроклимата в целях контроля их соответствия гигиеническим требованиям должны производиться в холодный период года – в дни с температурой наружного воздуха, отличающейся от средней температуры наиболее холодного месяца зимы не более чем на 5 °С, в теплый период года – в дни с температурой наружного воздуха, отличающейся от средней максимальной температуры наиболее жаркого месяца не более чем на 5 °С. Частота измерений в оба периода года определяется стабильностью производственного процесса, функционированием технологического и санитарно-технического оборудования.

При выборе участков и времени измерения необходимо учитывать все факторы, влияющие на микроклимат рабочих мест. Измерения показателей микроклимата следует проводить 3 раза в смену (в начале, середине и конце). При колебаниях показателей, связанных с технологическими и другими причинами, необходимо проводить дополнительные измерения при наибольших и наименьших величинах термических нагрузок на работающих. Измерения следует проводить на рабочих местах; если рабочим местом является несколько участков производственного помещения, то измерения осуществляются на каждом из них.

При наличии источников локального тепловыделения, охлаждения или влаговыведения измерения следует проводить на каждом рабочем месте в точках, минимально и максимально удаленных от источников термического



воздействия. В помещениях с большой плотностью рабочих мест при отсутствии источников локального тепловыделения, охлаждения или влаговыведения участки измерения температуры, относительной влажности и скорости движения воздуха должны распределяться равномерно по площади помещения:

до 100 м^2 – 4 участка,

$100\text{-}400 \text{ м}^2$ – 8 участков,

свыше 400 м^2 – расстояние между участками не должно превышать 10 м.

Точки измерения:

а) при работах, выполняемых сидя, температуру и скорость движения воздуха следует измерять на высоте 0,1 и 1,0 м, а относительную влажность воздуха – на высоте 1,0 м от пола или рабочей площадки;

б) при работах, выполняемых стоя, температуру и скорость движения воздуха – на высоте 0,1 и 1,5 м, а относительную влажность воздуха – на высоте 1,5 м;

в) при наличии источников теплового излучения интенсивность его на рабочих местах необходимо измерять от каждого источника, располагая датчик прибора перпендикулярно падающему потоку. Измерения следует проводить на высоте 0,5; 1,0 и 1,5 м от пола или рабочей площадки.

1.4.2. Приборы и методика измерений параметров микроклимата

Температуру воздуха измеряют с помощью термометров различной конструкции и термографов. В зависимости от назначения и пределов измерения температур применяются жидкостные термометры, у которых в качестве рабочей жидкости используется спирт, толуол, ртуть. Принцип действия этих приборов основан на изменении объема рабочей жидкости при нагревании или охлаждении воздуха.



Температуру воздуха при наличии источников теплового излучения и воздушных потоков на рабочем месте следует измерять аспирационными психрометрами Ассмана. Приборы для измерения температуры должны иметь диапазон измерения от -30 до $+50$ °С с предельным отклонением $\pm 0,2$ °С.

Для непрерывной регистрации температуры окружающего воздуха в течение определенного промежутка времени применяют термографы различной конструкции.

Влажность воздуха определяется содержанием в нем водяных паров. В атмосферном воздухе каждому значению температуры воздуха соответствует вполне определенное максимальное содержание водяного пара, а именно: чем выше температура воздуха, тем больше максимальное содержание водяного пара в единице объема. Когда количество водяного пара в воздухе при определенной температуре достигает своего максимального значения, воздух считается насыщенным.

Различают следующие виды влажности воздуха:

1) абсолютная влажность (A) – упругость водяных паров в мм.рт столба или масса водяных паров, содержащаяся при определенной температуре в единице объема воздуха (г/м^3);

2) максимальная влажность (P_C) – упругость водяных паров при полном насыщении воздуха влагой в мм рт. столба или максимально возможное количество водяных паров в воздухе при данных температуре и давлении (г/м^3);

3) относительная влажность (ϕ) – отношение абсолютной влажности к максимальной, выраженное в процентах,

$$\phi = \frac{A}{P_C} \cdot 100\%.$$

На самочувствие людей оказывает влияние степень насыщения воздуха парами влаги, т.е. относительная влажность, которая и определяет



возможность и эффективность теплоотдачи испарением влаги с поверхности кожи работающего.

При наличии источников теплового излучения и воздушных потоков на рабочем месте относительную влажность следует измерять аспирационным психрометром Ассмана.

Приборы для измерения относительной влажности должны иметь диапазон измерения по сухому термометру от -30 до $+50$ °С, по смоченному термометру от 0 до $+50$ °С с предельным отклонением $\pm 0,2$ °С.

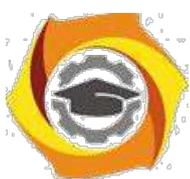
Психрометр Ассмана (рис. 1) состоит из двух ртутных термометров высокой точности – «сухого» и «смоченного», помещенных с целью экранирования от внешнего теплового облучения в двойные латунные трубки с зеркальной наружной поверхностью. По этим трубкам встроенный в верхней части психрометра вентилятор с пружинным затвором прокачивает вдоль термометров воздух, создавая вдоль них стандартный воздушный поток со скоростью 4 м/с. Резервуар «смоченного» термометра обернут кусочком батиста, который увлажняется однократно при проведении одного измерения. С поверхности увлажненного батиста под действием вертикальной аспирации (движения) воздуха испаряется влага, создавая вблизи термометра среду насыщенных водяных паров. Поэтому показания «смоченного» термометра есть $t_{вл}$, при которой данная влажность становится максимальной.

Благодаря обтеканию вертикального потока воздуха «сухой» термометр будет показывать температуру воздуха в расчетной точке, а показания «увлажненного» термометра будут зависеть от влажности воздуха тем меньше, чем ниже влажность, так как с уменьшением влаги в воздухе возрастает испарение воды с увлажненной ткани и поверхность резервуара термометра охлаждается в большей мере.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Порядок работы и определения относительной влажности с помощью психрометра Ассмана следующий:



- 1) смачивают водой из пипетки батист, облегающий резервуар «увлажненного» термометра;
- 2) заводят механизм вентилятора почти до отказа;
- 3) психрометр подвешивают на кронштейне в рабочей зоне на высоте измерения 1,0 или 1,5 м;
- 4) на 4-й минуте после пуска вентилятора производят отсчет по «сухому» и «увлажненному» термометрам;
- 5) относительная влажность воздуха определяется по показаниям «сухого» и «увлажненного» термометров по психрометрической таблице (табл. 1.5).

Температуру поверхностей следует измерять в случаях, когда рабочие места удалены от них на расстояние не более двух метров. Измерения проводятся контактными приборами типа электротермометров или дистанционными типа пирометров; приборы должны иметь диапазон измерения от 0 до +50 °С с предельным отклонением $\pm 0,5$ °С.

Интенсивность теплового облучения следует измерять приборами, обеспечивающими угол видимости датчика, близкий к полусфере (не менее 160°), и чувствительными в инфракрасной и видимой области спектра (актинометры, радиометры и т.д.), которые должны иметь при диапазоне измерения от 10 до 350 Вт/м² предельное отклонение $\pm 5,0$ Вт/м², в диапазоне более 350 Вт/м² - $\pm 50,0$ Вт/м².

Устройство актинометра основано на принципе термоприемника; в актинометре использована так называемая термобатарейка – пластинка, состоящая из ряда термоэлементов, спаянных между собой. Под действием лучистой энергии в термобатарее возникает термоэлектрический ток, который измеряется вмонтированным в прибор гальванометром. В данной работе измерений интенсивности теплового излучения производиться не будет.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

24

Таблица 1.5

Таблица для вычисления относительной влажности по аспирационному психрометру Ассмана

°C по сухому термометру	°C по увлажненному термометру																				
	Относительная влажность																				
	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
12	29	38	18	57	68	79	69	100													
13	28	31	10	19	59	69	79	89	100												
14	17	25	33	42	51	36	76	79	89	100											
15		20	27	36	44	52	61	71	80	90	100										
16		15	22	38	37	46	64	83	71	81	90	100									
17			17	24	32	39	47	55	64	72	81	90	100								
18			13	20	27	34	44	49	56	65	73	82	91	100							
19				15	22	29	36	43	30	58	66	74	83	91	100						
20					18	24	38	37	43	52	59	66	74	63	91	100					
21					14	20	26	32	39	46	53	60	67	75	83	92	100				
22						16	22	20	34	40	47	54	61	68	76	84	92	100			
23						13	18	24	30	36	42	48	55	62	69	76	84	92	100		
24							15	20	26	31	37	43	49	56	63	70	77	84	92	100	
25								17	22	27	33	38	44	50	57	63	70	77	84	92	100
26								14	19	24	29	34	40	46	52	57	64	71	77	85	92
27									16	21	25	30	36	41	47	52	58	65	73	78	85
28									13	18	22	27	32	37	42	48	53	58	65	72	78
29									11	15	19	24	28	33	38	43	49	54	60	6	72
30										13	17	21	25	30	34	39	44	40	55	61	67
31										10	14	18	22	27	35	36	40	45	50	56	62
32											12	16	20	24	28	32	37	41	46	51	57
33											10	14	17	21	25	29	33	38	42	47	52
34												12	15	19	22	26	30	34	39	43	48



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

35											10	13	16	20	24	27	31	35	40	44
----	--	--	--	--	--	--	--	--	--	--	----	----	----	----	----	----	----	----	----	----



Скорость движения воздуха измеряется анемометрами различной конструкции. Например, цифровой анемометр AR816 (рис. 2,а) предназначен для измерения скорости и температуры воздушного потока. Нажатием кнопки «MODE» включают прибор. Держа кнопку «MODE» в течение 3 секунд нажатием кнопки «SET» выбирают желаемые единицы измерения. При вращении крыльчатки на экране будет отображаться мгновенная скорость воздушного потока и его температура. Диапазон измерения скорости: $0 - 30 \frac{\text{М}}{\text{с}}$ с разрешением $0,1 \frac{\text{М}}{\text{с}}$.

Термоанемометр DT-318 предназначен также для измерения скорости и температуры воздушного потока (рис. 2, б). Диапазон измерения скорости: $1,0 - 30,0 \frac{\text{М}}{\text{с}}$ с разрешением $0,01 \frac{\text{М}}{\text{с}}$. Для включения или выключения прибора следует нажать кнопку «» (рис. 2,б). На основном индикаторе отображается измеренное значение скорости, на дополнительном – температуры воздушного потока.

1.5. Порядок проведения работы

1. Ознакомиться с устройством приборов и методикой измерения параметров микроклимата.
2. Измерить температуру и относительную влажность воздуха с помощью аспирационного психрометра Ассмана.
3. Измерить скорость движения воздуха с помощью анемометра.
4. Сравнить измеренные параметры микроклимата с гигиеническими нормативами.
5. Сделать заключение о состоянии условий труда по параметрам микроклимата и отнести их к тому или иному классу (подклассу).



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

6. Все данные занести в таблицу. Таблица должна быть такой, чтобы студент с её помощью мог ответить на все вопросы по данной работе.

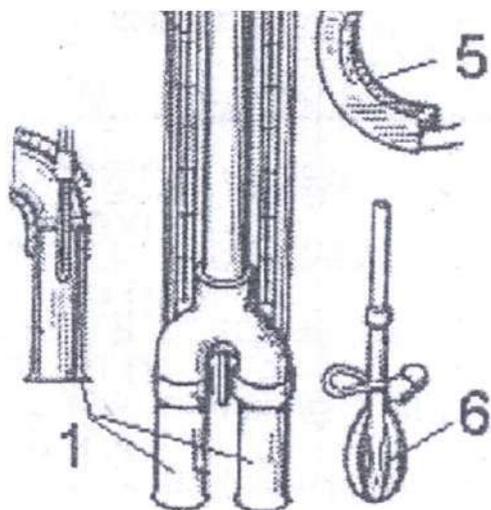
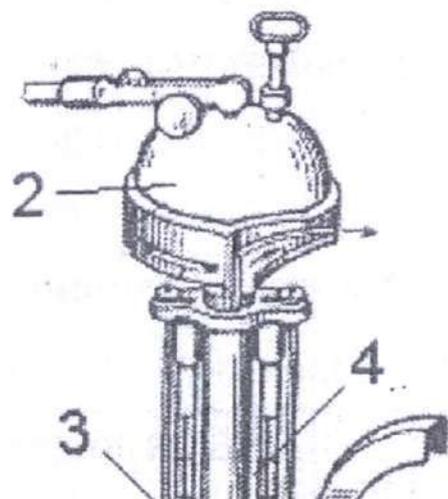
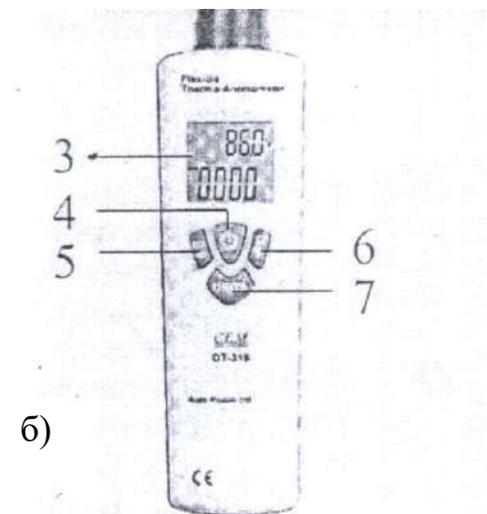
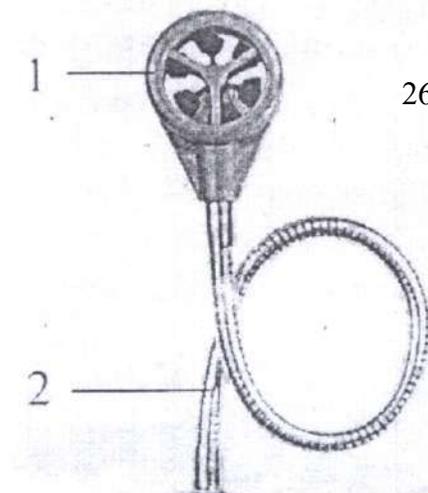


Рис. 1 Аспирационный психрометр Ассмана.
1 – трубки с двойными стенками; 2 – аспиратор;
3 – «сухой» термометр; 4 – «влажный» термометр;
5 – ветровой предохранитель; 6 – пипетка с водой



а)



б)

Рис. 2 а) цифровой анемометр AR-816;
б) термоанемометр DT-318:

1 – крыльчатка со встроенным термистором; 2 – гибкий зонд

ЖК-дисплей; 3 – кнопка - включение/выключение прибора;
5 – кнопка MAX/MIN - определение максимальных/минимальных значений
температуры и скорости воздушного потока;

6 – кнопка - удержание показаний на дисплее, включение/выключение
подсветки; 7 – кнопка UNITS - выбор единиц измерения температуры и



Негосударственное частное образовательное учреждение
высшего образования
«Технический **У**ниверситет»

скорости воздушного потока



Контрольные вопросы

1. Какими параметрами характеризуется микроклимат производственных помещений?
2. Что такое комфортный, нагревающий и охлаждающий микроклимат?
3. Каковы отличия оптимальных и допустимых параметров микроклимата?
4. От чего зависят допустимые и оптимальные нормы параметров микроклиматов?
5. Какова последовательность отнесения условий труда по микроклимату к тому или иному классу (подклассу)?
6. Перечислите основные требования к организации гигиенического контроля параметров микроклимата?
7. Приборы и методика измерений параметров микроклимата?



Лабораторная работа 2

Производственное освещение

Цель работы: 1. Изучить существующие санитарно-гигиенические нормы и требования к производственному освещению.

2. Изучить методики измерения параметров освещения и ознакомиться с приборами.

3. Научиться оценивать состояние производственного освещения на основании проведенных измерений.

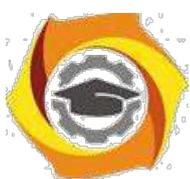
3.1. Общие сведения и нормирование

Естественное освещение может осуществляться через окно и другие боковые светопроемы в наружных стенах (боковое освещение), через застекленные световые фонари и перекрытия (верхнее освещение) или обоими способами одновременно (комбинированное освещение).

Искусственное освещение может быть двух видов: общее, когда осветительные устройства размещены таким образом, чтобы обеспечить достаточную освещенность в зоне производства работ и в проходах, и комбинированное, когда кроме общего освещения устанавливаются светильники местного освещения для создания более высоких уровней освещенности на рабочих местах, где выполняется напряженная зрительная работа.

Для некоторых географических районов России, а также для работ очень высокой точности СНиП 23-05-95 рекомендует использовать систему совмещенного освещения, то есть естественное освещение совмещать с искусственным.

Для нормирования искусственного освещения используется освещенность (Е) в люксах (лк). Нормы освещенности установлены СНиП 23-05-95[6] и приведены в таблице 3.1. В зависимости от точности все

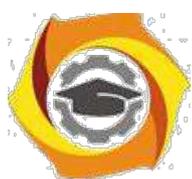


зрительные работы разбиты на восемь разрядов (I, II, ... VIII). В зависимости от сочетания фона и контрастности в пределах одной точности зрительные работы разбиты на подразряды (а, б, в, г). При наличии в одном помещении рабочих и вспомогательных зон следует предусматривать локализованное общее освещение рабочих зон и менее интенсивное освещение вспомогательных зон, относя их к разряду VIII а.

Таблица 2.1

**Нормированные значения освещенности при искусственном освещении
в производственных помещениях**

Характеристика зрительной работы	Наименьший или эквивалентный размер объекта различения, мм	Разряд зрительной работы	Подразряд зрительной работы	Контраст объекта с фоном	Характеристика фона	Освещенность, лк		
						при комбинированного освещения		при системе общего освещения
						всего	в том числе от общего	
1	2	3	4	5	6	7	8	9
Наивысшей точности	Менее 0,15	I	а б в г	Малый	Темный	5000	500	-
						4500	500	-
				Малый	Средний	4000	400	1250
				Средний	Темный	3500	400	1000
				Малый	Светлый	2500	300	750
				Средний	Средний			
				Большой	Темный	2000	200	600
				Средний	Светлый	1500	200	400
Очень высокой точности	от 0,15 до 0,30	II	а б в г	Малый	Темный	4000	400	-
						3500	400	-
				Малый	Средний	3000	300	750
								600
				Малый	Светлый	2000	200	500
				Средний	Средний			
				Большой	Темный	1500	200	400
				Средний	Светлый	1000	200	300
Высокой точности	От 0,30 до 0,50	III	а б в	Малый	Темный	2000	200	500
						1500	200	400
				Малый	Средний	1000	200	300
				Средний	Темный	750	200	200
				Малый	Светлый	750	200	300
				Средний	Средний			
				Большой	Темный	600	200	200



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

			г	Средний	Светлый	400	200	200
--	--	--	---	---------	---------	-----	-----	-----



					Большой Большой	Светлый Средний		
--	--	--	--	--	--------------------	--------------------	--	--

Продолжение таблицы 2.1

1	2	3	4	5	6	7	8	9
Средней точности	Свыше 0,50 до 1,0	IV	а	Малый	Темный	750	200	300
			б	Малый	Средний	500	200	200
			в	Средний Малый	Темный Светлый	400	200	200
			г	Средний Большой	Средний Темный			
Большой	Средний Большой	Светлый Светлый	-	-	200			
Малой точности	Свыше 1 до 5	V	а	Малый	Темный	400	200	300
			б	Малый	Средний	-	-	200
			в	Средний Малый	Темный Светлый	-	-	200
			г	Средний Большой	Средний Темный	-	-	200
Большой	Средний Большой	Светлый Светлый	-	-	200			
Большой	Средний	Светлый	-	-	200			
Грубая (очень малой точности)	Более 5	VI		Независимо от характе- ристик фона и контраста объекта с фоном				200
Работа со светящи- мися материалами в горячих цехах	Более 0,5	VII		Тоже		-	-	200
Общее наблюдение за ходом производ- ственного процесса: постоянное периодическое при постоянном пре- бывании людей в помещении периодическое при периодическом пребывании людей в помещении Общее наблюдение за инженерными комму-никациями		VIII	а	"		-	-	200
			б	"		-	-	75
			в	"		-	-	50
			г	"		-	-	20

Освещенность рабочей поверхности, создаваемая светильниками общего освещения в системе комбинированного, должна составлять не менее 10% нормируемой для комбинированного освещения. При этом освещенность должна быть не менее 200 лк при газоразрядных лампах, не менее 75 лк при лампах накаливания. Создавать освещенность от общего



освещения в системе комбинированного более 500 лк при газоразрядных лампах и более 150 лк при лампах накаливания без обоснования не следует.

В производственных помещениях освещенность проходов и участков, где работа не производится, должна составлять не более 25% нормируемой освещенности, но не менее 75 лк при газоразрядных лампах и не менее 30 лк при лампах накаливания.

Так как естественное освещение в помещении непрерывно и независимо от человека меняется, его невозможно характеризовать величиной освещенности рабочей поверхности. Для его оценки пользуются относительным показателем – коэффициентом естественной освещенности – КЕО; обозначается оно буквой "е" в отличие от освещенности "Е".

$$e = \frac{E_{вн}}{E_{нар}} \cdot 100, \%$$

где $E_{вн}$ – освещенность внутри помещения, лк;

$E_{нар}$ – освещенность вне помещения, на открытой площадке, лк.

В небольших помещениях при одностороннем боковом естественном освещении нормируется минимальное значение КЕО в точке, расположенной на пересечении вертикальной плоскости характерного разреза помещения и условной рабочей поверхности на расстоянии 1 м от стены, наиболее удаленной от световых проемов, а при двустороннем боковом освещении – в точке посередине помещения. При верхнем или комбинированном естественном освещении нормируется среднее значение КЕО в точках, расположенных на пересечении вертикальной плоскости характерного разреза помещения и условной рабочей поверхности (или пола). Первая и последняя точки принимаются на расстоянии 1 м от поверхности стен или осей колонн. Нормированные значения КЕО приведены в таблице 3.2

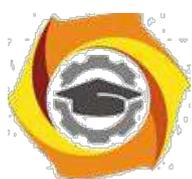
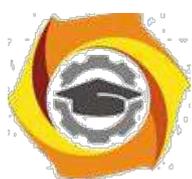


Таблица 2.2

Нормированное значение КЕО при естественном освещении
в производственных помещениях

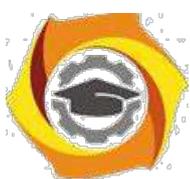
Характеристика зрительной работы	Наименьший или эквивалентный размер объекта различения, мм	Разряд зрительной работы	Естественное освещение		Совмещенное освещение	
			КЕО, е _н , %			
			при верхнем или комбинированном освещении	при боковом освещении	при верхнем или комбинированном освещении	при боковом освещении
1	2	3	4	5	6	7
Наивысшей точности	Менее 0,15	I	-	-	6,0	2,0
Очень высокой точности	Свыше 0,15 до 0,30	II	-	-	4,2	1,5
Высокой точности	Свыше 0,30 до 0,50	III	-	-	3,0	1,2
Средней точности	Свыше 0,50 до 1,00	IV	4	1,5	2,4	0,9
Малой точности	Свыше 1,00 до 5,00	V	3	1	1,8	0,6
Грубая (очень малой точности)	Более 5,00	VI	3	1	1,8	0,6
Общее наблюдение за ходом производственного процесса: - постоянное - периодическое при постоянном пребывании людей в помещении		VIII а	3	1	1,8	0,6
		VIII б	1	0,3	0,7	
		VIII в	0,7	0,2	0,5	0,2
Общее наблюдение за инженерными коммуникациями		VIII г	0,3	0,1	0,2	0,1

Помимо обеспечения нормативных показателей освещения к производственному освещению предъявляется ряд дополнительных



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

требований, связанных с переадаптацией глаза на различную освещенность.



Во-первых освещение должно быть равномерным. Это достигается для искусственного освещения равномерным распределением светильников по площади помещения, а для естественного – равномерным распределением световых проемов. Во-вторых источник света не должен ослеплять. Поэтому запрещается освещать голыми лампами, лампы должны находиться в специальной арматуре и это называется светильник или осветительное устройство. В-третьих между объектом различения и фоном должна быть хотя бы минимальная контрастность.

3.2. Классы условий труда по параметрам световой среды

Отнесение условий труда к классу (подклассу) условий труда при воздействии световой среды оценивается по освещенности рабочей поверхности для искусственного освещения и коэффициенту естественной освещенности для естественного освещения в соответствии с таблицей 3.3.

Таблица 2.3.

Классы условий труда при воздействии световой среды

Наименование показателя	Класс (подкласс) условий труда		
	допустимый	вредный	
	2	3.1	3.2
Искусственное освещение, Е, лк	$\geq E_n$	$\geq 0,5E_n$	$< 0,5E_n$
Естественное освещение КЕО(е), %	$\geq e_n$	$< e_n$	—

E_n и e_n - нормативные значения освещенности и коэффициента естественной освещенности.

При работе на открытой территории только в дневное время суток условия труда на рабочем месте по показателям световой среды признаются допустимыми условиями труда. При расположении рабочего места в нескольких рабочих зонах (в помещениях, на участках открытой территории) отнесение условий труда к тому или иному классу осуществляется с учетом



времени пребывания в разных зонах. На рабочих местах работников, в поле зрения которых присутствуют слепящие источники света, проводящих работу с объектами различения и рабочими поверхностями, обладающими направленно-рассеяным и смешанным отражением (металлы, пластмассы, стекло, глянцевая бумага), рекомендуется оценивать дополнительно такие показатели световой среды, как прямая и отраженная блесккость, пульсация.

3.3. Инструментарий и методы измерения показателей освещения

Перед измерением освещенности от искусственного освещения следует провести замену перегоревших ламп и чистку светильников. В противном случае это должно быть зафиксировано при оформлении результатов измерений.

Измерения КЕО проводят в помещениях, где мебель и оборудование не затеняет световые проемы, которые должны быть чистыми. В противном случае это должно быть зафиксировано при оформлении результатов измерений.

Перед измерениями выбирают и наносят контрольные точки для измерения освещенности на план помещения, сооружения или освещенного участка (или исполнительный чертеж осветительной установки) с указаниями размещения светильников.

Фотоэлектрические люксометры Ю-116 и Ю-117 предназначены для измерения освещенности, создаваемой естественным светом, лампами накаливания или любыми другими источниками света по СНиП 23-05-95. Принцип действия люксометра основан на явлении фотоэлектрического эффекта. Световой поток, падающий на фотоэлемент, вызывает фототок, который пропорционален величине этого светового потока. По отклонению стрелки гальванометра, проградуированного в люксах, судят о величине освещенности.

Люксометр Ю-116 предназначен для измерения освещенности от любых источников света, расположенных произвольно относительно его



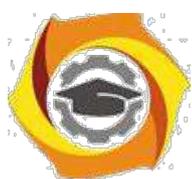
светоприемника. Для этого селеновый фотоэлемент изготовлен в форме круга, а на него надета полусферическая насадка из белой светорассеивающей пластмассы, обозначаемая буквой **К**. Для расширения диапазона измерения совместно с основной насадкой применяют три дополнительных, имеющие обозначения **М**, **Р** и **Т**. В результате совместного применения насадок в сочетаниях **КМ**, **КР** и **КТ** получается общий коэффициент ослабления светового потока – 10, 100 и 1000 соответственно.

Наиболее удобны для измерения освещенности цифровые люксметры, например, AR 813A. Он предназначен для измерения освещенности в диапазоне от 1 до 100 000лк. Он имеет 3 предела измерения с ручным переключением, автоматическую калибровку (установка нуля) и ЖК-дисплей с цифровым индикатором. Подвижное крепление датчика позволяет устанавливать фотоэлемент под различными углами.

Контрольные точки для измерения минимальной освещенности от рабочего освещения размещают в центре помещения, под светильниками, между светильниками и их рядами, у стен. При размещении контрольных точек на плане помещения их сетка не должна совпадать с сеткой размещения светильников. В случае совпадения сеток число контрольных точек на плане помещения целесообразно увеличить.

Контрольные точки при измерении коэффициента естественной освещенности помещений размещают на пересечении вертикальной плоскости характерного разреза помещения и условной рабочей поверхности (или пола). Первую и последнюю точки принимают на расстоянии 1м от поверхности наружных стен и внутренних перегородок (или оси колонн). Число контрольных точек зависит от ширины помещения и количества оконных проемов.

3.4 Проведение измерений и оценка результатов



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

При измерениях освещенности от любого источника света необходимо соблюдать следующие требования:



- на измерительный фотометрический датчик не должна падать тень от человека;

- измерительный прибор не должен располагаться вблизи сильных электромагнитных полей.

Освещенность на рабочем месте определяют прямыми измерениями в плоскости, указанной в нормах освещенности, или на рабочей поверхности оборудования. При комбинированном освещении рабочих мест освещенность измеряют сначала от светильников общего освещения, затем включают светильники местного освещения в их рабочем положении и измеряют суммарную освещенность от светильников общего и местного освещения.

При определении коэффициента естественной освещенности проводят одновременные измерения освещенности в контрольных точках внутри помещения $E_{вн}$ и наружной освещенности $E_{нар}$ на горизонтальной площадке, освещаемой всем светом небосвода в соответствии с методикой измерения.

После этого определяют класс условий труда в зависимости от параметров световой среды, пользуясь таблицей 3.3. Все данные заносят в таблицу. Таблица должна быть такой, чтобы студент с ее помощью мог ответить на все вопросы по данной работе.

Контрольные вопросы

1. Каким параметром нормируется искусственное освещение?
2. Каким параметром нормируется естественное освещение?
3. Размерность освещенности
4. Как рассчитать коэффициент естественной освещенности?
5. От чего зависит величина санитарной нормы для естественного и искусственного освещения?
6. Перечислите основные требования к производственному освещению?



7. Как производится отнесение условий труда к тому или иному классу (подклассу) при воздействии световой среды?
8. Как выбираются контрольные точки для измерений при оценке естественного и искусственного освещения?
9. Принцип работы люксметра?



Лабораторная работа 3

Производственный шум и вибрация

Цель работы: 1. Изучить принципы санитарно-гигиенического нормирования производственного шума и от чего зависит допустимый уровень шума.

1. Изучить приборы и методику измерения параметров шума.
2. Научиться оценивать состояние производственного шума на основании проведенных измерений.

4.1 Общие сведения и нормирование

Шум – совокупность звуков различной интенсивности, частоты и амплитуды колебаний, неблагоприятно воздействующих на органы слуха, центральную нервную и сердечно-сосудистую системы человека. Под воздействием шума понижается острота зрения, появляются головные боли и головокружение, изменяются ритмы дыхания и сердечной деятельности, повышается внутричерепное и кровяное давление, нарушается процесс пищеварения, происходит изменение объема внутренних органов, развивается тугоухость и глухота. Все эти патологии ведут к развитию профессиональной «шумовой» болезни.

Важным средством профилактики вредного воздействия шума на работающих является санитарное нормирование шума на рабочих местах, целью которого является установление научно-обоснованных допустимых уровней шума, которые при ежедневном систематическом воздействии в течение всего рабочего дня и многих лет не вызывают существенных заболеваний человека и не мешают его нормальной трудовой деятельности.

Нормирование допустимого уровня шума производится по санитарным нормам СН2.2.4/2.1.8.562-96[7] в зависимости от спектра и характера изменения шума по времени, вида и степени напряженности производственной деятельности человека, места пребывания человека во



время отдыха, лечения и т.д. Этим документом устанавливается классификация шумов по спектральным и временным характеристикам.

Шумы по характеру спектра подразделяются на широкополосные с непрерывным спектром шириной более одной октавы и тональные, в спектре которых имеются выраженные дискретные тона, измеряемые в 1/3 октавных полосах частот по превышению уровня в одной полосе над соседними не менее чем на 10дб.

По временным характеристикам шумы подразделяются на постоянные, уровень звука которых за 8-часовой рабочий день или за время измерения в помещениях жилых и общественных зданий, на территории жилой застройки изменяется во времени не более чем на 5дБА, и непостоянные, уровень звука которых изменяется более чем на 5 дБА.

В свою очередь, непостоянные шумы подразделяются на колеблющиеся во времени, прерывистые и импульсные.

Характеристикой постоянного шума на рабочих местах являются уровни звукового давления L в децибелах (дБ) в октавных полосах со среднегеометрическими частотами 31,5; 63; 125; 500; 1000; 2000; 4000; 8000 Гц, определяемые по формуле

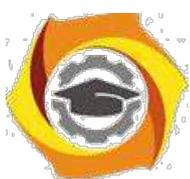
$$L = 20 \lg \frac{P}{P_0} \text{ (дБ)},$$

где P – среднеквадратичная величина звукового давления в октавной полосе, Па;

P_0 - $2 \cdot 10^{-5}$ Па – пороговое значение звукового давления (порог слышимости при частоте звука $f = 1000$ Гц)

В качестве характеристики постоянного широкополосного шума на местах допускается принимать уровень звука в дБА, измеренный на временной характеристике «медленно» шумомера, определяемый по формуле

$$L_A = 20 \lg \frac{P}{P_0} ,$$



где P_A – среднеквадратичная величина звукового давления с учетом коррекции «А» шумомера, Па.

Характеристикой непостоянного шума на рабочих местах является эквивалентный (по энергии) уровень звука в дБА. Эквивалентный уровень звука непостоянного шума – уровень звука постоянного широкополосного шума, который имеет такое же среднеквадратичное звуковое давление, что и данный непостоянный шум в течение определенного интервала времени.

При ориентировочной оценке за характеристику постоянного шума на рабочем месте допускается принимать уровень звука в дБА, при котором полученные значения соответствуют средней чувствительности органа слуха человека на различных частотах спектра (результаты объективных измерений приближены к субъективному восприятию органами слуха человека).

Предельно допустимые уровни звукового давления в октавных полосах на рабочих местах для широкополосного постоянного и непостоянного (кроме импульсного) шума принимаются по таблице 4.1, для тонального и импульсного шума – на 5 дБ меньше значений таблицы 4.1.

Таблица 3.1

Допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука на рабочих местах и для основных наиболее типичных видов трудовой деятельности

Виды трудовой деятельности, рабочее место	Уровни звукового давления (дБ) в октавных полосах со среднегеометрическими частотами (Гц)										Уровни звука и эквивалентные уровни звука в (дБА)
	31,5	63	125	250	500	1000	2000	4000	8000		
1	2	3	4	5	6	7	8	9	10	11	
1. Творческая, научная, рабочая деятельность в помещениях – КБ, расчетчиков, программистов ВЦ, лабораториях для	86	71	61	54	49	45	42	40	38	50	



Продолжение таблицы 3.1

1	2	3	4	5	6	7	8	9	10	11
2. Высококвалифицированная работа, измерительные и аналитические работы в лаборатории. Рабочие места в помещениях цехового управления, в рабочих комнатах, лабораториях.	93	79	70 63		58	55	52	50	49	60
3. Работа, требующая постоянного слухового контроля, диспетчерская работа. Машинописное бюро, участки точной сварки, в залах обработки информации на ВЦ	96	83	74	68	63	60	57	55	54	65
4. Работа, требующая сосредоточенности. Рабочие места за пультами в кабине технаблюдения и дистанционного управления, в лабораториях с шумным оборудованием, в помещениях ВЦ	103	91	83	77	73	70	68	66	64	75
5. Выполнение всех видов работ (за исключением в п.п. 1-4 и аналогичных им) на постоянных рабочих местах в производственных помещениях и на территории предприятий	107	95	87	82	78	75	73	71	69	80

При гигиеническом нормировании учитывают большую биологическую опасность тонального и импульсного шума путем ввода соответствующих поправок, зависящих от длительности воздействия этих шумов (табл.4.2).

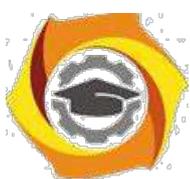


Таблица 3.2

Поправки к величине уровня шума в зависимости
от характера и времени воздействия, дБ

Время действия шума в смену (непрерывно или прерывисто), час	Характер шума	
	широкополосный	Тональный или импульсный
4	0	-5
1,5	+5	0
0,75	+10	+5
0,50	+15	+10
0,25	+20	+15

4.2. Классы условий труда при воздействии шума

Отнесение условий труда к тому или иному классу при воздействии шума осуществляется в зависимости от превышения фактических уровней шума над предельно допустимым уровнем (ПДУ). ПДУ звукового давления, звука и эквивалентного уровня звука на рабочих местах устанавливается в соответствии со следующей таблицей:

Наименование показателя	Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц										Уровень звука и эквивалентный уровень звука, дБА
	31,5	63	125	250	500	1000	2000	4000	8000		
Выполнение всех видов работ на рабочих местах	107	95	87	82	78	75	73	71	69	80	

При воздействии на работника постоянного шума отнесение условий труда к тому или иному классу осуществляется по результатам измерения уровней звукового давления (L) в октавных полосах со среднегеометрическими частотами 31,5; 63; 125; 250; 500; 1000; 2000; 4000; 8000 Гц.

Для оценки уровня шума допускается использовать уровень звука (L_A) в дБА в соответствии с таблицей 4.3.

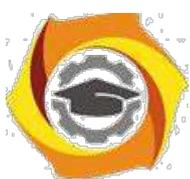


Таблица 3.3.

Классы условий труда при воздействии производственного шума

Наименование показателя, единица измерения	Класс (подкласс) условий труда					
	допусти- мый	вредный				опас- ный
		2	3.1	3.2	3.3	
Шум, эквивалентный уро- вень звука, дБА	≤ 80	$> 80-85$	$> 85-95$	$> 95-105$	$> 105-115$	> 115

При воздействии в течение рабочего дня (смены) на работника шумов с разными временными (постоянный шум, непостоянный шум – колеблющийся, прерывистый, импульсный) и спектральными (тональный шум) характеристиками в различных сочетаниях измеряют или рассчитывают эквивалентный уровень звука. Для получения сопоставимых данных измеренные или рассчитанные эквивалентные уровни звука импульсного и тонального шумов увеличиваются на 5дБА, после чего полученный результат можно сравнивать с ПДУ шума (табл. 4.3) без внесения в него понижающей поправки.

4.3. Инструментарий и методы измерения уровней шума

Системой стандартов безопасности труда предусмотрены пять методов измерения шумовых характеристик источников шума – два точных, два технических (эти методы требуют специальных измерительных помещений) и один ориентировочный метод. Наиболее широко распространен последний – для измерения шумовых характеристик источников шума на местах их эксплуатации.

Измерения в производственных помещениях выполняются при работе не менее 2/3 единиц технологического оборудования при наиболее характерном режиме работы. При этом должны работать вентиляция и другое оборудование, являющееся источником дополнительного шума.



На постоянных рабочих местах измерение производят в точках, соответствующих этим местам; на непостоянных рабочих местах – в точках (не менее трех), охватывающих возможно большую часть рабочей зоны.

Микрофон должен быть установлен в точке измерения на высоте 1,5 м над уровнем пола или рабочей площадки (при работе – стоя), либо на высоте головы работающего (при работе сидя), ориентирован в сторону источника шума и удален не менее чем на 0,5 м от человека, производящего измерения.

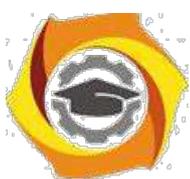
Для постоянных шумов измерения необходимо выполнять не менее 3-х раз в каждой точке, результаты которых затем усредняют. Минимальное расстояние до источника шума – 1 м.

Измерения на открытой площадке не должны производиться во время выпадения атмосферных осадков и при скорости ветра более 5 м/с. При скорости ветра 1–5 м/с следует применять ветрозащитный экран.

При измерении уровней шума следует применять шумомеры различных конструкций, среди которых измеритель шума и вибраций ВШВ-003 является наиболее современным и простым в использовании. ВШВ-003 устроен по принципу преобразования звуковых и механических колебаний исследуемых объектов в пропорциональные им электрические сигналы, усиливаемые и измеряемые с помощью измерительного прибора, проградуированного в децибелах.

В качестве преобразователя звуковых колебаний в электрические сигналы используется микрофон с капсулом М 101, имеющем в своем составе пьезоэлемент, преобразующий звуковые колебания.

Перед началом работы производится электрическая калибровка измерителя. Включение прибора производится переключателем «РОД РАБОТЫ» в положение «КОНТРОЛЬ ПИТАНИЯ», при этом стрелка показывающего прибора должна находиться в диапазоне 7-10 дБ шкалы $-\infty \div 10$ дБ. При положении **F** или **S** прибор готов к работе (осуществляется через пять минут после включения).



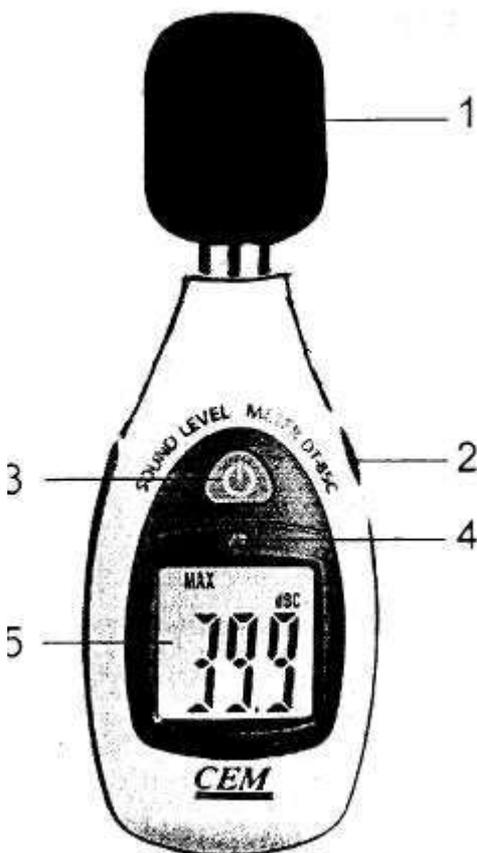
При измерении уровней звука на частотной характеристике **A** переключатели измерительного прибора устанавливают в положения:

ДЕЛИТЕЛЬ, dВ1- 80; ФИЛЬТРЫ - **A**;

ДЕЛИТЕЛЬ, dВ2- 50; РОД РАБОТЫ - **F**

Стрелку показывающего прибора выводят в сектор 0-10 шкалы децибел сначала переключателем ДЕЛИТЕЛЬ dВ1, а затем переключателем ДЕЛИТЕЛЬ dВ2. Для определения результатов измерения складывают показание светодиода по шкале dВ М 101 на передней панели прибора и показание на шкале децибел показывающего прибора.

При измерении уровней звукового давления в октавных полосах частот используется переключатель ФИЛЬТРЫ ОКТАВНЫЕ для включения необходимых октав. При этом переключатель ФИЛЬТРЫ устанавливают в положение ЛИН, кнопка ФИЛЬТРЫ ОКТАВНЫЕ – включена. Переключателем ДЕЛИТЕЛЬ dВ2 выводят стрелку показывающего прибора в сектор 0-10 dВ шкалы децибел, переключатель ДЕЛИТЕЛЬ dВ1 оставляют в том положении, которое он занимал при измерении уровней звука.



Результаты измерения определяют аналогично описанному выше. Для измерения уровня звука (дБА) используются цифровые шумомеры ДТ – 85А и ДТ -85С (рис.1). Особенности этих шумомеров состоят в том, что они удобны, определяют максимальное и минимальные значения уровня звука, имеют автоматическую подсветку дисплея.

Рис.1

- 1– микрофон с ветрозащитным экраном; 2 - кнопка для определения максимального/минимального значения; 3 – кнопка включение/выключение питания



4.4. Порядок проведения работы

1. Ознакомиться с устройством приборов и методикой измерения параметров производственного шума.
2. Найти предельно допустимые уровни (ПДУ) звукового давления, звука и эквивалентного уровня звука по заданию преподавателя.
3. Измерить уровень шума в октавных полосах и уровень звука в дБА на рабочем месте без звукоизоляции.
4. Измерить уровни шума на рабочем месте при наличии звукоизоляции. Оценить ее эффективность.
5. Сделать заключение о состоянии условий труда при воздействии шума и отнести их к тому или иному классу.
6. Все данные занести в таблицу. Таблица должна быть такой, чтобы студент с ее помощью мог ответить на все вопросы по данной работе.

Контрольные вопросы

1. Что такое шум?
2. Как подразделяются шумы по характеру спектра?
3. Как подразделяются шумы по временным характеристикам?
4. Какими параметрами характеризуется постоянный и непостоянный шум?
5. От чего зависят допустимые уровни шума?
6. Как осуществляется отнесение условий труда к тому или иному классу (подклассу) при воздействии шума?
7. Принцип работы шумомера и методика проведения измерений?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Методические указания для проведения лабораторных
работ по дисциплине

**Безопасность информационных процессов
в компьютерных системах и сетях**

Лабораторная работа № 1. Модели взаимодействия прикладной программы и программной закладки. Методы перехвата и навязывания информации. Методы внедрения программных закладок

Цель работы: ознакомиться с проблемами реализации политик безопасности в компьютерных системах на примере дискреционной модели.

Модели взаимодействия прикладной программы и программной закладки
Общая модель РПС можно представить в виде совокупности моделей, каждая из которых соответствует описанным выше типам РПС и характеризует действия злоумышленника, исходя из его образа мыслей и возможностей [ПАС].

1. *Модель «перехват».* Программная закладка встраивается (внедряется) в ПЗУ, ОС или прикладное программное обеспечение и сохраняет все или избранные фрагменты вводимой или выводимой информации в скрытой области локальной или удаленной внешней памяти прямого доступа. Объектом сохранения может быть клавиатурный ввод, документы, выводимые на принтер, или уничтожаемые файлы-документы. Для данной модели существенно наличие во внешней памяти места хранения информации, которое должно быть организовано таким образом, чтобы обеспечить ее сохранность на протяжении заданного промежутка времени и возможность последующего съема. Важно также, чтобы сохраняемая информация была каким-либо образом замаскирована от просмотра легальными пользователями.

2. *Модель «троянский конь».* Закладка встраивается в постоянно используемое программное обеспечение и по некоторому активизирующему событию моделирует сбойную ситуацию на средствах хранения информации или



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

в оборудовании компьютера (сети). Тем самым могут быть достигнуты две различные цели: во-первых, парализована нормальная работа компьютерной системы и, во-вторых, злоумышленник (например, под видом обслуживания или ремонта) может ознакомиться с имеющейся в системе или накопленной посредством использования модели «перехват» информацией. Событием, активизирующим закладку, может быть некоторый момент времени, либо сигнал из канала модемной связи (явный или замаскированный), либо состояние некоторых счетчиков (например, число запусков программ).

3. *Модель «наблюдатель».* Закладка встраивается в сетевое или телекоммуникационное программное обеспечение. Пользуясь тем, что данное ПО, как правило, всегда активно, программная закладка осуществляет контроль за процессами обработки информации на данном компьютере, установку и удаление закладок, а также съем накопленной информации. Закладка может инициировать события для ранее внедренных закладок, действующих по модели «тройанский конь».

4. *Модель «компрометация».* Закладка либо передает заданную злоумышленником информацию (например, клавиатурный ввод) в канал связи, либо сохраняет ее, не полагаясь на гарантированную возможность последующего приема или снятия. Более экзотический случай – закладка инициирует постоянное обращение к информации, приводящее к росту отношения сигнал/шум при перехвате побочных излучений.

5. *Модель «искажение или инициатор ошибок».* Программная закладка искажает потоки данных, возникающие при работе прикладных программ (выходные потоки), либо искажает входные потоки информации, либо инициирует (или подавляет) возникающие при работе прикладных программ ошибки.

6. *Модель «сборка мусора».* В данном случае прямого воздействия РПС может и не быть; изучаются «остатки» информации. В случае применения программной закладки навязывается такой порядок работы, чтобы максимизировать количество остающихся фрагментов ценной информации.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Злоумышленник получает либо данные фрагменты, используя закладки моделей 2 и 3, либо непосредственный доступ к компьютеру под видом ремонта или профилактики.

У рассмотренных различных по целям воздействия моделей закладок имеется важная общая черта - наличие операции записи, производимой закладкой (в оперативную или внешнюю память). При отсутствии данной операции никакое негативное влияние невозможно. Вполне понятно, что для направленного воздействия закладка должна также выполнять и операции чтения. Так, например, можно реализовать функцию целенаправленной модификации данных в каком-либо секторе жесткого диска, которая возможна только после их прочтения.

Таким образом, исполнение кода закладки может быть сопровождено операциями несанкционированной записи (НСЗ), например, для сохранения некоторых фрагментов информации, и несанкционированного считывания (НСЧ), которое может происходить отдельно от операций чтения прикладной программы или совместно с ними. При этом операции считывания и записи могут быть не связаны с получением информации, например считывание параметров устройства или его инициализация - закладка может использовать для своей работы и такие операции, в частности, для инициирования сбойных ситуаций или переназначения ввода вывода.

Возможные комбинации несанкционированных действий (НСЧ и НСЗ), а также санкционированных действий прикладной программы или операционной среды по записи (СЗ) или считыванию (СЧ) приведены в табл. 1.2 [ПАС].

Ситуации 1 - 4 соответствуют нормальной работе прикладной программы, когда закладка не оказывает на нее никакого воздействия.

Ситуация 5 может быть связана либо с разрушением кода прикладной программы в оперативной памяти ЭВМ, поскольку прикладная программа не выполняет санкционированные действия по записи и считыванию, либо с сохранением уже накопленной в ОП информации.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Ситуация 6 связана с разрушением или с сохранением информации (искажение или сохранение выходного потока), записываемой прикладной программой.

Ситуация 7 связана с сохранением информации (сохранение входного потока) считываемой прикладной программой.

Ситуация 8 связана с сохранением информации закладкой при считывании или записи информации прикладной программой.

Ситуация 9 не связана с прямым негативным воздействием, поскольку прикладная программа не активна, а закладка производит только НСЧ (процесс «настройки»).

Ситуация 10 может быть связана с сохранением выводимой информации в оперативную память.

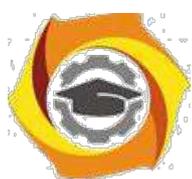
Ситуация 11 может быть связана с сохранением вводимой информации в оперативную память либо с изменением параметров процесса санкционированного чтения закладкой.

Ситуация 12 может быть связана с сохранением как вводимой, так и выводимой прикладной программой информации в оперативную память.

Ситуация 13 может быть связана с размножением закладки, сохранением накопленной в буферах ОП информации или с разрушением кода и данных в файлах, поскольку прикладная программа не активна.

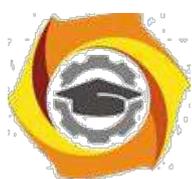
Таблица 1.2

<i>Номер ситуации</i>	<i>НСЧ</i>	<i>НСЗ</i>	<i>Действия РПС</i>	<i>СЧ</i>	<i>СЗ</i>
1	0	0	Нет	0	0
2	0	0	Нет	0	1
3	0	0	Нет	1	0
4	0	0	Нет	1	1
5	0	1	Изменение (разрушение) кода прикладной программы в ОП	0	0
6	0	1	Разрушение или сохранение выводимых	0	1



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

			прикладной программой данных		
7	0	1	Разрушение или сохранение вводимых прикладной программой данных	1	0



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

7

8	0	1	Разрушение или сохранение вводимых и выводимых данных	1	1
9	1	0	Нет	0	0
10	1	0	Перенос выводимых прикладной программой данных в ОП	0	1
11	1	0	Перенос вводимых прикладной программой данных в ОП	1	0
12	1	0	Перенос вводимых и выводимых данных в ОП	1	1
13	1	1	Процедуры типа «размножения вируса» (действия закладки независимо от операций прикладной программы)	0	0
14	1	1	Те же действия, что и в процедурах 6 – 8	0	1
15	1	1		1	0
16	1	1		1	1

Ситуации 14 - 16 могут быть связаны как с сохранением, так и с разрушением данных или кода и аналогичны ситуациям 6 - 8.

Несанкционированная запись закладкой может происходить:

в массив данных, не совпадающий с пользовательской информацией, - сохранение информации;

в массив данных, совпадающий с пользовательской информацией или ее подмножеством, - искажение, уничтожение или навязывание информации закладкой.

Следовательно, можно рассматривать три основные группы деструктивных функций, которые могут выполняться РПС [ПАС]:

сохранение фрагментов информации, возникающей при работе пользователя, прикладных программ, вводе/выводе данных, во внешней памяти



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

(локальной или удаленной) в сети или выделенном компьютере, в том числе сохранение различных паролей, ключей и кодов доступа, собственно конфиденциальных документов в электронном виде, либо безадресная



компрометация фрагментов ценной информации (модели «перехват», «компрометация»);

изменение алгоритмов функционирования прикладных программ (т.е. целенаправленное воздействие во внешней или оперативной памяти) - происходит изменение собственно исходных алгоритмов работы программ, например, программа разграничения доступа станет пропускать пользователей по любому паролю (модели «искажение», «тройанский конь»);

навязывание некоторого режима работы (например, при уничтожении информации - блокирование записи на диск, при этом информация, естественно, не уничтожается) либо замена записываемой информации данными, навязанными закладкой.

Методы внедрения рпс

Можно выделить следующие основные методы внедрения РПС [ПАС].

Маскировка закладки под «безобидное» программное обеспечение. Данный метод заключается в том, что программная закладка внедряется в систему под видом новой программы, на первый взгляд абсолютно безобидной. Программная закладка может быть внедрена в текстовый или графический редакторы, системную утилиту, компьютерную игру, хранитель экрана и т.д. После внедрения закладки ее присутствие в системе не нужно маскировать - даже если администратор заметит факт появления в системе новой программы, он не придаст этому значения, поскольку эта программа внешне совершенно безобидна.

Маскировка закладки под «безобидный» модуль расширения программной среды. Многие программные среды допускают свое расширение дополнительными программными модулями. Например, для операционных систем семейства Microsoft Windows модулями расширения могут выступать динамически подгружаемые библиотеки (DLL) и драйверы устройств. В таких модулях расширения может содержаться РПС, которое может потенциально внедрено в систему. Данный метод фактически является частным случаем предыдущего метода и отличается от него только тем, что закладка



представляет собой не прикладную программу, а модуль расширения программной среды.

Подмена закладкой одного или нескольких программных модулей атакуемой среды. Данный метод внедрения в систему программной закладки заключается в том, что в атакуемой программной среде выбирается один или несколько программных модулей, подмена которых фрагментами программной закладки позволяет оказывать на среду требуемые негативные воздействия. Программная закладка должна полностью реализовывать все функции подменяемых программных модулей.

Основная проблема, возникающая при практической реализации данного метода, заключается в том, что программист, разрабатывающий программную закладку, никогда не может быть уверен, что созданная им закладка точно реализует все функции подменяемого программного модуля. Если подменяемый модуль достаточно велик по объему или недостаточно подробно документирован, точно запрограммировать все его функции практически невозможно. Поэтому описываемый метод целесообразно применять только для тех программных модулей атакуемой среды, для которых доступна полная или почти полная документация. Оптимальной является ситуация, когда доступен исходный текст подменяемого модуля.

Прямое ассоциирование. Данный метод внедрения в систему программной закладки заключается в ассоциировании закладки с исполняемыми файлами одной или нескольких легальных программ системы. Сложность задачи прямого ассоциирования программной закладки с программой атакуемой среды существенно зависит от того, является атакуемая среда однозадачной или многозадачной, однопользовательской или многопользовательской. Для однозадачных однопользовательских систем эта задача решается достаточно просто. В то же время при внедрении закладки в многозадачную или многопользовательскую программную среду прямое ассоциирование закладки с легальным программным обеспечением является весьма нетривиальной задачей.



Косвенное ассоциирование. Косвенное ассоциирование закладки с программным модулем атакуемой среды заключается в ассоциировании закладки с кодом программного модуля, загруженным в оперативную память. При косвенном ассоциировании исполняемый файл программного модуля остается неизменным, что затрудняет выявление программной закладки.

Для того чтобы косвенное ассоциирование стало возможным, необходимо, чтобы устанавливающая часть закладки уже присутствовала в системе. Другими словами, программная закладка, внедряемая в систему с помощью косвенного ассоциирования, должна быть составной.



**Лабораторная работа № 2. Защита от закладок и дизассемблирования.
Способы встраивания защитных механизмов в программное обеспечение.
Понятие разрушающего программного воздействия.**

Цель работы: освоить технологию работы с дизассемблером и декомпилятором.

Основные принципы обеспечения безопасности ПО.

В качестве объекта обеспечения технологической и эксплуатационной безопасности ПО рассматривается вся совокупность его компонентов в рамках конкретной КС. В качестве доминирующей должна использоваться стратегия сквозного тотального контроля технологического и эксплуатационного этапов жизненного цикла компонентов ПО. Совокупность мероприятий по обеспечению технологической и эксплуатационной безопасности компонентов ПО должна носить, по возможности, конфиденциальный характер. Необходимо обеспечить постоянный, комплексный и действенный контроль за деятельностью разработчиков и пользователей компонентов ПО. Кроме общих принципов, обычно необходимо конкретизировать принципы обеспечения безопасности ПО на каждом этапе его жизненного цикла. Далее приводятся один из вариантов разработки таких принципов.

Принципы обеспечения технологической безопасности при обосновании, планировании работ и проектном анализе ПО.

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

Комплексности обеспечения безопасности ПО, предполагающей рассмотрение проблемы безопасности информационно - вычислительных процессов с учетом всех структур КС, возможных каналов утечки информации и несанкционированного доступа к ней, времени и условий их возникновения, комплексного применения организационных и технических мероприятий.



Планируемости применения средств безопасности программ, предполагающей перенос акцента на совместное системное проектирование ПО и средств его безопасности, планирование их использования в предполагаемых условиях эксплуатации.

Обоснованности средств обеспечения безопасности ПО, заключающейся в глубоком научно-обоснованном подходе к принятию проектных решений по оценке степени безопасности, прогнозированию угроз безопасности, всесторонней априорной оценке показателей средств защиты.

Достаточности защищенности программ, отражающей необходимость поиска наиболее эффективных и надежных мер безопасности при одновременной минимизации их стоимости.

Гибкости управления защитой программ, требующей от системы контроля и управления обеспечением безопасности ПО способности к диагностированию, опережающей нейтрализации, оперативному и эффективному устранению возникающих угроз,

Заблаговременности разработки средств обеспечения безопасности и контроля производства ПО, заключающейся в предупредительном характере мер обеспечения технологической безопасности работ в интересах недопущения снижения эффективности системы безопасности процесса создания ПО.

Документируемости технологии создания программ, подразумевающей разработку пакета нормативно-технических документов по контролю программных средств на наличие преднамеренных дефектов.

Принципы достижения технологической безопасности по в процессе его разработки.

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

Регламентации технологических этапов разработки ПО, включающей упорядоченные фазы промежуточного контроля, спецификацию программных модулей и стандартизацию функций и формата представления данных.



Автоматизации средств контроля управляющих и вычислительных программна наличие преднамеренных дефектов.

Создания типовой общей информационной базы алгоритмов, исходных текстов и программных средств, позволяющих выявлять преднамеренные программные дефекты.

Последовательной многоуровневой фильтрации программных модулей процессе их создания с применением функционального дублирования разработок и поэтапного контроля.

Типизации алгоритмов, программ и средств информационной безопасности, обеспечивающей информационную, технологическую и программную совместимость, на основе максимальной их унификации по всем компонентам и интерфейсам.

Централизованного управления базами данных проектов ПО и администрирование технологии их разработкис жестким разграничением функций, ограничением доступа в соответствии со средствами диагностики, контроля и защиты.

Блокирования несанкционированного доступа исполнителей и абонентов государственных и негосударственных сетей связи, подключенных к стендам для разработки программ.

Статистического учета и ведения системных журналов всех процессах разработки ПО с целью контроля технологической безопасности.

Использования только сертифицированных и выбранных в качестве единых инструментальных средств разработки программ для новых технологий обработки информации и перспективных архитектур вычислительных систем.

Дизассемблер — транслятор, преобразующий машинный код, объектный файл или библиотечные модули в текст программы на языке ассемблера.

По режиму работы с пользователем делятся на
Автоматические Интерактивные



Примером автоматических дизассемблеров может служить Sourcer. Такие дизассемблеры генерируют готовый листинг, который можно затем править в текстовом редакторе. Пример интерактивного — IDA. Он позволяет изменять правила дизассемблирования и является весьма удобным инструментом для исследования программ.

Дизассемблеры бывают **однопроходные и многопроходные**. Основная трудность при работе дизассемблера — отличить данные от машинного кода, поэтому на первых проходах автоматически или интерактивно собирается информация о границах процедур и функций, а на последнем проходе формируется итоговый листинг. Интерактивность позволяет улучшить этот процесс, так как просматривая дампы дизассемблируемой области памяти, программист может сразу выделить строковые константы, дать содержательные имена известным точкам входа, прокомментировать разобранные им фрагменты программы.

Чаще всего дизассемблер используют для анализа программы (или ее части), исходный текст которой неизвестен — с целью модификации, копирования или взлома. Реже — для поиска ошибок (багов) в программах и компиляторах, а также для анализа оптимизации создаваемого компилятором машинного кода. Обычно однопроходный дизассемблер (как и построчный ассемблер) является составной частью отладчика.

Защита от дизассемблирования

Первое направление защиты, как правило, реализуется значительно легче, чем второе, поэтому будет приведен лишь краткий обзор данного направления. При реализации защиты программ от дизассемблирования можно применять различные приемы.

Среди них наиболее часто используемым и эффективным приемом является зашифровка и \ или упаковка отдельных участков исходного кода или всего кода целиком, при этом необходимо позаботиться о распаковке \ расшифровке программы на точке входа. Таким образом, при просмотре



исполняемого машинного кода исполняемого файла вместо рабочего кода программы будет отображен лишь бессмысленный набор операций. При реализации защиты от дизассемблирования используется также множество приемов, которые реализуются с целью запутать потенциального взломщика. Можно привести несколько примеров такого вида приемов:

- увеличение исходного кода программы добавлением множества «бессмысленных» операций, а рабочий участок программы записать в определенное место этого множества;
- замена местами адресов обработчиков (векторов) прерываний, например, поменять местами вектор прерывания видео сервиса (INT 10h) с вектором прерывания сервиса DOS (INT 21h), после такой замены для вызова из программы какой-либо функции прерывания INT 21h необходимо пользоваться вызовом прерывания INT 10h.

Для достижения наиболее надежной и эффективной защиты используется комбинация нескольких приемов.

Защита от отладки

Для защиты программы от трассировки отладчиком также существует несколько способов. Наиболее распространенными являются два из них.

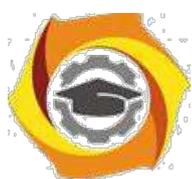
Первый способ.

Идея:

При трассировке программы команды выполняются по команде человека, поэтому длительность выполнения операций (время от начала одной операции до начала следующей) изменяется. Поэтому в программу можно включать точки для проверки времени выполнения одинаковых участков кода программы. Если время выполнения выполнения одинаковых участков различна, то это означает, что программа трассируется в данный момент, необходимо выйти из программы, иначе - продолжить выполнение.

Алгоритм реализации:

1. Запомнить текущее время;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

2. Выполнить контрольный участок кода;



3. Запомнить текущее время и разность текущего и предыдущего запомненного времени;
4. Выполнить контрольный участок кода повторно;
5. Сравнить разность текущего времени и предыдущего запомненного текущего времени с предыдущей запомненной разностью;
6. Если разности совпадают, продолжить выполнение, иначе – выйти из программы.
 - метаморфическое преобразование кода программы, позволяющее защитить программу от дизассемблирования и модификации;
 - защита ключом отдельных участков кода программы (поддерживается только в зарегистрированной версии);
 - полное разрушение логики защищенных фрагментов кода, не позволяющее анализировать программу с помощью дизассемблера или отладчика;
 - обнаружение и противодействие отладчикам SoftIce, NtIce, TD и др.;
 - защита точки входа;
 - защита от модификации кода;
 - защищенная работа с реестром, не позволяющая программам вроде RegMon определить, к какому ключу реестра обращается твоя программа;
 - технология "динамического импорта", которая разрушает имена всех импортируемых функций, а также не использует функцию GetProcAddress;
 - сжатие ресурсов и исполнимого кода приложения;
 - поддержка коротких серийных номеров (12 символов);
 - поддержка внешнего генератора серийных номеров с OLE/DLL-интерфейсом;
 - технология OneTouch Trial (о ней читай ниже).

Самое главное, что нас интересует – это метаморфическое преобразование кода программы и поддержка серийных номеров. Метаморфическое кодирование позволяет изменить код программы до



неузнаваемости и запутать отладчик и человека, который запустил этот отладчик.

Декомпилятор.

Он переводит двоичный код в символьный на языке команд какого-нибудь языка. Например, диасемблеры, деклиппер и многие другие. Эти средства появились раньше отладчиков, т.к. вначале не было архитектуры со встроенными средствами отлаживания программ. С помощью декомпиляторов можно изменять исходный код программы. Допустим необходимо внести крупные изменения в код программы. Прямая вставка двоичных кодов не помогает, т.к. нарушается расположение меток перехода и процедур. Программа – это линейка кода, по которой нужно перемещаться нелинейно, переходить с определенным смещением. Если линейка удлиняется из-за добавления чего-то в середине, все смещения будут показывать не туда куда нужно. Повторная перекомпиляция вписывает новые смещения.

Среди декомпиляторов можно выделить: Hacker-VIEW (HVIEW), IDA (интерактивный дизассемблер).

С помощью Hacker-VIEW можно посмотреть любой исполняемый файл по любому смещению. Можно выполнить какую-то часть программы. Это позволяет расшифровывать программы и обходить защиту от дизассемблирования. Этот декомпилятор «понимает» как старые форматы исполняемых файлов DOS-COM и DOS-EXE, так и форматы исполняемых файлов Windows.

IDA очень мощное средство работы с ассемблерными текстами программ. Обладает широким спектром возможностей, имеет более удобный интерфейс, чем Hacker-VIEW. Очень хорошо предусмотрена архитектура работы программ в Windows (такие вещи, как DLL, расширенный режим работы с памятью и т.д.).

Декомпиляторы программ занимают свое место в инструментарии взломщика. В основном это совместное использование с отладчиками.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Второе средство – отладчики.



Отладчики позволяют запускать отдельные части программы и следить за изменениями, которые она производит, за результатами ее работы.

Защите от отладки не стоит уделять много времени, т.к. все возможные хитрости и приемы уже известны и взломщикам и программистам. Так же и шифрование. Любой хакер, если получает заказ на взлом, имеет доступ к нормальной копии программы. То есть он ее либо может купить, либо попользоваться ею на компьютере покупателя.

Среди отладчиков выделим: SOFTICE и WINICE.

С появлением Windows отладка программ стала на порядок проще и намного удобнее дизассемблирования. Принципиально изменился стиль некоторых атак на защиту программ. Теперь не надо шаг за шагом смотреть на ассемблерный код, «продираться» сквозь дебри незначущих кодов и защит.

Теперь надо отловить нужное событие и понять как на него реагирует программа. Это, конечно, не всегда бывает так просто, как выглядит на словах. Как и ранее, отладка требует знание архитектуры операционной системы.

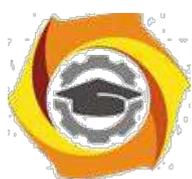
Неважно насколько сложным был бы механизм защиты, все сводится к простейшей проверке или дешифровке. И взлом, в случае с проверкой, можно разбить на два этапа: установка «брейков» на «подозрительные» флаги, обнаруженные в процедуре защиты; анализ обращений к флагам. По реакции программы можно судить флаг это или просто переменная.

Практическая часть.

1. Изучить теоретическую часть. Сделать записи в тетради.
2. Провести сравнение декомпилятора и отладчика. По данным составить таблицу сравнений.
3. Ответить на контрольные вопросы

Способы встраивания защитных механизмов в программное обеспечение

Встраивание защитных механизмов можно выполнить следующими



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ОСНОВНЫМИ СПОСОБАМИ:



вставкой фрагмента проверочного кода в исполняемый файл;
преобразованием исполняемого файла к неисполняемому виду (шифрование, архивация с неизвестным параметром и т.д.) и применением для загрузки не средств операционной среды, а некоторой программы, в теле которой и осуществляются необходимые проверки;

вставкой проверочного механизма в исходный код на этапе разработки и отладки программного продукта;
комбинированием указанных методов.

Применительно к конкретной реализации защитных механизмов для конкретной вычислительной архитектуры можно говорить о защитном фрагменте в исполняемом или исходном коде. К процессу и результату встраивания защитных механизмов можно предъявить следующие требования:

высокая трудоемкость обнаружения защитного фрагмента при статическом исследовании (особенно актуальна при встраивании в исходный код программного продукта);

высокая трудоемкость обнаружения защитного фрагмента при динамическом исследовании (при отладке и трассировке по внешним событиям);

высокая трудоемкость обхода или редуцирования защитного файла.

Возможность встраивания защитных фрагментов в исполняемый код обусловлена типовой архитектурой исполняемых модулей различных операционных сред, содержащих, как правило, адрес точки входа в исполняемый модуль. В этом случае добавление защитного фрагмента происходит следующим образом. Защитный фрагмент добавляется к началу или концу исполняемого файла, точка входа корректируется таким образом, чтобы при загрузке управление передалось дополнительному защитному фрагменту, а в составе защитного фрагмента предусматривается процедура возврата к оригинальной точке входа. Достаточно часто оригинальный исполняемый файл подвергается преобразованию. В этом случае перед возвратом управления оригинальной точке входа производится преобразование



образа оперативной памяти загруженного исполняемого файла к исходному виду.

В случае дополнения динамических библиотек возможна коррекция указанным образом отдельных функций.

Существенным недостатком рассмотренного метода является его легкая обнаруживаемость и в случае отсутствия преобразования оригинального кода исполняемого файла – легкая возможность обхода защитного фрагмента путем восстановления оригинальной точки входа.

Обфускация программ

В данном подразделе кратко затрагиваются вопросы, связанные с активно развивающимися теорией и практикой обфускации программ. Неформально говоря, под *обфускацией программ* здесь понимается преобразование программ с целью максимального затруднения их анализа и модификации, при сохранении, в то же время, их функциональных возможностей. Известные на сегодня методы обфускации, как правило, носят эмпирический характер и слабо обоснованы теоретически. В работе [CTL] приведена классификация методов обфускации, а в работе [BGI] предпринята, скорее всего, первая попытка формализации и теоретического обоснования задачи обфускации программ.

Неформально, *обфускатор* – это (эффективный, вероятностный) «компилятор», который в качестве входа имеет программу P и производит новую программу $O(P)$, которая имеет те же самые функциональные возможности, как и P , но, в то же время, программа $O(P)$ является «*неясной*», («*непонятной*») для противника (наблюдателя, постороннего лица) в некотором заранее определенном смысле. Обфускаторы, если будет доказано, что они существуют, могут применяться для защиты программного обеспечения и, кроме того, могут иметь широкую область криптографических и теоретико-сложностных приложений. Существование обфускаторов для этих приложений

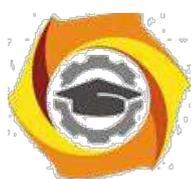


должно быть основано на формализации определения понятия «неясности» [BGI].

Методы компенсации угроз информационной безопасности

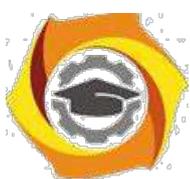
Таблица 15.2. Методы компенсации угроз информационной безопасности

Атаки, осуществляемые атакующей стороной класса ВН (внешний наблюдатель)		
A1	Перехват, накопление, анализ информации	защита трафика
A2	Разведка топологии сети	туннелирование трафика
A3	Атаки на систему управления сетью	аутентификация трафика, защита трафика, авторизованный доступ
A4	Проникновение в сеть с целью компрометации информации на серверах сети	аутентификация трафика, защита трафика, авторизованный доступ
A5	Атаки путем фальсификации или повторной передачи	аутентификация трафика, защита трафика и проверка целостности
Атаки, осуществляемые атакующей стороной класса ВА (внешний абонент)		
B1	Компрометация ресурсов сети для повышения эффективности атак нижележащих классов	Аутентификация трафика, авторизованный доступ
B2	Маскировка под пользователей своей сети	Аутентификация трафика, авторизованный доступ
Атаки, осуществляемые атакующей стороной класса ЗП (зарегистрированный пользователь) и ЗПП (зарегистрированный привилегированный пользователь)		
C1	Компрометация ресурсов сети для повышения эффективности атак нижележащих классов	Аутентификация трафика, авторизованный доступ
Атаки, осуществляемые атакующей стороной класса СА (системный администратор)		
D1	Атаки на систему защиты сети	



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Аут
ент
ифи
кац
ия
тра
фик
а,
соб
ыти
йно
е
про
ток
оли
ров
ани
е



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

22

D2	Усиление атак нижележащих классов путем эффективной маскировки под другого участника, кооперация с другими атакующими сторонами	Протоколирование действий оператора, сигнализация	действий оперативная
D3	Компрометация системы управления	Протоколирование действий оператора, ответственности, контроль	действий разделение взаимный
Атаки, осуществляемые атакующей стороной класса АБ (администратор безопасности)			
E1	Атаки на систему протоколирования сети	Протоколирование действий оператора, ответственности, контроль	действий разделение взаимный
E2	Атаки на систему защиты трафика и аутентификации	Протоколирование действий оператора, ответственности, контроль	действий разделение взаимный
E3	Компрометация сертификатов	Организационно-штатные мероприятия, организация учета и контроля	
E4	Усиление атак нижележащих классов путем маскировки под другого участника, кооперация с другими атакующими сторонами	Протоколирование действий оператора, ответственности, контроль	действий разделение взаимный

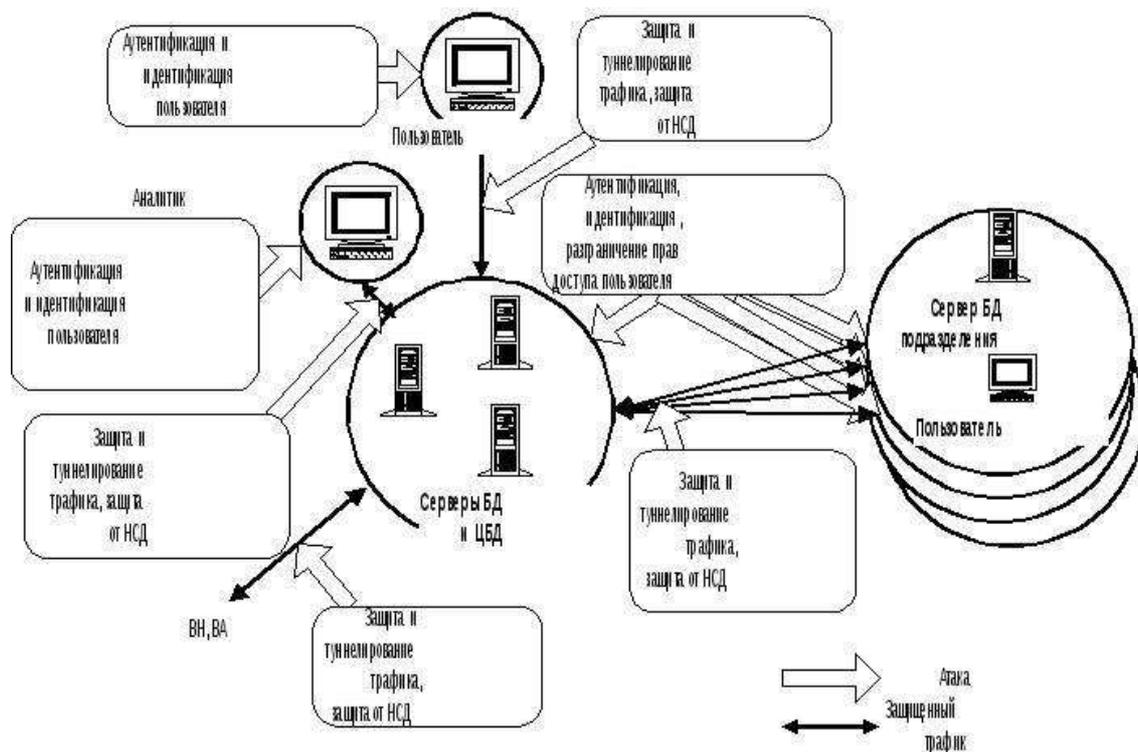
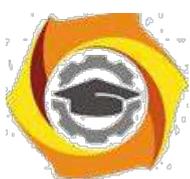
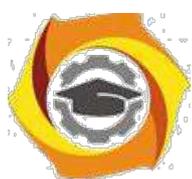


Рис. 1 - Методы компенсации угроз информационной безопасности

Понятие разрушающего программного воздействия

Существующая на сегодняшний день концепция построения защищенных компьютерных систем (КС) подразумевает использование в едином комплексе программных средств различного назначения. Так, система автоматизированного документооборота банка может использовать на одних и тех же аппаратных средствах (например, компьютере, аппаратуре передачи данных (модемах) и т.д.) взаимосвязанный комплекс программных средств: операционную среду, программные средства управления базой данных (СУБД), телекоммуникационные средства, средства обработки текстов (редакторы, текстовые процессоры), возможно, также средства антивирусного контроля, разграничения доступа к программам и данным, средства криптографической защиты передаваемой и хранимой информации, средства криптографической



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

идентификации и аутентификации (электронная цифровая подпись) и многое другое.



Важным моментом при работе прикладных программ, в особенности средств защиты информации, является необходимость потенциального невмешательства иных присутствующих в КС прикладных или системных программ в процесс обработки информации.

Напомним, что под несанкционированным доступом (НСД) к ресурсам защищенной КС понимаются действия по использованию, изменению и уничтожению исполняемых модулей и массивов данных, принадлежащих указанной системе, производимые субъектом, не имеющим права на такие действия. Данного субъекта будем называть злоумышленником (нарушителем). Остальные субъекты именуются легальными пользователями. Данное априорно деление предполагает несколько существенно важных моментов:

система имеет механизм различения злоумышленников и легальных пользователей;

в системе имеются пассивная и активная компоненты (исполняемые модули и данные), пользование которыми злоумышленником нежелательно;

в системе имеется механизм установления соответствия субъекта и информации, к которой он имеет доступ.

Как уже отмечалось, в настоящее время для интегрального обозначения процедур обеспечения безопасности информации употребляют термин "политика безопасности", а всевозможные ситуации нарушения априорно предписанных правил называют нарушениями безопасности.

Считая, что злоумышленник в совершенстве владеет всем программным и аппаратным обеспечением системы, можно предполагать, что несанкционированный доступ может быть вызван следующими причинами:

отключением или видоизменением защитных механизмов злоумышленником (сюда же можно отнести процедуры доступа "мимо" средств контроля, например, при нарушении уровня иерархии информационных объектов – доступ к объектам типа "файл" как к последовательности секторов и др.);



входом злоумышленника в систему под именем и с полномочиями легального пользователя.

В первом случае злоумышленник должен видоизменить защитные механизмы в системе (например, отключить программу запросов паролей пользователей). Во втором — каким-либо образом выяснить или подделать идентификатор реального пользователя (например, "подсмотреть" пароль, вводимый с клавиатуры). В обоих случаях НСД можно представить моделью опосредованного несанкционированного доступа – когда проникновение в систему осуществляется на основе некоторого воздействия, произведенного предварительно внедренной в систему программой (программами).

Например, злоумышленник пользуется информацией, которая извлечена из некоторого массива данных, созданного при совместной работе программного средства злоумышленника и системы проверки прав доступа (т.е. предварительно внедренная в систему программа при доступе легального пользователя перехватит его пароль и сохранит в заранее известном и доступном злоумышленнику месте, а затем злоумышленник воспользуется данным паролем для входа в систему). Либо злоумышленник изменит часть системы защиты так, чтобы она перестала выполнять свои функции. Например, модифицирует систему проверки прав доступа

так, чтобы она пропускала любого пользователя, или изменит программу шифрования вручную (или при помощи некоторой другой программы) таким образом, чтобы она перестала шифровать или изменила алгоритм шифрования на более простой.

Программой с потенциально опасными последствиями (badware – "вредные программы") назовем некоторую программу (осмысленный набор инструкций для какого-либо процессора), которая способна выполнить любое непустое подмножество перечисленных функций:

- 1) скрыть признаки своего присутствия в программной среде КС;



- 2) реализовать самодублирование, ассоциирование себя с другими программами и/или перенос своих фрагментов в иные (не занимаемые изначально указанной программой) области оперативной или внешней памяти;
- 3) разрушить (исказить произвольным образом) код программ (отличных от нее) в оперативной памяти КС;
- 4) перенести (сохранить) фрагменты информации из оперативной памяти в некоторые области оперативной или внешней памяти прямого доступа (локальных или удаленных);
- 5) имеет потенциальную возможность исказить произвольным образом, заблокировать и/или подменить выводимый во внешнюю память или в канал связи массив информации, образовавшийся в результате работы прикладных программ или уже находящийся во внешней памяти, либо изменить его параметры.

Программы с потенциально опасными последствиями можно (весьма условно) разделить на три класса.

1. Классические программы — "вирусы" (термин применен в 1984 г. Ф. Коэном). Особенность данного класса вредных программ заключается в ненаправленности их воздействия на конкретные типы прикладных программы, а также в том, что во главу угла ставится самодублирование вируса. Разрушение информации вирусом не направлено на конкретные программы и встречается у 10...20% вирусов.

2. Программы типа "программный червь" или "троянский конь" и фрагменты программ типа "логический люк". В данном случае имеет место обратная ситуация — самодублирование не всегда присуще такого рода программам или фрагментам программ, но они обладают возможностью перехвата конфиденциальной информации, или извлечения информации из сегментов систем безопасности, или разграничения доступа.

3. Программные закладки или разрушающие программные воздействия (РПВ) — обобщенный класс программ (в смысле отсутствия конкретных признаков) с потенциально опасными последствиями, обязательно



реализующие хотя бы один из п.п. 3 - 5 определения программы с потенциально опасными последствиями.

Далее наряду с термином "программа с потенциально опасными последствиями" будут использованы термины "закладка", "программная закладка" либо сокращение РПВ.

Кроме того, программные закладки можно классифицировать по методу и месту их внедрения и применения (т.е. по "способу доставки" в компьютерную систему):

закладки, ассоциированные с программно-аппаратной средой компьютерной системы (основная BIOS или расширенные BIOS);

закладки, ассоциированные с программами первичной загрузки (находящиеся в Master Boot Record или BOOT-секторах активных разделов) – загрузочные закладки;

закладки, ассоциированные с загрузкой драйверов DOS, драйверов внешних устройств других ОС, командного интерпретатора, сетевых драйверов, т.е. с загрузкой операционной среды;

закладки, ассоциированные с прикладным программным обеспечением общего назначения (встроенные в клавиатурные и экранные драйверы, программы тестирования компьютеров, утилиты и оболочки типа NORTON);

исполняемые модули, содержащие только код закладки (как правило, внедряемые в файлы пакетной обработки типа .BAT);

модули-имитаторы, совпадающие по внешнему виду с некоторыми программами, требующими ввода конфиденциальной информации — наиболее характерны для Unix-систем;

закладки, маскируемые под программные средства оптимизационного назначения (архиваторы, ускорители обмена с диском и т.д.);



закладки, маскируемые под программные средства игрового и развлекательного назначения (как правило, используются для первичного внедрения закладок).

Как видно, программные закладки имеют много общего с классическими вирусами, особенно в части ассоциирования себя с исполняемым кодом (загрузочные вирусы, вирусы-драйверы, файловые вирусы). Кроме того, программные закладки, как и многие известные вирусы классического типа, имеют развитые средства борьбы с отладчиками и дизассемблерами.

Для того чтобы закладка смогла выполнить какие-либо действия по отношению к прикладной программе или данным, она должна получить управление, т.е. процессор должен начать выполнять инструкции (команды), относящиеся к коду закладки. Это возможно только при одновременном выполнении двух условий:

1) закладка должна находиться в оперативной памяти до начала работы программы, которая является целью воздействия закладки, следовательно, она должна быть загружена раньше или одновременно с этой программой;

2) закладка должна активизироваться по некоторому общему как для закладки, так и для программы событию, т.е. при выполнении ряда условий в программно-аппаратной среде управление должно быть передано программе-закладке. Данное событие далее будем называть активизирующим.

Обычно выполнение указанных условий достигается путем анализа и обработки закладкой общих относительно закладки и прикладной программы воздействий (как правило, прерываний) либо событий (в зависимости от типа и архитектуры операционной среды). Причем прерывания должны сопровождать работу прикладной программы или работу всего

компьютера. Данные условия являются необходимыми (но недостаточными), т.е. если они не выполнены, то активизация кода закладки не произойдет и код не сможет оказать какого-либо воздействия на работу прикладной программы.



Кроме того, возможен случай, когда при запуске программы (активизирующим событием является запуск программы) закладка разрушает некоторую часть кода программы, уже загруженной в оперативную память (ОП), и, возможно, систему контроля целостности кода или контроля иных событий и на этом заканчивает свою работу. Данный случай не противоречит необходимым условиям.

С учетом замечания о том, что закладка должна быть загружена в ОП раньше, чем цель ее воздействий, можно выделить закладки двух типов.

1. Закладки резидентного типа — находятся в памяти постоянно с некоторого момента времени до окончания сеанса работы компьютера (выключения питания или перезагрузки). Закладка может быть загружена в память при начальной загрузке компьютера, загрузке операционной среды или запуске некоторой программы (которая по традиции называется вирусом-носителем или просто носителем), а также запущена отдельно.

2. Закладки нерезидентного типа — начинают работу, как и закладки резидентного типа, но заканчивают ее самостоятельно через некоторый промежуток времени или по некоторому событию, при этом выгружая себя из памяти целиком.

Контрольные вопросы:

1. Что такое дизассемблер?
2. Как происходит защита программ от дизассемблирования?
3. Как происходит защита программ от отладки?
4. Какие виды отладчиков вы знаете?
5. Что такое декомпилятор?
6. Какие он функции выполняет?
7. Что такое трассировка?
8. Какие виды дизассемблеров вам известны?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

9. Какие приемы дизассемблирования вам известны?



**Лабораторная работа № 3. Восстановление зараженных файлов.
Профилактика проникновения «троянских программ». Настройка
безопасности почтового клиента.**

Цель работы: ознакомиться с возможностями профилактики вирусов и восстановления заражённых файлов.

Краткие теоретические сведения.

Макровирусы заражают файлы – документы и электронные таблицы популярных офисных приложений [10].

Для анализа макровирусов необходимо получить текст их макросов. Для нешифрованных («не - стелс») вирусов это достигается при помощи меню Сервис/Макрос. Если же вирус шифрует свои макросы или использует «стелс»-приемы, то необходимо воспользоваться специальными утилитами просмотра макросов.

***Задание: восстановить файл, зараженный
макровирусом*** Алгоритм выполнения работы.

Для восстановления документов Word и Excel достаточно сохранить пораженные файлы в текстовый формат RTF, содержащий практически всю информацию из первоначальных документов и не содержащий макросы.

Для этого выполните следующие действия.

1. В программе **WinWord** выберите пункты меню «**Файл**» – «**Сохранить как**».
2. В открывшемся окне в поле «**Тип файла**» выберите «**Текст в формате RTF**» (рис. 7).

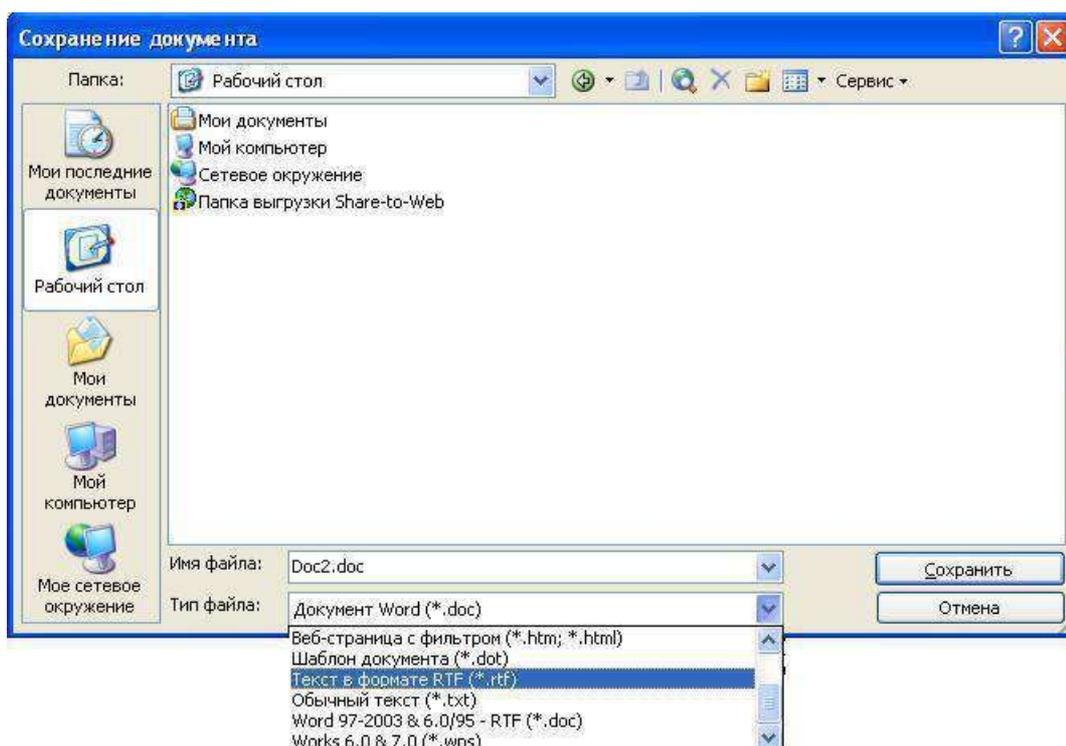
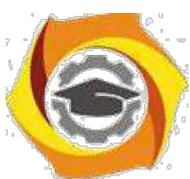


Рис. 7

3. Выберите команду **Сохранить**, при этом имя файла оставьте прежним.
4. В результате появится новый файл с именем существующего, но с другим расширением.
5. Далее закройте **WinWord** и удалите все зараженные Word-документы и файл-шаблон **NORMAL.DOT** в папке **WinWord**.
6. Запустите **WinWord** и восстановите документы из RTF-файлов в соответствующий формат файла (рис. 8) с расширением (.doc).
7. В результате этой процедуры вирус будет удален из системы, а практически вся информация останется без изменений.

Примечание:

- а) этот метод рекомендуется использовать, если нет соответствующих антивирусных программ;
- б) при конвертировании файлов происходит потеря невирусных макросов, используемых при работе. Поэтому перед запуском описанной процедуры следует сохранить их исходный текст, а после обезвреживания вируса – восстановить необходимые макросы в первоначальном виде.

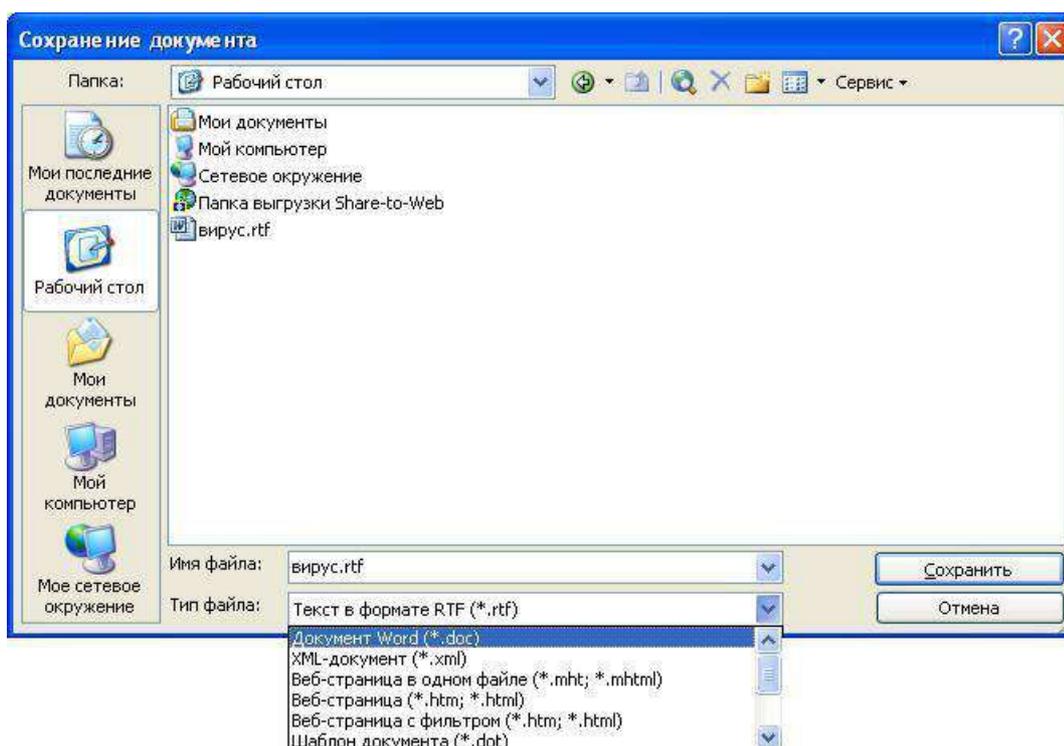
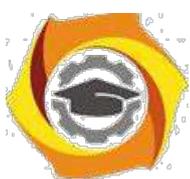
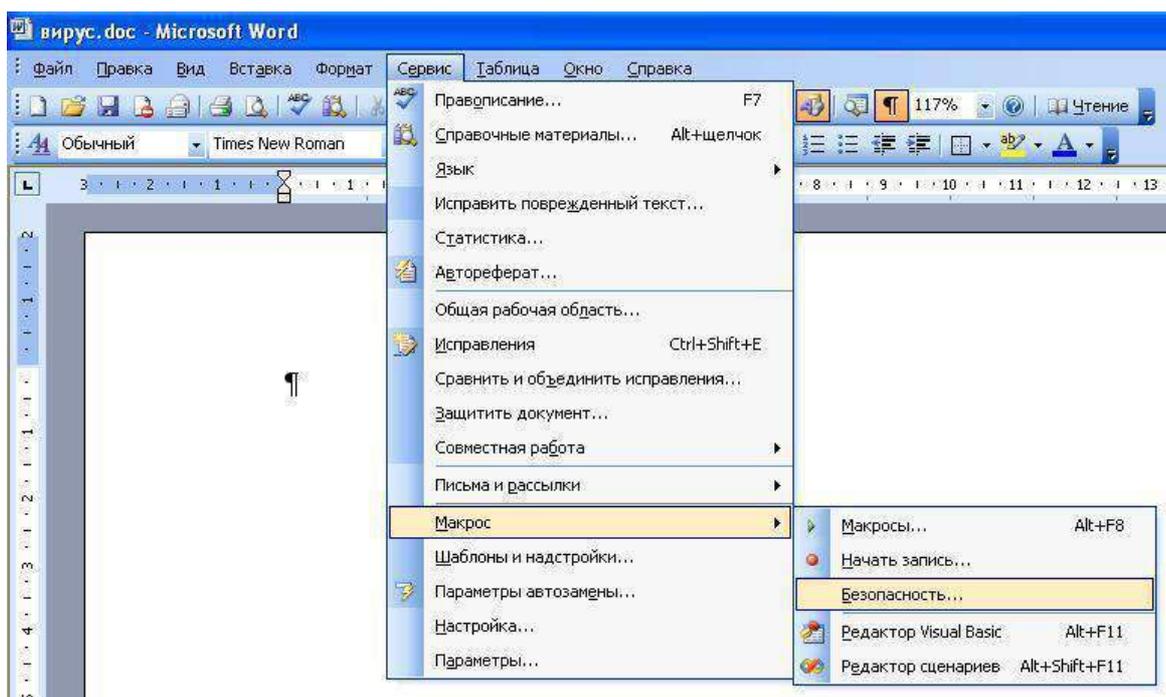


Рис. 8

8. Для последующей защиты файлов от макровирусов включите защиту от запуска макросов.

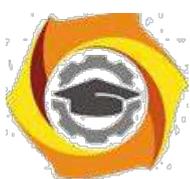
9. Для этого в **WinWord** выберите последовательно пункты меню: **Сервис - Макрос - Безопасность** (рис. 9).





**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Рис. 9



10. В открывшемся окне на закладке **Уровень безопасности** отметьте пункт **Высокая** (рис. 10).

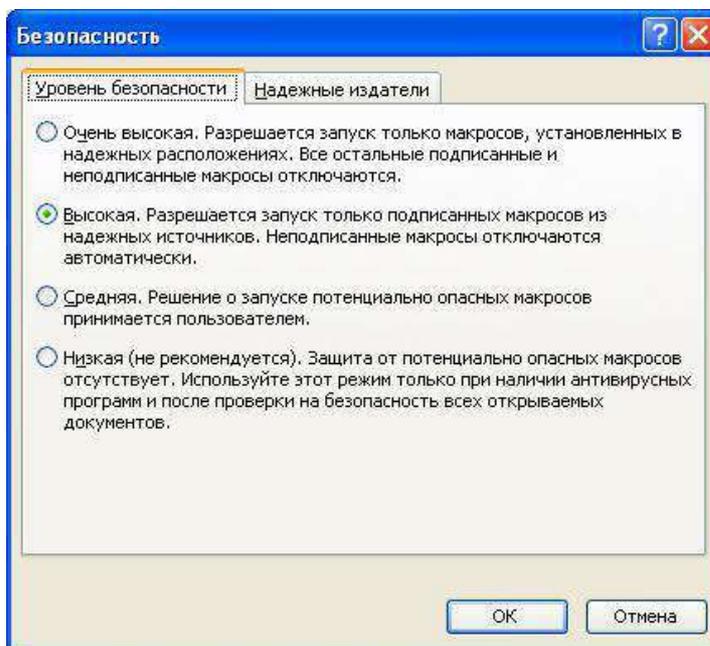


Рис. 10

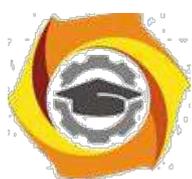
Задания для самостоятельной работы

1. Создайте файл **virus.doc** (содержание – чистый лист) и выполните алгоритм восстановления файла (в предположении его заражения макровирусом).
2. Зафиксируйте этапы работы, используя команду **PrintScreen** клавиатуры (скопированные таким образом файлы вставьте в новый Word-документ для отчета преподавателю).
3. Сравните размеры файлов **virus.doc** и **virus.rtf**, используя пункт контекстного меню **Свойства** (для этого выделите в **Проводнике** файл, нажмите правую кнопку мыши и выберите пункт **Свойства**).

Контрольные вопросы

1. Какие файлы заражают макровирусы?
2. Как просмотреть код макровируса?
3. Как восстановить файл, зараженный макровирусом?

Практическая работа 2 Профилактика проникновения «троянских программ»



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Краткие теоретические сведения



Главное отличие «троянских программ» от компьютерных вирусов состоит в том, что они не размножаются на зараженном компьютере и не имеют встроенных возможностей к самораспространению. «Троянские кони» засылаются пользователям (обычно через электронную почту) непосредственно их авторами под видом каких-нибудь полезных утилит. На самом деле они производят несанкционированное внедрение на компьютеры и в корпоративные сети различного рода нежелательных программ. Именно этой особенности «Троянские кони» обязаны своим названием.

Троянские программы различаются между собой по тем действиям, которые они производят на зараженном компьютере.

Backdoor – троянские утилиты удаленного администрирования

Троянские программы этого класса являются утилитами удаленного администрирования компьютеров в сети. По своей функциональности они во многом напоминают различные системы администрирования, разрабатываемые и распространяемые фирмами-производителями программных продуктов.

Единственная особенность этих программ заставляет классифицировать их как вредные троянские программы: отсутствие предупреждения об установке и запуске. При запуске «троянец» устанавливает себя в системе и затем следит за ней, при этом пользователю не выдается никаких сообщений о действиях троянца в системе. Более того, ссылка на «троянца» может отсутствовать в списке активных приложений. В результате «пользователь» этой троянской программы может и не знать о ее присутствии в системе, в то время как его компьютер открыт для удаленного управления.

Утилиты скрытого управления позволяют делать с компьютером все, что в них заложил автор: принимать или отсылать файлы, запускать и уничтожать их, выводить сообщения, стирать информацию, перезагружать компьютер и т. д. В результате эти троянцы могут быть использованы для обнаружения и передачи конфиденциальной информации, для запуска вирусов, уничтожения данных и т.п. – пораженные компьютеры оказываются открытыми для злоумышленных действий хакеров.



Таким образом, троянские программы данного типа являются одним из самых опасных видов вредоносного программного обеспечения, поскольку в них заложена возможность самых разнообразных злоумышленных действий, присущих другим видам троянских программ.

Отдельно следует отметить группу бэкдоров, способных распространяться по сети и внедряться в другие компьютеры, как это делают компьютерные черви. Отличает такие «троянцы» от червей тот факт, что они распространяются по сети не самопроизвольно (как черви), а только по специальной команде «хозяина», управляющего данной копией троянской программы.

Trojan-PSW – воровство паролей

Данное семейство объединяет троянские программы, «ворующие» различную информацию с зараженного компьютера, обычно – системные пароли (PSW – Password-Stealing-Ware). При запуске PSW-троянцы ищут системные файлы, хранящие различную конфиденциальную информацию (обычно номера телефонов и пароли доступа к интернету) и отсылают ее по указанному в коде «троянца» электронному адресу или адресам.

Существуют PSW-троянцы, которые сообщают и другую информацию о зараженном компьютере, например, информацию о системе (размер памяти и дискового пространства, версия операционной системы), тип используемого почтового клиента, IP-адрес и т.п. Некоторые троянцы данного типа «воруют» регистрационную информацию к различному программному обеспечению, коды доступа к сетевым играм и прочее.

Trojan-AOL – семейство троянских программ, «ворующих» коды доступа к сети AOL (America Online). Выделены в особую группу по причине своей многочисленности.

Trojan-Clicker – интернет-кликеры

Семейство троянских программ, основная функция которых – организация несанкционированных обращений к интернет-ресурсам (обычно к веб-страницам). Достигается это либо посылкой соответствующих команд



браузеру, либо заменой системных файлов, в которых указаны «стандартные» адреса интернет-ресурсов (например, файл hosts в MS Windows).

У злоумышленника могут быть следующие цели для подобных действий:

увеличение посещаемости каких-либо сайтов с целью увеличения показов рекламы;

организация DoS-атаки (Denial of Service) на какой-либо сервер;

привлечение потенциальных жертв для заражения вирусами или троянскими программами.

Trojan-Downloader – доставка прочих вредоносных программ

Троянские программы этого класса предназначены для загрузки и установки на компьютер-жертву новых версий вредоносных программ, установки «троянцев» или рекламных систем. Загруженные из интернета программы затем либо запускаются на выполнение, либо регистрируются «троянцем» на автозагрузку в соответствии с возможностями операционной системы. Данные действия при этом происходят без ведома пользователя.

Информация об именах и расположении загружаемых программ содержится в коде и данных троянца или скачивается троянцем с «управляющего» интернет-ресурса (обычно с веб-страницы).

Trojan-Dropper – инсталляторы прочих вредоносных программ

Троянские программы этого класса написаны в целях скрытной инсталляции других программ и практически всегда используются для «подсовывания» на компьютер-жертву вирусов или других троянских программ.

Данные троянцы обычно без каких-либо сообщений (либо с ложными сообщениями об ошибке в архиве или неверной версии операционной системы) сбрасывают на диск в какой-либо каталог (в корень диска C:, во временный каталог, в каталоги Windows) другие файлы и запускают их на выполнение.

Обычно структура таких программ следующая:



Основной код
Файл 1
Файл 2
...

«Основной код» выделяет из своего файла остальные компоненты (файл 1, файл 2, ...), записывает их на диск и открывает их (запускает на выполнение).

Обычно один (или более) компонент является троянской программой, и как минимум один компонент является «обманкой»: программой-шуткой, игрой, картинкой или чем-то подобным. «Обманка» должна отвлечь внимание пользователя и/или продемонстрировать то, что запускаемый файл действительно делает что-то «полезное», в то время как троянская компонента устанавливается в систему.

В результате использования программ данного класса хакеры достигают двух целей:

скрытная инсталляция троянских программ и/или вирусов;

защита от антивирусных программ, поскольку не все из них в состоянии проверить все компоненты внутри файлов этого типа.

Trojan-Proxy – троянские прокси-сервера

Семейство троянских программ, скрытно осуществляющих анонимный доступ к различным интернет-ресурсам. Обычно используются для рассылки спама.

Trojan-Spy – шпионские программы

Данные троянцы осуществляют электронный шпионаж за пользователем зараженного компьютера: вводимая с клавиатуры информация, снимки экрана, список активных приложений и действия пользователя с ними сохраняются в какой-либо файл на диске и периодически отправляются злоумышленнику.

Троянские программы этого типа часто используются для кражи информации пользователей различных систем онлайн-платежей и банковских систем.



Trojan – прочие троянские программы

К данным троянцам относятся те из них, которые осуществляют прочие действия, попадающие под определение троянских программ, т.е. разрушение или злонамеренная модификация данных, нарушение работоспособности компьютера и прочее.

В данной категории также присутствуют «многоцелевые» троянские программы, например, те из них, которые одновременно шпионят за пользователем и предоставляют проху-сервис удаленному злоумышленнику.

Trojan-Notifier – оповещение об успешной атаке

Троянцы данного типа предназначены для сообщения своему «хозяину» о зараженном компьютере. При этом на адрес «хозяина» отправляется информация о компьютере, например, IP-адрес компьютера, номер открытого порта, адрес электронной почты и т.п. Отсылка осуществляется различными способами: электронным письмом, специально оформленным обращением к веб-странице «хозяина», ICQ-сообщением.

Данные троянские программы используются в многокомпонентных троянских наборах для извещения своего «хозяина» об успешной инсталляции троянских компонент в атакуемую систему.

Реестр операционной системы Windows– это большая база данных, где хранится информация о конфигурации системы. Этой информацией пользуются как операционная система Windows, так и другие программы. Реестр содержит данные, к которым Windows XP постоянно обращается во время загрузки, работы и её завершения, а именно:

профили всех пользователей, то есть их настройки;

конфигурация оборудования, установленного в операционной системе.

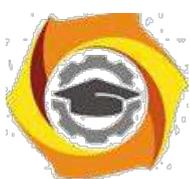
данные об установленных программах и типах документов, создаваемых каждой программой;

свойства папок и значков программ;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

данные об используемых портах.



Реестр имеет иерархическую древовидную структуру, состоящую из разделов, подразделов и ключей (параметров).

8. некоторых случаях восстановить работоспособность системы после сбоя можно, загрузив работоспособную версию реестра, но для этого, естественно, необходимо иметь копию реестра. Основным средством для просмотра и редактирования записей реестра служит специализированная утилита «**Редактор реестра**».

Файл редактора реестра находится в папке Windows. Называется он **regedit.exe**. Для того, чтобы запустить эту программу, необходимо выбрать **Пуск–Выполнить–regedit.exe**. После запуска появится окно редактора реестра. Вы увидите список из 5 разделов (рис. 11):

- HKEY_CLASSES_ROOT.
- HKEY_CURRENT_USER.
- HKEY_LOCAL_MACHINE.
- HKEY_USERS.
- HKEY_CURRENT_CONFIG.

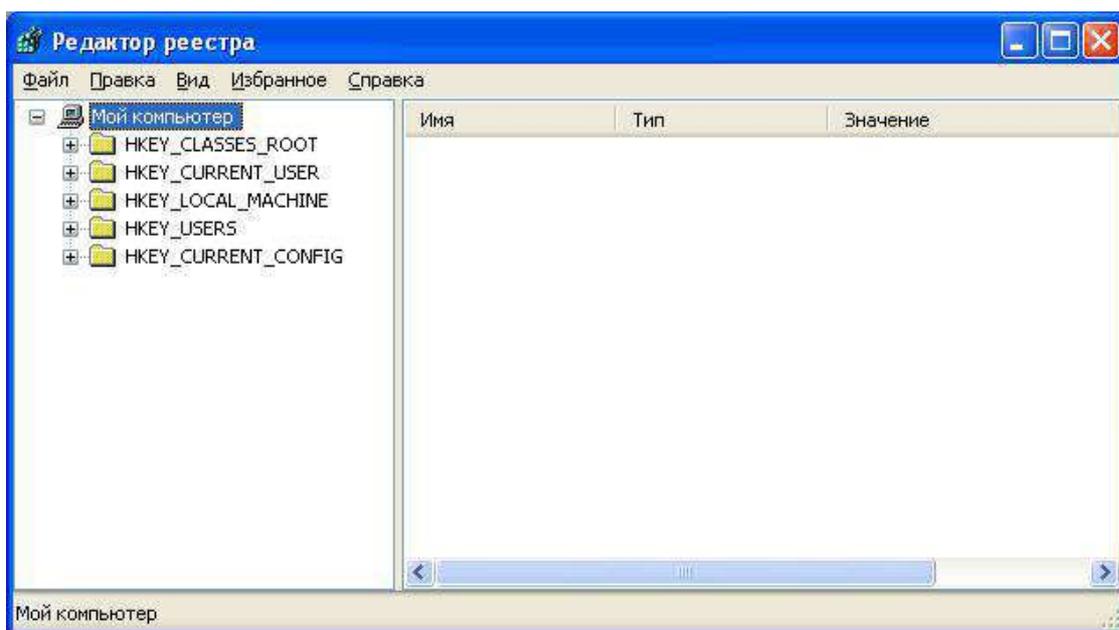


Рис. 11

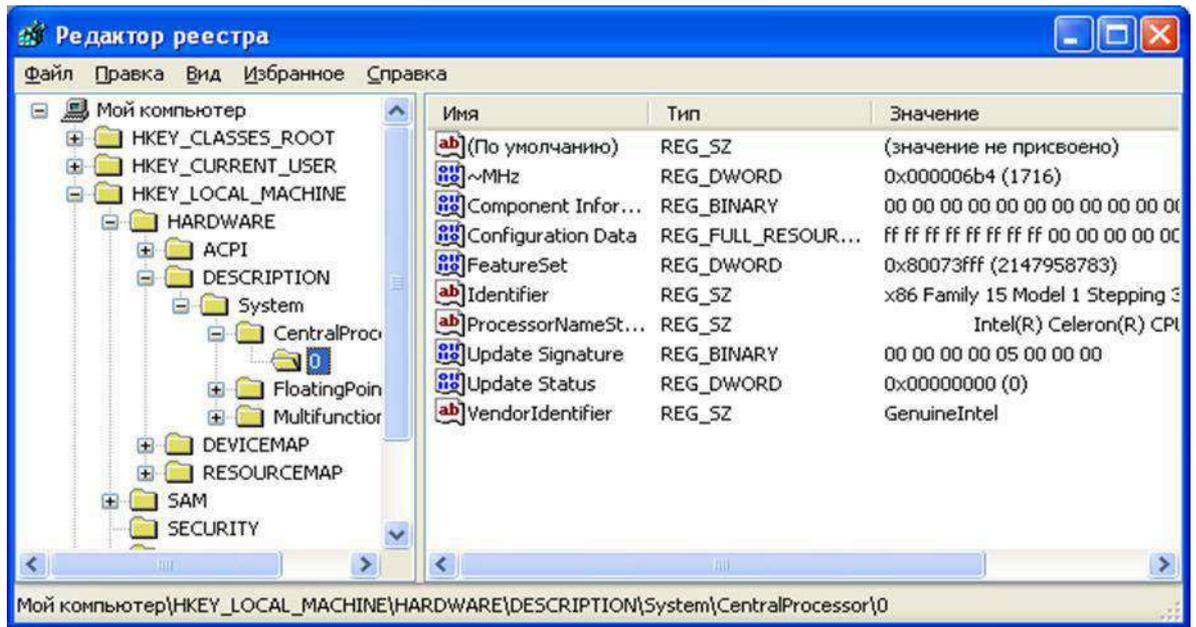


Рис. 12

Работа с разделами реестра аналогична работе с папками в Проводнике. Конечным элементом дерева реестра являются ключи или параметры, делящиеся на три типа (рис. 12):

строковые (напр. «C:\Windows»);

двоичные (напр. 10 82 AO 8F);

DWORD - этот тип ключа занимает 4 байта и отображается в шестнадцатеричном и в десятичном виде (например, 0x00000020 (32)).

В Windows системная информация разбита на так называемые ульи (hive). Это обусловлено принципиальным отличием концепции безопасности этих операционных систем. Имена файлов ульев и пути к каталогам, в которых они хранятся, расположены в разделе

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\hivelist(рис. 13).

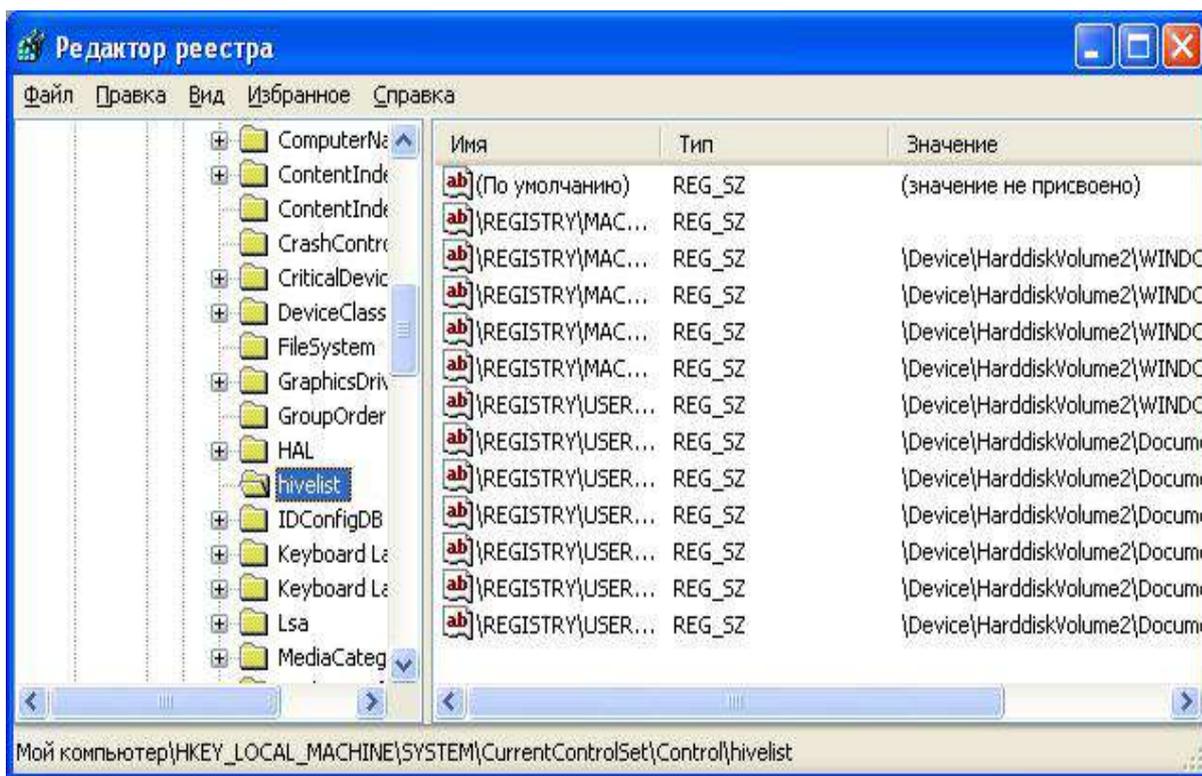
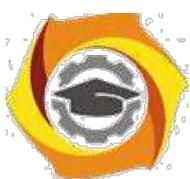


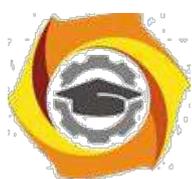
Рис. 13

/ таблице 1 даны краткие описания ульев реестра и файлов, в которых хранятся параметры безопасности.

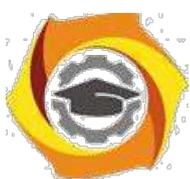
Таблица 5

Характеристика основных разделов системного реестра

HKEY_LOCAL_MACHINE\SAM	Содержит информацию SAM (Security Access Manager), хранящуюся в файлах SAM, SAM.LOG, SAM.SAV в папке %System-root%\System32\Config
HKEY_LOCAL_MACHINE\SECURITY	Содержит информацию безопасности в файлах SECURITY, SECURITY.LOG, SECURITY.SAV в папке \%\Systemroot%\System32\Config
HKEY_LOCAL_MACHINE\SYSTEM	Содержит информацию об аппаратных профилях этого подраздела. Информация хранится в файлах SYSTEM, SYSTEM.LOG, SYSTEM.SAV в папке \%\Systemroot%\System32\Config
HKEY_CURRENT_CONFIG	Содержит информацию о подразделе System этого улья, которая хранится в файлах SYSTEM.SAV и SYSTEM.ALT в



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**



	папке <code>\%Systemroot%\System32\Config</code>
HKEY_USERS\ DEFAULT	Содержит информацию, которая будет использоваться для создания профиля нового пользователя, впервые регистрирующегося в системе. Информация хранится в файлах <code>DEFAULT</code> , <code>DEFAULT.LOG</code> , <code>DEFAULT.SAV</code> в папке <code>\%Systemroot%\System32\Config</code>
HKEY_CURRE NT_USER	Содержит информацию о пользователе, зарегистрированном в системе на текущий момент. Эта информация хранится в файлах <code>NTUSER.DAT</code> и <code>NTUSER.DAT.LOG</code> , расположенных в каталоге <code>\%Systemroot%\Profiles\User name</code> , где <code>User name</code> – имя пользователя

Задание: проверить потенциальные места записей «троянских программ» в системном реестре операционной системы Windows 2000 (XP).

Алгоритм выполнения работы

Потенциальными местами записей «троянских программ» в системном реестре являются разделы, описывающие программы, запускаемые автоматически при загрузке операционной системы от имени пользователей и системы.

в Запустите программу **regedit.exe**.

в В открывшемся окне выберите ветвь **HKEY_LOCAL_MACHINE** и далее **Software\Microsoft\WindowsNT\CurrentVersion\Winlogon** (щелкнуть по значку «папка»).

в В правой половине открытого окна программы **regedit.exe** появится список ключей.

в Найдите ключ **Userinit(REG_SZ)** и проверьте его содержимое.

в По умолчанию (исходное состояние) 151 этот ключ содержит следующую запись **C:\WINDOWS\system32\userinit.exe** (рис. 14).

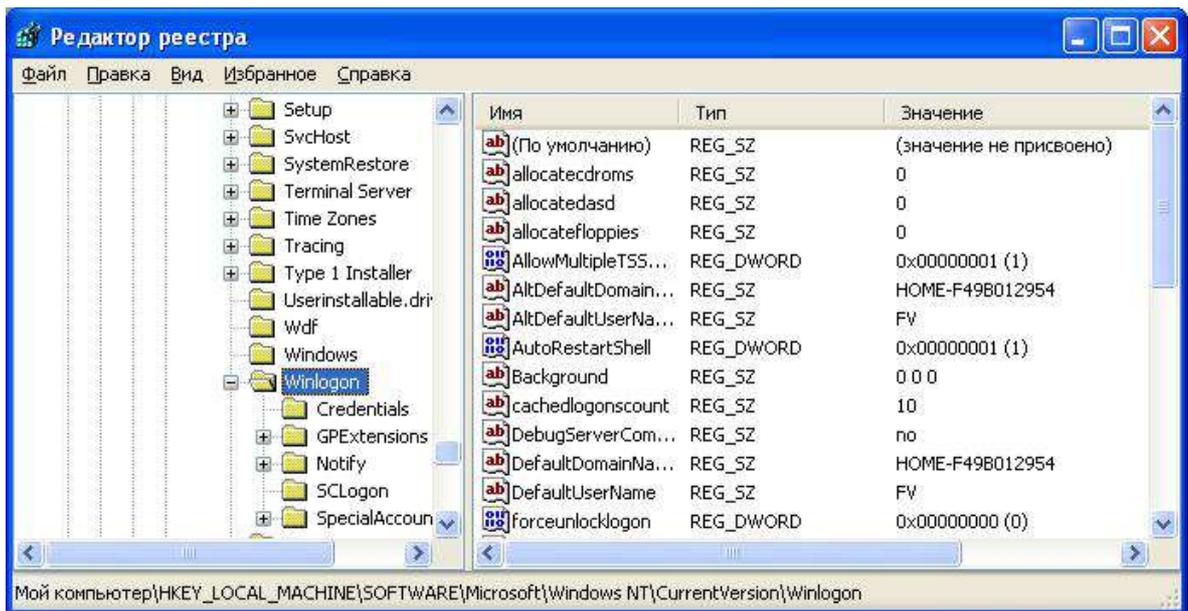


Рис. 14

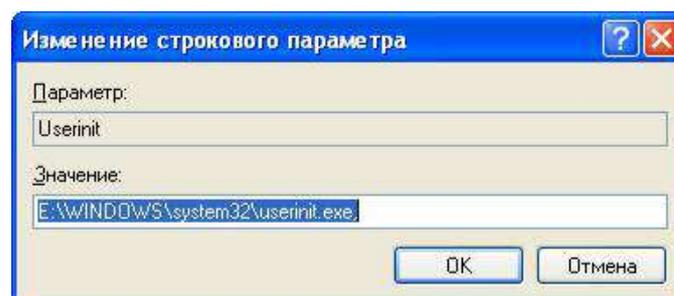
4) Если в указанном ключе содержатся дополнительные записи, то это могут быть «троянские программы».

5) В этом случае проанализируйте место расположения программы, обратите внимание на время создания файла и сопоставьте с Вашими действиями в это время.

4) Если время создания файла совпадает со временем Вашей работы в Интернете, то возможно, что в это время Ваш компьютер был заражен «троянским конем».

9. Для удаления этой записи необходимо дважды щелкнуть на названии ключа (или при выделенном ключе выбрать команду **Изменить** из меню **Правка** программы **regedit.exe**).

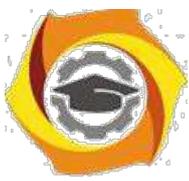
10. В открывшемся окне в поле **Значение** (рис. 15) удалите ссылку на подозрительный файл.





**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Рис. 15



10. Закройте программу **regedit.exe**.
11. Перейдите в папку с подозрительным файлом и удалите его.
12. Перезагрузите операционную систему и выполните пункты задания 1-4.

10) Если содержимое рассматриваемого ключа не изменилось, то предполагаемый «троянский конь» удален из Вашей системы.

Еще одним потенциальным местом записей на запуск «троянских программ» является раздел автозапуска **Run**.

Для его проверки выполните следующее.

- Запустите программу **regedit.exe**.
- В открывшемся окне выберите ветвь **HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\ ... (REG_SZ)** (рис. 16).

3.

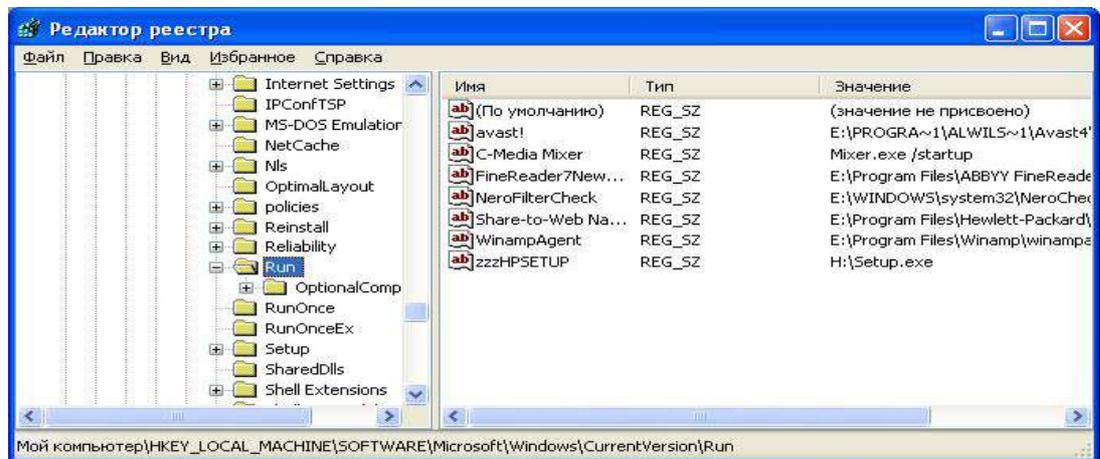


Рис. 16

3. В рассматриваемом примере автоматически запускается резидентный антивирус и его планировщик заданий, а также утилита, относящаяся к программе Nero (запись на CD).

6. Если в указанном разделе есть записи вызывающие подозрения, то выполните пункты 6-14 предыдущего задания.

Задания для самостоятельной работы

3 Проверьте содержимое ключа **HKEY_LOCAL_MACHINE\ Software\ Microsoft\ WindowsNT\ CurrentVersion\ Winlogon\ System (REG_SZ)**.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

45

7. Зафиксируйте этапы работы, используя команду PrintScreen клавиатуры.
16. Составьте отчет о результатах проверки.



Лабораторная работа № 4. Структура комплексной системы защиты информации от несанкционированного доступа (НСД); мониторинг и контроль окружающей среды; ведение специальной информационной базы данных КСИБ.

Цель работы: ознакомиться со структурой комплексной системы защиты информации от несанкционированного доступа.

На рынке защиты информации предлагается много отдельных инженерно-технических, программно-аппаратных, криптографических средств защиты информации. В литературе по защите информации можно найти описание методов и средств на их основе, теоретических моделей защиты. Однако для того, чтобы создать на предприятии условия эффективной защиты информации, необходимо объединить отдельные средства защиты в систему. При этом надо помнить, что главным элементом этой системы является человек. Причем человек является ключевым элементом системы и вместе с тем самым трудно формализуемым и потенциально слабым ее звеном.

Создание системы защиты информации (СЗИ) не является главной задачей предприятия, как, например, производство продукции и получение прибыли. Поэтому создаваемая СЗИ не должна приводить к ощутимым трудностям в работе предприятия, а создание СЗИ должно быть экономически оправданным. Тем не менее она должна обеспечивать защиту важных информационных ресурсов предприятия от всех реальных угроз.

В книге предложен комплексный подход к организации защиты информации (ЗИ) на предприятии. При этом объектом исследования является не только информационная система, но и предприятие в целом.

Рассматриваются концептуальные основы защиты информации, раскрывающие сущность, цели, структуру и стратегию защиты.

Анализируются источники, способы и результаты дестабилизирующего воздействия на информацию, а также каналы и методы несанкционированного



доступа к информации. Определяются методологические подходы к организации и технологическому обеспечению защиты информации на предприятии. Представлена архитектура, этапы построения, принципы управления комплексной системой защиты информации (КСЗИ). Особое внимание уделено проблеме «человеческого фактора».

Предложенный подход к защите информации обеспечит целостное видение проблемы, повышение качества, следовательно, и надежности защиты информации.

Следует подчеркнуть, что автор умышленно уходит от понятия «информационная безопасность», используя термин «защита информации».

Информационную безопасность принято рассматривать как обеспечение состояния защищенности:

4. личности, общества, государства от воздействия недоброкачественной информации;
5. информации и информационных ресурсов от неправомерного и несанкционированного воздействия посторонних лиц;
6. информационных прав и свобод гражданина и человека.

Поскольку в книге не рассматриваются вопросы защиты от воздействия недобросовестной информации, автор посчитал необходимым использовать более «узкий» термин.

- Сущность и задачи комплексной системы защиты информации

1.1. Подходы к проектированию систем защиты информации

Бытует мнение, что проблемы защиты информации относятся исключительно к информации, обрабатываемой компьютером. Это, по-видимому, связано с тем, что компьютер, и в частности персональный компьютер, является «ядром», центром хранения информации. Объект информатизации, по отношению к которому направлены действия по защите информации, представляется более широким понятием по сравнению с персональным компьютером. Что же представляет собой объект информатизации и каково его место на предприятии?



ГОСТ РФ 51275-99 определяет объект информатизации как «совокупность информационных ресурсов, средств и систем обработки информации, используемых в соответствии с заданной информационной технологией, средств обеспечения объекта информатизации, помещений или объектов (зданий, сооружений, технических средств), в которых они установлены, или помещения и объекты, предназначенные для ведения конфиденциальных переговоров»^[1].

Слово «совокупность» в данном определении указывает на то, что объект информатизации это единая информационная система, охватывающая в целом предприятие, учреждение, организацию.

В реальной жизни все эти отдельные «объекты информатизации» расположены в пределах одного предприятия и представляют собой единый комплекс компонентов, связанных общими целями, задачами, структурными отношениями, технологией информационного обмена и т. д.

Современное предприятие — большое количество разнородных компонентов, объединенных в сложную систему для выполнения поставленных целей, которые в процессе функционирования предприятия могут модифицироваться. Многообразие и сложность влияния внутренних и внешних факторов, которые часто не поддаются строгой количественной оценке, приводят к тому, что эта сложная система может обретать новые качества, не свойственные составляющим ее компонентам.

Характерной особенностью подобных систем является прежде всего наличие человека в каждой из составляющих ее подсистем и отдаленность (разделенность) человека от объекта его деятельности. Это происходит в связи с тем, что множество компонентов, составляющих объект информатизации, интегрально может быть представлено совокупностью трех групп систем: 1) люди (биосоциальные системы); 2) техника (технические системы и помещения, в которых они расположены); 3) программное обеспечение, которое является интеллектуальным посредником между человеком и техникой (интеллектуальные системы). Совокупность этих трех групп образует



социотехническую систему. Такое представление о социотехнических системах является достаточно широким и может быть распространено на многие объекты. Круг наших интересов ограничивается исследованием безопасности систем, предназначенных для обработки поступающей на их вход информации

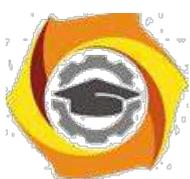
5. выдачи результата, т. е. социотехнических систем информационного типа.

Если обратиться к истории этой проблемы, то можно условно выделить три периода развития средств защиты информации (ЗИ):

- первый относится к тому времени, когда обработка информации осуществлялась по традиционным (ручным, бумажным) технологиям;
- второй — когда для обработки информации на регулярной основе применялись средства электронной вычислительной техники первых поколений;
- третий — когда использование средств электронно-вычислительной техники приняло массовый и повсеместный характер (появление персональных компьютеров).

8) 60–70 гг. проблема защиты информации решалась достаточно эффективно применением в основном организационных мер. К ним относились: режимные мероприятия, охрана, сигнализация и простейшие программные средства защиты информации. Эффективность использования этих средств достигалась за счет концентрации информации в определенных местах (спец. хранилища, вычислительные центры), что способствовало обеспечению защиты относительно малыми средствами.

«Рассосредоточение» информации по местам хранения и обработки обострило ситуацию с ее защитой. Появились дешевые персональные компьютеры. Это дало возможность построения сетей ЭВМ (локальных, глобальных, национальных и транснациональных), которые могут использовать различные каналы связи. Эти факторы способствуют созданию высокоэффективных систем разведки и получения информации. Они нашли отражение и в современных предприятиях.



Современное предприятие представляет собой сложную систему, в рамках которой осуществляется защита информации.

Рассмотрим основные особенности современного предприятия:

- сложная организационная структура;
- многоаспектность функционирования;
- высокая техническая оснащенность;
- широкие связи по кооперации;
- необходимость расширения доступа к информации;
- всевозрастающий удельный вес безбумажной технологии обработки информации;
- возрастающий удельный вес автоматизированных процедур в общем объеме процессов обработки данных;
- важность и ответственность решений, принимаемых в автоматизированном режиме, на основе автоматизированной обработки информации;
- высокая концентрация в автоматизированных системах информационных ресурсов;
- большая территориальная распределенность компонентов автоматизированных систем;
- накопление на технических носителях огромных объемов информации;
- интеграция в единых базах данных информации различного назначения 9.различной принадлежности;
- долговременное хранение больших объемов информации на машинных носителях;
- непосредственный и одновременный доступ к ресурсам (в т. ч. и к информации) автоматизированных систем большого числа пользователей различных категорий и различных учреждений;
- интенсивная циркуляция информации между компонентами автоматизированных систем, в том числе и удаленных друг от друга.



Таким образом, создание индустрии переработки информации, с одной стороны, создает объективные предпосылки для повышения уровня производительности труда и жизнедеятельности человека, с другой стороны, порождает целый ряд сложных и крупномасштабных проблем. Одной из них является обеспечение сохранности и установленного статуса информации, циркулирующей и обрабатываемой на предприятии.

1.2. Понятие комплексной системы защиты информации

Работы по защите информации у нас в стране ведутся достаточно интенсивно и уже продолжительное время. Накоплен существенный опыт. Сейчас уже никто не думает, что достаточно провести на предприятии ряд организационных мероприятий, включить в состав автоматизированных систем некоторые технические и программные средства — и этого будет достаточно для обеспечения безопасности.

Главное направление поиска новых путей защиты информации заключается не просто в создании соответствующих механизмов, а представляет собой реализацию регулярного процесса, осуществляемого на всех этапах жизненного цикла систем обработки информации при комплексном использовании всех имеющихся средств защиты. При этом все средства, методы и мероприятия, используемые для ЗИ, наиболее рациональным образом объединяются в единый целостный механизм — причем не только от злоумышленников, но и от некомпетентных или недостаточно подготовленных пользователей и персонала, а также нештатных ситуаций технического характера.

Основной проблемой реализации систем защиты является:

— с одной стороны, обеспечение надежной защиты, находящейся в системе информации: исключение случайного и преднамеренного получения информации посторонними лицами, разграничение доступа к устройствам и ресурсам системы всех пользователей, администрации и обслуживающего персонала;



— с другой стороны, системы защиты не должны создавать заметных неудобств пользователям в ходе их работы с ресурсами системы.

Проблема обеспечения желаемого уровня защиты информации весьма сложная, требующая для своего решения не просто осуществления некоторой совокупности научных, научно-технических и организационных мероприятий и применения специальных средств и методов, а создания целостной системы организационно-технологических мероприятий и применения комплекса специальных средств и методов по ЗИ.

На основе теоретических исследований и практических работ в области ЗИ сформулирован системно-концептуальный подход к защите информации.

Под системностью как основной частью системно-концептуального подхода понимается:

- системность целевая, т. е. защищенность информации рассматривается как основная часть общего понятия качества информации;
- системность пространственная, предлагающая взаимоувязанное решение всех вопросов защиты на всех компонентах предприятия;
- системность временная, означающая непрерывность работ по ЗИ, осуществляемых в соответствии планам;
- системность организационная, означающая единство организации всех работ по ЗИ и управления ими.

Концептуальность подхода предполагает разработку единой концепции как полной совокупности научно обоснованных взглядов, положений и решений, необходимых и достаточных для оптимальной организации и обеспечения надежности защиты информации, а также целенаправленной организации всех работ по ЗИ.

Комплексный (системный) подход к построению любой системы включает в себя: прежде всего, изучение объекта внедряемой системы; оценку угроз безопасности объекта; анализ средств, которыми будем оперировать при построении системы; оценку экономической целесообразности; изучение самой системы, ее свойств, принципов работы и возможность увеличения ее



эффективности; соотношение всех внутренних и внешних факторов; возможность дополнительных изменений в процессе построения системы и полную организацию всего процесса от начала до конца.

Комплексный (системный) подход — это принцип рассмотрения проекта, при котором анализируется система в целом, а не ее отдельные части. Его задачей является оптимизация всей системы в совокупности, а не улучшение эффективности отдельных частей. Это объясняется тем, что, как показывает практика, улучшение одних параметров часто приводит к ухудшению других, поэтому необходимо стараться обеспечить баланс противоречий требований и характеристик.

Комплексный (системный) подход не рекомендует приступать к созданию системы до тех пор, пока не определены следующие ее компоненты:

9. *Входные элементы.* Это те элементы, для обработки которых создается система. В качестве входных элементов выступают виды угроз безопасности, возможные на данном объекте;

2. *Ресурсы.* Это средства, которые обеспечивают создание и функционирование системы (например, материальные затраты, энергопотребление, допустимые размеры и т. д.). Обычно рекомендуется четко определять виды и допустимое потребление каждого вида ресурса как в процессе создания системы, так и в ходе ее эксплуатации;

3. *Окружающая среда.* Следует помнить, что любая реальная система всегда взаимодействует с другими системами, каждый объект связан с другими объектами. Очень важно установить границы области других систем, не подчиняющихся руководителю данного предприятия и не входящих в сферу его ответственности.

Характерным примером важности решения этой задачи является распределение функций по защите информации, передаваемой сигналами в кабельной линии, проходящей по территориям различных объектов. Как бы ни устанавливались границы системы, нельзя игнорировать ее взаимодействие с окружающей средой, ибо в этом случае принятые решения могут оказаться



бессмысленными. Это справедливо как для границ защищаемого объекта, так и для границ системы защиты;

7. Назначение и функции. Для каждой системы должна быть сформулирована цель, к которой она (система) стремится. Эта цель может быть описана как назначение системы, как ее функция. Чем точнее и конкретнее указано назначение или перечислены функции системы, тем быстрее и правильнее можно выбрать лучший вариант ее построения. Так, например, цель, сформулированная в самом общем виде как обеспечение безопасности объекта, заставит рассматривать варианты создания глобальной системы защиты. Если уточнить ее, определив, например, как обеспечение безопасности информации, передаваемой по каналам связи внутри здания, то круг возможных решений существенно сузится. Следует иметь в виду, что, как правило, глобальная цель достигается через достижение множества менее общих локальных целей (подцелей). Построение такого «дерева целей» значительно облегчает, ускоряет и удешевляет процесс создания системы;

> *Критерий эффективности.* Необходимо всегда рассматривать несколько путей, ведущих к цели, в частности нескольких вариантов построения системы, обеспечивающей заданные цели функционирования. Для того чтобы оценить, какой из путей лучше, необходимо иметь инструмент сравнения — критерий эффективности. Он должен: характеризовать качество реализации заданных функций; учитывать затраты ресурсов, необходимых для выполнения функционального назначения системы; иметь ясный и однозначный физический смысл; быть связанным с основными характеристиками системы и допускать количественную оценку на всех этапах создания системы.

Таким образом, учитывая многообразие потенциальных угроз информации на предприятии, сложность его структуры, а также участие человека в технологическом процессе обработки информации, цели защиты информации могут быть достигнуты только путем создания СЗИ на основе комплексного подхода.

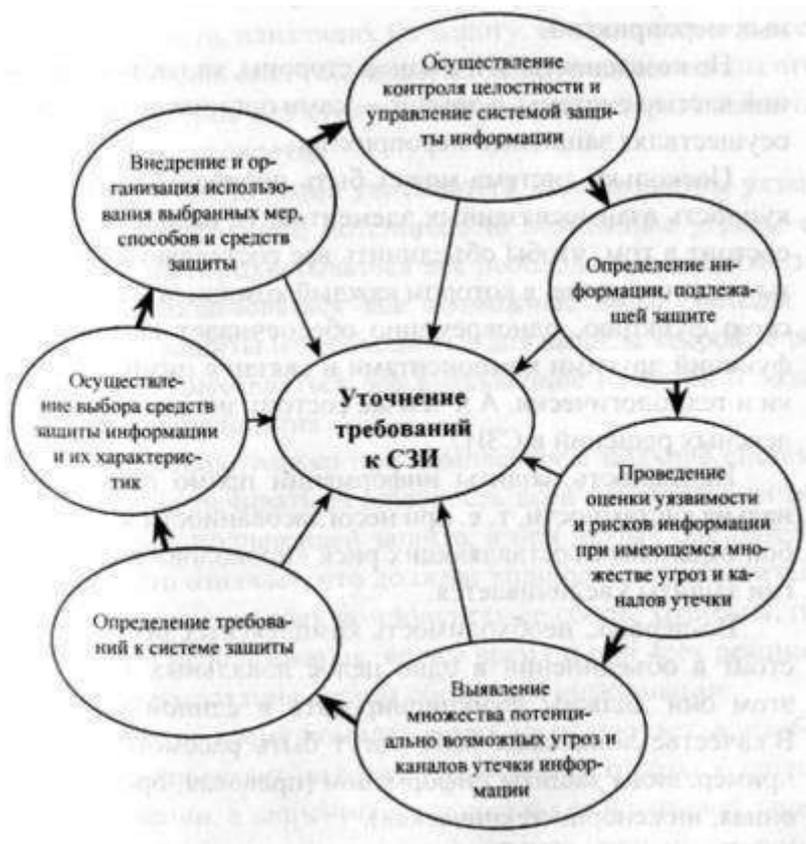
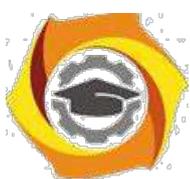


Рис. 1. Непрерывный цикл создания СЗИ

Процесс создания комплексной системы защиты информации может быть представлен в виде непрерывного цикла, так как это показано на рис. 1.

1.3. Назначение комплексной системы защиты информации

Главная цель создания системы защиты информации — ее надежность. Система ЗИ — это организованная совокупность объектов и субъектов ЗИ, используемых методов и средств защиты, а также осуществляемых защитных мероприятий.

Но компоненты ЗИ, с одной стороны, являются составной частью системы, с другой — сами организуют систему, осуществляя защитные мероприятия.

Поскольку система может быть определена как совокупность взаимосвязанных элементов, то назначение СЗИ состоит в том, чтобы объединить все составляющие защиты в единое целое, в котором каждый компонент, выполняя свою функцию, одновременно обеспечивает выполнение



функций другими компонентами и связан с ними логически и технологически. А в чем же состоит значимость комплексных решений в СЗИ?

Надежность защиты информации прямо пропорциональна системности, т. е. при несогласованности между собой отдельных составляющих риск «проколов» в технологии защиты увеличивается.

Во-первых, необходимость комплексных решений состоит в объединении в одно целое локальных СЗИ, при этом они должны функционировать в единой «связке». В качестве локальных СЗИ могут быть рассмотрены, например, виды защиты информации (правовая, организационная, инженерно-техническая).

Во-вторых, необходимость комплексных решений обусловлена назначением самой системы. Система должна объединить логически и технологически все составляющие защиты. Но из ее сферы выпадают вопросы полноты этих составляющих, она не учитывает всех факторов, которые оказывают или могут оказывать влияние на качество защиты. Например, система включает в себя какие-то объекты защиты, а все они включены или нет — это уже вне пределов системы.

Поэтому качество, надежность защиты зависят не только от видов составляющих системы, но и от их полноты, которая обеспечивается при учете всех факторов и обстоятельств, влияющих на защиту. Именно полнота всех составляющих системы защиты, базирующаяся на анализе таких факторов и обстоятельств, является вторым назначением комплексности.

При этом должны учитываться все параметры уязвимости информации, потенциально возможные угрозы ее безопасности, охватываться все необходимые объекты защиты, использоваться все возможные виды, методы и средства защиты и необходимые для защиты кадровые ресурсы, осуществляться все вытекающие из целей и задач защиты мероприятия.

В-третьих, только при комплексном подходе система может обеспечивать безопасность всей совокупности информации, подлежащей защите, и при любых обстоятельствах. Это означает, что должны защищаться все носители информации, во всех компонентах ее сбора, хранения, передачи и



использования, во все время и при всех режимах функционирования систем обработки информации.

то же время комплексность не исключает, а, наоборот, предполагает дифференцированный подход к защите информации, в зависимости от состава носителей, видов тайны, к которым отнесена информация, степени ее конфиденциальности, средств хранения и обработки, форм и условия проявления уязвимости, каналов и методов несанкционированного доступа к информации.

Таким образом, значимость комплексного подхода к защите информации состоит:

- в интеграции локальных систем защиты;
- в обеспечении полноты всех составляющих системы защиты;
- в обеспечении всеохватности защиты информации.

Исходя из этого, можно сформулировать следующее определение:

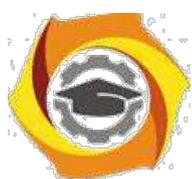
«Комплексная система защиты информации — система, полно и всесторонне охватывающая все предметы, процессы и факторы, которые обеспечивают безопасность всей защищаемой информации»^[2].

1.4. Принципы построения комплексной системы защиты информации

При построении любой системы необходимо определить принципы, в соответствии с которыми она будет построена. КСЗИ — сложная система, функционирующая, как правило, в условиях неопределенности, требующая значительных материальных затрат. Поэтому определение основных принципов КСЗИ позволит определить основные подходы к ее построению.

Принцип законности заключается в соответствии принимаемых мер законодательству РФ о защите информации, а в случае отсутствия соответствующих законов — другим государственным нормативным документам по защите.

5. соответствии с принципом полноты защищаемой информации защите подлежит не только информация, составляющая государственную,



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

коммерческую или служебную тайну, но и та часть несекретной информации,



утрата которой может нанести ущерб ее собственнику либо владельцу. Реализация этого принципа позволяет обеспечить и охрану интеллектуальной собственности.

Принцип обоснованности защиты информации заключается в установлении путем экспертной оценки целесообразности засекречивания и защиты той или другой информации, вероятных экономических и других последствий такой защиты исходя из баланса жизненно важных интересов государства, общества и граждан. Это, в свою очередь, позволяет расходовать средства на защиту только той информации, утрата или утечка которой может нанести действительный ущерб ее владельцу.

Принцип создания специализированных подразделений по защите информации заключается в том, что такие подразделения являются непременным условием организации комплексной защиты, поскольку только специализированные службы способны должным образом разрабатывать и внедрять защитные мероприятия и осуществлять контроль за их выполнением.

Принцип участия в защите информации всех соприкасающихся с ней лиц исходит из того, что защита информации является служебной обязанностью каждого лица, имеющего по роду выполняемой работы отношение к защищаемой информации, и такое участие дает возможность повысить качество защиты.

Принцип персональной ответственности за защиту информации требует, чтобы каждое лицо персонально отвечало за сохранность и неразглашение вверенной ему защищаемой информации, а за утрату или распространение такой информации оно несет уголовную, административную или иную ответственность.

Принцип наличия и использования всех необходимых правил и средств для защиты заключается в том, что КСЗИ требует, с одной стороны, участия в ней руководства предприятия и специальной службы защиты информации и всех исполнителей, работающих с защищаемой информацией, с другой стороны, использования различных организационных форм и методов защиты,



с третьей стороны, наличие необходимых материально-технических ресурсов, включая технические средства защиты.

Принцип превентивности принимаемых мер по защите информации предполагает априорное опережающее заблаговременное принятие мер по защите до начала разработки или получения информации. Из этого принципа вытекает, в частности, необходимость разработки защищенных информационных технологий.

Среди рассмотренных принципов едва ли можно выделить более, или менее важные. А при построении КСЗИ важно использовать их в совокупности.

Главная цель создания СЗИ — достижение максимальной эффективности защиты за счет одновременного использования всех необходимых ресурсов, методов и средств, исключающих несанкционированный доступ к защищаемой информации и обеспечивающих физическую сохранность ее носителей.

Организация — это совокупность элементов (людей, органов, подразделений) объединенных для достижения какой-либо цели, решения какой-либо задачи на основе разделения труда, распределения обязанностей и иерархической структуры.

СЗИ относится к системам организационно-технологического (социотехнического) типа, т. к. общую организацию защиты и решение значительной части задач осуществляют люди (организационная составляющая), а защита информации осуществляется параллельно с технологическим процессами ее обработки (технологическая составляющая).

Серьезным побудительным мотивом к проведению перспективных исследований в области защиты информации послужили те постоянно нарастающие количественные и качественные изменения в сфере информатизации, которые имели место в последнее время и которые, безусловно, должны быть учтены в концепциях защиты, информации.

Постановка задачи защиты информации в настоящее время приобретает ряд особенностей: во-первых, ставится вопрос о комплексной защите информации; во-вторых, защита информации становится все более актуальной



для массы объектов (больших и малых, государственной и негосударственной принадлежности); в-третьих, резко расширяется разнообразие подлежащей защите информации (государственная, промышленная, коммерческая, персональная и т. п.). Осуществление мероприятий по защите информации носит массовый характер, занимается этой проблемой большое количество специалистов различного профиля. Но успешное осуществление указанных мероприятий при такой их масштабности возможно только при наличии хорошего инструментария в виде методов и средств решения соответствующих задач. Разработка такого инструментария требует наличия развитых научно-методологических основ защиты информации.

Под научно-методологическими основами комплексной защиты информации (как решения любой другой проблемы) понимается совокупность принципов, подходов и методов (научно-технических направлений), необходимых и достаточных для анализа (изучения, исследования) проблемы комплексной защиты, построения оптимальных механизмов защиты и управления механизмами защиты в процессе их функционирования. Уже из приведенного определения следует, что основными компонентами научно-методологических основ являются принципы, подходы и методы. При этом под принципами понимается основное исходное положение какой-либо теории, учения, науки, мировоззрения; под подходом — совокупность приемов, способов изучения и разработки какой-либо проблемы; под методом — способ достижения какой-либо цели, решения конкретной задачи. Например, при реализации принципа разграничения доступа в качестве подхода можно выбрать моделирование, а в качестве метода реализации — построение матрицы доступа.

Общее назначение методологического базиса заключается в

- формировании обобщенного взгляда на организацию и управление КСЗИ, отражающего наиболее существенные аспекты проблемы;
- формировании полной системы принципов, следование которым обеспечивает наиболее полное решение основных задач;



— формировании совокупности методов, необходимых и достаточных для решения всей совокупности задач управления.

Предмет нашего исследования — рассмотрение различных аспектов обеспечения безопасности социотехнической системы, характерным примером которой является современный объект информатизации.

Поэтому состав научно-методологических основ можно определить следующим образом:

— так как речь идет об организации и построении КСЗИ, то общеметодологической основой будут выступать основные положения теории систем;

— так как речь идет об управлении, то в качестве научно-методической основы будут выступать общие законы кибернетики (как науки об управлении g) системах любой природы);

— так как процессы управления связаны с решением большого количества разноплановых задач, то в основе Должны быть принципы и методы моделирования больших систем и процессов их функционирования.

Состав научно-методологических основ комплексной системы защиты информации представлен на рис. 2.

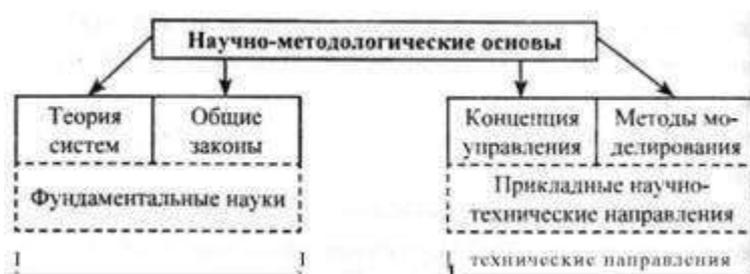


Рис. 2. Состав научно-методологических основ КСЗИ

2.2. Основные положения теории систем



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Считаю, что познать части без знания целого так же невозможно, как познать целое без знания его частей (Блез Паскаль 1623–1662).



Эти слова очень точно отражают суть теории систем. Но давайте по порядку.

Начнем с определения системы.

Система — совокупность или множество связанных между собой элементов.

Под системой может пониматься естественное соединение составных частей, самостоятельно существующих в природе, а также нечто абстрактное, порожденное воображением человека. Такой подход к определению понятия системы заранее предлагает существование связей между ее элементами.

Всякая система состоит из взаимосвязанных и взаимодействующих между собой и с внешней средой частей, в определенном смысле представляет собой замкнутое целое.

Система взаимодействует с внешней средой и может быть количественно оценена через свои входы и выходы.

Входами могут быть, в общем смысле, перерабатываемое сырье, его количество, состав, температура и т. д.; выходами могут быть количество готового продукта, его качество и т. п. (см. рис. 3).

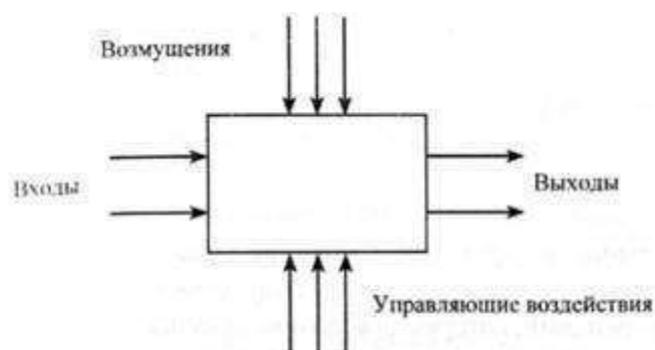


Рис. 3. Обобщенное представление системы



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Обычно система подвержена возмущениям, для их компенсации, т. е. для того, чтобы система работала в заданном направлении, используют управляющие воздействия.



Система — это достаточно сложный объект, который можно расчленить (провести декомпозицию) на составляющие элементы или подсистемы. Элементы связаны друг с другом и с окружающей средой объекта. Совокупность связей образует структуру системы. Система имеет алгоритм функционирования, направленный на достижение определенной цели.

Все системы можно условно разделить на малые и большие.

Малые системы однозначно определяются свойствами процесса и обычно ограничены одним типовым процессом, его внутренними связями, а также особенностями функционирования.

Большие системы представляют собой сложную совокупность малых (подсистем) систем и отличаются от них в количественном и качественном отношениях.

Рассмотрим составляющие системы и ее основные свойства.

Элементы — это объекты, части, компоненты системы. Причем их число ограничено.

Свойства — качества элементов, дающие возможность количественного описания системы, выражая ее в определенных величинах.

Связи — это то, что соединяет элементы и свойства системы в целое.

При анализе систем значительный интерес представляет изучение их структуры. Структура отражает наиболее существенные, устойчивые связи между элементами системы и их группами, которые обеспечивают основные свойства системы. То есть структура — это форма организации системы. Структура системы может претерпевать определенные изменения в зависимости от факторов (причин) внутренней и внешней природы, от времени.

Понятие «состояние» обычно выявляют на основании исследования, ситуационного анализа, исследуя, например, входные воздействия и выходные результаты системы.

Поведение системы характеризует возможность устойчивого, контролируемого перехода системы из одного состояния в другое.



Понятие «равновесие» определяется как способность системы в отсутствие внешних воздействий сохранять заранее заданное состояние.

Устойчивость характеризуется как способность системы возвращаться в состояние равновесия после того, как она была выведена из него под влиянием внешнего воздействия. На рисунке 4 схематично показана система в устойчивом и неустойчивом состояниях. Реально устойчивость систем может достигаться только в определенных пределах.

Понятие «развитие» характеризует совершенствование структуры и функций системы под влиянием внутренних факторов, в связи с чем поведение системы приобретает более упорядоченный и предсказуемый характер.

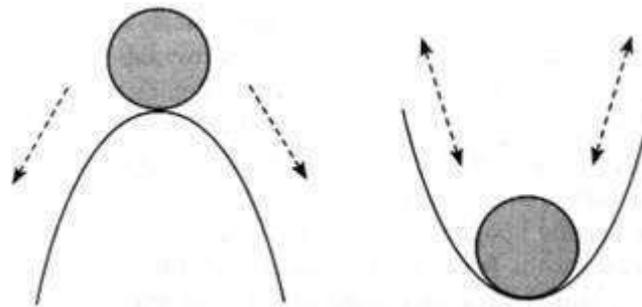


Рис. 4. Система в устойчивом состоянии (справа) и неустойчивом (слева)

Главное свойство системы в том, что она приобретает особенности, не свойственные ее элементам. Здесь можно привести множество примеров: компьютер, как система, состоящая из определенного набора деталей и программного обеспечения. И если все собрано и отлажено правильно (организована система), то получаем новые качества входящих в эту систему элементов. Это свойство называется принципом эмерджентности.

Общая теория систем — междисциплинарная область научных исследований, в задачи которой входит разработка обобщенных моделей систем, построение методологического аппарата, описание функционирования

11) поведения системных объектов, рассмотрение динамики систем, их поведения, развития, иерархического строения и процессов управления в



системах. Теория систем оперирует такими понятиями, как системный анализ и системный подход.

Системный анализ — это стратегия изучения сложных систем. В качестве метода исследования в нем используется математическое моделирование, а основным принципом является декомпозиция сложной системы на более простые подсистемы (принципы иерархии системы). В этом случае математическая модель строится по блочному принципу: общая модель подразделяется на блоки, которым можно дать сравнительно простые математические описания.

8) основе стратегии системного анализа лежат следующие общие положения: 1) четкая формулировка цели исследования; 2) постановка задачи по реализации этой цели и определение критерия эффективности решения задачи; 3) разработка развернутого плана исследования с указанием основных этапов и направлений решения задачи; 4) последовательное продвижение по всему комплексу взаимосвязанных этапов и возможных направлений; 5) организация последовательных приближений и повторных циклов исследований на отдельных этапах; 6) принцип нисходящей иерархии анализа и восходящей иерархии синтеза в решении составных задач и т. п.

Системный анализ позволяет организовать наши знания об объекте таким образом, чтобы помочь выбрать нужную стратегию либо предсказать результаты одной или нескольких стратегий, представляющихся целесообразными для тех, кто должен принимать решение.

С позиций системного анализа решаются задачи моделирования, оптимизации, управления и оптимального проектирования систем.

Особый вклад (важность) системного анализа в решении различных проблем заключается в том, что он позволяет выявить факторы и взаимосвязи, которые впоследствии могут оказаться весьма существенными, дает возможность спланировать методику наблюдений и построить эксперимент так, чтобы эти факторы были включены в рассмотрение, освещает слабые места гипотез и допущений. Как научный подход системный анализ создает



инструментарий познания физического мира и объединяет его в систему гибкого исследования сложных явлений.

Системный подход — направление методологии научного познания и социальной практики, в основе которого лежит рассмотрение объектов как систем. Системный подход ориентирует исследование на раскрытие целостности объекта, на выявление разных личных типов связей в нем и сведения в единую теоретическую картину.

Системный подход основан на представлении о системе как о чем-то целостном, обладающем новыми свойствами (качествами) по сравнению со свойствами составляющих ее элементов. Новые свойства при этом понимаются очень широко. Они могут выражаться, в частности, в способности решать новые проблемы или достигать новые цели. Для этого требуется определить границы системы, выделив ее из окружающего мира, и затем соответствующим образом изменить (преобразовать), или, говоря математическим языком, перевести систему в желаемое состояние. Академик В. М. Глушков выделил в системном подходе следующие этапы^[4]:

15. Постановка задачи (проблемы): определение Объекта исследования,
постановка целей, задание критериев для изучения объекта и управления им;
2. Очерчивание границ изучаемой системы и ее (первичная) структуризация. На этом этапе вся совокупность объектов и процессов, имеющих отношение к поставленной цели, разбивается на два класса — собственно изучаемая система и внешняя среда;
3. Составление математической модели изучаемой системы: параметризация системы, задание области определения параметров, установление зависимостей между введенными параметрами;
в Исследование построенной модели: прогноз развития изучаемой системы на основе ее модели, анализ результатов моделирования;
в Выбор оптимального управления.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Выбор оптимального управления как раз и позволяет перевести систему в желаемое (целевое) состояние и тем самым решить проблему.



Несмотря на четкую математическую трактовку системного подхода, он не получил, однако, однозначной практической интерпретации. В связи с этим развиваются несколько направлений его практической реализации. Наибольшее распространение получили АСУПовское и системотехническое направления, суть которых заключается в совершенствовании существующих систем управления. Для этого проводится их обследование (диагностическим анализ), выявляются недостатки и пути устранения последних, формируются мероприятия по совершенствованию систем, разрабатываются проекты систем, внедрение которых рассматривается как способ преобразования существующих систем управления.

Значительную роль в этих методах играют понятие системы, подсистемы, окружающей среды, классификация основных свойств и процессов в системах, классификация систем и т. д.

Остановимся на обобщенном определении системы.

Система, с одной стороны, может быть описана динамически как процесс, в с другой — статически, с точки зрения либо внешних, либо внутренних характеристик.

Кроме того, внутреннее строение системы может быть представлено в виде функциональных зависимостей и в виде структуры, реализующей эти зависимости.

Таким образом, можно выделить пять основных системных представлений: процессуальное, функциональное, макроскопическое, иерархическое и микроскопическое.

и процессуальном плане система рассматривается динамически как процесс, остальные системные представления отражают ее статический аспект.

и макроскопическом представлении описываются внешние характеристики системы, в функциональном, иерархическом и микроскопическом — внутренние.

Микроскопическое представление системы основано на понимании ее как совокупности взаимосвязанных элементов, неразложимых далее «кирпичиков».



Центральным понятием микрокосмического системного представления является понятие элемента. Конечно, в общем виде элемент лишь относительно неделим, однако для данной системы он является абсолютно неделимым. Элементы также могут быть рассмотрены как системы, но это будут системы другого типа, по отношению к исследуемой. Кроме того, система понимается как совокупность разнородных элементов, которые могут отличаться по принципу действия, техническому исполнению и ряду других характеристик. Система сводится к ансамблю простых частей.

Элементы системы обладают связями, которые объединяют их в целостную систему. Элементы могут существовать только в «связанном» виде — между элементами обязательно устанавливаются связи.

Например, в электрической цепи, если по ней не течет ток, нет электрических связей, следовательно, нет и элементов; когда цепь подключена к источнику электрической энергии, в ней образуются реальные электрические связи, и можно говорить о существовании элементов, которые они связывают.

Элементы в системе обязательно взаимодействуют, в результате одни свойства (переменные) изменяются, другие остаются неизменными (константы).

Важную роль в системных исследованиях играет поиск системообразующих связей, благодаря которым все элементы системы оказываются связанными воедино.

Функциональное представление системы связано с пониманием системы как совокупности функций (действий) для достижения определенной цели. Каждый элемент в системе выполняет определенную функцию.

Синонимом понятия «структура» для функционального представления служит понятие функциональной структуры, или организации.

Организация может быть реализована различными структурами (при этом функциональная сущность системы остается той же, меняется только способ реализации).



Для макроскопического представления характерно понимание системы как нерасчлененного целого. Здесь важно понятие системного окружения.

Под окружающей средой системы понимается совокупность всех объектов, изменение свойств которых влияет на систему и на которые влияет изменение свойств системы. Ни одна система объектов не может быть рассмотрена вне системного окружения. Системное окружение позволяет охарактеризовать систему множеством внешних связей (или внешней структурой), так и совокупностью внешних отношений.

Иерархическое представление системы (как иерархической упорядоченности) основано на понятии подсистемы, или единицы, которые следует отличать от понятия «элемент». Единица обладает функциональной спецификой целого (системы). Система может быть представлена в виде совокупности единиц, составляющих системную иерархию. (Единица может быть разложена на элементы.)

Можно выделить два типа функциональных связей между единицами системной иерархии — горизонтальные — между единицами одного уровня и вертикальные — между единицами различных уровней. Единицы каждого уровня описываются набором вертикальных и горизонтальных связей.

Процессуальное представление системы предполагает понимание системного объекта как совокупности процессов, характеризующихся последовательностью состояний во времени. Основным понятием здесь является понятие периода жизни — временного интервала, в течение которого функционирует данный процесс.

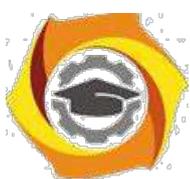
Комплексная система защиты информации — это система организационно-технологического типа. Она характеризуется рядом признаков.

КСЗИ — это система:

- искусственная, т. е. создана человеком;
- материальная, что подразумевает не только объективность ее существования, но и тот или иной уровень материальных и финансовых затрат на реализацию;



- открытая, т. е. возможно ее расширение;
- динамическая — подвержена старению, развитию, движению, прогрессу и регрессу, делению, слиянию и т. д.;
- вероятностная — система характеризуется вероятностью структуры, функции, целей, задач, ресурсов.



Лабораторная работа № 5. Настройка и использование межсетевого экрана. Создание VPN- подключения средствами Windows. Сетевые протоколы для секретной передачи данных.

Цель работы: ознакомиться с возможностями по настройке и использованию межсетевого экрана и создания VPN- подключения средствами Windows.

6. Настроить брандмауэр на работу с **Веб-сервером** (HTTP), FTP-сервером и зафиксировать соответствующее окно для отчета (PrinScreen).
5. Включить **журнал безопасности**.
6. После выполнения задания 1 и 2 подключиться к Интернету и посетить любой веб-сервер.
7. Просмотреть **журнал безопасности**.
8. Зафиксировать записи **журнала безопасности** для отчета. Сделать **ВЫВОДЫ**.

Рекомендации по выполнению заданий практического занятия

Для подготовки докладов на семинар рекомендуются следующие темы:

- и Технологии и методы защиты информации и информационных систем в таможенных органах РФ: идентификация и аутентификация.
- и Технологии и методы защиты информации и информационных систем в таможенных органах РФ: криптография и шифрование.
- и Технологии и методы защиты информации и информационных систем в таможенных органах РФ: регистрация и аудит.
- и Технологии и методы защиты информации и информационных систем в таможенных органах РФ: межсетевое экранирование; виртуальные частные сети (VPN).



- Основные направления и задачи обеспечения информационной безопасности таможенных органов РФ на период до 2020 года.

Для подготовки докладов формируются четыре творческие группы, каждая из которых готовит доклад по одной из перечисленных тем. По каждому докладу необходимо раскрыть содержание технологии (метода) защиты информации, его место роль в обеспечении защиты информации, характер угроз, в отношении которых применяется технология (метод) и примеры информационных систем (желательно таможенных информационных систем), в которых реализованы соответствующие технологии и методы защиты информации.

Доклад по последней теме готовит один из наиболее подготовленных студентов. Этот доклад завершает изучение дисциплины.

Все доклады должны сопровождаться презентациями.

Базовая теория по теме практического занятия и алгоритм его выполнения. Межсетевое экранирование повышает безопасность объектов внутренней сети за счет игнорирования неавторизованных запросов из внешней среды, тем самым, обеспечивая все составляющие информационной безопасности. Кроме функций разграничения доступа, экранирование обеспечивает регистрацию информационных обменов.

Функции экранирования выполняет **межсетевой экран** или брандмауэр (firewall), под которым понимают программную или программно-аппаратную систему, которая выполняет контроль информационных потоков, поступающих

- информационную систему и/или выходящих из нее, и обеспечивает защиту информационной системы посредством фильтрации информации. Фильтрация информации состоит в анализе информации по совокупности критериев и принятии решения о ее приеме и/или передаче.

Брандмауэр в **Windows** – это система защиты подключения к Интернету (Internet Connection Firewall, ICF), представляет собой программу настройки ограничений, регулирующих обмен данными между Интернетом и небольшой



сетью или локальным компьютером. Брандмауэр ICF необходимо установить для любого компьютера, имеющего прямое подключение к Интернету.

При включении брандмауэра для локального компьютера, подключенного к Интернету с помощью модема удаленного доступа, брандмауэр ICF обеспечивает защиту этого подключения.

Задание: Активизировать встроенный брандмауэр операционной системы Windows и настроить его параметры.

Все действия выполняются в режиме удаленного рабочего стола с учетной записью userXX, где XX номер.

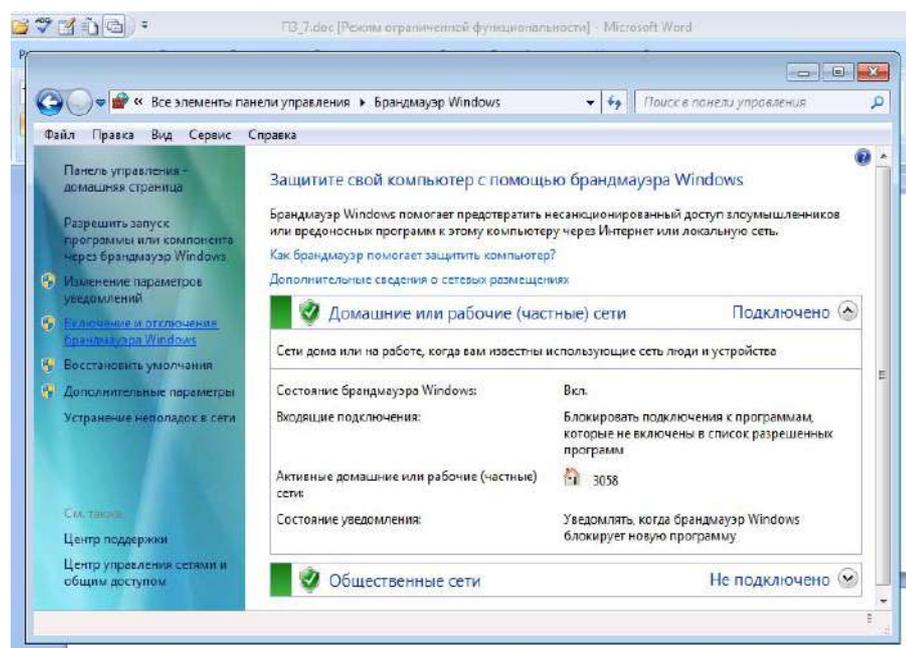
Алгоритм выполнения работы.

А) Активизация встроенного брандмауэра.

Для активизации встроенного брандмауэра операционной системы Windows выполните следующие действия.

1. Последовательно выполните **Пуск/Панель управления/Брандмауэр Windows.**

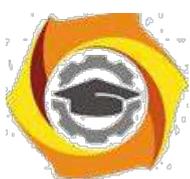
и В открывшемся окне выберите **Включение и отключение брандмауэр Windows (рисунок 1).**





**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Рисунок 1 – Окно настройки брандмауэра



3. Щелкните **ВключениебрандмауэраWindows** подкаждым сетевым размещением (рисунок 2), которое следует защитить, и нажмите кнопку **ОК**.

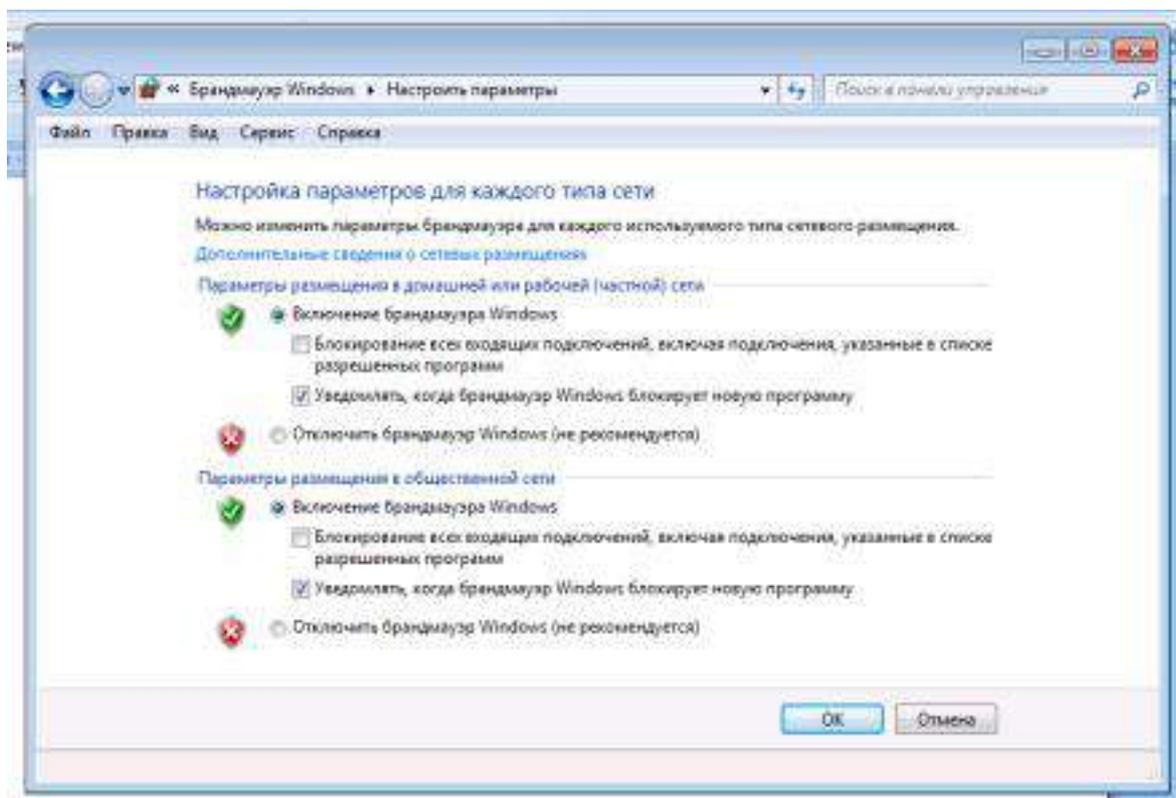


Рисунок 2 – Окно включения/отключения брандмауэра

- Если брандмауэр должен блокировать все, включая программы, которым ранее было разрешено устанавливать связь через брандмауэр, установите флажок **Блокирование всех входящих подключений**, включая подключения, указанные в списке разрешенных программ.

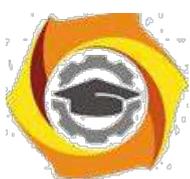
В) Настройка параметров брандмауэра.

Если требуется настроить разрешения для конкретной программы или компонента ОС, щелкните **Разрешить запуск программы или компонента через брандмауэр Windows** в левой панели.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- В открывшемся окне просмотрите разрешенные для соединения программы;



С Для добавления новой программы в открывшемся окне нажмите **Разрешить другую программу** и выберите любую программу.

С Нажмите **Ок** и убедитесь, что программа появилась в списке;
Для настройки брандмауэра в режиме расширенной безопасности выполните следующие действия.

4. Последовательно выполните **Пуск/Панель управления/Брандмауэр Windows**.

С В открывшемся окне выберите **Дополнительные параметры** (рисунок 3).

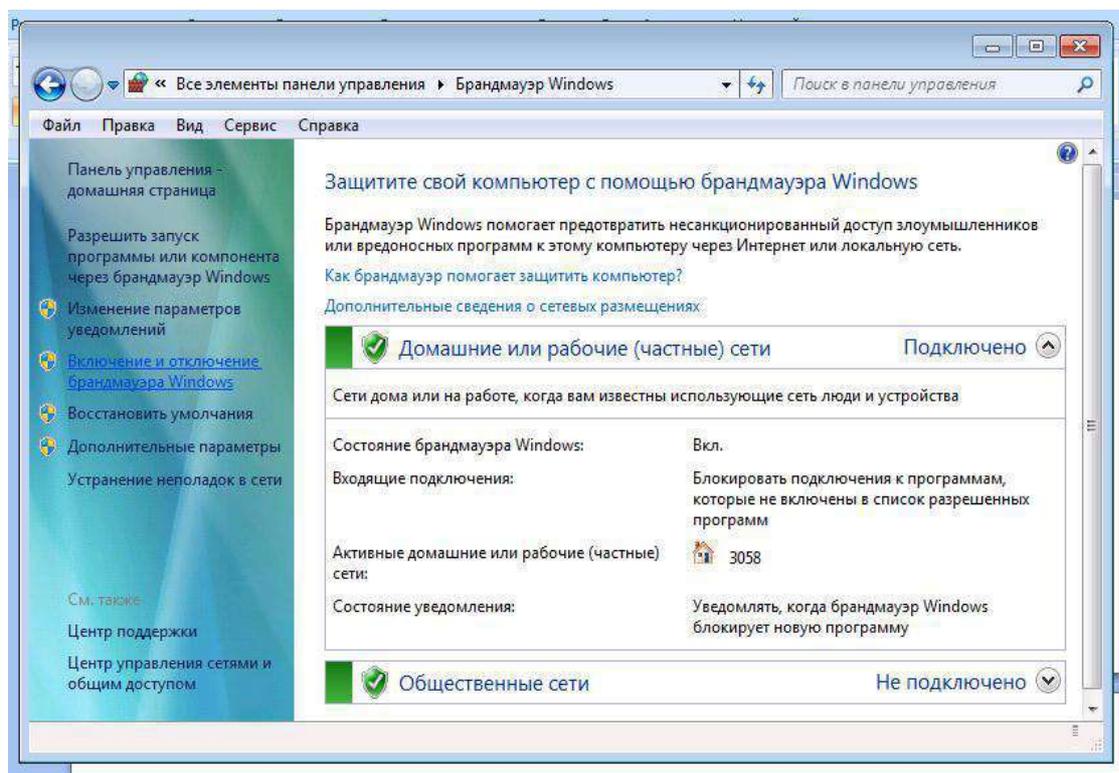


Рисунок 3 – Окно выбора дополнительных параметров брандмауэра

- В результате откроется окно **Брандмауэр Windows** в режиме **повышенной безопасности** (рисунок 4) с тремя полями.
- Изучите все закладки открытого окна, найдите закладку **Правила безопасности** и изучите порядок создания правил.

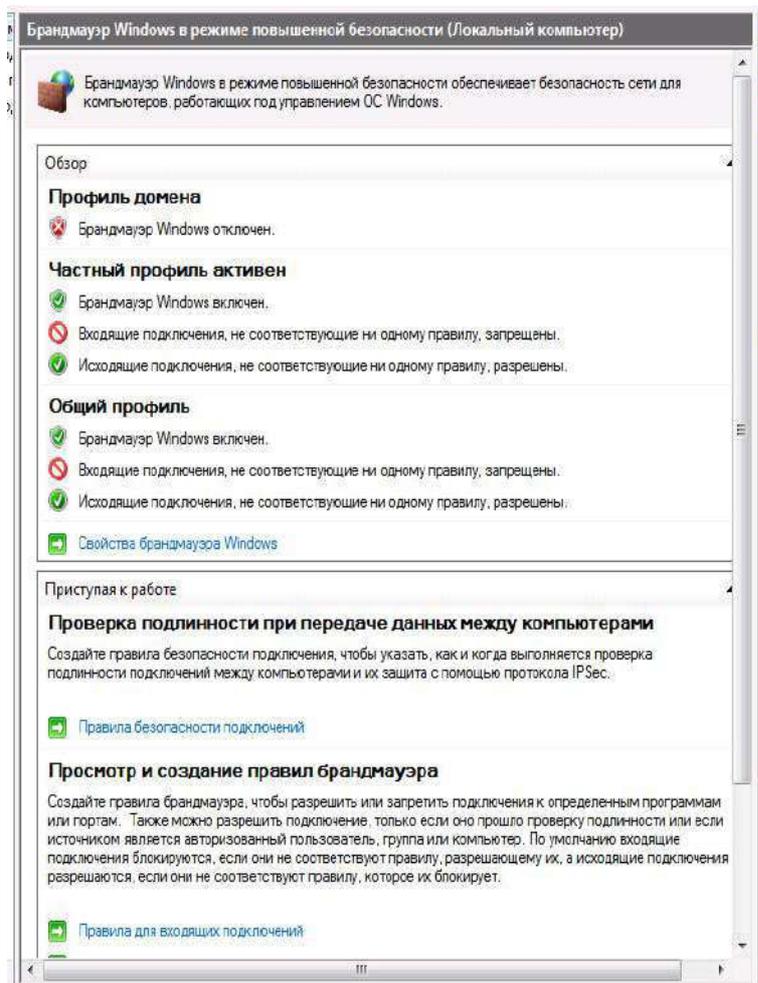
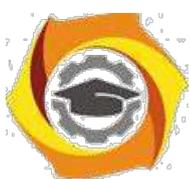


Рисунок 4 – Окно брандмауэра в режиме повышенной безопасности

Создание vpn-подключения средствами Windows 2000 (xp)

Технология виртуальных частных сетей (VPN – Virtual Private Network) является одним из эффективных механизмов обеспечения информационной безопасности при передаче данных в распределенных вычислительных сетях.

Виртуальные частные сети являются комбинацией нескольких самостоятельных сервисов (механизмов) безопасности: шифрования, экранирования и туннелирования.

Задание: создать VPN-подключение и выполнить его настройку. **Алгоритм выполнения работы А. Создание VPN-подключения.**

- Откройте компонент **Сетевые подключения**. Для этого выберите последовательно **Пуск – Панель управления – Сетевые подключения**.



2. Выберите пункт **Создание нового подключения** и нажмите кнопку **Далее**.

• В зависимости от операционной системы выполните следующие действия:

для Windows XP – в открывшемся окне выберите пункт **Подключить к сети на рабочем месте** (рис. 5, только для XP) и нажмите **Далее**. После этого выберите **Подключение к виртуальной частной сети** (рис. 6) и нажмите **Далее**.

для Windows 2000 – в открывшемся окне выберите пункт **Подключение к виртуальной частной сети через Интернет** и нажмите **Далее**.

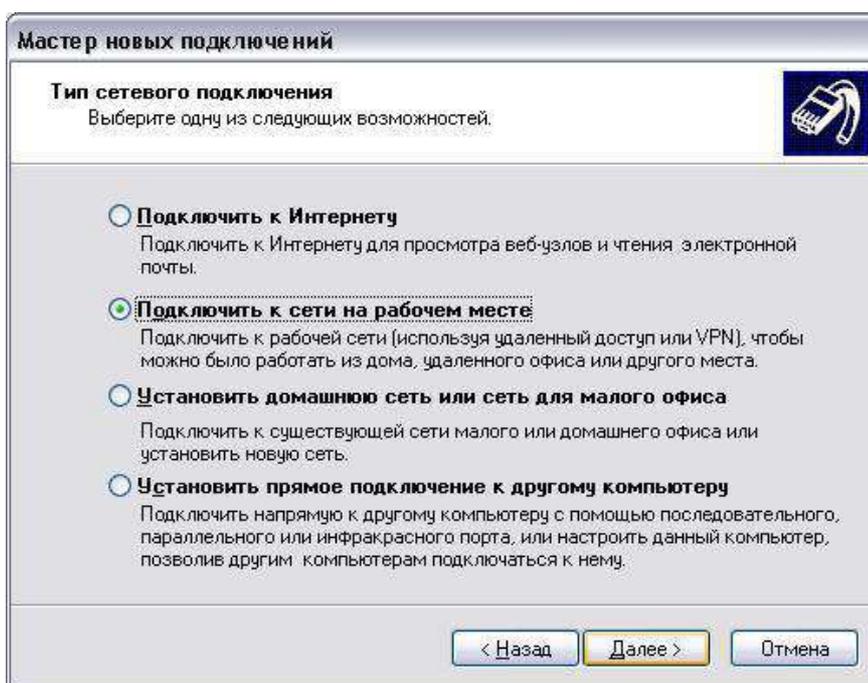


Рис. 5

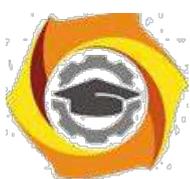
Введите имя подключения и перейдите к следующему шагу командой **Далее**.

Если перед установкой «туннельного доступа» требуется подключение



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- провайдеру услуг Интернета, то выберите (рис. 7).



Набрать номер для следующего предварительного подключения и, выбрав нужное подключение, нажмите Далее. В противном случае, выберите Не набирать номер для предварительного подключения и нажмите Далее.

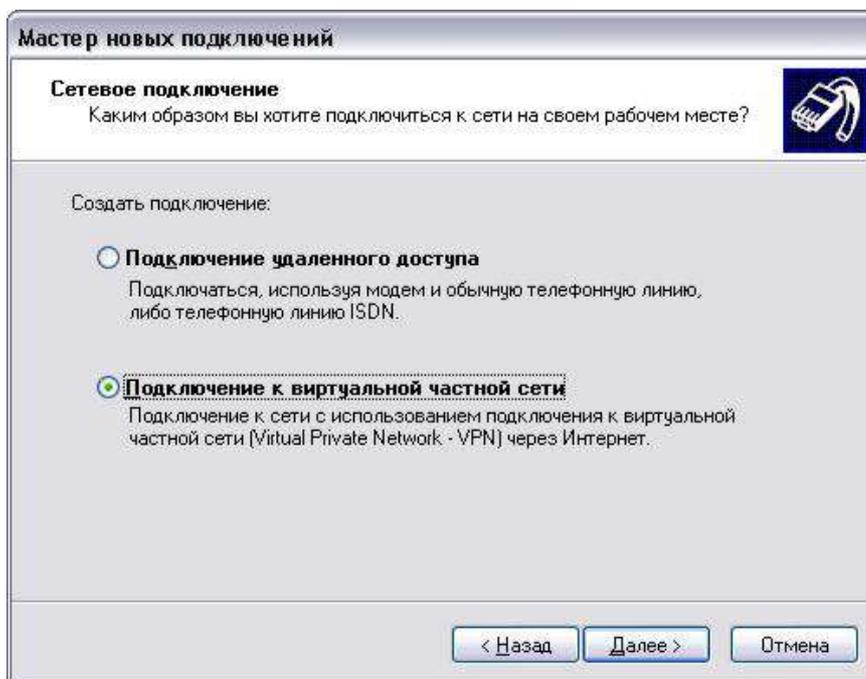
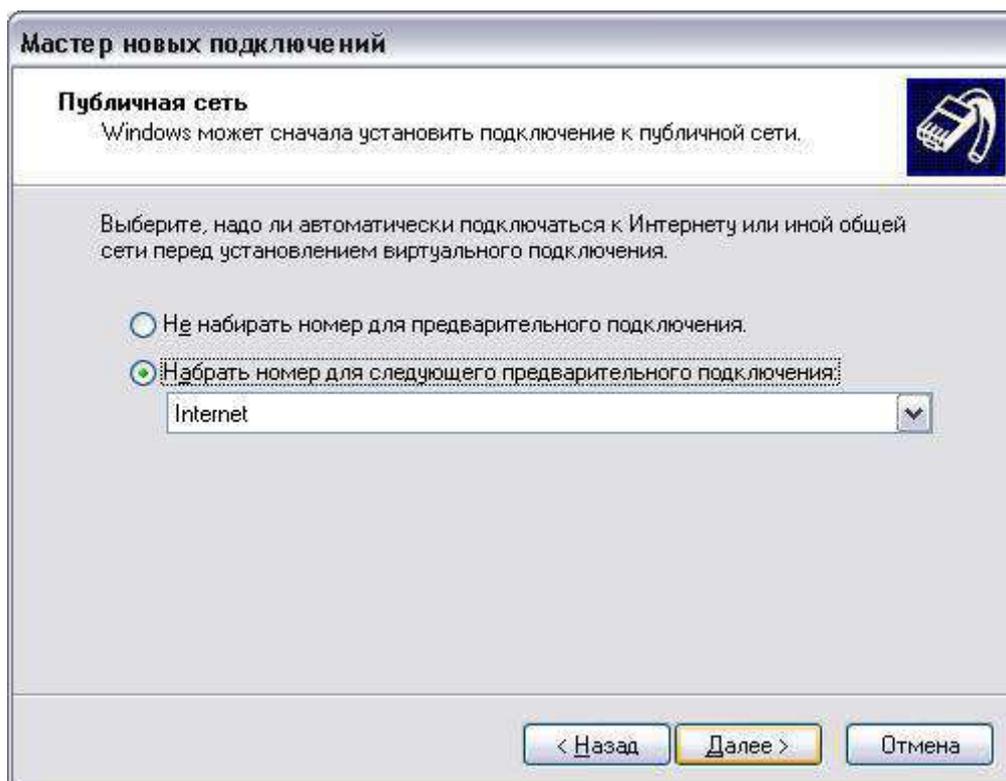
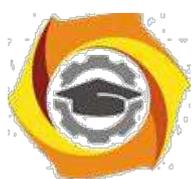


Рис. 6





**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Рис.7



- Введите имя узла (сети) или его IP-адрес (например 122.122.122.122), к которому идет подключение (рис.8).

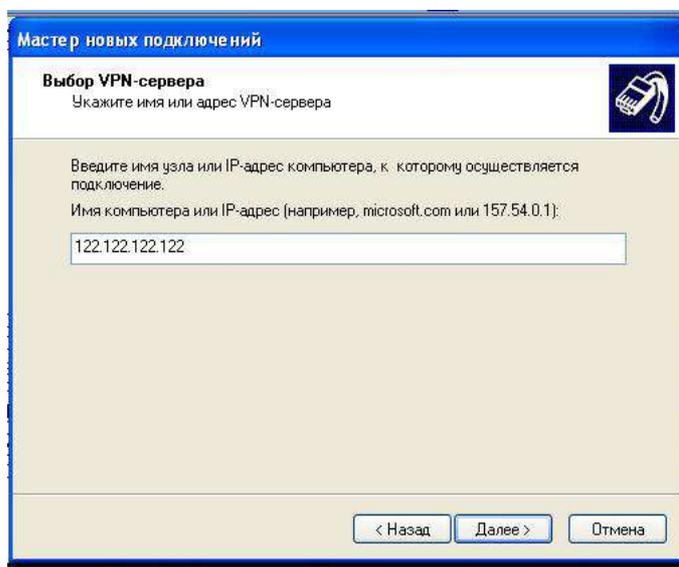
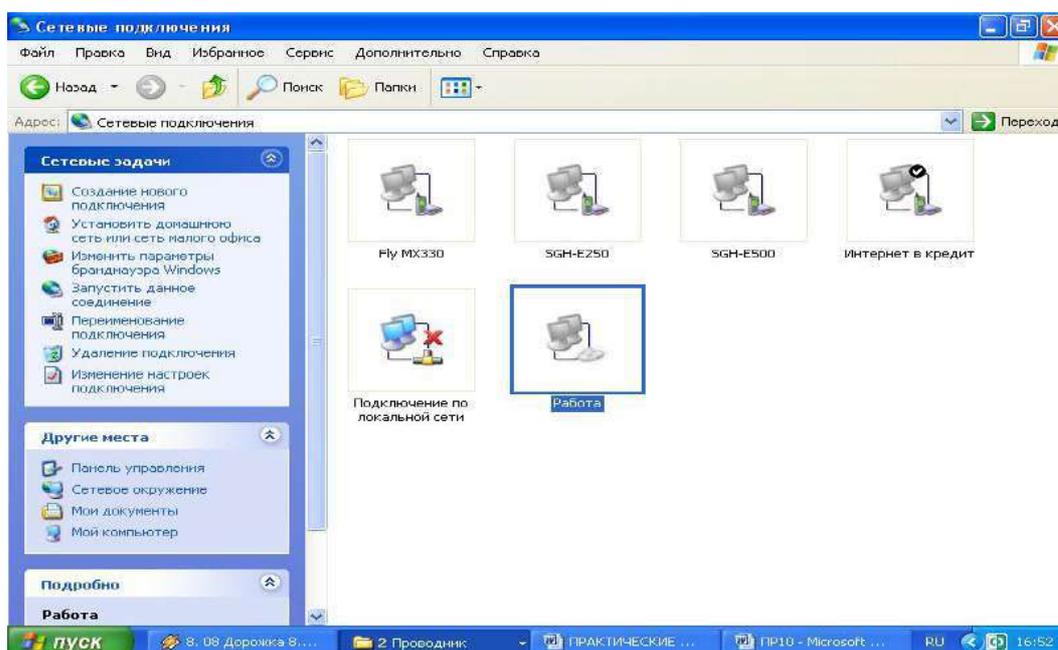
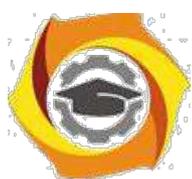


Рис. 8

- Завершите работу Мастера сетевых подключений.
- В результате в папке Подключения появится новое подключение (рис. 9).





**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Рис.9



В Для настройки параметров подключения выделите подключение VPN и вызовите его свойства из контекстного меню (нажатие правой клавиши мыши).

В Рассмотрите все имеющиеся параметры VPN-подключения и при необходимости воспользуйтесь соответствующими разделами справки.

Задания для самостоятельной работы

Создайте VPN-подключение к узлу с адресом 122.122.122.122 и зафиксируйте окно его свойств (Print Screen) на закладке Общие (как показано на рис. 5) в качестве отчета.

Контрольные вопросы

7. Какие механизмы безопасности используются при реализации VPN-подключения?
12. Что такое «туннель» и в чем состоит принцип «туннелирования»?
13. В чем заключаются защитные функции виртуальных частных сетей?

Практическое занятие №1. Нормативно-правовая база защиты компьютерных сетей от несанкционированного доступа. Компьютерные преступления и особенности их расследования.

Цель работы: ознакомиться с нормативно-правовой базой защиты компьютерных сетей от несанкционированного доступа.

Несанкционированный доступ – чтение, обновление или разрушение информации при отсутствии на это соответствующих полномочий.

Несанкционированный доступ осуществляется, как правило, с использованием чужого имени, изменением физических адресов устройств, использованием информации, оставшейся после решения задач, модификацией



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

программного и информационного обеспечения, хищением носителя информации, установкой аппаратуры записи.

Для успешной защиты своей информации пользователь должен иметь абсолютно ясное представление о возможных путях несанкционированного доступа. Основные типовые пути несанкционированного получения информации:

- хищение носителей информации и производственных отходов;
- копирование носителей информации с преодолением мер защиты;
- маскировка под зарегистрированного пользователя;
- мистификация (маскировка под запросы системы);
- использование недостатков операционных систем и языков

программирования;

· использование программных закладок и программных блоков типа "троянский конь";

- перехват электронных излучений;
- перехват акустических излучений;
- дистанционное фотографирование;
- применение подслушивающих устройств;



- злоумышленный вывод из строя механизмов защиты и т.д..

Для защиты информации от несанкционированного доступа применяются:

- 1) организационные мероприятия;
- 2) технические средства;
- 3) программные средства;
- 4) шифрование.

Организационные мероприятия включают в себя:

- пропускной режим;
- хранение носителей и устройств в сейфе (дискеты, монитор, клавиатура и т.д.);
- ограничение доступа лиц в компьютерные помещения и т.д..

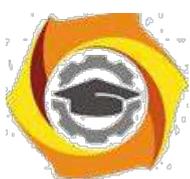
Технические средства включают в себя:

- фильтры, экраны на аппаратуру;
- ключ для блокировки клавиатуры;
- устройства аутентификации – для чтения отпечатков пальцев, формы руки, радужной оболочки глаза, скорости и приемов печати и т.д.;
- электронные ключи на микросхемах и т.д.

Программные средства включают в себя:

- парольный доступ – задание полномочий пользователя;
- блокировка экрана и клавиатуры с помощью комбинации клавиш в утилите Diskreet из пакета Norton Utilites;
- использование средств парольной защиты BIOS – на сам BIOS и на ПК в целом и т.д.

Шифрование – это преобразование (кодирование) открытой информации в зашифрованную, не доступную для понимания посторонних. Методы шифрования и расшифровывания сообщения изучает наука криптология, история которой насчитывает около четырех тысяч лет.



Защита информации в беспроводных сетях

Невероятно быстрые темпы внедрения в современных сетях беспроводных решений заставляют задуматься о надежности защиты данных[17].

Сам принцип беспроводной передачи данных заключает в себе возможность несанкционированных подключений к точкам доступа.

Не менее опасная угроза - вероятность хищения оборудования. Если политика безопасности беспроводной сети построена на MAC-адресах, то сетевая карта или точка доступа, украденная злоумышленником, может открыть доступ к сети.

Часто несанкционированное подключение точек доступа к ЛВС выполняется самими работниками предприятия, которые не задумываются о защите.

Решением подобных проблем нужно заниматься комплексно. Организационные мероприятия выбираются исходя из условий работы каждой конкретной сети. Что касается мероприятий технического характера, то весьма хорошей результат достигается при использовании обязательной взаимной аутентификации устройств и внедрении активных средств контроля.

В 2001 году появились первые реализации драйверов и программ, позволяющих справиться с шифрованием WEP. Самый удачный - PreShared Key. Но и он хорош только при надежной шифрации и регулярной замене качественных паролей (рис.1).



Рисунок 1 - Алгоритм анализа зашифрованных данных



Современные требования к защите

Аутентификация

В настоящее время в различном сетевом оборудовании, в том числе в беспроводных устройствах, широко применяется более современный способ аутентификации, который определен в стандарте 802.1x - пока не будет проведена взаимная проверка, пользователь не может ни принимать, ни передавать никаких данных.

Ряд разработчиков используют для аутентификации в своих устройствах протоколы EAP-TLS и PEAP, Cisco Systems, предлагает для своих беспроводных сетей, помимо упомянутых, следующие протоколы: EAP-TLS, PEAP, LEAP, EAP-FAST.

Все современные способы аутентификации подразумевают поддержку динамических ключей.

Главный недостаток LEAP и EAP-FAST - эти протоколы поддерживаются в основном в оборудовании Cisco Systems (рис. 2).

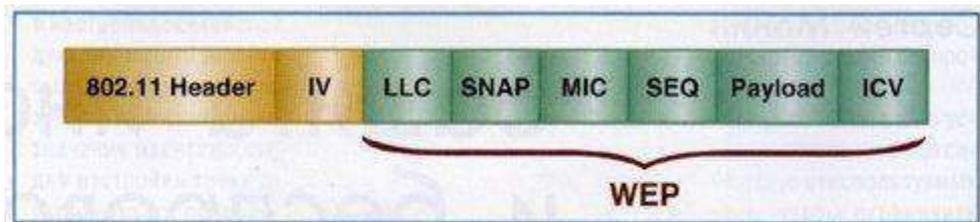


Рисунок 2 - Структура пакета 802.11x при использовании TKIP-PPK, MIC и шифрации по WEP.

Шифрование и целостность

На основании рекомендаций 802.11i Cisco Systems реализован протокол TKIP (Temporal Integrity Protocol), обеспечивающий смену ключа шифрования



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

РРК (Per Packet Keying) в каждом пакете и контроль целостности сообщений МІС (Message Integrity Check).



Другой перспективный протокол шифрования и обеспечения целостности - AES (Advanced Encryption Standard). Он обладает лучшей криптостойкостью по сравнению с DES и ГОСТ 28147-89. Он обеспечивает и шифрацию, и целостность.

Заметим, что используемый в нем алгоритм (Rijndael) не требует больших ресурсов ни при реализации, ни при работе, что очень важно для уменьшения времени задержки данных и нагрузки на процессор.

Стандарт обеспечения безопасности в беспроводных локальных сетях - 802,11i.

Стандарт Wi-Fi Protected Access (WPA) - это набор правил, обеспечивающих реализацию защиты данных в сетях 802.11х. Начиная с августа 2003 года соответствие стандартам WPA является обязательным требованием к оборудованию, сертифицируемому на звание Wi-Fi Certified.

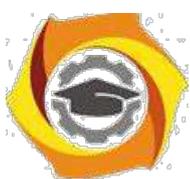
В спецификацию WPA входит измененный протокол TKIP-PPK. Шифрование производится на сочетании нескольких ключей - текущего и последующего. При этом длина IV увеличена до 48 бит. Это дает возможность реализовать дополнительные меры по защите информации, к примеру ужесточить требования к реассоциациям, реаутентификациям.

Спецификации предусматривают и поддержку 802.1х/EAP, и аутентификацию с разделяемым ключом, и, несомненно, управление ключами.

Рекомендуется распределять пользователей с разной степенью защищенности по разным виртуальным ЛС, и, в соответствии с этим, реализовывать политику безопасности (табл.3).

Таблица 3 - Способы реализации политики безопасности

Показатель	Способ			
	LEAP	EAP-FAST	PEAP	EAP-TLS
Поддержка современных ОС	Да	Да	Не все	Не все
Сложность ПО и ресурсоёмкость аутентификации	Низкая	Низкая	Средняя	Высокая



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

8

Сложность управления	Низкая *	Низкая	Средняя	Средняя
Single Sign on (единый логин в Windows)	Да	Да	Нет	Да
Динамические ключи	Да	Да	Да	Да
Одноразовые пароли	Нет	Да	Да	Нет
Поддержка баз пользователей не в формате MS Windows	Нет	Да	Да	Да
Fast Secure Роуминг	Да	Да	Нет	Нет
Возможность локальной аутентификации	Да	Да	Нет	Нет
* Сложность управления низкая, но необходима продуманная политика генерации паролей, что усложняет управление.				

При условии использования современного оборудования и ПО в настоящее время вполне возможно построить на базе стандартов серии 802.11x защищенную и устойчивую к атакам беспроводную сеть.

Почти всегда беспроводная сеть связана с проводной, а это, помимо необходимости защищать беспроводные каналы, необходимо обеспечивать защиты в проводных сетях. В противном случае сеть будет иметь фрагментарную защиту, что, по сути, является угрозой безопасности. Желательно использовать оборудование, имеющее сертификат Wi-Fi Certified, то есть подтверждающий соответствие WPA.

Нужно внедрять 802.11x/EAP/TKIP/MIC и динамическое управление ключами. В случае смешанной сети следует использовать виртуальные локальные сети; при наличии внешних антенн применяется технология виртуальных частных сетей VPN.

Необходимо сочетать как протокольные и программные способы защиты, так и административные.



Практическое занятие №2. Основные виды формальных моделей безопасности. Применение иерархического метода для построения защищенной операционной системы; исследование корректности систем защиты; методология обследования и проектирования защиты; модель политики контроля целостности.

Цель работы: ознакомиться с основными видами и формальными моделями безопасности.

Формальные модели безопасности

Наибольшее развитие получили два подхода, каждый из которых основан на своем видении проблемы безопасности и нацелен на решение определенных задач, — это формальное моделирование политики безопасности и криптография. Причем эти различные по происхождению и решаемым задачам подходы дополняют друг друга: криптография может предложить конкретные методы защиты информации в виде алгоритмов идентификации, аутентификации, шифрования и контроля целостности, а формальные модели безопасности предоставляют разработчикам основополагающие принципы, лежащие в основе архитектуры защищенной системы и определяющие концепцию ее построения.

Модель политики безопасности — формальное выражение политики безопасности. Формальные модели используются достаточно широко, потому что только с их помощью можно доказать безопасность системы, опираясь при этом на объективные и неопровержимые постулаты математической теории.

Основная цель создания политики безопасности ИС и описания ее в виде формальной модели — это определение условий, которым должно подчиняться поведение системы, выработка критерия безопасности и проведение формального доказательства соответствия системы этому критерию при соблюдении установленных правил и ограничений. На практике это означает, что только соответствующим образом уполномоченные пользователи получают



доступ к информации и смогут осуществлять с ней только санкционированные действия.

Среди моделей политик безопасности можно выделить два основных класса: дискреционные (произвольные) и мандатные (нормативные). В данном подразделе в качестве примера изложены основные положения наиболее распространенных политик безопасности, основанных на контроле доступа субъектов к объектам.

Дискреционная модель Харрисона—Руззо — Ульмана.

Модель безопасности Харрисона—Руззо — Ульмана, являющаяся классической дискреционной моделью, реализует произвольное управление доступом субъектов к объектам и контроль за распространением прав доступа.

В рамках этой модели система обработки информации представляется в виде совокупности активных сущностей — субъектов (множество S), которые осуществляют доступ к информации, пассивных сущностей — объектов (множество O), содержащих защищаемую информацию, и конечного множества прав доступа $R = \{r_1, \dots, r_n\}$, означающих полномочия на выполнение соответствующих действий (например, чтение, запись, выполнение).

Причем для того чтобы включить в область действия модели и отношения между субъектами, принято считать, что все субъекты одновременно являются и объектами. Поведение системы моделируется с помощью понятия «состояние». Пространство состояний системы образуется декартовым произведением множеств составляющих ее объектов, субъектов и прав — OSR . Текущее состояние системы Q в этом пространстве определяется тройкой, состоящей из множества субъектов, множества объектов и матрицы прав доступа M , описывающей текущие права доступа субъектов к объектам, — $Q = (S, O, M)$. Строки матрицы соответствуют субъектам, а столбцы — объектам, поскольку множество объектов включает в себя множество субъектов, матрица имеет вид прямоугольника. Любая ячейка матрицы $M[s, o]$ содержит набор прав субъекта s к объекту o , принадлежащих множеству



прав доступа R . Поведение системы во времени моделируется переходами между различными состояниями. Переход осуществляется путем внесения изменений в матрицу M с помощью команд следующего вида:

$$\left\{ \begin{array}{l} \text{command } a(x_1, \dots, x_k) \\ \text{If } r_1 \text{ in } M[x_{s_1}, x_{o_1}] \text{ and (условия выполнения команды)} \\ r_1 \text{ in } M[x_{s_2}, x_{o_2}] \text{ and} \\ \cdot \\ \cdot \\ r_m \text{ in } M[x_{s_m}, x_{o_m}] \\ \text{then} \\ op_1, op_2, \dots, op_n \text{ (операции, составляющие команду)} \end{array} \right.$$

Здесь a — имя команды; x_i — параметры команды, являющиеся идентификаторами субъектов и объектов; s_i и o_i — индексы субъектов и объектов в диапазоне от 1 до k ; op_i — элементарные операции.

Элементарные операции, составляющие команду, выполняются только в том случае, если все условия, означающие присутствие указанных прав доступа в ячейках матрицы M , являются истинными. В классической модели допустимы только следующие элементарные операции:

enter r into $M[s, o]$ (добавление субъекту s права r для объекта o)

delete r from $M[s, o]$ (удаление у субъекта s права r для объекта o)

create subject s (создание нового субъекта s)

create object o (создание нового объекта o)

destroy subject s (удаление существующего субъекта s)

destroy object o (удаление существующего объекта o)

Критерий безопасности модели Харрисона — Руззо — Ульмана формулируется следующим образом.

Для заданной системы начальное состояние $Q_0 = (S_0, O_0, M_0)$ является безопасным относительно права r , если не существует применимой к Q_0 последовательности команд, в результате которой право r будет занесено в ячейку матрицы M , в которой оно отсутствовало в состоянии Q_0 .

Нужно отметить, что все дискреционные модели уязвимы по отношению



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

к атаке с помощью «троянского коня», поскольку в них контролируются только



операции доступа субъектов к объектам, а не потоки информации между ними. Поэтому, когда «троянская» программа, которую нарушитель подсунил некоторому пользователю, переносит информацию из доступного этому объекта в объект, доступный нарушителю, то формально никакое правило дискреционной политики безопасности не нарушается, но утечка информации происходит.

Таким образом, дискреционная модель Харрисона—Руццо — Ульмана в своей общей постановке не дает гарантий безопасности системы, однако именно она послужила основой для целого класса моделей политик безопасности, которые используются для управления доступом и контроля за распределением прав во всех современных системах.

Типизованная матрица доступа. Другая дискреционная модель, получившая название «Типизованная матрица доступа» (*Type Access Matrix*— далее *TAM*), представляет собой развитие модели Харрисона—Руццо—Ульмана, дополненной концепцией типов, что позволяет несколько смягчить те условия, для которых возможно доказательство безопасности системы.

Состояние системы описывается четверкой $Q = (S, O, t, M)$, где S , O и M обозначают соответственно множество субъектов, объектов и матрицу доступа, а $t: O \rightarrow T$ — функция, ставящая в соответствие каждому объекту некоторый тип.

Состояние системы изменяется с помощью команд из множества C . Команды *TAM* имеют тот же формат, что и в модели Харрисона—Руццо—Ульмана, но всем параметрам приписывается определенный тип:

```
command a( $x_1: t_1, x_k: t_k$ )
  If  $r_1$  in  $M[x_{S_1}, x_{O_1}]$  and
     $r_1$  in  $M[x_{S_2}, x_{O_2}]$  and
    .
    .
     $r_m$  in  $M[x_{S_m}, x_{O_m}]$ 
  then
     $op_1, op_2, \dots, op_n$ 
```

(условия выполнения команды)

(операции, составляющие команду)



Перед выполнением команды происходит проверка типов фактических параметров, и, если они не совпадают с указанными в определении, команда не выполняется. Фактически введение контроля типов для параметров команд приводит к неявному введению дополнительных условий, так как команды могут быть выполнены только при совпадении типов параметров. В модели используются следующие шесть элементарных операций, отличающихся от аналогичных операций модели Харрисона—Руззо —Ульмана только использованием типизованных параметров:

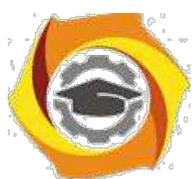
enter r into $M[s, o]$	}	(монотонные операции)
create subject s of type t		
create object o of type t		
delete r from $M[s, o]$	}	(немонотонные операции)
destroy subject s		
destroy object o		

Смысл элементарных операций совпадает со смыслом аналогичных операций их классической модели Харрисона — Руззо — Ульмана с точностью до использования типов.

Таким образом, *TAM* является обобщением модели Харрисона—Руззо—Ульмана, которую можно рассматривать как частный случай *TAM* с одним единственным типом, к которому относятся все объекты и субъекты. Появление в каждой команде дополнительных неявных условий, ограничивающих область применения команды только сущностями соответствующих типов, позволяет несколько смягчить жесткие условия классической модели, при которых критерий безопасности является разрешимым.

Несмотря на различающиеся подходы, суть всех моделей безопасности одинакова, поскольку они предназначены для решения одних и тех же задач. Целью построения модели является получение формального доказательства безопасности системы, а также определения достаточного критерия безопасности.

Безопасность системы определяется в равной степени тремя факторами:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

свойствами самой модели, ее адекватностью угрозам, воздействующим на



систему, и тем, насколько корректно она реализована. Поскольку при существующем разнообразии теоретических разработок в области теории информационной безопасности выбор модели, адекватной заданным угрозам, не является проблемой, последнее, решающее слово остается за ее реализацией в защищенной системе.

Иерархический метод построения защиты.

Создание гарантированно защищенных баз данных связано с некоторыми общими проблемами синтеза систем защиты. Если политика безопасности в базе данных не включает вопросов, связанных с взаимным выводом информации и каналов утечки, основанных на этом выводе, вопросов восстановления зашумленной информации путем повторных запросов в базу данных, то такая политика не является адекватной для безопасности информации. Однако, все эти вопросы нельзя включать в политику безопасности, поддерживаемую самой вычислительной системой, что приведет к симбиозу операционной системы и системы управления базой данных. Вместе с тем, сложилась такая практика, что производителями операционных систем и систем управления базами данных являются разные фирмы или организации, что делает такой симбиоз невозможным. Поэтому политика безопасности частично должна поддерживаться самой базой данных. Аналогичные проблемы возникают при модернизации защищенных систем и могут быть сформулированы как противоречие между единой оценкой защищенности всей системы и многопрофильностью подсистем, создаваемых различными производителями. С аналогичной проблемой мы сталкивались в тема 8. Однако теперь существенным образом возникает зависимость частей защиты друг от друга (ТСВ вычислительной системы управляет субъектами системы защиты (ТСВ), поддерживающей политику безопасности базы данных). Такая структура не вкладывается в систему независимых защищенных компонент распределенной сети. В параграфе 7.1 приводится описание модели ТСВ-подмножеств, иерархически связанных друг с другом, которые при выполнении некоторых дополнительных условий могут удовлетворять



условиям ТСВ системы. В п. 9.2 приведены наиболее распространенные архитектуры защищенных баз данных, в которых теория п.9.1 может быть реализована.

Иерархический метод построения защиты.

В данном п. рассматривается еще один пример иерархической декомпозиции сложных систем. Если в п. 2.2 мы рассматривали примеры, не связанные непосредственно с задачей защиты, то сейчас основное внимание будет посвящено иерархическому построению ТСВ в электронных системах обработки данных. Как и раньше, основная задача ТСВ - поддержка монитора обращений. Однако, в отличие от теории предыдущего параграфа, где мониторы обращения были независимыми, сейчас мы покажем, что одни ТСВ-подмножества могут использовать ресурсы в других ТСВ-подмножествах. А именно, рассмотрим следующую модель.

Определение. ТСВ-подмножество M есть совокупность программно-аппаратных ресурсов системы, которые управляют доступами множества S субъектов к множеству O объектов на основе четко определенной политики P и удовлетворяет свойствам:

- 1) M определяет и контролирует каждый доступ к объектам из O со стороны субъектов из S ;
- 2) M гарантировано защищено;
- 3) M достаточно просто устроено, чтобы существовала возможность проанализировать его системой тестов, полнота которых доказана.

Зависимость ТСВ-подмножеств состоит в том, что M использует ресурсы точно определенного множества более примитивных ТСВ-подмножеств (то есть предполагается заданным некоторый частичный порядок на ТСВ-подмножествах), для того, чтобы создать объекты из O , создать и поддерживать структуры данных и поддерживать политику P .

Если более примитивных ТСВ-подмножеств нет, то такое ТСВ-подмножество опирается в решении тех же задач на аппаратную часть.



Отметим, что кроме монитора обращений необходимо существование механизма поддержки этого монитора. Рассмотрим условия, при которых из ТСВ-подмножеств удастся собрать ТСВ системы (то есть доказать, что выполняются все требования к ТСВ).

Пусть даны ТСВ-подмножества $M(i)$, которые управляют доступом субъектов $S(i)$ к объектам $O(i)$ в соответствии с политикой безопасности $P(i)$. Предположим, что нет объектов в системе, которые контролируются более, чем одним ТСВ-подмножеством. Здесь принимается та же точка зрения, как в главе VI, состоящая в том, что ТСВ-подмножества могут экспортировать объекты, подлежащие управлению другим ТСВ-подмножеством. Однако принятый и переданный объекты - это разные объекты, контролируемые разными ТСВ-подмножествами. При этом политика безопасности $P(i)$ ТСВ-подмножества $M(i)$ может отличаться от политики безопасности $P(j)$ ТСВ-подмножества $M(j)$. Однако все вместе должны составлять единую политику P системы, причем каждое правило из P должно поддерживаться определенной системой ТСВ-подмножеств.

Необходимо также предполагать, что каждый доверенный субъект, то есть субъект, который может нарушать правила $P(i)$ является частью ТСВ-подмножества $M(i)$.

Рассмотрим ограничения на зависимости ТСВ-подмножеств.

Определение. ТСВ-подмножество A прямо зависит (в своей правильности) от ТСВ-подмножества B тогда и только тогда, когда доводы о подтверждении правильности A (верификация установки A обозначается vA) частично или полностью основаны на предположении, что B установлена верно в соответствии спецификацией B (обозначается sB).

Определение. ТСВ-подмножество A менее примитивно, чем ТСВ-подмножество B , если:

- а) A прямо зависит от B ;
- б) существует цепочка ТСВ-подмножеств от A к B такая, что каждый элемент цепи прямо зависит от предыдущего элемента цепи.



Рассмотрим примеры, поясняющие понятие зависимости, взятые из "Розовой книги".

Пример 1. Пусть ТСВ-подмножество B предоставляет услугу в виде "файла", которым B управляет в соответствии с политикой $P(B)$, а ТСВ-подмножество A использует B -файл для хранения информации. Если vA использует факт, что различные B -файлы действительно различаются и доступ к ним определяется политикой $P(B)$, то vA полагается на факт, что sB и B соответствуют друг другу. Тогда A прямо зависит от B .

Пример 2. Пусть A и B взаимно не доверяющие друг другу системы бронирования авиабилетов, расположенные отдельно и принадлежащие различным организациям. Предположим sA и sB дают возможность получить заказ на бронирование по сети от других систем, используя взаимно согласованный протокол. Пусть этот протокол полностью определен и соответствует sA и sB . Пусть также vA и vB с заданной степенью уверенности подтверждают, что A и B соответствуют своим спецификациям sA и sB . A не зависит в правильности своей установки от правильности установки B , так как sA - полная, то есть какая бы последовательность бит не пришла от B , sA определяет, что A должна делать, а vA демонстрирует, что именно это и делается. Аналогично, B не зависит от правильности A . Поэтому A и B не зависят одна от другой.

Пример 3. Пусть A является сервером электронной почты, а B управляет запросами в базу данных. Спецификация sA может совсем не упоминать систему управления базой данных, она просто определяет интерфейс почтовой системы. Однако в vA мы находим, что программа обеспечения A использует таблицы, предоставляемые B , для хранения сообщений A и B на различных, связанных машинах. Ни sB , ни vB не упоминают о почтовой системе. Как и в предыдущем примере, sB полностью определяет поведение B для всех получаемых последовательностей бит. Здесь A прямо зависит от B , но B не зависит от A . Отметим, что информационные потоки в обоих направлениях



являются законными и никоим образом не компрометируют целостность В. Зависимость находится в другой плоскости от потока данных.

Этот пример замечателен тем, что анализ структуры элементов не позволяет выявить существование зависимости, при этом архитектура системы внешне совпадает с приведенной в примере 2 архитектурой взаимно независимых систем. Кроме того, этот пример показывает, что описания интерфейса не достаточно для выявления зависимостей.

Пример 4. Пусть А и В взаимно зависимые системы. А зависит от В и В зависит от А. Это значит, что правильность установки vA доказывается из предположения, что В установлена правильно в соответствии с sB . Также правильность установки vB доказывается из предположения, что А установлена правильно в соответствии с sA . Пусть vA и vB подтверждают правильность установки А и В. Однако отсюда не следует, что А и В функционируют правильно.

В самом деле, если А и В функционируют правильно относительно sA и sB , то vA и vB поддерживают правильность установки. Если А установлено неправильно по отношению к sA и В установлено неправильно по отношению к sB , то никто не мешает возникновению ситуации, когда vA подтверждает правильность исходя из того, что В не соответствует sB . Аналогично, vB подтверждает правильность установки sB , хотя А не соответствует sA . Тогда можно доказать, что vA и vB подтверждают правильность А и В, хотя А и В установлены неверно.

Для того, чтобы понять возникающую здесь коллизию, рассмотрим две системы бронирования билетов на самолеты А и В, как в примере 2. Предположим, что А содержит информацию об исходных пунктах и времени вылета всех полетов в США, а В - в Европе. Пусть sA включает утверждение, что А обеспечивает пассажирам услугу в подборе вылета по транзиту, минимизирующем время ожидания следующего полета. Пусть sB предлагает аналогичные услуги. Из утверждения А соответствует sA и В соответствует sB можно вывести истинность утверждения, что А соответствует спецификации sA



в предоставлении услуги подбора транзитного рейса в Европе и Америке. Аналогичное утверждение можно вывести для В. Однако, если А хранит информацию в местном времени, а В - во времени по Гринвичу, то транзитный маршрут, составленный А и В на основе информации, полученной друг от друга, будет неправильным из-за различия во времени. То есть каждая система функционирует правильно, но результат неверен. Это происходит из-за того, что А и В зависимы.

Гарантии защищенности ТСВ-подмножеств имеют большое значение в проблеме построения единой ТСВ системы из ТСВ-подмножеств. В случае зависимых ТСВ-подмножеств менее примитивное ТСВ-подмножество может использовать объекты и субъекты, предоставленные более примитивным ТСВ-подмножеством. Потому первая проблема состоит в доказательстве того факта, что невозможны никакие изменения данных, критических для политики безопасности, или кодов ТСВ-подмножества. То есть никакая внешняя по отношению к ТСВ-подмножеству система (кроме, может быть, более примитивных ТСВ-подмножеств) не может инициировать произвольное изменение кодов ТСВ-подмножества или его структур данных. Вторая проблема состоит в доказательстве того, что данные, хранящиеся в ТСВ-подмножестве и критичные для политики безопасности, не могут изменяться иначе, чем в соответствии с логикой ТСВ-подмножества. Разумеется, эти доказательства возможны при условии правильности той информации, которая вносится в ТСВ-подмножество более примитивным ТСВ-подмножеством.

Ясно, что в доказательстве возможности построения единой ТСВ системы из ТСВ-подмножеств присутствуют условия не только на локальные свойства ТСВ-подмножеств, но также интегральные требования.

Суммируем перечень всех условий, которые необходимо выполнить, чтобы из семейства ТСВ-подмножеств можно было бы синтезировать ТСВ системы. Если, кроме нижеприведенных требований, выполняются требования какого-либо класса в стандарте "Оранжевая книга", то построенная таким образом система может быть аттестована по соответствующему классу.



Условия:

1. ТСВ-подмножества четко определены.
2. Политика безопасности системы распределена по ТСВ-подмножествам.
3. Каждое ТСВ-подмножество $M(i)$ включает доверенные процессы по отношению к своей политике безопасности $P(i)$.
4. Архитектура ТСВ-подмножеств четко определена .
5. Все ТСВ-подмножества занимают различные домены.
6. Во всех случаях примитивные ТСВ-подмножества поддерживают правильное функционирование монитора обращений в менее примитивном ТСВ-подмножестве.

Исследование корректности реализации и верификация ас

Понятие корректности или правильности подразумевает соответствие проверяемого объекта некоторому эталонному объекту или совокупности формализованных этапонных характеристик и правил. Корректность ПО при разработке наиболее полно определяется степенью соответствия предъявляемым к ней формализованным требованиям программной спецификации. В спецификациях отражается совокупность этапонных характеристик, свойств и условий, которым должна соответствовать программа. Основную часть спецификации составляют функциональные критерии и характеристики. Исходной программной спецификацией, которой должна соответствовать программа, является ТЗ.

При отсутствии полностью формализованной спецификации требований в качестве ТЗ, которому должна соответствовать АС и результаты ее функционирования, иногда используются неформализованные представления разработчика, пользователя или заказчика программ. Однако понятие корректности программ по отношению к запросам пользователя или заказчика сопряжено с неопределённостью самого эталона, которому должна соответствовать АС. Для сложных программ всегда существует риск обнаружить их некорректность (по мнению пользователя или заказчика) при



формальной корректности относительно спецификаций вследствие неточности самих спецификаций.

Традиционный взгляд на спецификацию требований заключается в том, что она представляет собой документ на естественном языке, который является интерфейсом между заказчиком и изготовителем. Хотя подготовке документа может предшествовать некоторое взаимодействие, именно этот документ в значительной степени выступает как "отправная точка" для изготовителя программ.

Таким образом, можно сделать вывод о том, что создание совокупности взаимоувязанных непротиворечивых спецификаций является необходимой базой для обеспечения корректности проектируемой программы. При этом спецификации должны:

быть формальными;

 позволять проверять непротиворечивость и полноту требований заказчика;

 служить основой для дальнейшего формализованного проектирования ОС.

Существует несколько подходов к определению спецификаций требований.

Спецификация как описание. Заказчик выдает спецификацию, чтобы изготовители могли снабдить его тем изделием, которое он желает, поэтому заказчик видит этот документ главным образом как описание системы, которую он желал бы иметь. В принципе, в описании должно быть указано, что должна и что не должна делать система. На практике обычно по умолчанию предполагается, что система должна делать то, что уточняется в спецификации, 3. не должна делать ничего более. В этом состоит главная проблема с описательной стороной спецификации. Предполагается, что заказчик всегда точно знает всё, что система должна и не должна делать. Более того, в дальнейшем предполагается, что заказчик полностью перенёс это знание в специфицированный документ.



Спецификация как предписание. Изготовитель смотрит на специфицированный документ как на набор составных частей, подлежащих сборке, чтобы разрешить проблему заказчика. Такой предписывающий взгляд обуславливается не только трудностями создания описательного документа (как указывалось выше), но и сведениями, которые умышленно или неумышленно расширяют или ограничивают свободу изготовителя.

Договорная методология. В рамках "описание заказчика - предписание изготовителю" спецификация рассматривается как формальное разделение между сторонами. Что касается заказчика, то он оговаривает минимально приемлемое, тогда как изготовитель - максимально требуемое. Договор предлагается и принимается при зарождении системы и заканчивается после завершения системы, когда заказчик принимает систему как отвечающую его минимальным требованиям. Во время изготовления системы в принципе не предполагается никаких взаимодействий, даже если изготовитель подозревает, что предписываемое не совсем соответствует тому, что заказчик желает видеть в действительности.

Спецификация как модель. Современные более строгие представления о спецификации трактуют ее как модель системы. При условии, что лежащая в основе модели семантика в достаточной мере обоснована, такая спецификация обеспечивает чёткую формулировку требований.

Соответствующие модели подходят также для автоматизированного контроля целостности и другого прогнозного анализа, который, в частности, обеспечит прекращение разработки системы, в принципе не способной удовлетворить требованиям.

Модели как описание системы имеют следующие отличительные черты по сравнению с другими способами формального описания:

- хорошее сочетание нисходящего и восходящего подходов к их разработке с возможностью выбора абстрактного описания;
- возможность описания параллельной, распределенной и циклической работы;



возможность выбора различных формализованных аппаратов для описания систем.

Основное преимущество использования формальной модели заключается в возможности исследования с ее помощью особенностей моделируемой системы. Основываясь на формальном методе разработки на математической модели и затем исследуя модель, можно выявить такие грани поведения системы, которые в противном случае не были бы очевидны до более поздних стадий. Так как целевым объектом проектирования является АС, то модель может описывать либо саму АС, либо ее поведение, т.е. внешние проявления функционирования АС. Модель, описывающая поведение АС по сравнению с моделью АС, обладает одним важным преимуществом-она может быть проверена и оценена как исполнителями, так и заказчиками, поскольку заказчики не знают, как должна работать АС, но зато они представляют, что она должна делать. В результате такого моделирования может быть проверена корректность спецификаций относительно исходной постановки задачи, т.е. ТЗ. Кроме того, критерии правильности считаются достаточными при условии, что спецификация представляет собой исчерпывающее описание "внешнего" поведения объекта при всех возможных (или запланированных) ситуациях его использования.

Как было отмечено выше, при разработке АС, особенно ее компонентов, представляющих систему защиты информации, для обеспечения высоких гарантий отсутствия неисправностей и последующего доказательства того, что система функционирует согласно требованиям ТЗ, используются формальные подходы к ее проектированию.

Формальное проектирование алгоритмов базируется, в основном, на языках алгоритмических логик, которые включают высказывание вида

$Q\{S\}R$.

читающееся следующим образом: "если до исполнения оператора S было выполнено условие Q, то после него будет R". Здесь Q называется предусловием, а R-постусловием. Эти языки были изобретены практически



одновременно Р. У. Флойдом (1967 г.), С. А. Р. Хоаром (1969 г.) и учеными польской логической школы (А. Сальвицкий и др., 1970 г.). Как предусловие, так и постусловие являются предикатами.

Рассмотрение программ в качестве некоего "преобразователя предикатов" позволяет прямо определить связь между начальными и конечными состояниями без каких-либо ссылок на промежуточные состояния, которые могут возникнуть во время выполнения программы.

Преимущество представления алгоритма в виде преобразователя предикатов состоит в том, что оно дает возможность:

- анализировать алгоритмы как математические объекты;
- дать формальное описание алгоритма, позволяющее интеллектуально охватить алгоритм;
- синтезировать алгоритмы по представленным спецификациям;
- провести формальное верифицирование алгоритма, т.е. доказать корректность его реализации.

Методология формальной разработки и доказательства корректности алгоритмов в настоящее время хорошо разработана и изложена в целом ряде работ. Вкратце суть этих методов сводится к следующему:

- разработка алгоритма проводится методом последовательной декомпозиции, с разбивкой общей задачи, решаемой алгоритмом, на ряд более мелких подзадач;
- критерием детализации подзадач является возможность их реализации с помощью одной конструкции ветвления или цикла;
- разбиение общей задачи на подзадачи предусматривает формулирование пред- и постусловий для каждой подзадачи с целью их корректного проектирования и дальнейшей верификации.

Для доказательства корректности алгоритма (верификация) формулируется математическая теорема $Q\{S\}R$, которая затем доказывается. Доказательство теоремы о корректности принято разбивать на две части. Одна часть служит для доказательства того, что рассматриваемый алгоритм вообще



может завершить работу (проводится анализ всех циклов). В другой части доказывается корректность постуловия в предположении, что алгоритм завершает работу.

Методология обследования и проектирования систем защиты при организации информационной безопасности

1 Теоретическая информация (тезисное изложение материала примерно на 20 страниц)

Требования нормативных документов к средствам защиты информации, проведение обследования и испытаний средств защиты информации.

2 Практический пример использования теоретического материала (примерно на 5 страниц)

3 Электронные слайды в MicrosoftOfficePowerPoint2003 по теоретическому материалу в виде рисунков

4 Глоссарий по теоретическому материалу (примерно на 5 страниц)

5 Список используемых информационных источников

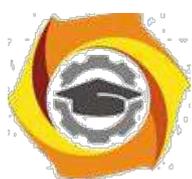
Управление процессами функционирования систем защиты при организации информационной безопасности

1 Теоретическая информация (тезисное изложение материала примерно на 20 страниц)

Определение процесса, функции системы защиты, администрирование системы защиты.

2 Практический пример использования теоретического материала (примерно на 5 страниц)

3 Электронные слайды в MicrosoftOfficePowerPoint2003 по теоретическому материалу в виде рисунков



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

4 Глоссарий по теоретическому материалу (примерно на 5 страниц)

5 Список используемых информационных источников



**Перечень примерной тематики практических работ по
теоретическому материалу**

1. Виды моделей безопасности. Модель дискреционного разграничения доступа
 - Виды моделей безопасности. Модель мандатного разграничения доступа.
 - Виды моделей безопасности. Модель безопасности информационных потоков.
 - Виды моделей безопасности. Модель ролевого разграничения доступа
 - Политика безопасности. Модель Белла-ЛаПадула и Low-Water-Mark
 - Политика безопасности. Модель БИБА
 - Примеры практической реализации. Программно-аппаратная реализация системы безопасности компьютерной информации. SecretNet,
 - Примеры практической реализации. Аппаратная реализация системы безопасности компьютерной информации. Ключи HASP.
 - Примеры практической реализации. Программная реализация системы безопасности компьютерной информации. Антивирусная защита, установка и настройка пакета Shadow Defender. Ревизор XP.
 - В Управление пользователями и правами в ОС Windows 2000,
 - и Построение парольных систем. Генераторы паролей. Оценка стойкости парольной защиты.
 - и Построение парольных систем. Защита документов в Microsoft Office. Защита архивов.
 - и Методы криптографической защиты. Принципы криптографической защиты информации. Традиционные симметричные криптосистемы.
 - и Методы криптографической защиты. Элементы криптоанализа.



- Методы криптографической защиты. Особенности реализации систем с несимметричными ключами, несимметричные криптосистемы шифрования, несимметричные алгоритмы.
- Модель политики контроля целостности. Функция хеширования, ЭЦП.

Код 5 (опд.Ф.11 рп2009 Организационное обеспечение информационной безопасности)

Анализ и оценка угроз информационной безопасности объекта

1 Теоретическая информация (тезисное изложение материала примерно на 20 страниц)

Оценка возможностей технических разведок и других источников угроз безопасности конфиденциальной информации. Риск-Менеджмент. Методы оценки уязвимости информации. Методы оценки достоверности информационной базы моделей прогнозирования значений показателей уязвимости информации. Методы оценки ущерба от реализации угроз ИБ. Эмпирический подход к оценке уязвимости информации. Практическая реализация модели «Угроза-Защита». Методы определения требований кЗИ

2 Практический пример использования теоретического материала (примерно на 5 страниц)

3 Электронные слайды в MicrosoftOfficePowerPoint2003 по теоретическому материалу в виде рисунков

4 Глоссарий по теоретическому материалу (примерно на 5 страниц)

5 Список используемых информационных источников

Модели контроля целостности

Рассмотрим модели безопасности, контролирующие целостность информации. В частности, модели Биба, использующиеся для синтеза механизмов контроля целостности информации в системе, а также модель Кларка – Вилсона (КВМ), которая является примером неформального



выражения политики безопасности. Последняя модель сформулирована в виде набора неформальных правил, и хотя в литературе она названа моделью безопасности, ее скорее можно назвать политикой контроля целостности.

Модель Биба

Кен Биба в середине семидесятых годов прошлого века сделал два наблюдения. Они были последовательно внесены в модель безопасности, которая с тех пор называется моделью целостности Биба (или просто моделью Биба). В контексте разговора о моделях контроля целостности запись наверх может представлять угрозу в том случае, если субъект с низким уровнем безопасности искажает или уничтожает данные в объекте, лежащем на более высоком уровне. Поэтому, исходя из задач целостности, можно потребовать, чтобы такая запись была запрещена. Кроме того, можно рассматривать чтение снизу как поток информации, идущий из объекта нижнего уровня и нарушающий целостность субъекта высокого уровня. Поэтому весьма вероятно, что и такое чтение необходимо запретить.

Биба выразил свою модель таким же способом, каким была выражена БЛМ, за тем исключением, что правила его модели являются полной противоположностью правилам БЛМ. Возможны три вариации модели Биба: мандатная модель целостности, модель понижения уровня субъекта и модель понижения уровня объекта. Фактически, общий термин «модель Биба» используется для обозначения любой или сразу всех трех моделей.

Мандатная модель целостности Биба

Ее часто называют инверсией БЛМ. Это довольно точное название, поскольку основные правила этой модели просто переворачивают правила БЛМ. Мы будем ссылаться на эти правила как «нет чтения снизу» (NRD) и «нет записи наверх» (NWU) и определим их в терминах субъектов, объектов и нового типа уровней безопасности – уровней целостности, над которыми может быть введено отношение преобладания.

Правило NRD мандатной модели целостности Биба определяется как запрет субъектам на чтение информации из объекта с более низким уровнем



целостности. Правило NWU мандатной модели целостности Биба определяется как запрет субъектам на запись информации в объект с более высоким уровнем целостности.

Одним из преимуществ этой модели является то, что она унаследовала многие важные характеристики БЛМ, включая ее простоту и интуитивность. Это значит, что проектировщики реальных систем могут легко понять суть этих правил и использовать их для принятия решений при проектировании. Кроме того, поскольку мандатная модель целостности Биба, подобно БЛМ, основана на простой иерархии, ее легко объяснить и изобразить пользователям системы.

- другой стороны, модель представляет собой очевидное противоречие с правилами NRU и NWD. Это значит, что если необходимо построить систему, которая предотвращает угрозы как секретности, так и целостности, то одновременное использование правил моделей БЛМ и Биба может привести к ситуации, в которой уровни безопасности и целостности будут использоваться противоположными способами.

Рассмотрим формальное описание модели Биба. Для этого опишем простые математические конструкции, которые помогут описать различные правила, составляющие мандатную модель целостности Биба.

Начнем с представления множества субъектов и объектов. Уровни целостности субъекта или объекта x обозначаются как уровень (x), и для них введено отношение преобладания. Используя эти определения, сформулируем правила NRD и NWU мандатной модели целостности Биба в терминах булевой функции разрешить:

NRD: $s, o \in O$: разрешить (s, o , чтение) уровень (o) > уровень (s).

Данный тип определения предусматривает условия, при которых функция разрешить принимает значение истинно. Определение утверждает, что для всех определенных субъектов и объектов операция чтения разрешена только в том случае, если выполняется условие преобладания. Правило NWU просто переворачивает использование отношения преобладания, как показано в следующем определении:



Практическое занятие №3. Основные этапы построения системы комплексной защиты вычислительных систем; анализ моделей нарушителя; угрозы информационно-программному обеспечению вычислительных систем и их классификация.

Цель работы: ознакомиться с основными этапами построения системы комплексной защиты вычислительных систем.

Система защиты информации должна создаваться совместно с создаваемой компьютерной системой. При построении системы защиты могут использоваться существующие средства защиты, или же они разрабатываются специально для конкретной АС. В зависимости от особенностей компьютерной системы, условий ее эксплуатации и требований к защите информации процесс создания КСЗИ может не содержать отдельных этапов, или содержание их может несколько отличаться от общепринятых норм при разработке сложных аппаратно-программных систем. Но обычно разработка таких систем включает следующие этапы:

- разработка технического задания;
- эскизное проектирование;
- техническое проектирование;
- рабочее проектирование;
- производство опытного образца.

Одним из **основных** этапов разработки КСЗИ является этап разработки технического задания. Именно на этом этапе решаются практически все специфические задачи, характерные именно для разработки КСЗИ.

Процесс разработки систем, заканчивающийся выработкой **технического задания**, называют научно-исследовательской разработкой, а остальную часть работы по созданию сложной системы называют **опытно-конструкторской разработкой**. Опытно-конструкторская разработка аппаратно-программных средств ведётся с применением систем автоматизации проектирования,



алгоритмы проектирования хорошо изучены и отработаны. Поэтому особый интерес представляет рассмотрение процесса научно-исследовательского проектирования.

Научно-исследовательская разработка ксзи

Целью этого этапа является разработка **технического задания** на проектирование КСЗИ. Техническое задание содержит основные технические требования к разрабатываемой КСЗИ, а также согласованные взаимные обязательства заказчика и исполнителя разработки. Технические требования определяют значения основных технических характеристик, выполняемые функции, режимы работы, взаимодействие с внешними системами и т. д.

Аппаратные средства оцениваются следующими характеристиками: быстродействие, производительность, ёмкость запоминающих устройств, разрядность, стоимость, характеристики надёжности и др. Программные средства характеризуются требуемым объёмом оперативной и внешней памяти, системой программирования, в которой разработаны эти средства, совместимостью с ОС и другими программными средствами, временем выполнения, стоимостью и т. д.

Получение значений этих характеристик, а также состава выполняемых функций и режимов работы средств защиты, порядка их использования и взаимодействия с внешними системами составляют основное содержание этапа научно-исследовательской разработки. Для проведения исследований на этом этапе заказчик может привлекать исполнителя или научно-исследовательское учреждение, либо организует совместную их работу.

Научно-исследовательская разработка начинается с анализа угроз безопасности информации, анализа защищаемой АС и анализа конфиденциальности и важности информации в АС.

Сначала производится анализ конфиденциальности и важности информации, которая должна обрабатываться, храниться и передаваться в АС. На основе анализа делается вывод о целесообразности создания КСЗИ. Если информация не является конфиденциальной и легко может быть восстановлена,



то создавать СЗИ нет необходимости. Не имеет смысла также создавать КСЗИ в АС, если потеря целостности и конфиденциальности информации связана с незначительными потерями.

В этих случаях достаточно использовать штатные средства АС и, возможно, страхование от утраты информации.

При анализе информации определяются потоки конфиденциальной информации, элементы АС, в которых она обрабатывается и хранится. На этом этапе рассматриваются также вопросы разграничения доступа к информации отдельных пользователей и целых сегментов АС. На основе анализа информации определяются требования к её защищённости. Требования задаются путём присвоения определённого грифа конфиденциальности, установления правил разграничения доступа.

Очень важная исходная информация для построения КСЗИ получается в результате анализа защищаемой АС. Так как КСЗИ является подсистемой АС, то взаимодействие системы защиты с АС можно определить как внутреннее, а взаимодействие с внешней средой - как внешнее (рис. 1).

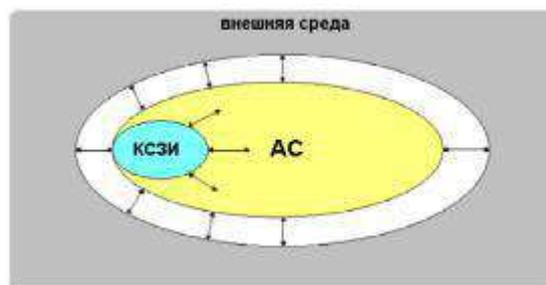


Рис. 1. Схема взаимодействия КСЗИ

Внутренние условия взаимодействия определяются архитектурой АС.

При построении КСЗИ учитываются:

географическое положение АС;

тип АС (распределённая или сосредоточенная);

структуры АС (техническая, программная, информационная и т. д.);

производительность и надёжность элементов АС;

типы используемых аппаратных и программных средств и режимы их работы;



угрозы безопасности информации, которые порождаются внутри АС (отказы аппаратных и программных средств, алгоритмические ошибки и т. п.).

Учитываются следующие внешние условия:

взаимодействие с внешними системами;

случайные и преднамеренные угрозы.

Анализ угроз безопасности является одним из обязательных условий построения КСЗИ. По результатам проведённого анализа строится модель угроз безопасности информации в АС.

Модель угроз безопасности информации в АС содержит систематизированные данные о случайных и преднамеренных угрозах безопасности информации в конкретной АС. Систематизация данных модели предполагает наличие сведений обо всех возможных угрозах, их опасности, временных рамках действия, вероятности реализации. Часто модель угроз рассматривается как композиция модели злоумышленника и модели случайных угроз.

Модели представляются в виде таблиц, графов или на вербальном уровне. При построении модели злоумышленника используются два подхода:

- модель ориентируется только на высококвалифицированного злоумышленника-профессионала, оснащённого всем необходимым и имеющего легальный доступ на всех рубежах защиты;
- модель учитывает квалификацию злоумышленника, его оснащённость (возможности) и официальный статус в АС.

Первый подход проще реализуется и позволяет определить верхнюю границу преднамеренных угроз безопасности информации.

Второй подход отличается гибкостью и позволяет учитывать особенности АС в полной мере. Градация злоумышленников по их квалификации может быть различной. Например, может быть выделено три класса злоумышленников:

- С высококвалифицированный злоумышленник-профессионал;



- в квалифицированный злоумышленник-непрофессионал;
- в неквалифицированный злоумышленник-непрофессионал.

Класс злоумышленника, его оснащённость и статус на объекте определяют возможности злоумышленника по несанкционированному доступу к ресурсам АС.

Угрозы, связанные с непреднамеренными действиями, хорошо изучены, и большая часть их может быть формализована. Сюда следует отнести угрозы безопасности, которые связаны с конечной надёжностью технических систем. Угрозы, порождаемые стихией или человеком, формализовать сложнее. Но с другой стороны, по ним накоплен большой объем статистических данных. На основании этих данных можно прогнозировать проявление угроз этого класса.

Модель злоумышленника и модель случайных угроз позволяют получить полный спектр угроз и их характеристик. В совокупности с исходными данными, полученными в результате анализа информации, особенностей архитектуры проектируемой АС, модели угроз безопасности информации позволяют получить исходные данные для построения модели КСЗИ.

Классификация угроз информационной безопасности

Угрозы можно классифицировать по нескольким критериям:

по аспекту информационной безопасности (доступность, целостность, конфиденциальность), против которого угрозы направлены в первую очередь;

по компонентам информационных систем, на которые угрозы нацелены (данные, программы, аппаратура, поддерживающая инфраструктура);

по способу осуществления (случайные/преднамеренные действия природного/техногенного характера);

по расположению источника угроз (внутри/вне рассматриваемой ИС).

Необходимость классификации угроз ИБ АС обусловлена тем, что архитектура современных средств автоматизированной обработки информации, организационное, структурное и функциональное построение информационно-



вычислительных систем и сетей, технологии и условия автоматизированной обработки информации такие, что накапливаемая, хранимая и обрабатываемая информация подвержена случайным влияниям чрезвычайно большого числа факторов, в силу чего становится невозможным формализовать задачу описания полного множества угроз. Как следствие, для защищаемой системы определяют не полный перечень угроз, а перечень классов угроз.

Классификация всех возможных угроз информационной безопасности АС может быть проведена по ряду базовых признаков.

4. По природе возникновения.

1.1. Естественные угрозы-угрозы, вызванные воздействиями на АС и ее компоненты объективных физических процессов или стихийных природных явлений, независимых от человека.

1.2. Искусственные угрозы- угрозы информационной безопасности АС, вызванные деятельностью человека.

По степени преднамеренности проявления.

2.1. Угрозы случайного действия и/или угрозы, вызванные ошибками или халатностью персонала. Например:

проявление ошибок программно-аппаратных средств АС;

некомпетентное использование, настройка или неправомерное отключение средств защиты персоналом службы безопасности;

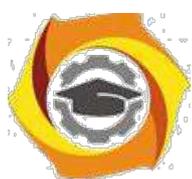
неумышленные действия, приводящие к частичному или полному отказу системы или разрушению аппаратных, программных, информационных ресурсов системы (неумышленная порча оборудования, удаление, искажение файлов с важной информацией или программ, в том числе системных и т. п.);

неправомерное включение оборудования или изменение режимов работы устройств и программ;

неумышленная порча носителей информации;

пересылка данных по ошибочному адресу абонента (устройства);

ввод ошибочных данных;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

неумышленное повреждение каналов связи.



2.2. Угрозы преднамеренного действия (например, угрозы действий злоумышленника для хищения информации).

По непосредственному источнику угроз.

3.1. Угрозы, непосредственным источником которых является природная среда (стихийные бедствия, магнитные бури, радиоактивное излучение и т.п.).

3.2. Угрозы, источником которых является человек:

внедрение агентов в число персонала системы (в том числе, возможно, и в административную группу, отвечающую за безопасность);

вербовка (путем подкупа, шантажа и т.п.) персонала или отдельных пользователей, имеющих определенные полномочия;

угроза несанкционированного копирования секретных данных пользователем АС;

разглашение, передача или утрата атрибутов разграничения доступа (паролей, ключей шифрования, идентификационных карточек, пропусков и т.п.).

3.3. Угрозы, непосредственным источником которых являются санкционированные программно-аппаратные средства:

запуск технологических программ, способных при некомпетентном пользовании вызывать потерю работоспособности системы (зависания) или зацикливания) или необратимые изменения в системе (форматирование или реструктуризацию носителей информации, удаление данных и т. п.);

возникновение отказа в работе операционной системы.

3.4. Угрозы, непосредственным источником которых являются несанкционированные программно-аппаратные средства:

нелегальное внедрение и использование неучтенных программ (игровых, обучающих, технологических и др., не являющихся необходимыми для выполнения нарушителем своих служебных обязанностей) с последующим необоснованным расходом ресурсов (загрузка процессора, захват оперативной памяти и памяти на внешних носителях);



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

заражение компьютера вирусами с деструктивными функциями.



2) По положению источника угроз.

4.1. Угрозы, источник которых расположен вне контролируемой зоны территории (помещения), на которой находится АС:

перехват побочных электромагнитных, акустических и других излучений устройств и линий связи, а также наводок активных излучений на вспомогательные технические средства, непосредственно не участвующие в обработке информации (телефонные линии, сети питания, отопления и т. п.);

перехват данных, передаваемых по каналам связи, и их анализ с целью выяснения протоколов обмена, правил вхождения в связь и авторизации пользователя и последующих попыток их имитации для проникновения в систему;

дистанционная фото- и видеосъемка.

4.2. Угрозы, источник которых расположен в пределах контролируемой зоны территории (помещения), на которой находится АС:

хищение производственных отходов (распечаток, записей, списанных носителей информации и т.п.);

отключение или вывод из строя подсистем обеспечения функционирования вычислительных систем (электропитания, охлаждения и вентиляции, линий связи и т.д.);

применение подслушивающих устройств.

4.3. Угрозы, источник которых имеет доступ к периферийным устройства АС(терминалам).

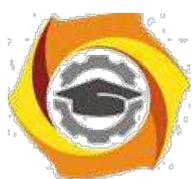
4.4. Угрозы, источник которых расположен в АС:

проектирование архитектуры системы и технологии обработки данных, разработка прикладных программ, которые представляют опасность для работоспособности системы и безопасности информации;

некорректное использование ресурсов АС.

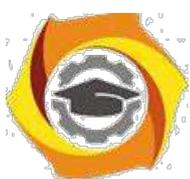
4. По степени зависимости от активности АС.

5.1. Угрозы, которые могут проявляться независимо от активности АС:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

вскрытие шифров криптозащиты информации;



хищение носителей информации (магнитных дисков, лент, микросхем памяти, запоминающих устройств и компьютерных систем).

5.2. Угрозы, которые могут проявляться только в процессе автоматизированной обработки данных (например, угрозы выполнения и распространения программных вирусов).

По степени воздействия на АС.

6.1. Пассивные угрозы, которые при реализации ничего не меняют в структуре и содержании АС (угроза копирования секретных данных).

6.2. Активные угрозы, которые при воздействии вносят изменения в структуру и содержание АС:

внедрение аппаратных спецвложений, программных "закладок" и "вирусов" ("троянских коней" и "жучков"), т.е. таких участков программ, которые не нужны для выполнения заявленных функций, но позволяют преодолеть систему защиты, скрытно и незаконно осуществить доступ к системным ресурсам с целью регистрации и передачи критической информации или дезорганизации функционирования системы;

действия по дезорганизации функционирования системы (изменение режимов работы устройств или программ, забастовка, саботаж персонала, постановка мощных активных радиопомех на частотах работы устройств системы и т.п.);

угроза умышленной модификации информации.

С По этапам доступа пользователей или программ к ресурсам АС.

7.1. Угрозы, которые могут проявляться на этапе доступа к ресурсам АС (например, угрозы несанкционированного доступа в АС).

7.2. Угрозы, которые могут проявляться после разрешения доступа к ресурсам АС (например, угрозы несанкционированного или некорректного использования ресурсов АС).

2. По способу доступа к ресурсам АС.

8.1. Угрозы, направленные на использование прямого стандартного пути доступа к ресурсам АС:



незаконное получение паролей и других реквизитов разграничения доступа (агентурным путем, используя халатность пользователей, подбором, имитацией интерфейса системы и т.д.) с последующей маскировкой под зарегистрированного пользователя ("маскарад");

несанкционированное использование терминалов пользователей, имеющих уникальные физические характеристики, такие как номер рабочей станции в сети, физический адрес, адрес в системе связи, аппаратный блок кодирования и т.п.

8.2. Угрозы, направленные на использование скрытого нестандартного пути доступа к ресурсам АС:

вход в систему в обход средств защиты (загрузка посторонней операционной системы со сменных магнитных носителей и т.п.);

угроза несанкционированного доступа к ресурсам АС путем использования недокументированных возможностей ОС.

6. По текущему месту расположения информации, хранимой и обрабатываемой в АС.

8. Угрозы доступа к информации на внешних запоминающих устройствах (например, угроза несанкционированного копирования секретной информации с жесткого диска).

9.2. Угрозы доступа к информации в оперативной памяти:

чтение остаточной информации из оперативной памяти;

чтение информации из областей оперативной памяти,

используемых операционной системой (в том числе подсистемой защиты) или другими пользователями, в асинхронном режиме, используя недостатки мультизадачных АС и систем программирования;

угроза доступа к системной области оперативной памяти со сторонних прикладных программ.

9.3. Угрозы доступа к информации, циркулирующей в линиях связи:



незаконное подключение к линиям связи с целью работы во время пауз в действиях законного пользователя от его имени с вводом ложных сообщений или модификацией передаваемых сообщений;

незаконное подключение к линиям связи с целью прямой подмены законного пользователя путем его физического отключения после входа в систему и успешной аутентификации с последующим вводом дезинформации и навязыванием ложных сообщений;

перехват всего потока данных с целью дальнейшего анализа не в реальном масштабе времени.

9.4. Угрозы доступа к информации, отображаемой на терминале или печатаемой на принтере (например, угроза записи отображаемой информации на скрытую видеокамеру). Вне зависимости от конкретных видов угроз или их проблемно-ориентированной классификации АС удовлетворяет потребности эксплуатирующих ее лиц, если обеспечиваются следующие свойства информации систем ее обработки.

4. качестве основного критерия будем использовать первый (по аспекту ИБ), привлекая при необходимости остальные.

Угроза доступности (отказа служб) возникает всякий раз, когда в результате преднамеренных действий, предпринимаемых другим пользователем или злоумышленником, блокируется доступ к некоторому ресурсу вычислительной системы. Реально блокирование может быть постоянным - запрашиваемый ресурс никогда не будет получен, или оно может вызывать только задержку запрашиваемого ресурса, достаточно долгую для того, чтобы он стал бесполезным. В этих случаях говорят, что ресурс исчерпан.

Доступность информации – свойство системы (среды, средств и технологии обработки), в которой циркулирует информация, характеризующееся способностью обеспечивать своевременный беспрепятственный доступ субъектов к интересующей их информации и готовность соответствующих автоматизированных служб к обслуживанию



поступающих от субъектов запросов всегда, когда возникает в этом необходимость.

Самыми частыми и самыми опасными (с точки зрения размера ущерба) являются непреднамеренные ошибки штатных пользователей, операторов, системных администраторов и других лиц, обслуживающих ИС.

Иногда такие ошибки и являются собственно угрозами (неправильно введенные данные или ошибка в программе, вызвавшая крах системы), иногда они создают уязвимые места, которыми могут воспользоваться злоумышленники. По некоторым данным, до 65% потерь – следствие непреднамеренных ошибок.

Пожары и наводнения не приносят столько бед, сколько безграмотность и небрежность в работе. Самый радикальный способ борьбы с непреднамеренными ошибками – максимальная автоматизация и строгий контроль.

Другие угрозы доступности классифицируем по компонентам ИС, на которые нацелены угрозы:

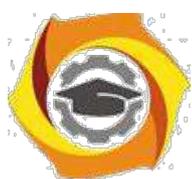
- отказ пользователей;
- внутренний отказ информационной системы;
- отказ поддерживающей инфраструктуры.

Обычно применительно к пользователям рассматриваются следующие угрозы:

- нежелание работать с информационной системой;

невозможность работать с системой в силу отсутствия соответствующей подготовки (недостаток компьютерной грамотности, неумение интерпретировать диагностические сообщения, неумение работать с документацией);

невозможность работать с системой в силу отсутствия технической поддержки (неполнота документации, недостаток справочной информации и т.п.).



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Основными источниками внутренних отказов являются:



отступление от установленных правил эксплуатации;

выход системы из штатного режима эксплуатации в силу случайных или преднамеренных действий пользователей или обслуживающего персонала (превышение расчетного числа запросов, чрезмерный объем обрабатываемой информации и т.п.);

ошибки при (пере)конфигурировании системы;

отказы программного и аппаратного обеспечения;

разрушение данных;

разрушение или повреждение аппаратуры.

По отношению к поддерживающей инфраструктуре рекомендуется рассматривать следующие угрозы:

нарушение работы (случайное или умышленное) систем связи, электропитания, водо- и/или теплоснабжения, кондиционирования;

разрушение или повреждение помещений;

невозможность или нежелание обслуживающего персонала и/или пользователей выполнять свои обязанности (гражданские беспорядки, аварии на транспорте, террористический акт или его угроза, забастовка и т.п.).

Весьма опасны так называемые "обиженные" сотрудники – нынешние и бывшие (они стремятся нанести вред организации-"обидчику", например: испортить оборудование; встроить логическую бомбу, которая со временем разрушит программы и/или данные; удалить данные). Необходимо следить за тем, чтобы при увольнении сотрудника его права доступа (логического и физического) к информационным ресурсам аннулировались.

Опасны, разумеется, стихийные бедствия и события, воспринимаемые как стихийные бедствия, – пожары, наводнения, землетрясения, ураганы, по статистике, (среди которых самый опасный – перебой электропитания) приходится 13% потерь, нанесенных ИС.

Некоторые примеры угроз доступности.

Угрозы доступности могут выглядеть грубо – как повреждение или даже разрушение оборудования (в том числе носителей данных) и может вызываться



естественными причинами (чаще всего – грозами). Опасны протечки водопровода и отопительной системы, в сильную жару, ломаются кондиционеры, установленные в серверных залах, набитых дорогостоящим оборудованием.

Общеизвестно, что периодически необходимо производить резервное копирование данных. Однако даже если это предложение выполняется, резервные носители обычно хранят небрежно, не обеспечивая их защиту от вредного воздействия окружающей среды.

Перейдем теперь к программным атакам на доступность.

качестве средства вывода системы из штатного режима эксплуатации может использоваться агрессивное потребление ресурсов (обычно – полосы пропускания сетей, вычислительных возможностей процессоров или ОЗУ). По расположению источника угрозы такое потребление подразделяется на локальное и удаленное. При просчетах в конфигурации системы локальная программа способна практически монополизировать процессор и/или физическую память, сведя скорость выполнения других программ к нулю.

Простейший пример удаленного потребления ресурсов – атака, получившая наименование "SYN-наводнение". Она представляет собой попытку переполнить таблицу "полуоткрытых" TCP-соединений сервера (установление соединений начинается, но не заканчивается), что приводит к затруднению установление новых соединений пользователей, то есть сервер блокируется.

По отношению к атаке "Papa Smurf" уязвимы сети, воспринимающие ping-пакеты с широковещательными адресами. Ответы на такие пакеты "съедают" полосу пропускания.

Удаленное потребление ресурсов в последнее время проявляется в особенно опасной форме – как скоординированные распределенные атаки, когда на сервер с множества разных адресов с максимальной скоростью направляются вполне легальные запросы на соединение и/или обслуживание.



Для выведения систем из штатного режима эксплуатации могут использоваться уязвимые места в виде программных и аппаратных ошибок. Например, известная ошибка в процессоре Pentium I дает возможность локальному пользователю путем выполнения определенной команды "подвесить" компьютер, так что помогает только аппаратный RESET.

Программа "Teardrop" удаленно "подвешивает" компьютеры, эксплуатируя ошибку в сборке фрагментированных IP-пакетов.

Угроза нарушения целостности включает в себя любое умышленное изменение информации, хранящейся в вычислительной системе или передаваемой из одной системы в другую, в том числе и несанкционированное изменение информации при случайных ошибках программного или аппаратного обеспечения. Санкционированными изменениями являются те, которые сделаны уполномоченными лицами с обоснованной целью (например, периодическая запланированная коррекция некоторой базы данных).

Целостность информации - существование информации в неискаженном виде (неизменном по отношению к некоторому фиксированному ее состоянию). Обычно субъектов интересует обеспечение более широкого свойства – достоверности информации, которое складывается из адекватности (полноты и точности) отображения состояния предметной области и непосредственно целостности информации, т.е. ее неискаженности.

На втором месте по размерам ущерба стоят кражи и подлоги. По данным газеты USA Today, еще в 1992 году в результате подобных противоправных действий с использованием персональных компьютеров американским организациям был нанесен общий ущерб в размере 882 миллионов долларов. В наши дни ущерб от такого рода действий вырос многократно.

К большинству случаев виновниками оказывались штатные сотрудники организаций, знакомые с режимом работы и мерами защиты, что подтверждает опасность внутренних угроз, хотя им уделяют меньшее внимание, чем внешним.



Существует различие между статической и динамической целостностью. В целях нарушения статической целостности злоумышленник может: ввести неверные данные; изменить данные.

Иногда изменяются содержательные данные, иногда – служебная информация. Показательный случай нарушения целостности имел место в 1996 году. Служащая Oracle (личный секретарь вице-президента) предъявила судебный иск, обвиняя президента корпорации в незаконном увольнении после того, как она отвергла его ухаживания. В доказательство своей правоты женщина привела электронное письмо, якобы отправленное ее начальником президенту. Содержание письма для нас сейчас не важно; важно время отправки. Дело в том, что вице-президент предъявил, в свою очередь, файл с регистрационной информацией компании сотовой связи, из которого явствовало, что в указанное время он разговаривал по мобильному телефону, находясь вдалеке от своего рабочего места. Таким образом, в суде состоялось противостояние "файл против файла". Очевидно, один из них был фальсифицирован или изменен, то есть была нарушена его целостность. Суд решил, что подделали электронное письмо (секретарша знала пароль вице-президента, поскольку ей было поручено его менять), и иск был отвергнут...

Угрозой целостности является не только фальсификация или изменение данных, но и отказ от совершенных действий. Если нет средств обеспечить "неотказуемость", компьютерные данные не могут рассматриваться в качестве доказательства.

Потенциально уязвимы с точки зрения нарушения целостности не только данные, но и программы. Внедрение рассмотренного выше вредоносного ПО – пример подобного нарушения.

Угрозами динамической целостности являются нарушение атомарности транзакций, переупорядочение, кража, дублирование данных или внесение дополнительных сообщений (сетевых пакетов и т.п.). Соответствующие действия в сетевой среде называются активным прослушиванием.



Угроза нарушения конфиденциальности заключается в том, что информация становится известной тому, кто не располагает полномочиями доступа к ней. Иногда, в связи с угрозой нарушения конфиденциальности, используется термин "утечка".

Конфиденциальность информации – субъективно определяемая (приписываемая) характеристика (свойство) информации, указывающая на необходимость введения ограничений на круг субъектов, имеющих доступ к данной информации, и обеспечиваемая способностью системы (среды) сохранять указанную информацию в тайне от субъектов, не имеющих полномочий доступа к ней. Объективные предпосылки подобного ограничения доступности информации для одних субъектов заключены в необходимости защиты их законных интересов от других субъектов информационных отношений.

Конфиденциальную информацию можно разделить на предметную и служебную. Служебная информация (например, пароли пользователей) не относится к определенной предметной области, в информационной системе она играет техническую роль, но ее раскрытие особенно опасно, поскольку оно чревато получением несанкционированного доступа ко всей информации, в том числе предметной.

Многим людям приходится выступать в качестве пользователей не одной, - целого ряда систем (информационных сервисов). Если для доступа к таким системам используются многозначные пароли или иная конфиденциальная информация, то наверняка эти данные будут храниться не только в голове, но и в записной книжке или на листках бумаги, которые пользователь часто оставляет на рабочем столе, а то и попросту теряет. И дело здесь не в неорганизованности людей, а в изначальной непригодности парольной схемы.

Невозможно помнить много разных паролей; рекомендации по их регулярной смене только усугубляют положение, заставляя применять несложные схемы чередования или стараться свести дело к двум-трем легко запоминаемым паролям.



Описанный класс уязвимых мест можно назвать размещением конфиденциальных данных в среде, где им не обеспечена необходимая защита. Угроза же состоит в том, что кто-то не откажется узнать секреты, которые сами просятся в руки. Помимо паролей, хранящихся в записных книжках пользователей, в этот класс попадает передача конфиденциальных данных в открытом виде (в разговоре, в письме, по сети), которая делает возможным перехват данных. Для атаки могут использоваться разные технические средства (подслушивание или прослушивание разговоров, пассивное прослушивание сети и т.п.), но идея одна – получить доступ к данным в тот момент, когда они наименее защищены.

Угрозу перехвата данных следует принимать во внимание не только при начальном конфигурировании ИС, но и, что очень важно, при всех изменениях. Еще один пример изменения, о котором часто забывают, – хранение данных на резервных носителях. Для защиты данных на основных носителях применяются развитые системы управления доступом; копии же нередко просто лежат в шкафах и получить доступ к ним могут многие.

Перехват данных – очень серьезная угроза, и если конфиденциальность действительно является критичной, а данные передаются по многим каналам, их защита может оказаться весьма сложной и дорогостоящей. Технические средства перехвата хорошо проработаны, доступны, просты в эксплуатации, а установить их, например на кабельную сеть, может кто угодно, так что эту угрозу нужно принимать во внимание по отношению не только к внешним, но и к внутренним коммуникациям.

Кражи оборудования являются угрозой не только для резервных носителей, но и для компьютеров, особенно портативных. Часто ноутбуки оставляют без присмотра на работе или в автомобиле, иногда просто теряют.

Опасной нетехнической угрозой конфиденциальности являются методы морально-психологического воздействия, такие как маскарад – выполнение действий под видом лица, обладающего полномочиями для доступа к данным



11. неприятным угрозам, от которых трудно защищаться, можно отнести злоупотребление полномочиями. На многих типах систем привилегированный пользователь (например системный администратор) способен прочитать любой (незашифрованный) файл, получить доступ к почте любого пользователя и т.д. Другой пример – нанесение ущерба при сервисном обслуживании. Обычно сервисный инженер получает неограниченный доступ к оборудованию и имеет возможность действовать в обход программных защитных механизмов.

Таковы основные угрозы, которые наносят наибольший ущерб субъектам информационных отношений.

На современном этапе развития информационных технологий под системы или функции защиты являются неотъемлемой частью комплекса по обработке информации. Информация не представляется "в чистом виде", на пути к ней имеется хотя бы какая-нибудь система защиты, и поэтому чтобы угрожать, атакующая сторона должна преодолеть эту систему. Однако не существует абсолютно стойкой системы защиты, вопрос лишь во времени и средствах, требующихся на ее преодоление. Исходя из данных условий, примем следующую модель: **защита информационной системы считается преодоленной, если в ходе ее исследования определены все уязвимости системы.** Поскольку преодоление защиты также представляет собой угрозу, для защищенных систем будем рассматривать ее **четвертый вид - угрозу раскрытия параметров АС**, включающей в себя систему защиты. С точки зрения практики любое проводимое мероприятие предваряется этапом разведки, в ходе которого определяются основные параметры системы, её характеристики, в результате чего уточняется поставленная задача и выбираются оптимальные технические средства.

Угрозу раскрытия можно рассматривать как опосредованную. Последствия ее реализации не причиняют какой-либо ущерб обрабатываемой информации, но дают возможность реализоваться первичным или непосредственным угрозам, перечисленным выше.



Практическое занятие №4. Этапы доступа к ресурсам вычислительной системы; использование простого пароля; использование динамически изменяющегося пароля; взаимная проверка подлинности и другие случаи опознания; способы разграничения доступа к компьютерным ресурсам; разграничение доступа по спискам.

Цель работы: ознакомиться с этапами доступа к ресурсам вычислительной системы.

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА И ЕЕ РЕСУРСЫ

Вычислительная система (ВС) - это взаимосвязанная совокупность аппаратных средств вычислительной техники и программного обеспечения, предназначенная для обработки информации.

Иногда под ВС понимают совокупность технических средств ЭВМ, в которую входит не менее двух процессоров, связанных общностью управления и использования общесистемных ресурсов (память, периферийные устройства, программное обеспечение и т.п.).

Ресурсы вычислительной системы

К ресурсам вычислительной системы относят такие средства вычислительной системы, которые могут быть выделены процессу обработки данных на определенный квант времени. Основными ресурсами ВС являются процессоры, области оперативной памяти, наборы данных, периферийные устройства, программы.

9. зависимости от ряда признаков различают следующие вычислительные системы (ВС):

однопрограммные и многопрограммные (в зависимости от количества программ, одновременно находящихся в оперативной памяти);

индивидуального и коллективного пользования (в зависимости от числа пользователей, которые одновременно могут использовать ресурсы ВС);



с пакетной обработкой и разделением времени (в зависимости от организации и обработки заданий);

однопроцессорные, многопроцессорные и многомашинные (в зависимости от числа процессоров);

сосредоточенные, распределенные (вычислительные сети) и ВС с теледоступом (в зависимости от территориального расположения и взаимодействия технических средств);

работающие или не работающие в режиме реального времени (в зависимости от соотношения скоростей поступления задач в ВС и их решения);

универсальные, специализированные и проблемно-ориентированные (в зависимости от назначения).

РЕЖИМЫ РАБОТЫ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

в Мультипрограммирование

в Режим реального времени

в Однопрограммный режим работы вычислительной системы (ВС)

в Мультипрограммный режим работы вычислительной системы (ВС)

6) Режим пакетной обработки

7) Режим коллективного доступа

Мультипрограммирование - это режим обработки данных, при котором ресурсы вычислительной системы предоставляются каждому процессу из группы процессов обработки данных, находящихся в ВС, на интервалы времени, длительность и очередность предоставления которых определяется управляющей программой этой системы с целью обеспечения одновременной работы в интерактивном режиме.

Режим реального времени - режим обработки данных, при котором обеспечивается взаимодействие вычислительной системы с внешними по отношению к ней процессами в темпе, соизмеримом со скоростью протекания этих процессов.



Этот режим обработки данных широко используется в системах управления и информационно-поисковых системах.

Аппаратные средства ЭВМ совместно с программным обеспечением образуют ВС. В зависимости от класса ЭВМ и вида операционной системы ВС могут работать в режимах **однопрограммном и мультипрограммном**.

5) **однопрограммном режиме** работы в памяти ЭВМ находится и выполняется только одна программ. Такой режим обычно характерен для микро-ЭВМ и персональных ЭВМ, то есть для ЭВМ индивидуального пользования.

6) **мультипрограммном (многопрограммном) режиме** работы в памяти ЭВМ находится несколько программ, которые выполняются частично или полностью между переходами процессора от одной задачи к другой в зависимости от ситуации, складывающейся в системе.

7) мультипрограммном режиме более эффективно используются машинное время и оперативная память, так как при возникновении каких-либо ситуаций в выполняемой задаче, требующих перехода процессора в режим ожидания, процессор переключается на другую задачу и выполняет ее до тех пор, пока в ней не возникает подобная ситуация, и т.д.

При реализации мультипрограммного режима требуется определять очередность переключения задач и выбирать моменты переключения, чтобы эффективность использования машинного времени и памяти была максимальной.

Мультипрограммный режим обеспечивается аппаратными средствами ЭВМ и средствами операционной системы. Он характерен для сложных ЭВМ, где стоимость машинного времени значительно выше, чем у микро-ЭВМ. Разработаны также мультипрограммные ОС, позволяющие одновременно следить за решением нескольких задач и повышать эффективность работы пользователя.

Режим пакетной обработки



11. зависимости от того, в каком порядке при мультипрограммном режиме выполняются программы пользователей, различают режимы пакетной обработки задач и коллективного доступа.

12. режиме пакетной обработки задачи выстраиваются в одну или несколько очередей и последовательно выбираются для их выполнения.

Режим коллективного доступа

13. режиме коллективного доступа каждый пользователь ставит свою задачу на выполнение в любой момент времени, то есть для каждого пользователя в такой ВС реализуется режим индивидуального пользования. Это осуществляется обычно с помощью квантования машинного времени, когда каждой задаче, находящейся в оперативной памяти ЭВМ, выделяется квант времени. После окончания кванта времени процессор переключается на другую задачу или продолжает выполнение прерванной в зависимости от ситуации в ВС. Вычислительные системы, обеспечивающие коллективный доступ пользователей с квантованием машинного времени, называют ВС с разделением времени.

Объектно-ориентированное программирование

Объектно-ориентированное программирование представляет собой метод программирования, который весьма близко напоминает наше поведение. Оно является естественной эволюцией более ранних нововведений в разработке языков программирования. Объектно-ориентированное программирование является более структурным, чем все предыдущие разработки, касающиеся структурного программирования. Оно также является более модульным и более абстрактным, чем предыдущие попытки абстрагирования данных и переноса деталей программирования на внутренний уровень. Объектно-ориентированный язык программирования характеризуется тремя основными свойствами:

11) **Инкапсуляция.** Комбинирование записей с процедурами и функциями, манипулирующими полями этих записей, формирует новый тип данных - объект.



- **Наследование.** Определение объекта и его дальнейшее использование для построения иерархии порожденных объектов с возможностью для каждого порожденного объекта, относящегося к иерархии, доступа к коду и данным всех порождающих объектов.

7. Полиморфизм. Присваивание действию одного имени, которое затем совместно используется вниз и вверх по иерархии объектов, причем каждый объект иерархии выполняет это действие способом, именно ему подходящим.

В разработке программного обеспечения, **стадии разработки программного обеспечения** используются для описания степени готовности программного продукта. Также стадия разработки может отражать количество реализованных функций, запланированных для определённой версии программы. Стадии либо могут быть официально объявлены и регламентируются разработчиками, либо иногда этот термин используется неофициально для описания состояния продукта.

Пре-альфа - Начальная стадия разработки — Период времени со старта разработки до выхода стадии Альфа (или до любой другой, если стадии Альфа нет). Также так называются программы, не вышедшие еще в стадию альфа или бета, но прошедшие стадию разработки, для первичной оценки функциональных возможностей в действии. В отличие от альфа и бета версий, пре-альфа может включать в себя не весь спектр функциональных возможностей программы. В этом случае, подразумеваются все действия выполняемые во время проектирования и разработки программы вплоть до тестирования. К таким действиям относятся — разработка дизайна, анализ требований, собственно разработка приложения, а также отладка отдельных модулей.

Альфа-тестирование - Внутреннее тестирование — Стадия начала тестирования программы в целом специалистами-тестерами, обычно не разработчиками программного продукта, но, как правило, внутри организации



или сообществе разрабатывающих продукт. Также это может быть стадия добавления новых функциональных возможностей. Программы на данной стадии могут применяться только для ознакомления с будущими возможностями.

Бета-тестирование - Публичное тестирование — Стадия активного бета-тестирования и отладки, прошедшей альфа-тестирование (если таковое было). Программы этого уровня могут быть использованы другими разработчиками программного обеспечения для испытания совместимости. Тем не менее программы этого этапа могут содержать достаточно большое количество ошибок.

Релиз-кандидат или RC (англ. release candidate) — стадия-кандидат на то, чтобы стать стабильной. Программы этой стадии прошли комплексное тестирование, благодаря чему были исправлены все найденные критические ошибки. Но в то же время, существует вероятность выявления ещё некоторого числа ошибок, не замеченных при тестировании.

Релиз или RTM (англ. release to manufacturing) — стабильная версия программы, прошедшая все предыдущие стадии, в которых исправлены основные ошибки, и готовая к применению.

Парольная защита компьютера

Для обеспечения безопасности ПК необходимо обеспечить защиту отдельных файлов и папок и предпринять действия по физической защите компьютера. Если на компьютере имеются конфиденциальные сведения, они должны храниться в безопасном месте.

Простейшими способами защиты компьютера являются его блокировка на время отсутствия на рабочем месте и настройка заставки, защищенной паролем.

Блокировка ПК

Нажмите комбинацию клавиш **CTRL+ALT+DELETE**.

4 В появившемся окне нажмите кнопку **Блокировка**. В результате отобразится диалоговое окно **Блокировка компьютера**. Теперь компьютер



заблокирован, что не позволит никому, кроме администратора и пользователя, заблокировавшего компьютер, войти в систему и открывать файлы и программы.

8. Для разблокирования ПК необходимо вновь нажать комбинацию клавиш **CTRL+ALT+DEL**, ввести пароль, а затем нажать кнопку **ОК**.

Защита файлов с помощью пароля экранной заставки

17. Щелчком правой кнопки мыши по свободному месту Рабочего стола откройте контекстное меню и выберите команду **Свойства**.

18. В открывшемся диалоговом окне **Свойства: Экран** перейдите на вкладку **Заставка**.

19. В раскрывающемся списке **Заставка** выберите какую-либо заставку.

20. Включите флажок **Защита паролем** и нажмите кнопку **Применить**.

При установке флажка **Защита паролем** работа ПК блокируется при активизации заставки. Для разблокирования ПК при возобновлении работы необходимо ввести пароль, который совпадает с паролем текущего пользователя.

Использование паролей

Эффективным средством защиты, используемым для управления входом в систему по учетным записям пользователей, а также организации доступа к компьютерам и ресурсам является пароль.

Пароль – это уникальный набор разрешенных символов, который должен быть введен пользователем для проверки его учетного имени и получения доступа к ресурсам ПК.

Для организации надежной защиты ПК пароль должен выбираться, исходя из следующих требований:

7. иметь длину не менее семи символов (наиболее надежные пароли состоят из 7 - 14 символов);

8. содержать символы каждой из трех следующих групп:

буквы (прописные и строчные) А, В, С,...; а, в, с,...; А, Б, В,...; а, б,

В,...



цифры 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

специальные символы `~!@#\$\$%^&*()_+={}|[\:";'<>

?,./

6. содержать хотя бы один специальный символ на позициях со второй по шестую;

7. существенно отличаться от ранее использованных паролей;

8. не содержать фамилии или имени пользователя;

9. не являться распространенным словом или именем.

Следует помнить, что программное обеспечение для взлома паролей использует один из трех подходов: *угадывание*, *подбор вариантов* и *автоматический перебор всех возможных комбинаций символов*. Имея достаточное количество времени можно взломать любой пароль методом автоматического перебора.

В Windows XP пароль учетной записи пользователя может содержать до 127 символов, однако если компьютер используется в локальной сети, где есть ПК с ОС Windows 95/98, поддерживающих пароли длиной до 14 знаков, использовать пароли большей длины нельзя.

Для обеспечения безопасности необходимо быть аккуратным при использовании паролей. Следует руководствоваться следующими рекомендациями:

никогда не записывайте свой пароль в каком-либо обозримом месте;

не сообщайте пароль никому;

не используйте сетевой пароль для других целей;

используйте разные пароли для входа в сеть и учетной записи администратора на компьютере;

изменяйте свой пароль каждые 60-90 дней;

измените пароль немедленно, если возникнут подозрения, что он был раскрыт;



будьте осторожны при сохранении пароля на компьютере. В некоторых диалоговых окнах имеется возможность сохранить пароль. Никогда не устанавливайте этот параметр.

Программные средства и способы защиты компьютерной информации

Защита от несанкционированного доступа к ресурсам компьютера: пароли, разграничение доступа.

Данные меры защиты предусматривают защиту доступа к дисковым накопителям информации, к клавиатуре и дисплею компьютера.

Защита жёстких и гибких магнитных дисков предусматривает:

защиту от любого несанкционированного доступа к диску;

разграничение доступа пользователей к файлам дисков;

контроль обращения к диску и проверка целостности защиты информации диска;

периодические проверки наличия вирусов на диске;

стирание в файлах остатков закрытой информации.

Защита дисков от несанкционированного доступа к ним осуществляется с помощью паролей. Пароль – это набор символов, который соответствует определенному объекту идентификации. Пароли для защиты дисков делятся:

по типу объектов идентификации: пароли пользователей, ресурсов

10. файлов;

по типу символов: цифровые, буквенные, смешанные;

по способу ввода в компьютер: с клавиатуры, с помощью мыши, со специальной ключевой дискеты;

по срокам применения: с неограниченным сроком, периодически сменяемые, разовые;

по длине: фиксированной и переменной длины.

Чем больше длина пароля и меньше срок его использования, тем выше уровень защиты соответствующего диска.



Парольная защита дисков осуществляется, специальными компьютерными программами, а также утилитой «Пароли» в Панели управления операционной системы Windows98.

Более детальная процедура доступа к файлам организуется на основе таблиц разграничения доступа пользователей к файлам по назначению файлов, а также по характеру работы пользователей (создание, чтение, редактирование, удаление файлов и другие). Разработаны специальные программы, предоставляющие пользователям только те файлы и те возможности работы с ними, которые указаны в таблице разграничения доступа. Разработан также ряд специальных программ, которые фиксируют даты и время обращения к диску, а также случаи «взлома» защиты диска, несанкционированной работы с ним.

Необходимость стирания в файлах остатков закрытой информации вызвана тем, что, во-первых, при удалении файла командами MS DOS, Norton Commander или Windows стирается только имя файла, а не сама информация на диске; во-вторых объём данных любого файла обычно меньше отведенного для него пространство на диске, поэтому в конце (в «хвосте») нового файла могут сохраниться остатки закрытой информации от предыдущего файла. Операцию стирания «хвостов» файлов осуществляет, например, утилита Wipe Info из пакета Norton Utilities.

Защита клавиатуры и дисплея применяется, когда пользователь отлучается с рабочего места на короткое время. Например, утилита Diskreet из пакета Norton Utilities блокирует клавиатуру и гасит экран при ее вводе. Ограничения снимаются введением пароля, т.е. когда пользователь возвращается к работе в предположении, что никто кроме него этот пароль не знает.

Защита дисков от копирования

Основное направление действий по защите от копирования дисков относится к дискетам, т.е. гибким дискам, так как такие диски легче похитить и скопировать. Применяются следующие способы защиты дискет от копирования:



парольная защита дискеты, при которой без ввода пароля дискета не копируется;

привязка информации к определенной дискете: фирме и типу дискеты, нестандартному способу форматирования дискеты, нанесению уникальных признаков на дискету, связанных с введением заранее помеченных «сбойных» участков;

привязка информации к определенному компьютеру, например, к его тактовой частоте, параметрам накопителей, дисплея, принтера и других устройств.

7. соответствии с задачами защиты от копирования имеется много программ защиты, в том числе, обеспечивающих многоуровневую систему защиты дискет от копирования.

Жесткие диски компьютеров чаще всего защищают от копирования с помощью использования паролей.

Разграничение доступа

После выполнения идентификации и аутентификации необходимо установить полномочия (совокупность прав) субъекта для последующего контроля санкционированного использования вычислительных ресурсов, доступных в АС. Такой процесс называется разграничением (логическим управлением) доступа.

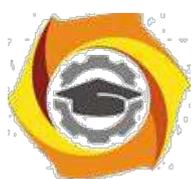
Обычно полномочия субъекта представляются: списком ресурсов, доступных пользователю, и правами по доступу к каждому ресурсу из списка. В качестве вычислительных ресурсов могут быть программы, информация, логические устройства, объем памяти, время процессора, приоритет и т. д.

Обычно выделяют следующие методы разграничения доступа:

разграничение доступа по спискам;

использование матрицы установления полномочий;

по уровням секретности и категориям;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

парольное разграничение доступа.



При разграничении доступа по спискам задаются соответствия:
каждому пользователю – список ресурсов и прав доступа к ним или
каждому ресурсу – список пользователей и их прав доступа к
данному ресурсу.

Списки позволяют установить права с точностью до пользователя. Здесь нетрудно добавить права или явным образом запретить доступ. Списки используются в большинстве ОС и СУБД.

Использование матрицы установления полномочий подразумевает применение матрицы доступа (таблицы полномочий). В указанной матрице (см. таблицу 2.7) строками являются идентификаторы субъектов, имеющих доступ в АС, а столбцами – объекты (информационные ресурсы) АС. Каждый элемент матрицы может содержать имя и размер предоставляемого ресурса, право доступа (чтение, запись и др.), ссылку на другую информационную структуру, уточняющую права доступа, ссылку на программу, управляющую правами доступа и др.

Таблица 2.7 - Фрагмент матрицы установления полномочий

Субъект	Каталог d:\Heap	Программа prty	Принтер
Пользователь 1	cdrw	e	w
Пользователь 2	r		w с 9:00 до 17:00

c – создание, d – удаление, r – чтение, w – запись, e – выполнение.

Данный метод предоставляет более унифицированный и удобный подход, т. к. вся информация о полномочиях хранится в виде единой таблицы, а не в виде разнотипных списков. Недостатками матрицы являются ее возможная громоздкость и не совсем оптимальное использование ресурсов (большинство клеток – пустые).

Разграничения доступа по уровням секретности и категориям состоят в том, что ресурсы АС разделяются в соответствии с уровнями секретности или категорий.



При разграничении по уровню секретности выделяют несколько уровней, например,: общий доступ, конфиденциально, секретно, совершенно секретно. Полномочия каждого пользователя задаются в соответствии с максимальным уровнем секретности, к которому он допущен. Пользователь имеет доступ ко всем данным, имеющим уровень (гриф) секретности не выше, чем он имеет.

При разграничении по категориям задается и контролируется ранг категории, соответствующей пользователю. Соответственно, все ресурсы АС декомпозируют по уровню важности, причем определенному уровню соответствует некоторый ранг персонала (типа: руководитель, администратор, пользователь).

Парольное разграничение, очевидно, представляет использование методов доступа субъектов к объектам по паролю. При этом используются все методы парольной защиты. Очевидно, что постоянное использование паролей создает неудобства пользователям и временные задержки. Поэтому указанные методы используют в исключительных ситуациях.

На практике обычно сочетают различные методы разграничения доступа. Например, первые три метода усиливают парольной защитой.

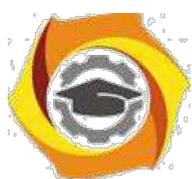
4. завершении подраздела заметим, что руководящие документы могут регламентировать два вида (принципа) разграничения доступа:

дискретное управление доступом;

мандатное управление доступом.

Дискретное управление доступом представляет собой разграничение доступа между поименованными субъектами и поименованными объектами. Субъект с определенным правом доступа может передать это право любому другому субъекту. Данный вид организуется на базе методов разграничения по спискам или с помощью матрицы.

Мандатное управление доступом регламентирует разграничение доступа субъектов к объектам, основанное на характеризуемой меткой конфиденциальности информации, содержащейся в объектах, и официальном



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

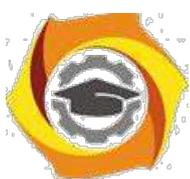
разрешении

(допуске)

субъектов

обращаться

к информации



такого уровня конфиденциальности. Иначе, для реализации мандатного управления доступом каждому субъекту и каждому объекту присваивают классификационные метки, отражающие их место в соответствующей иерархии. С помощью этих меток субъектам и объектам должны быть назначены классификационные уровни, являющиеся комбинациями уровня иерархической классификации и иерархических категорий. Данные метки должны служить основой мандатного принципа разграничения доступа. Ясно, что методы разграничения доступа по уровням секретности и категориям являются примерами мандатного управления доступом.

Методические рекомендации для магистрантов по организации и выполнению самостоятельной работы по "ГИА".

Тематика самостоятельной работы

№	Наименование работы
1	Выполнение выпускной квалификационной работы



Оформление и подготовка магистерской диссертации

В Общие положения

1.1. Магистерская диссертация (далее – МД) является выпускной квалификационной работой (далее – ВКР), подготовленной для публичной защиты и показывающей уровень профессиональной подготовки магистранта, умение самостоятельно вести научный поиск и решать практические задачи в сфере профессиональной деятельности обучающихся в магистратуре Технического университета УГМК (далее ТУ УГМК). Публичная защита осуществляется в рамках государственной итоговой аттестации (далее ГИА), которая проводится государственными экзаменационными комиссиями. Успешная публичная защита ВКР перед Государственной экзаменационной комиссией (далее ГЭК) наряду с освоением учебного плана соответствующей образовательной программы является главным основанием для присвоения соответствующей степени обучающимся ТУ УГМК.

1.2. ВКР по магистерской программе является решением производственной задачи, выданной предприятием магистранту на период обучения в магистратуре Технического Университета УГМК.

1.3. Подготовка магистерской диссертации и порядок ее защиты регламентированы в ТУ УГМК Положением о государственной итоговой аттестации по образовательным программам высшего образования и Стандартом УГМК – 044-2017.

1.4. Магистерская диссертация должна быть написана студентом самостоятельно, обладать внутренним единством, содержать новые научные результаты, сведения о практическом использовании полученных результатов. В процессе написания магистерской диссертации студент должен опираться на информацию, полученную в ходе прохождения практики и выполнения научно-исследовательской работы.

1.5. Магистерская диссертация должна содержать совокупность результатов и научных положений, выдвигаемых магистрантом для защиты, иметь внутреннее единство, свидетельствовать о способности автора самостоятельно вести научные исследования, используя теоретические знания и практические навыки, видеть профессиональные проблемы, знать методы и приемы их решения.

1.6. В силу специфики и прикладного характера обучения в Техническом университете УГМК тематика ВКР заранее установлена предприятиями.



с Этапы выполнения магистерской диссертации

Процесс выполнения магистерской диссертации состоит из последовательности этапов, которые в плане выполнения исследования по функциональному признаку разделены следующим образом:

- предварительный этап,
- исследовательский этап,
- этап оформления,
- этап защиты работы.

2.1. *Предварительный этап (1 семестр)*

На предварительном этапе определяется общая проблемная ситуация в выбранном направлении (объекте) исследований и выявляются противоречия, требующие разрешения; производится научный поиск, изучение литературных источников и т.п. Результатом является определение круга задач по выполнению производственной задачи, определенной предприятием. Уточняется с предприятием тема исследования с учетом ее значимости и своевременности, которые совместно с сутью проблемной ситуации определяют актуальность работы. Логическим завершением предварительного этапа является формулировка цели исследования и конкретных задач, которые необходимо решить и которые определяют стратегию и направление исследований (задачи в процессе работы могут корректироваться). Организующим началом в подготовке магистерской диссертации является её план. Сначала с предприятием обсуждается укрупненный перспективный план выполнения магистерской диссертации, определяется ее структура, необходимые главы и разделы. Затем магистрант тщательно прорабатывает все имеющиеся информационные материалы и научную литературу. Детально обсуждает все возникающие вопросы с научным руководителем.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Результатами предварительного этапа работы является составление Технического задания на выполнение магистерской диссертации (далее ТЗ). Форма ТЗ утверждается на выпускающей кафедре, пример ТЗ в приложении 1. В ТЗ указывается перечень требований, которые должны быть учтены при разработке магистерской диссертации, перечень документов, на основании которых разрабатывается ВКР, кем и когда будут утверждены эти документы; плановые сроки начала и окончания работ по выполнению ВКР; наименование работ по ВКР; порядок оформления и предъявления Заказчику результатов работ; производственная задача, цели и задачи ВКР, требования к ВКР и проекту в целом. По разработанному ТЗ разрабатывается график выполнения магистерского проекта. График может корректироваться в ходе выполнения проекта.

Магистранту обязательно необходимо сформировать список литературных источников (библиографию) по выбранной теме работы. Изучение литературы позволяет определить понятийный аппарат, ключевые слова, основные источники по выбранной теме. По результатам литературного обзора, как правило, оформляются статьи и доклады на конференциях.

Источником информации при проведении «литобзора» в первую очередь являются справочно-поисковые системы в сети Интернет, которые позволяют быстро найти требуемые тематические сайты и литературные источники, законодательные и нормативные материалы. Магистранты Технического университета УГМК имеют возможность изучения и поиска литературы в электронно-библиотечных системах, с которыми университет имеет соглашения о сотрудничестве (допуск к указанным системам будет предоставлен при зачислении). Это дает магистрантам дополнительные возможности безопасного и быстрого поиска по полнотекстовым базам данных литературы ведущих отечественных издательств.

ВКР магистранта размещается в системе управления обучением 1С для рассмотрения экспертами и контроля за ходом выполнения работы. Эксперту выдается логин и пароль для входа в систему через портал ТУ УГМК.

2.2. Исследовательский этап (2-4 семестры)

На исследовательском этапе производится выбор методов исследования и математического аппарата, т.е. инструмента, необходимого для достижения поставленной цели, а также методики и техники исследования.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Выполняется само исследование, которое помимо теоретической части должно содержать практическую реализацию поставленной задачи в виде проекта, подтверждающего достоверность, эффективность и практическую значимость полученных результатов исследования.

7. заключении этапа дается оценка полученных результатов, в том числе укрупненно-экономическая, формулируются выводы по работе, часть из которых должна определять научную новизну, другая – практическую значимость.

При завершении данного этапа (или промежуточного этапа) заведующие кафедрами совместно с научными руководителями, кураторами-экспертами и экспертами дирекций по направлениям осуществляют рассмотрение промежуточных результатов выполнения магистерских проектов для оценки и возможно корректировки проделанной работы. По итогам рассмотрения результатов работы оформляется протокол, в котором отмечаются достижения и основные вехи проекта, а также замечания и рекомендации по улучшению проекта

2.3. Этап оформления (4 семестр)

Определяется композиция (структура) и содержание диссертации, обеспечивающие логическую последовательность изложения.

8. соответствии с содержанием (оглавлением) оформляется предварительный вариант текста, включая иллюстрации и таблицы. Текст должен быть информативным и в то же время кратким и четким, оформлен в соответствии с требованиями к научной работе. Основные положения по оформлению научной работы приведены ниже.

На этом этапе оформляются основные структурные элементы МД и прочие документы. Производится научное редактирование текста, просмотр таблиц и иллюстраций магистрантом и научным руководителем, нормоконтроль МД нормоконтролером, окончательное оформление результатов исследований в виде завершённой диссертации, которая может быть представлена к защите.

2.4. Этап защиты (5 семестр)

9. Техническом университете УГМК этап защиты диссертации многоступенчатый. Он включает в себя процедуру обсуждения магистерской диссертации сначала на предприятии, а затем на кафедре с участием экспертов от профильных дирекций



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

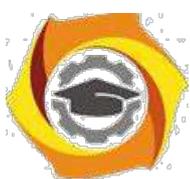
Холдинга. Данная процедура является своего рода предзащитой, на которой заинтересованные специалисты предприятий, ТУ УГМК и Холдинга определяют степень готовности ВКР, возможно делают замечания, дают свои рекомендации. Результатом этих процедур являются протоколы совещаний. Организационные моменты и сроки проведения этих процедур определяют заведующие кафедрами совместно с дирекциями по направлениям Холдинга.

Затем магистерская диссертация сдается научному руководителю, который дает письменное заключение (отзыв) о степени соответствия работы требованиям, предъявляемым к ВКР, а автора – искомой степени магистра. Далее диссертация передается внешнему рецензенту из числа компетентных в соответствующей отрасли науки ученых (доктору или кандидату наук в данной области). Рецензент дает отзыв на МД. В отзыве на основе изучения МД оценивается актуальность темы, степень обоснованности положений, выводов и рекомендаций, сформулированных в диссертации, их достоверность, новизна и практическая значимость дается общая оценка работы, в том числе с указанием недостатков и других замечаний, а также дается заключение о соответствии МД критериям для присуждения квалификации магистра. Рецензия и отзыв оглашается на заседании ГЭК при защите и обсуждении диссертации.

Магистрант готовит к защите доклад или его тезисы, а также иллюстративный материал (презентацию в электронном виде и распечатывает раздаточный материал для каждого члена ГЭК). Целесообразнее всего тезисы доклада и иллюстрации представить в виде слайдов или использовать компьютерное сопровождение доклада в режиме презентации. Готовятся ответы на замечания руководителя и рецензента, если они отмечены в отзывах (приложение 2).

Подготовка выпускной квалификационной работы завершается публичной защитой перед Государственной экзаменационной комиссией, которая принимает решение о присвоении магистранту квалификации магистра.

По желанию заказчика может быть организована прямая трансляция защиты ВКР
10. возможностью участия в обсуждении доклада с использованием средств видеокон-ференцсвязи.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

в Оформление магистерской диссертации

3.1. Оформление МД по ГОСТ Р 7.0.11- 2011 «Система стандартов по информации, библиотечному и издательскому делу. Диссертация и автореферат диссертации. Структура и правила оформления», 2.105-95 «Единая система конструкторской документации. Общие требования к текстовым документам», ГОСТ 15.101-98 «Порядок выполнения научно-исследовательских работ», ГОСТ 7.32-2017 - «Система стандартов по информации, библиотечному и издательскому делу. Отчёт о научно-исследовательской работе. Структура и правила оформления».

3.2. Вне зависимости от решаемой задачи и используемого подхода при проектировании, магистерская диссертация должны содержать следующие основные структурные элементы:

В титульный лист (приложение 3);

-задание на выполнение магистерской диссертации, утвержденное заведующим кафедрой (приложение 4);

с оглавление;

с текст диссертации:

с введение,

с обоснование актуальности выбранной темы и новизны работы,

с основная часть (разделение на главы обозначены в методических рекомендациях по направлениям подготовки),

с заключение;

с списки;

с приложения.

Общий объем диссертации составляет 80-100 стр. (без учета приложений).

3.2.1. Титульный лист МД оформляется одинаково для всех направлений подготовки. МД присваивается обозначение (шифр), состоящее из следующих номеров и кодов, разделяемых тире:

индекса МД – магистерская диссертация;

кода учебного заведения по общероссийскому классификатору предприятий и организаций (ОКПО);

код направления подготовки;



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

код основной профессиональной образовательной программы ТУ УГМК
(Реестр ОПОП ТУ УГМК, Приложение №6);

х номер зачетной книжки.

Например:



Шифр МД для студентов магистрантов:

МД — 39902925 - 130402 – 01 – 192 – 20

3.2.2. Оглавление это - перечень основных частей диссертации с указанием страниц, на которые их помещают. Заголовки в оглавлении должны точно повторять заголовки в тексте. Не допускается сокращать или давать заголовки в другой формулировке. Последнее слово заголовка соединяют отточием с соответствующим ему номером страницы в правом столбце оглавления.

3.2.3. Текст диссертации:

Введение к диссертации включает в себя следующие основные структурные элементы:

- актуальность темы исследования;
- степень ее разработанности;
- цели и задачи;
- предмет и объект;
- научную новизну;
- теоретическую и практическую значимость работы;
- методологию и методы исследования;
- положения, выносимые на защиту;
- степень достоверности и апробацию результатов.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Объем введения рекомендуется ограничить 1-3 страницами текста.

Основная часть диссертации состоит из трех глав иногда четырех глав (разделение на главы обозначены в методических рекомендациях по направлениям подготовки), общим рекомендованным объемом 70-80 страниц. Каждая глава должна заканчиваться выводами. Содержание основного текста должно точно соответствовать заявленной теме работы и полностью раскрывать данную тему и сформулированные вопросы исследования. Главы должны быть сопоставимыми по объему и включать в себя теоретический и практический материал.

Теоретический материал представляет собой обобщение и обзор теоретических аспектов рассматриваемой предметной области и обязательно включает в себя обзор научной зарубежной и отечественной литературы по теме диссертации. При теоретическом обзоре ведущее место занимает рассмотрение различных теоретических подходов, методов и моделей. Обосновывается их применимость или невозможность использования в выпускной квалификационной работе, то есть дается критическая оценка современного состояния научной мысли применительно к состоянию исследуемой проблемы.

При наличии дискуссионности подходов в какой-либо предметной области необходимо показать отличающиеся точки зрения, отметив ту, которой будет придерживаться и которую в дальнейшем развивает магистрант. Так же необходимо охарактеризовать состояние подобных проблем в других корпорациях, в Холдинге, за рубежом и накопленный опыт их решения.

Уместно определиться с дефинициями, дать однозначное определение используемых терминов, категорий, привести, если необходимо, принятые классификации, систематизировать факторы, оказывающие влияние на исследуемый объект (предприятие).

Практический материал представляется в последующих главах, определенных 3. зависимости от тематики ВКР и включает описание результатов исследования на основе рассмотренных ранее методов, методик и моделей. Предоставление практических рекомендаций на основе анализа результатов расчетов.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Проектная часть диссертации чаще всего представляет собой обоснование автором разработанных им алгоритмов решения проблемы и включает в себя совокупность конкретных практических рекомендаций, направленных на повышение эффективности деятельности исследуемого предприятия. Здесь же приводятся результаты расчета экономического эффекта от реализации предлагаемых технологических и технических решений.

7. заключения диссертации излагают итоги выполненного исследования, рекомендации, перспективы дальнейшей разработки темы, внедрения проекта. Рекомендованный объем 1-2 страницы. Стил ь изложения материалов работы должен быть академическим, без риторических вопросов, многоточий, обращений к читателю и лирических отступлений. Речь должна идти от третьего лица. Не следует писать: «Я получил следующие результаты:...». Надо писать: «Были получены следующие результаты:...». Либо: «Автором были получены следующие результаты:...». Надо писать: «В данной работе были получены следующие результаты:...» и т.п. Когда описывается текущее состояние дел в изучаемой области или научной группе, в которой выполнялась работа, следует использовать настоящее время, а когда речь идет о результатах, полученных лично автором, следует использовать прошедшее время. Например: «Рассматриваемая экономическая модель не позволяет оценить параметры Предложенная модель позволила устранить эти недостатки».

Заключение представляет собой последовательное, логически стройное изложение полученных итогов и их соотношение с общей целью и задачами, сформулированными во «Введении». Это – не простое механическое суммирование выводов, которыми заканчивается каждая глава, а формулирование того нового, существенного, что составляет (с точки зрения автора) новизну, теоретическую и практическую значимость выполненного исследования. Хорошо написанное заключение отражает уровень профессиональной и научной квалификации её автора.

Тексты ВКР, за исключением текстов ВКР, содержащих сведения, составляющие государственную тайну, размещаются в электронно-библиотечной системе ТУ УГМК и проверяются на объём заимствования, в том числе содержательного, выявления неправомерных заимствований и не должен превышать 30%. Порядок разме-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

щения текстов выпускных квалификационных работ в электронно-библиотечной системе, проверки на объём заимствования, устанавливается специальной инструкцией. Контроль и ответственность за проведение данной процедуры возлагается на научного руководителя. Перед процедурой размещения студент должен написать аннотацию на ВКР (приложение 5).

3.3. Параметры страницы диссертации.

Работа должна быть выполнена печатным способом с использованием компьютера и принтера на одной стороне листа белой бумаги одного сорта формата А4 (размер 210 на 297 мм).

Размер шрифта основного текста – 14 пунктов (на рисунках и в таблицах допускается применение более мелкого размера шрифта), гарнитура – Times New Roman. Междустрочный интервал – полуторный, интервалы перед и после абзацев – не применяются. Отступ первой строки абзацев – 1,25 см, отступы справа и слева – не применяются. Выравнивание основного текста – по ширине, с автоматической расстановкой переносов. Текст работы следует печатать, соблюдая следующие размеры полей: левое – 30 мм, правое – 15 мм, верхнее и нижнее – 20 мм.

Ориентация страниц основного текста – книжная. Для рисунков и таблиц допускается применение альбомной ориентации страниц. Размещение текста на страницах альбомной ориентации не рекомендуется (за исключением подписей и комментариев к таблицам и иллюстративным материалам). Текст ВКР должен быть сброшюрован.

Основной текст должен быть разделен на главы и параграфы или разделы и подразделы, которые нумеруют арабскими цифрами. Слово «Глава» в заголовке глав не ставится. После номера раздела, подраздела, пункта и подпункта в тексте точку не ставят. Если раздел или подраздел имеет только один пункт или пункт имеет один подпункт, то нумеровать его не следует.

Каждую главу диссертации начинают с новой страницы. Следующий параграф внутри одной главы начинается на том же листе, где закончился предыдущий.

3.4. Заголовки оглавления, введения, глав основной части ВКР, заключения и списка литературы следует печатать заглавными (прописными) буквами и распола-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

гать по центру строки без точки в конце (не подчеркивая, размер шрифта – 14, полужирный). Заголовки подразделов и пунктов следует начинать с абзацного отступа и печатать с прописной буквы без точки в конце (не подчеркивая, размер шрифта – 14, полужирный).

Если заголовок включает несколько предложений, их разделяют точками. Переносы слов в заголовках не допускаются. Заголовки отделяют от текста сверху и снизу двумя интервалами. Не допускается размещать заголовок на одной странице, а текст – на другой.

Все страницы диссертации, включая иллюстрации и приложения, нумеруются арабскими цифрами по порядку без пропусков и повторений. Первой страницей считается титульный лист, на котором нумерация страниц не ставится, следующая страница нумеруется «2» и т.д. Порядковый номер страницы проставляют в центре в нижней части листа без точки (гарнитура Times New Roman, размер шрифта – 12).

В тексте документа не допускается:

применять обороты разговорной речи, техницизмы, профессионализмы;

применять для одного и того же понятия различные научно-технические термины, близкие по смыслу (синонимы), а также иностранные слова и термины при наличии равнозначных слов и терминов в русском языке;

применять произвольные словообразования;

применять сокращения слов, кроме установленных правилами русской орфографии, соответствующими государственными стандартами, а также в данном документе;

сокращать обозначения единиц физических величин, если они употребляются без цифр, за исключением единиц физических величин в головках и боковиках таблиц 8. в расшифровках буквенных обозначений, входящих в формулы и рисунки.

В тексте документа, за исключением формул, таблиц и рисунков, не допускается:

- применять математический знак минус (-) перед отрицательными значениями величин (следует писать слово "минус");

- применять знак "∅" для обозначения диаметра (следует писать слово "диа-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

метр"). При указании размера или предельных отклонений диаметра на чертежах, помещенных в тексте документа, перед размерным числом следует писать знак "∅";

- применять без числовых значений математические знаки, например > (больше), < (меньше), = (равно), ≥ (больше или равно), ≤ (меньше или равно), ≠ (не равно), а также знаки N (номер), % (процент).

Все иллюстрирующие материалы выпускной квалификационной работы (рисунки, чертежи, схемы, диаграммы, графики, фотографии) называют *рисунками*. На все рисунки должны быть ссылки в тексте работы. При ссылках на иллюстрации следует писать «...приведены на рисунке 1.2» или «на рисунке 1.2 приведен ...». Рисунки следует располагать непосредственно после текста, в котором они упоминаются впервые, или на следующей странице.

При цветном исполнении рисунков следует использовать принтер с возможностью цветной печати. При использовании в черно-белой печати следует применять черно-белую штриховку элементов рисунка.

Нумерация рисунков идет арабскими цифрами в пределах раздела. Номер рисунка состоит из номера раздела и порядкового номера иллюстрации, разделенного точкой. Иллюстрации должны иметь наименование и пояснительные данные (подрисуночный текст). Слово «Рисунок» и наименование (через тире) располагают по центру следующим образом: «Рисунок 1.2 – Классификация рынков» (см. пример 1). Не допускается размещать подрисуночный текст на одной странице, а рисунок – на другой.

Иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения. Например - Рисунок А.3.

Иллюстрации, при необходимости, могут иметь наименование и пояснительные данные (подрисуночный текст). Слово «Рисунок» и наименование помещают после пояснительных данных.

Рисунки располагаются непосредственно после их упоминания в тексте, или на следующей странице, если рисунок не размещается на данной странице.



Пример 1



Рисунок 1.2 – Классификация рынков

3.5. Таблицы применяют для лучшей наглядности и удобства сравнения показателей. Название таблицы должно отражать ее содержание, быть точным, кратким. Название таблицы следует помещать над таблицей слева, без абзацного отступа в одну строку с ее номером через тире (см. пример 2). Не допускается размещать название таблицы на одной странице, а таблицу – на другой.

Ширину таблицы и полей устанавливают аналогично основному тексту (за исключением случаев, когда требуется применение других параметров). Цифры в графах таблиц должны проставляться так, чтобы разряды чисел во всей графе были расположены один под другим, если они относятся к одному показателю. Для этого числовые значения, как правило, выравнивают по правому краю ячеек таблицы. В качестве разделителей разрядов рекомендуется использовать пробелы. В одной графе должно быть соблюдено одинаковое количество десятичных знаков для всех значений величин. Не допускается перенос числовых значений в ячейках таблицы на две и более строки.

Пример 2

Таблица 3.13 – Динамика величины основных фондов ПАО «Мегафон»

Наименование показателя	Ед. изм.	Отчетный период		
		I пол. 2014 г.	II пол. 2014 г.	I пол. 2015 г.
Первоначальная стоимость основных фондов, млрд. руб.	Млрд. руб.	13 092,05	13 840,23	14 578,35
Остаточная стоимость основных фондов, млрд. руб.		9 059,25	9 564,35	10 047,87

Нумерация таблиц идет в пределах раздела арабскими цифрами. Номер таблицы состоит из номера раздела и порядкового номера таблицы, разделенных точкой.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Таблицу следует располагать непосредственно после текста, в котором она упоминается впервые, или на следующей странице. На каждую таблицу в тексте должна быть ссылка. При этом следует полностью писать слово «таблица» с указанием её номера («в таблице 2.1 приводится...»).

Заголовки граф и строк таблицы следует писать с прописной буквы в единственном числе, а подзаголовки граф – со строчной буквы, если они составляют одно предложение с заголовком, или с прописной буквы, если они имеют самостоятельное значение. В конце заголовков и подзаголовков таблиц точки не ставят.

Таблицу с большим количеством строк (граф) допускается делить на части и размещать на нескольких страницах. В этом случае на каждой странице повторяются заголовки соответствующих граф (строк).

Если все показатели, приведенные в графах таблицы, выражены в одной и той же единице физической величины, то ее обозначение необходимо помещать над таблицей справа, а при делении таблицы на части - над каждой ее частью.

Если в большинстве граф таблицы приведены показатели, выраженные в одних и тех же единицах физических величин (например, в миллиметрах, вольтах), но имеются графы с показателями, выраженными в других единицах физических величин, то над таблицей следует писать наименование преобладающего показателя и обозначение его физической величины, например, "Размеры в миллиметрах", "Напряжение в вольтах", а в подзаголовках остальных граф приводить наименование показателей и (или) обозначения других единиц физических величин в соответствии с рисунком 4.

Пример 2.2

Таблица 4.2 – Показатели переработки руды

Наименование продукта	Выход	Содержание меди	В процентах
			Извлечение меди
Медный концентрат	10	20	95
Отвальные хвосты	90	0,2	5

Интервалы чисел в тексте записывают со словами «от» и «до (имея в виду «От ... до ... включительно»)), если после чисел указана единица физической величины или



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

числа представляют безразмерные коэффициенты, или через дефис, если числа представляют порядковые номера. Например, «... толщина слоя должна быть от 0,5 до 20 мм...» или «...таблицы 27-12, рисунки 1-14...»

Для составления *формул* и уравнений используется редактор формул Microsoft Word. Уравнения и формулы следует выделять из текста в отдельную строку. Формулы размещают посередине строки. Для этого формулу выравнивают в тексте по правому краю, а затем абзацными отступами (кнопкой «Tab») смещают формулу от ее номера на середину строки. Выше и ниже каждой формулы должно быть оставлено по одной свободной строке. Пояснение значений символов и числовых коэффициентов следует приводить непосредственно под формулой в той же последовательности, в которой они даны в формуле. Первую строку пояснения начинают со слова «где» (без двоеточия после него).

Формулы следует нумеровать в пределах раздела арабскими цифрами в круглых скобках в крайнем правом положении на строке. Номер формулы состоит из номера раздела и порядкового номера формулы, разделенных точкой и заключенных в круглые скобки, например, (3.1). Номер формулы записывают на уровне формулы справа. Ссылки на формулы в тексте дают в скобках. Пример: «... в формуле (3.1)». Ниже приводится пример оформления формулы и ссылки на нее.

Пример 3

Будущая стоимость вклада (денежных средств), учитывающая фактор риска рассчитывается по формуле

$$S_R = P \times [(1 + A_n) \times (1 + RP_n)]^n, \quad (3.1)$$

где S_R – будущая стоимость вклада (денежных средств), учитывающая фактор риска, руб.;

P – первоначальная сумма вклада, руб.;

A_n – безрисковая норма доходности на финансовом рынке, выраженная десятичной дробью,

RP_n – уровень премии за риск по конкретному финансовому инструменту (финансовой операции), выраженной десятичной дробью;

n – количество интервалов, по которым осуществляется начисление процентов, единиц.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

В документе следует применять стандартизованные единицы физических величин, их наименования и обозначения в соответствии с ГОСТ 8.417.

Наряду с единицами СИ, при необходимости, в скобках указывают единицы ранее применявшихся систем, разрешенных к применению. Применение в одном документе разных систем обозначения физических величин не допускается.

Недопустимо отделять единицу физической величины от числового значения (переносить их на разные строки или страницы), кроме единиц физических величин, помещаемых в таблицах.

Материал, дополняющий основной текст диссертации, допускается помещать в *приложениях*. В качестве приложения могут быть представлены: графический материал, таблицы, формулы, карты, рисунки, фотографии и другой иллюстративный материал.

Приложения оформляют как продолжение работы на ее последующих страницах. Приложения должны иметь общую с остальной частью работы сквозную нумерацию страниц. Отдельный том приложений должен иметь самостоятельную нумерацию. В тексте диссертации на все приложения должны быть даны ссылки. Приложения располагают в порядке ссылок на них в тексте диссертации. Приложения должны быть перечислены в оглавлении диссертации с указанием их номеров, заголовков и страниц.

Каждое приложение следует начинать с новой страницы с указанием сверху по середине страницы слова «Приложение». Приложение должно иметь заголовок, который записывают отдельной строкой с прописной буквы по середине страницы.

Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ъ, Ы, Ь. После слова «Приложение» следует буква, обозначающая его последовательность. Если в отчете одно приложение, оно обозначается «Приложение А».

3.6.Списки.

При оформлении *нумерованных списков* в магистерской диссертации следует ограничиться тремя видами списков: нумерованным списком, содержащим в одном пункте несколько предложений, нумерованным списком, содержащим в одном



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

пункте одно предложение, двухуровневым нумерованным списком. Использование других нумерованных списков не рекомендуется.

Приведем нумерованный список, содержащий в одном пункте несколько предложений:

1. Каждый пункт нумерованного списка, содержащего в одном пункте несколько предложений должен начинаться с большой буквы и заканчиваться точкой.
2. Номер пункта данного списка выравнивается по левому краю без абзацного отступа. После номера ставится точка и делается отступ для написания текста.
3. Текст списка выравнивается по ширине. В тексте используются переносы. Левая граница второй и последующих строк внутри текста пункта выравниваются по первой букве текста первой строки.
4. Все первые буквы текста первых строк пунктов списка должны быть выровнены между собой.

Приведем пример нумерованного списка, содержащего в одном пункте одно предложение:

- 1) тест списка начинается с маленькой буквы и заканчивается точкой с за-пятой;
- 2) номер пункта списка выравнивается по левому краю с абзацным отступом;
- 3) после номера ставится точка и делается отступ для написания текста;
- 4) текст списка выравнивается по ширине;
- 5) в тексте используются переносы;
- 6) левая граница второй и последующих строк внутри текста пункта выравниваются по первой букве текста первой строки;
- 7) все первые буквы текста первых строк пунктов списка должны быть выровнены между собой;
- 8) последнее предложение оканчивается точкой.

Двухуровневый нумерованный список, используется при необходимости перечислений внутри списка, нумеруется прописными буквами русского алфавита за исключением букв «ё, з, й, о, ч, ь, ы, ь».

Приведем пример такого списка:

- a) тест списка начинается с маленькой буквы и заканчивается точкой с за-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

пятой;

- б) номер пункта списка выравнивается по левому краю с абзацным отступом; в) после номера ставится точка и делается отступ для написания текста; г) номер второго уровня выравнивается по тексту первого уровня;
 - 1) текст списка выравнивается по ширине;
 - 2) в тексте используются переносы;
 - 3) левая граница второй и последующих строк внутри текста пункта выравниваются по первой букве текста первой строки; д) все первые буквы текста первых строк пунктов списка должны быть выровнены между собой;
- е) последнее предложение оканчивается точкой.

Ненумерованные списки используются для перечислений. Обычно в таком списке один пункт, одно предложение.

Приведем пример такого списка:

- в качестве маркера списка используется тире;
- применение других типов маркеров недопустимо;
- остальное оформление списка аналогично нумерованному списку, содержащему в одном пункте одно предложение.

3.7. Библиографические ссылки.

Ссылки на использованные источники следует указывать порядковым номером библиографического описания источника в списке использованных источников. Порядковый номер ссылки заключают в квадратные скобки.

Список литературы является составной частью справочного аппарата исследования и служит источником библиографической информации о документах, использованных автором в ходе исследования. Ссылки на источники необходимо указывать во всех случаях рассмотрения, упоминания или цитирования в тексте диссертации других произведений. Они требуются для идентификации и поиска источников, на которые ссылается автор. Библиографические ссылки составляются на любые опубликованные и неопубликованные документы, в том числе представленные в электронном виде.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

В список литературы включаются те источники, которые использованы магистрантом при написании выпускной квалификационной работы. Включение в библиографический список тех источников, ссылки на которые по тексту отсутствуют, недопустимо. Общее количество источников в списке литературы должно быть не менее 35-50. Для качественного и всестороннего изучения рассматриваемых в работе вопросов магистранту необходимо использовать как отечественную, так и зарубежную литературу. В библиографическом списке, помимо классических учебных пособий и монографий должна обязательно присутствовать научная периодика, а также специализированные тематические ресурсы Интернет, использованные при проведении исследования.

Список литературы необходимо нумеровать арабскими цифрами и печатать с абзацного отступа. Библиографические записи необходимо располагать в следующем порядке:

- законодательные и нормативные акты (в порядке убывания юридической силы);
- иная литература (в порядке упоминания в тексте диссертации, либо в алфавитном порядке).

Ниже приводятся примеры оформления библиографических записей в списке литературы:

Статьи из журналов и сборников

1. Адорно Т. В. К логике социальных наук // Вопросы философии. - 1992. - № 10. - С. 76 – 86.

Монографии

2. Тарасова В. И. Политическая история Латинской Америки учеб. для вузов. - 2-е изд. - М.: Проспект, 2006. - 412 с.

Интернет-ресурсы



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

4. Рынок тренингов Новосибирска: своя игра [Электронный ресурс]. Режим доступа: <http://nsk.fdme.ru/news/2006/07/03/2121> (дата обращения: 17.10.2012) *Нормативно-правовые акты*

5. О противодействии терроризму: федер. закон Рос. Федерации от 6 марта 2006 г. №35-ФЗ: принят Гос. Думой Федер. Собр. Рос. Федерации 26 февр. 2006 г.: одобр. Советом Федерации Федер. Собр. Рос. Федерации 1 марта 2006 г. // Рос. газ. - 2006. - 10 марта.

6. Трудовой кодекс Российской Федерации [Электронный ресурс]: от 30.12.2001 г. № 197–ФЗ: [по состоянию на 22 нояб. 2011 г.] // Официальный интернет-портал правовой информации / Гос. система правовой информ. Режим доступа: <http://pravo.gov.ru/proxy/ips/?docbody=&prevcklink=1&&nd=102073987> (дата обращения: 11.12.2011).

Описание книг:

1. Книги одного автора

Патласов О. Ю. Антикризисное управление. Финансовое моделирование и диагностика банкротства коммерческой организации: учеб. пособие / О. Ю. Патласов. – М.: Книжный мир, 2009. – 509 с. – ISBN: 978-5-8041-0383-6.

2. Книги двух авторов

Еникеев М. И. Общая, социальная и юридическая психология: краткий энцикл. слов. / М. И. Еникеев, О. Л. Кочетков. – М.: Юридическая литература, 1997. – 447 с. – ISBN 5-7260-0891-X.

3. Книги трех авторов

Орехов В. И. Антикризисное управление: учеб. пособие / В. И. Орехов, К. В. Балдин, Т. Р. Орехова. – 2-е изд., испр. – М.: ИНФРА-М, 2012. – 540 с.

4. Более трех авторов

Основы антикризисного управления предприятиями: учеб. пособие для студ. высш. учеб. заведений / Н. Н. Кожевников [и др.]; под ред. Н. Н. Кожевникова. – 2-е изд., стер. – М.: Академия, 2007. – 496 с.

5. Книги, в которых автор не указан

В таких изданиях за косой чертой пишется фамилия редактора, составителя или другого ответственного лица.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Антикризисное управление: учеб. пособие / под ред. Э. М. Короткова. – М.: ИНФРА-М, 2010. – 240 с.

6. Стандарты

1. ГОСТ 7.53–2001. Издания. Международная стандартная нумерация книг. – Взамен ГОСТ 7–53–86; введ. 2002–07–01. – Минск: Межгос. совет по стандартизации, метрологии и сертификации; М.: Изд-во стандартов, 2002. – 3 с.

2. ГОСТ Р 517721–2001. Аппаратура радиоэлектронная бытовая. Входные и выходные параметры и типы соединений. Технические требования. – Введ. 2002–01–01. – М.: Изд-во стандартов, 2001. – IV, 27 с.: ил.; 29 см.

7. Диссертации и авторефераты

1. Вишняков И. В. Модели и методы оценки коммерческих банков в условиях неопределенности: дис. ... канд. экон. наук: 08.0013: защищена 12.02.02: утв. 24.06.02 / Вишняков Илья Владимирович. – М., 2002. – 234 с.

2. Махов А. Н. Разработка эффективной технологии сушки стеклонаполненных полиамидов: автореферат дис. ... канд. техн. наук: 05.17.06 / Махов Александр Николаевич. – Саратов, 2009. – 18 с. – Библиогр.: с. 18 (8 назв.).

8. Отчеты о научно-исследовательской работе

Формирование генетической структуры стада: отчет о НИР (промежуточ.): 42-44 / Всерос. науч.-исслед. ин-т животноводства; рук. Попов В.А.; исполн.: Алешин Г.П. [и др.]. – М., 2001. – 75 с. – Библиогр.: с. 72–74. – № ГР 01840051145. – Инв. № 04534333943.

9. Патентные документы

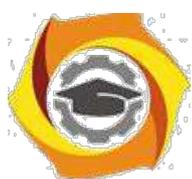
1. Пат. 2187888 Российская Федерация, МПК7 Н 04 В 1/38, Н 04 J 13/00. Приемопередающее устройство / Чугаева В.И.; заявитель и патентообладатель Воронеж. науч.-исслед. ин-т связи. – № 2000131736/09; заявл. 18.12.00; опубл. 20.08.02, Бюл. № 23 (II ч.). – 3 с. : ил.

2. Заявка 1095735 РФ, МПК7 В 64 G 1/00. Одноразовая ракета-носитель / Тернер Э. В. (США); заявитель Спейс Системз/Лорал, инк. – № 2000108705/28; заявл. 07.04.00; опубл. 10.03.01, Бюл. № 7 (I ч.). – 5 с. : ил.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

3. А. с. 1007970 СССР, МКИЗ В 25 J 15/00. Устройство для захвата неориентированных деталей типа валов / В. С. Ваулин, В. Г. Кемайкин (СССР). – № 3360585/25–08; заявл. 23.11.81; опубл. 30.03.83, Бюл. № 12. – 2 с. : ил



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Приложение 1 к Методическим рекомендациям
по оформлению и подготовке магистерской диссертации в
Негосударственном частном образовательном учреждении
высшего образования «Технический университет УГМК»

ТЕХНИЧЕСКОЕ ЗАДАНИЕ
на выполнение ВКР (магистерской диссертации)

(Наименование темы магистерской диссертации)

(Название предприятия)

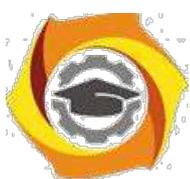
(ФИО магистранта)



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Содержание:

1.	ОБЩИЕ СВЕДЕНИЯ	26
1.1.	Формулировка производственной задачи	26
_____	26
_____	26
1.2.	Тема ВКР (магистерской диссертации)	26
1.3.	Наименование предприятия, заказчика	26
1.4.	Магистрант, исполнитель	26
1.5.	Куратор-эксперт от предприятия-заказчика	26
1.6.	Перечень документов, на основании которых разрабатывается ВКР, кем и когда утверждены эти документы	26
1.7.	Порядок оформления и предъявления Заказчику результатов работ	28
2.	ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ МАГИСТЕРСКОЙ ДИССЕРТАЦИИ	28
3.	ЦЕЛИ ВКР	28
_____	28
4.	ЗАДАЧИ ВКР	28
_____	28
5.	ТРЕБОВАНИЯ К ВКР	28
5.1.	Проведение анализа достижений науки, техники и технологии в области решения поставленной производственной задачи	29
_____	29
5.2.	Проведение анализа текущего состояния по теме ВКР	29
_____	29
5.3.	Проведение анализа публикаций по теме ВКР	29
_____	29
5.4.	Проведение анализа передового опыта по теме ВКР	29
_____	29
5.5.	Постановка целей, задач научного исследования по теме ВКР	29
_____	29
5.6.	Обоснование методов научного исследования по теме ВКР	29
_____	29
5.7.	Проведение экспериментальных исследований по теме ВКР	29
_____	29
5.8.	Разработка схемных, технических, технологических решений по теме ВКР ..	29
_____	29
5.9.	Обоснование выбора разработанных схемных, технических, технологических, организационных решений по теме ВКР	29
_____	29
6.	Требования к технико-экономическому обоснованию проекта (выбор и расчет соответствующих технико-экономических показателей)	7
7.	Требования к методике подтверждения результативности проекта	7



1. Общие сведения

Настоящий документ содержит перечень требований, которые должны быть учтены при разработке выпускной квалификационной работы (ВКР), являющейся решением производственной задачи, выданной предприятием магистранту на период обучения в магистратуре Технического Университета УГМК.

1.1. Формулировка производственной задачи

1.2. Тема ВКР (магистерской диссертации)

1.3. Наименование предприятия, заказчика

1.4. Магистрант, исполнитель

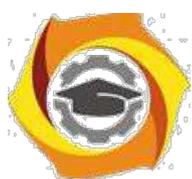
ФИО	Должность	Название предприятия (подразделения)
<hr/>	<hr/>	<hr/>

1.5. Куратор-эксперт от предприятия-заказчика

ФИО	Должность	Название предприятия (подразделения)
<hr/>	<hr/>	<hr/>

1.6. Перечень документов, на основании которых разрабатывается ВКР, кем и когда утверждены эти документы

Проект разрабатывается на основании требований к результатам освоения основной образовательной программе прикладной магистратуры «Управление и устойчивое развитие электроэнергетики предприятия» по направлению 13.04.02 Электроэнергетика и электротехника для руководителей энергетической службы предприятий, энергетиков подразделений, инженеров-электриков энергетических служб предприятий ОАО «УГМК», утвержденной 20.09.2018г



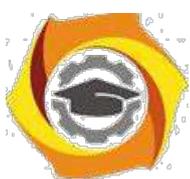
**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Плановые сроки начала и окончания работ по выполнению ВКР (магистерской диссертации)

Плановые сроки начала и окончания работ по выполнению ВКР (магистерской диссертации) приведены в таблице 1.

Таблица 1

№ ра-пп бот	Наименование работ	2019				2020								2021					2022											
		ноябрь	декабрь	январь	февраль	март	апрель	май	июнь	июль	август	сентябрь	октябрь	ноябрь	декабрь	январь	февраль	март	апрель	май	июнь	июль	август	сентябрь	октябрь	ноябрь	декабрь	январь	февраль	март
1.	Формулировка производственной задачи Подготовка																													
2.	по-дробного плана вы-полнения магистер-ской диссертации, формулировка темы ВКР Анализ																													
3.	существую-щего состояния по теме ВКР																													
4.	Анализ публикаций по теме ВКР																													
5.	Анализ передового опыта по теме ВКР																													
6.	Постановка целей, задач научного ис-следования по теме ВКР																													
7.	Разработка мето-дики исследования по теме ВКР																													
8.	Выполнение НИР по теме ВКР (тео-ретическая часть)																													
9.	Разработка ТЗ на выполнение маги-стерской диссертации Согласование ТЗ на выполнение маги-																													
10.	стерской диссертации с экспертом службы директора по энергетике																													
11.	Доклад о промежу-точных результатах выполнения маги-стерской диссертации перед эксперт-ной комиссией Холдинга.																													
12.	Выполнение НИР по теме ВКР (экспериментальная часть)																													
13.	Разработка 1 ВКР Раздела																													
14.	Разработка 2 ВКР Раздела																													



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

№ пп	Наименование работ	2019		2020												2021					2022									
		ноябрь	декабрь	январь	февраль	март	апрель	май	июнь	июль	август	сентябрь	октябрь	ноябрь	декабрь	январь	февраль	март	апрель	май	июнь	июль	август	сентябрь	октябрь	ноябрь	декабрь	январь	февраль	март
15.	Разработка Раздела 3 ВКР																													
16.	Разработка Раздела 4 ВКР																													
17.	Разработка раздела 5 ВКР																													
18.	Экспертиза проекта на предприятии																													
19.	Экспертиза в службе директора по энергетике																													
20.	Защита ВКР (магистерской диссертации)																													

1.7. Порядок оформления и предъявления Заказчику результатов работ

ВКР оформляется в виде магистерской диссертации и передается заказчику в установленные учебным планом сроки.

Приемка работ осуществляется экспертной комиссией предприятия.

Рецензия на магистерскую диссертацию готовится назначенным экспертом от дирекции по энергетике.

Защита магистерской диссертации осуществляется в НЧОУ ВО «Технический университет УГМК» перед Государственной экзаменационной комиссией, утвержденной приказом.

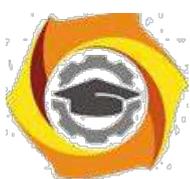
2. Основания для разработки магистерской диссертации

2.1. Производственная задача от предприятия на период обучения в магистратуре.

3. Цели ВКР

4. Задачи ВКР

5. Требования к ВКР



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

5.1. Проведение анализа достижений науки, техники и технологии в области решения поставленной производственной задачи

5.2. Проведение анализа текущего состояния по теме ВКР

5.3. Проведение анализа публикаций по теме ВКР

5.4. Проведение анализа передового опыта по теме ВКР

5.5. Постановка целей, задач научного исследования по теме ВКР

5.6. Обоснование методов научного исследования по теме ВКР

5.7. Проведение экспериментальных исследований по теме ВКР

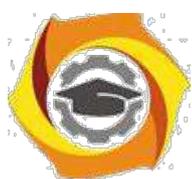
5.8. Разработка схемных, технических, технологических решений по теме ВКР

5.9. Обоснование выбора разработанных схемных, технических, технологических, организационных решений по теме ВКР

6. Требования к технико-экономическому обоснованию проекта (выбор и расчет соответствующих технико-экономических показателей)

7. Требования к методике подтверждения результативности проекта

СОСТАВИЛИ:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Наименование организации	Должность исполнителя	Фамилия И. О.	Подпись	Дата

СОГЛАСОВАНО:

Наименование организации	Должность исполнителя	Фамилия И. О.	Подпись	Дата

СОГЛАСОВАНО:

Эксперт от службы директора по энергетике ОАО «УГМК»	Должность	Фамилия И. О.	Подпись	Дата



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Приложение 2 к Методическим рекомендациям
по оформлению и подготовке магистерской диссертации в
Негосударственном частном образовательном учреждении
высшего образования «Технический университет УГМК»

Доклад и презентация

Доклад по результатам магистерской диссертации является публичным выступлением перед членами ГЭК и присутствующими на открытой защите.

По структуре доклад можно разделить на три части. Каждая часть представляет собой самостоятельный смысловой блок, хотя в целом они логически взаимосвязаны и отражают содержание проведенного исследования.

В начале доклада соискатель должен представить работу и научного руководителя: «Уважаемый председатель! Уважаемые члены государственной экзаменационной комиссии! Позвольте представить вашему вниманию результаты магистерской диссертации на тему: «...». Научный руководитель - ... ».

Первая часть доклада опирается на введение и первую главу диссертации. В этой части характеризуется актуальность выбранной темы и, на основе проведенного анализа предметной области, формулируется цель и задачи исследования.

Вторая, самая большая по объему часть, в последовательности, установленной логикой проведенного исследования, раскрывает и доказывает существенность каждого научного положения диссертационной работы, выносимого на защиту. Переход от первой части доклада ко второй может быть выражен фразой: «Разрешите остановиться на выдвигаемых положениях более подробно».

Заканчивается доклад заключительной частью, которая может начинаться следующим образом: «Суммируя вышесказанное, следует отметить, что научная новизна результатов исследования заключается в следующем: 1)..., 2)..., 3)...и т.д.». Таким образом, соискатель, не повторяя частные обобщения, сделанные ранее во время доклада, останавливает свое внимание на принципиальных отличиях полученных им научных результатов от тех, которые уже известны науке. Кроме этого, в заключительной части доклада следует отметить практическую значимость результатов исследования: «Практическая значимость диссертационного исследования заключается в...», а также апробацию диссертационной работы: «Основные результаты исследования одобрены... используются... рекомендованы...».

Доклад может заканчиваться словами: «Доклад окончен. Спасибо за внимание».

В общей сложности доклад должен составлять не более 10 минут. На бумажном носителе доклад, рассчитанный на 10 мин. занимает до 5 страниц текста, шрифт Times New Roman 14, полуторный интервал.

Презентация магистерской диссертации предназначена для сопровождения иллюстративным материалом доклада студента перед членами ГЭК. Презентация должна быть увязана с докладом. Оформление презентации рекомендуется выполнять с соблюдением корпоративного стиля, принятого в ООО «УГМК Холдинг», предприятия, Технического университета УГМК.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Приложение 3 к Методическим рекомендациям
по оформлению и подготовке магистерской диссертации в
Негосударственном частном образовательном учреждении
высшего образования «Технический университет УГМК»



**Негосударственное частное образовательное
учреждение высшего образования
«Технический университет УГМК»
Кафедра информационных технологий**

ДОПУСТИТЬ К ЗАЩИТЕ:
Зав. кафедрой, канд.пед.наук,
доцент

_____ Горбатов С.В.

« ____ » _____ 2024 г.

Иванов Иван Иванович

**РАЗРАБОТКА МОДЕЛИ РАЗВИТИЯ ИТ-ИНФРАСТРУКТУРЫ
ООО «МЕДНОГОРСКИЙ МЕДНО-СЕРНЫЙ КОМБИНАТ»**

МД — 39902925 - 130402 – 01 – 192 – 22

Выпускная квалификационная работа

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Профиль: «Информационные технологии в промышленности»

Научный руководитель
канд.физ.-мат.наук.

_____	_____
Подпись	ФИО
_____	_____

Консультант

Нормоконтролер

_____	_____
Подпись	ФИО
_____	_____

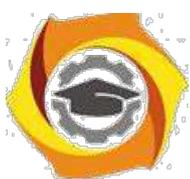
Куратор-эксперт

_____	_____
Подпись	ФИО
_____	_____

Студент гр. Эн-2310и

_____	_____
Подпись	ФИО

г. Верхняя Пышма 2024



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Приложение 4 к Методическим рекомендациям
по оформлению и подготовке магистерской диссертации в
Негосударственном частном образовательном учреждении
высшего образования «Технический университет УГМК»

**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет УГМК»**

Кафедра _____
Направление _____

УТВЕРЖДАЮ

Зав. кафедрой _____

« » _____ 202... г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

Студента _____ группы _____
(фамилия, имя, отчество)

1. Тема ВКР _____

_____ -

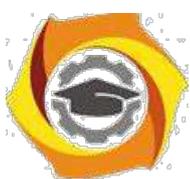
Утверждена приказом директора университета от « ____ » _____ 201... г. № _____

2. Научный руководитель _____
(ФИО, должность, ученое звание, ученая степень)

3. Исходные данные к ВКР _____
(полное наименование организации УГМК, перечень материалов необходимых для выполнения
работы, справочная, научная, методическая литература, ресурсы Интернет)

4. Содержание пояснительной записки (перечень подлежащих разработке вопросов):

5. Перечень демонстрационных материалов: _____



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

6. Консультанты по ВКР с указанием относящихся к ним разделов

Раздел	Консультант	Подпись, дата	
		задание выдал	задание принял

7. Календарный план

Наименование этапов выполнения ВКР	Сроки выполнения этапов ВКР	Отметка о выполнении

Научный руководитель _____
(подпись) _____ ФИО

Задание принял к исполнению _____
(подпись)

8. ВКР закончена « ____ » _____ 202.... г.

Пояснительная записка и все материалы просмотрены

Оценка консультантов: а) _____ б) _____
в) _____ г) _____

Считаю возможным допустить _____
к защите его выпускной квалификационной работы в Государственной экзаменационной
комиссии

Научный руководитель _____

9. Допустить _____ к защите вы-
пускной квалификационной работы в Государственной экзаменационной комиссии (протокол
экспертной комиссии № _____ от « ____ » _____ 202... г.)
(протокол заседания кафедры № _____ от « ____ » _____ 202. г.)

Зав. кафедрой _____



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Приложение 5 к Методическим
рекомендациям
**по оформлению и подготовке магистерской
диссертации в Негосударственном частном
образовательном учреждении высшего образования
«Технический университет УГМК»**

**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет УГМК»**

АННОТАЦИЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Ф.И.О.

выпускника _____

Направление подготовки _____

Название программы _____

Форма обучения _____

Тема выпускной квалификационной работы (ВКР) _____

Краткое описание содержания работы:

В первой главе содержится

Во второй главе

В третьей главе (при наличии)



Результат работы (основные выводы)

Иностранный язык: английский язык

Методические указания по практическим занятиям ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1

1. Цели занятия:

Повторить грамматический материал (порядок слов в английском предложении, времена группы Simple (Present, Past, Future) в действительном и страдательном залоге; типы вопросительных предложений: общий и альтернативный вопросы. Формировать навыки употребления грамматической структуры Simple Tenses в повествовательной и вопросительной формах в письменной речи. Развивать навыки использования речевого этикета в диалогической речи.

2. Рекомендации:

Изучить справочный материал [3] с.5 – 9, а также [1] с. 5 – 11, [2] с.87 – 98.

3. Порядок выполнения работы:

3.1. Прочитать и перевести текст «Education in Russia»; ответить на вопросы к тексту. Используя дополнительный раздаточный материал прочитать и перевести текст «MTUCI – my university».

3.2. Повторить грамматический материал: Времена группы Simple (Present, Past, Future) в действительном и страдательном залоге, утвердительная и отрицательная форма. Выполнить письменно упражнения см. п. 2 на тему Simple Tenses.

3.3. Повторить Типы вопросительных предложений: общий и альтернативный вопросы. Записать вопросительные предложения: общий и альтернативный вопросы к тексту.

3.4. Изучить фразы и выражения, используемые при знакомстве, согласно английскому рече-вому этикету. Составить мини-диалоги на тему «Знакомство».

4. Контрольные вопросы:

4.1. What is the national system of higher education?

4.2. What objects does the higher education have?

4.3. Is the education important for people? Why?

4.4. What is necessary to enter a higher educational institution?

4.5. What types of educational institutions are there in the Russian Federation?

4.6. How does MTUCI differ from the other Russian universities?

4.7. Рассказать об образовании и использовании времен группы Simple.

4.8. Назвать основные особенности речевого этикета, характерные для англоязычного населения и используемые при знакомстве.

5. Литература:



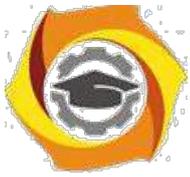
**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

11. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
12. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
13. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2

1. Цели занятия:

Повторить грамматический материала (порядок слов в английском предложении, времена группы Continuous (Present, Past, Future) в действительном и страдательном залоге. Форми-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ровать навыки употребления грамматической структуры Continuous Tenses в повествовательной и вопросительной формах в письменной речи. Развивать навыки ведения дискуссии.

2. Рекомендации:

Изучить справочный материал [1] с.11 – 18, 26 – 35, [2] с.99 – 107, [3] с.5 – 9.

3. Порядок выполнения работы:

3.1. Прочитать и перевести текст «Russia and the English speaking countries»; ответить на вопросы к тексту.

3.2. Повторить грамматический материал: Времена группы Continuous (Present, Past, Future) в действительном и страдательном залоге, утвердительная и отрицательная форма. Выполнить письменно упражнения см. п. 2 на тему Continuous Tenses. (*перевод с русского языка на английский, тренировочные упражнения*).

3.3. Провести дискуссию “вопрос-ответ” (разновидность простого собеседования, где применяется определенная форма постановки вопросов для собеседования с участниками дискуссии-диалога, позволяющая собеседникам перейти от желания говорить к пониманию индивидуальных интересов, состояний, отношений).

Тема дискуссии «Russia on the Political World Map». Использовать также общий и альтернативный типы вопросов при ведении дискуссии.

Правила ведения дискуссии: 1) Моя цель не в том, чтобы “победить”, а в том, чтобы прийти в наилучшему решению; 2) критиковать идеи, а не людей; 3) побуждать каждого из участников к тому, чтобы участвовать в обсуждении; 4) стремиться осмыслить и понять оба взгляда на проблему.

Чтобы сделать выступление кратким и ясным, воспользуйтесь следующей формулой:

1 – выразить свою позицию (в чем заключается ваша точка зрения) – Я считаю, что...(I think that..., I believe that..., To my mind...)

2 – обоснование (на чем вы основываетесь, довод в поддержку вашей позиции) - ...потому, что...(Because...)

3 – пример (факты, иллюстрирующие ваш довод) - ...например (for example..., for instance).

4 – следствие (вывод, что надо сделать, призыв к принятию вашей позиции) - ...поэтому...(That is why...,)

Все ваше выступление, таким образом, может состоять из двух-четырёх предложений и занимать 1-2 минуты.

4. Контрольные вопросы:

4.1. Speak about Russia as one of the largest countries of the world, its political and economical features (official name, geographical characteristics, political state).

4.2 What are the main English speaking countries?

4.3 Speak about Britain, its political and economical features (official name, geographical characteristics, political state).

4.4 Speak about the USA, its political and economical features (official name, geographical characteristics, political state).

4.2. Рассказать об образовании и использовании времен группы Continuous.

В 3. Назвать основные типы вопросов в английском языке. Назвать правила ведения дискуссии.

В Литература:

с Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

с Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

с З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3

1. Цели занятия:

Повторить грамматического материала (порядок слов в английском предложении, времена группы Simple, Continuous (Present, Past, Future) в действительном и страдательном залоге; типы вопросительных предложений: специальные и разделительный вопросы. Формировать навыки анализа текста. Развивать способности монологической речи.

2. Рекомендации:

Изучить справочный материал [1] с.121 – 127, [2] с.26 – 35.

3. Порядок выполнения работы:

- 3.1. Прочитать и перевести текст «Holidays, customs and traditions of the UK and the USA»; ответить на вопросы к тексту.
- 3.2. Выполнить письменный анализ текста: записать основную идею текста; разбить текст на логические части и озаглавить их.
- 3.3. Выполнить письменно упражнение 2.
- 3.4. Повторить типы вопросительных предложений: специальные и разделительный вопросы. Составить специальные (2) и разделительный (1) вопросы к тексту.
- 3.5. Рассказать о себе (Introducing myself) и ответить на вопросы одногруппников (ведение диалога на общие темы: My studies. My home town).

4. Контрольные вопросы:

- 4.1. What is the role of customs and traditions for the lives of different nations?
- 4.2. What common customs do Russia and other countries have?
- 4.3. What is the intercultural difference?
- 4.4. Рассказать о видах вопросительных предложений в английском языке.
- 4.5. Рассказать об особенностях, которые характеризуют порядок слов в вопросительных предложениях (в зависимости от типа вопроса).

5. Литература:

и Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

и Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

и З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4

1. Цели занятия:

Повторить грамматический материал (порядок слов в английском предложении, времена группы Perfect (Present, Past, Future) в действительном и страдательном залоге. Развивать навыки употребления грамматической структуры Perfect Tenses в повествовательной и вопросительной формах в письменной речи. Развивать навыки диалогической речи.

2. Рекомендации:

Изучить справочный материал [1] с. 18 – 35,[2] с.107 – 111, [3] с. 12 – 13.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

3. Порядок выполнения работы:

- 3.1. Прочитать и перевести текст «Progressive organizations and movements: the UNO»; ответить на вопросы к тексту.
- 3.2. Выполнить письменно упражнение 3.
- 3.3. Повторить типы вопросительных предложений: общий, альтернативный, специальные и разделительный вопросы. Составить мини-диалоги на тему «Progressive organizations and movements: their role in social and cultural lives of different nations».

4. Контрольные вопросы:

- 4.1. What progressive movements and organizations do you know?
- 4.2. What goals do they have?
- 4.3. What importance do they have for the world community?
- 4.4. What is the UNO?
- 4.5. What does the UNO's operation consist in?
- 4.6. Рассказать об образовании и использовании времен группы Perfect.

5. Литература:

- x Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
- x Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
- x З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5

1. Цели занятия:

Контролировать уровень усвоения учебного материала модуля. Тренировать навыки использования английского языка для межличностного общения в устной речи.

2. Порядок выполнения работы:

2.1. Проведение ролевой игры «International conference».

Конференция проходит в Москве, присутствуют представители различных государств. Председатель объявляет повестку дня «Role of progressive movements and organizations in solving global problems», предлагает внести свои предложения. Участники конференции выступают со своими докладами (2-4 минуты) об одной из прогрессивных организаций, движений, задают вопросы каждому докладчику, вносят свои дополнения. После окончания конференции председатель подводит итоги. В ходе конференции используются вопросы домашнего задания и соответствующие речевые формулы.

Студенты сами выбирают страну и организацию, которую хотят представить на конференции в виде доклада, готовят небольшой доклад, а также вопросы, которые они хотели бы задать каждой из стран-участников. Основные организации и движения: ООН, ЮНЕСКО, Армия спасения, Международный Красный Крест, Солдатские матери, Гринпис, Зеленые, Живая планета.

Один студент, по желанию, будет выполнять роль председателя, ведущего конференции и оценит доклад каждого участника. Его задача правильно начать, направлять ход конференции и закончить событие.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Рекомендуемые речевые формулы

Chairman	Председатель
Dear Friends!	Дорогие друзья!
May I take your attention, please?	Прошу вашего внимания.
We are glad to welcome ...	Мы рады приветствовать...
I declare our conference open	Я объявляю нашу конференцию открытой
As you know, the theme of our conference is...	Как вы знаете, тема нашей конференции...
According to the agenda we are to hear several reports.	В соответствии с повесткой дня нам предстоит заслушать несколько докладов.
I ask the speakers to be brief as our time is short.	Прошу докладчиков быть краткими, так как наше время ограничено.
Well, now I give the floor to the first speaker...	А сейчас я предоставляю слово первому докладчику...
Are there any questions to the speaker?	Есть ли какие-либо вопросы к докладчику?
Are there any additions, remarks or comments on the report?	Есть ли какие-либо добавления, замечания или комментарии по докладу?
We thank our speaker for his interesting report/presentation.	Благодарим нашего докладчика за его интересный доклад / выступление
Next we'll hear from our second speaker...	Следующим мы послушаем нашего второго докладчика...
Before declaring our conference closed, I'd like to say that it was held at a rather high/low level.	Перед тем, как объявить конференцию закрытой, мне бы хотелось сказать, что она прошла на довольно высоком/низком уровне.
All the reports/papers were of great interest.	Все доклады были очень интересны.
Among the most noteworthy reports/papers were those made by...	Среди докладов, заслуживающих наибольшего внимания, были те, которые сделали ...
Allow me to thank all our reporters/speakers once more and wish them further progress.	Позвольте мне еще раз поблагодарить всех наших докладчиков и пожелать им дальнейших успехов
And now I declare our conference closed	А сейчас я объявляю нашу конференцию закрытой
Speaker	Докладчик
In my report/paper I want/ I'd like to...	В своем докладе я хочу/ мне бы хотелось...
I think, I can answer your question	Я думаю, что смогу ... ответить на ваш вопрос
Sorry, I am not ready to answer your question now	Сожалею, я не готов пока ответить на ваш вопрос
I can't agree with you.	Я не могу согласиться с вами.
Audience	Аудитория
Could the speaker explain or tell us about...?	Не мог бы докладчик объяснить или рассказать нам о ...?
And what about...?	А как насчет...?
Are you sure that...?	Вы уверены, что...?
May I ask you a question?	Могу я задать вам вопрос?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 6

1. Цели занятия:

Изучить грамматический материал (Неличные формы глаголов: причастие настоящего времени, герундий, отглагольное существительное (Participle I, Gerund, Verbal Noun); развивать навыки употребления грамматических явлений в письменной речи. Формировать навыки анализа текста. Формировать у студентов умение использовать язык в межличностном общении.

2. Рекомендации:

Изучить справочный материал [1] с. 37 – 43, [2] с.120 -- 124, [3] с. 17 – 18.

3. Порядок выполнения работы:

3.1. Изучить грамматический материал: Неличные формы глаголов: причастие настоящего времени, герундий, отглагольное существительное (Participle I, Gerund, Verbal Noun). Выполнить упражнения 1 – устно, 2 – письменно.

3.2. Прочитать и перевести текст «International cooperation»; ответить на вопросы к тексту. Сформулировать основную идею текста, определить ключевые вопросы, затронуты в тексте.

3.3. Используя ключевые слова, записать 5 вопросов разных типов по теме, затронутой в тексте. Письменно прокомментировать утверждение «Nowadays distance is a great obstacle for various contacts between the Globe nations».

3.4. Построить и реализовать диалог на тему «In spite of great distance the Earth people are successfully cooperating in many fields», используя речевые обороты для выражения согласия / несогласия / сомнения / отказа.

4. Контрольные вопросы:

- 4.1. What purpose does the international cooperation have?
- 4.2. What is the international cooperation based on?
- 4.3. What functions does it perform?
- 4.4. What importance does this type of cooperation have for the world community?
- 4.5. How is the international cooperation realized?
- 4.6. Рассказать о неличных формах английского глагола.

5. Литература:

в Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

в Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

в З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 7

1. Цели занятия:

Закрепить грамматический материал. Формировать навыки использования английского языка в письменной речи. Формировать навыки анализа информации (на базе учебного текста). Формировать у студентов умение использовать язык в межличностном общении на социальные и культурные темы.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

2. Рекомендации:

Изучить справочный материал [1] с. 45 – 46, [2] с.126,129, [3] с. 19 – 20.

3. Порядок выполнения работы:

3.1. Письменно выполнить грамматические упражнения 4,6.

3.2. Прочитать и перевести текст «What is a computer?». Выполнить задания к тексту: установить соответствие предложений содержанию текста; сформулировать основную идею текста; ответить на вопросы к тексту. Используя ключевые слова, написать тезисы (**тезис** - это мысль, сформулированная в виде одного или нескольких предложений) по теме «Great inventions and their role for our civilization».

3.3. Подготовить и реализовать диалог по теме текста (использовать тезисы в качестве плана).

4. Контрольные вопросы:

4.1. What is a computer?

4.2. What does a computer system consist of?

4.3. What types of computers are there?

4.4. Назовите основные выражения речевого этикета, используемые для выражения совета, и ответные реплики.

5. Литература:

2 Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

3 Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

4 З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 8

1. Цели занятия:

Контролировать уровень усвоения учебного материала модуля. Тренировать навыки использования английского языка для межличностного общения.

2. Порядок выполнения работы:

2.1. Выполнить тест за модуль №1.

2.2. Изучить речевые обороты, используемые для выражения потребностей и рекомендаций.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 9

1. Цели занятия:

Закрепить грамматический материал. Формировать навыки использования английского языка в письменной речи. Формировать навыки анализа информации (на базе учебного текста). Формировать у студентов умение использовать язык в межличностном общении на социальные и культурные темы.

2. Рекомендации:

Изучить справочный материал [1] с. 45 – 46, [2] с.126,129, а также [3] с. 19 – 20.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

3. Порядок выполнения работы:

- 3.1. Письменно выполнить грамматические упражнения 2,6.
- 3.2. Прочитать и перевести текст «What is inside a PC system?». Выполнить задания к тексту: установить соответствие предложений содержанию текста; сформулировать основную идею текста; ответить на вопросы к тексту.
- 3.3. Подготовить устный развернутый ответ на один из контрольных вопросов.

4. Контрольные вопросы:

- 4.1. What is the CPU?
- 4.2. What does the CPU consist of?
- 4.3. What determines the power and performance of a computer?
- 4.4. What is the difference between RAM and ROM?
- 4.5. What is the purpose of buses and cards?

5. Литература:

4. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
5. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
6. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 10

1. Цели занятия:

Формировать навыки использования английского языка в письменной речи. Формировать навыки анализа информации (на базе учебного текста). Формировать у студентов умение использовать язык в межличностном общении на профессиональные темы. Формировать навыки восприятия иностранной речи на слух.

2. Рекомендации:

Изучить справочный материал и разобранные примеры (3) с. 21 – 23, (2) с. 120 – 126.

3. Порядок выполнения работы:

- 3.1. Прочитать и перевести текст «In a computer shop»; ответить на вопросы к тексту.
- 3.2. Прослушать диалог «Choosing the right computer»; выполнить задания к прослушиваемому материалу.
- 3.3. Выполнить письменно упражнения 4,5.

4. Контрольные вопросы:

- 4.1. List main important technical characteristics of a PC.
- 4.2. List questions people can need when buying a computer.
- 4.3. List the difference between a notebook, laptop, desktop PC.

5. Литература:

8. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
9. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

9. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 11

1. Цели занятия:

1.1. Контролировать уровень усвоения учебного материала модуля. Тренировать навыки использования английского языка для межличностного общения в устной речи.

1.2. Реализовать ролевую игру «In a shop» (buying a computer). Реализуйте диалог между продавцом магазина и клиентом, намеревающимся приобрести компьютер.

2. Порядок выполнения работы:

2.1. Проведение ролевой игры «In a shop (buying a computer)».

- игре студенты участвуют по очереди в парной работе. При этом один из студентов является продавцом специализированного магазина, другой – покупатель, который желает приобрести ПК для конкретных целей. роль продавца заключается в том, чтобы узнать от покупателя максимум информации о цели приобретения компьютера, способе его эксплуатации и т.п., чтобы порекомендовать наиболее подходящую для данного покупателя модель (ноутбук, планшет, планшетфон, ПК). В ходе игры используются вопросы домашнего задания и соответствующие речевые формулы.

Студенты готовят тезисы, а также вопросы, которые они могли бы использовать, исполняя ту или иную роль. Один студент, по желанию, будет выполнять роль наблюдателя (эту функцию может выполнять преподаватель) и оценит речевое поведение каждого участника. Задача студентов правильно начать, направлять ход диалога и закончить беседу результативно.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 12

1. Цели занятия:

Изучить грамматический материал: герундий, оборот *used to + infinitive*, особенностях использования в речи. Формировать навыки работы с текстом: анализ информации, написание резюме (на базе учебного текста). Формировать навыки письменной речи. Формировать навыки восприятия иностранной речи на слух.

2. Рекомендации:

Изучить справочный материал в [1] с. 51 – 55, (3) с. 30, [2] с. 115 – 118.

3. Порядок выполнения работы:

3.1. Прочитать и перевести текст «Interacting with your computer»; ответить на вопросы к тексту. Выполнить упражнения после текста.

3.2. Прослушать диалог «Speech recognition systems»; выполнить задания к прослушиваемому материалу.

3.3. Дать развернутый письменный ответ на тему «Role of Speech recognition systems now and in future».

4. Контрольные вопросы:

4.1. What are the ways to interact with a PC?

4.2. Which devices can be used to interact with a computer?

4.3. List functions and features of the most common input devices.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- 4.4. What are the benefits of speech recognition systems?
- 4.5. Who would benefit most from advances in speech recognition technology?
- 4.6. What is the future of this kind of technology?
- 4.7. Расскажите о герундии, обороте *used to + infinitive*, особенностях использования в речи.

5. Литература:

- Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
- Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
- З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 13

1. Цели занятия:

Изучить грамматический материал: сравнительная и превосходная степени сравнения прилагательных и наречий. Формировать навыки работы с текстом: анализ информации. Формировать навыки письменной речи. Формировать навыки восприятия иностранной речи на слух.

2. Рекомендации:

Изучить справочный материал в [1] с. 51 – 55, (3) с. 30, [2] с. 115 – 118.

3. Порядок выполнения работы:

- 3.1. Выполнить грамматические упражнения: 3 – устно, 4, 5 – письменно.
- 3.2. Прочитать и перевести текст «The eyes of a computer»; ответить на вопросы к тексту.
- 3.3. Прослушать диалог «Ergonomics»; выполнить задания к прослушиваемому материалу.
- 3.4. Дать развернутый письменный ответ по теме “Usefulness of ergonomics”.

4. Контрольные вопросы:

- 4.1. Which devices are called “the eyes of a computer”? Why?
- 4.2. What types of displays are there?
- 4.3. What are main characteristics of a display?
- 4.4. What is ergonomics?
- 4.5. What are its main principles?
- 4.6. Рассказать об образовании сравнительной и превосходной степеней сравнения прилагательных и наречий.

5. Литература:

- и Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
- и Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
- и З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 14

1. Цели занятия:

Изучить грамматический материал (Модальные глаголы и их эквиваленты (Modals and relatives), повелительное наклонение); развивать навыки употребления грамматических явлений ее письменной речи. Формировать навыки устной речи. Формировать навыки восприятия иностранной речи на слух.

2. Рекомендации:

Изучить справочный материал в [1] с. 55 - 65, [3] с. 30, [2] с. 115 – 118.

3. Порядок выполнения работы:

- 3.1. Выполнить грамматические упражнения: 10 – устно, 7,8 – письменно.
- 3.2. Прочитать и перевести текст «Types of magnetic drives. Optical storage: CDs and DVDs. Flash memory»; ответить на вопросы к тексту.
- 3.3. Подготовить и реализовать мини-диалоги по теме «Storage devices», используя обороты речевого этикета для выражения мнения / суждения / согласия / несогласия.
- 3.4. Дать развернутый письменный ответ на один из контрольных вопросов по теме текста.

4. Контрольные вопросы:

- 4.1. What types of magnetic drives are there?
- 4.2. What types of optical storage are there?
- 4.3. What is flash memory?
- 4.4. Рассказать о повелительном наклонении.
- 4.5. Расскажите о модальных глаголах, особенностях их использования в речи.

5. Литература:

В Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

В Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

В З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 15

1. Цели занятия:

Изучить грамматический материал (неопределённые местоимения *some, any, no*, особенности их употребления в речи); развивать навыки употребления грамматических явлений в письменной речи. Формировать навыки письменной речи, ведения переписки. Формировать навыки восприятия иностранной речи на слух.

2. Рекомендации:

Изучить справочный материал в [1] с. 55 - 65, [3] с. 30, [2] с. 115 – 118.

3. Порядок выполнения работы:

- 3.1. Выполнить грамматические упражнения: 1,3 – устно, 2 – письменно.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

3.2. Прослушать диалог «Describing flash drives»; выполнить задания к прослушиваемому материалу.

3.3. Прочитать и перевести текст электронного письма. Изучить правила оформления делового письма. Написать письмо-ответ, используя изученные речевые обороты.

3.4. Дать развернутый письменный ответ на один из контрольных вопросов по теме текста.

4. Контрольные вопросы:

4.1. What is the storage capacity of the flash drive?

4.2. How is it connected to the computer?

4.3. What are the advantages of a flash drive over a DVD or an external hard drive?

4.4. Рассказать о неопределённых местоимениях *some, any, no*. Особенности их употребления в речи.

4.5. Назвать основные правила написания делового письма.

5. Литература:

4. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

5. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

6. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 16

1. Цели занятия:

Контролировать уровень усвоения учебного материала модуля. Тренировать навыки монологической речи. Контролировать формирование английского языка как языка специальности при переводе текстов профессиональной направленности.

2. Порядок выполнения работы:

2.1. Выполнение теста за модуль №2.

2.2. Ответ устной темы «I/O Devices».

2.3. Ответ внеаудиторного чтения.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 17

1. Цели занятия:

Развивать умение логично и аргументированно строить письменную речь. Развивать навыки диалогической речи. Формировать навыки письменной речи, ведения переписки. Формировать навыки восприятия иностранной речи на слух.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72 ,[3] с 33 -- 35, [2] с. 135 – 136.

3. Порядок выполнения работы:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- 3.1. Прочитать и перевести текст «The functions of the operating system. GUI operating system»; ответить на вопросы к тексту, выполнить лексические задания к тексту.
- 3.2. Дать развернутый устный ответ на один из контрольных вопросов по теме текста.
- 3.3. Написать краткое изложение текста.
- 3.4. Прослушать диалог «Windows Vista»; выполнить задания к прослушиваемому материалу.

4. Контрольные вопросы:

- 4.1. What is an operating system? What types of OS are there?
- 4.2. What is GUI operating system?
- 4.3. Which Microsoft platform is used for pockets PCs, mobiles and portable media centres?
- 4.4. Describe the OS Windows Wista.
- 4.5. What are the advantages of WW?
- 4.6. Is WW still popular? Why?

5. Литература:

5. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
6. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
7. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 18

1. Цели занятия:

Изучить грамматический материал (слова с –ing окончанием: Participle I, Gerund, adjective); развивать навыки употребления грамматических явлений в устной и письменной речи. Развивать умение логично и аргументированно строить письменную речь.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 36 – 39.

3. Порядок выполнения работы:

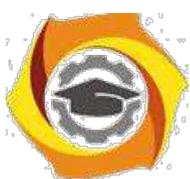
- 3.1. Выполнить грамматические упражнения: 1 – устно, 2, 3 – письменно.
- 3.2. Прочитать и перевести текст «Computer graphics»; ответить на вопросы к тексту.
- 3.3. Дать развернутый устный ответ на один из контрольных вопросов по теме текста.

4. Контрольные вопросы:

- 4.1. What types of graphics are described in the text?
- 4.2. Which graphics file format are mentioned?
- 4.3. What are benefits of using graphics?
- 4.4. What type of graphics is used to make maps or 3-D models?
- 4.5. Рассказать о словах с –ing окончанием: Participle I, Gerund, adjective.

5. Литература:

- 4) Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
- 5) Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

В.З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие
(Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 19

1. Цели занятия:

Закрепить грамматический материал (инфинитив, его формы, функции в предложении и способах перевода на русский язык); развивать навыки употребления грамматических явлений в устной и письменной речи. Развивать умение использовать английский язык в письменной речи. Формировать навыки диалогической речи.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 40 – 43.

3. Порядок выполнения работы:

- 3.1. Выполнить грамматические упражнения: 7 – устно, 9, 13 – письменно.
- 3.2. Прочитать и перевести текст «What is desktop publishing?»; ответить на вопросы к тексту.
- 3.3. Подготовьте устный развернутый ответ на вопрос What is desktop publishing?
- 3.4. Написать электронное неофициальное письмо.

4. Контрольные вопросы:

- 4.1. What types of software is used for the creation of DTP documents?
- 4.2. What are three differences between DTP software and word processor?
- 4.3. What is PDF and what can it do?
- 4.4. Which program is necessary to view a PDF document?
- 4.5. Рассказать о формах инфинитива, его функциях в предложении и способах перевода на русский язык.

5. Литература:

2. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
3. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
4. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 20

1. Цели занятия:

Закрепить грамматический материал (инфинитив, его формы, функции в предложении и способах перевода на русский язык); развивать навыки употребления грамматических явлений в устной и письменной речи. Развивать умение использовать английский язык в письменной речи. Формировать навыки диалогической речи. Формировать навыки восприятия иностранной речи на слух.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с.44 – 45.

3. Порядок выполнения работы:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- 3.1. Выполнить грамматические упражнения: 4 – устно, 5, 6 – письменно.
- 3.2. Прочитать и перевести текст «Multimedia magic»; ответить на вопросы к тексту.
- 3.3. Прослушать диалог «Components and system requirements»; выполнить задания к прослу-шиваемому материалу.
- 3.4. Подготовьте мини-диалоги-обсуждение по теме «Applications of multimedia».

4. Контрольные вопросы:

- 4.1. What is multimedia?
- 4.2. Where is multimedia used?
- 4.3. What opportunities does multimedia bring with it?
- 4.4. What is multimedia system?

5. Литература:

- 77Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
- 78Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
- 79З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 21

1. Цели занятия:

Закрепить грамматический материал (систему времен английского языка (English tenses)); развивать навыки употребления грамматических явлений в устной и письменной речи. Развивать умение читать и переводить тексты профессиональной направленности. Формировать навыки монологической речи (давать описание, характеристику).

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 46 – 47.

3. Порядок выполнения работы:

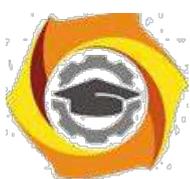
- 3.1. Выполнить грамматические упражнения: 5,7– письменно.
- 3.2. Прочитать и перевести текст «Internet FAQs. Email features»; ответить на вопросы к тексту, выполнить задания к тексту.
- 3.3. Подготовить мини-диалог-обсуждение по теме «Internet and its uses».

4. Контрольные вопросы:

- 4.1. What is the Internet? What are its origins?
- 4.2. What ways to get online are there?
- 4.3. How fast are today's Internet connections?
- 4.4 What does TCP/IP mean?
- 4.5. What are main Internet protocols?

5. Литература:

3. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
4. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

/ З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие
(Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 22

1. Цели занятия:

Изучить грамматический материал (словообразование: префиксы e- и cyber-). Развивать умение читать и переводить профессиональные тексты. Формировать навыки ведения переписки на английском языке. Формировать навыки логично и аргументированно трюить устную и письменную речь.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 48 – 49.

3. Порядок выполнения работы:

- 3.1. Выполнить грамматические упражнения: 4,5– письменно.
- 3.2. Прочитать и перевести тексты «Tour the collectives of Cyberspace»; ответить на вопросы к текстам, выполнить задания к тексту.
- 3.3. Прослушать диалог «E-commerce & on-line banking», выполнить задания к прослушиваемому материалу.
- 3.4. Напишите неофициальное письмо, содержащее предложение по использованию технологии MPLS для коммерческого предприятия.
- 3.5. Дать устный развернутый ответ на один из вопросов, данных к учебному тексту.

4. Контрольные вопросы:

- 4.1. What cyberspace is described in the text?
- 4.2. What communications possibilities does the Internet provide?
- 4.3. What is e-commerce? Which precautions should be met?
- 4.4. What is online banking? Which precautions should be met when using online banking?
- 4.5. Рассказать о словообразовании: префиксы e- и cyber-.

5. Литература:

- / Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
- / Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
- / З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 23

1. Цели занятия:

Контролировать уровень усвоения учебного материала модуля. Тренировать навыки монологической речи.

2. Порядок выполнения работы:

- 2.1. Выполнение теста за модуль №1 (2 семестр).
- 2.2. Ответ устной темы «Multimedia».



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 24

1. Цели занятия:

Изучить грамматический материал (синтаксическая конструкция «for+n/p+toV», способы ее перевода на русский язык); развивать навыки употребления грамматических явлений в устной и письменной речи. Формировать навыки использования английского языка в устной и письменной речи.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 50 – 52.

3. Порядок выполнения работы:

3.1. Выполнить грамматические упражнения: 1 – устно, 2,3 – письменно.

3.2. Прочитать и перевести текст «Virtual meetings.Chat & conferencing»; выполнить задания и ответить на вопросы к тексту.

3.3. Дать развернутый письменный ответ на один из контрольных вопросов.

3.4. Подготовить развернутый устный ответ по теме «Internet as a virtual room for communicating».

4. Контрольные вопросы:

4.1. Which virtual meetings are mentioned in the text?

4.2. Why is videoconferencing so useful for virtual workgroups?

4.3. Which technology enables people to make phone calls over the Internet?

4.4. What is the difference between web chat rooms and Instant Messaging?

4.5. Рассказать о способах перевода предложений с конструкцией «for+n/p+toV» на русский язык.

5. Литература:

/ Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

/ Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

/ З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 25

1. Цели занятия:

Повторить грамматический материал (пассивный залог английских времен); развивать навыки употребления грамматических явлений в устной и письменной речи. Формировать навыки восприятия иностранной речи на слух. Развивать навыки системно анализировать информацию (на базе учебного текста). Формировать навыки использования английского языка в письменной речи. Формировать навыки монологической речи.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с.54 – 56.

3. Порядок выполнения работы:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- 3.1. Прочитать и перевести текст «Security and privacy of the Internet»; выполнить задания к тексту и ответить на вопросы.
- 3.3. Прослушать диалог «Safety online for children», выполнить задания к прослушиваемому материалу.
- 3.3. Написать несколько тезисов по услышанной, прочитанной информации (тезис - это мысль, сформулированная в виде одного или нескольких предложений).
- 3.4. Подготовить развернутое устное сообщение по теме «The Internet and children: pros and cons».

4. Контрольные вопросы:

- 4.1. Why is security so important on the Internet?
- 4.2. How can we protect our personal data in the computer?
- 4.3. What is a virus? How can it enter a computer system?
- 4.4. Who is a hacker?
- 4.5. Which risks are there for children on the Internet?
- 4.6. Which solutions are developed to improve security for children online?
- 4.7. Рассказать о формах пассивного залога.

5. Литература:

- / Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
- / Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
- / З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 26

1. Цели занятия:

Повторить грамматический материал (инфинитив, его функции, употребление в предложении); развивать навыки употребления грамматических явлений в письменной речи. Формировать навыки анализа информации, написание резюме (на базе учебного текста). Тренировать диалогическую речь. Формировать навыки восприятия иностранной речи на слух.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с.56 – 57.

3. Порядок выполнения работы:

- 3.1. Выполнить грамматические упражнения: 1,3– письменно.
- 3.2. Прочитать и перевести текст «Computer languages»; выполнить задания к тексту и ответить на вопросы к тексту, сформулировать основную идею текста. Написать резюме текста. Резюме - это краткое изложение текста. В резюме содержится основная идея текста и определенные ключевые слова и фразы, доказывающие и поясняющие её. Для того чтобы написать резюме нужно научиться работать с текстом.

Существуют определенные правила, которые позволяют правильно написать резюме, так чтобы при этом текст сохранил свой смысл, а резюме приобрело краткую форму.

Правила написания резюме текста:

В Внимательно прочитайте текст и попытайтесь его осмыслить. Это позволит вам составить общее представление о тексте, сформулировать его главную идею.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- Разделите текст на логические абзацы. Для этого найдите ключевые фразы, которые разделяют текст на логические части.

\endash Упростите текст: определите ключевые слова, попробуйте их заменить синонимами, так чтобы при этом не исказить смысл текста; сократите длинные предложения, разбив на не-сколько коротких; уберите лишние слова (эпитеты, союзы, наречия), которые не имеют смысловой важности для текста.

\endash Постарайтесь рассказать весь текст в несколько предложениях. Если сложно сократить весь текст сразу, тогда можно делать это по абзацам. Каждый логический абзац сокращайте по-степенно до нескольких фраз.

\endash Прочитайте текст и удостоверьтесь, что вы ничего не упустили.

3.3. Прослушать текст «Steps in programming», выполнить задания к прослушиваемому материалу.

4. Контрольные вопросы:

- 4.1. What is a computer language?
- 4.2. Which types of computer languages are there?
- 4.3. What is the function of an assembler?
- 4.4. What is the difference between a compiler and an interpreter?
- 4.5. List the steps of programming.
- 4.6. Рассказать об инфинитиве: функции, употребление в предложении.

5. Литература:

3. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
4. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
5. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 27

1. Цели занятия:

Повторить грамматический материал (слова с окончанием –ed: Participle II, Past Simple); развивать навыки употребления грамматических явлений в письменной речи. Формировать навыки анализа информации (на базе учебного текста). Тренировать способность логично и аргументированно строить устную речь.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с.59 – 60.

3. Порядок выполнения работы:

- 3.1. Выполнить грамматические упражнения: 1,3– устно, 4 – письменно.
- 3.2. Прочитать и перевести текст «Java Applets. The Java language»; сформулировать главную идею текста, ответить на вопросы к тексту.
- 3.3. Прослушать текст «The history of Java».
- 3.4. Написать тезисы по услышанной информации.
- 3.5. Подготовить устный развернутый ответ по вопросу «What is Java?»

4. Контрольные вопросы:

- 4.1. What is Java?
- 4.2. What are main characteristics of Java language?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- 4.3. Why is Java popular?
- 4.4. Which alternatives to Java exist?
- 4.4. What did you know about the history of development of the Java language?
- 4.5. Рассказать о словах с окончанием –ed: Participle II, Past Simple.

5. Литература:

4. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
5. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
6. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 28

1. Цели занятия:

Изучить грамматический материал (сослагательное наклонение, употребление Present и Past Subjunctive); развивать навыки употребления грамматических явлений в письменной речи. Формировать навыки анализа информации (на базе учебного текста). Тренировать способность использовать английский язык для межличностного общения на профессиональные темы.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 61– 63.

3. Порядок выполнения работы:

- 3.1. Выполнить грамматические упражнения: 1 – устно, 3 4 – письменно.
- 3.2. Прочитать и перевести текст «The conversion of symbolic languages»; ответить на вопросы к тексту, разделить текст на логические части, озаглавить части.
- 3.3. Провести дискуссию “вопрос-ответ” (разновидность простого собеседования, где применяется определенная форма постановки вопросов для собеседования с участниками дискуссии-диалога, позволяющая собеседникам перейти от желания говорить к пониманию индивидуальных интересов, состояний, отношений).

Тема дискуссии « Necessity of conversion of symbolic languages».

Правила ведения дискуссии: 1) Моя цель не в том, чтобы “победить”, а в том, чтобы прийти к наилучшему решению; 2) критиковать идеи, а не людей; 3) побуждать каждого из участников к тому, чтобы участвовать в обсуждении; 4) стремиться осмыслить и понять оба взгляда на проблему.

Чтобы сделать выступление кратким и ясным, воспользуйтесь следующей формулой:

1 – выразить свою позицию (в чем заключается ваша точка зрения) – Я считаю, что...(I think that..., I believe that..., To my mind...)

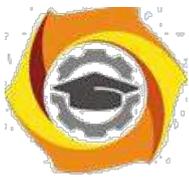
2 – обоснование (на чем вы основываетесь, довод в поддержку вашей позиции) - ...потому, что...(Because...)

3 – пример (факты, иллюстрирующие ваш довод) - ...например (for example..., for instance).

4 – следствие (вывод, что надо сделать, призыв к принятию вашей позиции) - ...поэтому...(That is why...,)

Все ваше выступление, таким образом, может состоять из двух-четырех предложений и занимать 1-2 минуты.

4. Контрольные вопросы:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- 4.1. What is conversion of symbolic languages?
- 4.2. What results does the conversion have?
- 4.3. Рассказать о сослагательном наклонении. Как употребляются Present и Past Subjunctive?

5. Литература:

2. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
3. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
4. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 29

1. Цели занятия:

Закрепить грамматический материал (Условные предложения); развивать навыки употребления грамматических явлений в письменной речи. Тренировать способность логично и аргументированно строить письменную речь. Формировать у студентов умение использовать язык в межличностном общении на профессиональные темы.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 64 – 65.

3. Порядок выполнения работы:

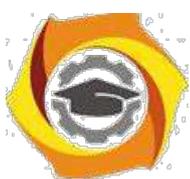
- 3.1. Выполнить грамматические упражнения: 1 – устно, 2, 3 – письменно.
- 3.2. Прочитать и перевести текст «Running the computer program. Testing the computer program»; ответить на вопросы к тексту.
- 3.3. Прочитать статью на английском языке; написать краткое изложение статьи на английском языке.
- 3.4. Дать устный развернутый ответ по теме «Types and purposes of computer programs», используя контрольные вопросы в качестве плана ответа.

4. Контрольные вопросы:

- 4.1. Which computer programs make computer operate?
- 4.2. What is a control program?
- 4.3. What is language processor?
- 4.4. What are the service programs?
- 4.5. Which errors/bugs may a computer program have?
- 4.6. What tools are there to debug the program?

5. Литература:

- №Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
- №Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
- №З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 30

1. Цели занятия:

Контролировать уровень усвоения учебного материала модуля. Тренировать навыки монологической речи. Контролировать формирование английского языка как языка специальности при переводе профессиональных текстов.

2. Порядок выполнения работы:

- 2.1. Выполнение теста за модуль №2 (2 семестр).
- 2.2. Ответ устной темы «The Internet as a means of communication».
- 2.3. Презентация на тему «Problems of security and privacy of the Internet».

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 31

1. Цели занятия:

Повторить грамматический материал (Пассивная форма временных групп Simple, Perfect, Continuous, модальных глаголов); развивать навыки употребления грамматических явлений в письменной речи. Тренировать способность логично и аргументированно строить устную речь. Формировать у студентов умение использовать язык в межличностном общении на профессиональные темы. Формировать навыки восприятия иностранной речи на слух.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 67 – 69.

3. Порядок выполнения работы:

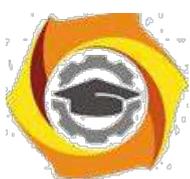
- 3.1. Выполнить грамматические упражнения: 1 – устно, 2, 3 – письменно.
- 3.2. Прочитать и перевести текст «ICT systems. Channels of communication»; выполнить задания, ответить на вопросы к тексту.
- 3.3. Прослушать текст «VoIP Technology»; выполнить задания к прослушиваемому материалу. Записать тезисы по услышанной информации.
- 3.4. Дать устный развернутый ответ на один из контрольных вопросов.

4. Контрольные вопросы:

- 4.1. What does ICT stand for?
- 4.2. What ICT systems do you know?
- 4.3. What are telecommunications?
- 4.4. What are mobile communications?
- 4.5. What does VoIP stand for?
- 4.6. How is VoIP technology realized?

5. Литература:

15. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
16. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
17. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.



ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 32

1. Цели занятия:

Тренировать навыки монологической речи, публичных выступлений.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 62 – 69.

3. Порядок выполнения работы:

- 3.1. Провести дискуссию «ответ-вопрос» по теме «Advantages of VoIP Technology».
- 3.2. Представить презентацию по теме «Описание и обсуждение мобильных устройств» (моб. телефон, смартфон, планшет, планшетфон и др.).

4. Литература:

3. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 33

1. Цели занятия:

Изучить грамматический материал (Синтаксические конструкции с неличными формами глаголов: независимый причастный оборот (Complexes with the verbals: Absolute participial construction)); развивать навыки употребления грамматических явлений в письменной речи. Тренировать способность логично и аргументированно строить устную речь.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 68.

3. Порядок выполнения работы:

- 3.1. Прочитать и перевести тексты «A letter of application», «A CV»; ответить на вопросы к тексту.
- 3.2. Прослушать диалоги «IT professionals», «A job interview».
- 3.3. Подготовить устный развернутый ответ на один из контрольных вопросов.
- 3.4. Написать CV.

4. Контрольные вопросы:

- 4.1. What is a letter of application?
- 4.2. What is a CV?
- 4.3. What are common professions in the ICT field?
- 4.4. Which qualities and abilities are important for those jobs?
- 4.5. What are the most widespread questions an interviewer can ask at a job interview?

5. Литература:

6. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

7. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 34

1. Цели занятия:

Закрепить грамматический материал (Фразовые глаголы); развивать навыки употребления грамматических явлений в письменной речи. Тренировать способность логично и аргументированно строить устную речь. Формировать навыки восприятия иностранной речи на слух. Формировать у студентов умение использовать язык в межличностном общении на профессиональные темы.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с.56 – 57.

3. Порядок выполнения работы:

- 3.1. Выполнить грамматические упражнения: 5 – письменно. Самостоятельно составить и записать 10 предложений с фразовыми глаголами.
- 3.2. Прочитать и перевести текст «Networks. Networking FAQs»; выполнить задания, ответить на вопросы к тексту.
- 3.3. Прослушать диалог «Small networks», выполнить задания к прослушиваемому материалу.
- 3.4. Дать устный развернутый ответ по теме «Networking».

4. Контрольные вопросы:

- 4.1. What is a computer network?
- 4.2. What types of networks do you know?
- 4.3. What are the benefits of using networks?
- 4.4. What does PAN stand for?
- 4.5. What is the difference between architecture and topology of a network?
- 4.6. What are small networks?
- 4.7. What elements can a small network consist of?

5. Литература:

8. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
9. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
10. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 35

1. Цели занятия:

Контролировать уровень усвоения учебного материала модуля. Тренировать навыки монологической речи.

2. Порядок выполнения работы:

- 2.1. Ответ устной темы «Information and communication technologies».
- 2.2. Выполнение теста за модуль №1 (3 семестр).



ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 36

1. Цели занятия:

Тренировать способность логично и аргументированно строить устную речь. Формировать у студентов умение использовать язык в межличностном общении на специальные темы в устной и письменной речи.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 56 – 57.

3. Порядок выполнения работы:

- 3.1. Прочитать и перевести текст «Fast Ethernet. TCP/IP»; выполнить задания к тексту, ответить на вопросы к тексту.
- 3.2. Дать устный развернутый ответ на один из контрольных вопросов.

4. Контрольные вопросы:

- 4.1. What is Ethernet?
- 4.2. How fast can Ethernet be?
- 4.3. Where is Ethernet used?
- 4.4. What is TCP/IP?
- 4.5. What is TCP/IP used for?
- 4.6. How does TCP/IP work?

5. Литература:

В Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

В Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

В З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 37

1. Цели занятия:

Тренировать способность логично и аргументированно строить устную и письменную речь. Формировать у студентов умение использовать язык в межличностном общении на специальные темы в устной и письменной речи. Формировать навыки восприятия иностранной речи на слух.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 54 – 55.

3. Порядок выполнения работы:

- 3.1. Прочитать и перевести тексты «Wireless networking», «Broadband communications»; выполнить задания, ответить на вопросы к тексту.
- 3.2. Написать аннотацию текста «Wireless networking».
- 3.3. Дать устный развернутый ответ по теме «Networking».



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

3.4. Дать развернутый письменный ответ на один из контрольных вопросов.

4. Контрольные вопросы:

- 4.1. What is wireless networking? What is the purpose of wireless networks?
- 4.2. How is wireless networking realized?
- 4.3. List examples of wireless networks.
- 4.4. What cable alternatives are there for networking?
- 4.5. What is broadband communications?
- 4.7. Which telecommunications services can be provided through broadband communications?

5. Литература:

ООО Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

ООО Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ООО З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 38

1. Цели занятия:

Повторить грамматический материал: способы передачи будущего времени (Future Simple, Future Perfect, Present Continuous, to be going to). Тренировать способность логично и аргументированно строить устную речь. Формировать у студентов умение использовать язык в межличностном общении на специальные темы в устной и письменной речи. Формировать навыки восприятия иностранной речи на слух.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 53 – 54.

3. Порядок выполнения работы:

- 3.1. Выполнить грамматические упражнения: 4 – письменно.
- 3.2. Прочитать и перевести текст «Future trends of technologies»; выполнить задания, ответить на вопросы к тексту.
- 3.3. Прослушать диалог «RFID tags», выполнить задания к прослушиваемому материалу.
- 3.4. Подготовить и провести дискуссию по теме «Future trends of technologies».

4. Контрольные вопросы:

- 4.1. What are the main trends of new technologies developments?
- 4.2. What is Artificial Intelligence? Where do experts suppose to use it?
- 4.3. What is biometrics? Where can it be applied?
- 4.4. What is pervasive computing?
- 4.5. Which advances are expected in the telecoms industry?
- 4.6. What does RFID stand for?
- 4.7. How can RFID tags be used? Where and Why?
- 4.8. What are active RFID tags?
- 4.9. What is the purpose of RFID chips?

5. Литература:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

2. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
3. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
4. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 39

1. Цели занятия:

Тренировать способность логично и аргументированно строить устную речь. Формировать у студентов умение использовать язык в межличностном общении на специальные темы в устной и письменной речи. Формировать навыки восприятия иностранной речи на слух.

2. Рекомендации:

Изучить справочный материал в [1] с. 65 – 72, (3) с. 52 – 53.

3. Порядок выполнения работы:

- 3.1. Прочитать и перевести текст «Architecture of future information systems»; ответить на вопросы к тексту.
- 3.2. Написать конспект текста.
- 3.3. Дать устный развернутый ответ по теме «Architecture of future information systems», используя контрольные вопросы в качестве плана ответа.

4. Контрольные вопросы:

- 4.1. What are the causes to change architecture of the information systems?
- 4.2. What is the concept of future information systems?
- 4.3. What kind of architecture is proposed for future information systems?
- 4.4. Which problems should be overcome?
- 4.5. Which features must future information systems possess?

5. Литература:

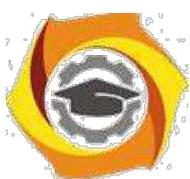
6. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
7. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
8. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 40

1. Цели занятия:

Тренировать способность логично и аргументированно строить устную речь. Формировать способность анализировать информацию. Формировать у студентов умение использовать язык в межличностном общении на специальные темы в устной и письменной речи.

2. Рекомендации:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Изучить справочный материал в [1] с. 65 – 72, (3) с. 51 – 52.

3. Порядок выполнения работы:

- 3.1. Прочитать и перевести текст «The future of information technology»; выполнить задания, ответить на вопросы к тексту.
- 3.2. Прослушать диалог «Healthcare», выполнить задания к прослушиваемому материалу.
- 3.3. Дать устный развернутый ответ по теме «How can technologies influence the people's life?», используя контрольные вопросы в качестве плана ответа.
- 3.4. Прочитать статью на русском языке, написать резюме статьи на английском языке.

4. Контрольные вопросы:

- 4.1. Which spheres of human's life will be affected by information technologies in future?
- 4.2. Can IT influence the way people do their work?
- 4.3. Will the information infrastructure be changed in future?
- 4.4. How can the Internet be changed in future?
- 4.5. In what ways is the telecoms and IT industry bad for the environment?
- 4.6. In what ways could the telecoms and IT industry be good for the environment?
- 4.7. How could user equipment be made greener?
- 4.8. Do you think digital technology has improved society?
- 4.9. How do you think technology could help society in the next 50 years?

5. Литература:

7. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
8. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
9. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 41

1. Цели занятия:

Контролировать сформированность компетенции (проверить готовность студентов к итоговому контролю полученных знаний (экзамен)): контролировать уровень усвоения учебного материала дисциплины; контролировать формирование английского языка как языка специальности при переводе профессиональных текстов; контролировать навыки монологической речи.

2. Порядок выполнения работы:

- 2.1. Ответ устной темы «Future prospects of IT».
- 2.2. Выполнение итоговой контрольной работы.
- 2.3. Ответ внеаудиторного чтения.



Заочная форма обучения

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1

1. Цели занятия:

Повторить грамматический материал (порядок слов в английском предложении), введение нового материала (функции глаголов to be, to have, to do). Формировать у студентов навыки использования английского языка для межличностного общения в социально-повседневных ситуациях.

2. Рекомендации:

Изучить справочный материал [3] с.5 – 9, а также [1] с. 5 – 11, [2] с.7 – 8.

3. Порядок выполнения работы:

- 3.1. Используя речевые обороты, составьте диалог «Знакомство» (парная работа).
- 3.2. Повторить основные аспекты грамматики английского языка. Изучить функции глаголов to be, to have, to do. Выполнить грамматический анализ следующих предложений:
 - к This is a device working on a direct current.
 - к Noise is a serious problem for radio communication practically of any kind.
 - к The scientists had completed his research by the beginning of the conference.
 - к Many people have an opportunity to use computers.
 - к We have to connect this wire to the plate.

4. Контрольные вопросы:

- 4.1. Какова специфика речевого поведения англоговорящих людей?
- 4.2. Назовите основные типы оборотов речевого этикета.
- 4.3. Назовите основные речевые обороты, используемые для знакомства и ответных реплик.
- 4.4. Назовите речевые обороты, используемые для приветствия и выражения, его сопровождающие.
- 4.5. Назовите некоторые вопросы, используемые для поддержания диалога (ice-breaking questions).
- 4.6. Назовите специфику порядка слов в английском предложении.
- 4.7. Каков порядок слов в простом (утвердительном и отрицательном) предложении?
- 4.8. Как образуется отрицательная форма глагола?
- 4.9. Каков порядок слов в вопросительном предложении?
- 4.10. С чего начинается перевод предложений с английского языка на русский?

5. Литература:

10. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
11. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
12. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.



ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2

1. Цели занятия:

Формировать у студентов английский язык как язык специальности, формировать навыки технического перевода. Развивать навыки речевого этикета и диалогической речи. Тренировать способность логично и аргументированно строить устную речь. Формировать у студентов умение использовать язык в межличностном общении на специальные темы в устной и письменной речи.

2. Рекомендации: Изучить справочный материал [1] с. 18 – 35, [2] с. 10 – 11, [3] с. 3 – 10.

3. Порядок выполнения работы:

3.1. Прочтите текст, выпишите выражения, указывающие на неформальный стиль общения.
3.2. Перевести предложения, сделать отрицательную и вопросительную формы данных предложений.

1. A gas laser is to perform numerous jobs in various spheres of our life.
 2. SMS is one of the most used services in second-generation mobile telecommunications systems.
 3. The laboratory have developed a new receiving machine.
 4. The laying of underground transmission lines is being constantly increased.
 5. Electric charges are acted upon by forces when they move in the magnetic field.
- 3.3. Подготовить мини-диалоги по теме «Знакомство» (парная работа).

4. Контрольные вопросы:

- 4.1. Назовите основные правила неофициального общения в устной форме.
- 4.2. Назовите основные виды неформального общения в письменной форме (чат, мессенджер), особенности его оформления.
- 4.3. Каков порядок слов в утвердительном, отрицательном и вопросительном предложениях?
- 4.4. Назовите основные правила неофициального общения в устной форме.
- 4.5. Назовите основные виды неформального общения в письменной форме (чат, мессенджер), особенности его оформления.
- 4.6. Каков порядок слов в утвердительном, отрицательном и вопросительном предложениях?
- 4.7. Как спрягаются глаголы to be, to have, to do во временах группы Simple?
- 4.8. Какие значения глагола to do вы знаете?
- 4.9. В каком случае глаголы to be и to have приобретают значение долженствования?
- 4.10. Когда глагол to be используется в качестве вспомогательного глагола?
- 4.11. Когда глагол to have используется в качестве вспомогательного глагола?
- 4.12. Какие ещё значения глаголов to be, to have вы знаете?

5. Литература:

1. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
2. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
3. Практика английского языка. Методическое пособие для студентов-заочников -38с. Ростов-на-Дону: СКФ МТУСИ, 2016.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3

1. Цели занятия:

Формировать у студентов английский язык как язык специальности, формировать навыки технического перевода. Развивать навыки речевого этикета и диалогической речи. Формировать навыки письменной речи на повседневные и профессиональные темы.

2. Рекомендации: Изучить справочный материал в [1] с. 19 – 20, [2] с.10 – 14.

3. Порядок выполнения работы:

3.1. Прослушать диалог. Записать фразы, используемые для уточнения информации, отказа, выражения благодарности.

3.2. Составить диалог по одной из устных тем.

4. Контрольные вопросы:

4.1. Назовите основные правила неофициального общения в устной и письменной форме.

4.2. Назовите основные речевые обороты, используемые для выражения:

а) согласия / несогласия;

б) мнения / намерения;

в) уточнения информации;

г) отказа / разрешения.

5. Литература:

1. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

2. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

3. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

4. Практика английского языка. Методическое пособие для студентов-заочников -38с. Ростов-на-Дону: СКФ МТУСИ, 2016.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4

1. Цели занятия:

Формировать у студентов английский язык как язык специальности, формировать навыки технического перевода. Развивать навыки речевого этикета и диалогической речи. Формировать навыки письменной речи на повседневные и профессиональные темы.

2. Рекомендации:

Изучить справочный материал в [1] с. 55 - 65, [3] с. 18 - 27.

3. Порядок выполнения работы:

3.1. Прочитать и перевести текст по специальности. Составить вопросы к тексту.

3.2. Записать вопросительные предложения по одной из изученных тем. Подготовить ответы на вопросы, используя изученные речевые обороты. Подготовить мини-диалоги по теме.

4. Контрольные вопросы:

4.1. Назовите основные правила официального общения в устной форме (диалог, монолог).



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- 4.2. Назовите речевые обороты, используемые для выражения согласия / несогласия / сомнения / отказа.
- 4.3. Назовите виды вопросительных предложений и способы их образования.
- 4.4. Как образуются времена группы Continuous в действительном и страдательном залоге?
- 4.5. Как образуются времена группы Perfect в действительном и страдательном залоге?
- 4.6. Как переводится глагол-сказуемое во временах групп Continuous и Perfect?
- 4.7. Какие есть неличные формы глаголов в английском языке?

5. Литература:

1. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
2. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
3. Практика английского языка. Методическое пособие для студентов-заочников -38с. Ростов-на-Дону: СКФ МТУСИ, 2016.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5

1. Цели занятия:

Формировать у студентов английский язык как язык специальности, формировать навыки технического перевода. Развивать навыки речевого этикета и диалогической речи. Формировать навыки письменной речи на повседневные и профессиональные темы.

2. Рекомендации:

Изучить справочный материал в [1] с. 15 - 18, [2] с. 25 – 28.

3. Порядок выполнения работы:

- 3.1. Прочитать и перевести аутентичный текст (заранее подготовленный студентами в качестве домашнего чтения).
- 3.2. Представить устный рассказ на одну из тем, данных в самостоятельной работе. Ответить на вопросы по теме.
- 3.3. Подготовить мини-диалоги на базе заготовленных устных тем.
- 3.4. Записать вопросительные предложения по одной из изученных тем. Подготовить ответы на вопросы, используя изученные речевые обороты.
- 3.5. Подготовить мини-диалоги по теме (групповая работа: формирование решения технической проблемы, обсуждение способов ее решения).

4. Контрольные вопросы:

- 4.1. Назовите основные правила официального общения в устной форме (диалог, монолог).
- 4.2. Назовите речевые обороты, используемые для выражения мнения / совета / предложения.

5. Литература:

1. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
2. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
3. Практика английского языка. Методическое пособие для студентов-заочников -38с. Ростов-на-Дону: СКФ МТУСИ, 2016.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 6

1. Цели занятия:

Проверить готовность студентов к итоговому контролю полученных знаний (экзамен): повторить грамматический материал; закрепить умения перевода и анализа англоязычных текстов; закрепить способность аргументировано и ясно строить устную и письменную речь.

2. Рекомендации:

Изучить справочный материал в [1] с. 55 - 65, [3] с. 22 - 27.

3. Порядок выполнения работы:

- 3.1. Выполнить рубежный тест на изученный материал.
- 3.2. Составить диалог с использованием изученных речевых оборотов. Написать отчет о проделанной работе и принятом решении (на основе проблемы, обсужденной в диалоге).
- 3.3. Просмотреть грамматический материал контрольной работы. Выполнить перевод и анализ предложений (1 - 10), данных для закрепления материала (стр. 22, Практика английского языка. Методическое пособие для студентов-заочников технических специальностей).

4. Контрольные вопросы:

- 4.1. Какие группы времён английского глагола вы знаете?
- 4.2. Какие синтаксические конструкции вы знаете?
- 4.3. Какие синтаксические конструкции с неличными формами глагола вы знаете?
- 4.4. Как наличие конструкции в предложении влияет на его перевод?
- 4.5. Какие есть особенности перевода синтаксических конструкций на русский язык?
- 4.6. Какие значения могут иметь слова *it, one, that*?
- 4.7. Какие усилительные конструкции вы знаете?

5. Литература:

1. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
2. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
3. Практика английского языка. Методическое пособие для студентов-заочников -38с. Ростов-на-Дону: СКФ МТУСИ, 2016.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 7

1. Цели занятия:

Формировать у студентов английский язык как язык специальности, формировать навыки технического перевода. Развивать навыки речевого этикета и диалогической речи. Формировать навыки письменной речи на повседневные и профессиональные темы.

2. Рекомендации:

Изучить справочный материал в [1] с. 55 - 65, [2] с. 27 – 29.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

3. Порядок выполнения работы:

- 3.1 Прочитать и перевести деловое письмо. Выписать фразы, используемые в письме для обращения, комплементарной концовки, прощания. Выписать слова и словосочетания, указывающие на официальный стиль письма.
- 3.2. Написать официальное письмо, используя деловое письмо как образец.
- 3.3. Прочитать и перевести неофициальное письмо. Указать отличия официального и неофициального стилей корреспонденции. Написать ответ на данное письмо, используя изученные фразы и выражения.
- 3.4. Прослушать диалог. Записать фразы, используемые для уточнения информации, отказа, выражения благодарности.
- 3.5. Ответ внеаудиторного чтения, устной темы.

4. Контрольные вопросы:

- 4.1. Назовите основные правила официального общения в устной форме (диалог, монолог).
- 4.2. Назовите основные правила официального общения в письменной форме.
- 4.3. Каковы правила ведения переписки на английском языке?
- 4.4. Назовите основные виды делового письма (деловое письмо, служебная записка, отчет), особенности его оформления.
- 4.5. Какие группы времён английского глагола вы знаете?
- 4.6. Какие синтаксические конструкции с неличными формами глагола вы знаете? Как наличие конструкции в предложении влияет на его перевод?

5. Литература:

1. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.
2. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.
3. Практика английского языка. Методическое пособие для студентов-заочников -38с. Ростов-на-Дону: СКФ МТУСИ, 2016.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 8

1. Цели занятия:

Проверить готовность студентов к итоговому контролю полученных знаний (зачет): повторить грамматический материал; закрепить умения перевода и анализа англоязычных текстов; закрепить способность аргументировано и ясно строить устную и письменную речь.

2. Рекомендации:

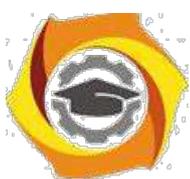
Изучить справочный материал в [1] с. 55 - 65, [3] с. 3-35.

3. Порядок выполнения работы:

- 3.1. Выполнить перевод технического текста с английского на русский язык.
- 3.2. Прочитать статью на русском языке. Написать аннотацию русского текста на английском языке.

4. Контрольные вопросы:

- 4.1. Назовите основные правила написания аннотации



Информатика

Методические рекомендации

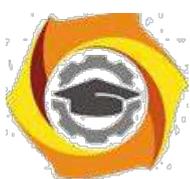
Введение

Дисциплина «Информатика» относится к математическому и естественнонаучному циклу, для изучения этой дисциплины необходимы знания из области высшей математики и умения использовать средства интегрированной системы программирования.

Сдача работы включает два этапа:

в Предварительно работа сдается в ЦОКР и регистрируется. Преподаватель проверяет правильность ее оформления и выполнения и делает соответствующую запись о допуске или не допуске к защите на титульном листе. В случае не допуски студент принимает меры для устранения недостатков и замечаний и повторно сдает контрольную работу. Если до начала зачета/экзамена контрольная работа не получила допуска к защите или не была защищена, то студент к сдаче зачета/экзамена на допускается.

в Если контрольная работа допущена к защите – студент должен во время защиты по требованию преподавателя пояснить, как он выполнял те или иные пункты задания. Такая проверка проводится для выяснения самостоятельного выполнения контрольной работы обучаемым. Если студент не может пояснить определенные этапы выполнения своей работы – он получает незачет.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Выбор варианта

Контрольная работа №2 включает выполнение четырёх задач. Ниже, приведены задачи, из которых студент должен выбрать четыре в соответствии с двумя последними цифрами номера зачетной книжки

Последние цифры номера зачетной книжки	Номер задачи/ Вариант
01, 16, 31, 46, 61, 76, 91	1/1, 2/1, 3/1, 4/1
02, 17, 32, 47, 62, 77, 92	1/2, 2/2, 3/2, 4/2
03, 18, 33, 48, 63, 78, 93	1/3, 2/3, 3/3, 4/3
04, 19, 34, 49, 64, 79, 94	1/4, 2/4, 3/4, 4/4
05, 20, 35, 50, 65, 80, 95	1/5, 2/5, 3/5, 4/5
06, 21, 36, 51, 66, 81, 96	1/6, 2/6, 3/6, 4/6
07, 22, 37, 52, 67, 82, 97	1/7, 2/7, 3/7, 4/7
08, 23, 38, 53, 68, 83, 98	1/8, 2/8, 3/8, 4/8
09, 24, 39, 54, 69, 84, 99	1/9, 2/9, 3/9, 4/9
10, 25, 40, 55, 70, 85	1/10, 2/10, 3/10, 4/10
11, 26, 41, 56, 71, 86	1/11, 2/11, 3/11, 4/11
12, 27, 42, 57, 72, 87	1/12, 2/12, 3/12, 4/12
13, 28, 43, 58, 73, 88	1/13, 2/13, 3/13, 4/13
14, 29, 44, 59, 74, 89	1/14, 2/14, 3/14, 4/14
15, 30, 45, 60, 75, 90	1/15, 2/15, 3/15, 4/15

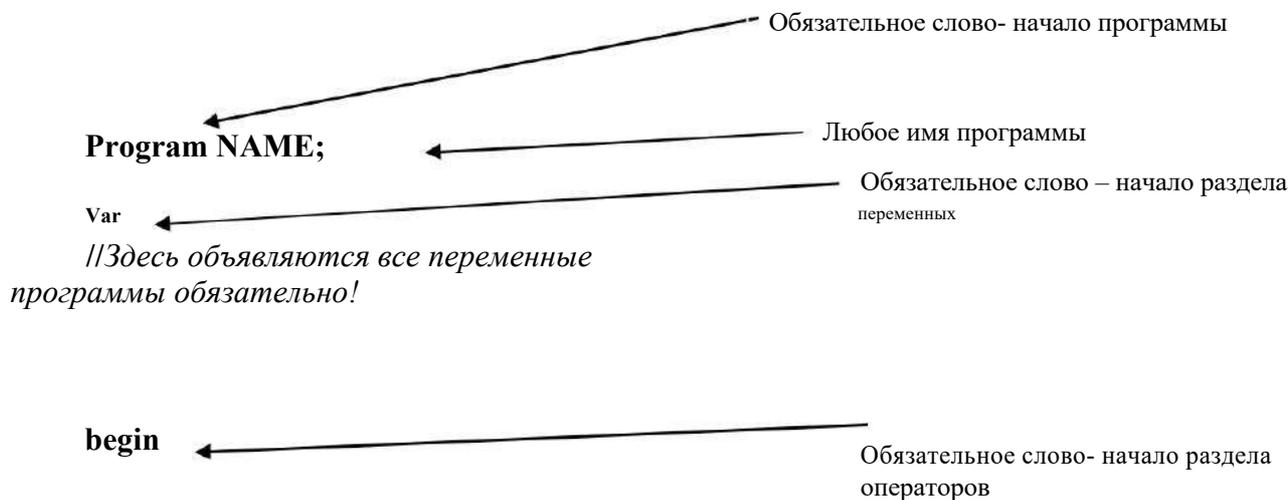
Общие сведения о языке Pascal

Для выполнения заданий контрольной работы подойдет любая сборка языка Pascal, которую можно легко найти в Интернете [3]. Это свободно распространяемое ПО.

Все примеры в данных методических указаниях приведены с использованием Pascal ABC [3].

Программа в языке Pascal состоит из нескольких обязательных частей (разделов).

Чтобы программа была работоспособной в ней обязательно должно быть:





Негосударственное частное образовательное учреждение высшего образования «Технический университет»

5 Здесь записывают исполняемые операторы, или выводят данные и т.п.

которые что-то вычисляют, вводят

end.

Обязательное слово- конец раздела операторов (**обязательно ставится точка**)

Это минимальный шаблон любой Pascal программы.

Ввод данных

7. настоящей контрольной работе необходимо будет данные вводить с клавиатуры, т.е. набирать необходимые значения и нажимать «Enter».

Для этого используются 2-а оператора: «Писать» и «Читать». Например, необходимо задать значения $z = 0.3$ с клавиатуры. Для этого пишем в разделе операторов:

Begin

write('z=');

Напишем на экран **z=**, куда потом будем вводить число

read(z);

Набираем на клавиатуре число **0.3** (**обязательно через точку!**) и нажимаем Enter. После этого число присваивается переменной z.

end.

10. результате, набранное на клавиатуре число присваивается переменной z. Физически это значение записывается в ячейку оперативной памяти, которой присвоено наименование z.

Вывод данных

Результаты решения каждой задачи, а также какие то промежуточные значения необходимо будет выводить на экран (говорят еще «распечатать»). Для этого есть оператор, который в переводе с английского, означает «писать».

Предположим, нужно вычислить значение функции $y = \sin x$ (обратите внимание, что это математическая запись функции!) и напечатать на экран ее значение (все это конечно же нужно выполнять в разделе исполняемых операторов!):

begin

y:=sin(x);
writeln("y=", y);

end.

Оператор **writeln('y=', y);** трактуется так: написать на экран текст «y=» и значение переменной y, затем переместить курсор на следующую строку. Тем самым, следующее выводимое на экран значение, если это будет указано в программе, будет выведено на строку ниже, чем значение функции y, т.е. вывод будет происходить «в столбик». Это потому, что в



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

конец оператора указано **ln-** переход на следующую строку. Если этот префикс отсутствует, то дальше вся информация будет записываться «в строку».

Возведение в степень

Операции возведения в степень в Паскале нет. Для выполнения этой операции можно использовать стандартные функции экспоненты и натурального логарифма- $\exp(x)$ и $\ln(a)$, воспользовавшись соотношением:

$$= a^n = e^{\ln a^n}, \text{ тогда } y = \exp(n * \ln(a)).$$

10. операторах программы эта строчка выглядит так:

```
y:=exp(n*ln(a));
```

Циклы

Циклические конструкции нужны для того, чтобы делать повторяющиеся вычисления, т.е. вычислять что-то более одного раза.

Например, нужно вычислить значение функции $z=\cos x$ (это, снова, математическая запись функции, которая отличается от записи ее в программе!) для x , изменяющегося на отрезке от -1.2 до 14.8 (или отрезок может быть задан так: [-1.2; 14.8]) с шагом 2.4.

Можно, конечно, написать в программе несколько одинаковых строчек для вычисления функции z , пользуясь следующим правилом:

Задаем начальное значение x

Вычисляем z

Печатаем z на экран

Добавляем к x шаг

Вычисляем z

Печатаем z на экран

Добавляем к x шаг

Вычисляем z

Печатаем z на экран

- и т.д.

Но тогда программа превратится в большое количество одинаковых строчек и будет трудно читаема. Чтобы в программе была простая узнаваемая структура вводят такую конструкцию как **ЦИКЛ**.

Любой цикл это описание задания: от какого и до какого значения и что вычислять.

Т.е. любой цикл имеет заголовок (предписание), где указывается какую переменную (называется «**переменная цикла**») изменять от какого и до какого значения (или указывается условие для нее) и с каким шагом изменять переменную цикла. И «тело» цикла, где происходят сами вычисления:

ЗАГОЛОВОК ЦИКЛА

begin

ТЕЛО ЦИКЛА



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

end;

Видно, что тело цикла взято в «операторные скобки», где `begin` –открывающая скобка, `end;`- закрывающая. Это делается для того, чтобы условия, указанные в заголовке цикла действовали для всех операторов, указанных в операторных скобках. В противном случае, заголовок цикла будет действовать только на первый следующий за ним оператор и будет казаться, что все условия указаны верно, а цикл не считает.

Для приведенного примера словесная формулировка цикла может быть следующей:

Присваиваем переменной X начальное значение -1.2 (левая граница интервала);
До тех пор, пока X меньше или равен 14.8 (правая граница интервала) делать

`begin`

Вычисляем значение Z при указанном X;
Увеличиваем X на величину шага: $X+2.4$;
Печатаем Z на экран;

`end;`

В операторах программы это будет выглядеть так:

`begin`

`x:= -1.2;`

`while (x<=14.8) do`

`begin`

`z:=cos(x);`
`writeln('z=',z);`
`x:=x+2.4;`
`end;`

`end.`

Заголовок цикла

Тело цикла

Обратите внимание, что жирным шрифтом выделен раздел исполняемых операторов программы.

**Контрольное задание №1
по теме «Программирование алгоритмов линейной структуры»**

Задание:

- написать на языке Паскаль программу для вычисления функции в соответствии с вариантом, округлить полученный результат, выделить целую часть полученного результата;
- выполнить программу, подставив значения исходных данных в соответствии с вариантом задания.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

**Методические требования
выполнению контрольного задания №1:**

и предусмотреть в программе объявление типов всех используемых переменных: переменные с именами i, j, k, l, m, n объявить данными целого типа, остальные переменные - вещественного типа с одинарной точностью;

и предусмотреть ввод значений исходных данных с клавиатуры по запросу, обеспечив при этом вывод подсказки;

и предусмотреть в программе вывод исходных данных и результаты внутри рамки из произвольных символов в центре экрана, обеспечив перед этим его очистку;

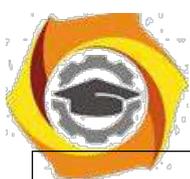
и для округления числа (в таблице 1 эта операция обозначается символами " $()$ ") использовать встроенную функцию **ROUND**;

и для выделения целой части числа (отсекания дробной части числа) использовать встроенную функцию **TRUNC**. В таблице 1 эта операция обозначается символами " $[]$ ";

и после выполнения программы и проверки полученного результата сделать скриншоты окна среды программирования с введенным текстом программы и скриншот окна запущенной программы.

Таблица 1 - варианты задания №1 по теме «Программирование алгоритмов линейной структуры»

Вариант	Формулы для вычислений	Исходные данные	
		X	Y
1.	$z = \frac{pi}{2} - \sqrt{2x} - \frac{x+y^2}{0.75tg x+y }$ $m = (z)$ $n = [z]$	13.003	-4.408
2.	$d = \frac{xy^2 - \sqrt{ x^2 - 2.5 * 10^{-3} y }}{2 \sin xy}$ $k=(d)$ $L=[d]$	1.3802	-1.9
3.	$f = 120 \frac{\lg(x+y)}{1}$ $i = f$ $j = [f]$ $x - 0.45 \sin(x - 8y)$	12.675	6.9
4.	$a = 0.8 \cdot 10^{-5} (xe^{-x(y-1.2)} yx)^3$ $i = (a)$ $j = [a]$	82.578	1.4517
5.	$R = \sqrt[4]{\frac{2x+y}{0.1x}} * \cos(\sin^2 x)$ $i = (R)$ $j = [R]$	5.422	-0.1537



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

6.	$c = \frac{(y - \pi) \cos \frac{\pi}{4}}{1 + \ln 1 - y } + \sqrt[3]{\frac{xy}{1 + 0.3x}}$	$i = (c)$	$j = [c]$	13.737	11.560
7.	$t = \cos \frac{\pi * \sin^2(3x - 5y)}{5 * 3.5(x + \pi)}$	$i = (t)$	$j = [t]$	4.72	21.35
8.	$t = \cos \frac{\pi * \sin^2(x - 8y)}{7 * 2.7(x - \pi)}$	$i = (t)$	$j = (t)$	3.59	17.53
9.	$d = \frac{(1 - e^{xy})^2}{0.7 \lg 1 - x^2 }$,	$m = (d)$	$n = [d]$.	1.674	-0.533
10.	$h = \frac{xy + \sin x}{ 1 - y * \ln x}$	$k = (h)$	$n = [h]$	32.01	-0.4917
11.	$U = \frac{xe^{xy} + 4 \cos^2 x}{ x(x - y)(2x + y) }$	$m = (u)$	$n = [u]$	1.472	-0.4342
12.	$b = \sqrt[3]{\frac{x + y}{0.2x}} * \sin(\operatorname{tg}^2 x)$	$i = (b)$	$j = [b]$	8.402	-0.22226
13.	$S = 0.910^{-4} (ye^{-x(y-0.8)} + y^2 x)^3$	$m = (S)$	$n = [S]$	75.447	1.2328
14.	$b = \frac{0.72 * \operatorname{tg} x + y }{x^2 + y} - \sqrt{2x} + \pi/2$	$k = (b)$	$L = [b]$	11.872	1.4517
15.	$a = 3 \lg(x + y) + x - \frac{1}{0.68 * \sin(x - 7y)}$	$i = (a)$	$j = [a]$	12.345	7.5

Пример отчета по заданию №1.

Задание №1. Программирование алгоритмов линейной структуры. Математические вычисления. Задание, формула для вычислений, исходные данные приведены в таблице 1.

Таблица 1 – Исходные данные варианта 1

Вариант	Формулы для вычислений	Исходные данные	
1	$t = \cos \frac{\pi * \sin^2(x - 8y)}{7 * 2.7(x - \pi)}$ $i = (t)$ $j = (t)$	8.19	7.23



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

ее Листинг программы на языке Pascal представлен в листинге 1.

Листинг 1 - Программирование алгоритма линейной структуры

```
Program test1_1;  
VAR i,j: integer;  
c,x,y: real;  
Begin  
Writeln ('Введите числа x и y (для разделения чисел используйте пробел, для подтверждения ввода - Enter:'); ReadLn (x,y);  
  
c:=(y-pi)*cos(pi/4)/(1+ln(1-y))+ ln(x*y)/(1+0.3*x);  
...  
End.
```

В Блок-схема программы 1 приведена на рисунке 1.

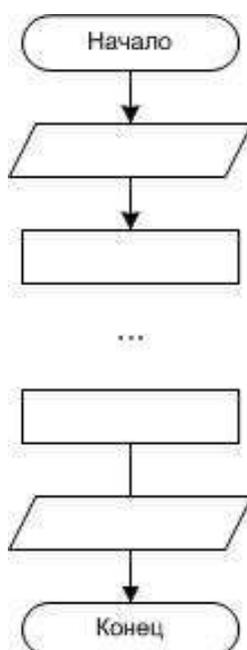


Рисунок 1 – Программа алгоритма линейной структуры

7. Скриншот окна среды программирования с введенным текстом программы представлен на рисунке 2.

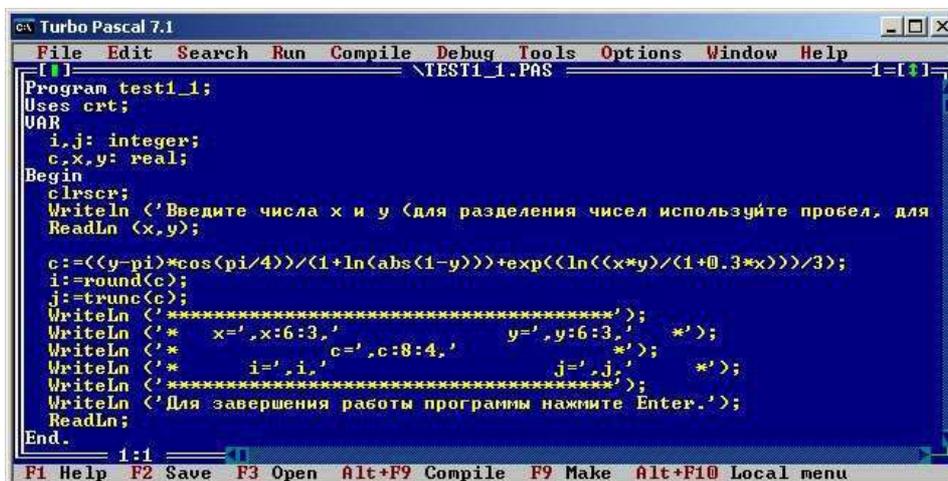


Рисунок 2 – Программа алгоритма линейной структуры



8. Скриншот окна запущенной программы представлен на рисунке 3.

```
C:\ Turbo Pascal 7.1
Введите числа x и y (для разделения чисел используйте пробел, для подтверждения
ввода - Enter):
12.737 10.560
*****
* x=12.737 y=10.560 *
* c= 4.6432 j=4 *
*****
Для завершения работы программы нажмите Enter.
```

Рисунок 3 – Программа алгоритма линейной структуры

Контрольное задание №2 по теме «Программирование алгоритмов разветвляющейся структуры»

Задание:

- б) написать на языке Паскаль программу для вычисления функции в соответствии с вариантом;
- 7) подготовить наборы x , y , a, b, c, d для тестирования всех ветвей программы;
- 8) выполнить программу, подставив все подготовленные наборы данных для тестирования.

Методические требования выполнению контрольного задания №2:

В предусмотреть в программе объявление типа всех используемых переменных, считая все их переменными вещественного типа;

В предусмотреть ввод значений переменных x , y , z , входящих в наборы исходных данных для тестирования, с клавиатуры по запросу программы;

В при выполнении домашнего задания подготовить наборы изменяемых x , y и неизменяемых a , b , c , d переменных исходных данных для тестирования всех ветвей программы;

В предусмотреть в программе вывод исходных данных и результатов выполнения программы в произвольном формате;

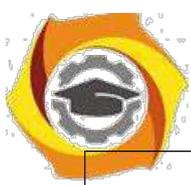
В после выполнения программы и проверки полученного результата сделать скриншоты: окна среды программирования с введенным текстом программы и окна запущенной программы.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

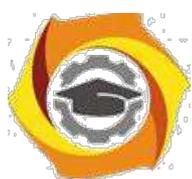
Таблица 2 - варианты задания №2 по теме «Программирование алгоритмов разветвляющейся структуры»

Вариант	Формулы для вычислений и условия их реализации
1.	$t = \begin{cases} \max(3^{x+a}, c+d, \min(x, a)), & \text{если } x > 3 \text{ и } xy < 2 \\ \min(x, a, c, y, d), & \text{если } x > 3 \text{ и } xy > 2 \\ C + dx, & \text{в противном случае} \end{cases}$
2.	$G = \begin{cases} 4 \min(5^{2+x}, x+y, -4y), & \text{если } 2 < x+y < 3 \\ \max(e^{x-y}, 3 \ln x, \min(x, y)), & \text{если } x+y > 3 \\ 3.2 \sin(xy) \max(x, y) & \text{в противном случае} \end{cases}$
3.	$f = \begin{cases} \max(3 \sin x, 2 \min(x, y), x+a), & \text{если } x < 1 \text{ и } y > 1 \\ \min\left(\frac{x + e^{x+a}}{3}, \max(x, y)\right), & \text{если } x > 1 \text{ и } y > 1 \\ 4 \sin xy, & \text{в противном случае} \end{cases}$
4.	$R = \begin{cases} \max\left(a, \frac{x}{a}, \sin^2 a\right) & \text{если } a > 0 \text{ и } x > 1 \\ \min(a, ax + \sin b, \max(a, b)) & \text{если } a < 0 \text{ и } x > 1 \\ \max(a, b) & \text{в противном случае} \end{cases}$
5.	$T = \begin{cases} 4 \min(5^{3+x}, x+y, -4y), & \text{если } 2 < x+y < 3 \\ \max(e^{x-y}, 3 \ln x, \min(x, y)), & \text{если } x+y > 3 \\ 2.5 \sin(xy) \max(x, y) & \text{в противном случае} \end{cases}$
6.	$Z = \begin{cases} \max\left(c, \frac{x}{c}, \sin c\right) & \text{если } c > 0 \text{ и } x > 2 \\ \min(c, cx + \sin b, \max(c, b)) & \text{если } c < 0 \text{ и } x > 2 \\ \max(c, b) & \text{в противном случае} \end{cases}$
7.	$Z = \begin{cases} \max\left(a, \frac{x}{a}, \cos a\right) & \text{если } a > 0 \text{ и } x > 1 \\ \min(a, ax + \sin b, \max(a, b)) & \text{если } a < 0 \text{ и } x > 1 \\ \max(a, b) & \text{в противном случае} \end{cases}$



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

8.	$\alpha = \begin{cases} x^3 + \lg(xy)^{cd}, & \text{если } xy > 3 \\ 3 \min(x, y, \max(cx, dy)), & \text{если } 2 < xy < 3 \\ 2^{cd} - x, & \text{если } xy < 2 \end{cases}$
9.	$\beta = \begin{cases} xy + ab, & \text{если } xy > 0 \\ b - \min(ax, y), & \text{если } xy = 0 \\ \max(x^3, e^y, \sqrt{\ln y^2}), & \text{если } xy < 0 \end{cases}$
10.	$\begin{cases} \min(ax, y, \max(\sqrt{x}, \sin y^2)) & \text{если } x > 0 \quad y > 2 \\ \max(b, x, y) & \text{в противном случае} \end{cases}$
11.	$P = \begin{cases} \max(3^{x+a}, c + d, \min(x, a)), & \text{если } x > 3 \text{ и } xy < 2 \\ \min(x, a, c, y, d), & \text{если } x > 3 \text{ и } xy > 2 \\ 3c + dx, & \text{в противном случае} \end{cases}$
12.	$T = \begin{cases} \min(ax, y, \max(\sqrt{x}, \sin y)) & \text{если } x > 0 \text{ и } y > 2 \\ \max(b, x, y) & \text{в противном случае} \end{cases}$
13.	$Z = \begin{cases} \max(a, \frac{x}{a}, \sin a) & \text{если } a > 0 \text{ и } x > 1 \\ \min(a, ax + \sin b, \max(a, b)) & \text{если } a < 0 \text{ и } x > 1 \\ \max(a, b) & \text{в противном случае} \end{cases}$
14.	$b = \begin{cases} 2 \ln x \max(\min(x, y), \sin cx, \cos d), & \text{если } 0 < x < 1 \text{ и } 0 < y < 1 \\ \min(e^{x+1}, y), & \text{если } x < 0 \text{ и } y < 0 \\ a + x, & \text{в противном случае} \end{cases}$
15.	$s = \begin{cases} \max(2 \sin x, 3 \min(x, y), x + a), & \text{если } x < 1 \text{ и } y > 1 \\ \min\left(\frac{x + e^{2x+a}}{3}, \max(x, y)\right), & \text{если } x > 1 \text{ и } y > 1 \\ 4 \cos xy, & \text{в противном случае} \end{cases}$



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**



Пример отчета по заданию №2.

Задание №2. Программирование алгоритмов разветвляющейся структуры.

Использование условного оператора.

5. Задание, формула для вычислений, исходные данные приведены в таблице 2

Таблица 2 – Исходные данные задания 2

Вариант	Формулы для вычислений и условия их реализации
1	$\alpha = \begin{cases} x^3 + \lg(xy)^{cd}, & \text{если } xy > 3 \\ 3 \min(x, y, \max(cx, dy)), & \text{если } 2 < xy < 3 \\ 2^{cd} - x, & \text{если } xy < 2 \end{cases}$

80 Листинг программы на языке Pascal

```
Program test2_2;
Uses crt;
VAR b,c,x,z: real;
Begin
clrscr;
b:=5;
c:=7;
WriteLn ('Введите число x (для подтверждения ввода - Enter):');
ReadLn (x);
  If x>2 then
begin if c>0 then
begin
if c>sin(c) then z:=c else z:=sin(c);
...
end
else if c=0 then
begin if b>c then z:=b else z:=c end
end
else
```



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

```
begin if b>c then z:=b else z:=c end;  
WriteLn ('b=',b:6:3,' c=',c:6:3,' x=',x:6:3,' z=',z:6:3);  
ReadLn;  
End.
```

5. Блок-схема программы 2 приведена на рисунке 4.

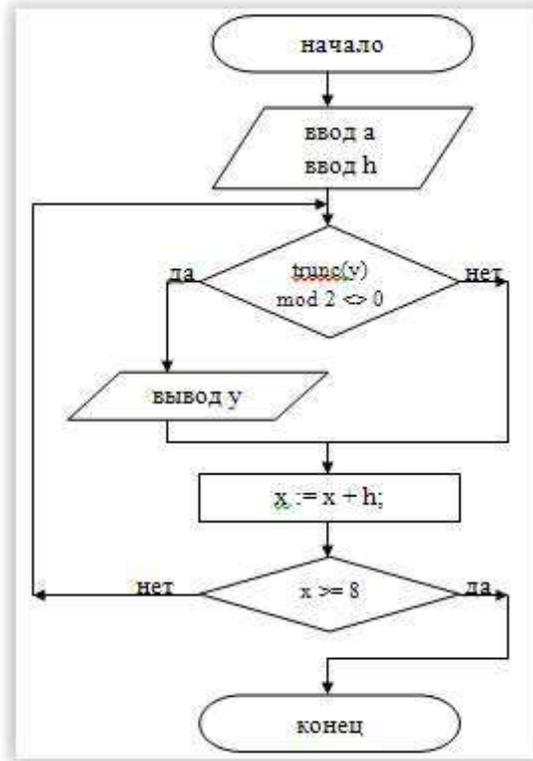


Рисунок 4 – Программа алгоритма разветвляющейся структуры

/ Скриншот окна среды программирования с введенным текстом программы приведен на рисунке 5.



Рисунок 5 – Исходный код программы 2



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Скриншот окна запущенной программы представлен на рисунке 6.

```
c:\ D:\TEST2_2.EXE
Введите число x (для подтверждения ввода - Enter):
0
b= 5.000 c= 7.000 x= 0.000 z= 7.000
Для завершения работы программы нажмите Enter.
```

Рисунок 6 – Скриншот окна запущенной программы 2

**Контрольное задание №3 по теме
«Программирование алгоритмов регулярной циклической структуры»**

Цикл – последовательность команд, которая повторяется до тех пор, пока не будет выполнено заданное условие. Повторяющиеся действия на каждом шагу цикла называются «телом цикла».

Известны три типа циклических алгоритмов: цикл с параметром (арифметический цикл или цикл со счетчиком), цикл с предусловием и цикл с постусловием.

/ цикл с предусловием: количество шагов цикла заранее не определено и зависит от входных данных задачи.

/ цикл с постусловием: не определено число повторений тела цикла, оно зависит от входных данных задачи. тело цикла будет выполняться до тех пор, пока значение условного выражения ложно. как только оно становится истинным, выполнение команды прекращается.

/ в цикле с параметром (арифметическом цикле) число его шагов (повторений) однозначно определяется правилом изменения параметра.

Задание:

В написать на языке Паскаль программу для задачи заданного типа;

В выполнить программу, подставив значения исходных данных в соответствии с вариантом задания.

Методические требования

1. выполнению контрольного задания №3:

∴ ενδαση при выполнении заданий решить задачу одного из следующих типов: табуляция (получить таблицу значений функции $y=f(x)$ при изменении x на отрезке $[a;b]$ с шагом h) или сумма или произведение (вычислить сумму или



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

произведение значений функции $y=f(x)$, удовлетворяющих заданным ограничениям, при изменении x на отрезке $[a;b]$ с шагом h);

6. объявить в программе тип используемых переменных (целый или вещественный) в соответствии с содержанием задачи. Использовать вещественные переменные с одинарной точностью;

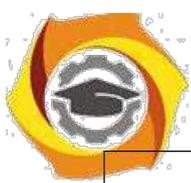
7. использовать в качестве параметра цикла переменную целого типа (счётчик числа повторений цикла);

8. во всех вариантах заданий вывести значения аргумента и функции в виде таблицы с заголовком столбцов;

9. после выполнения программы и проверки полученного результата сделать скриншоты: окна среды программирования с введенным текстом программы и окна запущенной программы.

Таблица 3 - варианты задания №3 по теме «Программирование алгоритмов регулярной циклической структуры»

Вариант	Тип задачи и ограничения	Функция	отрезок	шаг
7.	Табуляция	$y = \begin{cases} x^2 + 1, & \text{если } x < 0 \\ \cos x, & \text{если } 0 < x < 1.5 \\ \sin(x-1), & \text{если } x > 1.5 \end{cases}$	[-5;5]	0.4
8.	Произведение $Y > 0$	$y = x^4 + x^3 - 10x^2 - 34x - 25$	[-2; 0]	0.1
9.	Табуляция	$\begin{cases} e^x, & \text{если } x > 2 \\ x + 4, & \text{если } -2 < x < 2 \\ 0, & \text{если } x < -2 \end{cases}$	[-3;3]	0.4
10.	Сумма $y > 0$	$y = \frac{1}{x + 2\pi} - \sin x$	[- π , π]	$\pi/8$
11.	Табуляция	$y = \begin{cases} x^2 - 1, & \text{если } x < 0 \\ \cos x, & \text{если } 0 < x < 1.5 \\ \sin(x-1), & \text{если } x > 1.5 \end{cases}$	[-6;4]	0.6
6.	Табуляция	$y = \begin{cases} e^x & \text{если } x > 1 \\ 2x - 1 & \text{если } x < 0 \\ & \text{в противном случае} \end{cases}$	[-3;3]	0,3
9.	Произведение $Y > 0$	$y = x^4 + x^3 - 10x^2 - 34x - 25$	[-1;1]	0.1



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

5.	Табуляция	$y = \begin{cases} \sin x, & \text{если } -1 < x < 1 \\ 5 \cos x, & \text{в противном случае} \end{cases}$	[-4;4]	0.6
6.	Сумма $y > 0$	$y = \frac{1}{x + 2\pi} \sin x$	[-π;π]	π/4
7.	Табуляция	$y = \begin{cases} x^2 - 1, & \text{если } x < 0 \\ \cos x, & \text{если } 0 < x < 1.5 \\ \sin(x-1), & \text{если } x > 1.5 \end{cases}$	[-4;4]	0.5
8.	Табуляция	$y = \begin{cases} e^x, & \text{если } x > 2 \\ x + 4, & \text{если } -2 < x < 2 \\ 0, & \text{если } x < -2 \end{cases}$	[-2;2]	0.2 5
9.	Максимум	$N_0 = X e^{-x}$	[0.1;1.5]	0.1
10.	Произведение $Y > 0$	$18. \quad = x^4 + x^3 - 10x^2 - 34x - 25$	[-2;2]	0.2
11.	Табуляция	$y = \begin{cases} x^2 - 1, & \text{если } x < 0 \\ \cos x, & \text{если } 0 < x < 1.5 \\ \sin(x-1), & \text{если } x > 1.5 \end{cases}$	[-5;5]	0.4
12.	Табуляция	$y = \begin{cases} \sin x, & \text{если } -1 < x < 1 \\ 5 \cos x, & \text{в противном случае} \end{cases}$	[-2;2]	0.2 5

Оформление задания выполняется по аналогии с заданием 1 и 2.

**Контрольное задание №4 по теме
«Программирование алгоритмов итеративной циклической структуры»**

Задание:

4. написать на языке Паскаль программу согласно варианта;
- б) выполнить программу, подставив значения исходных данных в соответствии с вариантом задания

**Методические требования
выполнению контрольного задания №4:**

8. объявить в программе тип используемых переменных (целый или вещественный) в соответствии с содержанием задачи. Использовать вещественные переменные с удвоенной точностью;

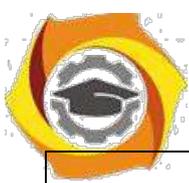


**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

11. считать, что заданная точность ϵ обеспечена, если модуль разности двух последовательных значений искомой величины меньше ϵ ;
12. выводить все изменяющиеся переменные при каждом повторении цикла, предусмотрев вывод имени и значения каждой переменной;
13. после выполнения программы и проверки полученного результата сделать скриншоты: окна среды программирования с введенным текстом программы и окна запущенной программы.

Таблица 4 - варианты задания №4 по теме «Программирование алгоритмов итеративной циклической структуры»

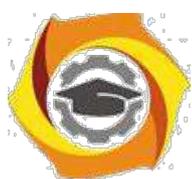
Вариант	Задание
В 000	Вычислить и вывести те члены последовательности (\quad) значения которых больше $\epsilon=0.001$, при $x=0.4$ Вычислить с точностью $\epsilon =$ значение функции $y = \sqrt{\quad}$ при $x = 2$, воспользовавшись итерационной формулой Герона:
3	сравнить результат со значением, полученным с помощью соответствующей встроенной функции. Вычислить $\sin 0.4$ с точностью $\epsilon =$, воспользовавшись разложением в ряд: $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^n \frac{x^{2n-1}}{(2n+1)!}$
	Сравнить результат со значением, полученным с помощью соответствующей встроенной функции
4	Вычислить и вывести те члены последовательности (\quad) значения которых больше $\epsilon=0.001$, при $x=0.2$.
5.	Найти наибольшее целое положительное число n , удовлетворяющее условию:
6.	Вычислить с точностью $\epsilon =$, значение функции $y = \sqrt{\quad}$ при $x = 3$, воспользовавшись итерационной формулой Герона: сравнить результат со значением, полученным с помощью соответствующей встроенной функции.
	20



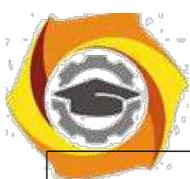
**Негосударственное частное образовательное
учреждение
высшего образования**

«Технический университет»

7	<p>Вычислить $\sin 0.5$ с точностью $e =$ _____, воспользовавшись разложением в ряд:</p> <p style="text-align: center;">_____</p>
8	<p>Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.</p> <p>Найти наибольшее целое положительное число n, удовлетворяющее условию:</p> $3n^3 - 730n < 5$
9	<p>Вычислить с точностью $e=10^{-5}$ значение функции $y = \frac{1}{\sqrt{x}}$ при $x=2$, воспользовавшись итерационной формулой:</p> $y_{i+1} = 1.5y_i - 0.5xy_i^3 ; \quad i = 0, 1, 2, \dots ; \quad y_0 = 1$ <p>Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.</p>
10	<p>Вычислить с точностью до $e=10^{-5}$ константу Эйлера (основание натурального логарифма), воспользовавшись разложением в ряд, $x=1.1$:</p> $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots$ <p>Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.</p>
11	<p>Вычислить $\cos 0.7$ с точностью до $e=10^{-5}$, воспользовавшись разложением в ряд:</p> $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!}$ <p>Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.</p>
12	<p>Вычислить и вывести те члены последовательности, значения которых больше $e = 0.001$, при $x = 0,7$.</p> $1, \frac{x}{2!}, \frac{x^2}{3!}, \dots, \frac{x^n}{(n+1)!}, \dots$
13	<p>Вычислить $\cos 0.45$ с точностью до $e=10^{-5}$, воспользовавшись разложением в ряд:</p> $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!}$ <p>Сравнить результат со значением, полученным с помощью</p>



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**



	соответствующей встроенной функции.
14	<p>Вычислить с точностью до $\epsilon=10^{-5}$ константу Эйлера (основание натурального логарифма), воспользовавшись разложением в ряд, $x=3.2$:</p> $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$ <p>Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.</p>
15	<p>Вычислить и вывести те члены последовательности, значения которых больше $\epsilon = 0.001$, при $x = 0.85$</p> $1, \frac{x}{2!}, \frac{x^2}{3!}, \dots, \frac{x^n}{(n+1)!}, \dots$

Оформление задания выполняется по аналогии с заданием 1 и 2.

9. Методические указания по выполнению задания №5 «Основы использования Microsoft Office Excel (Open Libre)»

Выполните последовательно следующие действия:

10. Выбрать функцию (табл. 7.1) и исходные данные (табл. 7.2) по варианту № bc (константы варианта: $A=a$, $B=b$, $C=c$, где a , b , c – крайние 3 цифры студенческого билета, если $bc > 60$, то № варианта = $bc - 60$).

11. Задать изменение X по формуле: $X_{i+1} = X_i + h$, где: $i=1, 2, \dots, 20$, X_1 – начальное значение и h – шаг определяются из табл.2.

12. Используя электронные таблицы Microsoft Office Excel или Open (Libre), вычислить значение функции для полученных X , и по этим данным построить график функции.

13. Вычислить Y_{\max} – наибольшее и Y_{\min} – наименьшее значения функции.

14. Определить величину K по формуле: $K = (Y_{\max} - Y_{\min}) / M$, где: M определяется из табл. 7.2.

15. Определить какую долю в процентах от общего числа значений функции (их всего 20 шт.) составляют ее значения, большие K .

16. Определить какую долю в процентах от общего числа значений функции составляют ее значения, которые меньше, либо равные числу K .

17. Проверить, что в сумме все доли дают 100%.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

к По этим данным построить круговую объемную диаграмму, принимая все значения функции за 100%.

Таблица 7.1 - Формулы для вычисления функции

- | | |
|---|--|
| <p>1. $y = \frac{A+B}{2A-B}(A+C) \sin^2 x;$</p> <p>2. $y = x_{AC} + e^{-B} + \sqrt{D}; \quad D=7.8;$</p> <p>3. $y = \frac{A+Bx-Cx^2}{\cos x + e^{\sqrt{x}}} + x^{A+B};$</p> <p>4. $y = \cos B / \cos x \cdot \left(\frac{AC}{x^A}\right)^2$</p> <p>5. $y = \frac{x^A}{B} \sqrt{x^C + 1} \frac{1}{A^x + 1};$</p> <p>6. $T = A \frac{1}{x} + C \sqrt{\sin B + x};$</p> <p>7. $y = \sqrt[4]{x^B + A/C} A - 2C ;$</p> <p>8. $y = \cos \frac{A}{B} + e^x + \frac{1+x}{\sin^2 x};$</p> <p>9. $y = B_x + \frac{A}{x^C} + \sqrt{ A-x };$</p> <p>10. $y = 2 \sqrt{A^x + \frac{4x^2}{A^2 + B - C}};$</p> <p>11. $y = \frac{Ae^{\sin x} + \cos(B+C)}{\sin(x+B)};$</p> <p>12. $y = \sqrt{\sin^2(x^2 + A) + C^x};$</p> <p>13. $y = \cos B/100 \cos x \cdot \left(\frac{AC}{x^A}\right)^2$</p> <p>14. $y = \frac{x^A}{1 + \frac{x^B}{1+x^C}};$</p> | <p>31. $y = \frac{x^A}{1 + \frac{x^B}{x^C}};$</p> <p>13. $y = \frac{A+Bx+Cx^2 + (A)^4}{(1-\operatorname{tg}x)^2(B)} ;$</p> <p>33. $z = \frac{x^2 + Bx - 3C }{\ln(x^3 + BC + A)};$</p> <p>34. $y = \frac{A+B}{A+3B}(2A+B) \sin^2 2x;$</p> <p>35. $y = \sqrt[4]{x^B + A/C} A - 2C ;$</p> <p>36. $y = B_x + \frac{A}{x^C} + \sqrt{ A-x };$</p> <p>37. $y = \cos \frac{A}{B} + e^x + \frac{1+x}{\sin^2 x};$</p> <p>38. $z = \frac{1 + (A/C)^3}{\frac{e^2}{x^x} (1+BC)};$</p> <p>39. $y = \frac{\sqrt[4]{x} + Ax + 6,8 \sin x}{x + e^{B+C}};$</p> <p>4. $y = \frac{1}{\cos x + \ln \operatorname{tg} B^A} + Cx;$</p> <p>41. $y = \frac{A+B + \cos^2(x) + x^{A-B}}{C + \sqrt{A+B}};$</p> <p>42. $y = \sqrt[3]{(x-AB)^2} + 2e^{C/x-1};$</p> <p>43. $y = \sqrt{x + \sqrt{x - AC}} \sqrt[3]{Bx};$</p> <p>6. $y = \frac{Ae^{\sin x} + \cos(B+C)}{\sin(x+B)};$</p> |
|---|--|



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

$$15. y = e^{-\frac{x}{B}} (\cos 2Ax + \sin x);$$

$$16. y = \frac{A+B}{A+3B} (2A+B) \sin^2 2x;$$

$$17. y = x^{A/C} + e^{-2B} + \sqrt{3D}; \quad D=7,8;$$

$$18. y = \frac{A+x-Cx^2}{\cos 2x + e^{\sqrt{x}}} + x^{A-B};$$

$$19. y = \cos \frac{A}{B} + e^x + \frac{1+x}{\sin^2 x};$$

$$20. y = \frac{x/A}{B} \sqrt{x^{-C} + 1} \frac{1}{A^{-x} + 1};$$

$$21. y = A \frac{1}{x} + B \sqrt{\sin C + 2x};$$

$$22. y = \sqrt{\sin^2(x^2 + A) + C^{-x}};$$

$$23. T = A \frac{1}{x} + C \sqrt{\sin B + x};$$

$$24. z = A^{-2x} + \frac{B}{x^C} + \sqrt{2A-x};$$

$$25. y = \frac{Ae^{\sin x} + \cos(B+C)}{\sin(x+B)};$$

$$26. y = \sqrt{A^{-x} + \frac{4x^2}{A^{-2} + B - C}};$$

$$4. y = \sqrt{\sin^{-2}(x^2 + A) + C^{-x}};$$

$$28. y = x^{A/C} + e^{-2B} + \sqrt{3D}; \quad D=7,8;$$

$$29. y = \frac{x^B}{1 + \frac{x^A}{1+x^{-C}}};$$

$$30. y = \frac{x^A}{B} \sqrt{x^C + 1} \frac{1}{A^x + 1};$$

$$45. T = A \frac{1}{x} + C \sqrt{\sin B + x};$$

$$46. y = \frac{A+Bx-Cx^2}{\cos x + e^{\sqrt{x}}} + x^{A+B};$$

$$47. y = \frac{x^A}{B} \sqrt{x^C + 1} \frac{1}{A^x + 1};$$

$$48. z = \frac{x^2 + |4Bx + 3/C|}{\ln|x^3 + BC + A|};$$

$$49. y = B_x + \frac{A}{x^C} + \sqrt{|A-x|};$$

$$50. y = e^{-\frac{x}{B}} (\cos 2Ax + \sin x);$$

$$51. y = \cos \frac{A}{B} + e^x + \frac{1+x}{\sin^2 x};$$

$$52. y = -\sqrt{x^B + A/C} |A-2C|;$$

$$53. y = \cos |B/100| \cos x \quad \left(\frac{AC}{x^A} \right)^2$$

$$54. y = \cos \frac{A}{B} + e^x + \frac{1+x}{\sin^2 x};$$

$$55. y = \frac{1}{\cos 2x} + \ln \left| \operatorname{tg} \frac{4A}{B} \right| + C/x;$$

$$56. z = \frac{A+B^x + x^{A-B}}{C + \sqrt{A+B}};$$

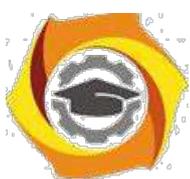
$$4. T_2 = -\sqrt{|x-AB|} + 2e^{x-C};$$

$$58. y = \sqrt{2/x} + \sqrt{x-AC} \sqrt[3]{Bx};$$

$$5. y_1 = \sin(1 - \sqrt[4]{A/x} \sqrt{C});$$

$$6. z = e^{-\sin Bx} - \operatorname{tg} \left(\frac{C}{A} \right) \sqrt[3]{\cos Bx};$$

Таблица 7.2 - Исходные данные по вариантам



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

№ Варианта	Начальное значение X_1	Шаг h	M	№ Варианта	Начальное значение X_1	Шаг h	M
1	0,1	2,05	2	31	1,25	0,52	4
2	0,0006	0,2	3	32	0,95	3,46	5
3	0,0006	0,2	4	33	1,81	3,52	6
4	0,5	0,03	5	34	0,5	0,03	7
5	2,98	0,42	6	35	0,52	0,05	6
6	4,04	0,48	7	36	1,26	0,2	5
7	0,524	0,05	6	37	0,66	0,09	4
8	0,658	0,09	5	38	0,001	0,2	3
9	1,26	0,2	4	39	0,73	0,38	2
10	0,966	0,7	3	40	0,82	0,39	2
11	1,2	0,02	2	41	0,92	0,4	3
12	1,48	0,08	2	42	0,38	0,01	4
13	0,5	0,03	3	43	0,78	0,43	5
14	1,25	0,52	4	44	1,2	0,02	6
15	0,14	0,08	5	45	4,04	0,48	7
16	0,5	0,03	6	46	1,26	0,2	6
17	0,3	4,25	7	47	2,98	0,43	5
18	0,1	0,05	6	48	3,44	0,44	2
19	0,66	0,09	5	49	1,26	0,2	3
20	0,5	0,03	4	50	0,14	0,08	4
21	0,97	1,7	3	51	0,66	0,09	5
22	1,48	0,08	2	52	0,52	0,05	6
23	4,04	0,48	2	53	0,5	0,03	7
24	1,2	0,02	3	54	0,66	0,09	6
25	1,2	0,02	4	55	-17,48	4,86	5
26	24,23	8	5	56	-18,09	4,92	4
27	1,25	0,52	6	57	-18,7	4,98	3
28	0,3	0,15	7	58	-19,33	5,04	2
29	27,32	0,43	6	59	-19,96	5,11	5
30	2,98	9	5	60	-20,59	5,17	6

Пример выполнения задания

1. Определим функцию $z = e^{-\sin Bx} - \operatorname{tg} \left(\frac{C}{A} \right) + \sqrt[3]{\cos Bx}$; и исходные данные

варианта

a	b	c	h	M	X_1
5	6	1	5,17	6	-20,59

4. Посчитаем X_j



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

x
-20,59
=P11,4+\$A\$116
-10,25
-5,08
0,09
5,26
10,43
15,6
20,77
25,94
31,11
36,28
41,45
46,62
51,79
56,96
62,13
67,3
72,47
77,64

4. Для рассчитанных X-ов вычислим значения функции и построим ее график

Т.к., функция сложная, то вычислим отдельно каждое слагаемое, а потом сложим их со своими знаками. Так, гораздо легче устранять возможные ошибки.

На рисунках 1-3 показаны формулы для вычисления каждого из слагаемых

Данные	i	x	E	tg	Корень	z
a	1	-20,59	=EXP(-SIN(\$A\$112*C109))			-0,58252
	5	2	-15,42	0,372431031	0,20271	-0,36899
b	3	-10,25	0,378481296	0,20271	0,61856	0,79434
	6	4	-5,08	0,447021348	0,20271	0,84017
c	5	0,09	0,598017063	0,20271	0,95012	1,34543
	1	6	5,26	0,866255348	0,20271	0,99653
h	7	10,43	1,282997321	0,20271	0,98937	2,06966
	5,17	8	15,6	1,828356173	0,20271	0,92732
		9	20,77	2,373283466	0,20271	0,7953
		10	25,94	2,695032797	0,20271	0,5076
		11	31,11	2,625193416	0,20271	-0,63962
		12	36,28	2,202447729	0,20271	-0,84979
		13	41,45	1,635291969	0,20271	-0,95489
		14	46,62	1,125219822	0,20271	-0,99767
		15	51,79	0,760242177	0,20271	-0,98707
		16	56,96	0,535902455	0,20271	-0,92114
		17	62,13	0,416040478	0,20271	-0,78327
		18	67,3	0,369924463	0,20271	-0,47199
		19	72,47	0,383629844	0,20271	0,65924
		20	77,64	0,461412231	0,20271	0,859
						1,1177



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Рисунок 1- Формула для 1-го слагаемого

Данные	i	x	E	tg	Корень	z
a	1	-20,59	0,426987735	=TAN(SAS114/SAS110)		
5	2	-15,42	0,372431031	0,20271	-0,53871	-0,36899
b	3	-10,25	0,378481296	0,20271	0,61856	0,79434
6	4	-5,08	0,447021348	0,20271	0,84017	1,08449
c	5	0,09	0,598017063	0,20271	0,95012	1,34543
1	6	5,26	0,866255348	0,20271	0,99653	1,66008
h	7	10,43	1,282997321	0,20271	0,98937	2,06966
5,17	8	15,6	1,828356173	0,20271	0,92732	2,55297
	9	20,77	2,373283466	0,20271	0,7953	2,96587
	10	25,94	2,695032797	0,20271	0,5076	2,99992
	11	31,11	2,625193416	0,20271	-0,63962	1,78286
	12	36,28	2,202447729	0,20271	-0,84979	1,14995
	13	41,45	1,635291969	0,20271	-0,95489	0,47769
	14	46,62	1,125219822	0,20271	-0,99767	-0,07516
	15	51,79	0,760242177	0,20271	-0,98707	-0,42953
	16	56,96	0,535902455	0,20271	-0,92114	-0,58795
	17	62,13	0,416040478	0,20271	-0,78327	-0,56994
	18	67,3	0,369924463	0,20271	-0,47199	-0,30478
	19	72,47	0,383629844	0,20271	0,65924	0,84016
	20	77,64	0,461412231	0,20271	0,859	1,1177

Рисунок 2- Формула для 2-го слагаемого

Данные	i	x	E	tg	Корень	z	Zmax	Zmin
a	1	-20,59	0,426987735	0,20271	=СТЕПЕНЬ(COS(SAS112*C109);1/3)			
5	2	-15,42	0,372431031	0,20271	-0,53871	-0,36899		
b	3	-10,25	0,378481296	0,20271	0,61856	0,79434		
6	4	-5,08	0,447021348	0,20271	0,84017	1,08449		
c	5	0,09	0,598017063	0,20271	0,95012	1,34543		
1	6	5,26	0,866255348	0,20271	0,99653	1,66008		
h	7	10,43	1,282997321	0,20271	0,98937	2,06966		
5,17	8	15,6	1,828356173	0,20271	0,92732	2,55297		
	9	20,77	2,373283466	0,20271	0,7953	2,96587		
	10	25,94	2,695032797	0,20271	0,5076	2,99992		
	11	31,11	2,625193416	0,20271	-0,63962	1,78286		
	12	36,28	2,202447729	0,20271	-0,84979	1,14995		
	13	41,45	1,635291969	0,20271	-0,95489	0,47769		
	14	46,62	1,125219822	0,20271	-0,99767	-0,07516		
	15	51,79	0,760242177	0,20271	-0,98707	-0,42953		
	16	56,96	0,535902455	0,20271	-0,92114	-0,58795		
	17	62,13	0,416040478	0,20271	-0,78327	-0,56994		
	18	67,3	0,369924463	0,20271	-0,47199	-0,30478		
	19	72,47	0,383629844	0,20271	0,65924	0,84016		
	20	77,64	0,461412231	0,20271	0,859	1,1177		

Рисунок 3- Формула для 3-го слагаемого

В итоге получим значения функции для заданных X-ов и построим для них ее график



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Данные	i	x	z	
a	1	-20,59	-0,58252	
	5	2	-15,42	-0,36899
b	3	-10,25	0,79434	
	6	4	-5,08	1,08449
c	5	0,09	1,34543	
	1	6	5,26	1,66008
h	7	10,43	2,06966	
	5,17	8	15,6	2,55297
		9	20,77	2,96587
		10	25,94	2,99992
		11	31,11	1,78286
		12	36,28	1,14995
		13	41,45	0,47769
		14	46,62	-0,07516
		15	51,79	-0,42953
		16	56,96	-0,58795
		17	62,13	-0,56994
		18	67,3	-0,30478
		19	72,47	0,84016
		20	77,64	1,1177



4. Вычислим наибольшее и наименьшее значения функции

z	Zmax	Zmin	z	Zmax	Zmin	K
-0,58252	=МАКС(G109:G128)		-0,58252	2,999924	=МИН(G109:G128)	
-0,36899			-0,36899			
0,79434			0,79434			
1,08449			1,08449			
1,34543			1,34543			
1,66008			1,66008			
2,06966			2,06966			
2,55297			2,55297			
2,96587			2,96587			
2,99992			2,99992			
1,78286			1,78286			
1,14995			1,14995			
0,47769			0,47769			
-0,07516			-0,07516			
-0,42953			-0,42953			
-0,58795			-0,58795			
-0,56994			-0,56994			
-0,30478			-0,30478			
0,84016			0,84016			
1,1177			1,1177			

Zmax	Zmin
2,999924	-0,58795

4. получим в итоге

Посчитаем K

Zmax	Zmin	K	Z>K
2,999924	-0,58795	=(H109-I109)/6	



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

К
0,59798

1. получим

– 9. Определим количество значений функции, больших К

Для этого используем функцию, которая при выполнении условия записывает в ячейку это значение, иначе 0.

z	Zmax	Zmin	K	Z>K	Z<=K
-0,58252	2,999924	-0,58795	0,59798	-ЕСЛИ(G109>=0,59798;0,58252;0)	0
-0,36899				0	-0,36899
0,79434				0,79434	0
1,08449				1,08449	0
1,34543				1,34543	0
1,66008				1,66008	0
2,06966				2,06966	0
2,55297				2,55297	0
2,96587				2,96587	0
2,99992				2,99992	0
1,78286				1,78286	0
1,14995				1,14995	0
0,47769				0	0,47769
-0,07516				0	-0,07516
-0,42953				0	-0,42953
-0,58795				0	-0,58795
-0,56994				0	-0,56994
-0,30478				0	-0,30478
0,84016				0,84016	0
1,1177				1,1177	0

Для того, чтобы определить сколько это составляет % от общего числа значений функции, нужно посчитать все не равные 0 ее значения, которые больше К, умножить на 100% и разделить на общее количество значений функции (20).

Данные	i	x	z	Zmax	Zmin	K	Z>K	Z<=K
a	1	-20,59	-0,58252	2,999924	-0,58795	0,59798	0	-0,58252
	5	2	-15,42	-0,36899			0	-0,36899
b	3	-10,25	0,79434				0,79434	0
	6	4	-5,08	1,08449			1,08449	0
c	5	0,09	1,34543				1,34543	0
	1	6	5,26	1,66008			1,66008	0
h	7	10,43	2,06966				2,06966	0
	5,17	8	15,6	2,55297			2,55297	0
		9	20,77	2,96587			2,96587	0
		10	25,94	2,99992			2,99992	0
		11	31,11	1,78286			1,78286	0
		12	36,28	1,14995			1,14995	0
		13	41,45	0,47769			0	0,47769
		14	46,62	-0,07516			0	-0,07516
		15	51,79	-0,42953			0	-0,42953
		16	56,96	-0,58795			0	-0,58795
		17	62,13	-0,56994			0	-0,56994
		18	67,3	-0,30478			0	-0,30478
		19	72,47	0,84016			0,84016	0
		20	77,64	1,1177			1,1177	0

Проверка =СЧЁТЕСЛИ(K109;K128;"<0")*100/8128

Оставшиеся значения – это те, которые меньше, либо равные К и для их определения применим следующую формулу:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Данные	i	x	z	Zmax	Zmin	K	Z>K	Z<=K		
a	1	-20,59	-0,58252	2,999924	-0,58795	0,59798	0	=ЕСЛИ(G109<=-\$109;G109;0)		
	5	-15,42	-0,36899				0	-0,36899		
b	3	-10,25	0,79434				0,79434	0		
	6	-5,08	1,08449				1,08449	0		
c	5	0,09	1,34543				1,34543	0		
	1	5,26	1,66008				1,66008	0		
h	7	10,43	2,06966				2,06966	0		
	5,17	15,6	2,55297				2,55297	0		
		20,77	2,96587				2,96587	0		
		25,94	2,99992				2,99992	0		
		31,11	1,78286				1,78286	0		
		36,28	1,14995				1,14995	0		
		41,45	0,47769				0	0,47769		
		46,62	-0,07516				0	-0,07516		
		51,79	-0,42953				0	-0,42953		
		56,96	-0,58795				0	-0,58795		
		62,13	-0,56994				0	-0,56994		
		67,3	-0,30478				0	-0,30478		
		72,47	0,84016				0,84016	0		
		77,64	1,1177				1,1177	0		
						Проверка	60	40	100	

И аналогично определим их долю в процентах от общего числа значений функции.

Понятно, что в сумме эти 2-е цифры дают 100% значений функции- это и есть проверка.

По этим 2-м цифрам строим круговую объемную диаграмму:





**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Требования к оформлению контрольной работы

Контрольная работа выполняется в электронном и печатном виде с учётом возможностей, которыми располагают для оформления MS Office: Word или OpenOffice и LibreOffice: Writer.

Бумажный носитель сдается на проверку, а файл с электронной версией работы (форматов: .doc / .docx / .odt и .pas) предъявляется на собеседовании.

Отчет, распечатываемый на бумажный носитель, включает в себя следующие части:

- Титульная страница.
 - Содержание.
 - Введение, в котором кратко излагаются сведения - какие программные продукты и, соответственно, их возможности использовались для подготовки материала контрольной работы.
 - Основная часть должна включать распечатку:
 - отчет по заданию №1 на тему «Программирование алгоритмов линейной структуры»;
 - отчет по заданию №2 на тему «Программирование алгоритмов разветвляющейся структуры»;
 - отчет по заданию №3 на тему «Программирование алгоритмов регулярной циклической структуры»;
 - отчет по заданию №4 на тему «Программирование алгоритмов итеративной циклической структуры».
 - Заключение, в котором подводятся итоги работы.
 - Библиографический список (не менее 5 наименований), литература и Интернет-ресурсы, которые использовались при выполнении контрольной работы. Библиографический список оформляется согласно требованиям ГОСТР 7.0.5–2008 «Библиографическая ссылка».
- Общие требования и правила составления».

Оформление контрольной работы должно соответствовать требованиям стандарта:

Поля на листе формата А4: левое поле – 30 мм, правое – 15 мм, верхнее – 20 мм и нижнее – 25 мм.

Все страницы (кроме титульной) нумеруются в правом нижнем углу арабскими цифрами.

Текст в распечатанном виде набирается шрифтом TimesNewRoman, 14пт, интервал полуторный, абзацный отступ 1,25 см, выравнивание по ширине страницы.

Все листы пробиваются дыроколом и подшиваются в скоросшиватель. CD-диск помещается в отдельный прозрачный файл в конце отчета.

***Практическое занятие №1. Классификация и формы представления моделей.
Информационная модель объекта.***

/ среде MS Excel создать информационную модель простейшей ЛВС. Исходные данные:

- 1.1 Число серверов – 1.
- 1.2 Число отделов – 4.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

1.3 Число ПЭВМ (рабочих станций, персональных компьютеров - РС) в каждой рабочей группе равно $4+i$.

1.4 Занимаемая полоса пропускания линии связи при передаче данных в процессе общения пары клиент – сервер равна:

В направлении ПЭВМ – сервер – 1,35 МБит/с;

В направлении сервер – ПЭВМ – 5 МБит/с.

1.5 Занимаемая полоса пропускания линии связи при передаче данных в процессе общения пары клиент – сервер равна:

– в направлении ПЭВМ -Internet - 1,9 МБит/с;

– в направлении Internet – ПЭВМ – 3,8 МБит/с.

1.6 Занимаемая полоса пропускания линии связи при передаче данных между компьютерами – 1,5 МБит/с.

Для представленных данных необходимо:

10. Построить схему сети;

11. Создать шаблон позволяющий рассчитывать значения трафика в каждой линии

связи.

12. Вывести на график пропускные способности в различных линиях связи.

13. Увеличивая параметры занимаемой полосы пропускания на 30%, с шагом 0.1, представить изменение пропускной способности в точках подключения к серверу и выходу в интернет.

14. Сделать выводы о возможности процессов и целесообразности этого процесса.

Контрольные вопросы ПЗ1(УК-1):

12. Классификация моделей.

13. Основные этапы разработки и исследования моделей на компьютере.

14. Сущность математического моделирования.

15. Сущность компьютерного моделирования.

16. Создать простейшие модели объектов и процессов в виде изображений и чертежей, динамических (электронных) таблиц.

17. Провести компьютерные эксперименты с использованием готовых моделей объектов и процессов.



Практическое занятие №2. Вычислительные сети. Основные понятия. Построение и компоненты. Основные топологии.

Цель работы: знать назначение и классификацию программного обеспечения вычислительных сетей, основные возможности сетевых операционных сред, уметь использовать некоторые сетевые прикладные программные пакеты для решения сетевых задач.

1.2 Указания по оформлению отчета:

Отчет должен содержать: титульный лист, цель работы; ответы на контрольные вопросы; выводы.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Компьютерной сетью называют совокупность узлов (компьютеров, терминалов, периферийных устройств), имеющих возможность информационного взаимодействия друг с другом с помощью специального коммуникационного оборудования и программного обеспечения.

Средства передачи и обработки информации ориентированы в ней на коллективное использование общесетевых ресурсов – информационных, программных, аппаратных.

Компьютерные сети могут работать в различных режимах: обмена данными между абонентами сети, запроса и выдачи информации, сбора информации пакетной обработки данных по запросам пользователей с удаленных терминалов, в диалоговых режимах.

Таким образом, с появлением сетей ЭВМ разрешены две очень важные проблемы:

10. обеспечение в принципе неограниченного доступа к ЭВМ пользователей независимо от территориального расположения,

11. возможность оперативного перемещений больших массивов информации на любые расстояния, позволяющий своевременно получать данные для принятия тех или иных решений.

Использование вычислительных сетей дает предприятию следующие возможности:

1. Разделение дорогостоящих ресурсов;
2. Улучшение доступа к информации;
3. Быстрое и качественное принятие решений;
4. Совершенствование коммуникаций;
5. Свобода в территориальном размещении компьютеров.

Программное обеспечение сетей ЭВМ в расширенном варианте составляют:

13. сетевые операционные системы;
14. сетевые драйвера, протоколы, службы и другое дополнительное программное обеспечение сетевых интерфейсов;
15. прикладное сетевое программное обеспечение.

Под сетевыми операционными системами понимают такие операционные системы, которые обеспечивают пользователям распределенный доступ к сетям ЭВМ.

Во вторую группу входит большой круг всевозможного программного обеспечения в основном изготовителя данного интерфейса (сетевой платы, модема и т.п.) для обеспечения правильной работы сетевого устройства.

При этом под драйвером понимается программа, непосредственно взаимодействующая с интерфейсом - сетевым адаптером и операционной системой (ОС). Драйвер сетевого адаптера взаимодействует с ОС через систему протоколов и служб, которые могут находиться как в самих ОС, так и поставляться вместе с устройством.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

При этом под сетевым протоколом понимается набор правил поведения сетевых узлов при передаче-приеме информации.

Под сетевыми службами понимается набор программного обеспечения сетевого обеспечения узкоспециального назначения, например:

№ клиенты сетей - позволяют подключаться, обозревать и пользоваться сетевыми ресурсами соответствующих сетей,

№ службы контроля трафика сетей,

№ службы использования доступа к разделяемым ресурсам,

№ доменные службы и др.

Круг прикладного сетевого программного обеспечения составляют всевозможные сетевые приложения.

Каждый компьютер работает под управлением собственной операционной системы, а какая-либо «общая» операционная система, распределяющая работу между компьютерами сети, отсутствует. Взаимодействие между компьютерами сети происходит за счет передачи сообщений через сетевые адаптеры и каналы связи. С помощью этих сообщений один компьютер обычно запрашивает доступ к локальным ресурсам другого компьютера. Такими ресурсами могут быть как данные, хранящиеся на диске, так и разнообразные периферийные устройства — принтеры, модемы, факс-аппараты и т.д. Разделение локальных ресурсов каждого компьютера между всеми пользователями сети — основная цель создания вычислительной сети.

Каким же образом сказывается на пользователе тот факт, что его компьютер подключен к сети? Прежде всего, он может пользоваться не только файлами, дисками, принтерами и другими ресурсами своего компьютера, но и аналогичными ресурсами других компьютеров, подключенных к той же сети. Правда, для этого недостаточно снабдить компьютеры сетевыми адаптерами и соединить их кабельной системой. Необходимы еще некоторые добавления к операционным системам этих компьютеров. На тех компьютерах, ресурсы которых должны быть доступны всем пользователям сети, необходимо добавить модули, которые постоянно будут находиться в режиме ожидания запросов, поступающих по сети от других компьютеров.

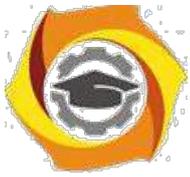
Обычно такие модули называются программными серверами (server), так как их главная задача — обслуживать (serve) запросы на доступ к ресурсам своего компьютера. На компьютерах, пользователи которых хотят получать доступ к ресурсам других компьютеров, также нужно добавить к операционной системе некоторые специальные программные модули, которые должны вырабатывать запросы на доступ к удаленным ресурсам и передавать их по сети на нужный компьютер. Такие модули обычно называют программными клиентами (client).

Собственно же сетевые адаптеры и каналы связи решают в сети достаточно простую задачу — они передают сообщения с запросами и ответами от одного компьютера к другому, а основную

работу по организации совместного использования ресурсов выполняют клиентские и серверные части операционных систем.

Пара модулей «клиент – сервер» обеспечивает совместный доступ пользователей к определенному типу ресурсов, например к файлам. В этом случае говорит, что пользователь имеет дело с файловой службой (service). Обычно сетевая операционная система поддерживает несколько видов сетевых служб для своих пользователей — файловую службу, службу печати, службу электронной почты, службу удаленного доступа и т. п.

Термины «клиент» и «сервер» используются не только для обозначения программных модулей, но и компьютеров, подключенных к сети. Если компьютер предоставляет свои ресурсы другим компьютерам сети, то он называется сервером, а если он их потребляет — клиентом. Иногда один и тот же компьютер может одновременно играть роли и сервера, и клиента.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Сетевые службы всегда представляют собой распределенные программы, состоящие из нескольких взаимодействующих частей, причем каждая часть, как правило, выполняется на отдельном компьютере сети.

До сих пор речь шла о системных распределенных программах. Однако в сети могут выполняться и распределенные пользовательские программы - приложения. Распределенное приложение также состоит из нескольких частей, каждая из которых выполняет какую-то определенную законченную работу по решению прикладной задачи. Например, одна часть приложения, выполняющаяся на компьютере пользователя, может поддерживать специализированный графический интерфейс, вторая - работать на мощном выделенном компьютере и заниматься статистической обработкой введенных пользователем данных, а третья - заносить полученные результаты в базу данных на компьютере с установленной стандартной СУБД. Распределенные приложения в полной мере используют потенциальные возможности распределенной обработки, предоставляемые вычислительной сетью, и поэтому часто называются сетевыми приложениями.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

5. Охарактеризовать сетевые операционные системы по следующей схеме:

- 1) платность,
- 2) доступ к исходному коду,
- 3) многоплатформенность,
- 4) мультизадачность,
- 5) количество пользователей,
- 6) функции управления сетью,
- 7) интерфейс работы,
- 8) потребляемые ресурсы.

Контрольные вопросы ПЗ2(УК-1):

9. Что такое вычислительная сеть?
10. Что такое компоненты вычислительных сетей?
11. Что такое построение вычислительных сетей?
12. Что такое информационно-вычислительная сеть?
13. Что такое топология сети? Какие основные виды топологий сетей существуют?
14. Каким образом составляются различные конфигурации сетей? Какие сетевые устройства это реализуют?
15. Каким образом информация распространяется в сети? Для чего используется команда ping?
16. Как соотносятся понятия «сеть», «корпоративная сеть» и «подсеть»?
17. Что такое маска подсети и как она задается?
18. Что такое шлюзы подсети и для чего они используются?
19. Что такое DNS сервер подсети и для чего он используется?
20. Из чего состоит IP адрес конкретного ПК и как он задается?
21. Что такое рабочая группа, как ее создать и на что это влияет?



***Практическое занятие №3. Основные методы защиты информации в компьютерных сетях.
Виды электронной подписи.***

Задание 1 Защита документов, созданных в Microsoft Word.

Цель работы: Изучение возможностей современных средств защиты документов, созданных в среде приложений OS Microsoft. Закрепление теоретического материала. Изучение способов и систем защиты файлов и файловых систем.

Учебно-наглядные пособия и ТСО: ПЭВМ, ОС Windows, пакет Microsoft Office, методическое пособие. ЗАДАНИЕ

Создать документ в приложении Word офисного пакета Microsoft Office Используя свойства и возможности приложения Word, защитить созданный файл паролем. Используя настройки атрибутов файла, ограничить доступ к файлу пользователей сети. Используя настройки атрибутов файла, сделать файл «скрытым».

Задание 2 Защита документов, созданных в Microsoft Excel.

Цель работы: Изучение возможностей современных средств защиты документов, созданных в среде приложений OS Microsoft. Закрепление теоретического материала. Изучение способов и систем защиты файлов и файловых систем.

Учебно-наглядные пособия и ТСО: ПЭВМ, ОС Windows, пакет Microsoft Office, методическое пособие. ЗАДАНИЕ

Создать документ в приложении Excel офисного пакета Microsoft Office Используя свойства и возможности приложения Excel, защитить созданный файл паролем. Используя настройки атрибутов файла, ограничить доступ к файлу пользователей сети. Используя настройки атрибутов файла, сделать файл «скрытым».

Ход выполнения лабораторной работы:

14. Изучить приведенный в методическом описании к лабораторной работе материал. 2. Ответить на контрольные вопросы. 3. Представить отчет по лабораторной работе преподавателю.

Контрольные вопросы:

В Перечислите свойства файлов (документов), создаваемых в MS Office. 2. В чем заключаются особенности файлов (документов), создаваемых в MS Word? 3. Перечислите порядок установки пароля к созданному файлу в MS Word.

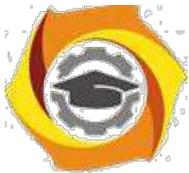
Задание 3 Защита документов, созданных в Microsoft Access. Защита файла паролем.

Цель работы: Изучение возможностей современных средств защиты документов, созданных в среде приложений OS Microsoft. Закрепление теоретического материала. Изучение способов и систем защиты файлов и файловых систем.

Учебно-наглядные пособия и ТСО: ПЭВМ, ОС Windows, пакет Microsoft Office, методическое пособие. ЗАДАНИЕ 1

Создать документ в приложении Access пакета Microsoft Office Используя свойства и возможности приложения Access, защитить созданный файл паролем. Используя настройки атрибутов файла, ограничить доступ к файлу пользователей сети. Используя настройки атрибутов файла, сделать файл «скрытым».

ЗАДАНИЕ 2



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Выполнить программу на одном из языков программирования (например, PASCAL), осуществляющую функцию защиты файла паролем. Ход выполнения задания: 1. Составить алгоритм 2. Использовать условные операторы 3. Создать необходимые циклы, один из которых использует функцию сравнения пароля 1 цикл на запуск программы используя число ввода пароля до 3 4. Завершение программы неудачей, если число ввода неверного пароля превысило $N=3$ 5.

Можете использовать следующие текстовые сообщения (примерные): «ВВЕДИТЕ ПАРОЛЬ ДЛЯ ВХОДА В ПРОГРАММУ» (Начало выполнения загрузки) «ПАРОЛЬ НЕВЕРНЫЙ! ИСПОЛЬЗУЙТЕ ЕЩЕ ОДНУ ПОПЫТКУ» (Если пароль введен некорректно) «ДОБРО ПРЯВЫСИЛИ ДОПУСТИМОЕ ЧИСЛО ПОПЫТОК! ДО СВИДАНИЯ!» (Если количество неверных попыток ввода пароля превысило допустимое число $N=3$)

Ход выполнения лабораторной работы:

ООО Изучить приведенный в методическом описании к лабораторной работе материал. 2. Выполнить распечатку кода программы, оформить отчет в установленной форме по разделам. 3. Ответить на контрольные вопросы. 4. Представить программу по разделу 2 и отчет по лабораторной работе преподавателю.

Контрольные вопросы:

7. Перечислите свойства файлов (документов), создаваемых в MS Office. 2. В чем заключаются особенности файлов (документов), создаваемых в MS Access? 3. Перечислите порядок установки пароля к созданному файлу в MS Access. 4. Назовите назначение основных модулей в блок-схеме алгоритма разработанной Вами программы. 5. При каких условиях в общем случае может быть обеспечен доступ к сетевому ресурсу?

Контрольные вопросы ПЗЗ(УК-1):

10. Основные методы защиты информации в компьютерных сетях.
11. Виды электронной подписи.
12. Что такое методы защиты информации в компьютерных сетях?
13. Симметричное и асимметричное кодирование.
14. Открытый и закрытый коды.
15. Что такое электронная подпись?
16. Виды электронной подписи.



Практическое занятие №4. Разработка алгоритмов линейной, разветвляющейся, циклической структуры. Графическая реализация. ЕСПД.

Линейный алгоритм. Ввод и вывод информации

Теоретическая часть

Алгоритм - это последовательность действий, которые необходимо выполнить, чтобы решить поставленную задачу.

Программа же представляет собой набор команд на языке, понятном исполнителю, реализующий некоторый алгоритм. В нашем случае исполнителем является компьютер, а языком программирования будет язык высокого уровня Pascal. К сожалению, любой язык высокого уровня удобен только человеку, пишущему или отлаживающему программу, но совершенно непонятен компьютеру. Программа на таком языке называется исходным текстом и хранится во внешнем файле с расширением .pas.

Для перевода программы на язык низкого уровня, понятный исполнителю-компьютеру, существуют специальные программы-переводчики - компиляторы. Результатом работы компилятора (иными словами, результатом процесса компиляции) является исполняемый код, который записывается в файл с расширением .exe.

Линейным принято называть вычислительный процесс, в котором этапы вычислений выполняются в линейной последовательности и каждый этап выполняется только один раз. На схеме блоки размещаются сверху вниз в порядке их выполнения. Для таких процессов характерно, что направление вычислений не зависит от исходных данных или промежуточных результатов.

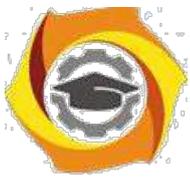
Линейные процессы имеют место, например, при вычислении арифметических выражений.

В состав среды разработчика Turbo Pascal входят:

- текстовый редактор, в котором можно набирать тексты программ;
- компилятор, превращающий исходные тексты в исполняемый код;
- отладчик, помогающий обнаруживать и исправлять ошибки в программе

Из многочисленных возможностей, предоставляемых средой Turbo Pascal, мы упомянем лишь самые важные - те, без которых написание программ становится совсем уж затруднительным.

18. Нажатие клавиш F1, Alt+F1, Ctrl+F1 открывает экранную подсказку.
19. Нажатие клавиши F2 позволяет сохранить исходный текст программы.
20. Нажатие клавиши F3 открывает диалог выбора нужного файла (по умолчанию, отображаются только файлы с расширением .pas).
21. Нажатие клавиши Alt+F5 показывает консоль (см. п. "Ввод и вывод: консоль" ниже) с результатами работы программы.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

к Нажатие клавиши Ctrl+F9 начинает процесс выполнения программы. Если она еще не была откомпилирована, предварительно будет вызван компилятор.

к Клавиши F7 и F8 обеспечивают трассировку - пошаговое выполнение программы, позволяющее проследить за процессом ее выполнения.

к Дополнительное окно Debug/Watch показывает текущее состояние выбранных переменных.

Любая Pascal-программа может состоять из следующих **блоков** (напомним, что квадратными скобками здесь и далее помечены необязательные части):

```
program <имя_программы>;  
  [ uses <имена_подключаемых_модулей>;]  
  [ label <список_меток>;]  
  [ const <имя_константы> = <значение_константы>;]  
  [ type <имя_типа> = <определение_типа>;]  
  [ var <имя_переменной> : <тип_переменной>;]  
  [ procedure <имя_процедуры>  
    <описание_процедуры>;] [ function <имя_функции>  
    <описание_функции>;] begin {начало основного тела  
  программы} <операторы>  
end. (* конец основного тела программы *)
```

Сразу же необходимо сделать важную оговорку: поздние версии компиляторов языка Pascal уже не требуют указывать название программы, то есть строку

```
program <имя_программы>;
```

вообще говоря, можно опустить. Но это возможно только в том случае, если вся программа содержится в одном модуле-файле. Если же программа состоит из нескольких самостоятельных кусков - модулей, то каждый из них должен иметь заголовок (program или unit).

Комментарии

Помимо отступов, большие логически замкнутые блоки программы удобно разделять строками-комментариями, содержащими информацию о смысле последующего блока. Комментарий - это строка (или несколько строк) из произвольных символов, заключенная в фигурные скобки:

```
{ комментарий }
```

Другой вариант оформления комментария:

```
(* комментарий *)
```

Внутри самого комментария символы } или *) встречаться не должны.

Во время компиляции программы комментарии игнорируются. Следовательно, их можно добавлять в любом месте программы. Можно даже разорвать оператор вставкой комментария. Кроме того, все, что находится после ключевого слова end., завершающего текст программы, компилятор тоже воспринимает как комментарий.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Переменные и типы данных

Переменная - это программный объект, значение которого может изменяться в процессе работы программы.

Тип данных - это характеристика диапазона значений, которые могут принимать переменные, относящиеся к этому типу данных.

Все используемые в программе переменные должны быть описаны в специальном разделе **var** по следующему шаблону:

```
var <имя_переменной_1> [, <имя_переменной_2, _>] : <имя_типа_1>;
    <имя_переменной_3> [, <имя_переменной_4, _>] : <имя_типа_2>;
```

Для удобства программистов в языке Pascal существует множество стандартных типов данных и плюс к тому возможность создавать новые типы.

Конструируя новые типы данных на основе уже имеющихся (стандартных или опять-таки определенных самим программистом), нужно помнить, что любое здание должно строиться на хорошем фундаменте. Поэтому сейчас мы и поговорим об этом "фундаменте".

На основании базовых типов данных строятся все остальные типы языка Pascal, которые так и называются: конструируемые.

Разделение на базовые и конструируемые типы данных в языке Pascal

показано в таблице:

	Порядковые (дискретные) типы данных					
		Арифметические типы данных			Адресные типы данных	Структурированные типы данных
		Целые	Вещественные			
Базовые типы данных	Логический boolean	Символьный (литерный) char	shortint t byte integer word longint	real single double extended comp	Нетипизированный указатель pointer	
Конструируемые типы		Перечисляемый week = (su, mo, tu, we, th, fr, sa);			Типизированный указатель ^<тип>	Массив array Строка string Запись record Файл text file Процедурный Объектный
	Типы данных, конструируемые программистом					

Типы данных, конструируемые программистом, описываются в разделе **type** по следующему шаблону:



```
type <имя_типа> = <описание_типа>;
```

Например:

```
type lat_bukvy = 'a'..'z','A'..'Z';
```

Базовые типы данных являются стандартными, поэтому нет нужды описывать их в разделе type. Однако при желании это тоже можно сделать, например, дав длинным определениям короткие имена. Скажем, введя новый тип данных

```
type int = integer;
```

можно немного сократить текст программы.

Порядковые типы данных

Среди базовых типов данных особо выделяются порядковые типы. Такое название можно обосновать двояко:

5. Каждому элементу порядкового типа может быть сопоставлен уникальный (порядковый) номер. Нумерация значений начинается с нуля. Исключение - типы данных `shortint`, `integer` и `longint`. Их нумерация совпадает со значениями элементов.

7. Кроме того, на элементах любого порядкового типа определен порядок (в математическом смысле этого слова), который напрямую зависит от нумерации. Таким образом, для любых двух элементов порядкового типа можно точно сказать, который из них меньше, а который - больше.

Стандартные подпрограммы, обрабатывающие порядковые типы данных

Только для величин порядковых типов определены следующие функции и процедуры:

5. Функция **ord(x)** возвращает порядковый номер значения переменной `x` (относительно того типа, к которому принадлежит переменная `x`).

6. Функция **pred(x)** возвращает значение, предшествующее `x` (к первому элементу типа неприменима).

7. Функция **succ(x)** возвращает значение, следующее за `x` (к последнему элементу типа неприменима).

8. Процедура **inc(x)** возвращает значение, следующее за `x` (для арифметических типов данных это эквивалентно оператору `x:=x+1`).

9. Процедура **inc(x,k)** возвращает `k`-е значение, следующее за `x` (для арифметических типов данных это эквивалентно оператору `x:=x+k`).

10. Процедура **dec(x)** возвращает значение, предшествующее `x` (для арифметических типов данных это эквивалентно оператору `x:=x-1`).

11. Процедура **dec(x,k)** возвращает `k`-е значение, предшествующее `x` (для арифметических типов данных это эквивалентно оператору `x:=x-k`).

Целочисленные типы данных

Тип данных	Количество		Диапазон
	байтов	битов	
<code>shortint</code>	1	8	-128..127



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

byte	1	8	0..255	$0..2^8-1$
integer	2	16	-32768..32767	$-2^{15}..2^{15}-1$
word	2	16	0..65535	$0..2^{16}-1$
longint	4	32	-2147483648..2147483647	$-2^{31}..2^{31}-1$

Вещественные типы данных

Напомним, что эти типы данных являются арифметическими, но не порядковыми.

Тип	Количество байтов	Диапазон (абсолютной величины)
single	4	$1.5 \cdot 10^{-45} .. 3.4 \cdot 10^{38}$
real	6	$2.9 \cdot 10^{-39} .. 1.7 \cdot 10^{38}$
double	8	$5.0 \cdot 10^{-324} .. 1.7 \cdot 10^{308}$
extended	10	$3.4 \cdot 10^{-4932} .. 1.1 \cdot 10^{4932}$
comp	8	$-2^{63}+1 .. 2^{63}-1$

Стандартные арифметические функции

арифметическим операциям примыкают и стандартные арифметические функции. Их список

5. кратким описанием мы приводим в таблице.

	Описание	Тип аргумента	Тип результата
abs(x)	Абсолютное значение (модуль) числа	Арифметический	Совпадает с типом аргумента
arctan(x)	Арктангенс (в радианах)	Арифметический	Вещественный
cos(x)	Косинус (в радианах)	Арифметический	Вещественный
exp(x)	Экспонента (e^x)	Арифметический	Вещественный
frac(x)	Взятие дробной части числа	Арифметический	Вещественный
int(x)	Взятие целой части числа	Арифметический	Вещественный
ln(x)	Натуральный логарифм (по основанию e)	Арифметический	Вещественный
odd(x)	Проверка нечетности числа	Целый	boolean
pi	Значение числа	-	Вещественный
round(x)	Округление к ближайшему целому	Арифметический	Целый
trunc(x)	Округление "вниз" - к ближайшему меньшему целому	Арифметический	Целый
sin(x)	Синус (в радианах)	Арифметический	Вещественный
sqr(x)	Возведение в квадрат	Арифметический	Вещественный
sqrt(x)	Извлечение квадратного корня	Арифметический	Вещественный

Операторы

Оператором называется (минимальная) структурно законченная единица программы.

Важно! Все операторы языка Pascal должны заканчиваться знаком ";" (точка с запятой), и ни один оператор не может разрываться этим знаком. Единственная возможность не ставить после оператора ";" появляется в том случае, когда сразу за этим оператором следует ключевое слово end.

7. простейшим операторам языка Pascal относятся:

a:= b; - присваивание переменной a значения переменной b. В правой части присваивания может находиться переменная, константа, арифметическое выражение или вызов функции.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

5. ; - пустой оператор, который можно вставлять куда угодно, а также вычеркивать откуда угодно, поскольку на целостность программы это никак не влияет.

6. Операторные скобки, превращающие несколько операторов в один:

7. begin

8. <несколько операторов>

end;

Везде далее, где в записи конструкций языка Pascal мы будем использовать обозначение <один_оператор>, его следует понимать как "один оператор или несколько операторов, заключенные в операторные скобки begin - end".

Ввод и вывод: консоль

Любой алгоритм должен быть результативным. В общем случае это означает, что он должен сообщать результат своей работы потребителю: пользователю-человеку или другой программе (например, программе управления принтером). Мы не будем описывать здесь внутренние автоматические процессы, использующие сигналы непрерывно функционирующих программ, а сосредоточим внимание на взаимодействии программы и человека, то есть на процессах ввода информации с клавиатуры и вывода ее на экран.

5. программировании существует специальное понятие консоль, которое обозначает клавиатуру при вводе и монитор при выводе.

Ввод с консоли

Для того чтобы получить данные, вводимые пользователем вручную (то есть с консоли), применяются команды

read(<список_ввода>) и readln(<список_ввода>).

Первая из этих команд считывает все предложенные ей данные, оставляя курсор в конце последней строки ввода, а вторая - сразу после окончания ввода переводит курсор на начало следующей строки. В остальном же их действия полностью совпадают.

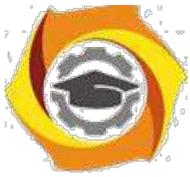
Список ввода - это последовательность имен переменных, разделенных запятыми.

Например, при помощи команды

```
readln(k,x,c,s); {k:byte; x:real; c:char; s:string}
```

программа может получить с клавиатуры данные сразу для четырех переменных, относящихся к различным типам данных.

Вводимые значения необходимо разделять пробелами, а завершать ввод - нажатием клавиши Enter. Ввод данных заканчивается в тот момент, когда последняя переменная из списка ввода получила свое значение. Следовательно, вводя данные при помощи приведенной выше команды, вы можете нажать Enter четыре раза - после каждой из вводимых переменных, - либо же только один раз,



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

предварительно введя все четыре переменные в одну строчку (обязательно нужно разделить их пробелами).

Типы вводимых значений должны совпадать с типами указанных переменных, иначе возникает ошибка. Поэтому нужно внимательно следить за правильностью вводимых данных.

Вообще, вводить с клавиатуры можно только данные базовых типов (за исключением логического). Если же программе все-таки необходимо получить с консоли значение для boolean-величины, придется действовать более хитро: вводить оговоренный символ, а уже на его основе присваивать логической переменной соответствующее значение. Например:

```
repeat
  writeln('Согласны ли Вы с этим утверждением? y - да, n - нет');
  readln(c);      {c:char}
  case c of
    'y': b:= true;
    'n': b:= false;
    else writeln('Ошибка!');
  end;
until (c='n')or(c='y');
```

Второе исключение: строки, хотя они и не являются базовым типом, вводить тоже разрешается.

Признаком окончания ввода строки является нажатие клавиши Enter, поэтому все следующие за нею переменные необходимо вводить с новой строчки.

Вывод на консоль

Сделаем одно важное замечание: ожидая от человека ввода с клавиатуры, не нужно полагать, что он окажется ясновидящим и просто по мерцанию курсора на черном экране догадается, какого типа переменная нужна ожидающей программе. Старайтесь всегда придерживаться правила: "лысый" ввод недопустим! Перед тем как считывать что-либо с консоли, необходимо сообщить пользователю, что именно он должен ввести: смысл вводимой информации, тип данных, максимальное и минимальное допустимые значения и т.п.

Примером неплохого приглашения служит, скажем, такая строчка:

Введите два вещественных числа ($0.1 < x, y < 1000000$) - длины катетов.

Впрочем, и ее можно улучшить, сообщив пользователю не только допустимый диапазон ввода, но и ожидаемую точность (количество знаков после запятой).

Средства, позволяющие организовать выдачу информации на экран, мы здесь и рассмотрим.

Для того чтобы вывести на экран какое-либо сообщение, воспользуйтесь процедурой

`write(<список_вывода>)` или `writeln(<список_вывода>)`.

Первая из них, напечатав на экране все, о чем ее просили, оставит курсор в конце выведенной строки, а вторая переведет его в начало следующей строчки.

Список вывода может состоять из нескольких переменных, записанных через запятую; все эти переменные должны иметь тип либо базовый, либо строчный. Например, `writeln(a,b,c);`



Форматный вывод

Если для вывода информации воспользоваться командой, приведенной в конце предыдущего пункта, то выводимые символы окажутся "слеplенными". Чтобы этого не случилось, нужно либо позаботиться о пробелах между выводимыми переменными:

```
writeln(a,' ',b,' ',c);
```

либо задать для всех (или хотя бы для некоторых) переменных формат вывода:

```
writeln(a:5,b,c:20:5);
```

Первое число после знака ":" обозначает количество позиций, выделяемых под всю переменную, а второе - под дробную часть числа. Десятичная точка тоже считается отдельным символом.

Если число длиннее, чем отведенное под него пространство, количество позиций будет автоматически увеличено. Если же выводимое число короче заданного формата, то спереди к нему припишутся несколько пробелов. Таким образом можно производить вывод красивыми ровными столбиками, а также следить за тем, чтобы переменные не сливались.

Например, если $a = 25$, $b = 'x'$, а $c = 10.5$, то после выполнения команды `writeln(a:5,'b,c:10:5)` на экране или в файле будет записано следующее (подчерки в данном случае служат лишь для визуализации пробелов):

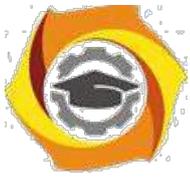
```
__ _ 25 _x_ _ 10.50000
```

Особенно важен формат при выводе вещественных переменных. К примеру, если не указать формат, то число 10.5 будет выведено как 1.0500000000E+0001. Такой формат называется записью с плавающей точкой.

Если же задать только общую длину вещественного числа, не указывая длину дробной части, то оно будет занимать на экране заданное количество символов (в случае надобности, спереди будет добавлено соответствующее количество пробелов), но при этом останется в формате плавающей точки. Минимальной длиной для вывода вещественных чисел является 10 (при формате `_x.xE+uuuu`). Первая позиция зарезервирована под знак "-".

Необходимо помнить, что в случае недостаточной длины вывода число будет автоматически округлено, например (подчерк служит для визуализации пробела):

Оператор форматного вывода	Результат вывода на экран
<code>write (125.2367:10);</code>	<code>_ 1.3E+0002</code>
<code>write (125.2367:11);</code>	<code>_ 1.25E+0002</code>
<code>write (125.2367:12);</code>	<code>_ 1.252E+0002</code>
<code>write (125.2367:13);</code>	<code>_ 1.2524E+0002</code>
<code>write (125.2367:14);</code>	<code>_ 1.25237E+0002</code>
<code>write (125.2367:15);</code>	<code>_ 1.252367E+0002</code>
<code>write (125.2367:16);</code>	<code>_ 1.2523670E+0002</code>



Пример простейшей программы на языке Pascal

```
program start;  
var s: string;  
begin  
  write('Пожалуйста, введите Ваше имя: ');  
  readln(s);  
  writeln('Мы рады Вас приветствовать, ',s,'!');  
end.
```

Во время работы этой программы на экране появится примерно следующее:

Пожалуйста, введите Ваше имя: Иван Иванович

Мы рады Вас приветствовать, Иван Иванович!

Практическая часть

5. Пусть даны длины сторон треугольника. Вычислите его площадь.
6. Вычислите расстояние между двумя точками на плоскости с данными координатами (x_1, y_1) и (x_2, y_2) .
7. Введите положительное число **a**. Вычислите:
площадь равностороннего треугольника со стороной **a**; площадь квадрата со стороной **a**; площадь круга с диаметром **a**.
5. Вычислите длину окружности, площадь круга, объем шара заданного радиуса.
6. Пусть даны числа a, b, c . Найти площадь треугольника, две стороны которого равны a, b , а угол между этими сторонами равен c . Считать, что c – это: радианная мера; градусная мера.

2. Пусть известны длины сторон a, b, c треугольника. Вычислите высоты этого треугольника по формулам: $h_a = \frac{2}{a} \sqrt{p(p-a)(p-b)(p-c)}$ где $p = \frac{a+b+c}{2}$

$$h_b = \frac{2}{b} \sqrt{p(p-a)(p-b)(p-c)} \quad h_c = \frac{2}{c} \sqrt{p(p-a)(p-b)(p-c)}$$

И Даны координаты вершин некоторого треугольника. Вычислить его периметр.

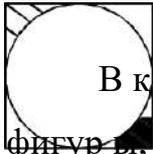


Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

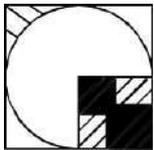
В квадрат вписана окружность. Определить площадь заштрихованной части фигуры, если известна длина сторон квадрата.



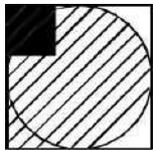
В квадрат вписана окружность. Определить площадь заштрихованной части фигуры, если известен радиус окружности.



В квадрат вписана окружность. Определить площадь заштрихованной части фигуры, если известна длина стороны квадрата.



В квадрат вписана окружность. Определить площадь заштрихованной части фигуры, если известна длина стороны квадрата.



Даны два ненулевых числа. Найти их сумму, разность, произведение и частное.

Даны два числа. Найти среднее арифметическое их квадратов и среднее арифметическое их модулей.

Найти периметр и площадь прямоугольного треугольника, если даны длины его катетов a и b .

Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.

Найти длину окружности и площадь круга заданного радиуса R . В качестве значения π использовать 3.14.

Найти площадь кольца, внутренний радиус которого равен R_1 , а внешний радиус равен R_2 ($R_1 < R_2$). В качестве значения π использовать 3.14.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Дана сторона равностороннего треугольника. Найти площадь этого треугольника и радиусы вписанной и описанной окружностей.

Дана площадь круга. Найти длину окружности, ограничивающей этот круг.
· качестве значения π использовать 3.14.

Найти периметр и площадь равнобедренной трапеции с основаниями a и b ($a > b$) и углом α при большем основании (угол дан в радианах).

Найти периметр и площадь прямоугольной трапеции с основаниями a и b ($a > b$) и острым углом α (угол дан в радианах).

Определить объём и площадь боковой поверхности цилиндра с заданными радиусом основания R и высотой H .

Вычислите длину окружности, площадь круга и объём шара одного и того же заданного радиуса.

По координатам трёх вершин некоторого треугольника найдите его площадь и периметр.

Вычислите периметр и площадь прямоугольного треугольника по длинам двух его катетов

Зная два катета прямоугольного треугольника, найти гипотенузу и углы треугольника.

27. Известна гипотенуза c и прилежащий угол α прямоугольного треугольника. Найти площадь треугольника.

и Диагональ квадрата равна d . Вычислить площадь и периметр квадрата.

и Площадь квадрата равна S . Вычислить сторону квадрата a , диагональ d и площадь S_1 описанного вокруг квадрата круга.

30. В равнобедренном треугольнике известно основание c и угол при нем α . Найти площадь треугольника S и величину боковой стороны a .

· Известна диагональ ромба. Вычислить его площадь и периметр.

· Задан периметр квадрата. Вычислить его сторону, диагональ и площадь.

· В равнобедренном треугольнике известно основание c и высота h . Найти площадь и периметр треугольника.

· Известна гипотенуза c и противолежащий угол β прямоугольного треугольника. Найти периметр треугольника.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- В прямоугольном треугольнике известен катет b и площадь S . Вычислить периметр треугольника.
- в прямоугольном треугольнике известен катет b и площадь S . Вычислить величину гипотенузы c и второго катета a .

Контрольные вопросы ПЗ4(УК-1):

- В Какими свойствами обладает алгоритм?
- В Что понимается под результативностью алгоритма?
- В В чем заключается свойство массовости?
- В Что означает свойство определенности алгоритма?
- В Какими элементами задается алгоритм?
- В Что представляет собой «исполнитель алгоритма»?
- В Что является основной характеристикой исполнителя?
- В Что относится к среде исполнителя?
- В Поясните общую методику записи содержания алгоритма.
- В Какие существуют формы представления алгоритмов?
- В Что представляет собой словесный способ записи алгоритма?
- В Что называется блок-схемой алгоритма?
- В Поясните особенности составления блок-схемы алгоритма.
- В Когда используется межстраничный соединитель?
- В Что представляет собой псевдокод?



*Практическое занятие №5. Понятия об основных методах ввода/вывода данных.
Разработка программ линейных алгоритмов. Понятия об основных методах
отладки (тестирования) программы.*

Условные конструкции

Теоретическая часть

К операторам, позволяющим из нескольких возможных вариантов выполнения программы (ветвей) выбрать только один, относятся `if` и `case`.

Условный оператор `if`

Оператор `if` выбирает между двумя вариантами развития событий:

```
if <условие>  
  then <один_оператор>  
  [else <один_оператор>];
```

Обратите внимание, что перед словом `else` (когда оно присутствует, конечно же) символ ";" не ставится - ведь это разорвало бы оператор на две части.

Условный оператор `if` работает следующим образом:

- Сначала вычисляется значение <условия> - это может быть любое выражение, возвращающее значение типа `boolean`.

В Затем, если в результате получена "истина" (`true`), то выполняется оператор, стоящий после ключевого слова `then`, а если "ложь" (`false`) - без дополнительных проверок выполняется оператор, стоящий после ключевого слова `else`. Если же `else`-ветвь отсутствует, то не выполняется ничего.

Что же произойдет, если написать несколько вложенных операторов `if`?

в случае, когда каждый оператор `if` имеет собственную `else`-ветвь, все будет в порядке. А вот если некоторые из них этой ветви не имеют, может возникнуть ошибка. Компилятор языка `Pascal` всегда считает, что `else` относится к самому ближайшему оператору `if`. Таким образом, если написать

```
if i>0 then if s>2  
  then s:= 1  
  else s:= -1;
```

подразумевая, что `else`-ветвь относится к внешнему оператору `if`, то компилятор все равно воспримет эту запись как

```
if i>0 then if s>2  
  then s:= 1  
  else s:= -1  
  else;
```

Ясно, что таким образом правильного результата получить не удастся.

Для того чтобы избежать подобных ошибок, стоит всегда (или по крайней мере при наличии нескольких вложенных условных операторов) указывать оба ключевых слова, даже если одна из



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

ветвей будет пустовать. Так вы застрахуетесь от одной из частых "ошибок по невнимательности", которые очень сложно найти в процессе отладки программы.

Итак, исходный вариант нужно переписать следующим образом:

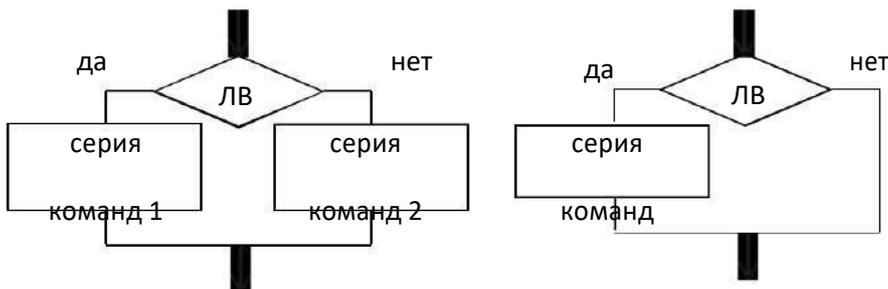
```
if i>0 then if s>2  
    then s:=1  
    else  
    else s:=-1;
```

либо так:

```
if i>0 then begin if s>2  
    then s:=1  
end  
else s:=-1;
```

Вообще же, если есть возможность переписать несколько вложенных условных операторов как один оператор выбора, это стоит сделать.

Изображение условного оператора в виде блок-схемы выглядит следующим образом: (слева – полное ветвление *если-то-иначе*, справа – неполный вариант ветвления *если-то*)



Практическая часть

к Даны три целых числа. Возвести в квадрат отрицательные числа и в третью степень — положительные (число 0 не изменять).

Из трех данных чисел выбрать наименьшее.

Из трех данных чисел выбрать наибольшее.

Из трех данных чисел выбрать наименьшее и наибольшее.

Перераспределить значения переменных X и Y так, чтобы в X оказалось меньшее из этих значений, а в Y — большее.

Даны две переменные целого типа: A и B. Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

О Даны две переменные целого типа: А и В. Если их значения не равны, то присвоить каждой переменной максимальное из этих значений, а если равны, то присвоить переменным нулевые значения.

О Даны три переменные: X, Y, Z. Если их значения упорядочены по убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное.

О Даны три переменные: X, Y, Z. Если их значения упорядочены по возрастанию или убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное.

О Даны целочисленные координаты точки на плоскости. Если точка не лежит на координатных осях, то вывести 0. Если точка совпадает с началом координат, то вывести 1. Если точка не совпадает с началом координат, но лежит на оси OX или OY, то вывести соответственно 2 или 3.

В Даны вещественные координаты точки, не лежащей на координатных осях OX и OY. Вывести номер координатной четверти, в которой находится данная точка.

В На числовой оси расположены три точки: А, В, С. Определить, какая из двух последних точек (В или С) расположена ближе к А, и вывести эту точку и ее расстояние от точки А.

В Даны четыре целых числа, одно из которых отлично от трех других, равных между собой. Вывести порядковый номер этого числа.

В Дан номер некоторого года (положительное целое число). Вывести соответствующий ему номер столетия, учитывая, что, к примеру, началом 20 столетия был 1901 год.

В Дан номер некоторого года (положительное целое число). Вывести число дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются).

3) Для данного x вычислить значение следующей функции f , вещественные значения: -1 если $x \leq 0$, $f(x) = x$ если $0 < x < 2$, 4 если $x \geq 2$.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Для данного x вычислить значение следующей функции f , принимающей значения целого типа: 0 , если $x < 0$, $f(x) = 1$, если x принадлежит $[0,1)$, $f(x) = [2,3)$, ... , -1 если x принадлежит $[1,2)$, $f(x) = [3,4)$,

Дано целое число, лежащее в диапазоне от -999 до 999 . Вывести строку — словесное описание данного числа вида "отрицательное двузначное число", "нулевое число", "положительное однозначное число" и т.д.

в Дано целое число, лежащее в диапазоне от 1 до 9999 . Вывести строку — словесное описание данного числа вида "четное двузначное число", "нечетное четырехзначное число" и т.д.

20. Два прямоугольника заданы длинами сторон. Определите, можно ли первый прямоугольник целиком разместить во втором.

21. Решите квадратное уравнение $ax^2 + bx + c = 0$

22. Составить программу, печатающую значение `true`, если указанное высказывание является истинным, и `false` в противном случае: данная тройка натуральных чисел a, b, c не является тройкой Пифагора, т.е. $c^2 = a^2 + b^2$

23. Составить программу, печатающую значение `true`, если указанное высказывание является истинным, и `false` в противном случае: данные числа x, y являются координатами точки, лежащей в первой координатной четверти.

4. Составить программу, печатающую значение `true`, если указанное высказывание является истинным, и `false` в противном случае: данная тройка натуральных чисел a, b, c является тройкой Пифагора, т.е. $c^2 = a^2 + b^2$.

5. Составить программу, печатающую значение `true`, если указанное высказывание является истинным, и `false` в противном случае: число c является средним арифметическим чисел a и b .

6. Составить программу, печатающую значение `true`, если указанное высказывание является истинным, и `false` в противном случае: число c является средним геометрическим чисел a и b .

7. Составить программу, печатающую значение `true`, если указанное высказывание является истинным, и `false` в противном случае: сумма двух натуральных чисел кратна 2 .



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

4. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: произведение натуральных чисел a и b кратно числу c .
5. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: данное натуральное число a кратно числу b , но не кратно числу c .
6. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: числа a и b выражают длины катетов одного прямоугольного треугольника, c и d — другого. Эти треугольники являются подобными.
7. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: данные числа c и d являются соответственно квадратом и кубом числа a .
6. Определите, какая из точек $M_1(x_1, y_1)$ или $M_2(x_2, y_2)$ расположена ближе к началу координат. Укажите координаты этой точки.
7. Определите, какая из двух фигур (круг или квадрат) имеет большую площадь. Известно, что сторона квадрата равна a , радиус круга r . Выведите название большей фигуры и её площадь.
8. Определите, попадает ли точка $M(x, y)$ в круг радиусом R с центром в точке (x_0, y_0) .
8. Заданы площади круга и квадрата. Определите, поместится ли квадрат в круге.
9. Заданы площади круга и квадрата. Определите, поместится ли круг в квадрате.

Контрольные вопросы ПЗ5(УК-1):

- У Понятия об основных методах ввода/вывода данных.
- У Разработка программ линейных алгоритмов.
- У Понятия об основных методах отладки (тестирования) программы.
- У Что такое ввод данных?
- У Что такое вывод данных?
- У Что такое область значений данных?
- У Что такое тип данных?
- У Что такое отладка программ?
- У Что такое контрольный вариант расчета ?



Практическое занятие №6 Разработка программ алгоритмов разветвляющейся и циклической структуры.

Оператор case позволяет сделать выбор между несколькими вариантами:

```
case <переключатель> of
    <список_констант> : <один_оператор>;
    [<список_констант> : <один_оператор>;]
    [<список_констант> : <один_оператор>;]
    [else <один_оператор>;]
end;
```

Замечание: Обратите внимание, что после else двоеточие не ставится.

Существуют дополнительные правила, относящиеся к структуре этого оператора:

1. Переключатель должен относиться только к порядковому типу данных, но не к типу longint.
2. Переключатель может быть переменной или выражением.
3. Список констант может задаваться как явным перечислением, так и интервалом или их объединением.
4. Повторение констант не допускается.
5. Тип переключателя и типы всех констант должны быть совместимыми.

Пример оператора выбора:

```
case symbol(* :char *) of
    'a'..'z', 'A'..'Z' : writeln('Это латинская буква');
    'а'..'я', 'А'..'Я' : writeln('Это русская буква');
    '0'..'9' : writeln('Это цифра');
    ' ',#10,#13,#26 : writeln('Это пробельный символ');
    else writeln('Это служебный символ'); end;
```

Выполнение оператора case происходит следующим образом:

вычисляется значение переключателя;
полученный результат проверяется на принадлежность к тому или иному списку констант;

если такой список найден, то дальнейшие проверки уже не производятся, а выполняется оператор, соответствующий выбранной ветви, после чего управление передается оператору, следующему за ключевым словом end, которое закрывает всю конструкцию case.

если подходящего списка констант нет, то выполняется оператор, стоящий за ключевым словом else. Если else-ветви нет, то не выполняется ничего.



Практическая часть

По заданному номеру месяца m вывести на печать его название.

Дан номер месяца (1 — январь, 2 — февраль, ...). Вывести название соответствующего времени года ("зима", "весна" и т.д.).

Дан номер месяца (1 — январь, 2 — февраль, ...). Вывести число дней этом месяце для невисокосного года.

Дано целое число в диапазоне 0 – 9. Вывести строку — название соответствующей цифры на русском языке (0 — "ноль", 1 — "один", 2 — "два", ...).

Дано целое число в диапазоне 1 – 5. Вывести строку — словесное описание соответствующей оценки (1 — "плохо", 2 — "неудовлетворительно", 3 — "удовлетворительно", 4 — "хорошо", 5 — "отлично").

Вычислить значение функции y по одной из формул: x^{10}/g , $g_2 \cdot x + 6 \cdot x - g_3 \cdot (\cos x + g - x)^2$, причем x и g вводятся с клавиатуры

Арифметические действия над числами пронумерованы следующим образом: 1 — сложение, 2 — вычитание, 3 — умножение, 4 — деление. Дан номер действия и два числа A и B (B не равно нулю). Выполнить над числами указанное действие и вывести результат.

Единицы длины пронумерованы следующим образом: 1 — дециметр, 2 — километр, 3 — метр, 4 — миллиметр, 5 — сантиметр. Дан номер единицы длины и длина отрезка L в этих единицах (вещественное число). Вывести длину данного отрезка в метрах.

- Единицы массы пронумерованы следующим образом: 1 — килограмм, 2 — миллиграмм, 3 — грамм, 4 — тонна, 5 — центнер. Дан номер единицы массы масса тела M в этих единицах (вещественное число). Вывести массу данного тела в килограммах.

Вычислить значение функции y по одной из формул: \sqrt{x} , x^6 ; $\cos x + x^7$
3. причем x и g вводятся с клавиатуры.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

7. Автобус остановился на развилке дорог. Он может поехать в четырех направлениях (1 — север, 2 — запад, 3 — юг, 4 — восток). Дано число N — заданное направление. Вывести, куда поедет автобус.

- Элементы окружности пронумерованы следующим образом: 1 — радиус (R), 2 — диаметр (D), 3 — длина (L), 4 — площадь круга (S). Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данной окружности (в том же порядке). В качестве значения P_i использовать 3.14.

13. Вычислить значение функции y по одной из формул: $x^5 - 10$, $x^{9 \cdot x}$; $\sqrt{x^2} - \sqrt{x}$, причем x вводится с клавиатуры.

Элементы равнобедренного прямоугольного треугольника пронумерованы следующим образом: 1 — катет (a), 2 — гипотенуза (c), 3 — высота, опущенная на гипотенузу (h), 4 — площадь (S). Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данного треугольника (в том же порядке).

- Элементы равностороннего треугольника пронумерованы следующим образом: 1 — сторона (a), 2 — радиус вписанной окружности (R_1), 3 — радиус описанной окружности (R_2), 4 — площадь (S). Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данного треугольника (в том же порядке).

- Вычислить значение функции y по одной из формул: x^2 , $2 \cdot x + 3 \cdot x^2$; $(\sin x + \cos x)^2$, причем x вводится с клавиатуры.

С При введении времени года вывести общее количество дней данного времени года.

С Дано целое число в диапазоне 20 – 30, определяющее возраст (в годах). Вывести строку — словесное описание указанного возраста, обеспечив правильное согласование числа со словом "год", например: 20 — "двадцать лет", 21 — "двадцать один год", 22 — "двадцать два года".

в Определить количество дней в месяце високосного года по номеру месяца.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

С По заданному названию времени года вывести на печать название месяцев для данного времени года.

21. Вычислить значение функции y по одной из формул: g^3 , $2 \cdot g \cdot e_x - h$; $5 \cdot (\cos x + h)^2$, причем x , h и g вводятся с клавиатуры.

Написать программу, которая анализирует данные о возрасте и относит человека к одной из четырех групп: дошкольник, ученик, работник, пенсионер. Возраст вводится с клавиатуры.

Написать программу, которая по номеру дня недели (целому числу от 1 до 7) выдает в качестве результата количество уроков в вашем классе в этот день.

24. Написать программу, которая по вводимому числу от 1 до 11 (номеру класса) выдает соответствующее значения «Привет, k -классник». Например, если $k=4$, то «Привет, четвероклассник».

25. Вычислить значение функции y по одной из формул: g ; $g \cdot x$; $g \cdot \blacksquare$, причем x и g вводятся с клавиатуры.

26. Вычислить значение функции y по одной из формул: g^2 , $g \cdot e_x + h$; $g \cdot (\sin x + h)^2$, причем x , h и g вводятся с клавиатуры.

Дан номер месяца (1 — январь, 2 — февраль, ...). Вывести число дней этом месяце для високосного года.

С клавиатуры вводится символ. Определить что это за символ (заглавная буква, строчная буква или цифра).

с Дан номер курса k (1 – первый курс, 2 – второй курс, ...). Вывести на экран, сколько лет вам осталось учиться в нашем колледже. Например, если $k=1$, то «Осталось учиться 3 года».

30. Вычислить значение функции y по одной из формул: x^3+1 , $x^{2 \cdot x}$; $\sqrt{x} + \frac{1}{x}$, причем x вводится с клавиатуры.

31. При введении времени года вывести его месяцы и количество дней в каждом.

32. Даны номера названий времен года (1 — зима, 2 — весна, 3 — лето, 4 — осень). Вывести названия месяцев, соответствующих каждому времени года.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

33. Вычислить значение функции y по одной из формул: g^4 , $g \cdot e_{2x} + 2 \cdot h$;

$4 \cdot g \cdot (\sin x + h)^2$, причем x , h и g вводятся с клавиатуры.

В Написать программу, которая по вводимому числу от 1 до 4 (номеру курса) выдает соответствующее значения «Привет, ты учишься k -м курсе, до получения диплома осталось $4-k$ года (лет)». Например, если $k=1$, то «Привет, ты учишься 1-м курсе, до получения диплома осталось 3 года (лет)».

В Арифметические действия над числами пронумерованы следующим образом: 1 — возведение в степень 2, 2 — вычисление квадратного корня, 3 —

умножение на 5, 4 — деление на 10. Дан номер действия и число A .

Выполнить указанное действие и вывести результат.

36. Написать программу, которая по номеру дня недели (целому числу от

1 до 7) выдает в качестве результата перечень уроков в вашем классе в этот день.

**Методические указания к
ЛАБОРАТОРНЫМ РАБОТАМ
по дисциплине**

**Информационные технологии и программирование
(Основы алгоритмизации и программирования)
Методические указания к лабораторной работе 1**

Исследование технологий создания и использования массивов в языке

ТП 7.0

2.1.1 Цели занятия

Выработать навыки составления программ для создания и обработки массивов данных.

2.1.2 Теоретические основы и пример выполнения



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Массив – набор переменных, имеющих одно и то же базовое имя и отличающихся друг от друга числовым признаком (индексом).

Таким образом, массив – средство, позволяющее удобно размещать в памяти большое количество однородной информации.

Описание массива:

VAR Имя массива : ARRAY[нач.индекс .. конеч.индекс] OF тип данных ;

Индексы – любой дискретный тип, обычно Integer.

Обращение к элементам массива – по имени и индексу:

Mes: Array [1..12] of integer;

Mes[1]:=31; mes[2]:=28; mes[3]:=31; ... mes[12]:=31;

Примеры описания массивов:

Type T1 = array[1..100] of real;

Day=array[1..365] of integer;

Table=array [char] of boolean;

Var temp: T1;

S: table;

Примеры обращения к элементам массива:



```
Temp[1]:=2.5;
```

```
temp[i]:=(temp [i-1]+temp[i+1])/2;
```

```
S['a']:= true;
```

Свойства массивов:

19. Каждый компонент массива может быть явно обозначен и к нему имеется прямой доступ.
20. Число компонентов массива определяется при его описании и в дальнейшем не меняется.
21. В качестве индекса можно использовать любое выражение, имеющее тип, совместимый с типом индекса.

Элементы массива размещаются в памяти последовательно, друг за другом. Каждому элементу отводится в памяти столько места, сколько простой переменной того же типа: `arr: array[1..3] of integer;`

Многомерные массивы

В качестве элементов массива может быть значения любого типа. В частности, ими могут быть другие массивы.

Пример описания многомерного массива:

```
Var V2: array[1..10] of array[1..20] of integer;
```

```
V2: array[1..10, 1..20] of integer;
```

Таким образом, массив V2 можно рассматривать двояко:

массив, каждый элемент которого представляет, в свою очередь, массив типа `integer`;

один двумерный массив (матрица).

обращения к элементам массива будут разными:

A) `V2: array[1..10] of array[1..20] of byte;`



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

V2[1] (первый массив с элементами от 1 до 20)



V2[5] (пятый массив с элементами от 1 до 20)

И т.д.

Для адресации, например, к 5-му элементу массива V2[5], нужно записать V2[5][5].

Б). V2: array[1..10, 1..20] of integer;

Паскаль допускает единственное возможное действие над массивом в целом: использование его в операторе присваивания. При этом типы обоих массивов в этом случае д.б. одинаковы.

```
Type Vector = array[1..10] of word;
```

```
Var Vect1, Vect2: Vector;
```

```
Vect1:=Vect2;
```

```
Vect1[i+1]:=100;
```

Элемент массива считается переменной. Он может получать значения в операторе присваивания, а также участвовать в выражениях, допустимых для данного типа переменных.

Типы массивов. Особенности использования массивов в программах.

Типичные ошибки при использовании массивов

Пример 1:

```
Program pr1;
```

```
Var vect: array [1..5] of integer;
```

```
Sum, i : integer;
```

```
Begin
```

```
For i:=0 to 5 DO sum:= vect[i]+sum {не существует элем. Vect[0]} End.
```

Компилятор выдаст: Error201: Range check error.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Пример 2: Просуммировать все элементы массива.



Program error2;

Var Vect: array[1..5] of integer;

Sum, i : integer;

Begin

I:=1;

While i<5 DO begin sum:=sum+vect[i];

I:=i+1;

End;

End.

Ошибка: переменная i не «добегает» до верхней границы диапазона массива, т.е. Vect[5] не будет участвовать в суммировании. Нужно: $I \leq 5$.

В качестве примера выполнения задания приводится программа для вычисления суммы элементов одномерного массива (листинг 2.1).

Листинг 2.1 – Программа вычисления суммы элементов одномерного массива

```
program msg1;
const n=15; {число элементов массива}
var      a : array [1..n] of real;
          summa: real;

begin
  summa := 0;
  for i:=1 to n do
    begin
      readln (a[i]);
      summa := summa+a[i]
    end;
  writeln ('сумма ', n, ' элементов массива равна ', summa)
end.
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

2.1.3 Порядок проведения исследований



Алгоритм выполнения заданий ЛР следующий:

6. выбрать 5 задач по следующему правилу: номер по журналу- первая задача; номер каждой последующей задачи определяется прибавлением цифры 3 к номеру первой задачи, который только что вычислили (если достигнуто окончание списка вариантов задач, то перейти в его начало);
7. составить программы решения всех задач;
8. оформить отчет для всех задач в целом, включив в него задание, блок-схему алгоритма (в электронном виде), тексты программ и skrin-shert результатов выполнения каждой задачи и представить его на проверку.

2.1.4 Варианты заданий

22. Найти N элементов массива X, в котором $X_1 = X_2 = X_3 = 2$; а все последующие элементы вычисляются по формуле: $X_k = X_{k-2} - X_{k-3} + 1/K$.
Вычислить значения элементов массива Z по формуле :
 $Z = \cos X + \operatorname{tg} X$, где X меняется на отрезке [1;15] с шагом 0,92
3. Вычислить и напечатать значения функции $Y = A_k^2 + A_k - \sin A_k$ где элементы массива A вводятся с клавиатуры .
4. Найти сумму положительных значений элементов массива W, вводимого с клавиатуры.
5. Составить массив из положительных значений функции $Z = \cos X * \sin X$ для X, изменяющегося на отрезке [-5,10] с шагом 0,67.
6. Ввести с клавиатуры информацию о температуре воздуха за 2 недели. Записать в массив. Определить, сколько раз за это время она была ниже нуля.
7. Рост студентов представить в виде массива. Рост девушек закодировать со знаком "-", а рост юношей со знаком "+". Определить средний рост мальчиков.
8. Составить массив B из отрицательных значений функции $Z = \cos(x)/\sin(x-2)$ для x, изменяющегося на отрезке [5;-10] с шагом 0,67.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

9. Вычислить последовательность N чисел Фибоначчи и записать ее в мас-сив

$$F_0 = F_1 = 1; F_{i+1} = F_i + F_{i-1}$$



10. Вычислить N элементов массива X :

$$X_k = X_{k-1} + (1/2)X_{k-2}, \text{ где первые элементы уже известны } X_1 = 0, X_2 = 0,25.$$

11. Написать программу нахождения N элементов массивов X и Y , пользуясь формулами: $X_k = 3X_{k-1} + k$, $Y_k = X_{k-1} + Y_{k-1}$, $X_0 = 1$, $Y_0 = 2$.

12. Найти N элементов массива $X_1 = X_2 = X_3 = 1$; $X_k = X_{k-1} + X_{k-3} - 1/K$.

13. Найти сумму N элементов массива $X_1 = X_2 = X_3 = 2$; $X_k = X_{k-2} - X_{k-3} + 1/K$

14. Вычислить значения элементов массива Z по формуле :

$$Z = \cos X + \ln X, \text{ где } X \text{ меняется на отрезке } [1;15] \text{ с шагом } 0,92 \text{ и найти их сумму}$$

15. Вычислить сумму значений функции $Y_k = A_k^2 + A_k - \sin A_k$

где элементы массива A вводятся с клавиатуры .

16. Рассчитать сумму N значений элементов массива B , формуле :

17. Найти сумму отрицательных значений элементов массива W , вводимого с клавиатуры.

18. Найти сумму значений элементов массива W с четными индексами вводимого с клавиатуры.

19. Ввести с клавиатуры информацию о температуре воздуха за 2 недели. Определить, сколько раз за это время она была ниже нуля.

20. Найти сумму значений элементов массива A с нечетными индексами вводимого с клавиатуры.

21. Рассчитать сумму N значений элементов массива B , заполненного рандомно.

22. Составить массив B из отрицательных значений функции $Z = \cos(X)/\sin(X-2)$ для X , изменяющегося на отрезке $[5;-10]$ с шагом $0,67$ и найти его сумму.

15. Вычислить последовательность N чисел Фибоначчи:

$$F_0 = F_1 = 1; F_{i+1} = F_i + F_{i-1} \text{ и записать ее в массив. Найти сумму чисел с нечетными номерами.}$$



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

24. Написать программу нахождения элементов массивов X и Y ,



пользуясь формулами: $X_k = 3X_{k-1} + K$, $Y_k = X_{k-1} + Y_{k-1}$, $X_0 = Y_0 = 1$ и найти их сумму.

25. Найти N элементов массива $X_1 = X_2 = X_3 = 1$; $X_k = X_{k-1} + X_{k-3} - 1/K$ и найти их сумму.



Методические указания к лабораторной работе 2

Выявление закономерностей использования процедур и функций в решении прикладных задач

2.2.1 Цели занятия

Исследовать особенности составления методов пользователя. Изучить правила составления методов пользователя.

2.2.2 Теоретические основы и пример выполнения

Покажем основы составления пользовательских методов на примере языка С# [5].

Вначале рассмотрим процедуры и функции - методы класса.

Долгое время процедуры и функции играли не только функциональную, но и архитектурную роль. Весьма популярным при построении программных систем был метод функциональной декомпозиции "сверху вниз", и сегодня еще играющий важную роль. Но с появлением ООП архитектурная роль функциональных модулей отошла на второй план.

Для объектно-ориентированных языков, к которым относится и язык С#, в роли архитектурного модуля выступает класс. Программная система строится из модулей, роль которых играют классы, но каждый из этих модулей имеют содержательную начинку, задавая некоторую абстракцию данных.

Синтаксически в описании метода различают две части - описание заголовка и описание тела метода:

заголовок_метода

тело_метода

Рассмотрим синтаксис заголовка метода:

[атрибуты][модификаторы]{void| тип_результата_функции}

имя_метода([список_формальных_аргументов])



Имя метода и список формальных аргументов составляют сигнатуру метода. Заметьте, в сигнатуру не входят имена формальных аргументов - здесь важны типы аргументов. В сигнатуру не входит и тип возвращаемого результата.

Квадратные скобки (метасимволы синтаксической формулы) показывают, что атрибуты и модификаторы могут быть опущены при описании метода. У метода есть модификатор доступа и у него четыре возможных значения, из которых пока рассмотрим только два - `public` и `private`. Модификатор `public` показывает, что метод открыт и доступен для вызова клиентами и потомками класса. Модификатор `private` говорит, что метод предназначен для внутреннего использования в классе и доступен для вызова только в теле методов самого класса. Заметьте, если модификатор доступа опущен, то по умолчанию предполагается, что он имеет значение `private` и метод является закрытым для клиентов и потомков класса.

Обязательным при описании заголовка является указание типа результата, имени метода и круглых скобок, наличие которых необходимо и в том случае, если сам список формальных аргументов отсутствует. Формально тип результата метода указывается всегда, но значение `void` однозначно определяет, что метод реализуется процедурой. Тип результата, отличный от `void`, указывает на функцию. Вот несколько простейших примеров описания методов:

```
void A() {...};
```

```
int B() {...};
```

```
public void C() {...};
```

Методы А и В являются закрытыми, а метод С - открыт. Методы А и С реализованы процедурами, а метод В - функцией, возвращающей целое значение.

Список формальных аргументов метода может быть пустым, и это довольно типичная ситуация для методов класса. Список может содержать фиксированное число аргументов, разделяемых символом запятой.

Рассмотрим теперь синтаксис объявления формального аргумента:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

[ref|out|params]тип_аргумента имя_аргумента



Обязательным является указание типа и имени аргумента. Заметьте, никаких ограничений на тип аргумента не накладывается. Он может быть любым скалярным типом, массивом, классом, структурой, интерфейсом, перечислением, функциональным типом.

Несмотря на фиксированное число формальных аргументов, есть возможность при вызове метода передавать ему произвольное число фактических аргументов. Для реализации этой возможности в списке формальных аргументов необходимо задать ключевое слово `params`.

Оно задается один раз и указывается только для последнего аргумента списка, объявляемого как массив произвольного типа. При вызове метода этому формальному аргументу соответствует произвольное число фактических аргументов.

Содержательно, все аргументы метода разделяются на три группы: входные, выходные и обновляемые. Аргументы первой группы передают информацию методу, их значения в теле метода только читаются. Аргументы второй группы представляют собой результаты метода, они получают значения в ходе работы метода. Аргументы третьей группы выполняют обе функции. Их значения используются в ходе вычислений и обновляются в результате работы метода. Выходные аргументы всегда должны сопровождаться ключевым словом `out`, обновляемые - `ref`. Что же касается входных аргументов, то, как правило, они задаются без ключевого слова, хотя иногда их полезно объявлять с параметром `ref`.

Заметьте, если аргумент объявлен как выходной с ключевым словом `out`, то в теле метода обязательно должен присутствовать оператор присваивания, задающий значение этому аргументу. В противном случае возникает ошибка еще на этапе компиляции.

Процедуры и функции связываются теперь с классом, они обеспечивают функциональность данных класса и называются методами класса. Главную роль в программной системе играют данные, а функции лишь служат данным.



В C# процедуры и функции существуют только как методы некоторого класса, они не существуют вне класса.

2.2.3 Порядок проведения исследований

Алгоритм выполнения заданий ЛР следующий:

- В выбрать 5 задач по следующему правилу: номер по журналу- первая задача; номер каждой последующей задачи определяется прибавлением цифры 3 к номеру первой задачи, который только что вычислили (если достигнуто окончание списка вариантов задач, то перейти в его начало);
- В составить методы (функции) решения всех задач, поместить их в класс Form1;
- В оформить отчет для всего приложения в целом, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения каждой задачи и представить его на проверку.

2.2.4 Варианты заданий

Составить процедуры и функции, или методы возвращающие и не возвращающие значения для следующих заданий.

000 Найти среднее арифметическое положительных чисел, введенных с клавиатуры. Всего ввести N различных чисел.

8. Ввести с клавиатуры N чисел. Найти сумму тех из них, которые принадлежат интервалу (2;9).

17. Для N введенных с клавиатуры чисел найти сумму положительных кратных 3.

18. Для арифметической прогрессии 4, 9, 14, 19... найти первые n членов этой прогрессии.

19. Найти сумму отрицательных значений функции $Z = \sin(5-x)/\cos(x-2)$ для x, изменяющегося на отрезке [-5,12] с шагом 1,23.

22. Найти среднее арифметическое отрицательных чисел, введенных с кла-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

виатуры. Всего ввести N различных чисел.



к Найти среднее арифметическое чисел, принадлежащих отрезку $[2,184]$, кратных 2 и введенных с клавиатуры. Всего ввести N различных чисел.

14. Найти сумму значений функции, больших 2 $Z = \sin(1/x) + 5\cos(1/(x-3)) + x$

для x , изменяющегося на отрезке $[-3,8]$ с шагом 1,123.

6. Найти n членов последовательности $x_1 = x_2 = x_3 = 1$; $x_k = x_{k-1} + x_{k-3}$.

Вычислить последовательность N чисел $A_0 = x$, $A_1 = 2$, $A_k = A_{k-1} - A_{k-2}$.

Для $x_1 = 0,3$ и $x_2 = -0,3$ найти $x_k = k + \sin(x_{k-2})$ для k , изменяющегося следующим образом: $k = 3, 4, \dots, 14$.

Составить таблицу перевода дюймов в сантиметры для расстояний от 12. до 13 дюймов с шагом 1.

Вывести на печать значения функции, меньшие 2, $Z = \sin(1/x) + 5\cos(x-3) + x$ для x , изменяющегося на отрезке $[-7,4]$ с шагом 1.

Напечатать таблицу значений функции $Y = \operatorname{tg}(x/b) + x/(b-2)$ для x , изменяющегося от 0 до 10 с шагом 1 (b - произвольное число).

6. Вычислить N -й член последовательности $x_k = x_{k-2} - x_{k-1}$, $x_0 = 2,4$ $x_1 = 3,8$.

Составить таблицу перевода суток (от 1 до 7) в часы, минуты, секунды.

Вычислить N -й член последовательности $x_k = x_{k-1} + (2/3)x_{k-2} + 1$, $x_1 = -1$, $x_2 = 1,38$.

Напечатать значения функции $z = 1/(x-2) + 1/(x-5) + \ln(12,8-X)$ для x , изменяющегося на отрезке $[-4,14]$ с шагом 1,152.

Вывести на печать отрицательные значения функции $z = \sin(5-x)/\cos(x-9)$ для x , изменяющегося на отрезке $[-6,13]$ с шагом 1,541 (учесть область допустимых значений функции).

6. Из N введенных с клавиатуры чисел напечатать кратные 3 и меньшие 58.

8. Ввести с клавиатуры N чисел. Напечатать те из них, которые принадлежат интервалу $(1,11)$ и являются четными.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

7. Из N введенных с клавиатуры чисел напечатать положительные, кратные 3.



3. Вывести на печать значения функции $z = \sin(x/(x-2))$, находящиеся в интервале $(-0,4;0,8)$ для x , изменяющегося от 8 до -6 с шагом 1,235.

4. Ввести с клавиатуры N чисел. Напечатать те из них, которые принадлежат интервалу $(2;9)$.

И Для геометрической прогрессии 2, 6, 18, 54, 162 ... определить первые членов этой прогрессии.

Ввести с клавиатуры N чисел. Напечатать те из них, которые не принадлежат интервалу $(1;5)$.

- Найти n членов последовательности $x_1 = x_2 = x_3 = 1; x_k = x_{k-1} - 2x_{k-3}$.
- Вычислить последовательность N чисел $A_0 = x, A_1 = 2, A_k = A_{k-1} + A_{k-2}$
- Составить таблицу перевода килограммов (от 1 до 13) в граммы с шагом

· Найти сумму значений функции $Y = \cos(x/A) + x/(A-2)$ для x , изменяющегося от 2 до 13 с шагом 1 (A - произвольное число).



Методические указания к лабораторной работе 3

Исследование управляющих операторов C#

2.3.1 Цели занятия

Выработать умения и навыки по составлению программ с использованием базовых операторов языка C#.

2.3.2 Теоретические основы и пример выполнения

2.3.2.1 Консольный ввод и вывод. Преобразования типов

Для вывода данных различных типов используется метод Write() класса Console.

Можно выводить данные потоком:

```
int x=12547;  
Console.WriteLine("x="+x);
```

И будет напечатан текст x= и значение этой переменной:

```
x=12547
```

Возможен также форматный вывод, когда используются, так называемые, метки-заполнители (листинг 2.2)

Листинг 2.2- Форматный вывод данных

```
void frmt()  
{  
    int x=12547;  
    double y = 529465797.45963;  
    Console.WriteLine("{0:d} \n{1:f5}", x, y);  
}
```

Результат:

```
12547  
529465797.45963
```

Получение системного времени:

```
System.DateTime dat = Convert.ToDateTime("15.03.2003").
```

Ввод и вывод в консоли осуществляется всегда в типе string. Чтобы ввести число какого-то типа (int, double, float) нужно использовать метод Parse, пара-



метром которого должен быть метод `ReadLine()`, который прочитывает введенную строку, т.е. используется вот такая конструкция:

```
ТИП Переменная=ТИП.Parse(Console.ReadLine());
```

Пример ввода с клавиатуры:

```
int переменнаяA=int.Parse(Console.ReadLine());
```

```
double переменнаяX= double. Parse(Console.ReadLine());
```

Запись переменных и выражений в языке `C#` имеет определенную структуру (листинг 2.3).

Листинг 2.3- Инициализация констант и переменных

```
public void SimpleVars ()
{
//Объявления локальных переменных
int x, s; //без инициализации
int y =0, u = 77; //обычный способ инициализации
//допустимая инициализация
float w1=0f, w2 = 5.5f, w3 =w1+ w2 + 125.25f;
```

Листинг 2.3- Продолжение

```
//допустимая инициализация в объектном стиле
int z= new int();
//Недопустимая инициализация.
//Конструктор с параметрами не
определен //int v = new int(77);
x=u+y; //теперь x инициализирована
if(x> 5) s = 4;
for (x=1; x<5; x++)s=5;
//Инициализация в if и for не рассматривается,
//поэтому s считается неинициализированной переменной
//Ошибка компиляции:использование неинициализированной переменной
//Console.WriteLine("s= {0}",s);
} //SimpleVars
//Инициализация переменных. Глобальные переменные
//fields
```



```
int x,y; //координаты точки
string name; //имя точки
//конструктор с параметрами
public Testing(int x, int y, string name)
{
this.x = x; this.y = y; this.name = name;
}
//Локальные переменные
5) <summary>
6) Анализ области видимости переменных
7) </summary>
8) <param name="x"></param>
public void ScopeVar(int x)
{
//int x=0;
int y =77; string s = name;
if (s=="Точка1")
{
```

Листинг 2.3- Продолжение

```
int u = 5; int v = u+y; x +=1;
Console.WriteLine("y= {0}; u={1};
v={2}; x={3}", y,u,v,x);
}
else
{
int u= 7; int v= u+y;
Console.WriteLine("y= {0}; u={1}; v={2}", y,u,v);
}
//Console.WriteLine("y= {0}; u={1}; v={2}",y,u,v);
//Локальные переменные не могут быть статическими.
//static int Count = 1;
//Ошибка: использование sum до объявления
//Console.WriteLine("x= {0}; sum ={1}", x,sum);
int i;long sum =0;
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

```
for (i=1; i<x; i++)
```



```
{
//ошибка: коллизия имен: y
//float y = 7.7f;
sum +=i;
}
Console.WriteLine("x= {0}; sum ={1}", x, sum);
} //ScopeVar
//Константы
и <summary>
и Константы
и </summary>
public void Constants()
{
const int SmallSize = 38, LargeSize =58;
const int MidSize = (SmallSize +
LargeSize)/2; const double pi = 3.141593;
//LargeSize = 60; //Значение константы нельзя
изменить. Console.WriteLine("MidSize= {0}; pi={1}",
Листинг 2.3- Продолжение
MidSize, pi);
}
```

2.3.2.2 Операции отношения и условный оператор

Операции отношения можно просто перечислить - в объяснениях они не нуждаются. Всего операций 6 (==, !=, <, >, <=, >=):

· - «равно»;

!= - «неравно»;

· - «И»;

В - «ИЛИ»;

++x; или x++; --x; или x--; -инкремент, декремент.

- C#, как и в C++, разрешены условные выражения. Конечно, без них можно обойтись, заменив их условным оператором. Вот простой пример их ис-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

пользования, поясняющий синтаксис их записи (Листинг 2.4).



Листинг 2.4-Условное выражение

```
int a = 7, b= 9, max;  
max= (a>b) ? a:b;  
Console.WriteLine("a = " + a + "; b= " + b +  
"; max(a,b) = " + max);
```

Условное выражение начинается с условия, заключенного в круглые скобки, после которого следует знак вопроса и пара выражений, разделенных двоеточием " : ". Условием является выражение типа bool. Если оно истинно, то из пары выражений выбирается первое, в противном случае результатом является значение второго выражения. В данном примере переменная max получит значение 9.

Более длинная форма записи (листинг 2.5).

Листинг 2.5- Классическая форма записи условного оператора

```
if (условие) {прямая ветка}  
else {побочная ветка}
```

Листинг 2.5- Продолжение

```
if (x<=0 || y>=4 && d!=m || s==5) { Con-  
sole.WriteLine("g="+g);}  
else { Console.WriteLine("x="+x);}
```

При необходимости требуемый тип данных можно указать явно перед переменной или перед выражением:

```
//cast  
int p;  
p = (int)x;  
//b = (bool)x;
```

В классе Math и содержатся различные математические функции:

В тригонометрические функции - Sin, Cos, Tan;

В обратные тригонометрические функции - ASin, ACos, ATan, ATan2 (sinx, cosx);



- в гиперболические функции - Tanh, Sinh, Cosh;
- в экспоненту и логарифмические функции - Exp, Log, Log10;
- в модуль, корень, знак - Abs, Sqrt, Sign;
- в функции округления - Ceiling, Floor, Round;
- в минимум, максимум, степень, остаток - Min, Max, Pow, IEEEReminder.

В листинге 2.6 демонстрируется использование этих функций.

Листинг 2.6 – Математические конструкции языка C#

```
public void MathFunctions()  
{  
    double a, b, t, t0, dt, y;  
    string NameFunction;  
    Console.WriteLine("Введите имя F(t) исследуемой функции  
a*F(b*t)" + " (sin, cos, tan, cotan)"); NameFunction =  
    Console.ReadLine(); Console.WriteLine("Введите  
параметр a (double)");  
    a= double.Parse(Console.ReadLine());  
    Console.WriteLine("Введите параметр b (double)");
```

Листинг 2.6 – Продолжение

```
b= double.Parse(Console.ReadLine());  
    Console.WriteLine("Введите начальное время  
t0(double)"); t0= double.Parse(Console.ReadLine());  
    const int points = 10;  
    dt = 0.2;  
    for(int i = 1; i<=points; i++)  
    {  
        t = t0 + (i-1)* dt;  
        switch (NameFunction)  
        {  
            case ("sin"): y = a*Math.Sin(b*t); break;  
            case ("cos"): y = a*Math.Cos(b*t); break;  
            case ("tan"): y = a*Math.Tan(b*t); break;  
            case ("cotan"): y = a/Math.Tan(b*t); break;
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

```
case ("ln"): y = a*Math.Log(b*t); break;
```



```
case ("tanh"): y = a*Math.Tanh(b*t); break;
default: y=1; break;
} //switch

Console.WriteLine ("t = " + t + "; " + a + "*" +
NameFunction + "(" + b + "*t) = " + y + ";");
} //for

double u = 2.5, v = 1.5, p, w;
p = Math.Pow(u, v);
w = Math.IEEERemainder(u, v); Console.WriteLine
("u = " + u + "; v = " + v +
"; power(u, v) = " + p + "; remainder(u, v) = " + w);
} //MathFunctions
```

2.3.2.3 Операторы языка C#

Блок или составной оператор выглядит следующим образом:

```
{оператор_1 ...
оператор_N}
```

Это нужно, если под одним условием или в одном цикле, или по еще какой-либо причине должны выполняться сразу несколько операторов. Тогда их берут в фигурные скобки и управление сначала передается этому блоку операторов, а потом только выполняются все операторы, следующие за ним.

Оператор if

```
if(выражение_1) оператор_1
else if(выражение_2) оператор_2
...
else if(выражение_K) оператор_K
else оператор_N
if(выражение1) if(выражение2) if(выражение3) ...
```

Оператор switch

```
switch(выражение)
{
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

case константное_выражение_1: [операторы_1 оператор_перехода_1]



...

```
case константное_выражение_K: [операторы_K оператор_перехода_K]  
[default: операторы_N оператор_перехода_N]  
}
```

Ветвь default может отсутствовать.

В качестве примера выбора из множества альтернатив приведем программу, которая определяет период жизни в зависимости от возраста – age (листинг 2.7).

Листинг 2.7- Пример использования оператора switch

```
public void SetPeriod()  
{  
if ((age > 0)&& (age <7))period=1;  
else if ((age >= 7)&& (age <17))period=2;
```

Листинг 2.6- Продолжение

```
else if ((age >= 17)&& (age <22))period=3;  
else if ((age >= 22)&& (age <27))period=4;
```

Листинг 2.7- Продолжение

```
else if ((age >= 27)&& (age <37))period=5;  
else period =6;  
}  
public void SetStatus()  
{  
switch (period)  
{  
case 1: status = "child"; break;  
case 2: status = "schoolboy"; break;  
case 3: status = "student"; break;  
case 4: status = "junior researcher"; break;  
case 5: status = "senior researcher"; break;  
case 6: status = "professor"; break;  
default : status = "не определен"; break;  
}
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

```
Console.WriteLine("Имя = {0}, Возраст = {1}, Статус = {2}",
```



```
name, age, status);} }
```

Оператор goto имеет простой синтаксис и семантику:

```
goto [метка|case константное_выражение|default];
```

Далее рассмотрим операторы цикла. Первый из них- это цикл со счетчиком:

```
for(инициализаторы; условие; список_выражений) оператор
```

Для демонстрации этого цикла покажем программу, которая в передаваемом ей тексте определяет палиндромы. Палиндромом называется симметричная строка текста, читающаяся одинаково слева направо и справа налево (листинг 2.8).

Листинг 2.8- Определение палиндромов. Демонстрация цикла for

```
public bool Palindrom(string str)
{
for (int i =0,j =str.Length-1; i<j; i++,j--)
if(str[i]!=str[j]) return(false);
return(true);} }
```

Далее, рассмотрим цикл с предусловием:

```
while(выражение)
```

```
{тело цикла}
```

и цикл с постусловием:

```
do
```

```
{тело цикла}
```

```
while(выражение);
```

Продемонстрируем эти типы циклов в программе, которая работает следующим образом (листинг 2.9):

к Внешний цикл - образец многократно решаемой задачи.

к Завершение цикла определяется в диалоге с пользователем.

Листинг 2.9- Пример циклов с пред- и постусловием

```
public void Loop()
{
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

```
string answer, text;
```



```
do
{
Console.WriteLine("Введите слово");
text = Console.ReadLine(); int i =0,
j = text.Length-1;
while ((i<j) && (text[i] == text[j]))
{i++; j--;}
if (text[i] == text[j])
Console.WriteLine(text + " - это
палиндром!"); else
Console.WriteLine(text + " - это не палиндром!");
Console.WriteLine("Продолжим? (yes/no)"); answer
= Console.ReadLine(); }

while(answer == "yes");
}
```

О завершим рассмотрение циклов циклом `foreach`. Новым видом цикла, не унаследованным от C++, является цикл `foreach`, удобный при работе с массивами, коллекциями и другими подобными контейнерами данных.

Его синтаксис:

`foreach`(тип идентификатор `in` контейнер) оператор

В приведенном ниже примере показана работа с трехмерным массивом, заполненным случайными элементами. Массив создается с использованием циклов типа `for`, а при нахождении суммы его элементов, минимального и максимального значения используется цикл `foreach` (листинг 2.10).

Листинг 2.10- Использование цикла `foreach`

```
public void SumMinMax()
{
int [,,] arr3d = new int[10,10,10];
Random rnd = new Random(); for (int
i =0; i<10; i++)
for (int j =0; j<10; j++)
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

```
for (int k =0; k<10; k++)
```



```
arr3d[i,j,k]= rnd.Next(100);  
long sum =0; int min=arr3d[0,0,0], max=arr3d[0,0,0];  
foreach(int item in arr3d) { sum +=item;  
  
if (item > max) max = item;  
else if (item < min) min = item;  
}  
Console.WriteLine("sum = {0}, min = {1}, max = {2}",  
sum, min, max);  
}
```

2.3.3 Порядок проведения исследований

Алгоритм выполнения задач практического занятия следующий:

- в выбрать 5 задач по следующему правилу: номер по журналу- первая задача; номер каждой последующей задачи определяется прибавлением цифры 3 к номеру первой задачи, который только что вычислили (если достигнуто окончание списка вариантов задач, то перейти в его начало);
- в составить методы (функции) решения всех задач;
- в оформить отчет для всего приложения в целом, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения каждой задачи и представить его на проверку.

2.3.4 Варианты заданий

В Составить программу для перевода длины в метрах в длину в сантиметрах, определив функцию, выполняющую это преобразование и передав длину в метрах в качестве параметра.

4) Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив функцию, выполняющую это действие, и передавая массивы в качестве параметра.

5) Даны числа S, T. Получить с использованием функции пользователя $F(T, -2S, 1.17) + F(2.2, T, S - T)$ где $F(A, B, C) = (2A - B - \sin(C)) / (5 + C)$



к Составить программу перевода двоичной записи натурального числа в десятичную, описав соответствующую функцию с параметром. Перевод осуществлять для чисел, вводимых с клавиатуры. Признак конца ввода - число 0. 5.

Даны числа S, T. Получить с использованием функции пользователя с параметрами:

$$G(1, \sin(S)) + 2G(T * S, 24) - G(5, -S), \text{ где } G(A, B) = (2A + B * B) / (A * B * 2 + B * 5).$$

7. Составить программу для расчета значений гипотенузы треугольника, определив функцию, выполняющую этот расчет. Катеты передаются в качестве параметров.

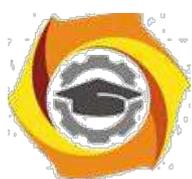
в Найти периметр десятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, которые передаются функции в качестве параметров из основной программы.

24. Найти периметр шестиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами. Координаты передаются функции в качестве параметров из основной программы.

8. Найти площадь пятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, и процедуру вычисления площади треугольника по трем сторонам. Описать функции с соответствующими формальными параметрами.

8. Составить программу вывода на экран всех натуральных чисел, не превосходящих N и делящихся на каждую из своих цифр. Описать соответствующую функцию, получающую из основной программы в качестве параметра натуральное число и возвращающую TRUE, если оно удовлетворяет указанному условию.

9. Используя подпрограмму - функцию, составить программу для нахождения максимального из трех чисел. Числа передаются функции в каче-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

стве параметров.



10. Используя подпрограмму - функцию, составить программу для печати знаков трех чисел, введенных с клавиатуры и передаваемых функции в качестве параметра.

У Используя подпрограмму - функцию, составить программу для возведения чисел в целую положительную степень. Число передается функции в качестве параметра из основной программы. Расчет вести для чисел, пока не будет введено число, равное 0.

6. Используя подпрограмму - функцию, составить программу для вычисления функции $Z=(X1+Y1)/(X1*Y1)$, где $X1$ - первый корень уравнения $X^2-4*X-1=0$; $Y1$ - первый корень уравнения $2*Y^2 + A*Y - A^2 = 0$ (A - произвольное).

' Задав функцию, вывести на печать средние арифметические двух массивов, введенных с клавиатуры. Массив передается функции в качестве параметра.

Задав функцию, рассчитать и вывести на печать максимальные значения в трех парах чисел, вводимых с клавиатуры. Пара чисел передается функции в качестве параметра.

Найти периметр восьмиугольника, координаты вершин которого заданы. Определить функцию вычисления расстояния между двумя точками, заданными своими координатами. Координаты передать функции в качестве параметров.

В Даны четыре пары чисел. Получить с использованием функции пользователя наибольший общий делитель для каждой пары.

и Даны числа A, B, C . Получить с использованием функции пользователя наименьшее значение. Числа передаются функции из основной программы в качестве параметров.

- Даны числа $x = 1, 2, \dots, N$. Получить с использованием функции пользователя значения $3*P(X+3)*P(X)$ для заданных x , где $P(X) = 10*X^3 - 14*X^2 + 12*X - 2$.



- Составить программу для расчета значений катета треугольника, определив функцию, выполняющую этот расчет. Гипотенуза и второй катет передаются в качестве параметров.

4. Даны целые числа a, b, c, d . Проверить с использованием функции пользователя их четность. Число для проверки передается в функцию в качестве параметра из основной программы.

8. Для каждого из 10 введенных с клавиатуры чисел напечатать сообщение: является ли оно простым или нет, описав функцию логического типа, возвращающую значение “ИСТИНА”, если число, переданное ей в качестве параметра, является простым.

- Даны числа S, T . Получить с использованием функции пользователя $Y(T, S) = G(12, S) + G(T, S) - G(2S - 1, S * T)$, где $G(A, B) = (2 * A + B * B) / (A * B * 2 + B * 5)$.

- Определите функцию, определяющую, какой целой степенью числа 2 является ее аргумент (если число не является степенью двойки - выдать соответствующее сообщение).

- Определите функцию, подсчитывающую сумму N первых элементов целочисленного массива A . N и массив A передать в качестве параметров.

- Вычислить количество простых чисел, не превосходящих заданного N . Описать функцию логического типа, возвращающую значение true, если число простое и false в противном случае.

С Используя подпрограмму - функцию с параметрами, составить программу для вычисления функции $F(X, Y) = (2X^3 - 4 * X^2 + X + 1) / (9 * Y^3 + Y + 4) + 3 * Y^2 + 5 * Y$.

С Составить программу для перевода веса в граммах в вес в килограммах, определив функцию, выполняющую это преобразование. Вес в граммах передается функции в качестве параметра.

в Даны числа S, T . Получить с использованием функции пользователя $G(12, S) + G(T, S) - G(2S - 1, S * T)$ где $G(A, B) = (2 * A + B * B) / (A * B * 2 + B * 5)$.



Методические указания к лабораторной работе 4

Исследование управляющих операторов Java

2.4.1 Цели занятия

Выработать умения и навыки составлять типовые программы решения задач на языке программирования Java с использованием основных управляющих операторов.

2.4.2 Теоретические основы и пример выполнения

Рассмотрим условный оператор.

if ... else - после служебного слова if должно располагаться логическое выражение. Логическое выражение должно быть взято в круглые скобки. В противном случае, либо выполняются операторы, стоящие за служебным словом else, либо, если нет служебного слова else, выполнение оператора if прекращается.

Оператор If дает возможность в зависимости от условия выполнять ту или иную ветвь программы. Синтаксис оператора следующий:

If условие выражение1 else выражение2;

Условие должно давать результат в виде логического значения истинности или ложности. Выражение1 будет выполняться если условие истинно. Выражение2 будет выполняться если условие ложно.

Существует сокращенный вариант оператора:

If условие выражение1

Пример. Определить, является ли введенное число днем недели, т.е. входит ли число в диапазон от 1 до 7 (листинг 2.11).

Листинг 2.11- Пример написания условий

```
package lr;
import java.util.Scanner;
public class l1 {
public static void main(String[] args) {
int a;
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

```
System.out.println("введите число");
```



```
Scanner scan=new Scanner(System.in);
a=scan.nextInt();
if ((a<1)|| (a>7)){
System.out.println("введенное число не является днем недели
");
}
else{
System.out.println("введенное число является днем недели
");
}
}}}
```

Выражение условия $(A < 1) \parallel (A > 7)$ будет давать TRUE, если выполняется $A < 1$ или $A > 7$ - в этом случае выполняется ветка `printf('Error ',A);`, иначе ветка `printf('OK ',A);`.

Далее перейдем к циклическому алгоритму (цикл с параметром - FOR).

Оператор `for` организует цикл:

`for(выражение 1; выражение 2; выражение 3)`

оператор;

выражение 3 (приращение) вычисляется после каждого прогона цикла;

выражение 1 (инициализация) вычисляется перед началом цикла;

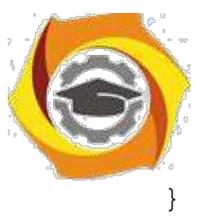
выражение 2 (условие) - до и после каждого прогона цикла.

Оператор тела цикла выполняется до тех пор, пока истинно выражение 2.

Любое из выражений, или же все три, могут отсутствовать, но при этом должны сохраняться все точки с запятыми. Если выражение 2 опущено, то считается, что оно всегда истинно (листинг 2.12).

Листинг 2.12- Фрагмент программы вычисления суммы чисел от 0 до 9

```
package lr;
public class Lr3 {
public static void main(String[] args)
{ int addCounter=0;
for(int counter=0; counter<10;counter++){
System.out.println(" counter="+counter );
addCounter+=counter;
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**



```
System.out.println(" Сумма чисел от 0 до 9="+addCounter );  
}
```

Далее, рассмотрим оператор цикла с предусловием.

Для организации цикла с предусловием используется оператор `while`:

`while(выражение)`

оператор;

Оператор тела цикла выполняется до тех пор, пока истинно выражение, записанное в скобках.

Выражение вычисляется до начала и после каждого прогона цикла. Цикл не выполняется ни разу, если выражение ложно (равно 0) (листинг 2.13).

Листинг 2.13- Фрагмент программы вычисления суммы чисел от 0 до 9

```
package lr;  
public class Lr4prim {  
    public static void main(String[] args)  
    { int counter=0, addCounter=0;  
      while(counter++<9) {  
          System.out.println(" counter="+counter );  
          addCounter+=counter;  
      }  
      System.out.println(" Сумма чисел от 0 до 9="+addCounter );  
    }  
}
```

Завершим рассмотрение циклов оператором цикла с постусловием.

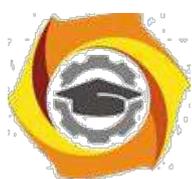
Цикл с пост-условием `do while` имеет вид:

`do`

оператор;

`while(условие);`

Оператор тела цикла выполняется до тех пор, пока истинно условие. Условие проверяется после выполнения оператора тела цикла (листинг 2.14).



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Листинг 2.14- Цикл с предусловием



```
package lr;
import java.math.BigInteger;
import java.util.Scanner;
public class Lr5prim {
    public static void main(String[] args)
        { int n;
        long x=1;
        System.out.println("Расчет факториала числа. Введите
число= " );
        Scanner scan=new Scanner(System.in);
        n=scan.nextInt();
        int Z=n;
    do {
        x*=n;
```

Листинг 2.14- Продолжение

```
System.out.println("число= " + n);

    } while (--n > 0);
        System.out.println("Факториал " + Z+"="+x);
    }
}
```

Оператор `continue` возвращает управление на начало цикла, пропуская стоящие после него операторы цикла. Оператором `break` можно завершить цикл.

Оператор безусловного перехода `goto` метка; передает управление на оператор, перед которым стоит метка. Метка представляет собой идентификатор с двоеточием и может стоять перед любым выполняемым оператором.

Как пример выполнения задания лабораторной работы покажем программу для решения одного из вариантов заданий, где нужно вывести на печать положительные значения функции $z = \sin(x) + 5\cos(x-2)$ для x изменяющегося на отрезке $[5, -10]$ с шагом 1,2 (листинг 2.15).

Листинг 2.15 – Пример выполнения варианта лабораторной работы

```
package lr;
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

```
public class Lr5laba {
```



```
public static void main(String[] args) {
    System.out.println("Вывести на печать положительные значения
функции  $z = \sin(x) + 5\cos(x-2)$  для  $x$  изменяющегося на отрезке  $[5; -10]$  с
шагом 1,2.");
    double x=5.0;
    double z;
    do
    {
        z=Math.sin(x)+5*Math.cos(x-2);
        if (z > 0){
            System.out.print("Положительное значение функции Z="+z+ "
при x=");          System.out.print(x+"\n");}
        x = x - 1.2;
    }while (x>=-10);}}
```

2.4.3 Порядок проведения исследований

Алгоритм выполнения заданий лабораторной работы следующий:

- С выбрать 5 задач по следующему правилу: номер по журналу- первая задача; номер каждой последующей задачи определяется прибавлением цифры 3 к номеру первой задачи, который только что вычислили (если достигнуто окончание списка вариантов задач, то перейти в его начало);
- С составить методы (функции) решения всех задач;
- С оформить отчет для всего приложения в целом, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения каждой задачи и представить его на проверку.

2.4.4 Варианты заданий

6. Вывести на печать положительные значения функции $y = \sin(x) + 5\cos(x-2)$ для x изменяющегося на отрезке $[-5, 12]$ с шагом 1,2.

Вывести на печать значения функции $z = \text{tg}(2x) - \sin(x)$ для x изменяющегося на отрезке $[-3, 3]$ с шагом 0,3.



25. Ввести с клавиатуры и напечатать модули N чисел; если введено отрицательное число, ввод и печать прекратить.

Вывести на печать значения функции $z = \ln(x) + \operatorname{tg}(2x)$, большие 1, для x изменяющегося на отрезке $[3, 8]$ с шагом 0,9.

В Определить, является ли натуральное число N степенью числа 5 или нет.

В Напечатать значения функции $y = \ln(x + 12/x)$, где значения x вводятся с клавиатуры. При вводе числа, не входящего в область определения функции, вычисления прекратить.

с Напечатать значения функции $y = \ln(x - 1/x)$, где значения x вводятся с клавиатуры. При вводе числа, не входящего в область определения функции, вычисления прекратить.

33. Для x из интервала $(-2; 8)$ с шагом 0,75 вычислить $y = (4x - 3x + \operatorname{tg}(x))/A$, где A вводится с клавиатуры.

Вывести на печать значения функции $z = \sin(x) + \cos(x)$, находящиеся в интервале $(-0,2; 0,8)$ для x изменяющегося на отрезке $[4, -6]$ с шагом 0,91.

В Дано натуральное число N . Получить наименьшее число вида 4^k , большее N .

3) Для x из интервала $(2; 8)$ с шагом 0,75 вычислить $y = (4x - 3x + \cos(x))/A$, где A вводится с клавиатуры.

в Найти первый член последовательности $\ln(9n)/(n \cdot n)$, меньший 1, для n изменяющегося следующим образом: $n = 1, 2, 3, \dots$

б) Определить, является ли натуральное число N степенью числа 3 или нет.

2. Вывести на печать отрицательные значения функции $z = \cos(x) - 5\sin(x - 2)$ для x изменяющегося на отрезке $[-3, 11]$ с шагом 0,9.

5. Ввести с клавиатуры и напечатать квадраты N чисел, если введено кратное 3 положительное число, ввод и печать прекратить.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

6. Вывести на печать отрицательные значения функции $z = \operatorname{tg}(x) + 5\cos(x-2)$ для x изменяющегося на отрезке $[12, 1]$ с шагом 1,2.



17. Ввести с клавиатуры и напечатать N чисел, если введено равное нулю или кратное 2 число, ввод и печать прекратить.

18. Вывести на печать значения функции $z = \ln(|x|) + \operatorname{tg}(2x)$, большие 2 для x изменяющегося на отрезке $[-3, -8]$ с шагом 0,9.

19. Найти первый отрицательный член последовательности $\sin(\operatorname{tg}(n/2))$ для n изменяющегося на следующим образом: $n=1,2,3\dots$

Напечатать значения функции $y = \ln(x+12/x)$, где значения x вводятся с клавиатуры. При вводе числа, не входящего в область определения функции, вычисления прекратить.

21. Найти первую цифру в целом положительном числе.

22. Дано натуральное число N . Получить наибольшее число вида 3^k , меньшее N .

3) Вывести на печать значения функции $z = \sin(x) + \cos(x)$, находящиеся в интервале $(-0,3; 0,7)$ для x изменяющегося на отрезке $[-4,6]$ с шагом 0,91.

4) Дано натуральное число N . Получить наименьшее число вида 5^k , большее N .

С Для x из интервала $(-2;8)$ с шагом 0,75 вычислить $y = (4x - 3x + \operatorname{tg}(x))/A$, где A вводится с клавиатуры.

3. Найти первый член последовательности $\ln(9n/(n*n+1))$, меньший 0, для n изменяющегося на следующим образом: $n=1,2,3\dots$

3. Определить, является ли натуральное N степенью числа 4 или нет.

4. Вывести на печать положительные значения функции $z = \sin(x) - 5\cos(x-2)$ для x изменяющегося на отрезке $[5,-12]$ с шагом 1,2.

10. Напечатать значения функции $Y = \sqrt{2x^2 - x^3}$ для произвольных x , вводимых с клавиатуры. При вводе числа, не входящего в область определения функции, ввод и печать прекратить.

3. Найти первый отрицательный член последовательности $\cos(\operatorname{ctg}(n))$ для n



Искусство деловых отношений

Методические указания по практическим занятиям УКАЗАНИЯ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ

Практическое занятие 1

Тема : История этических учений

Цель работы: Ознакомиться с ретроспективой взглядов философов различных времен на нравственно-этические проблемы. Сформулировать свое отношение к ним и выявить приоритеты. Закрепить знания по разделу «История этических учений», повысить свой интеллектуальный и культурный уровень.

Ход работы:

- 1)Ознакомьтесь с ретроспективой высказываний. Вдумайтесь в каждое из них.
- 2)Проинтерпретируйте каждое высказывание.
- 3)Сформулируйте свое отношение к данным точкам зрения.
- 4)Укажите высказывание, которое вызвало у вас наибольший интерес, с которым вы больше всего согласны. Мотивируйте свой ответ.
- 5)Укажите высказывание, которое не нашло отклика в вашей душе или вызвало противоречивые чувства, либо вы с ним не согласны. Мотивируйте свой ответ.
- 6)В конце работы обязательно сделайте вывод, этические воззрения какого периода кажутся вам наиболее приемлемыми или интересными.
- 7)Оформите письменный отчет.

Высказывания, предлагаемые для рассмотрения.

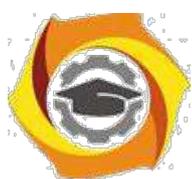
М.Монтень : « Мы берем на хранение чужие мысли и знания, только и всего. Нужно, однако, сделать их собственными. Мы уподобляемся человеку, который, нуждаясь в огне, отправился за ним к соседу, и , найдя у него прекрасный, яркий огонь, стал греться у чужого очага, забыв о своем намерении разжечь очаг у себя дома... И если можно быть учеными чужой ученостью, то мудрыми мы можем быть лишь собственной мудростью».

Аристотель: «Созерцательная деятельность разума...отличается значительностью, существует ради себя самой, не стремится ни к какой внешней цели и заключает в себе одной ей свойственное наслаждение, которое усиливается энергией; если она, сверх того, является самоудовлетворенною, заключающую в себе покой и лишенную тревожений, насколько это возможно человеку, то она и есть истинное блаженство для человека.»

Фома Аквинский : «Бог...есть первая причина, приводящая в движение как естественные причины, так и причины доброй воли. Подобно тому как, приводя в движение естественные причины, он не вызывает того, чтобы их акты утратили естественный характер. Точно так же, сообщая движения причинам доброй воли, он не ликвидирует добровольности вызываемых им действий, а скорее вызывает эту добровольность, ибо Бог в каждой вещи действует в соответствии с ее особенностью».

Апостол Павел: «Если я говорю языками человеческими и ангельскими, а любви не имею, то я – медь звенящая... Если имею дар пророчества, имею всякое познание и веру, но не имею любви – то я ничто. И даже если я раздам все свое имущество, а тело свое отдам на сожжение, но не имею любви – то я ничто. Любовь долготерпит, милосердствует, любовь не завидует, не превозносится, не гордится, не бесчинствует, не ищет своего и не раздражается, не мыслит зла и не радуется неправде, а сорадуется истине, все покрывает, всему верит, всего надеется, все переносит. Любовь никогда не престаёт, хотя и пророчества прекратятся, и языки умолкнут , и знание упразднится... Благодать Господа нашего Иисуса Христа с вами, и любовь моя со всеми вами во Христе Иисусе. Аминь.»

И.Кант: «Идеал служит первообразом для всестороннего определения подражаний ему, у нас нет иного руководства для наших поступков, кроме поведения этого божественного в нас, с



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

которым мы сравниваем себя, оцениваем себя таким образом и улучшаемся, никогда, однако, не имея возможности стать на один уровень с ним. Идеалы дают руководство к разуму, который



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

нуждается в понятии совершенного представителя своего рода, чтобы оценить и измерять соответственно этому понятию степень и недостатки несовершенного.»

Гегель: «Нравственность есть идея свободы... Нравственность есть иногда понятие свободы, ставшее наличным миром и природой самосознания». В.С.Соловьев : «Внутреннее основание 4.безусловное начало нравственной деятельности есть без сомнения любовь.»

Маркс: « ...мы утверждаем, что всякая теория морали являлась до сих пор продуктом данного экономического развития общества... Мораль истинно человеческая станет возможной лишь на такой ступени развития общества, когда не только будет уничтожена противоположность классов, но и изгладится след ее из практической жизни.»

Ф.Ницше: «Мораль – есть тормоз на пути развития человеческой личности... Мораль – удел плебеев и рабов и является порождением рабской зависимости...Существуют два типа морали: мораль господ и мораль рабов. Все действия и поступки человека складываются благодаря тому типу морали, к которому он принадлежит... Добро и зло – понятия относительные, это чисто человеческие мерки.»

Х.Штейнталь, М.Лацарус: «Человек научается не только своими современниками, но еще больше из опыта протекших столетий и тысячелетий и вполне зависит от них в своих мыслях, чувствах и воле».

Примечание. Выполнение этой работы не требует от вас фундаментальных знаний истории 5. философии, от вас требуется умение выделить главное из прочитанного. Поэтому следуйте следующему алгоритму выполнения задания:

- 1.Внимательно, вдумчиво прочитайте высказывание.
- 2.Определите основную мысль этого высказывания.
- 3.Сформулируйте ее своими словами.
- 4.Запишите собственную трактовку высказывания.
- 5.Выскажите свое мнение по поводу прочитанного.(Согласны вы с ним или нет и почему.)

Запишите его.

- 6.Сделайте вывод по работе :что он а вам дала, чему научила.

Проинтерпретировать – это значит сформулировать своими словами слова другого человека (проще сказать – пересказать.)

Ретроспектива – это демонстрация, выставка работ, выполненных художником в течение жизни. Ретроспектива высказываний – это набор и демонстрация высказываний философов различных времен.

Внимание! Помните, что в конце каждой работы должны обязательно быть выводы по работе, которые необходимо связать с целью работы!

Доклады на темы:

5. Искусство деловых отношений в Европе;
6. Американский стиль деловых отношений;
Японские традиции деловых отношений.

Практическое занятие 2

Тема: Основные нравственные ценности

Цель работы: Попытаться сформулировать свою систему нравственных ценностей и приоритеты в ней. Поразмышлять над нравственными проблемами современности.

Ход работы:

Составьте свой список нравственных ценностей, классифицируя их по критериям : «поп-рок-добродетель». Оцените свой нравственный уровень по пятибальной системе в соответствии с этой классификацией. Попросите оценить вас кого-либо из близких: друга, маму, сестру, брата и т. д. Изменилась ли картина? Сделайте выводы.

5. Сформулируйте свое отношение к нравственным проблемам современности путем градации их по степени важности с вашей точки зрения. (Список проблем смотри ниже).



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

5. Напишите сочинение – миниатюру на тему «Самая актуальная нравственная проблема современности».

Нравственные проблемы современности: -

проблема взаимоотношения поколений;
- проблема патриотического воспитания, отсутствия патриотизма; - проблема однополюсной любви; - насилие в обществе и семье; - духовная деградация общества; - терроризм; - алкоголизм и наркомания; - проституция и порнография; - кризис семьи и брака; - деньги – мировой эквивалент всего?

- проблема самореализации личности;
- проблема эвтаназии; - проблема абортот; - проблема лжи и лицемерия;
- агрессия СМИ, проблема интервенции сознания в самосознание;
- аномия; - безнравственность современного бизнеса; - проблема веры и безверия; - проблема ранних браков; - проблема одиночества; - проблема экологии;

- проблема разбазаривания духовного наследия;

Задание 1.

Комментарии к выполнению задания:

1) Внимательно прочитайте список нравственных проблем современности;

2) Определите степень важности каждой проблемы и перепишите их, нумеруя по порядку, в свой список. Например:

1) Проблема терроризма

2) Проблема духовной деградации общества

3) Проблема насилия и т.д.

4) Обязательно допишите несколько своих проблем, то, что важно для вас сегодня. Задание

3.

Сочинение следует писать на тему, которую вы обозначили № 1 в задании № 2. Текст сочинения должен быть небольшим (объем-1 лист от руки). Тип текста: рассуждение. Озаглавить сочинение следует так, как названа проблема в списке или более распространено. Например: «Проблема терроризма - основная нравственная проблема современности.»

Примечания

патриотизм – это любовь к Родине. деградация – это движение назад, утрата качеств, когда-то имеющихся. эвтаназия – это добровольный уход из жизни при помощи другого человека. (Просьба неизлечимого больного к врачу – сделать ему смертельную инъекцию, отключить аппарат, поддерживающий дыхание и т.д.) СМИ – средства массовой информации. аномия – отсутствие четких представлений о морали и нравственности, стирание граней между добром и злом.

Доклады на темы:

1 Критерии надежности и достоверности информации.

2 История формирования культуры деловых отношений в России.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие 3

Тема работы: Христианское учение о морали.

Цель работы: Ознакомиться с основными постулатами христианской морали.

Сформулировать свое отношение к ним. Повышать свой культурный уровень.

Ход работы: 1) Найдите в Библии 10 заповедей христианства. Запишите их на современном русском языке. Напишите, как вы понимаете 1 и 2 заповеди. (См. Библия, Ветхий Завет, книга Моисея «Исход» гл. 20).

Изучите послание апостола Павла к Коринфянам». Ответьте на вопросы : - Какие слова апостола Павла вызвали у вас сочувствие и понимание? - Проинтерпретируйте его слова о любви, б. взаимоотношениях мужчин и женщин. Согласны ли вы с этими словами? Аргументируйте свой ответ.

Напишите сочинение – миниатюру на тему «Что такое вера? Какую роль она играет в жизни человека»

Комментарии к выполнению работы

Цель работы – ознакомление. Поэтому задание №1 выполнить очень легко. Достаточно найти Библию, прочитать заповеди и записать их. Вдумайтесь, что означают 1 и 2 заповедь и напишите свое мнение об этом. Приведите примеры соблюдения и несоблюдения этих христианских заповедей. Выполняете ли вы их сами? Задание № 2 более объемно. Сначала внимательно прочитайте текст. Попытайтесь проникнуть в его содержание, хотя, конечно, содержание Евангелия очень сложно и многопланово и осознать его суть сразу очень сложно. Тем не менее когда-то надо начинать. Затем выпишите несколько цитат из текста ,которые нашли отклик в вашей душе, просто понравились вам. Затем выпишите несколько цитат (или одну большую цитату), иллюстрирующих взаимоотношения мужчин и женщин. Запишите свое мнение по этому вопросу и сравните свою точку зрения и точку зрения апостола Павла. *Текст «Послания святого апостола Павла к коринфянам» вы найдете в Библии, Новый Завет.(см .содержание.)

Задание № 3 похоже на задание семинарской работы №2. Требования к тексту те же .(См. практич.раб.№2).Написать сочинения вам помогут вопросы:

- Что такое вера (с вашей точки зрения?)
- Какую роль она играет в жизни человека? - Нужна ли вера человеку? - Легко ли жить, ни во что не веря?
- Что такое атеизм?
- Почему безверие – это страшно?
- Чем верующий человек отличается от неверующего? И т. д.

Это лишь примерные вопросы. Вы можете совершенно самостоятельно написать сочинение по другим вопросам или по своему плану. Можете видоизменить тему. Например: «Мировые религии. Сходства и различия.» Можете написать о той религии, которую вы исповедуете или интересуетесь.

Доклады на темы:

- 1 Общение как элемент деловых отношений.
- 2 Современные тенденции развития деловой этики.

Практическое занятие 4

Тема работы: Национальные особенности этики.

Цель работы: Ознакомление с национальными особенностями нравственности; понятием «национальной нравственности», сравнительный анализ подходов к некоторым этическим проблемам на основе фольклора.

Ход работы:



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Ознакомьтесь и составьте краткий конспект статьи «Народная нравственность». Выделите те цитаты(+), которые вам наиболее понравились, а те, что вызвали несогласие, обозначьте знаком (-).

Проведите сравнительный анализ предложенных вам пословиц и поговорок различных народов по следующему плану: а) внимательно прочтите их, вдумайтесь в их содержание; б) сгруппируйте эти пословицы по темам: о старших, о богатстве и т.д. в) сделайте вывод: отличаются ли взгляды разных народов на различные явления и категории нравственности.

Что такое народная нравственность? Понятно, что она идет из глубины истории, от корней рода человеческого. Моральные отношения зарождаются не сразу с образованием общества. Медленно и постепенно табу и обычаи трансформируются в механизмы моральной регуляции. Эти первоначальные формы нравственности и закладывают фундамент народной мудрости. К числу простейших нравственных представлений относятся: уважение к старшим, патриотизм, трудолюбие, коллективизм. Отношение к старшим – почтенное, уважительное – свойственно разным народам и разным временам. Это то, что объединяет все нравственные системы. Старость – это знания и мудрость, это жизненный опыт и связь поколений. Особое место в культуре старших у всех народов занимает мать. Даже у тех народов, у которых женщина – существо второго сорта (азиатские страны) – женщина – мать крайне уважаема и почитаема. (...) (...) С установлением патриархата роль женщины постепенно отходит на второй план. Она хранительница очага, мужчины же ее глава. Его обязанность – обеспечивать женщину. В арабской, китайской традиции считалось допустимым иметь столько жен, сколько ты можешь прокормить, однако считалось, что безнравственно жениться вовсе, если ты беден. Роль женщины азиатской системы морали двойка. С одной стороны – это беспрекословное подчинение мужчине и его воле. Мужья имели право даже убивать своих жен, без каких-либо последствий. До сих пор в некоторых азиатских странах процветает традиция заключать браки «между семьями», когда только что родившихся детей – мальчика и девочку – объявляют мужем и женой (причем девочку могли отдать замуж за долги, по обещанию и т.д.). По достижению совершеннолетия они женятся и никто не спрашивает их желания. Муж – это «хозяин». И в то же время женщина – мать – это полубожество. Проклятие матери страшнее любой казни. Слово матери беспрекословно. К тому же известны множество красивейших подтверждений любви мужчин – мусульман к женщине. Особенно ярко это прослеживается в фольклоре. Ни у одной народности в мире нет такого прекрасного эпоса, какой есть у азиатских народов: «Сказки Шахерезады», в которых воспеваются не только женская красота и находчивость, сколько женская мудрость. А известнейший мавзолей Тадж-Махал в Индии, построенный царем в честь своей любимой наложницы и вовсе является седьмым чудом света. Таким образом, думается, что азиатская традиция отношения к женщинам находится в «плёну» расхожих представлений и до конца не изучена. В христианской (европейской) морали отношение к женщине регламентируется так же двойка. Достаточно вспомнить слова Апостола Павла: «Жены, повинуйтесь своим мужьям, яко Господу. Мужья, любите своих жен. Ибо как Господь возлюбил церковь свою, так и вы возлюбите своих жен, да будут двое – одна плоть.» Слова, подтверждающие единство мужского и женского начала и их взаимопроникновение и гармонию. Это же отношение мы встречаем и в европейском фольклоре: «Каков муж – такова и жена.» «Жена мужем красна». Повинуйтесь, но лишь тому, кто тебя любит, как самого себя. Интересен взгляд индийской философии на взаимоотношения мужчины и женщины. В трактовке индийских философов женское начало ИНЬ и мужское начало ЯНЬ должны находиться в безусловной гармонии. Знаменитый круг, в котором как бы сливаются воедино две капельки воды – таковы должны быть отношения мужчины и женщины. Семейная мораль различных народов так же несколько отличается. В европейской традиции морали семья создается для продолжения рода, для того, чтобы выполнить предназначение человека на земле – продолжить род человеческий. В азиатской традиции семьи с большим количеством жен создается и для любовного удовлетворения и для того, чтобы иметь детей и для престижа. Чем больше жен – тем ты богаче. Таким образом, разные народы по-разному воспринимают моральные установки, относящиеся к вопросу взаимоотношения полов. Тем не менее, общего в них тоже очень много. Понятие «традиции» в конечном счете складывается из понятия народная нравственность. Все праздники, которые мы и сегодня празднуем, имеют в истории какой-либо



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

нравственный подтекст. Патриотизм. Чувство Родины, исконная связь с конкретной землей составляют аксиому нравственного сознания. У всех народов и во все времена культ Родины и ее защиты был одним из основных культов. Нет народа, который бы не дорожил землей своих предков, домом, где он вырос, не стремился бы приукрасить ее, защитить. Безусловное единение человека с той природой, местностью, где он родился отражается в народном эпосе и фольклоре. Трудолюбие и богатство. Идея неразрывного единства человека и земли является источником нравственной идеи трудолюбия. Отношение к труду часто выступает критерием нравственности. Ни у одного народа не в почете ленивые, фольклор всех народов высмеивает лень и паразитизм. «Лень – мать всех пороков» (лат.) О богатстве же у разных народов разное представление. Зачастую в фольклоре встречается негативное отношение к богатству – богатые спесивы, высокомерны и суетны. Дело тут не в осуждении богатства как такового, а в 25 порицании богатства незаслуженного. Народная мудрость одобряет не роскошь, а трудовую зажиточность, достаток. Коллективизм. Первое звено коллектива – семья. Семья и ее культ – тоже общее для многих народов. В европейской философской традиции семья воспринимается как оплот человека, неверность супругов карается и духовными и общественными законами. Азиатская мораль относится к вопросу семьи так же. Далее коллективизм проявляется в отношении к обществу. Общество есть сфера самовыражения личности, и личности составляют общество. Но общество может и раздавить. Народная мудрость противоречива как и сама жизнь. Далеко не все заповеди, которые провозглашаются выполняются. Специфика национальных образцов нравственности Национальные особенности нравственных систем заключаются в их форме. Иногда национальное является синонимом нравственного. Национальные особенности проявляются в психологии народа. В последнее время широко внедряется в обиход понятие «менталитет», «ментальность», под которым понимаются особенности сознания народа, определяющие его поведение. Очень велико влияние на формирование нравственности религиозных факторов, который зачастую формирует народную, этническую нравственность. Национальная форма нравственности усваивается нами с рождения. И это не случайно. Мораль везде занимается одним и тем же: гуманизацией, возвышением личности, созданием идеальной перспективы развития. Именно поэтому многие нравственные принципы многих народов совпадают. Деление морали по национальному признаку противоречат сущности морали. В любом стаде, как утверждает пословица, есть «паршивая овца». Моральные критерии могут варьироваться, но в целом они опираются на одни и те же принципы. Не убий, не укради, почитай отца и мать твою, не лги и т.д. В XX веке различия в народной нравственности стираются все больше, в связи с процессами глобализации и интеграции. Но именно благодаря интеграции XX век осознал ценность многообразия этносов.

Пословицы для сравнительного анализа.

Слова старших и их наставления не высказываются зря. (туркмен.) Отцу достается сын для славы.(тюркск.) Ремесло отца сыну подобает.(туркмен.) Что стар – что млад.(руссск.) Старый конь борозды не испортит, но и глубоко не вспашет.(руссск.)

Дым отечества ярче огня чужбины (лат). Мой дом – моя крепость(англ). Чем быть шахом в чужой стране, лучше нищим в своей(тюрк.) Разлученный с любимой один раз плачет. Разлученный с Родиной – всю жизнь (туркмен.)

Без труда не вытащишь рыбку из пруда (русск.) Без труда нет наслажденья (тюркск.) Лодырю и порог кажется вершиной горы (тюркск). Работать кое-как, так и умереть кое как.(Дагест).

Сытый голодного не разумеет. (Русск) Доброе имя значит больше, чем богатство.(Лат). Богатство слаще отца и матери, но жизнь дороже золота, полного золота.(тюркск) Не с деньгами жить, а с добрыми людьми.(руссск)

С волками жить – по- волчьи выть. (Русск .) Попал в стаю – лай не лай, а хвостом виляй (руссск). С кем поведешься – от того и наберешься.(Русск.) Не плюй на общество – оно утрется, а плюнет – утонешь (англ).



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Практическое занятие 5

Тема: Этносоциология

4. Этнос - субъекты исторического развития.

5. Социальные проблемы национальных отношений.

ЭТНОС (с греч. - племя, народ) - исторически сложившаяся на определенной территории устойчивая совокупность людей, обладающих общими чертами и стабильными особенностями культуры (включая язык) и психологического склада, а также сознанием своего единства и отличия от других подобных образований (самосознание). Этносоциология - область социологии, изучающая генезис, сущность, функции, общие закономерности развития этносов, межэтнические (межнациональные) отношения и разрабатывающая методологические принципы из исследования.

Наряду с категорией "ЭТНОС", мы часто встречаем категорию "НАЦИЯ". Нация (от лат. - народ) - тип этноса, характерный для развитого, классового (капиталистического) общества.

Среди социальных отношений видное место принадлежит национальным и межнациональным отношениям.

Почему важно изучение этносоциологии в современных условиях вообще, в национальных регионах в особенности?

В чем причина национальных конфликтов в мире и в Российской Федерации?

Какие проблемы, стоящие перед чувашской культурой и языком Вы можете назвать?

Темы рефератов и докладов:

Этносоциология как отрасль социологической науки. Возможно ли окончательное решение национального вопроса?

Исторические и географические условия как факторы формирования этнических групп.

Этнические процессы в Российской Федерации.

Межнациональные конфликты: причины и следствия.

Взаимосвязь социально-экономических, политических и национальных отношений.

Динамика социальных изменений внутри этноса.

Проблемы самоопределения народов.

Национальные движения за возрождение и развитие национальной культуры.

Основные направления гармонизации национальных отношений.

Практическое занятие 6

Тема работы: Этика современного бизнеса.

Цель работы: ознакомление с основными нормами этики делового человека.

Ход работы:

4. Дайте определение этикета, речевого этикета и ситуативного этикета (используйте учебную литературу). 27

5. Проанализируйте предложенные вам ситуации и сформулируйте свое решение каждой ситуации, учитывая этический аспект проблемы.

Ситуации для анализа с точки зрения этического аспекта:

Ситуация № 1. Вы работаете в табачной компании, точно зная, что курение вызывает рак. Как вы поступите, если вам будет предложена другая, но менее оплачиваемая работа?

Ситуация № 2. Вы – сотрудник фирмы по производству косметики. Для того, чтобы увеличить объем продаж менеджер вашей компании предложит написать на тубике, что это новый вид товара, который обладает чудодейственным эффектом омоложения, благодаря новейшим технологиям, хотя товар остался тем же самым. Как вы отнесетесь к его предложению?

Ситуация № 3. Еще вчера вы с удовольствием обсуждали с сотрудниками своего начальника. И вдруг он неожиданно уволился и вас назначили на его место. Как вы поведете себя, зная, что думают про начальство ваши подчиненные?

Ситуация № 4. Ваш конкурент по бизнесу на время выбыл из строя из-за тяжелых семейных обстоятельств. У вас есть прекрасный шанс «съесть его». Как вы поступите?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Ситуация № 5. Вы - начальник фирмы, в которой торговые агенты (так сложилось) только мужчины. Вдруг к вам на собеседование приходит женщина, квалификация которой явно выше ваших агентов. Возьмете ли вы ее на работу?

Ситуация № 6. Ваш партнер по бизнесу предлагает не заплатить по контракту поставщикам продукции, так как прибыль, полученная в результате реализации этой продукции, оказалась гораздо ниже предполагаемой. Как вы отреагируете?

Ситуация № 7. В вашей фирме работает несколько сотрудников преклонного возраста, которые уже не справляются с объемом работы. Но до пенсии им еще не хватает года-двух. Как вы поступите?

Ситуация № 8. Вы – сотрудник аптекарской фирмы. Вы закупили крупную партию медикаментов. Вдруг до вас дошла информация, что часть из них поддельная, хотя при закупке вам были предоставлены соответствующие документы. Найти тех, кто продал вам фальшивку, уже не удалось. Как вы поступите: умолчите и реализуете продукцию, не задумываясь о последствиях, или уберете их из продажи, понеся немалые убытки?

Ситуация № 9 28 Вы – владелец ателье по пошиву кожаных изделий. Вы только что выполнили заказ на эксклюзивную модель кожаного пиджака. Но через неделю заказчик пришел и продемонстрировал, что кожа «осыпалась» и пиджак пришел в негодность. Как вы поступите?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Заочная форма обучения

Практическое занятие 1

Тема : История этических учений

Цель работы: Ознакомиться с ретроспективой взглядов философов различных времен на нравственно-этические проблемы. Сформулировать свое отношение к ним и выявить приоритеты. Закрепить знания по разделу «История этических учений», повысить свой интеллектуальный и культурный уровень.

Ход работы:

- 1) Ознакомьтесь с ретроспективой высказываний. Вдумайтесь в каждое из них.
- 2) Проинтерпретируйте каждое высказывание.
- 3) Сформулируйте свое отношение к данным точкам зрения.
- 4) Укажите высказывание, которое вызвало у вас наибольший интерес, с которым вы больше всего согласны. Мотивируйте свой ответ.
- 5) Укажите высказывание, которое не нашло отклика в вашей душе или вызвало противоречивые чувства, либо вы с ним не согласны. Мотивируйте свой ответ.
- 6) В конце работы обязательно сделайте вывод, этические воззрения какого периода кажутся вам наиболее приемлемыми или интересными.
- 7) Оформите письменный отчет.

Высказывания, предлагаемые для рассмотрения.

М.Монтень : « Мы берем на хранение чужие мысли и знания, только и всего. Нужно, однако, сделать их собственными. Мы уподобляемся человеку, который, нуждаясь в огне, отправился за ним к соседу, и , найдя у него прекрасный, яркий огонь, стал греться у чужого очага, забыв о своем намерении разжечь очаг у себя дома... И если можно быть учеными чужой ученостью, то мудрыми мы можем быть лишь собственной мудростью».

Аристотель: «Созерцательная деятельность разума...отличается значительностью, существует ради себя самой, не стремится ни к какой внешней цели и заключает в себе одной ей свойственное наслаждение, которое усиливается энергией; если она, сверх того, является самоудовлетворенною, заключающую в себе покой и лишённую тревожений, насколько это возможно человеку, то она и есть истинное блаженство для человека.»

Фома Аквинский : «Бог...есть первая причина, приводящая в движение как естественные причины, так и причины доброй воли. Подобно тому как, приводя в движение естественные причины, он не вызывает того, чтобы их акты утратили естественный характер. Точно так же, сообщая движения причинам доброй воли, он не ликвидирует добровольности вызываемых им действий, а скорее вызывает эту добровольность, ибо Бог в каждой вещи действует в соответствии с ее особенностью».

Апостол Павел: «Если я говорю языками человеческими и ангельскими, а любви не имею, то я – медь звенящая... Если имею дар пророчества, имею всякое познание и веру, но не имею любви – то я ничто. И даже если я раздам все свое имущество, а тело свое отдам на сожжение, но не имею любви – то я ничто. Любовь долготерпит, милосердствует, любовь не завидует, не превозносится, не гордится, не бесчинствует, не ищет своего и не раздражается, не мыслит зла и не радуется неправде, а сорадуется истине, все покрывает, всему верит, всего надеется, все переносит. Любовь никогда не престаёт, хотя и пророчества прекратятся, и языки умолкнут , и знание упразднится... Благодать Господа нашего Иисуса Христа с вами, и любовь моя со всеми вами во Христе Иисусе. Аминь.»

И.Кант: «Идеал служит первообразом для всестороннего определения подражаний ему, у нас нет иного руководства для наших поступков, кроме поведения этого божественного в нас, с которым мы сравниваем себя, оцениваем себя таким образом и улущаемся, никогда, однако, не



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

имея возможности стать на один уровень с ним. Идеалы дают руководство к разуму, который нуждается в понятии совершенного представителя своего рода, чтобы оценить и измерять соответственно этому понятию степень и недостатки несовершенного.»

Гегель: «Нравственность есть идея свободы... Нравственность есть иногда понятие свободы, ставшее наличным миром и природой самосознания». В.С.Соловьев : «Внутреннее основание 4.безусловное начало нравственной деятельности есть без сомнения любовь.»

Маркс: «...мы утверждаем, что всякая теория морали являлась до сих пор продуктом данного экономического развития общества... Мораль истинно человеческая станет возможной лишь на такой ступени развития общества, когда не только будет уничтожена противоположность классов, но и изгладится след ее из практической жизни.»

Ф.Ницше: «Мораль – есть тормоз на пути развития человеческой личности... Мораль – удел плебеев и рабов и является порождением рабской зависимости... Существуют два типа морали: мораль господ и мораль рабов. Все действия и поступки человека складываются благодаря тому типу морали, к которому он принадлежит... Добро и зло – понятия относительные, это чисто человеческие мерки.»

Х.Штейнталь, М.Лацарус: «Человек научается не только своими современниками, но еще больше из опыта протекших столетий и тысячелетий и вполне зависит от них в своих мыслях, чувствах и воле».

Примечание. Выполнение этой работы не требует от вас фундаментальных знаний истории 5. философии, от вас требуется умение выделить главное из прочитанного. Поэтому следуйте следующему алгоритму выполнения задания:

- 1.Внимательно, вдумчиво прочитайте высказывание.
- 2.Определите основную мысль этого высказывания.
- 3.Сформулируйте ее своими словами.
- 4.Запишите собственную трактовку высказывания.
- 5.Выскажите свое мнение по поводу прочитанного.(Согласны вы с ним или нет и почему.)

Запишите его.

- 6.Сделайте вывод по работе :что он а вам дала, чему научила.

Проинтерпретировать – это значит сформулировать своими словами слова другого человека (проще сказать – пересказать.)

Ретроспектива – это демонстрация, выставка работ, выполненных художником в течение жизни. Ретроспектива высказываний – это набор и демонстрация высказываний философов различных времен.

Внимание! Помните, что в конце каждой работы должны обязательно быть выводы по работе, которые необходимо связать с целью работы!

Практическое занятие 2

Тема: Основные нравственные ценности

Цель работы: Попытаться сформулировать свою систему нравственных ценностей и приоритеты в ней. Поразмышлять над нравственными проблемами современности.

Ход работы:

Составьте свой список нравственных ценностей, классифицируя их по критериям : «по-рок-добродетель». Оцените свой нравственный уровень по пятибалльной системе в соответствии с этой классификацией. Попросите оценить вас кого-либо из близких: друга, маму, сестру, брата и т. д. Изменилась ли картина? Сделайте выводы.

4. Сформулируйте свое отношение к нравственным проблемам современности путем градации их по степени важности с вашей точки зрения. (Список проблем смотри ниже).

5. Напишите сочинение – миниатюру на тему «Самая актуальная нравственная проблема современности».

Нравственные проблемы современности: -
проблема взаимоотношения поколений;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

4. проблема патриотического воспитания, отсутствия патриотизма;
5. проблема однополой любви; - насилие в обществе и семье;
6. духовная деградация общества; - терроризм;
7. алкоголизм и наркомания; - проституция и порнография;
8. кризис семьи и брака;
9. деньги – мировой эквивалент всего?
10. проблема самореализации личности;
11. проблема эвтаназии; - проблема абортот;
12. проблема лжи и лицемерия;
13. агрессия СМИ, проблема интервенции сознания в самосознание;
14. аномия; - безнравственность современного бизнеса;
15. проблема веры и безверия;
16. проблема ранних браков;
17. проблема одиночества;
проблема экологии;
18. проблема разбазаривания духовного наследия;

Задание 1.

Комментарии к выполнению задания:

- 1) Внимательно прочитайте список нравственных проблем современности;
- 2) Определите степень важности каждой проблемы и перепишите их, нумеруя по порядку, в свой список.

Например:

- 1) Проблема терроризма
- 2) Проблема духовной деградации общества
- 3) Проблема насилия и т.д.
- 4) Обязательно допишите несколько своих проблем, то, что важно для вас сегодня.

Задание 2.

Сочинение следует писать на тему, которую вы обозначили № 1 в задании № 2. Текст сочинения должен быть небольшим (объем-1 лист от руки). Тип текста: рассуждение. Озаглавить сочинение следует так, как названа проблема в списке или более распространено. Например: «Проблема терроризма - основная нравственная проблема современности.»

Примечания

патриотизм – это любовь к Родине. деградация – это движение назад, утрата качеств, когда-то имеющихся. эвтаназия – это добровольный уход из жизни при помощи другого человека. (Просьба неизлечимого больного к врачу – сделать ему смертельную инъекцию, отключить аппарат, поддерживающий дыхание и т.д.) СМИ – средства массовой информации. аномия – отсутствие четких представлений о морали и нравственности, стирание граней между добром и злом.

История

Методические указания по практическим занятиям

I. Введение

1. Цели изучения дисциплины

Целью изучения дисциплины «История» является формирование знаний об отечественной истории, о специфике истории как комплексной дисциплины гуманитарного цикла и формы общественной деятельности

2. Планируемые результаты обучения



Изучение дисциплины направлено на формирование у выпускника способности решать профессиональные задачи **проектного** типа деятельности.

II. Указания к практическим занятиям

Методические указания к практическим занятиям очной формы обучения

Модуль I.

Практическое занятие 1. Методологические вопросы истории

Целью проводимого занятия является формирование знаний: специфика определения предмета истории; основные историософские идеи историков; базовые методы исторического познания; выяснение всех историографических и источниковедческих вопросов; понимание хода мировой истории и места в нем России.

Форма проведения: контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

Вопросы к контрольной работе:

- е Предмет и значение исторического познания.
- е Историософия русской истории
- е Источниковедение и историография отечественной истории
- е Методология исторического познания.
- е Формационный и цивилизационный подходы в исторической науке.

В Дискуссии западников и славянофилов о месте России в мире и тенденциях ее исторического развития.

Дополнительные вопросы для обсуждения и самопроверки:

- с Поясните, в чем состоит специфика предмета исторической науки?
- с Каково значение исторического познания?
- с Изложите основные историософские теории русской истории
- с Поясните особенность и методы источниковедения?
- с Какие есть периоды развития историографии отечественной истории?
- с В чем состоит проблема объективности в исторической науке?
- с Проведите сравнительную характеристику формационного и цивилизационного подходов в исторической науке?

Литература:

- Отечественная история: учебник для бакалавров / И.Н. Кузнецов. – 9-е изд., испр. и доп. – М.: издательско-торговая корпорация «Дашков и К», 2018. – 816 с.
<https://znanium.com/catalog/product/414990>
- История России: Учебник / Ш.М. Мунчаев, В.М. Устинов. - 7-е изд., перераб. и доп. - Москва : Норма: НИЦ ИНФРА-М, 2018. - 608 с. <https://znanium.com/catalog/product/966207>
- История: учеб. пособие / Ю.А. Шестаков. — М. : РИОР : ИНФРА-М, 2019. — 248 с.
<https://znanium.com/catalog/product/1014722>
- История России XVIII — начала XX века : учебник / М.Ю. Лачаева, Л.М. Ляшенко, В.Е. Воронин, А.П. Синелобов ; под ред. М.Ю. Лачаевой. — Москва : ИНФРА-М, 2019. — 648с.
<https://znanium.com/catalog/product/1023725>



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

10. История: от древних цивилизаций до конца XX в.: Учебник / Оришев А.Б., Тарасенко В.Н. - Москва: ИЦ РИОР, НИЦ ИНФРА-М, 2019. - 276 с.
<https://znanium.com/catalog/product/1048076>

Практическое занятие 2. Славяне. Киевский и Московский периоды русской истории.

Целью проводимого занятия является формирование знаний: изучение начального и средневекового периодов русской истории; уяснение генезиса русской государственности, экономической системы, идеологии; хроники правления первых князей; основные категории развития древней и средневековой русской истории; понимание особенности развития России в 17 в.; внутренняя и внешняя политика первых Романовых; культура Киевской и Московской Руси.

Форма проведения: контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

Вопросы к контрольной работе:

14. Проблемы этногенеза славян.
15. Происхождение Российского государства.
16. Хроника первых веков
17. Феодалная раздробленность и иностранная интервенция на Русь в XII-XV вв.
18. Монголо-татары и Русь: проблемы взаимовлияния. Было ли иго? Современные оценки.
19. Политика Московских князей.
20. Личность и деятельность Ивана III.
21. Воссоединение земель вокруг Москвы в XIV- первой половине XV вв.
22. Московское царство в XVI в. Иван IV.
23. Причины и ход Смутного времени.
24. Россия в XVII веке.
25. Культура Древней и Московской Руси

Дополнительные вопросы для обсуждения и самопроверки:

- в В чем состоят основные проблемы в выявлении этногенеза славян?
- в Какова природа государства и признаки образования государства Киевская Русь?
- в Изложите специфику дискуссии вокруг норманской теории?
- в Расскажите хронику истории первых князей?
- в Обоснуйте идею высокоразвитости культуры Древней Руси?
- в Какова была внешняя политика Руси в XI-XIII вв.?
- в Изложите особенность борьбы северо-восточной Руси с крестоносцами?
- в Изложите основные факторы феодалной раздробленности на территории Руси в XII - XIII вв.?
- в Укажите на специфику экономического развития древней Руси?.
- в Какие существовали проблемы во взаимоотношениях Орды и Руси?
- в Опишите основные направления политики Московских князей?
- в Каковы были предпосылки и альтернативы воссоединения русских земель?
- в Расскажите хронику Куликовской битвы?
- в В чем состояла внешняя политика Ивана III и Василия III?
- в Расскажите об особенностях личности царя Ивана IV.
- в В чем суть местничества?
- в Изложите основные реформы внутренней политики Ивана IV?
- в Каковы причины и характер Смутного времени в России?
- в Назовите основные социально-экономические и политические особенности развития России после Смутного времени XVII в.?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

В В чем особенность Соборного уложения 1649 г.?

В Каково место Крестьянской войны под предводительством С. Разина в истории России XVII века?

В В чем сущность Церковного раскола?

В Опишите специфику культуры России, традиции и новации, отношение государства и церкви в России в XVI-XVII вв.?

Литература:

с Отечественная история: учебник для бакалавров / И.Н. Кузнецов. – 9-е изд., испр. и доп. – М.: издательско-торговая корпорация «Дашков и К», 2018. – 816 с.

<https://znanium.com/catalog/product/414990>

с История России: Учебник / Ш.М. Мунчаев, В.М. Устинов. - 7-е изд., перераб. и доп. - Москва : Норма: НИЦ ИНФРА-М, 2018. - 608 с. <https://znanium.com/catalog/product/966207>

с История: учеб. пособие / Ю.А. Шестаков. — М. : РИОР : ИНФРА-М, 2019. — 248 с.

<https://znanium.com/catalog/product/1014722>

с История России XVIII — начала XX века : учебник / М.Ю. Лачаева, Л.М. Ляшенко, В.Е. Воронин, А.П. Синелобов ; под ред. М.Ю. Лачаевой. — Москва : ИНФРА-М, 2019. – 648с.

<https://znanium.com/catalog/product/1023725>

с История: от древних цивилизаций до конца XX в.: Учебник / Оришев А.Б., Тарасенко В.Н. - Москва:ИЦ РИОР, НИЦ ИНФРА-М, 2019. - 276 с.

<https://znanium.com/catalog/product/1048076>

Практическое занятие 3. Россия в XVIII веке

Целью проводимого занятия является формирование знаний: внешняя политика Петра I; внутренняя политика в первой четверти XVIII в.; период дворцовых переворотов; значение петровских культурных преобразований для последующей отечественной истории; развитие России во 2-й половине XVIII в.; влияние идей Просвещения на русскую культуру.

Форма проведения: контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

Вопросы к контрольной работе:

и Личность Петра I.

и Петр I: борьба за преобразование традиционного общества в России.

и Северная война.

и Культурная реформа.

и Построение регулируемого государства.

и Послепетровская эпоха XVIII в.

и Основные направления «европеизации» страны.

и Эпоха дворцовых переворотов.

и Реформы Екатерины II.

и Крестьянская война Е. Пугачева.

и Внешняя политика России во 2-ой половине XVIII в.

и «Екатерининский век» в истории России.

и Внешняя политика Екатерины II.

и Россия при Павле I.

Дополнительные вопросы для обсуждения и самопроверки:

х Опишите личность Петра I.

х Как проводилась борьба за преобразование традиционного общества в России?



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- в Опишите причины и этапы и итоги Северной войны?
- в В чем состояла культурная реформа Петра I?
- в Как Петр выстраивал регулируемое государство?
- в Охарактеризуйте эпоху дворцовых переворотов.
- в Отметьте главные реформы Екатерины II.
- в Почему произошла крестьянская война Е. Пугачева?
- в Какова внешняя политика России во 2-ой половине XVIII в.?
- в Чем характеризуется «Екатерининский век» в истории России?
- в Отметьте основные направления внешней политики Екатерины II?
- в Как развивалась Россия при Павле I?

Литература:

- 6 Отечественная история: учебник для бакалавров / И.Н. Кузнецов. – 9-е изд., испр. и до-полн., - М.: издательско-торговая корпорация «Дашков и К», 2018. – 816 с.
<https://znanium.com/catalog/product/414990>
- 7 История России: Учебник / Ш.М. Мунчаев, В.М. Устинов. - 7-е изд., перераб. и доп. - Москва : Норма: НИЦ ИНФРА-М, 2018. - 608 с. <https://znanium.com/catalog/product/966207>
- 8 История: учеб. пособие / Ю.А. Шестаков. — М. : РИОР : ИНФРА-М, 2019. — 248 с.
<https://znanium.com/catalog/product/1014722>
- 9 История России XVIII — начала XX века : учебник / М.Ю. Лачаева, Л.М. Ляшенко, В.Е. Воронин, А.П. Синелобов ; под ред. М.Ю. Лачаевой. — Москва : ИНФРА-М, 2019. – 648с.
<https://znanium.com/catalog/product/1023725>
- 10 История: от древних цивилизаций до конца XX в.: Учебник / Оришев А.Б., Тарасенко В.Н. - Москва: ИЦ РИОР, НИЦ ИНФРА-М, 2019. - 276 с.
<https://znanium.com/catalog/product/1048076>

Практическое занятие 4. Россия в XIX веке.

Целью проводимого занятия является формирование знаний: особенности развития России в XIX в.; «золотой век» русской культуры; возникновение новых идейных течений; внешняя политика России в XIX в.; значение реформ второй половины XIX в.; развитие капитализма в России; социальные противоречия в XIX в.; культура во 2-й пол. XIX в.

Форма проведения: контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

Вопросы к контрольной работе:

- 8. Внутренняя и внешняя политика царизма в нач. XIX в.
- 9. Отечественная война 1812 г.
- 11. Кризис крепостнической системы.
- 12. Кавказская война.
- 13. Культура в 1-й четверти XIX века.
- 14. Политическая реакция и реформы в период правления Николая I.
- 15. Крымская война.
- 16. Реформы Александра II.
- 10. Развитие капитализма в пореформенной России.
- 11. Общественные движения России 70-х гг. XIX в.
- 12. Официальная идеология самодержавия и общественная мысль России второй четверти и середины XIX в.
- 13. Причины свертывания реформаторского процесса.
- 14. Особенности пореформенного социально-экономического развития России.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- Национальный вопрос в России и политика властей в конце XIX-начале XX в.
- Основные общественно-политические движения второй половины XIX в. (народничество, домарксистский социализм, либерализм, консерватизм).
- Россия на рубеже XIX-XX веков: особенности социально-экономического развития

Дополнительные вопросы для обсуждения и самопроверки:

- Как развивался царизм в нач. XIX в?
- Охарактеризуйте реформы Александра I в их замысле и реализации.
- Опишите причины, хронику и итоги Отечественной войны 1812 г?
- В чем сказывался кризис крепостнической системы?
- Какими чертами отличалась культура в 1-й четверти XIX века?
- Какие вы знаете реформы Николая I?
- Назовите причины и ход Крымской войны?
- Какова роль реформ Александра II?
- В чем проявляется особенность развитие капитализма в пореформенной России?
- Опишите общественное движение России в 70-х гг 19 века?
- В чем суть идеологии официальной народности?
- Каковы причины свертывания реформаторского процесса?
- Назовите особенности пореформенного социально-экономического развития России.
- В чем проявляются особенности социально-экономического развития?

Литература:

и Отечественная история: учебник для бакалавров / И.Н. Кузнецов. – 9-е изд., испр. и доп. - М.: издательско-торговая корпорация «Дашков и К», 2018. – 816 с.

<https://znanium.com/catalog/product/414990>

и История России: Учебник / Ш.М. Мунчаев, В.М. Устинов. - 7-е изд., перераб. и доп. - Москва : Норма: НИЦ ИНФРА-М, 2018. - 608 с. <https://znanium.com/catalog/product/966207>

и История: учеб. пособие / Ю.А. Шестаков. — М. : РИОР : ИНФРА-М, 2019. — 248 с.

<https://znanium.com/catalog/product/1014722>

и История России XVIII — начала XX века : учебник / М.Ю. Лачаева, Л.М. Ляшенко, В.Е. Воронин, А.П. Синелобов ; под ред. М.Ю. Лачаевой. — Москва : ИНФРА-М, 2019. – 648с.

<https://znanium.com/catalog/product/1023725>

и История: от древних цивилизаций до конца XX в.: Учебник / Оришев А.Б., Тарасенко В.Н. - Москва:ИЦ РИОР, НИЦ ИНФРА-М, 2019. - 276 с.

<https://znanium.com/catalog/product/1048076>

Практическое занятие 5. Россия в первой четверти XX века.

Целью проводимого занятия является формирование знаний: понимание комплексов причин первой и второй русской революции; значение I-й мировой войны; развитие культуры в начале XX века.

Форма проведения: контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

Вопросы к контрольной работе:

ее Личность и деятельность императора Николая II.

ее Российский капитализм в начале XX в.

ее Русско-японская война

ее Революция 1905-1907 гг.: поиск путей общественного прогресса.

ее Россия после поражения революции.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- В Политические партии и интеллигенция об опыте революции и способах модернизации страны. Сборник «Вехи».
- В Реформы Столыпина.
- В Первая мировая война
- В Становление российской многопартийности (до февраля 1917 г.).

Дополнительные вопросы для обсуждения и самопроверки:

8. Охарактеризуйте личность и деятельность императора Николая II?
9. В чем сущность российского капитализма в начале XX в.?
10. Назовите причины, этапы и героев русско-японской войны?
11. В чем суть революция 1905-1907 гг?
12. Какие реформы проводил Столыпин?
13. Какие причины и итоги Первой мировой войны?
14. Чем определяется становление российской многопартийности (до февраля 1917 г.)?

Литература:

9. Отечественная история: учебник для бакалавров / И.Н. Кузнецов. – 9-е изд., испр. и доп. – М.: издательско-торговая корпорация «Дашков и К», 2018. – 816 с.
<https://znanium.com/catalog/product/414990>
10. История России: Учебник / Ш.М. Мунчаев, В.М. Устинов. - 7-е изд., перераб. и доп. - Москва : Норма: НИЦ ИНФРА-М, 2018. - 608 с.
<https://znanium.com/catalog/product/966207>
11. История: учеб. пособие / Ю.А. Шестаков. — М. : РИОР : ИНФРА-М, 2019. — 248 с. <https://znanium.com/catalog/product/1014722>
12. История России XVIII — начала XX века : учебник / М.Ю. Лачаева, Л.М. Ляшенко, В.Е. Воронин, А.П. Синелобов ; под ред. М.Ю. Лачаевой. — Москва : ИНФРА-М, 2019. — 648с. <https://znanium.com/catalog/product/1023725>
13. История: от древних цивилизаций до конца XX в.: Учебник / Оришев А.Б., Тарасенко В.Н. - Москва:ИЦ РИОР, НИЦ ИНФРА-М, 2019. - 276 с.
<https://znanium.com/catalog/product/1048076>

Модуль II

Практическое занятие 6. Революция 1917г. СССР в 20-30 годы XX века.

Целью проводимого занятия является формирование знаний: цивилизационное измерение революции 1917 г.; социалистическая идеология; хроника гражданской войны; судьба последнего русского царя Николая II; государственное строительство Советской России; особенность НЭПа; Сталин и его внутренняя, экономическая и внешняя политика; формировании тоталитарного сталинского режима; политика индустриализации и коллективизации; международный статус Советской России.

Форма проведения: контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

Вопросы к контрольной работе:

- 9) Причины революции в России
- 10) Политические силы в Февральской революции.
- 11) Расклад власти в 1917 году
- 12) Октябрьская революция 1917 года: идеология и властные методы большевиков.
- 13) Причины гражданской войны.
- 14) Гражданская война в России: причины, участники, их идеология и политика, итоги.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- В Новая экономическая политика.
- В Национально-государственное строительство и образование СССР.
- В Предпосылки возникновения тоталитарного режима.
- В Формирование однопартийного политического режима.
- В «Великий перелом»: сущность, практика и итоги.
- В Форсирование социально-экономических процессов на рубеже 20-х-30-х гг. XX в.
- В Демонтаж НЭПа и политика «большого скачка».
- В Насильственная коллективизация и её последствия.
- В Ускоренный рынок в развитии промышленности
- В Проблема темпов в социалистическом строительстве.
- В Индустриализация и ее результаты.
- В Аграрная политика конца 20-х – начала 30-х годов XX в.
- В Насильственная сплошная коллективизация, ее истоки и последствия для страны.
- В Общественный строй СССР в 30-е годы XX в. Современные оценки.
- В Режим личной власти Сталина. Сопротивление сталинизму.
- В Внешняя политика СССР в 20-30-х гг. XX в. (до 1939 г.).

Дополнительные вопросы для обсуждения и самопроверки:

6. Назовите причины революционных событий в России.
7. Какие политические силы участвовали в Февральской революции?
8. Какие движущие силы октябрьской революции 1917 года?
9. Изложите идеологию и властные методы большевиков.
10. Каковы причины гражданской войны?
11. Определите идеологические последствия революции 1917 года.
12. Назовите основные программные направления новой экономической политики?
13. Какая была сформирована национально-государственная структура в СССР?
14. Каковы предпосылки возникновения тоталитарного режима?
15. В чем специфика однопартийного политического режима в Советском Союзе?
16. Как осуществлялось форсирование социально-экономических процессов на рубеже 20-х – 30-х гг?
17. Поясните проблемы демонтажа НЭПа и политику «большого скачка».
18. Зачем большевики проводили насильственную коллективизацию?
19. В чем состояло ускорение в развитии промышленности?
20. Охарактеризуйте общественный строй СССР в 30-е годы? Дайте ему современные оценки?
21. Изложите внешнюю политику СССР в 20-30-х гг. (до 1939 г.)?

Литература:

- 81 Отечественная история: учебник для бакалавров / И.Н. Кузнецов. – 9-е изд., испр. и доп., - М.: издательско-торговая корпорация «Дашков и К», 2018. – 816 с.
<https://znanium.com/catalog/product/414990>
- 82 История России: Учебник / Ш.М. Мунчаев, В.М. Устинов. - 7-е изд., перераб. и доп. - Москва : Норма: НИЦ ИНФРА-М, 2018. - 608 с. <https://znanium.com/catalog/product/966207>
- 83 История: учеб. пособие / Ю.А. Шестаков. — М. : РИОР : ИНФРА-М, 2019. — 248 с.
<https://znanium.com/catalog/product/1014722>
- 84 История России XVIII — начала XX века : учебник / М.Ю. Лачаева, Л.М. Ляшенко, В.Е. Воронин, А.П. Синелобов ; под ред. М.Ю. Лачаевой. — Москва : ИНФРА-М, 2019. — 648с.
<https://znanium.com/catalog/product/1023725>



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

6. История: от древних цивилизаций до конца XX в.: Учебник / Оришев А.Б., Тарасенко В.Н. - Москва:ИЦ РИОР, НИЦ ИНФРА-М, 2019. - 276 с.
<https://znanium.com/catalog/product/1048076>

Практическое занятие 7. Великая Отечественная война.

Целью проводимого занятия является формирование знаний: генезис фашистской идеологии / ее развитие в Германии; начало II-й мировой войны; начало и основные периоды, сражения, Великой Отечественной войны; значение победы в войне для дальнейшего развития СССР.

Форма проведения: контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

Вопросы к контрольной работе:

- / Причины II Мировой войны.
- / Основные этапы и боевые события ВОВ
- / Итоги Великой Отечественной войны.

Дополнительные вопросы для обсуждения и самопроверки:

- / Укажите причины II Мировой войны?
- / Объясните основные этапы и боевые события ВОВ?
- / Каковы итоги Великой Отечественной войны?

Литература:

- / Отечественная история: учебник для бакалавров / И.Н. Кузнецов. – 9-е изд., испр. и доп., - М.: издательско-торговая корпорация «Дашков и К», 2018. – 816 с.
<https://znanium.com/catalog/product/414990>
- / История России: Учебник / Ш.М. Мунчаев, В.М. Устинов. - 7-е изд., перераб. и доп. - Москва : Норма: НИЦ ИНФРА-М, 2018. - 608 с. <https://znanium.com/catalog/product/966207>
- / История: учеб. пособие / Ю.А. Шестаков. — М. : РИОР : ИНФРА-М, 2019. — 248 с.
<https://znanium.com/catalog/product/1014722>
- / История России XVIII — начала XX века : учебник / М.Ю. Лачаева, Л.М. Ляшенко, В.Е. Воронин, А.П. Синелобов ; под ред. М.Ю. Лачаевой. — Москва : ИНФРА-М, 2019. – 648с.
<https://znanium.com/catalog/product/1023725>
- / История: от древних цивилизаций до конца XX в.: Учебник / Оришев А.Б., Тарасенко В.Н. - Москва:ИЦ РИОР, НИЦ ИНФРА-М, 2019. - 276 с.
<https://znanium.com/catalog/product/1048076>

Практическое занятие 8. СССР в 50-80 гг. XX века

Целью проводимого занятия является формирование знаний: развитие СССР в послевоенные годы; холодная война и геополитическое устройство в XX в.; основные внутриполитические периоды развития советской истории; культура в СССР.

Форма проведения: контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

Вопросы к контрольной работе:

- В Противоречия развития страны в 1953-1985 гг.
- В Вторая мировая война и поляризация послевоенного мира.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- \endash Холодная война.
- \endash Первые попытки демократических преобразований.
- \endash Политика «оттепели» Хрущева.
- \endash Разоблачение культа личности.
- \endash Нарастание застойных явлений в социально-экономическом и общественно-политическом развитии страны.
- \endash Внешняя политика СССР в 60-70-е гг
- \endash Социально-экономическое и политическое развитие СССР в середине 60-х - начале 80-х гг. XXв.

Дополнительные вопросы для обсуждения и самопроверки:

15. Обоснуйте, в чем состояли противоречия развития страны в 1953-1985 гг.?
16. Каково значение холодной войны для внешней политики СССР?
17. В чем был замысел политики «оттепели» Хрущева?
18. Что означает разоблачение культа личности?
19. В чем проявляло себя нарастание застойных явлений в социально-экономическом и общественно-политическом развитии страны в 60-70-е гг XX в.?
20. Опишите социально-экономическое и политическое развитие СССР в середине 60-х - начале 80-х гг XXв.?

Литература:

18. Отечественная история: учебник для бакалавров / И.Н. Кузнецов. – 9-е изд., испр. и до-полн., - М.: издательско-торговая корпорация «Дашков и К», 2018. – 816 с.
<https://znanium.com/catalog/product/414990>
19. История России: Учебник / Ш.М. Мунчаев, В.М. Устинов. - 7-е изд., перераб. и доп. - Москва : Норма: НИЦ ИНФРА-М, 2018. - 608 с.
<https://znanium.com/catalog/product/966207>
20. История: учеб. пособие / Ю.А. Шестаков. — М. : РИОР : ИНФРА-М, 2019. — 248 с. <https://znanium.com/catalog/product/1014722>
21. История России XVIII — начала XX века : учебник / М.Ю. Лачаева, Л.М. Ляшенко, В.Е. Воронин, А.П. Синелобов ; под ред. М.Ю. Лачаевой. — Москва : ИНФРА-М, 2019. – 648с. <https://znanium.com/catalog/product/1023725>
22. История: от древних цивилизаций до конца XX в.: Учебник / Оришев А.Б., Тарасенко В.Н. - Москва:ИЦ РИОР, НИЦ ИНФРА-М, 2019. - 276 с.
<https://znanium.com/catalog/product/1048076>

Практическое занятие 9. Перестройка: от частичных преобразований к смене модели общественного развития. Современная Россия

Целью проводимого занятия является формирование знаний: причины распада СССР; смысл политики перестройки; либерализация экономической системы России; идеологическое брожение в 90-е гг. XX в.; развитие современной России.

Форма проведения: контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

Вопросы к контрольной работе:

12. Перестройка в СССР: причины, сущность, этапы, реализация и итоги.
13. Парад суверенитетов.
14. Чеченская война.
15. Курс правительства России на ускорение перехода к рыночной экономике.



4. Россия на современном этапе.

Дополнительные вопросы для обсуждения и самопроверки:

16. Изложите причины необходимости проведения реформ и перестройки в СССР?
17. Как проходил парад суверенитетов? Каковы его современные последствия?
18. Укажите сущность и хронику Чеченской войны?
19. Поясните, в чем особенность современного курса правительства России?
20. Как можно охарактеризовать историю России на современном этапе?

Литература:

№Отечественная история: учебник для бакалавров / И.Н. Кузнецов. – 9-е изд., испр. и до-полн., - М.: издательско-торговая корпорация «Дашков и К», 2018. – 816 с.

<https://znanium.com/catalog/product/414990>

№История России: Учебник / Ш.М. Мунчаев, В.М. Устинов. - 7-е изд., перераб. и доп. - Москва : Норма: НИЦ ИНФРА-М, 2018. - 608 с. <https://znanium.com/catalog/product/966207>

№История: учеб. пособие / Ю.А. Шестаков. — М. : РИОР : ИНФРА-М, 2019. — 248 с. <https://znanium.com/catalog/product/1014722>

№История России XVIII — начала XX века : учебник / М.Ю. Лачаева, Л.М. Ляшенко, В.Е. Воронин, А.П. Синелобов ; под ред. М.Ю. Лачаевой. — Москва : ИНФРА-М, 2019. – 648с. <https://znanium.com/catalog/product/1023725>

№История: от древних цивилизаций до конца XX в.: Учебник / Оришев А.Б., Тарасенко В.Н. - Москва:ИЦ РИОР, НИЦ ИНФРА-М, 2019. - 276 с. <https://znanium.com/catalog/product/1048076>

2. Методические указания к практическим занятиям заочной формы обучения

Практическое занятие 1. Славяне. Киевский и Московский периоды русской истории.

Целью проводимого занятия является формирование знаний: изучение начального и средне-векового периодов русской истории; уяснение генезиса русской государственности, эконо-мической системы, идеологии; хроники правления первых князей; основные категории раз-вития древней и средневековой русской истории; понимание особенности развития России в 17 в.; внутренняя и внешняя политика первых Романовых; культура Киевской и Московской Руси.

Форма проведения: контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

Вопросы к контрольной работе:

22. Проблемы этногенеза славян.
23. Происхождение Российского государства.
24. Хроника первых веков
25. Феодалная раздробленность и иностранная интервенция на Русь в XII-XV вв.
26. Монголо-татары и Русь: проблемы взаимовлияния. Было ли иго? Современные оценки.
27. Политика Московских князей.
28. Личность и деятельность Ивана III.
29. Воссоединение земель вокруг Москвы в XIV- первой половине XV вв.
30. Московское царство в XVI в. Иван IV.
31. Причины и ход Смутного времени.
32. Россия в XVII веке.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

12. Культура Древней и Московской Руси

Дополнительные вопросы для обсуждения и самопроверки:

9. В чем состоят основные проблемы в выявлении этногенеза славян?
10. Какова природа государства и признаки образования государства Киевская Русь?
11. Изложите специфику дискуссии вокруг норманской теории?
12. Расскажите хронику истории первых князей?
13. Обоснуйте идею высокоразвитости культуры Древней Руси?
14. Какова была внешняя политика Руси в XI-XIII вв.?
15. Изложите особенность борьбы северо-восточной Руси с крестоносцами?
16. Изложите основные факторы феодальной раздробленности на территории Руси в XII - XIII вв.?
17. Укажите на специфику экономического развития древней Руси?.
18. Какие существовали проблемы во взаимоотношениях Орды и Руси?
19. Опишите основные направления политики Московских князей?
20. Каковы были предпосылки и альтернативы воссоединения русских земель?
21. Расскажите хронику Куликовской битвы?
22. В чем состояла внешняя политика Ивана III и Василия III?
23. Расскажите об особенностях личности царя Ивана IV.
24. В чем суть местничества?
25. Изложите основные реформы внутренней политики Ивана IV?
26. Каковы причины и характер Смутного времени в России?
27. Назовите основные социально-экономические и политические особенности развития России после Смутного времени XVII в.?
28. В чем особенность Соборного уложения 1649 г.?
29. Каково место Крестьянской войны под предводительством С. Разина в истории России XVII века?
30. В чем сущность Церковного раскола?
31. Опишите специфику культуры России, традиции и новации, отношение государства и церкви в России в XVI-XVII вв.?

Литература:

23. Отечественная история: учебник для бакалавров / И.Н. Кузнецов. – 9-е изд., испр. и до-полн., - М.: издательско-торговая корпорация «Дашков и К», 2018. – 816 с.
<https://znanium.com/catalog/product/414990>
24. История России: Учебник / Ш.М. Мунчаев, В.М. Устинов. - 7-е изд., перераб. и доп. - Москва : Норма: НИЦ ИНФРА-М, 2018. - 608 с.
<https://znanium.com/catalog/product/966207>
25. История: учеб. пособие / Ю.А. Шестаков. — М. : РИОР : ИНФРА-М, 2019. — 248 с. <https://znanium.com/catalog/product/1014722>
26. История России XVIII — начала XX века : учебник / М.Ю. Лачаева, Л.М. Ляшенко, В.Е. Воронин, А.П. Синелобов ; под ред. М.Ю. Лачаевой. — Москва : ИНФРА-М, 2019. — 648с. <https://znanium.com/catalog/product/1023725>
27. История: от древних цивилизаций до конца XX в.: Учебник / Оришев А.Б., Тарасенко В.Н. - Москва:ИЦ РИОР, НИЦ ИНФРА-М, 2019. - 276 с.
<https://znanium.com/catalog/product/1048076>

Практическое занятие 2. Советская Россия.

Целью проводимого занятия является формирование знаний: особенности развития России в 16. в.; специфика формирования советского государства в 20-30-е гг. XX века; знание причин и хода ВОВ; развитие СССР в 50-80-е гг. XX века.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Форма проведения: контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

Вопросы к контрольной работе:

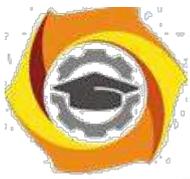
- В СССР в 20-30-е гг. 20 века.
- В Великая Отечественная война.
- В СССР в 50-80-е гг. 20 века.
- В Современный этап развития русской истории.

Дополнительные вопросы для обсуждения и самопроверки:

- 000 Назовите причины революционных событий в России.
- 000 Какие политические силы участвовали в Февральской революции?
- 000 Какие движущие силы октябрьской революции 1917 года?
- 000 Изложите идеологию и властные методы большевиков.
- 000 Каковы причины гражданской войны?
- 000 Определите идеологические последствия революции 1917 года.
- 000 Назовите основные программные направления новой экономической политики?
- 000 Какая была сформирована национально-государственная структура в СССР?
- 000 Каковы предпосылки возникновения тоталитарного режима?
- 000 В чем специфика однопартийного политического режима в Советском Союзе?
- 000 Как осуществлялось форсирование социально-экономических процессов на рубеже 20-х – 30-х гг?
- 000 Поясните проблемы демонтажа НЭПа и политику «большого скачка».
- 000 Зачем большевики проводили насильственную коллективизацию?
- 000 В чем состояло ускорение в развитии промышленности?
- 000 Охарактеризуйте общественный строй СССР в 30-е годы? Дайте ему современные оценки?
- 000 Изложите внешнюю политику СССР в 20-30-х гг. (до 1939 г.)?
- 000 Укажите причины II Мировой войны?
- 000 Объясните основные этапы и боевые события ВОВ?
- 000 Каковы итоги Великой Отечественной войны?
- 000 Обоснуйте, в чем состояли противоречия развития страны в 1953-1985 гг.?
- 000 Каково значение холодной войны для внешней политики СССР?
- 000 В чем был замысел политики «оттепели» Хрущева?
- 000 Что означает разоблачение культа личности?
- 000 В чем проявляло себя нарастание застойных явлений в социально-экономическом и общественно-политическом развитии страны в 60-70-е гг?
- 000 Опишите социально-экономическое и политическое развитие СССР в середине 60-х - начале 80-х гг?

Литература:

9. Отечественная история: учебник для бакалавров / И.Н. Кузнецов. – 9-е изд., испр. и доп. – М.: издательско-торговая корпорация «Дашков и К», 2018. – 816 с.
<https://znanium.com/catalog/product/414990>
10. История России: Учебник / Ш.М. Мунчаев, В.М. Устинов. - 7-е изд., перераб. и доп. - Москва : Норма: НИЦ ИНФРА-М, 2018. - 608 с.
<https://znanium.com/catalog/product/966207>
11. История: учеб. пособие / Ю.А. Шестаков. — М. : РИОР : ИНФРА-М, 2019. — 248 с. <https://znanium.com/catalog/product/1014722>



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

20. История России XVIII — начала XX века : учебник / М.Ю. Лачаева, Л.М. Ляшенко, В.Е. Воронин, А.П. Синелобов ; под ред. М.Ю. Лачаевой. — Москва : ИНФРА-М, 2019. — 648с.
<https://znanium.com/catalog/product/1023725>

21. История: от древних цивилизаций до конца XX в.: Учебник / Оришев А.Б., Тарасенко В.Н. - Москва:ИЦ РИОР, НИЦ ИНФРА-М, 2019. - 276 с.
<https://znanium.com/catalog/product/1048076>

III. Методические рекомендации по курсу

Курс по дисциплине «История» рассчитан на изучение в течение одного семестра, включает лекционные, практические занятия и самостоятельную работу студентов.

Для контроля уровня сформированности компетенций, качества знаний, умений и навыков особое внимание уделяется стимулированию самостоятельной работы студентов. Содержание дисциплины «История» разделено на два модуля, по окончании изучения которых осуществляется текущий контроль усвоения учебного материала. В течение семестра проводятся модульно-рейтинговые мероприятия, в том числе, и в форме компьютерного тестирования для проверки самостоятельной работы студентов.

Изучение Истории является трудоемкой задачей, в связи с чем, предусматривается обязательное посещение лекционных занятий для получения системы целостных знаний. Рекомендованные учебники и учебные пособия не всегда являются достаточными источниками знаний, так как, во-первых, учебный курс носит междисциплинарный характер,

23. во-вторых, лекционный материал непосредственно дает квинтэссенцию ответов на экзаменационные вопросы.

После лекций проводятся практические занятия, они, прежде всего, предназначены для закрепления теоретических знаний и выработки таких практических навыков, как, аргументированное отстаивание собственного видения рассматриваемых проблем, владение приемами ведения дискуссии, полемики и диалога. Специфика курса заключается в том, что многие темы требуют дискуссионного обсуждения. На практические занятия выносятся материал, дополняющий и углубляющий лекционный, а также, материал, не вошедший в лекционный курс, поэтому при подготовке к практическим занятиям необходимо самостоятельное про-чтение и осмысление рекомендуемых работ.

Темы курса рассчитаны на закрепление и систематизацию лекционного материала, самостоятельное освоение дополнительной литературы, а также, нацелены на формирование навыков по практическому применению полученных знаний в сфере образования и общественной коммуникации.

Дисциплина «История» призвана сформировать историческое мышление будущего специалиста, а также заложить методологические и духовно-нравственные установки для его формирования.

Методические указания по выполнению курсовой работы по дисциплине

Введение

Проблемы управления программными проектами впервые проявились в 60х начале 70х годов, когда провалились многие большие проекты по разработке программных продуктов. Были зафиксированы задержки в создании ИС, оно было



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ненадежным, затраты на разработку в несколько раз превосходили первоначальные оценки, созданные программные системы часто имели низкие показатели производительности. Причины провалов коренились в тех подходах, которые использовались в управлении проектами. Применяемая методика была основана на опыте управления техническими проектами и оказалась неэффективной при разработке программного обеспечения.

Имеется существенная разница между профессиональной разработкой ИС и любительским программированием. Необходимость управления программными проектами вытекает из того факта, что процесс создания профессионального ПО всегда является субъектом бюджетной политики организации, где оно разрабатывается, и имеет временные ограничения. Работа *руководителя программного проекта* по большому счету заключается в том, чтобы гарантировать выполнение этих бюджетных и временных ограничений с учетом бизнес целей организации относительно разрабатываемой ИС.

Руководители проектов призваны спланировать все этапы разработки программного продукта. Они также должны контролировать ход выполнения работ и соблюдения всех требуемых стандартов. Постоянный контроль за ходом выполнения работ необходим для того, чтобы процесс разработки не выходил за временные и бюджетные ограничения. Хорошее управление не гарантирует успешного завершения проекта, но плохое управление обязательно приведет к его провалу. Это может выразиться в задержке сроков сдачи готовой ИС, в превышении сметной стоимости проекта и в несоответствии готового ПО спецификации требований.



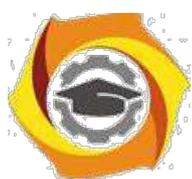
Процесс разработки ИС существенно отличается от процессов реализации технических проектов, что порождает определенные сложности в управлении программными проектами:

в *Программный продукт нематериален.* Программное обеспечение нематериально, его нельзя увидеть или потрогать. Руководитель программного проекта не видит процесс "роста" разрабатываемого ПО. Он может полагаться только на документацию, которая фиксирует процесс разработки программного продукта.

В *Не существует стандартных процессов разработки ПО.* На сегодняшний день не существует четкой зависимости между процессом создания ПО и типом создаваемого программного продукта. Другие технические дисциплины имеют длительную историю, процессы разработки технических изданий многократно опробованы и проверены. Процессы создания большинства технических систем хорошо изучены. Изучением же процессов создания ПО специалисты занимаются только последнее время. Поэтому пока нельзя точно предсказать, на каком этапе процесса разработки ПО могут возникнуть проблемы, угрожающие всему программному проекту.

с *Большие программные проекты это часто "одноразовые" проекты.* Большие программные проекты, как правило, значительно отличаются от проектов, реализованных ранее. Поэтому, чтобы уменьшить неопределенность в планировании проекта, руководители проектов должны обладать очень большим практическим опытом. Но постоянные технологические изменения в компьютерной технике и коммуникационном оборудовании обесценивают предыдущий опыт. Знания и навыки, накопленные опытом, могут не востребоваться в новом проекте.

Перечисленные отличия могут привести к тому, что реализация проекта выйдет из временного графика или превысит бюджетные ассигнования. Программные системы зачастую оказываются новинками как в "идеологическом", так и в техническом плане. Поэтому, предвидя возможные проблемы в реализации программного проекта, следует всегда помнить, что многим из них свойственно выходить за рамки временных и бюджетных ограничений.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Процесс управления разработкой информационной системы



Невозможно описать и стандартизировать все работы, выполняемые в проекте по созданию ИС. Эти работы весьма существенно зависят от организации, где выполняется разработка ИС, и от типа создаваемого программного продукта. Но всегда можно выделить следующие:

Написание предложений по созданию ПО.

Планирование и составление графика работ по созданию ПО.

Оценивание стоимости проекта.

Подбор персонала.

Контроль за ходом выполнения работ.

Написание отчетов и представлений.

Первая стадия программного проекта может состоять из написания предложений по реализации этого проекта. Предложения должны содержать описание целей проектов и способов их достижения. Они также обычно включают в себя оценки финансовых и временных затрат на выполнение проекта. При необходимости здесь могут приводиться обоснования для передачи проекта на выполнение сторонней организации или команде разработчиков.

Написание предложений — очень ответственная работа, так как для многих организаций вопрос о том, будет ли проект выполняться самой организацией или разрабатываться по контракту сторонней компанией, является критическим. Не существует каких либо рекомендаций по написанию предложений, многое здесь зависит от опыта.

На этапе *планирования проекта* определяются процессы, этапы и полученные на каждом из них результаты, которые должны привести к выполнению проекта. Реализация этого плана приведет к достижению целей проекта. Определение стоимости проекта напрямую связано с его планированием, поскольку здесь оцениваются ресурсы, требующиеся для выполнения плана.

Контроль за ходом выполнения работ (мониторинг проекта) — это непрерывный процесс, продолжающийся в течение всего срока реализации проекта. Руководитель должен постоянно отслеживать ход реализации проекта и сравнивать



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

фактические и плановые показатели выполнения работ с их стоимостью. Хотя многие организации имеют механизмы формального мониторинга работ, опыт-



ный руководитель может составить ясную картину о стадии развитии проекта просто путем неформального общения с разработчиками.

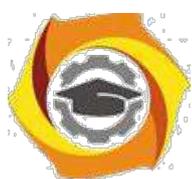
Неформальный мониторинг часто помогает обнаружить потенциальные проблемы, которые в явном виде могут обнаружиться позднее. Например, ежедневное обсуждение хода выполнения работ может выявить отдельные недоработки в создаваемом программном продукте. Вместо ожидания отчетов, в которых будет отражен факт "пробуксовки" графика работ, можно обсудить со специалистами намечающиеся программистские проблемы и не допустить срыва графика работ.

х течение реализации проекта обычно происходит несколько формальных контрольных проверок хода выполнения работ по созданию ПО. Такие проверки должны дать общую картину хода реализации проекта в целом и показать, насколько уже разработанная часть ПО соответствует целям проекта.

Время выполнения больших программных проектов может занимать несколько лет. В течение этого времени цели и намерения организации, заказавшей программный проект, могут существенно измениться. Может оказаться, что разрабатываемый программный продукт стал уже ненужным либо исходные требования к создаваемому ПО просто устарели и их необходимо кардинально менять. в такой ситуации руководство организации разработчика может принять решение о прекращении разработки ПО или об изменении проекта в целом с тем, чтобы учесть изменившиеся цели и намерения организации заказчика.

Руководители проектов обычно обязаны сами *подбирать исполнителей* для своих проектов. В идеальном случае профессиональный уровень исполнителей должен соответствовать той работе, которую они будут выполнять в ходе реализации проекта. Однако во многих случаях руководители должны полагаться на команду разработчиков, которая далека от идеальной. Такая ситуация может быть вызвана следующими причинами:

11 Бюджет проекта не позволяет привлечь высококвалифицированный персо-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

нал. В таком случае за меньшую плату привлекаются менее квалифицированные специалисты.



10. Бывают ситуации, когда невозможно найти специалистов необходимой квалификации как в самой организации разработчике, так и вне ее. Например, в организации "лучшие люди" могут быть уже заняты в других проектах.

17. Организация хочет повысить профессиональный уровень своих работников. В этом случае она может привлечь к участию в проекте неопытных или недостаточно квалифицированных работников, чтобы они приобрели необходимый опыт и поучились у более опытных специалистов.

Таким образом, почти всегда подбор специалистов для выполнения проекта имеет определенные ограничения и не является свободным. Вместе с тем необходимо, чтобы хотя бы несколько членов группы разработчиков имели квалификацию и опыт, достаточные для работы над данным проектом. В противном случае невозможно избежать ошибок в разработке ПО.

Руководитель проекта обычно обязан посылать *отчеты* о ходе его выполнения как заказчику, так и подрядным организациям. Это должны быть краткие документы, основанные на информации, извлекаемой из подробных отчетов о проекте. В этих отчетах должна быть та информация, которая позволяет четко оценить степень готовности создаваемого программного продукта.

15. в рамках дисциплины «управление жизненным циклом информационных систем» выделены следующие роли в группе по разработке ПО ИС:

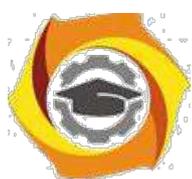
Руководитель – общее руководство проектом, написание документации, общение с заказчиком ИС

Системный аналитик – разработка требований (составление технического задания, проекта программного обеспечения)

Тестер – составление плана тестирования и аттестации готового ПО (продукта), составление сценария тестирования, базовый пример, проведение мероприятий по плану тестирования

Разработчик – моделирование компонент программного обеспечения, кодирование

Планирование проекта разработки программного обеспечения



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Эффективное управление жизненным циклом информационной системы
напрямую зависит от правильного планирования работ, необходимых для его вы-

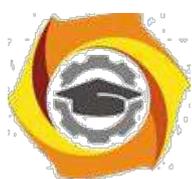


полнения. План помогает руководителю предвидеть проблемы, которые могут возникнуть на каких либо этапах создания ПО, и разработать превентивные меры для их предупреждения или решения. План, разработанный на начальном этапе проекта, рассматривается всеми его участниками как руководящий документ, выполнение которого должно привести к успешному завершению проекта. Этот первоначальный план должен максимально подробно описывать все этапы реализации проекта.

Процесс планирования начинается, исходя из описания системы, с определения проектных ограничений (временные ограничения, возможности наличного персонала, бюджетные ограничения и т.д.). Эти ограничения должны определяться параллельно с оцениванием проектных параметров, таких как структура и размер проекта, а также распределением функций среди исполнителей. Затем определяются этапы разработки и то, какие результаты документация, прототипы, подсистемы или версии программного продукта) должны быть получены по окончании этих этапов. Далее начинается циклическая часть планирования. Сначала разрабатывается график работ по выполнению проекта или дается разрешение на продолжение использования ранее созданного графика. После этого проводится контроль выполнения работ и отмечаются расхождения между реальным и плановым ходом работ.

Далее, по мере поступления новой информации о ходе выполнения проекта, возможен пересмотр первоначальных оценок параметров проекта. Это, в свою очередь, может привести к изменению графика работ. Если в результате этих изменений нарушаются сроки завершения проекта, должны быть пересмотрены (и согласованы с заказчиком ПО) проектные ограничения.

Конечно, большинство руководителей проектов не думают, что реализация их проектов пройдет гладко, без всяких проблем. Желательно описать возможные проблемы еще до того, как они проявят себя в ходе выполнения проекта. Поэтому лучше составлять "пессимистические" графики работ, чем "оптимистические". Но,



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

конечно, невозможно построить план, учитывающий все, в том числе случай-ные, проблемы и задержки выполнения проекта, поэтому и возникает необходи-

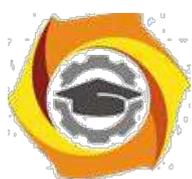


мость периодического пересмотра проектных ограничений и этапов создания программного продукта.

План проекта должен четко показать ресурсы, необходимые для реализации проекта, разделение работ на этапы и временной график выполнения этих этапов. В некоторых организациях план проекта составляется как единый документ, содержащий все виды планов, описанных выше. В других случаях план проекта описывает только технологический процесс создания ПО. В таком плане обязательно присутствуют ссылки на планы других видов, но они разрабатываются отдельно от плана проекта.

Детализация планов проектов очень различается в зависимости от типа разрабатываемого программного продукта и организационной структуры разработчика. Но в любом случае большинство планов содержат следующие разделы.

- *Введение.* Краткое описание целей проекта и проектных ограничений (бюджетных, временных и т.д.), которые важны для управления проектом.
 - *Организация выполнения проекта.* Описание способа подбора команды разработчиков и распределение обязанностей между членами команды.
 - и *Анализ рисков.* Описание возможных проектных рисков, вероятности их проявления и стратегий, направленных на их уменьшение.
 - и *Аппаратные и программные ресурсы, необходимые для реализации проекта.* Перечень аппаратных средств и программного обеспечения, необходимого для разработки программного продукта. Если аппаратные средства требуется закупать, приводится их стоимость совместно с графиком закупки ее поставки.
 - В *Разбиение работ на этапы.* Процесс реализации проекта разбивается на отдельные процессы, определяются этапы выполнения проекта, приводится описание результатов ("выходов") каждого этапа и контрольные отметки.
15. *График работ.* В этом графике отображаются зависимости между отдельными процессами (этапами) разработки ПО, оценки времени их выполнения



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

14. распределение членов команды разработчиков по отдельным этапам.



15) *Механизмы мониторинга и контроля за ходом выполнения проекта.* Описываются предоставляемые руководителем отчеты о ходе выполнения работ, сроки их предоставления, а также механизмы мониторинга всего проекта.

План должен регулярно пересматриваться в процессе реализации проекта. Одни части плана, например график работ, изменяются часто, другие более стабильны. Для внесения изменений в план требуется специальная организация документопотока, позволяющая отслеживать эти изменения.

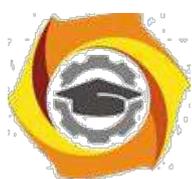
Общие сведения о требованиях к информационным системам

Проблемы, которые приходится решать специалистам в процессе создания программного обеспечения, очень сложны. Природа этих проблем не всегда ясна, особенно если разрабатываемая программная система инновационная. В частности, трудно чётко описать те действия, которые должна выполнять система. Описание функциональных возможностей и ограничений, накладываемых на систему, называется требованиями к этой системе, а сам процесс формирования, анализа, документирования и проверки этих функциональных возможностей и ограничений – разработкой требований.

Требования подразделяются на пользовательские и системные. Пользовательские требования – это описание на естественном языке (плюс поясняющие диаграммы) функций, выполняемых системой, и ограничений, накладываемых на неё. Системные требования – это описание особенностей системы (архитектура системы, требования к параметрам оборудования и т.д.), необходимых для эффективной реализации требований пользователя.

Первые шаги по разработке требований к информационным системам анализ осуществимости

Разработка требований — это процесс, включающий мероприятия, необходимые для создания и утверждения документа, содержащего спецификацию системных требований. Для новых программных систем процесс разработки требований должен начинаться с анализа осуществимости. Началом такого анализа яв-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ляется общее описание системы и ее назначения, а результатом анализа — отчет, в котором должна быть четкая рекомендация, продолжать или нет процесс разра-



ботки требований проектируемой системы. Другими словами, анализ осуществимости должен осветить следующие вопросы.

В Отвечает ли система общим и бизнес целям организации заказчика и организации разработчика?

22. Можно ли реализовать систему, используя существующие на данный момент технологии и не выходя за пределы заданной стоимости?

85. Можно ли объединить систему с другими системами, которые уже эксплуатируются?

Критическим является вопрос, будет ли система соответствовать целям организации. Если система не соответствует этим целям, она не представляет никакой ценности для организации. В то же время многие организации разрабатывают системы, не соответствующие их целям, либо не совсем ясно понимая эти цели, либо под влиянием политических или общественных факторов.

Проблемы, которые приходится решать специалистам в процессе создания, информационных систем очень сложны. Природа этих проблем не всегда ясна, особенно если разрабатываемая программная система инновационная. В частности, трудно чётко описать те действия, которые должна выполнять система. Описание функциональных возможностей и ограничений, накладываемых на систему, называется требованиями к этой системе, а сам процесс формирования, анализа, документирования и проверки этих функциональных возможностей и ограничений – разработкой требований.

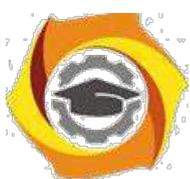
Требования подразделяются на пользовательские и системные. Пользовательские требования – это описание на естественном языке (плюс поясняющие диаграммы) функций, выполняемых системой, и ограничений, накладываемых на неё. Системные требования – это описание особенностей системы (архитектура системы, требования к параметрам оборудования и т.д.), необходимых для эффективной реализации требований пользователя.

Разработка требований



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Разработка требований — это процесс, включающий мероприятия, необходимые для создания и утверждения документа, содержащего спецификацию си-



стемных требований. Различают четыре основных этапа процесса разработки требований:

7. анализ технической осуществимости создания системы,
8. формирование и анализ требований,
9. специфицирование требований и создание соответствующей документации,
10. аттестация этих требований.

Основным документом, регламентирующим взаимоотношения ИС-службы и бизнес-подразделений предприятия, является соглашение об уровне сервиса (Service Level Agreement – SLA). В данном документе дается качественное и количественное описание ИТ-сервисов, как с точки зрения службы ИС, так и с точки зрения бизнес-подразделений.

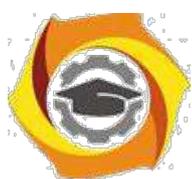
Соглашение об уровне сервиса определяет взаимные ответственности поставщика ИТ-сервиса и пользователей этого сервиса.

Типовая модель SLA должна включать следующие разделы:

- / определение предоставляемого сервиса, стороны, вовлеченные в соглашение, и сроки действия соглашения;
- / доступность ИТ-сервиса;
- / число и размещение пользователей и/или оборудования, использующих данный ИТ-сервис;
- / описание процедуры отчетов о проблемах;
- / описание процедуры запросов на изменение.

Спецификации целевых уровней качества сервиса, включая:

- / средняя доступность, выраженная как среднее число сбоев на период предоставления сервиса;
- / минимальная доступность для каждого пользователя;
- / среднее время отклика сервиса;
- / максимальное время отклика для каждого пользователя;
- / средняя пропускная способность;
- / описания расчета приведенных выше метрик и частоты отчетов;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

/ описание платежей, связанных с сервисом;



- / ответственности клиентов при использовании сервиса (подготовка, поддержка соответствующих конфигураций оборудования, программного обеспечения или изменения только в соответствии с процедурой изменения);
- / процедура разрешения споров, связанных с предоставлением сервиса.

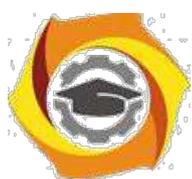
Существенной частью SLA является каталог сервисов. Каталог ИТ-сервисов представляет собой документ, в котором сформулированы все ИТ-сервисы, предоставляемые пользователям, при необходимости указывается цена услуги, общий порядок обращения за услугой. Каталог включает информацию описательную и операционную.

Как правило, в описывающей части содержится следующая информация:

- / имя сервиса;
- / ссылки на связанные сервисы;
- / описание сервисов, функций, границ предоставления сервисов, профилей пользователей;
- / поддерживаемые платформы или инфраструктуры;
- / характеристики доступности, производительности;
- / процедуры поддержки;
- / метрики;
- / процедуры мониторинга.

б. операционной части приводят:

- / имя владелец сервиса;
- / профиль клиента;
- / зависимости от других сервисов;
- / модель Operations Level Agreement (OLA);
- / детальная информация о технической инфраструктуре, необходимой для обеспечения сервиса;
- / единицы инфраструктуры, рассматриваемые как активы;
- / план поддержания целостности, улучшения качества сервисов, развития возможностей;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

/ результаты аудита;



/ информация о ценах.

SLA позволяет установить формализованные критерии оценки результатов деятельности ИС-службы, установить единообразные и обязательные для всех участников процесса процедуры оценки результатов деятельности ИС-службы.

Сервисный подход к управления ИС-службой требует определенной зрелости как для самой ИС-службы, так и для бизнес-заказчиков. При этом следует учитывать ряд факторов:

- В требуется определенный уровень развития управления процессами и сервисами ИТ-службы предприятия, который предполагает, что процессы и ИТ-сервисы являются измеримы;
- В бизнес должен быть готов воспринимать некоторые "стандартные услуги" ИТ-службы как набор управляемых сервисов, выдвигать адекватные требования к уровню качества их предоставления, участвовать в повышении их качества;
- В обеспечение прозрачности ценообразования ИТ-сервисов, при которой ИТ-служба должна обосновывать формирование цены ИТ-сервиса и возможные пути её снижения;
- В наличие исключительных ситуаций, которые трудно предусмотреть заранее, процедуры выхода из них;
- В процессы, люди, взгляды подвержены изменениям. SLA, как и бизнес, должен адекватно изменяться при изменении внутренних и внешних факторов.

Следует отметить, что модель ITSM может применяться для предприятий с ИТ-службами различного размера: от 1 – 5 сотрудников до нескольких десятков сотрудников.

Для малых предприятий ролевой подход, принятый в ITSM, допускает совмещение одним и тем же сотрудником сколь угодно большого количества ролей в пределах его возможностей и компетенции. В предельном случае модель ITSM может использовать ИС-служба, состоящая из одного человека. Инструменталь-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ные программные средства, которые используются для управления ИТ-инфраструктурой, могут варьироваться в широких пределах: от офисных пакетов,



\endash простейшем случае, до специализированных инструментальных средств при большом размере ИС-службы.

Впервые термин Совокупная стоимость владения (Total Cost of Ownership – TCO) был введен Полем Страссманом, под TCO он понимал денежные затраты на обслуживание, модернизацию, ремонт, приобретение новых программных продуктов (ПП) для технического средства (например, компьютера), или поддержание в рабочем состоянии ПП (например, базы данных) за все предполагаемое или фактическое время его существования.

Наиболее простым определением TCO ИС является следующее: это затраты, связанные с *приобретением, внедрением и использованием* ИС. При этом необходимо рассматривать *первоначальные и последующие* затраты, в совокупности определяя их как *единые затраты* на информационную систему в процессе ее создания и эксплуатации.

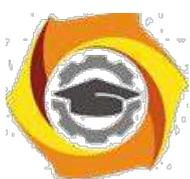
Любое предприятие с помощью автоматизации стремится повысить эффективность ведения своего бизнеса. Одно из главных условий достижения данной цели — «разумные» (т. е. не больше, но и не меньше) ИТ-затраты, которые точно так же, как и любые другие, требуют планирования, учета и контроля. Исходя из этого, для отечественных предприятий и ИТ-менеджеров неотъемлемыми и требующими детального рассмотрения являются вопросы, связанные с проблемой снижения совокупной стоимости владения информационной системой.

TCO первоначально разрабатывалась как средство расчета стоимости владения компьютером на Wintel-платформе и благодаря усилиям Gartner Group и Interpose эта методика стала основным инструментом для подсчета TCO и в других областях компьютерных технологий. Например, сейчас имеются методики расчета TCO документооборота, различных аппаратных платформ, сетей, ПО. Каждая из методик имеет свою специфику расчета, поэтому приведем лишь общую технологию расчета TCO.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Прежде всего мы попытаемся определить исходное число операторских мест и портов IVR (от англ. Interactive Voice Response – "интерактивная голосовая справка"), необходимое для того, чтобы начать работу. Для правильного расчета



нужно понять, какие задачи должен решать контакт-центр и в какой степени можно автоматизировать общение с абонентами, заменив операторов IVR.

Количество портов IVR не должно превышать число мест агентов более чем в два раза, так как в противном случае может образоваться огромная неуправляемая очередь, в которой абоненты будут долго ожидать ответа оператора.

21. в общем случае количество портов IVR не должно превышать число мест агентов более чем в два раза, так как в противном случае может образоваться огромная неуправляемая очередь, в которой абоненты будут долго ожидать ответа оператора. Исключением является случай, когда IVR используется в качестве автоматической информационно-справочной службы, в которой абонент может получить, например, информацию о состоянии своего лицевого счета, пополнить свой счет, послушать перечень и стоимость предоставляемых услуг и многое другое.

Предлагаемое пособие поможет качественно выполнить курсовую работу, целью которой является:

23. углубить знания теории и практики управления и администрирования информационных систем;

24. закрепить практические навыки написания программных кодов на языке высокого уровня Паскаль либо (по выбору студента) в среде визуального программирования Delphi, Lazarus или C++, на которых написаны большая часть программных кодов современных ОС.



Организационно-методические указания по выполнению курсовой работы

21. Курсовая работа оформляется с помощью текстового процессора MS Word на листах формата А4, шрифт 12, одинарный интервал, поля: сверху, снизу, справа – 15 мм, слева – 25 мм.

№ В записке должны быть приведены все использованные таблицы, схемы алгоритмов и иллюстрации.

№ Пояснительная записка должна иметь вид законченного документа и включать следующие обязательные пункты:

3.1. Титульный лист.

3.2. Лист утвержденного индивидуального задания на курсовую работу.

3.3. Содержание.

3.4. Теоретическая часть - обзорный – реферат на тему индивидуального задания согласно табл.1. по дисциплине «Управление и администрирование информационных систем». Вариант задания – номер студента по журналу группы. Студент должен самостоятельно выбрать литературу, использовать несколько источников, выделить основное и законспектировать. Не допускается дословное копирование текста из учебников и интернет – источников. Уровень оригинальности, полученный при проверке на сайте <https://www.antiplagiat.ru/> должен быть не менее 60%. Объем – 10-15 листов формата А4, включая рисунки, графики и таблицы. Список использованной литературы должен включать в себя не менее 10 источников не старше 5 лет, и быть оформлен в соответствии с ГОСТ 7.1 – 2003, .

3.5. Практическая часть - разработка компьютерного теста контроля знаний
33. формате GIFT по одному из разделов изучаемой дисциплины в соответствии с индивидуальным заданием согласно табл.1, а также рабочего программного модуля тестирующей программы. Тестирующая программа должна использовать в качестве исходных данных файл тестовых вопросов в формате GIFT, загружаемый



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

специальной директивой с использованием пароля доступа, позволяющую в режиме диалога проверить знания обучаемого.



3.6. Объем пояснительной записки к практической части – 15-20 печатных листов формата А4. Она должна включать следующие пункты:

3.6.1. постановка задачи и сравнительная характеристика существующих методов построения тестирующих программ;

3.6.2. обоснованный (исходя из возможностей выбранного средства программирования) выбор метода построения диалога;

3.6.3. исходные данные (тематика тестов, перечень проверяемых тем, вопросов и ответов);

3.6.4. разработка общего сценария тест-программы;

3.6.5. схемы алгоритмов функционирования программы;

3.6.6. текст программного модуля;

3.6.7. руководство программиста и руководство пользователя;

3.6.8. рекомендации по дальнейшему совершенствованию интерфейса и тест-программы в целом;

3.7. заключение (с необходимыми выводами);

3.8. список используемой литературы.

32. При защите курсовой работы студент демонстрируется функционирование разработанной программы путем проведения контрольного тестирования.

1.1 Особенности выполнения практической части курсовой работы

При разработке тестирующей программы студент самостоятельно выбирает уровень сложности разрабатываемой программы. Требования к программе, выполнение которых при условии успешной защиты курсовой работы позволит студенту получить оценку «Удовлетворительно», «Хорошо» или «Отлично» приводятся ниже.

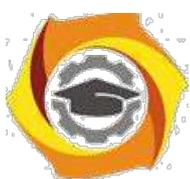
Минимальные требования для получения оценки «Удовлетворительно»

Используя возможности выбранного средства программирования разработать тестирующую программу, позволяющую в режиме диалога проверить знания обучаемого по одному из разделов изучаемых на кафедре дисциплин (в соответ-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ствии с вариантом индивидуального задания) с выставлением оценки. Формат файла вопросов – GIFT, кодировка UTF-8. Количество вопросов – не менее 40, из



которых (по случайному закону) для тестирования выбираются 10 вопросов. В состав вопросов должны входить вопросы следующих типов:

Множественный выбор – 15 шт.;

Краткий ответ – 10 шт.;

Верно-неверно – 5 шт.;

На соответствие – 5 шт.;

Числовой ответ – 5 шт..

28. в вопросах типа «Множественный выбор» правильный ответ выбирается из

4 – 5 вариантов ответов, при этом неправильные ответы не должны явно или радикально отличаться от правильных. После выбора ответа – автоматический переход к следующему вопросу без указания, правильно или нет ответил тестируемый. В конце на мониторе высвечивается оценка с комментарием, сколько процентов составили правильные ответы и на какие вопросы (номера) даны неверные ответы. Критерии выставления итоговой оценки:

17. Отлично – более чем на 85% вопросов получены правильные ответы;

18. Хорошо – более чем на 70% вопросов получены правильные ответы;

19. Удовлетворительно – более чем на 55% вопросов получены правильные

ответы;

20. Неудовлетворительно – более чем на 45% вопросов получены неправиль-

ные ответы.

Дополнительные требования для получения оценки «Хорошо»

Использовать текстовые и графические возможности выбранного средства программирования. Количество вопросов – не менее 50, из которых (по случайному закону) для тестирования выбираются 15 вопросов. В конце, помимо вышеуказанного, высвечиваются правильные ответы на те вопросы, по которым тестируемый дал неверный ответ, высвечивается оценка с комментарием, сколько процентов составили правильные ответы и на какие вопросы (номера) даны неверные ответы.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Дополнительные требования для получения оценки «Отлично»

Использовать текстовые, графические и аудио возможности выбранного средства программирования. Количество вопросов – не менее 60, из которых (по



случайному закону) для тестирования выбираются 20 вопросов. Ответ оценивается по дифференцированной 10-ти бальной шкале в зависимости от сложности вопроса, при неправильном ответе начисляется ноль баллов.

В конце высвечиваются:

максимально возможная сумма баллов;

набранная тестируемым сумма баллов и ее % от максимума;

правильные ответы на те вопросы, по которым тестируемый дал неверный ответ;

рекомендации по самостоятельному изучению материала с указанием литературных источников вплоть до страниц;

итоговая оценка за тест;

Описание формата GIFT

Формат GIFT - это наиболее подходящий формат для экспорта текстовых вопросов в текстовый файл. Он разработан для облегчения создания вопросов. GIFT поддерживает вопросы множественного выбора, верно/неверно, краткий ответ, вопросы на соответствие, численные вопросы и вопросы с пропущенными словами. Вопросы различных типов могут быть совмещены в одном файле, формат также поддерживает названия вопросов, комментарии к вариантам ответов, отзыв и процентное оценивание.

Правила создания файла в формате GIFT

ooo Любой файл GIFT должен быть закодирован в формате UTF-8

ooo В одном файле могут содержаться любые типы вопросов
(множественный выбор сопоставление и т.д)

ooo Между собой вопросы разделяются как минимум одной пустой строкой. Пустая строка определяет что начался новый вопрос.

Специальные символы ~ = # { } : контролируются фильтром и не могут быть использованы в тесте вопроса. Они участвуют в разделении частей вопроса,



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

12. называются "Символы управления." Но иногда вам приходится использовать эти символы в тексте вопроса, например, в математических формулах. Путь для решения таких проблем - "пропуск" символов управления. Он заключается в том,



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

23

что перед символом управления необходимо поставить обратный слеш "\". При обработке вопроса обратный слеш удаляется и не отображается.

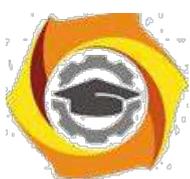
Таблица специальных символов формата GIFT

Symbols	Use
// text	Комментарий в конце строки (пометка на память для себя, нигде не появится, опционально)
::title::	Заголовок вопроса (опционально)
text	Текст вопроса (станет названием вопроса, если название не указано)
[...format...]	Формат куска текста. Варианты [HTML-код], [обычный] и [уценка]..
{	Фигурная скобка указывает что начался блок с ответами
}	Конец блока ответов
{T} or {F}	True или false ответа; используется в типе вопроса верно/неверно
{ ... =right ... }	Правильный ответ на множественный выбор внутри блока вариантов ответа (=правильный ответ)
{ ... ~wrong ... }	Правильный ответ на множественный выбор внутри блока вариантов ответа (~неправильный ответ)
{ ... =item -> match ... }	Ответ на вопросы соответствия
#feedback text	Ответ обратной связи на соответствие, с заполнением пустых полей или числовые ответы
####general feedback	Общие отзывы
{#	Начало цифрового ответа
answer:tolerance	Числовой ответ принимается в пределах \pm допустимых пределах
low..high	Нижнее и верхнее значения диапазона принимаемых числовых ответов
	N процент кредита на один из нескольких числовых



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

\pm диапазонов в пределах допуска от ответа



<code>\character</code>	Обратный слеш отменяет действие символов ~, =, #, {, }, and :
<code>\n</code>	Помещает строку в текст вопроса — символ заменяет пустую строку между вопросами

Описание типов вопросов

Множественный выбор

Описание: Студент выбирает ответ на вопрос из нескольких предложенных ему вариантов, причём вопросы могут предполагать один или сразу несколько правильных ответов.

Для вопросов типа "Множественный выбор" неправильные варианты ответов начинаются со знака тильды(~), правильные - знаком равенства (=).

Для удобства варианты ответов могут быть написаны каждый с новой строки.

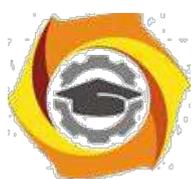
ПРИМЕР

```
::Вопрос 1::Кто является главным героем сказки «Айболит»?  
{  
~Синяя Борода  
~Джефферсон  
=Айболит  
~Конёк-Горбунок  
}
```

Можно использовать формат пропущенного слова, для этого в текст вопроса автоматически должна вставляться линия пропущенного слова (____), а варианты ответов показываются ниже

ПРИМЕР

```
::Вопрос 1::Ленин {~похоронен =родился ~живёт} в Симбирске
```



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**



Для множественного выбора могут заданы веса после тильды (~). Значение веса с обеих сторон заключается в знак % (например, %50%). Эта опция может быть скомбинирована с комментариями на варианты ответов.

ПРИМЕР

::Вопрос 1::Какие цвета используются в светофоре?

{
~синий
~%50%красный
~%50%желтый
~фиолетовый
~белый
}

Краткий ответ

Ответом на вопрос является слово или короткая фраза, допускается несколько правильных ответов с различными оценками. Ответы оцениваются путем сравнения с разными образцами ответов, в которых могут использоваться подстановочные знаки.

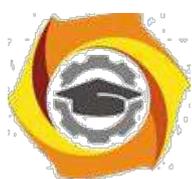
Ответы в вопросе "Краткий ответ" начинаются знаком равенства(=), показывающим правильный ответ. Ответы не должны содержать тильду.

Если в тексте вопроса использовать комбинацию из _ (нижней линии), то поле ввода ответа появится сразу в тексте. Длина поля зависит от количества _

ПРИМЕР

::Вопрос 1::Кто является героем сказки «Айболит»?{

=Айболит
=Бармалей
}



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**



::Вопрос 1:: Ленин _____ в Симбирске {
=родился
}

Верно/неверно

При ответе на вопрос, студент выбирает между двумя вариантами «Верно» и «Неверно».

Ответ должен быть написан как {TRUE} или {FALSE}, или сокращённо {T} или {F}.

ПРИМЕР

::Вопрос 1:: Ленин родился в Симбирске {TRUE}

На соответствие

Каждому элементу ответов первой группы нужно сопоставить элемент ответов второй группы.

Вопросы на соответствие не поддерживают процентное оценивание.

Совпадающие пары начинаются знаком (=) и разделяются знаком "->".

Должны быть как минимум три совпадающие пары.

::Название вопроса:: Укажите ... {

=подвопрос1 -> подответ1

=подвопрос2 -> подответ2

=подвопрос3 -> подответ3

ПРИМЕР

::Вопрос 1:: Укажите столицы государств: {

=Канада -> Оттава

=Италия -> Рим

=Япония -> Токио

=Индия -> Нью Дели



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

}



Числовой ответ

Так же, что и краткий ответ, только на выполнение вычислительных операций, числовой ответ может иметь заданный интервал предельно допустимой погрешности отклонения от правильного значения.

Секция ответа в числовом вопросе должна начинаться с решетки (#). Числовой ответ может включать погрешность, которая пишется после правильного ответа и отделяется двоеточием. Например, если правильный ответ находится в диапазоне от 1.5 до 2.5, тогда вопрос должен быть написан так: {#2:0.5}. Эта запись показывает что 2 с допуском 0.5 - правильный ответ (т.е. диапазон от 1.5 до 2.5). Если погрешность не определена, то по умолчанию она устанавливается в ноль.

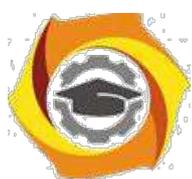
ПРИМЕР

::Вопрос 1:: В каком году родился Пушкин А.С.?

{#1799}

::Вопрос 2::Значение числа Пи (4 цифры после запятой)?

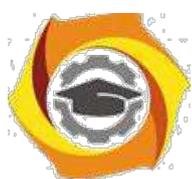
{#3.1415:0.0005}



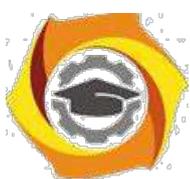
2 Задания на курсовую работу

Таблица 1. Индивидуальные задания для курсовой работы

Вариант №	Содержание задания №1 курсовой работы
1	CASE-технологий разработки информационной системы
2	Фазы развития информационных систем..
3	Концептуальная фаза жизненного цикла информационной системы
4	Подготовка технического предложения.
5	Проектирование информационной системы.
6	Разработка технического задания информационной системы.
7	Ввод системы в эксплуатацию.
8	Основные процессы жизненного цикла: разработка, эксплуатация, сопровождение.
9	Вспомогательные процессы жизненного цикла.
10	Организационные процессы.
11	Модели жизненного цикла информационной системы.
12	Структура жизненного цикла информационной системы.
13	Каскадная модель жизненного цикла системы: основные этапы разработки
14	Сервис ИТ в деятельности службы ИС.
15	Функциональные области управления службой ИС.
16	Организационная структура службы ИС. Плоская структура службы ИС
17	Организационная структура службы ИС. Развернутая структура службы ИС.
18	Организационная структура службы ИС. Дивизиональная структура службы ИС.
19	Функции службы ИС и параметры сервиса ИТ.
20	Процессы службы ИС и преодоление ограничений функционального подхода.
21	ITIL/ITSM – концептуальная основа процессов службы ИС. Проект ITIL
22	Модели управления сетевыми ресурсами
23	Служба Active Directory Service
24	Системы автоматизации учета заявок и поддержки пользователей информационных систем
25	Администрирование пользователей. Политики безопасностей, их реализация в операционных системах
26	Методы анализа бизнес-плана проекта по разработке информационной системы в среде Project Expert
27	Классификация и сравнительный анализ программных средств оценки и учета затрат на разработку программного обеспечения



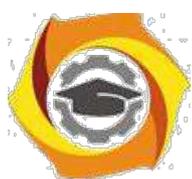
**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

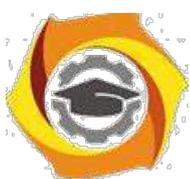
29

29	Обеспечение целостности данных. Резервное копирование и восстановление данных. Стратегии резервного копирования
30	Оценка совокупной стоимостью владения и ее применение для управления информационной службой
Вариант №	Содержание задания №2 курсовой работы (тематика разрабатываемого теста)
22.	Операционная система Windows. Интерфейс пользователя. Управление Windows. Рабочий стол, окна Windows, файловая система, специальные папки Рабочего стола.
23.	Операционная система Windows. Панель задач. Диалоговые окна. Контекстное меню. Справочная система. Корзина. Подготовка к выключению компьютера. Действия при «зависании» компьютера.
24.	Операционная система Windows. Проводник Windows. Основные операции с объектами. Запуск программ. Создание ярлыка. Изменение значка ярлыка.
25.	Операционная система Windows. Форматирование дисков. Настройка экрана. Панель управления Настройка панели задач. Поиск файлов.
26.	Текстовый процессор MSWord. Работа с окнами. Курсор ввода. Панели инструментов. Координатные линейки. Строка состояния. Режимы отображения документа. Полосы прокрутки. Выход из Microsoft Word.
27.	Текстовый процессор MSWord. Создание нового документа. Открытие документа. Сохранение документа. Закрытие документа. Ввод текста. Выделение фрагмента текста.
28.	Текстовый процессор MSWord. Редактирование текста. Отмена операций над текстом. Копирование текста. Перемещение текста. Вставка символа. Поиск и замена текста. Контекстное меню.
29.	Текстовый процессор MSWord. Форматирование текста. Изменение параметров шрифта. Изменение интервала и положения символов. Форматирование абзацев. Установление позиций табуляции.
30.	Текстовый процессор MSWord. Упорядочение списков. Стили форматирования. Оформление страниц документа. Установление параметров страницы. Вставка разрывов страниц. Нумерация страниц.
24.	Текстовый процессор MSWord. Установление колонтитулов. Создание многоколоночного документа. Печать документов.
25.	Текстовый процессор MSWord. Структура таблицы Word произвольной конфигурации. Способы создания таблиц. Форматирование таблицы. Использование в таблице формул. Изменение структуры таблицы.
26.	Создание составных (интегрированных) документов. Понятие составного документа. Создание составного документа через буфер обмена по технологии OLE. Внедрение и связывание объектов.
к	Текстовый процессор MSWord. Вставка графических объектов (рисунков, файлов). Использование автофигур, объектов WordArt. Группирование объектов и настройка изображений.
к	Табличный процессор MS Excel. Основные понятия интерфейс табличного процессора. Данные, хранимые в ячейках электронной таблицы. Относительные и абсолютные ссылки.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

к Табличный процессор MS Excel. Функциональные возможности табличных процес-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

30

	соров. Типовые операции с рабочими книгами и листами. Типовые операции с бло-ками ячеек.
15.	Табличный процессор MS Excel. Сортировка и фильтрация данных. Понятие о спис-ке (базе данных Excel). Сортировка данных в списке. Автофильтрация и расширен-ный фильтр.
16.	Табличный процессор MS Excel. Структурирование таблиц и создание сводных таб-лиц. Автоструктурирование таблиц. Создание сводных таблиц с помощь. Мастера сводных таблиц.
17.	Табличный процессор MS Excel. Технология работы в электронной таблице. Проек-тирование электронной таблицы. Объединение электронных таблиц. Макросы как средство автоматизации работы.
18.	Табличный процессор MS Excel. создание и заполнение таблицы постоянными дан-ными и формулами Характеристика режимов и команд. Графические возможности MS Excel.
19.	Табличный процессор MS Excel. Построение, редактирование и форматирование диаграмм. Построение тренда. Операции с листами рабочих книг.
20.	Табличный процессор MS Excel. Организация расчётов в электронной таблице. Ра-бота с формулами. Функции Мастер функций. Массивы формул.
21.	Табличный процессор MS Excel. Операции с элементами таблицы. Вставка и удале-ние элементов таблицы. Копирование и перемещение данных. Поиск данных. Замена данных.
22.	Табличный процессор MS Excel. Формат чисел. Маски форматов. Выравнивание со-держимого ячеек. Установление шрифта. Изменение размеров строк и столбцов. Оформление таблиц.
23.	Табличный процессор MS Excel. Группирование элементов таблицы. Работа с окна-ми. Разделение окон. Создание нового окна. Фиксация подокон. Присвоение имени ячейке, диапазону или формуле. Создание примечаний.
24.	Табличный процессор MS Excel. Работа с базами данных. Сортировка данных. Фор-мы данных. Фильтрация (выборка) данных. Установление диапазона критериев. Ав-то-фильтр. Расширенный фильтр.
25.	Язык программирования Pascal. Алфавит и базовые типы данных. Структура про-граммы. Стандартные функции. Запись аналитических зависимостей. Константы и переменные. Ввод-вывод данных.
26.	Основные понятия алгоритмического языка. Оператор перехода. Условный опера-тор. Оператор варианта. Элементы структурного программирования. Язык программирования Pascal. Цикл с предусловием, цикл с постусловием, цикл с



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

27. параметром. Операторы завершения цикла. Массивы, строки. Процедуры и функции.
Язык программирования Pascal. Перечисляемый тип данных. Интервальный тип
28. данных. Модули. Множества, Записи. Файлы: текстовые файлы, компонентные фай-лы, бестиповые файлы.
29. Язык программирования Pascal. Указатели. Динамические переменные. Динамиче-ские структуры данных. Очереди. Линейные списки.

Методические указания
по организации и проведению учебной практики

1 Общие положения

Учебная практика студентов является обязательной составной частью процесса обучения и представляет собой учебные занятия в виде самостоятельной работы. Учебная практика проводится со студентами очного и заочного факультетов.

Способ проведения учебной практики – стационарно, форма проведения учебной практики – дискретно. Место проведения практики – компьютерные лаборатории кафедры ИВТ ТУ УГМК.

Практика проводятся в соответствии с учебным планом и графиком учебного процесса.

Цель учебной практики

Целью учебной практики является получение первичных профессиональных умений и навыков студентами, в том числе, первичных умений и навыков научно-исследовательской деятельности.

Учебная практика направлена на формирование у обучающихся первичных профессиональных умений и навыков решать следующие профессиональные задачи в *проектном* виде профессиональной деятельности:

- способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности (ОПК-2);
- способен осваивать методики использования программных средств для решения практических задач (ОПК-9).

Задачи практики

Задачами учебной практики являются:

- закрепление теоретических знаний, полученных в вузе в процессе начального периода обучения;



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- приобретение опыта общения и взаимодействия в коллективе в процессе совместной деятельности, умения учитывать позиции других участников деятельности;
- приобретение навыков сбора, обработки, анализа и систематизации научно-технической информации по теме исследования при выполнении индивидуального задания;
- изучение современной аппаратуры, программных продуктов и методов исследования;
- участие в проведении экспериментальных исследований для решения научно-исследовательских и учебных задач.

Обязанности руководителя практики

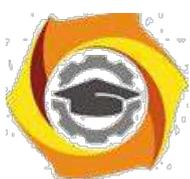
Руководитель учебной практики назначается приказом по ТУ. В его обязанности входит:

- проведение со студентами организационных мероприятий перед началом практики (инструктаж о порядке прохождения практики, выдача индивидуальных заданий, разъяснение требований к отчетности по практике, инструктаж по технике безопасности);
- осуществление контроля за обеспечением удовлетворительных условий самостоятельной работы студентов;
- осуществление контроля за работой студентов в период практики, просмотр дневников студентов;
- просмотр материалов студентов для отчета по практике и оказание помощи в их подготовке;
- проверка дневников и отчетов на заключительном этапе учебной практики;
- прием зачета по результатам практики студентов;
- оформление письменного отчета о прошедшей практике с замечаниями и предложениями по её совершенствованию.

Обязанности студента во время практики

Студент при прохождении учебной практики обязан:

- подчиняться правилам внутреннего трудового распорядка, действующим в ТУ, выполнять распоряжения преподавателя - руководителя практики;
- изучить и строго соблюдать правила охраны труда и техники безопасности, правила пожарной безопасности, производственной санитарии и охраны окружающей среды;
- полностью и своевременно выполнять все задания, предусмотренные программой практики;
- нести ответственность за выполненную работу и ее результаты;
- работать над выполнением индивидуального задания;



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- вести дневник по практике с ежедневной записью всех видов работы и последующей подписью руководителя об их выполнении;
- по окончании практики написать отчет с соблюдением требований и правил оформления, изложенных в разделе 3;
- своевременно представить руководителю практики дневник и письменный отчет о выполнении всех заданий, защитить его и получить оценку по практике и отзыв в дневнике (на пояснительной записке также должны стоять оценка и роспись руководителя).

Для оформления отчета студенту выделяется в конце практики 3-4 дня (18- 24 часов);

2 Подготовка к практике

Перед началом учебной практики с группой проводится собрание, на котором студентам доводятся сроки практики, последовательность её прохождения и порядок отчётности по ней. Проводит собрание руководитель практики, на собрании присутствуют декан факультета и заведующий кафедрой ИВТ.

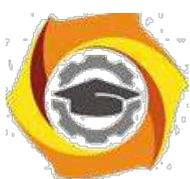
На собрании выдаются дневники практики (приложение А), и производится оформление первой части дневника. В частности заполняются:

- титульный лист;
- направление на практику;
- план практики;
- индивидуальное задание.

После оформления дневника в разделе направление на практику ставится подпись декана.

Типовой план проведения практики должен содержать следующие пункты:

- 1 Прохождение инструктажа по правилам и мерам безопасности.
- 2 Изучение требований безопасности на рабочем месте.
- 3 Проведение поиска и изучение нормативной и правовой документации в области инфокоммуникаций в соответствии со списком рекомендованных источников.
- 4 Изучение назначения, состава, характеристик и правил эксплуатации компьютерного оборудования, в объеме требований программы учебной практики.
- 5 Выполнение индивидуального задания.
- 6 Оформление дневника и отчёта по учебной практике.
7. Предоставление дневника и отчёта о прохождении учебной практики руководителю, их защита.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Все пункты плана практики устанавливаются руководителем практики. Данные пункты являются базовыми, но они могут быть дополнены или изменены исходя из индивидуальных особенностей студента или в целях научно-исследовательской работы в ТУ.

В индивидуальное задание выносятся вопросы, которые должны быть отражены в отчете по практике.

Типовой вариант индивидуального задания для прохождения учебной практики:

1 Пройти инструктажа по правилам и мерам безопасности.

2 Изучить требования безопасности на рабочем месте.

3 Изучить состав и устройство компьютера:

- типы компьютеров и их состав;
- особенности и сравнение устройств;
- устройства, предложенные для сборки компьютера; - проверка работоспособности компьютера.

4 Изучить одну из предложенных ниже операционных систем (ОС) (по указанию руководителя практики):

4.1 ОС Windows:

- понятие ОС, основные функции, назначение;
- характеристики и основные возможности ОС Windows;
- архитектура ОС;
- состав стандартных программ и их краткая характеристика; - панель управления, возможности командной строки.

4.2 ОС Linux:

- теоретические основы ОС Linux;
- обзор версий и скачивание дистрибутивов;
- подготовка загрузочного флеш-накопителя и установка OS Linux;
- работа с файловым менеджером, изучение сетевых настроек Network

Manager;

- разграничение прав, работа в терминале, изучение его возможностей; - установка и удаление программ, обновление системы.

5 Изучить одну из предложенных ниже тем (по указанию руководителя практики):

5.1 BIOS и виртуальные машины:

- виды BIOS, основные настройки;
- характеристика виртуальных машин;
- установка Oracle VM VirtualBox;
- особенности установки ОС Windows на виртуальную машину.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

5.2 Периферийные устройства компьютера:

- состав, назначение и характеристики периферийных устройств компьютера;
- подключение периферийных устройств, установка драйверов; - проверка работоспособности.

5.3 Локальные вычислительные сети (ЛВС):

- виды, основные характеристики ЛВС, состав сетевого оборудования;
- разворачивание проводной ЛВС с помощью предложенных

устройств;

- настройка и проверка работоспособности сети в ОС Windows.

5.4 Локальные вычислительные сети (ЛВС):

- виды, основные характеристики ЛВС, состав сетевого оборудования;
- разворачивание беспроводной ЛВС с помощью предложенных

устройств;

- настройка и проверка работоспособности сети в ОС Linux.

8. Оформить дневник и отчет по производственной практике.

9. Отчитаться о прохождении учебной практики перед руководителем практики и получить от него отзыв о выполненной работе и оценку.

Данные пункты могут быть дополнены или изменены в целях научноисследовательской работы в ТУ.

3 Прохождение практики

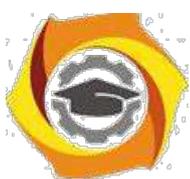
При прохождении практики студент должен выполнить все намеченные планом практики мероприятия и полностью выполнить индивидуальное задание на практику.

В процессе практики студент заполняет в дневнике раздел «ДНЕВНИК» и по мере накопления учебного материала пишет отчет по практике. В окончательном виде отчет должен соответствовать нижеизложенным требованиям.

Требования к содержанию и оформлению отчета по практике Рекомендуемый объем отчета 25-35 листов машинописного текста.

Отчет в обязательном порядке должна содержать следующие элементы:

- титульный лист (приложение Б);
- задание;
- содержание;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

- основная часть (разделы, подразделы);
- список использованных источников, в т. ч. нормативных, проектных и справочных материалов.

При необходимости отчет также может содержать:

- перечень сокращений, символов и специальных терминов с их определениями;
- приложения.

Оформление текста

Отчет по практике оформляется в соответствии с методическими указаниями по учебной практике, ГОСТ 2.105-95 и Положением о практиках студентов ТУ УГМК.

Текст отчета печатается на одной стороне листа белой бумаги формата

A4, при этом размеры полей: правое - 1,5 см, верхнее и нижнее - 2 см, левое - 3 см. Шрифт «TimesnewRoman», размер шрифта - 14, печать через 1,5 интервала. Абзацный отступ – 1,25 см. Цвет шрифта - черный. Заголовки

«СОДЕРЖАНИЕ», «СПИСОК ИСТОЧНИКОВ» и «СПИСОК СОКРАЩЕНИЙ» размещаются по центру строки и набираются прописными буквами, а названия «Разделов» и «Подразделов» размещаются как обычный текст с абзацного отступа и набираются строчными буквами с первой прописной. Допускается название раздела набирать шрифтом размера - 16. Разделы и подразделы нумеруются арабскими цифрами, точка после номера раздела (подраздела) и после наименования раздела не ставится.

1 Наименование раздела

1.1 Наименование подраздела

.....

1.2 Наименование подраздела

.....

2 Наименование раздела

2.1 Наименование подраздела

.....

2.2 Наименование подраздела

.....

Каждый раздел начинается с новой страницы. Названия разделов, подразделов должны соответствовать содержанию и быть оформлены единообразно во всем документе.

Страницы работы нумеруются арабскими цифрами (нумерация сквозная по всему тексту). Номер страницы проставляют на нижней части листа по центру или справа страницы. Титульный лист включается в общую нумерацию, при этом номер на нем не ставится.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Иллюстрации и таблицы, расположенные на отдельных листах, включают в общую нумерацию страниц отчета по практике.

Оформление формул

В формулах в качестве символов следует применять обозначения, установленные соответствующими государственными стандартами. Желательно для набора формул использовать встроенный в текстовый редактор «Word» редактор формул «Equations» или отдельную программу, встраиваемую в текстовый редактор, например, «MathType». В формулах буквы русского и греческого алфавитов и цифры пишутся прямым шрифтом, а буквы латинского алфавита – курсивом. Размер основного шрифта в формулах должен быть, как и во всем тексте пояснительной записки, равен 14. Пояснения символов и числовых коэффициентов в формулах, если они не пояснены ранее, должны быть приведены непосредственно под формулой. Пояснение каждого символа дается с новой строки в той последовательности, в которой символы приведены в формуле. Первая строка пояснения начинается со слова "где" без двоеточия после него.

Пример 1. Сила тока по закону Ома, вычисляются по формуле:

$$I = \frac{U}{R}, \text{ А,} \quad (1)$$

где U – напряжение, В,

R – сопротивление, Ом.

Формулы нумеруются сквозной нумерацией арабскими цифрами, которые записываются на уровне формулы справа в конце строки в круглых скобках.

Переносить формулы на следующую строку допускается только на знаках выполняемых операций, причем знак в начале следующей строки повторяют. При переносе формулы на знаке умножения применяют знак умножения в виде "х".

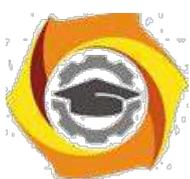
Ссылки в тексте на формулы дают по их порядковым номерам, указываемым в скобках, например, в формуле (1).

Оформление иллюстраций

Все иллюстрации (чертежи, графики, диаграммы, фотографии), включаемые в текстовый материал записки, именуется рисунками.

Иллюстрации нумеруются арабскими цифрами сквозной нумерацией.

Иллюстрации могут иметь наименование и пояснительные данные (подрисуночный текст). Точка после подрисуночного текста не ставится. Слово «Рисунок» и наименование помещают после рисунка и пояснительных данных (если они есть), выравнивают по центру и располагают, как показано на рисунке 1.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**



1 - 24-х контактный модуль питания, который подключается непосредственно к материнской плате и подает по разным линиям проводов на нее напряжения, в 5, 3 и 12 вольт, необходимые для функционирования различных устройств компьютера.

2 - 4-х контактный разъем питания на 12 вольт, обеспечивающий питание центрального процессора

3 - питание «SATA»

4 - питание стандарта «molex»

5 - разъем питания для дисководов

Рисунок 1 - Шлейфы и кабели, используемые для соединения
компонентов системного блока

При ссылках на иллюстрации следует писать: «...как показано на рисунке1».

Допускается размещение рисунка в альбомном расположении страницы. В этом случае подрисовочная надпись должна быть ориентирована таким образом, чтобы читалась при повороте открытого отчета по часовой стрелке на 90 градусов.

Оформление таблиц

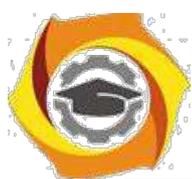
Таблицу, в зависимости от ее размера, помещают под текстом, в котором впервые дана ссылка на нее, или на следующей странице, а, при необходимости, в приложении к документу.

Название таблицы должно отражать её содержание, быть точным, кратким.

Название помещают над таблицей (выравнивание по левому краю), начинают с заглавной буквы, не подчеркивают. Таблицы необходимо нумеровать арабскими цифрами сквозной нумерацией. Высота строк таблицы должна быть не менее 8 мм. Ниже приведен пример оформления таблицы.

Таблица 1 – Параметры диэлектриков

Наименование материала диэлектрика	$E_{пр} \cdot 10^6, В/см$	ϵ
1. Моноксид кремния	3	2



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

2. Моноокись германия	1	3
3. Боросиликатное стекло	4	0,4
4. Стекло электровакуумное	3	1,5

В последний день практики дневник и отчет сдаются руководителю практики.

4. Отчетность по практике

Зачёт по учебной практике проводится в назначенное деканатом время.

К зачёту допускаются только те студенты, которые представили правильно оформленные дневник и отчёт.

Дневник по учебной практике считается оформленным правильно если:

- все страницы дневника заполнены и содержат информацию в достаточной степени раскрывающую рассматриваемые вопросы;
- записи выполнены читаемым почерком, без серьёзных ошибок, допускаются незначительные поправки.

Отчёт по учебной практике считается оформленным правильно если:

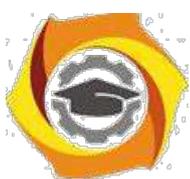
- содержание отчёта позволяет судить о выполненных студентом действиях при проведении практики;
- отчёт содержит анализ правовых и научно-технических источников в соответствии с индивидуальным заданием;
- в отчете приведены развернутые ответы по всем пунктам индивидуального задания;
- оформление отчета соответствует установленным требованиям.

Оценка за учебную практику определяется по уровню усвоения студентом всех компетенций. Определение знаний, умений и навыков производится путём получения ответов на задаваемые контрольные вопросы в рамках рабочей программы по практике.

Студенты, получившие отрицательный отзыв о прохождении практики или не выполнившие программу практики, направляются на практику повторно в период каникул, а в случае серьезных нарушений могут быть отчислены из ТУ.

Студенты, не явившиеся на зачет или не защитившие отчеты по практике, ликвидируют задолженность в сроки, установленные деканатом.

Задолженность по практике является академической задолженностью.



5 Рекомендованные источники

- 1 Федеральный закон от 07.07.2003 N 126-ФЗ (ред. от 13.07.2015) "О связи" (с изм. и доп., вступ. в силу с 10.01.2016).
- 2 Федеральный закон от 27 июля 2006 г. N 149-ФЗ "Об информации, информационных технологиях и о защите информации" (с изм. и доп., вступ. в силу с 10.01.2016).
- 3 ГОСТ 34.936-91 Информационная технология. Локальные вычислительные сети. Определение услуг уровня управления доступом к среде.
- 4 ГОСТ Р 53724-2009 Качество услуг связи. Общие положения.
- 5 Симонович С.В. Информатика. Базовый курс. Учебник для вузов.- СПб.: Питер, 2015.
- 6 Гаврилов М.В., Климов В.А. Информатика и информационные технологии.- М.: Юрайт, 2014.
- 7 Мартемьянов Ю.Ф. Операционные системы. Концепции построения и обеспечения безопасности: учебное пособие. - М.: Горячая линия - Телеком, 2011.
- 8 Баринов В.В. Технологии разработки и создания компьютерных сетей на базе аппаратуры D-LINK: учебное пособие для вузов. - М.: Горячая линия - Телеком, 2013.
- 9 Проскурин В.Г. Защита в операционных системах: учебное пособие для вузов. - М.: Горячая линия - Телеком, 2014.
- 10 Девянин П.Н. Модели безопасности компьютерных систем. Управление доступом и информационными потоками: учебное пособие для вузов. - М.: Горячая линия - Телеком, 2014.
- 11 Львов В.Л. Методические указания по организации и проведению учебной практики. - РнД: ТУ УГМК, 2017.

Методические указания

к лабораторным работам

по дисциплине

«Методы и средства защиты компьютерной информации»

Лабораторная работа 1. Исследование свойств межсетевого экрана с пакетной фильтрацией

1.1 Цель работы: Получение навыков конфигурирования межсетевого экрана с пакетной фильтрацией и исследование его свойств.

1.2 Перечень оборудования:

Маршрутизаторы Cisco;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

ПК с установленным ПО Cisco Packet Tracer;
Локальная сеть.

1.3 Задание:

В Cisco Packet Tracer создать сеть, состоящую из пяти подсетей. В одной из подсетей установить два сервера, один из которых должен быть сконфигурирован как FTP, а другой – как WEB-сервер.

Всем компьютерам подсети 192.169.X.0 (где X – номер студента в списке группы) предоставить полный доступ ко всем серверам.

Всем компьютерам подсети 192.169.X+1.0 предоставить доступ только к FTP-серверу по протоколу FTP.

Всем компьютерам подсети 192.169.X+2.0 предоставить доступ только к WEB-серверу.

Компьютерам оставшейся подсети запретить доступ к внешним ресурсам.

1.4 Указания к проведению работы.

Как известно [7], разделяют межсетевые экраны (FireWall) с пакетной фильтрацией и с сохранением состояний (экраны прикладного уровня). Рассмотрим сначала использование пакетной фильтрации.

Межсетевые экраны с фильтрацией пакетов представляют собой маршрутизаторы (например, Cisco) или работающие на сервере программы, сконфигурированные таким образом, чтобы фильтровать входящие и исходящие пакеты. Поэтому такие экраны называют иногда пакетными



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

фильтрами. Фильтрация осуществляется путем анализа IP-адреса источника в получателя, а также портов входящих TCP- и UDP-сегментов и сравнением их с сконфигурированной таблицей правил. Эти межсетевые экраны просты в использовании, дешевы, оказывают минимальное влияние на производительность вычислительной системы. Основным недостатком является их уязвимость при подмене адресов IP. Во всей линейке оборудования Cisco Systems пакетная фильтрация реализована с помощью так называемых списков контроля доступа (Access Control List).

Списки доступа ACL могут быть созданы для всех сетевых протоколов, функционирующих на маршрутизаторе, например IP или IPX, и устанавливаются на интерфейсах маршрутизаторов. Запрет или разрешение сетевого трафика через интерфейс маршрутизатора реализуется на основании анализа совпадения определенных условий. Для этого списки доступа представляются в виде последовательных записей, в которых используют адреса и протоколы. Сетевые фильтры (списки доступа) создаются для входящих или исходящих пакетов на основании анализируемых параметров (адреса источника, адреса назначения, протокола и номера порта верхнего уровня), указанных в списке доступа ACL (рисунок 1.1).

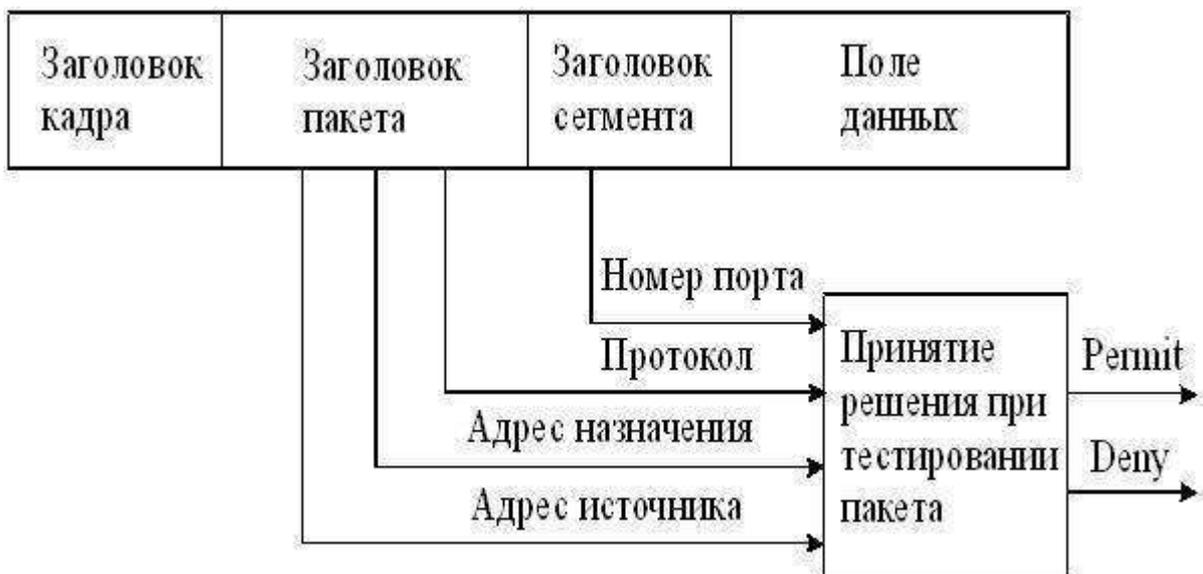


Рисунок 1.1 – Анализ заголовков пакета



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Как видно из рисунка 1.1, на основе проведенного анализа служебной информации, устройство, реализующее межсетевое экранирование, принимает решение о дальнейшей передаче (permit) или о фильтрации (deny).

Списки доступа могут быть определены для каждого установленного на интерфейсе протокола и для каждого направления сетевого трафика (исходящего и входящего). Поэтому для входящего и исходящего трафиков через интерфейс создаются отдельные списки.

Если списки доступа не формируются на маршрутизаторе, то все проходящие через маршрутизатор пакеты будут иметь доступ к сети.

Список доступа ACL составляется из утверждений (условий), которые определяют, следует ли пакеты принимать или отклонять во входных и выходных интерфейсах маршрутизатора. Программное обеспечение IOS Cisco проверяет пакет последовательно по каждому условию. Если условие, разрешающее продвижение пакета, расположено наверху списка, никакие условия, добавленные ниже его, не будут запрещать продвижение пакета. Если в списке доступа необходимы дополнительные условия, то список целиком должен быть удален и создан новый с новыми условиями.

Функционирование маршрутизатора по проверке соответствия принятого пакета требованиям списка доступа производится следующим образом. Когда кадр поступает на интерфейс, маршрутизатор проверяет IP-адрес. Если адрес назначения соответствует адресу интерфейса, то маршрутизатор извлекает (декапсулирует) из кадра пакет и проверяет его на соответствие условиям списка ACL входного интерфейса. При отсутствии запрета или отсутствии списка доступа пакет инкапсулируется в новый кадр второго уровня и отправляется интерфейсу следующего устройства.

Проверка условий (утверждений) списка доступа производится последовательно. Если текущее утверждение верно, пакет обрабатывается в соответствии с командами **permit** или **deny** списка доступа, остальная часть условий ACL не проверяется. Если все утверждения ACL неверны, то неявно



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

заданная по умолчанию команда **deny any** (запретить все остальное) в конце списка не позволит передавать дальше по сети несоответствующие пакеты.

Существуют разные типы списков доступа: стандартные (standard ACLs), расширенные (extended ACLs) и именованные (named ACLs). Когда список доступа конфигурируется на маршрутизаторе, каждый список должен иметь уникальный идентификационный номер или уникальное имя. Номер идентифицирует тип созданного списка доступа и должен находиться в пределах определенного диапазона, заданного для этого типа списка (таблица 1.1).

Таблица 1.1 – Диапазоны идентификационных номеров ACL

Диапазон номеров	Название списка доступа
1-99	IP standard access-list
100-199	IP extended access-list
1300-1999	IP standard access-list (extended range)
2000-2699	IP extended access-list (extended range)
600-699	Appletalk access-list
800-899	IPX standard access-list
900-999	IPX extended access-list

12 стандартном списке доступа для принятия решения в IP-пакете анализируется только адрес источника сообщения, чтобы фильтровать сеть (IPX-стандарт может фильтровать как адрес источника, так и назначения).

Расширенные списки доступа (Extended Access Lists) проверяют как IP-адрес источника, так и IP-адрес назначения, поле протокола в заголовке пакета сетевого уровня и номер порта в заголовке транспортного уровня.

Таким образом, для каждого протокола, для каждого направления трафика и для каждого интерфейса может быть создан свой список доступа.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Исходящие фильтры не затрагивают трафик, который идет из местного маршрутизатора.

Из рекомендаций по установке списков доступа можно отметить следующее. Стандартные списки доступа рекомендуется устанавливать по возможности ближе к адресату назначения, а расширенные – ближе к источнику. Поэтому стандартные списки доступа должны блокировать устройство назначения и располагаться поближе к защищаемой сети, а расширенные списки доступа должны быть установлены близко к источнику сообщений.

Список доступа производит фильтрацию пакетов по порядку, поэтому в строках списков следует задавать условия фильтрации, начиная от специфических условий и заканчивая общими. Условия списка доступа обрабатываются последовательно от вершины списка к основанию, пока не будет найдено соответствующее условие. Если никакое условие не найдено, тогда пакет отклоняется и уничтожается, поскольку неявное условие **deny any** (запретить все остальное) присутствует неявно в конце любого списка доступа. Не удовлетворяющий списку доступа пакет протокола IP будет отклонен и уничтожен, при этом отправителю будет послано сообщение ICMP. Новые записи (линии) всегда добавляются в конце списка доступа.

Конфигурирование списков доступа производится в два этапа:

18. Создание списка доступа в режиме глобального конфигурирования.

19. Привязка списка доступа к интерфейсу в режиме детального конфигурирования интерфейса.

Формат команды создания стандартного списка доступа следующий:

Router(config)#access-list {№} {permit / deny} {адрес источника}.

Списки доступа могут фильтровать как трафик, входящий в маршрутизатор (in), так и трафик, исходящий из маршрутизатора (out).

Направление трафика указывается при привязке списка доступа к интерфейсу.

Формат команды привязки списка к интерфейсу следующий:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Router(config-if)#{протокол} access-group {номер} {in или out}.

После привязки списка доступа его содержимое не может быть изменено. Не удовлетворяющий администратора список доступа должен быть удален командой **no access-list** и затем создан заново.

Расширенный список доступа создается командой:

**Router(config)#access-list {№} {permit / deny} {трансп.протокол}
{адр.ист} {адр.пол.} eq {№ порта или название прикладного протокола}**

Правила назначения в списке доступа номера порта (или, что тоже самое, прикладного протокола) представлены в таблице 1.2.

Таблица 1.2 – Правила назначения прикладных протоколов

Обозначение	Действие
lt n	Все номера портов, меньшие n.
gt n	Все номера портов, большие n.
eq n	Порт n
neq n	Все порты, за исключением n.
range n m	Все порты от n до m включительно.

Распространенные прикладные протоколы и соответствующие им стандартные номера портов приведены в таблице 1.3.

Таблица 1.3 – Номера портов некоторых прикладных протоколов

Номер порта	Транспортный протокол	Прикладной протокол	Ключевое слово в команде access-list
20	TCP	FTP	data ftp_data
21	TCP	Управление сервером FTP	ftp
22	TCP	SSH	
23	TCP	Telnet	telnet
25	TCP	SMTP	Smtп
53	UDP, TCP	DNS	Domain
67, 68	UDP	DHCP	nameserver
69	UDP	TFTP	Tftp
80, 8080	TCP	HTTP (WWW)	www
110	TCP	POP3	pop3
161	UDP	SNMP	Snmp



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

Рассмотрим пример создания стандартного списка доступа для сети, схема которой показана на рисунке 1.2. На рисунке укажем узлы, находящиеся в подсетях 192.168.0.0/24 и 192.168.1.0/24.

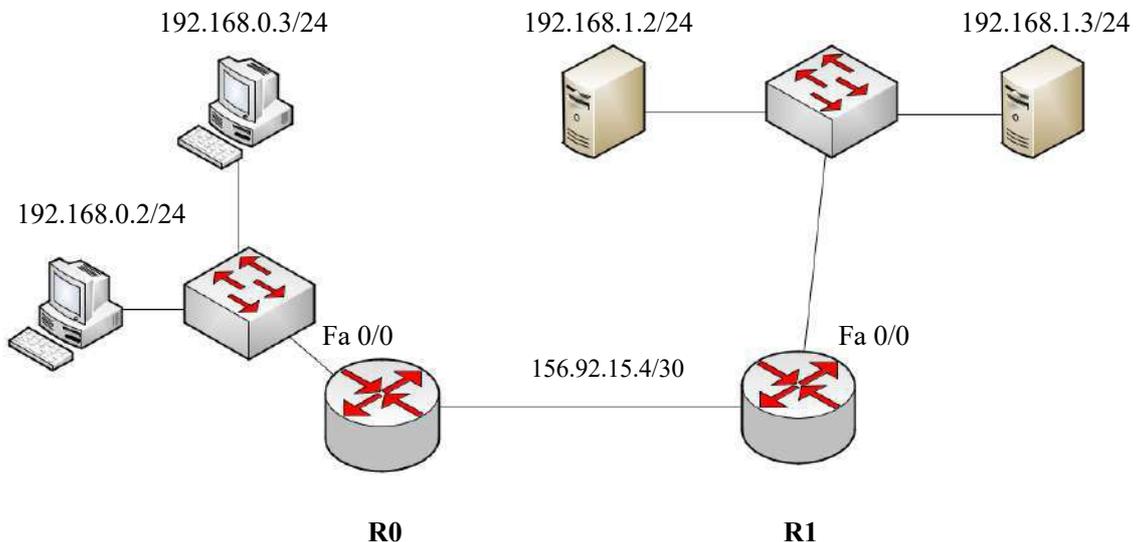


Рисунок 1.2 – Схема сети

Предположим, что к серверу, находящемуся в подсети 192.168.1.0/24 по адресу 192.168.1.2/24, доступ из подсети 192.168.0.0/24 разрешен только компьютеру 192.168.0.2/24. Это правило можно сконфигурировать с использованием стандартного списка доступа на интерфейсе Fa 0/0 маршрутизатора R1.

Для этого в режиме глобального конфигурирования на маршрутизаторе R1 необходимо выполнить следующие команды:

```
Router1(config)#access-list 10 permit 192.168.0.2
```

```
Router1(config)#interface fa 0/0
```

```
Router1(config-if)#ip access-group 10 out
```

Первая команда создает на маршрутизаторе список доступа с номером 10, который разрешает (permit) передачу пакетов с адресом источника 192.168.0.2.

Вторая команда является командой перехода к конфигурированию интерфейса Fa 0/0.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Третья команда привязывает список доступа с номером 10 к интерфейсу Fa 0/0 и указывает на направление передачи – исходящее (out).

Созданный таким образом список доступа будет состоять из двух строк. Первая строка в явной форме разрешает передавать на интерфейс маршрутизатора Fa 0/0 пакеты с адресом источника 192.168.0.2. Вторая строка в неявном виде запрещает (deny any) передавать на этот интерфейс все остальные пакеты.

Проанализируем действия маршрутизатора R1 при поступлении на его внешний интерфейс пакета после создания списка доступа.

Если пакет поступил из подсети 156.92.15.4 и предназначен серверу 192.168.1.2, маршрутизатор, определив по таблице маршрутизации выходной интерфейс, передает этот пакет в буфер интерфейса Fa 0/0.

Затем анализируется список, начиная с первой строки. Если источник имеет адрес 192.168.0.2 (совпадение в первой строке списка произошло), пакет инкапсулируется в кадр Ethernet и передается серверу. Если источник имеет любой другой адрес (совпадения в первой строке списка не произошло), происходит обращение ко второй неявной строке списка (deny any), и пакет отбрасывается.

16. случае, если необходимо обеспечить доступ к серверу и второго компьютера подсети 192.168.0.0/24 с адресом 192.168.0.3, команды конфигурирования будут выглядеть следующим образом:

```
Router1(config)#access-list 10 permit 192.168.0.2
```

```
Router1(config)#access-list 10 permit 192.168.0.3
```

```
Router1(config)#interface fa 0/0
```

```
Router1(config-if)#ip access-group 10 out
```

Очевидно, что список доступа теперь содержит три строки – две явные и одну неявную.

Очевидно, что рассмотренный способ конфигурирования списков доступа удобен в том случае, если доступ к какому-либо ресурсу (серверу) необходимо обеспечить небольшому количеству источников



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

(компьютеров). Если же, например, в подсети 192.168.0.0/24 значительное количество компьютеров, такое конфигурирование становится неудобным - подверженным ошибкам, так как для каждого из них необходимо отдельно создавать строку списка.

Поэтому при создании списков доступа можно использовать wildcard маски. В этом случае в строке списка может содержаться указание на передачу или фильтрацию пакетов не с адресами конечных узлов, а с адресами сетей (подсетей), в которые они входят.

Правило использования масок в этом случае можно сформулировать следующим образом – нулевые значения разрядов маски означают требование обработки соответствующих разрядов адреса, а единичные значения разрядов маски означают игнорирование соответствующих разрядов адреса. Например, если wildcard маска имеет вид 0.0.0.0, то проверять условие необходимо для всех разрядов адреса источника прибывшего пакета. Если же маска имеет вид 0.0.0.255, то проверять условие необходимо только для первых трех байтов адреса источника.

Предположим, что доступ к тому же серверу (адрес 192.168.1.2) должны получить все компьютеры подсети 192.168.0.0/24. В этом случае на маршрутизаторе R1 необходимо выполнить команды:

```
Router1(config)#access-list 10 permit 192.168.0.0 0.0.0.255
```

```
Router1(config)#interface fa 0/0
```

```
Router1(config-if)#ip access-group 10 out
```

Необходимо отметить, что, если нужно разрешить какому-либо одному узлу из другой подсети (например, 192.168.2.2/24) доступ к этому же серверу, создаваемый список необходимо дополнить командой

```
Router1(config)#access-list 10 permit 192.168.2.2
```

```
0.0.0.0 или, что то же самое, командой
```

```
Router1(config)#access-list 10 permit host 192.168.2.2
```

Создание списков доступа очень похоже на создание «белых» и «черных» списков на телефоне. При создании «белого» списка принимать



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

разрешено только вызовы от источников, номера которых внесены в «белый» список, остальные вызовы отбрасываются. При использовании «черного» списка отбрасываются только вызовы от источников, внесенных в список, остальные вызовы принимаются. Основное отличие от телефонных вызовов состоит в том, что в списки **permit** и **deny** вносятся не телефонные номера, а значения заголовков различных уровней.

Используя эту аналогию с «белыми» и «черными» списками телефона, можно отметить, что рассмотренные способы аналогичны созданию в телефоне «белых» списков – указанные в списке доступа адреса являются разрешенными, остальные – запрещенными.

В ряде случаев более удобным является использование аналогии «черного» списка – разрешено передавать данные от всех, кроме тех, кто указан в черном списке.

Предположим, что к тому же серверу необходимо обеспечить доступ всем компьютерам, кроме одного, имеющего адрес 192.168.0.15. Конфигурирование такого списка будет иметь вид:

```
Router1(config)#access-list 11 deny host 192.168.0.15
```

```
Router1(config)#access-list 11 permit any
```

```
Router1(config)#interface fa 0/0
```

```
Router1(config-if)#ip access-group 11 out
```

Напомним, что по умолчанию у создаваемых списков доступа неявно присутствует заключительная строка **deny any** – запретить все. В данном случае мы заменили эту строку на **permit any** – разрешить все. Соответственно, доступ к серверу будет разрешен всем, кроме компьютера с адресом 192.168.0.15.

Рассмотрим теперь применение расширенного списка для конфигурирования маршрутизатора R1 (рисунок 3.2), при этом должны быть выполнены следующие условия:

- компьютеру 192.168.0.2/24 необходимо предоставить доступ к web-серверу с адресом 192.168.1.2 по протоколу WWW;



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

и всем компьютерам подсети 192.168.0.0/24 необходимо предоставить доступ к FTP-серверу с адресом 192.168.1.3 по протоколу FTP.

Команды конфигурирования в этом случае будут выглядеть следующим образом:

```
Router1(config)#access-list 110 permit tcp host 192.168.0.2 host 192.168.1.2 eq www
```

```
Router1(config)#access-list 110 permit tcp 192.168.0.0 0.0.0.255 host 192.168.1.3 eq ftp
```

```
Router1(config)#interface fa 0/0
```

```
Router1(config-if)#ip access-group 110 out
```

Очевидно, что указанный способ аналогичен созданию «белого» списка в телефоне, так как третье неявное условие, находящееся в конце списка, блокирует все, что не разрешено.

Рассмотрим пример, когда удобнее использовать аналогию «черного» списка в телефоне.

На маршрутизаторе R1 должны быть выполнены следующие условия:

компьютеру 192.168.0.2/24 необходимо запретить доступ к серверу с адресом 192.168.1.2 по протоколу WWW, но разрешить доступы к другим сервисам;

всем компьютерам подсети 192.168.0.0/24 необходимо запретить доступ к серверу с адресом 192.168.1.3 по протоколу FTP, но разрешить доступ к другим сервисам.

Команды конфигурирования в этом случае будут иметь вид:

```
Router1(config)#access-list 110 deny tcp host 192.168.0.2 host 192.168.1.2 eq www
```

```
Router1(config)#access-list 110 deny tcp 192.168.0.0 0.0.0.255 host 192.168.1.3 eq ftp
```

```
Router1(config)#access-list 110 permit ip any any
```

```
Router1(config)#interface fa 0/0
```

```
Router1(config-if)#ip access-group 110 out
```

Запись **permit ip any any** означает, что весь остальной трафик от любого источника к любому получателю должен передаваться.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Просмотреть созданные на маршрутизаторе списки доступа можно по команде **show access-list**, а списки, настроенные на конкретных интерфейсах, командами **show ip interfaces** или **show running-config**. На рисунке 1.3 показаны настроенные списки доступа для рассмотренного здесь примера.

```
IOS Command Line Interface
```

```
Router#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#access-list 110 deny tcp host 192.168.0.2 host 192.168.1.2 eq www

Router(config)#access-list 110 deny tcp 192.168.0.0 0.0.0.255 host 192.168.1.3 eq ftp
Router(config)#access-list 110 permit ip any any
Router(config)#interface fa 0/0
Router(config-if)#ip access-group 110 out
Router(config-if)#^Z
Router#
%SYS-5-COMFIG_I: Configured from console by console

Router#show access-list
Extended IP access list 110
    deny tcp host 192.168.0.2 host 192.168.1.2 eq www
    deny tcp 192.168.0.0 0.0.0.255 host 192.168.1.3 eq ftp
    permit ip any any
Router#
```

Рисунок 1.3 – Конфигурирование списка доступа и его просмотр

Списки доступа также желательно использовать и для конфигурирования удаленного доступа к устройствам.

В настоящее время чаще используются не нумерованные, а именованные списки доступа. Удобство использования именованных списков доступа заключается прежде всего в том, что названию списка можно придать определенный смысл (INTERNET, ADMIN, FTP, и т.д.). Так как именованный список не имеет номера, который однозначно определяет его вид (таблица 3.1), при создании такого списка необходимо явно указать,



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

какой именно список создается – стандартный или расширенный. Команда создания именованного списка доступа имеет вид:

ip access-list <standard/extended> <имя>

<правило 1>

<правило 2>

<правило n>

Параметр **standard/extended** указывает на вид создаваемого списка, а правила прописываются аналогично нумерованным спискам.

1.5 Отчет по работе:

Демонстрация функционирования МСЭ.



Лабораторная работа 2. Исследование свойств межсетевого экрана с сохранением состояний

2.1 Цель работы: Получение навыков конфигурирования межсетевого экрана с сохранением состояний и исследование его свойств.

2.2 Перечень оборудования:

Маршрутизаторы Cisco;

ПК с установленным ПО Cisco Packet Tracer;

Локальная сеть.

2.3 Задание:

3.4.1 Используя Cisco Packet Tracer, построить сеть, показанную на рисунке 3.4. Произвести конфигурирование сетевых устройств для обеспечения доступа всех серверов из внутренней сети.

3.4.2 Убедиться в возможности доступа во внутреннюю сеть извне с использованием произвольного протокола.

3.4.3 Настроить инспектирование TCP-трафика и сконфигурировать список доступа. Убедиться в доступности серверов из внутренней сети и недоступности ресурсов внутренней сети извне.

2.4 Указания по проведению работы

Такие межсетевые экраны еще называют экранами с сохранением сессий (statefull firewall). Суть заключается в том, что при запросе на установление соединения (например, TCP-сессии) маршрутизатор запоминает эту сессию и при поступлении извне пакета сверяет его со всеми текущими сессиями. Если принятый извне пакет относится к какой-либо текущей сессии, он продвигается во внутреннюю сеть, в противном случае – отбрасывается.

Для конфигурирования межсетевого экранирования на устройствах Cisco необходимо в явном виде указать, трафик каких протоколов должен отслеживаться (инспектироваться). Для этого используется команда

ip inspect name <имя правила> <название протокола>



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Данная команда выполняется в режиме глобального конфигурирования.

Аналогично спискам доступа, созданное правило необходимо привязать к интерфейсу с указанием направления передачи:

R1(config)#int fa0/0 – переход в режим конфигурирования интерфейса fa 0/0;

R1(config-if)#ip inspect <имя правила> <in/out> - привязка правила к интерфейсу с указанием направления передачи.

Необходимо отметить, что правило можно привязывать как к внутреннему, так и ко внешнему интерфейсу маршрутизатора, однако направление передачи должно соответствовать направлению запросов из внутренней сети ко внешней. Соответственно, если правило привязывается к внутреннему интерфейсу, направление передачи – входящее (in), если к внешнему – исходящее (out).

Приведем пример межсетевого экранирования для сети, показанной на рисунке 2.1. Для удобства разместим маршрутизатор R1 во внешней сети, в которой также располагаются два сервера – web-сервер и ftp-сервер. Произведем настройку всего оборудования таким образом, чтобы из внутренней сети были доступны оба сервера.

Создадим правило для инспектирования запросов к web-серверу (протокол HTTP) с именем HTTP и привяжем его к внутреннему интерфейсу fa0/0 с входящим направлением передачи (рисунок 2.2).



Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»

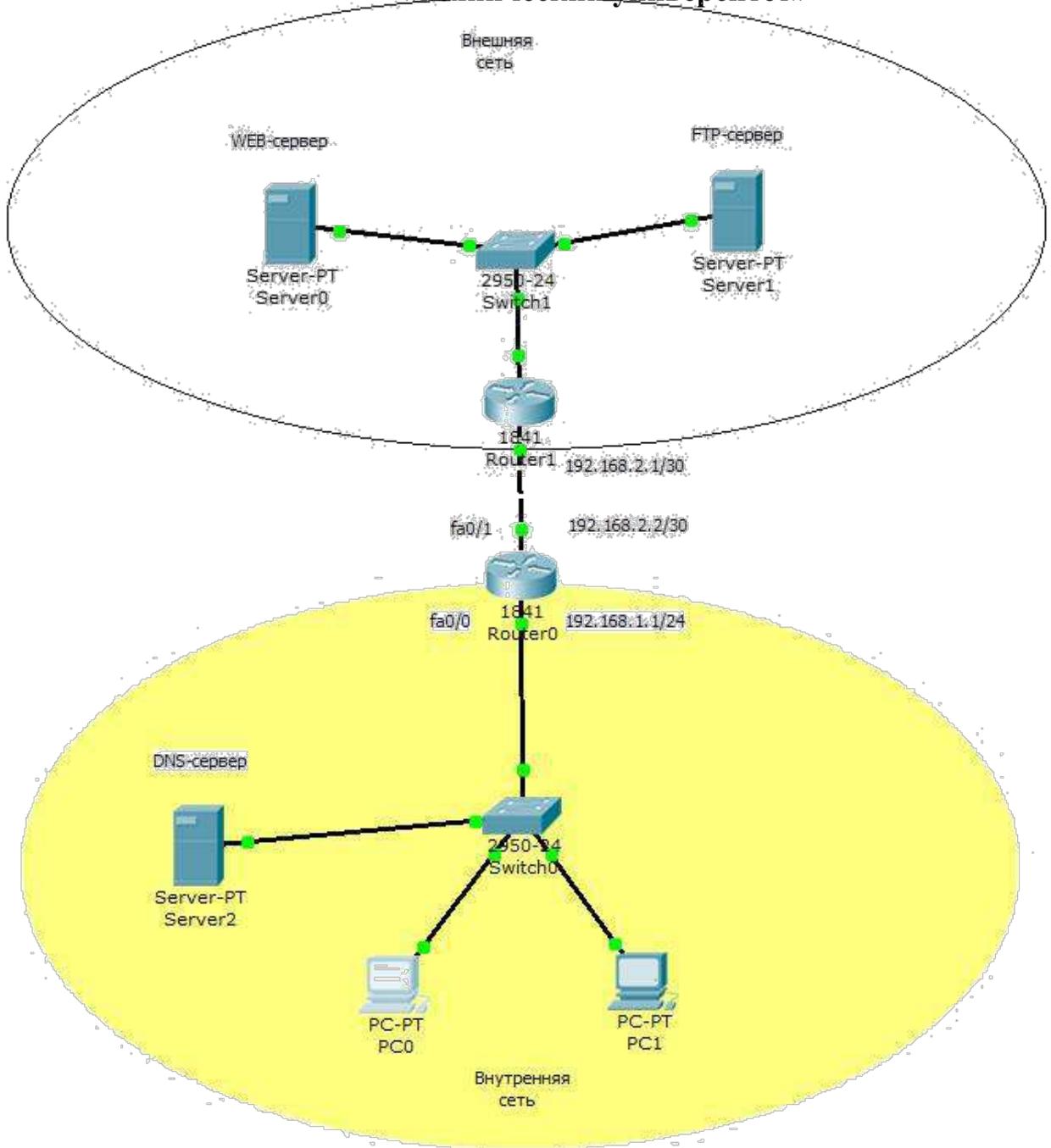


Рисунок 2.1 – Пример сети

```
Router>en
Router#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip inspect name HTTP http
Router(config)#int fa0/0
Router(config-if)#ip inspect HTTP in
Router(config-if)#
```

Рисунок 2.2 – Конфигурирование инспектирования протокола HTTP



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Следует отметить (и это очень важно!), что инспектирование трафика необходимо применять совместно со списками доступа. В нашем примере, когда списки доступа не были созданы, http-запросы, поступающие из внутренней сети, будут инспектироваться. Однако если из внешней сети также поступит http-запрос, он пройдет во внутреннюю сеть, так как не существует списка доступа, обеспечивающего фильтрацию этого запроса.

Для проверки этого разместим во внешней сети ПК с адресом 213.80.65.4 и попробуем соединиться с сервером внутренней сети по протоколу HTTP (рисунок 2.3).

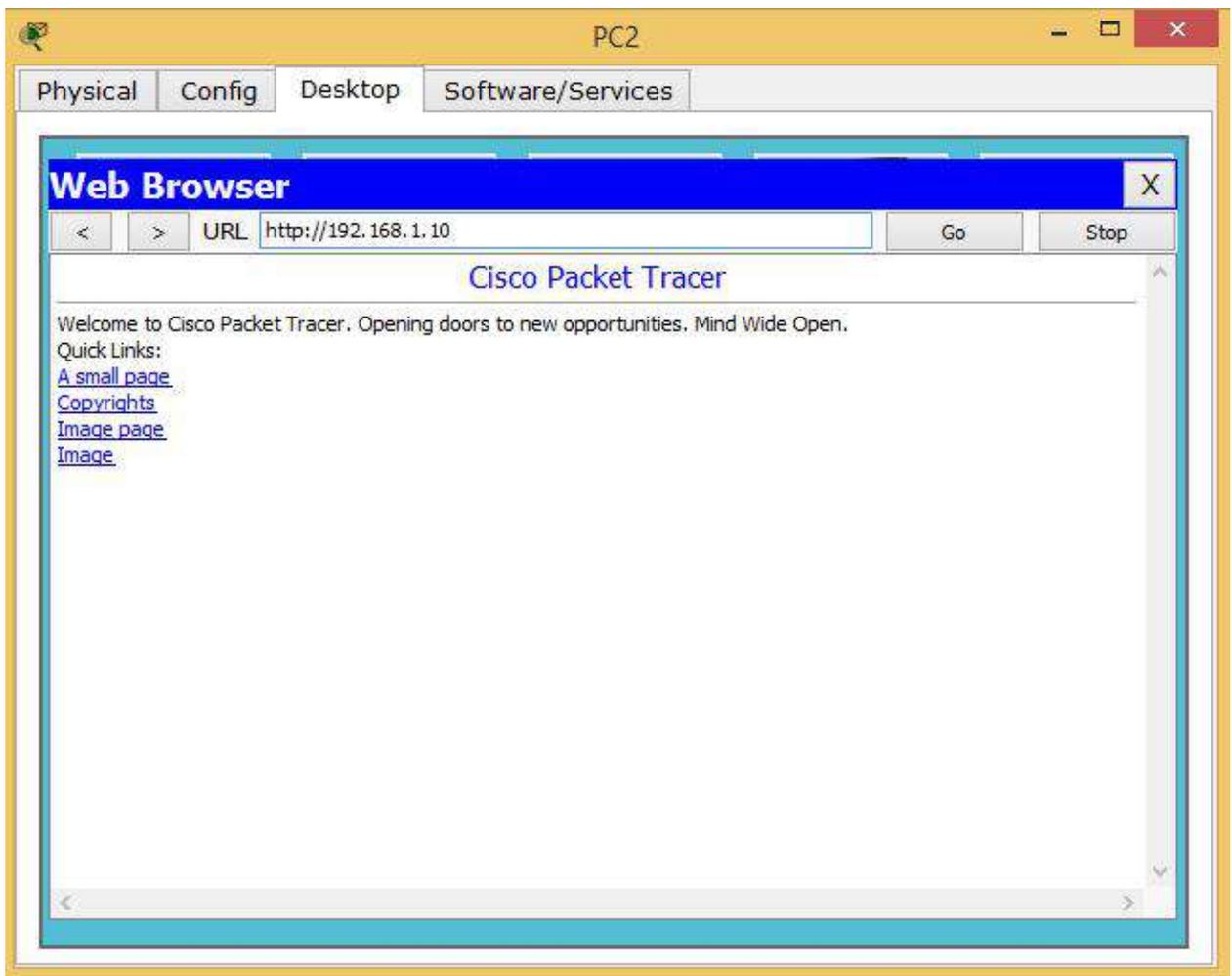


Рисунок 2.3 – Соединение с сервером внутренней сети по протоколу HTTP

В качестве внутреннего сервера мы использовали DNS-сервер с адресом 192.168.1.10/24. Как видно из рисунка 2.3, соединение прошло успешно.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Для защиты внутренней сети создадим на маршрутизаторе R0 список доступа, запрещающий передачу всех IP-пакетов, и привяжем его к внешнему интерфейсу fa0/1 с указанием входящего направления (рисунок 2.4).

```
Router>en
Router#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip access-list ex FRW
Router(config-ext-nacl)#deny ip any any
Router(config-ext-nacl)#
```

Рисунок 2.4 – Создание списка доступа

Теперь снова попытаемся послать HTTP-запрос из внешней сети (рисунок 2.5).

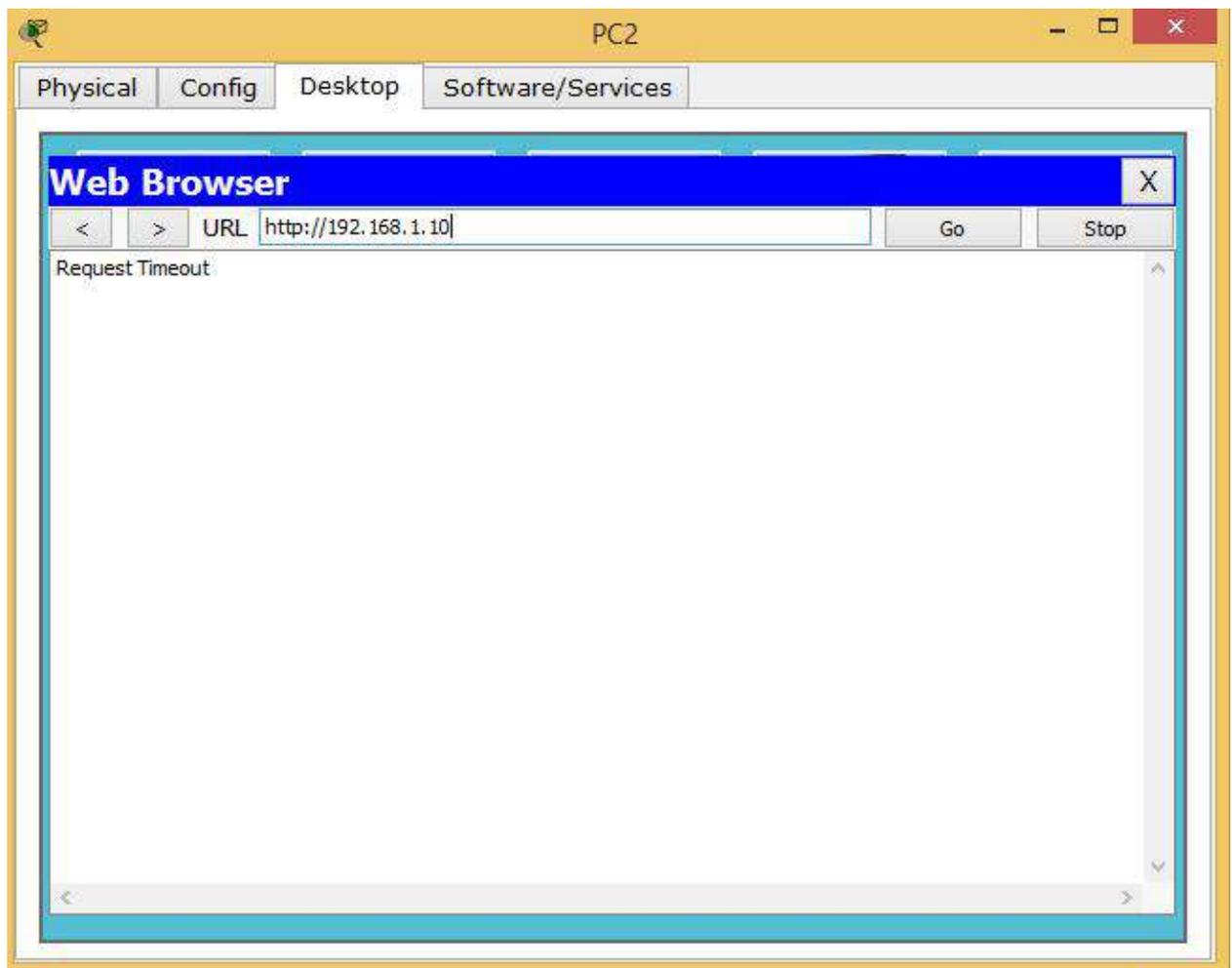


Рисунок 2.5 – Отсутствие соединения с сервером внутренней сети по протоколу HTTP



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Как видно из рисунка 2.5, из внешней сети сервер не доступен. При этом из внутренней сети web-сервер остается доступным.

С учетом созданного списка доступа FRW остальные протоколы (кроме HTTP) не инспектируются. Следовательно, если послать из внутренней сети запрос по какому-либо другому протоколу, ответ получен не будет, так как его заблокирует список доступа на внешнем интерфейсе маршрутизатора. Попробуем получить из внутренней сети доступ к ftp-серверу (рисунок 2.6).

```
Packet Tracer PC Command Line 1.0
PC>(Disconnecting from ftp server)

Packet Tracer PC Command Line 1.0
PC>ftp 213.80.65.3
Trying to connect...213.80.65.3

%Error opening ftp://213.80.65.3/ (Timed out)

Packet Tracer PC Command Line 1.0
PC>(Disconnecting from ftp server)

Packet Tracer PC Command Line 1.0
PC>
```

Рисунок 2.6 – Попытка соединения с ftp-сервером

Как видим, попытка не увенчалась успехом. Если же настроить инспектирование FTP-трафика, доступ к ftp-серверу будет возможен. Иллюстрировать здесь это не будем, так как Cisco Packet Tracer в силу своей ограниченной функциональности это не поддерживает. Поэтому



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

инспектирование разных видов трафика необходимо производить либо на реальном оборудовании, либо с использованием симулятора GNS3.

Однако Cisco Packet Tracer поддерживает инспектирование TCP-трафика. Так как и протокол HTTP, и протокол FTP использует для передачи TCP-сегменты, после настройки TCP-инспектирования доступны окажутся и http и ftp серверы (рисунки 2.7, 2.8).

```
Router(config)#ip inspect name HTTP tcp
Router(config)#
Router(config)#^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

Рисунок 2.7 – Настройка инспектирования TCP-трафика

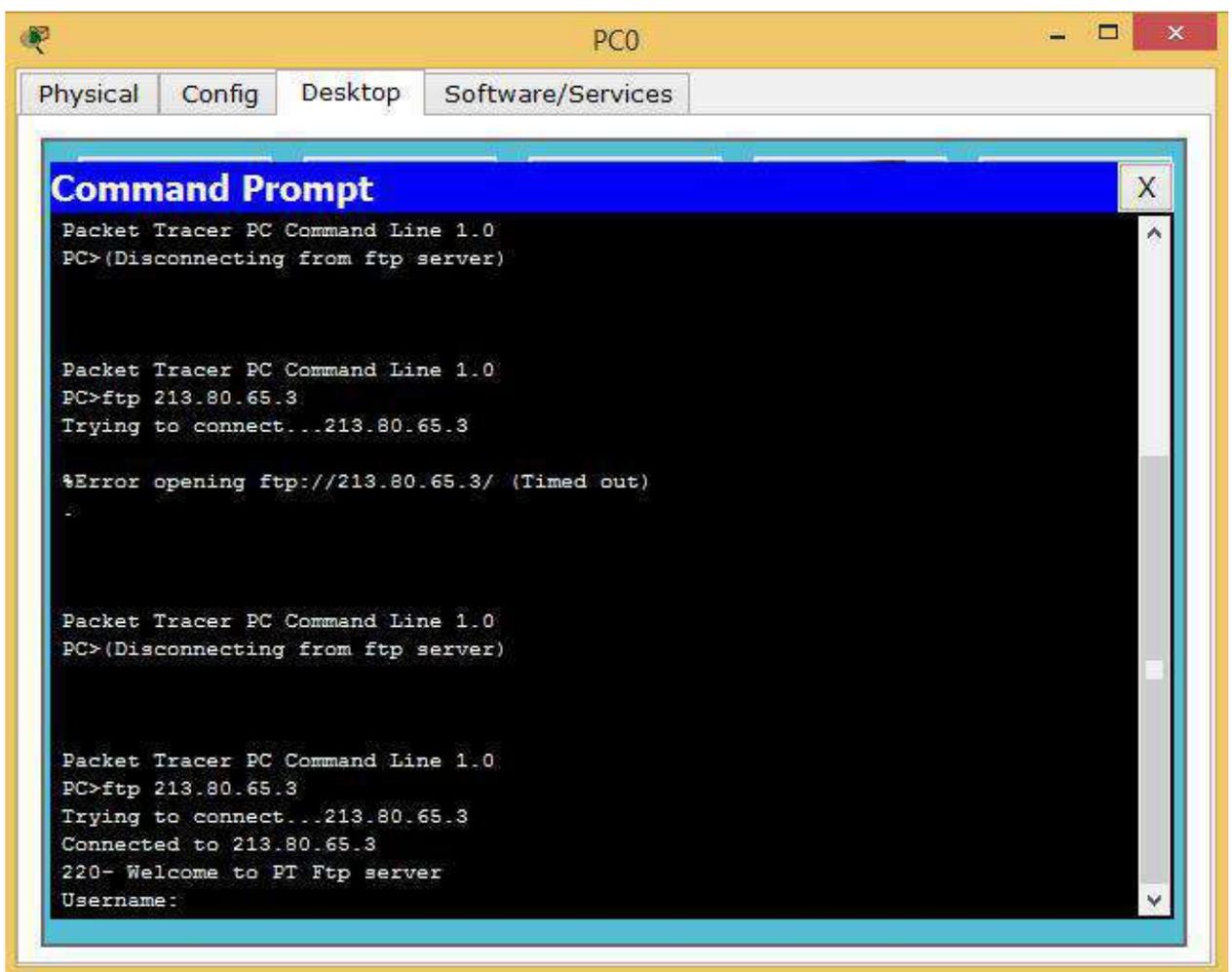


Рисунок 2.8 – Доступ к ftp-серверу после настройки инспектирования TCP-трафика



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Рассмотренные настройка межсетевого экрана являются базовыми, а более тонкие настройки применяются в случае, если производится разделение сети с различными зонами безопасности, которые рассмотрим на следующем занятии.

2.5 Отчет по работе:

Демонстрация функционирования МСЭ.



Лабораторная работа 3. Исследование свойств межсетевого экрана ZBFW

1.1 Цель работы: Получение навыков конфигурирования межсетевого экрана ZBPF и исследование его свойств.

1.2 Перечень оборудования:

Локальная сеть;

ПК с установленным ПО Cisco Packet Tracer;

Маршрутизаторы Cisco.

1.3 Задание:

Собрать сеть, показанную на рисунке 3.1, настроить адресацию по следующим исходным данным:

- внутренняя сеть – 192.168.X.0;
- демилитаризованная зона – 192.168.X+10.0;
- внешняя сеть – 213.80.X.0.

Проверить связность сети.

Сконфигурировать на маршрутизаторе межсетевой экран ZBFW, обеспечивающий выполнение следующих правил:

- из внутренней сети разрешены все запросы к внешней сети;
- к демилитаризованной зоне разрешены запросы из внутренней сети по протоколам Telnet и SSH и только с адресов, принадлежащих внутренней сети;
- из внешней сети запросы во внутреннюю сеть запрещены; ○ из внешней сети запросы в демилитаризованную зону разрешены только по протоколу HTTP.

Проверить работоспособность межсетевого экрана.

1.4 Указания к проведению работы.

Zone-Based Policy Firewall (ZBFW) – относительно новое направление на маршрутизаторах под управлением операционной системы Cisco IOS для



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

конфигурирования правил доступа между сетями. До появления этой технологии трафик фильтровался с помощью списков доступа ACL (рассмотренных в параграфе 3.1) и динамической инспекции трафика Context-Based Access Control (СВАС) (в настоящем пособии не рассматривается). И ACL и правила СВАС применяются непосредственно на физические интерфейсы, что во многих случаях не способствует масштабируемости и гибкости сетевых решений. Такая модель ограничивает степень детализации политик межсетевого экрана и вызывает путаницу правильного применения политики межсетевого экранирования, особенно в случаях, когда политика межсетевого экрана должна применяться между несколькими интерфейсами. Zone-Based Policy Firewall меняет конфигурацию межсетевого экрана от старой интерфейсной модели на более гибкую модель, основанную на зонах безопасности.

Зона безопасности состоит из набора различных интерфейсов, которые должны иметь одинаковую политику сетевой безопасности или, иначе говоря, одинаковый уровень доверия. В каждую из зон может входить один или несколько интерфейсов. После создания зон настраиваются правила для взаимодействий между зонами. Такой подход облегчает настройки правил межсетевого экрана, так как правила определяются не для отдельных интерфейсов, а для множества интерфейсов, входящих в одну зону. Кроме того в Zone-Based Policy Firewall используется язык Cisco Policy Language (CPL), который позволяет более гибко, чем в предыдущих версиях межсетевого экрана, настраивать правила фильтрации трафика.

В большинстве случаев сеть делится, как минимум, на три зоны:

- / внутренняя зона, где расположены пользователи (inside);
- / внешняя зона (Интернет – outside);
- / демилитаризованная зона, где расположены серверы, к которым должен быть обеспечен доступ извне (dmz).

Важно, что по умолчанию весь трафик между различными зонами будет запрещен, весь трафик внутри зоны – разрешен.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

На первом этапе настройки межсетевого экрана необходимо создать зоны. Зоны создаются командой, выполняемой в режиме глобального конфигурирования:

zone security <имя зоны>

После этого необходимо создать пары зон, между которыми будет передаваться трафик:

zone-pair security <имя пары> source <имя зоны> destination <имя зоны>

Необходимо отметить, что пары зон являются однонаправленными. То есть, если в нашей сети предполагается двунаправленная передача данных между внутренней и внешней зонами, необходимо создать две пары зон. Например (считаем, что внутренней зоне было присвоено имя IN, а внешней – OUT, имя пар IN_OUT и OUT_IN):

zone-pair security IN_OUT source IN destination OUT

zone-pair security OUT_IN source OUT destination IN

Как указывалось выше, по умолчанию передача трафика между созданными парами зон запрещена. Для формирования разрешенного для передачи трафика необходимо определить критерии, по которым отсортировывается нужный трафик. Для этого используется так называемый class-map (дословно – классовая карта). Class-map определяет, какой именно трафик будет инспектироваться (проходить между зонами, также проходить будут ответы на этот трафик). Фильтроваться трафик может по критериям с 3-го по 7-й уровней модели OSI (т.е. начиная от IP-адреса и заканчивая трафиком определенного приложения или сервиса прикладного уровня). Определяться трафик может списками доступа, значением CoS, типом протокола и еще рядом других параметров. Критериев может присутствовать одновременно несколько. При этом можно указать, должен ли трафик попадать под все эти критерии (match-all) или под любой из них (match-any). Таким образом, основная задача class-map – отфильтровать необходимый тип трафика.

Для создания class-map используется команда:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

class-map type inspect match-all/match-any <имя class-map>

После этого мы попадаем в конфигурирование созданного class-map, и далее необходимо использовать команду **match** с указанием критериев, по которым отсортировывается трафик:

/ **access-group** - стандартный, расширенный или именованный список доступа, который может фильтровать трафик на основании IP адреса и порта источника и приемника. Это единственный способ выделить трафик от конкретного источника к конкретному получателю;

/ **protocol** - это протоколы уровня 4 (TCP, UDP, ICMP), а также прикладные сервисы, такие как HTTP, SMTP, DNS, и т.д. Может быть указан любой известный или определяемый пользователем сервис;

/ **class-map** - подчиненный класс, который предоставляет дополнительные критерии соответствия;

/ **not** - определяет, что любой трафик, который не соответствует указанному сервису или протоколу, или листу доступа, будет выбран в данном class-map.

Важно, что критерии вводятся списком, и порядок обработки списка последовательный, как и у списков доступа. Например, если при конфигурировании class-map match-any мы использовали команды

match protocol http

match protocol tcp

то при обработке пакета сначала будет проверено его соответствие протоколу HTTP. Если найдено соответствие, то далее будет инспектироваться этот трафик, и следующее условие не будет проверяться. Если же команды поменять местами, то пакет сначала попадет под инспектирование трафика TCP.

Политики межсетевого экранирования определяются командой **policy-map**. Команда **policy-map** определяет действие, которое будет произведено с отфильтрованным с помощью команды **class-map** трафиком. Существует три основных действия, которые применимы к классифицированному трафику:



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Drop – Трафик, обрабатываемый этим действием, отбрасывается и никакого уведомления на удаленный хост не высылается (в противоположность классическим листам доступа (ACL), когда высылается ICMP-сообщение Host Unreachable). Каждая карта политик имеет скрытый класс `class-default`, для которого сконфигурировано действие **Drop** (аналогично строке `deny any any` в любом списке доступа).

Pass – Пропускает трафик, не включая инспекцию протокола. Это действие позволяет маршрутизатору пересылать трафик из одной зоны в другую, при этом он не отслеживает состояние соединений или сессий. Это действие разрешает прохождение трафика только в одном направлении. Чтобы обратный трафик был передан, должна быть соответствующая политика и для него. Это действие полезно для таких протоколов, как IPSec ESP, IPSec AH, ISAKMP и других по своей сути безопасных протоколов с предсказуемым поведением.

Inspect - Включает динамическую инспекцию для трафика, который проходит от зоны источника к зоне приемника, и автоматически разрешает обратный трафик даже для сложных протоколов, таких как H.323. Например, если трафик передается из зоны IN в зону OUT, маршрутизатор поддерживает информацию о соединениях или сеансах для TCP и UDP трафика. Поэтому маршрутизатор разрешает обратный трафик из зоны OUT / зону IN в качестве ответов на запросы соединений из IN в OUT.

Формат команды:

policy-map type inspect <имя policy-map >

После этого мы попадаем в режим конфигурирования созданного `policy-map`, в котором указываем, какой именно `class-map` должен обрабатываться, и затем указываем необходимое действие:

class type inspect <имя class-map>

inspect/pass/drop

Теперь созданные политики необходимо применить к парам зон, которые уже были созданы ранее (пары зон можно создать и на этом шаге):



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

**zone-pair security <имя пары> source <имя зоны> destination <имя зоны>
service-policy type inspect <имя policy-map >**

Осталось в явном виде указать маршрутизатору, какие его интерфейсы относятся к какой зоне:

interface <имя интерфейса>

zone-member security <имя зоны>

Приведем пример конфигурирования межсетевого экранирования на примере сети, показанной на рисунке 3.1.



Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

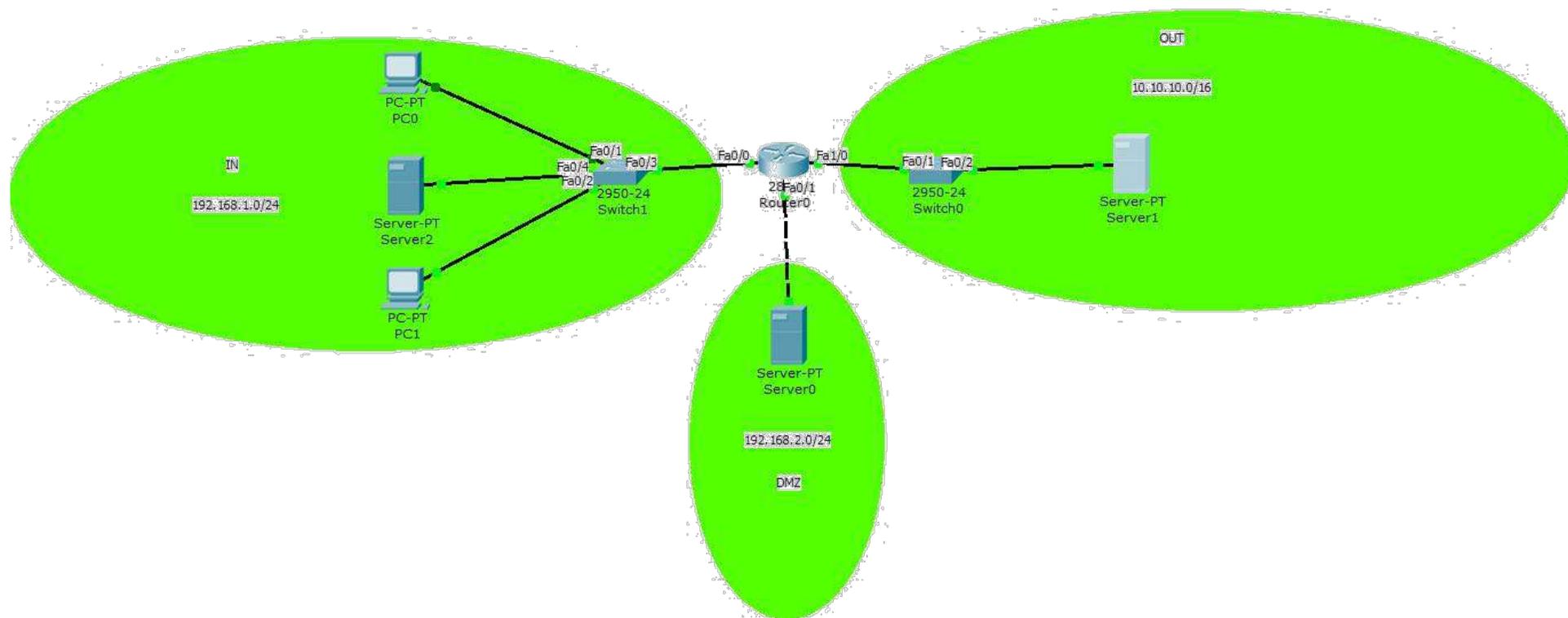


Рисунок 3.1 – Пример сети



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Для простоты будем считать, что интерфейсам маршрутизатора присвоены первые адреса из адресного пространства подсетей (192.168.1.1/24 в подсети IN, 10.10.10.1/16 в подсети OUT, и т.д.)

Сначала необходимо определить зоны:

R0(config)#zone security IN

R0(config)#zone security OUT

R0(config)#zone security DMZ

Так как созданные зоны пока не привязаны ни к каким интерфейсам, никакого ограничения в передаче трафика после создания зон не произойдет.

Сначала сконфигурируем межсетевой экран для информационного обмена между зонами IN и OUT. Считаем, что из внутренней сети разрешены любые запросы к серверу, находящемуся во внешней сети. Таким образом, при создании class-map в этом направлении удобно использовать стандартный список доступа, разрешающий передавать данные от всех рабочих станций подсети 192.168.1.0/24:

R0(config)#access-list permit 10 192.168.1.0 0.0.0.255

Создаем class-map для трафика, удовлетворяющего условию списка доступа 10, присвоив самому class-map имя FROM-IN:

R0(config)#class-map type inspect match-any FROM-

IN и указываем, какой трафик входит в созданный class-map:

R0(config-cmap)#match access-group 10

Создаем policy-map с именем FROM-INSIDE (в принципе, имена class-map и policy-map могут совпадать, здесь специально выбраны разные имена):

R0(config)#policy-map type inspect FROM-INSIDE

указываем, какой class-map должна обрабатывать политика:

R0(config-pmap)#class type inspect FROM-IN

и указываем нужное действие – инспектировать:



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

R0(config-pmap-c)#inspect

Теперь обращаемся к интерфейсам для указания, к каким зонам они относятся. Одновременно можно назначить интерфейсам IP-адреса и включить их, если этого не было сделано раньше:

R0(config)#int fa0/0

R0(config-if)#ip address 192.168.1.1 255.255.255.0

R0(config-if)#no shutdown

R0(config-if)#zone-member security IN

R0(config)#int fa1/0

R0(config-if)#ip address 10.10.10.1 255.255.0.0

R0(config-if)#no shutdown

R0(config-if)#zone-member security OUT

Так как интерфейсы маршрутизатора теперь принадлежат к разным зонам, передача трафика между ними запрещена. Для разрешения передачи между ними трафика в соответствии с созданной политикой создадим зонную пару с именем IN-TO-OUT:

**R0(config)#zone-pair security IN-TO-OUT source IN destination
OUT**

и применим к ней созданную политику:

R0(config-sec-zone-pair)#service-policy type inspect FROM-INSIDE

Заметим, что пару для обратного направления OUT-IN мы не создавали. Это связано с тем, что по условиям задачи любой трафик извне запрещен, за исключением ответов на запросы, которые поступили из внутренней сети (они инспектируются). Проверить работоспособность сконфигурированного межсетевого экрана достаточно просто – если из внутренней сети послать любой запрос (например, ping или http-запрос), то внутренний компьютер должен получить ответ (рисунок 3.2). Если же послать запрос с внешнего сервера, ответа не будет – запрос будет отброшен маршрутизатором (рисунок 3.3).



Негосударственное частное образовательное
учреждение
высшего образования

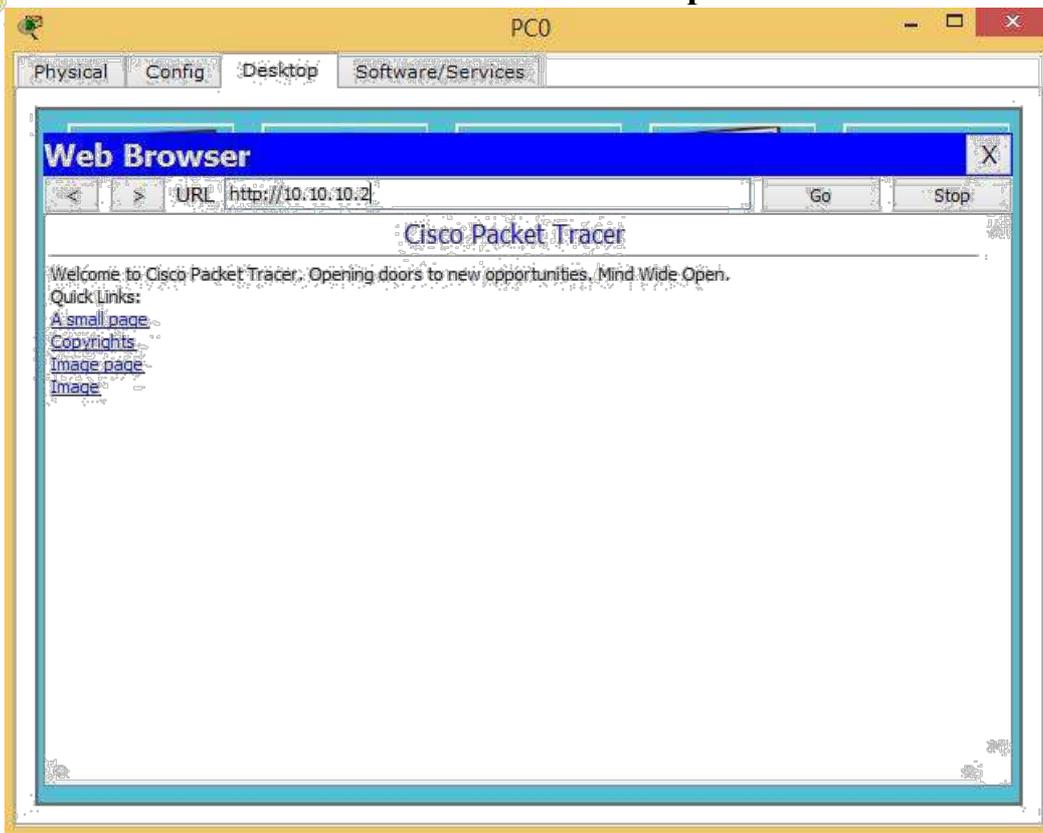


Рисунок 3.2 – Получение ответа при запросе из внутренней сети

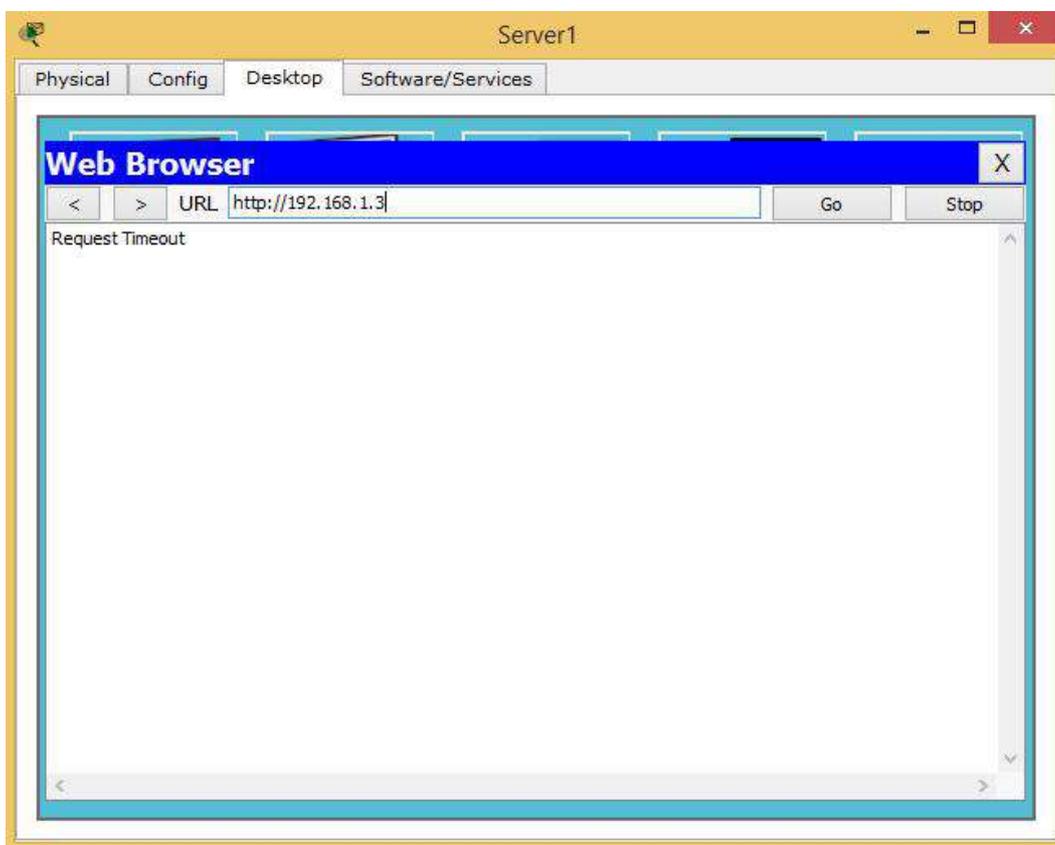


Рисунок 3.3 – Отсутствие ответа при запросе из внешней сети



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Перейдем теперь к настройке демилитаризованной зоны. Здесь ситуация несколько иная – к серверам демилитаризированной должен быть обеспечен доступ как извне (например, по протоколу HTTP, если это web-сервер), так и изнутри (например, по протоколу SSH или Telnet для конфигурирования сервера). Соответственно, в этом случае необходимо создавать как минимум две пары зон – OUT-DMZ, IN-DMZ – с различными политиками. Если же предполагается наличие доступа из демилитаризованной зоны, необходимо создавать пары DMZ-OUT или DMZ-IN.

Предположим, что для нашей сети необходимо обеспечить следующие условия:

В Доступ из внешней сети к серверу, находящемуся в DMZ, возможен только по протоколу HTTP;

\endash Доступ из внутренней сети к серверу, находящемуся в DMZ,

возможен по протоколам SSH, Telnet, HTTP, и только с тех IP-адресов, которые относятся к адресному пространству внутренней сети 192.168.1.0/24.

Привяжем интерфейс маршрутизатора fa0/1 к созданной ранее зоне DMZ, одновременно назначим ему IP-адрес и включим его:

R0(config)#int fa0/1

R0(config-if)#ip address 192.168.2.1 255.255.255.0

R0(config-if)#no shutdown

R0(config-if)#zone-member security DMZ

Очевидно, что для передачи трафика из внутренней сети к серверу DMZ необходимо создать несколько class-map, так как в каждом из них одновременно должно выполняться как минимум два условия:

22. источником трафика является внутренняя сеть и используется протокол SSH;



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

25. источником трафика является внутренняя сеть и используется протокол Telnet;

26. источником трафика является внутренняя сеть и используется протокол HTTP.

Создадим class-map для передачи трафика из внутренней сети к серверу DMZ по протоколу SSH с именем IN-DMZ-SSH:

```
R0(config)#class-map type inspect match-all IN-DMZ-  
SSH R0(config-cmap)#match access-group 10 R0(config-  
cmap)#match protocol ssh
```

Создадим class-map для передачи трафика из внутренней сети к серверу DMZ по протоколу Telnet с именем IN-DMZ-TLN:

```
R0(config)#class-map type inspect match-all IN-DMZ-  
TLN R0(config-cmap)#match access-group 10 R0(config-  
cmap)#match protocol telnet
```

Создадим class-map для передачи трафика из внутренней сети к серверу DMZ по протоколу HTTP с именем IN-DMZ-HTTP:

```
R0(config)#class-map type inspect match-all IN-DMZ-  
HTTP R0(config-cmap)#match access-group 10 R0(config-  
cmap)#match protocol http
```

Создадим policy-map с именем IN-DMZ, в которой укажем на необходимость инспектирования трафика, удовлетворяющего созданным class-map:

```
R0(config)#policy-map type inspect IN-DMZ  
R0(config-pmap)#class type inspect IN-DMZ-SSH  
R0(config-pmap-c)#inspect  
R0(config-pmap-c)#exit  
R0(config-pmap)#class type inspect IN-DMZ-TLN  
R0(config-pmap-c)#inspect  
R0(config-pmap-c)#exit
```



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

```
R0(config-pmap)#class type inspect IN-DMZ-HTTP
```

```
R0(config-pmap-c)#inspect
```

```
R0(config-pmap-c)#exit
```

```
R0(config-pmap)#
```

Для доступа к серверу DMZ из внешней сети должен использоваться только протокол HTTP, поэтому создадим один class-map с именем OUT-DMZ:

```
R0(config)#class-map type inspect match-any OUT-DMZ
```

```
R0(config-cmap)#match protocol http
```

Создадим policy-map с таким же названием и с указанием инспектировать трафик:

```
R0(config)#policy-map type inspect OUT-DMZ
```

```
R0(config-pmap)#class type inspect OUT-DMZ
```

```
R0(config-pmap-c)#inspect
```

Осталось создать пары зон и применить к ним созданные политики.

Пара IN-TO-DMZ:

```
R0(config)#zone-pair security IN-TO-DMZ source IN destination
```

```
DMZ R0(config-sec-zone-pair)#service-policy type inspect IN-DMZ
```

Пара зон OUT-TO-DMZ:

```
R0(config)#zone-pair security OUT-TO-DMZ source OUT destination
```

```
DMZ R0(config-sec-zone-pair)#service-policy type inspect OUT-DMZ
```

3.5 Отчет по работе:

Демонстрация функционирования МСЭ.



Лабораторная работа 4. Исследование VPN-туннеля

1.1 Цель работы: Получение навыков конфигурирования VPN-туннеля и исследование его свойств.

1.2 Перечень оборудования:

Локальная сеть;

ПК с установленным ПО Cisco Packet Tracer;

Маршрутизаторы Cisco.

1.3 Задание:

Собрать сеть, состоящую из двух маршрутизаторов.

Сконфигурировать VPN-туннель.

Проверить работоспособность туннеля.

1.4 Указания к проведению работы.

Здесь рассмотрим только классификацию VPN по назначению, так как именно от этого зависит настройка.

Remote Access VPN – используется для создания защищённого канала между сегментом корпоративной сети (центральным офисом или филиалом) и одиночным пользователем, который, работая дома, подключается к корпоративным ресурсам с домашнего компьютера, корпоративного ноутбука, смартфона или интернет-киоска. В терминологии Cisco такой пользователь называется Teleworker.

Site-to-Site VPN – используется для создания защищенного канала между различными сегментами корпоративной сети (центральным офисом и филиалами) через незащищенную сеть (Интернет).

Независимо от типа VPN при его конфигурировании создается туннель. Туннелирование – это процесс, в ходе которого создается защищенное логическое соединение между двумя конечными точками посредством инкапсуляции различных протоколов. При туннелировании данные упаковываются вместе со служебными заголовками в новый



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

«конверт» для обеспечения конфиденциальности и целостности всей передаваемой информации.

Понятие «туннелирование» включает в себя ряд терминов:

34. транспортируемый протокол (протокол-«пассажир») – протокол, в котором представлены данные, которые необходимо передать через туннель;
35. транспортирующий (несущий) протокол – протокол, на базе которого данные доставляются через некоторую транзитную сеть (например, через Интернет);
36. протокол инкапсуляции – протокол, описывающие правила инкапсуляции данных транспортируемого протокола в пакет транспортирующего протокола.

Протокол транзитной сети является несущим, а протокол объединяемых сетей — транспортируемым. Пакеты транспортируемого протокола помещаются в поле данных пакетов несущего протокола с помощью протокола инкапсуляции. Пакеты-«пассажиры» не обрабатываются при транспортировке по транзитной сети никаким образом (по сути, являются полем данных для транспортирующего протокола). Инкапсуляцию выполняет пограничное устройство (маршрутизатор или шлюз), которое находится на границе между исходной и транзитной сетями. Извлечение пакетов транспортируемого протокола из несущих пакетов выполняет второе пограничное устройство, расположенное на границе между транзитной сетью и сетью назначения. Пограничные устройства указывают в несущих пакетах свои адреса, а не адреса узлов в сети назначения.

Туннель может быть использован не только для создания VPN, но и когда две сети с одной транспортной технологией необходимо соединить через сеть, использующую другую транспортную технологию. При этом пограничные маршрутизаторы, которые подключают объединяемые сети к



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

транзитной, упаковывают пакеты транспортируемого протокола объединяемых сетей в пакеты транспортирующего протокола транзитной сети. Второй пограничный маршрутизатор выполняет обратную операцию. Например, с использованием туннелирования можно передавать трафик протоколов IPX, IPv6, AppleTalk через IPv4-сеть. В этом случае IPX, IPv6, AppleTalk будут являться транспортируемыми протоколами, а IPv4 – транспортирующим протоколом.

Туннелирование может использоваться на разных уровнях модели OSI, например, на канальном, или сетевом. Здесь будем рассматривать только туннелирование на сетевом уровне.

Одним из наиболее распространенных протоколов инкапсуляции, используемом для создания туннелей на сетевом уровне, является протокол GRE – Generic Routing Encapsulations. GRE – это протокол, разработанный компанией Cisco Systems, позволяющий инкапсулировать пакеты разного типа внутри IP-туннелей. Благодаря этому создается виртуальный канал «точка-точка» между маршрутизаторами Cisco поверх IP-сети. Необходимо отметить, что само по себе туннелирование не обеспечивает функционал VPN – данные транспортируемого протокола остаются незашифрованными и легко поддаются перехвату.

Инкапсуляция в соответствии с протоколом GRE в случае, если транспортируемым протоколом является IP, иллюстрируется рисунком 4.1.



Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»

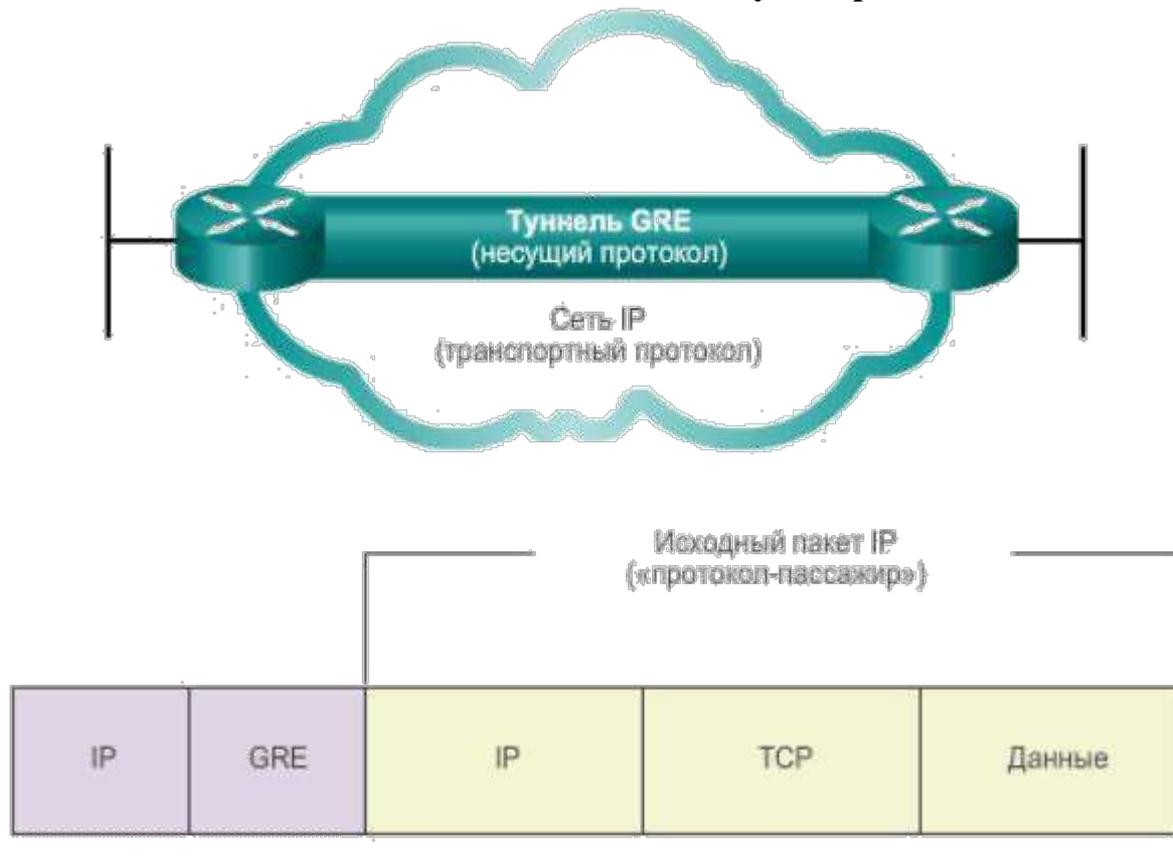


Рисунок 4.1 – Инкапсуляция с использованием GRE

Рассмотрим пример, заимствованный из [8].

Имеется общедоступная сеть, к которой выполнено подсоединение двух локальных сетей LAN1 и LAN2 с использованием маршрутизаторов Router_A и Router_B. Отметим, что в локальных сетях может использоваться адресация из диапазона частных немаршрутизируемых адресов, например, сеть, подключенная к Router_A, имеет адрес 192.168.1.0/24, а сеть, подключенная к Router_B – адрес 192.168.3.0/24, рисунок 4.2.

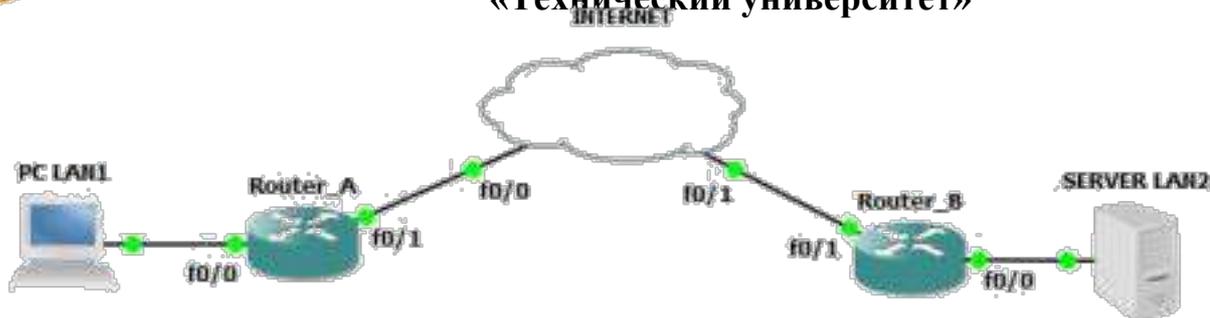


Рисунок 4.2 – Пример объединения двух частных сетей

Адресация в сети следующая:

Router_A:

f0/0 – 192.168.1.1/24

f0/1 – 11.11.11.11/8

Router_B:

f0/0 – 192.168.3.1/24

f0/1 – 33.33.33.33/8

Кроме реальных адресов, в рассматриваемой сети необходима адресация туннеля. Так как туннель представляет собой вырожденную сеть (точка-точка), можно использовать для него адрес 192.168.2.0/30.

Рассмотрим команды конфигурирования маршрутизатора Router_A:

```
Router_A(config)#interface fastEthernet0/0
```

```
Router_A(config-if)#ip address 192.168.1.1 255.255.255.0
```

```
Router_A(config-if)#no shutdown
```

```
Router_A(config-if)#exit
```

```
Router_A(config)#interface fastEthernet0/1
```

```
Router_A(config-if)#ip address 11.11.11.11 255.0.0.0
```

```
Router_A(config-if)#no shutdown
```

```
Router_A(config-if)#exit
```

Это были обычные команды конфигурирования интерфейсов маршрутизатора. Теперь необходимо сконфигурировать создаваемый туннель:



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Router_A(config)#interface Tunnel0 — вход в конфигурирование интерфейса Tunnel0;

Router_A(config-if)#ip address 192.168.2.1 255.255.255.252 — назначение IP-адреса для интерфейса Tunnel0;

Router_A(config-if)#tunnel source 11.11.11.11 — указание на IP-адрес интерфейса, являющегося источником туннеля;

Router_A(config-if)#tunnel destination 33.33.33.33 — указание на IP-адрес окончания туннеля;

Router_A(config-if)#exit

Router_A(config)#ip route 192.168.3.0 255.255.255.0 Tunnel0 — добавление статического маршрута ко второй локальной сети LAN2;

Router_A(config)#ip route 0.0.0.0 0.0.0.0 11.11.11.12 — добавление статического маршрута в направлении внешнего интерфейса.

Команда **tunnel source 11.11.11.11** предполагает возможность в качестве адреса источника туннеля указать не только IP-адрес, но и указание на физический интерфейс:

Router_A(config-if)#tunnel source fa0/1

Более того, если в качестве симулятора использовать Cisco Packet Tracer, можно будет указать в качестве источника только физический интерфейс.

Команда **ip route 192.168.3.0 255.255.255.0 Tunnel0** в Cisco Packet Tracer также не может быть выполнена, вместо параметра **Tunnel0** необходимо использовать IP-адрес другого «конца» туннеля, в нашем случае 192.168.2.2.

Проведем аналогичную конфигурацию для маршрутизатора Router_B:

Router_B(config)#interface fastEthernet0/0

Router_B(config-if)#ip address 192.168.3.1 255.255.255.0

Router_B(config-if)#no shutdown



```
Router_B(config-if)#exit
Router_B(config)#interface fastEthernet0/1
Router_B(config-if)#ip address 33.33.33.33 255.0.0.0
Router_B(config-if)#no shutdown
Router_B(config-if)#exit
Router_B(config)#interface Tunnel0
Router_B(config-if)#ip address 192.168.2.2 255.255.255.252
Router_B(config-if)#tunnel source 33.33.33.33
Router_B(config-if)#tunnel destination 11.11.11.11
Router_B(config-if)#exit
Router_B(config)#ip route 192.168.1.0 255.255.255.0 Tunnel0
Router_B(config)#ip route 0.0.0.0 0.0.0.0 33.33.33.34
```

После успешного установления туннеля компьютеры из подсети 192.168.3.0/24 оказываются доступны из подсети 192.168.1.0/24, рисунок 4.3. Анализ содержимого пакета, передаваемого по туннелю, показан на рисунке 4.4.

Из рисунка 4.4 видно, что пакет с адресами источника и назначения 192.168.1.2 и 192.168.3.2 соответственно, вложен в пакет с адресами источника и назначения 11.11.11.11 и 33.33.33.33.

Интерфейс туннеля с номером 0 можно просмотреть на маршрутизаторе с использованием команды **show running-config**, рисунок 4.5.

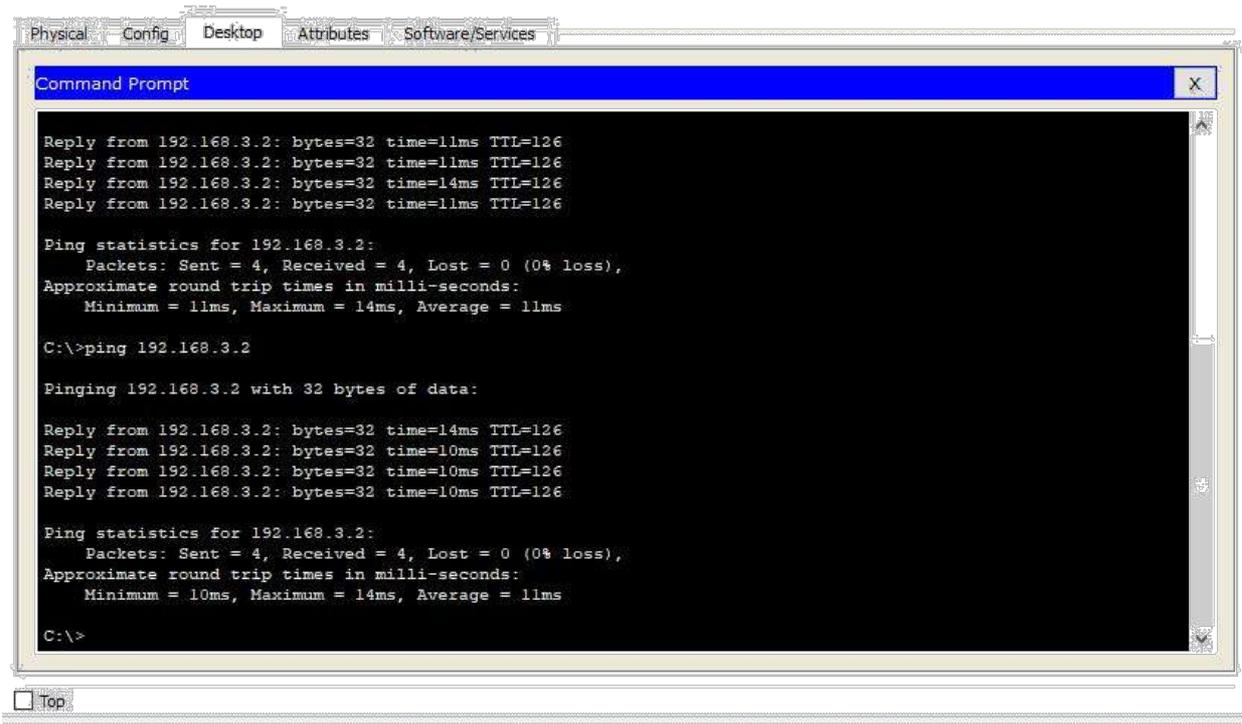


Рисунок 4.3 – Доступ ко второй локальной сети из первой с использованием туннеля

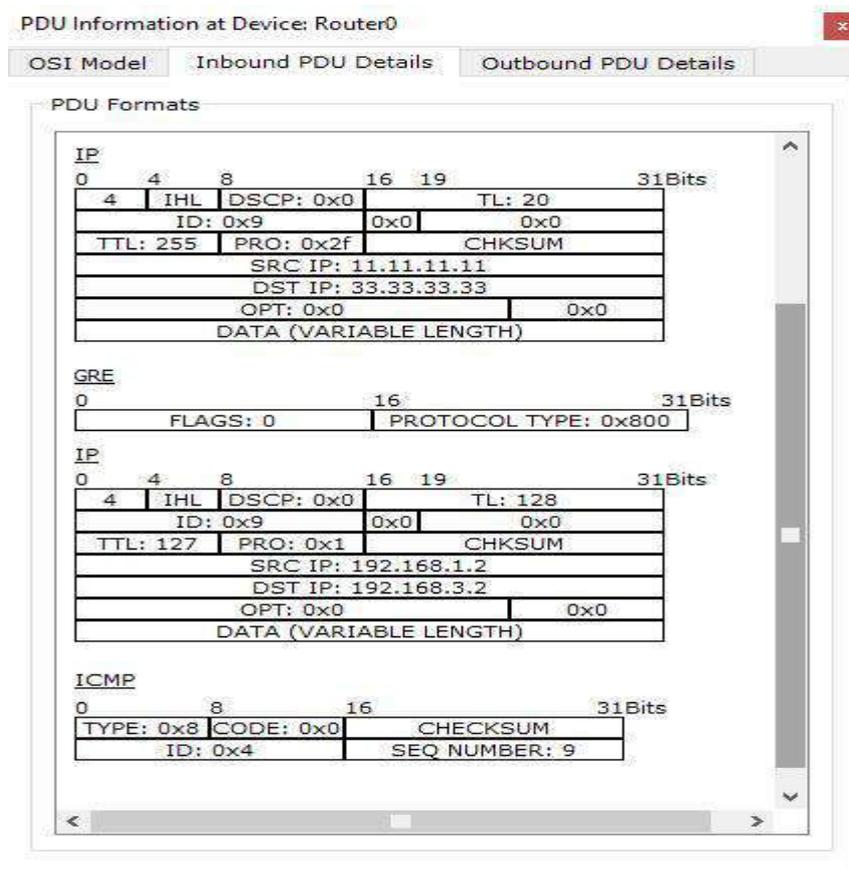


Рисунок 4.4 – Анализ содержимого пакета, передаваемого по туннелю



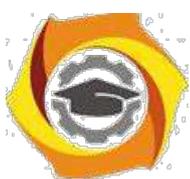
**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

```
Router1
Physical Config CLI Attributes
IOS Command Line Interface
!
interface Tunnel0
 ip address 192.168.2.1 255.255.255.252
 mtu 1476
 tunnel source FastEthernet0/1
 tunnel destination 33.33.33.33
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 11.11.11.11 255.0.0.0
 duplex auto
 speed auto
!
interface Vlan1
 no ip address
 shutdown
!
--More--
```

Рисунок 4.5 – Просмотр интерфейса tunnel0 на маршрутизаторе

Как указывалось выше, туннелирование само по себе не обеспечивает защиту передаваемых данных. Для этого используется семейство протоколов IP Security (IPSec).

IPSec — это комплект протоколов, касающихся вопросов шифрования, аутентификации и обеспечения защиты при транспортировке IP-пакетов; в его состав сейчас входят почти 20 предложений по стандартам и 18 RFC. Задача IPSec сводится к тому, чтобы выбрать конкретные алгоритмы и механизмы и настроить соответствующим образом устройства, участвующие в создании безопасного соединения. IPSec находит применение в организации VPN-соединений.



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

При создании защищенного канала участникам данного процесса необходимо произвести следующие действия:

33. Аутентифицировать друг друга.
34. Сгенерировать и обменяться ключами.
35. Договориться о том, с помощью каких протоколов шифровать данные.
36. Начать передавать данные в зашифрованный туннель.

IPsec, как уже было указано ранее, состоит из нескольких протоколов, каждый из которых отвечает за конкретную стадию установления IPsec-туннеля. Обобщенная архитектура IPsec представлена на рисунке 4.6.

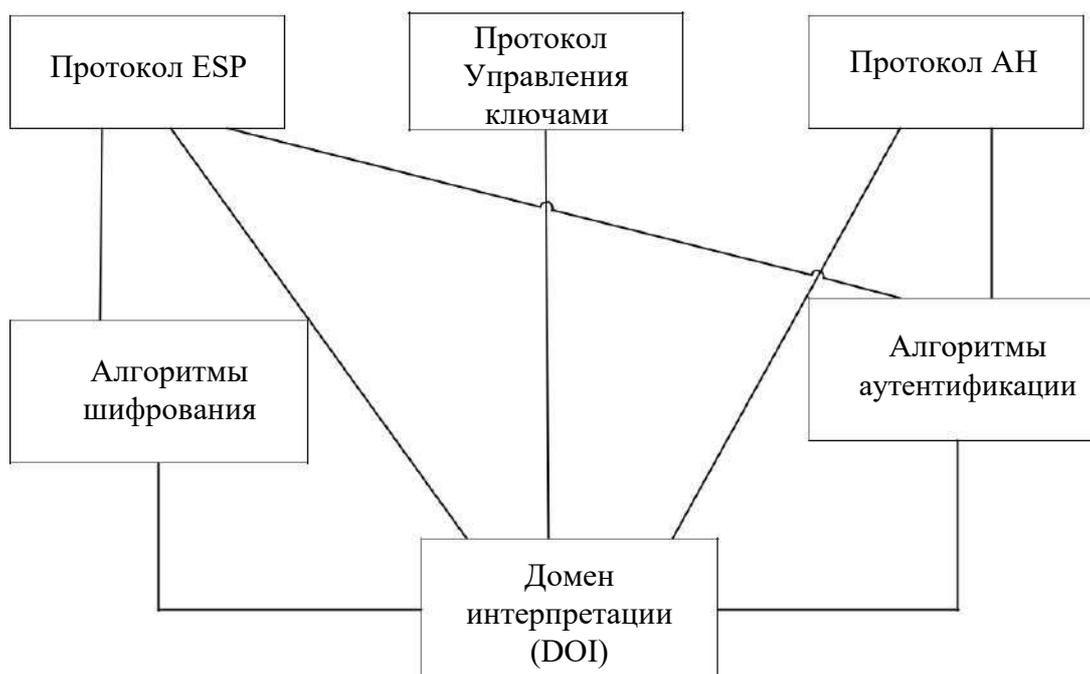


Рисунок 4.6 – Архитектура IPsec

Для управления ключами могут использоваться различные протоколы. Как известно, для шифрования и аутентификации необходимы ключи. Самым простым способом является ручная конфигурация ключей, однако такой способ плохо масштабируется. Поэтому на практике широко



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

используется IKE – Internet Key Exchange — протокол обмена ключами, который позволяет участникам динамически аутентифицировать друг друга, согласовывать использование SA (об SA речь пойдет ниже), генерировать ключи и обмениваться ими, используя алгоритм Диффи-Хеллмана. IKE использует ISAKMP (Internet Security Association Key Management Protocol – протокол управления ключами ассоциации безопасности Интернет, разработан Национальным агентством безопасности США - NSA). Протокол ISAKMP сам по себе не регламентирует какой-либо конкретный алгоритм обмена ключами, он содержит в себе описание ряда сообщений, которые позволяют согласовать использование алгоритмов обмена ключами. IKE пришел в 1998 г. на смену более ранним протоколам — ISAKMP/Oakley.

Протокол ISAKMP , описанный в документе RFC 2408, позволяет согласовывать алгоритмы и математические структуры (так называемые мультипликативные группы, определенные на конечном поле) для процедуры обмена ключами Диффи-Хеллмана, а также процессов аутентификации. Протокол Oakley, описанный в RFC 2412, основан на алгоритме Диффи — Хеллмана и служит для организации непосредственного обмена ключами.

Для выполнения аутентификации сторон в IKE применяются два основных способа.

Первый способ основан на использовании разделяемого ключа. Перед инициализацией IPSec-устройств, образующих безопасные ассоциации, в их базы данных помещается предварительно распределенный разделяемый ключ. Цифровая подпись на основе односторонней хэш-функции, например, MD5, использующей в качестве аргумента этот предварительно распределенный ключ, доказывает аутентичность противоположной стороны.



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Второй способ основан на использовании технологии цифровой подписи и цифровых сертификатов стандарта X.509. Каждая из сторон подписывает свой цифровой сертификат своим закрытым ключом и передает эти данные противоположной стороне. Если подписанный сертификат расшифровывается открытым ключом отправителя, то это удостоверяет тот факт, что отправитель, предоставивший данные, действительно обладает ответной частью данного открытого ключа — соответствующим закрытым ключом.

Однако следует отметить, что для удостоверения аутентичности стороны нужно еще убедиться в аутентичности самого сертификата, и для этого сертификат должен быть подписан не только его владельцем, но и некоторой третьей стороной, выдавшей сертификат и вызывающей доверие. В архитектуре IPSec эта третья сторона именуется органом сертификации CA (Certification Authority). Этот орган призван засвидетельствовать подлинность обеих сторон и должен пользоваться полным доверием сторон, а его открытый ключ — известен всем узлам, использующим его сертификаты для удостоверения личностей друг друга. Таким образом, IKE является комбинацией протоколов ISAKMP, Oakley и SKEME. Протокол ISAKMP описывает базовую технологию аутентификации, обмена ключами и согласования остальных параметров IPSec-туннеля при создании безопасного соединения, однако сами протоколы аутентификации сторон и обмена ключами в нем детально не определены. Поэтому при разработке протокола IKE общие правила и процедуры протокола ISAKMP дополнены процедурами аутентификации и обмена ключами, взятыми из протоколов Oakley и SKEME. Поскольку протокол IKE использует для управления ассоциациями алгоритмы и форматы протокола ISAKMP, названия этих протоколов иногда используют как синонимы.



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Следующий протокол на рисунке 4.6 – АН (Authentication Header – заголовок аутентификации) отвечает за аутентификацию источника и проверку целостности данных.

И, наконец, протокол ESP (Encapsulating Security Payload – безопасная инкапсуляция полезной нагрузки) занимается непосредственно шифрованием данных, а также может обеспечивать аутентификацию источника и проверку целостности данных.

Здесь необходимо ввести в рассмотрение термин SA – Security Association. SA в общем смысле представляет собой набор параметров защищенного соединения (например, алгоритм шифрования, ключ шифрования, и т.д.), который может использоваться обеими сторонами соединения. У каждого соединения есть ассоциированный с ним SA. При этом SA носит односторонний характер, то есть для взаимодействия двух объектов между ними должно быть установлено, как минимум, две SA. Стандарты IPSec позволяют шлюзам использовать как одну ассоциацию SA для передачи трафика всех взаимодействующих через общедоступную сеть хостов, так и создавать для этой цели произвольное число ассоциаций SA, например, по одной на каждое соединение TCP.

Для идентификации каждой SA предназначен индекс параметров безопасности SPI (Security Parameters Index). Этот индекс включается в заголовки защищенных IPSec-пакетов, чтобы принимающая сторона смогла правильно их расшифровать и аутентифицировать, воспользовавшись указанной безопасной ассоциацией.

Рассмотрим, каким образом организуется защищенное соединение между участниками информационного обмена (например, между двумя пограничными маршрутизаторами локальных сетей, в терминах IPSec они называются шлюзами безопасности – Security Gateway).



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

29. Участникам надо договориться, какие алгоритмы/механизмы защиты они будут использовать для своего защищенного соединения, для чего используется протокол IKE. Этот процесс состоит из двух фаз:

1а) участники аутентифицируют друг друга и договариваются о параметрах установки вспомогательного соединения (тоже защищенного),

предназначенного только для обмена информацией о желаемых/поддерживаемых алгоритмах шифрования и прочих деталях будущего IPSec-туннеля. Такой вспомогательный туннель называется ISAKMP-туннелем. Таким образом, первая фаза служит для создания вспомогательного защищенного ISAKMP-туннеля, через который будут передаваться параметры для будущего IPSec-туннеля.

1б) уже доверяющие друг другу участники договариваются о том, как строить основной туннель для передачи данных. Они по очереди предлагают друг другу варианты и, если приходят к согласию, устанавливают основной туннель (IPSec-туннель).

21. Участники получили IPSec-туннель с параметрами, которые устраивают их обоих, и направляют туда потоки данных, подлежащие шифрованию.

22. Периодически, в соответствии с настроенным временным интервалом

(Lifetime), обновляются ключи шифрования для основного туннеля. Для этого участники вновь связываются по ISAKMP-туннелю, проходят вторую фазу и устанавливают новые SA.

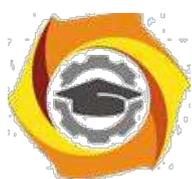
IPSec предлагает различные методы защиты трафика. В каждом узле, поддерживающем IPSec, используются базы данных (БД) двух типов:

база данных безопасных ассоциаций SAD (Security Associations Database);

база данных политики

безопасности SPD (Security Policy Database).

При установлении SA две вступающие в обмен стороны принимают ряд соглашений, регламентирующих процесс передачи потока данных



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

между ними. Соглашения представляются в виде набора параметров. Для SA такими параметрами являются, в частности, тип и режим работы протокола защиты (AH или ESP), методы шифрования, секретные ключи, значение текущего номера пакета в ассоциации и другая информация.

Объединение служебной информации в рамках SA предоставляет пользователю возможность сформировать разные классы защиты, предназначенные, например, для электронного общения с разными «собеседниками». Другими словами, применение структур SA открывает путь к построению множества виртуальных частных сетей, различающихся своими параметрами.

Наборы текущих параметров, определяющих все активные ассоциации, хранятся на обоих оконечных узлах защищенного канала в виде SAD. Каждый узел IPSec поддерживает две базы SAD — одну для исходящих ассоциаций, другую — для входящих.

SPD задает соответствие между IP-пакетами и установленными для них правилами обработки. При обработке пакетов БД SPD используются совместно с БД SAD. SPD представляет собой упорядоченный набор правил, каждое из которых включает совокупность фильтров и допустимых политик безопасности. Фильтры служат для отбора пакетов, а политики безопасности задают требуемую обработку. Такая БД формируется и поддерживается на каждом узле, где реализуется IPSec.

Необходимо отметить, что первая фаза 1а) может проходить в одном из двух режимов – основном (main mode) и агрессивном (aggressive mode).

ООО основном режиме (main mode) первая фаза состоит из следующих этапов:

13. Обмен параметрами безопасности будущего ISAKMP-туннеля (алгоритмы шифрования, методы аутентификации, способ обмена секретными ключами, срок жизни SA);

31. Генерация каждым из участников общего секретного ключа;



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

27. Обмен общими секретными ключами с использованием алгоритма Диффи-Хеллмана;

28. Взаимная аутентификация участников.

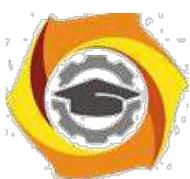
агрессивном режиме (aggressive mode) в первый пакет сразу помещается вся необходимая информация для установления ISAKMP-туннеля. Получатель посылает в ответ все, что необходимо для завершения обмена, после чего первому узлу необходимо лишь подтвердить соединение.

Агрессивный режим быстрее позволяет установить туннель, но при этом он менее безопасный, потому что стороны обмениваются информацией до того как безопасное соединение установлено.

АН (Authentication Header) — протокол IPSec, предназначенный для аутентификации отправителя, контроля целостности данных и, опционально, для предотвращения атак в виде повторной посылки пакета (reply). По сути это обычный опциональный заголовок, располагающийся между основным заголовком IP-пакета и полем данных.

АН (как и рассматриваемый ниже ESP) может работать в одном из двух режимов – транспортном и туннельном. В первом случае (транспортный режим) механизмы безопасности применяются только для транспортного уровня и выше. Соответственно, заголовок сетевого уровня (IP) остается без защиты, более того, он остается таким же, как и был в исходном пакете, за исключением изменения некоторых полей. Например, меняется поле Next Header, указывающее на то, заголовок какого протокола следует за IP-заголовком.

к туннельном режиме обеспечивается защита также и данных сетевого уровня. Это обеспечивается путем добавления нового IP-заголовка. После определения SA истинные адреса хостов отправления и назначения (и другие служебные поля) полностью защищаются от



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

модификаций, а в новый заголовок выставляются адреса и другие данные для шлюзов отправления/получения.

На рисунке 4.7 показаны исходный пакет, пакеты после обработки протоколом АН в транспортном и туннельном режимах.

IP-заголовок	TCP/UDP-заголовок	Данные
--------------	-------------------	--------

а) исходный пакет

IP-заголовок	АН-заголовок	TCP/UDP-заголовок	Данные
--------------	--------------	-------------------	--------

б) пакет после обработки в транспортном режиме

Новый IP-заголовок	АН-заголовок	IP-заголовок	TCP/UDP-заголовок	Данные
-----------------------	--------------	--------------	-------------------	--------

в) пакет после обработки в туннельном режиме

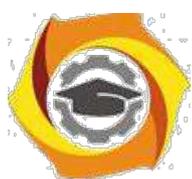
Рисунок 4.7 – Форматы пакетов

Формат АН-заголовка показан на рисунке 4.8.

0	8	16	31
Next Header	Payload Len	Резерв	
Security Parameters Index (SPI)			
Sequence Number (SN)			
Authentication Data			

Рисунок 4.8 – Формат заголовка АН

Первые 8 бит заголовка (поле Next Header) содержат номер, соответствующий протоколу следующего уровня. Номер для каждого протокола назначает организация IANA (Internet Assigned Numbers Authority). Например, номер TCP - 6, ESP - 50, АН - 51 и т.д.



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Поле Payload Len указывает длину заголовка АН в 32-битных словах. Необходимость этого поля связана с тем, что поле Authentication Data имеет переменную длину, соответственно, и сам АН-заголовок имеет переменную длину.

SPI – это 32-битный индикатор, который однозначно идентифицирует ту SA, в рамках которой передается данный пакет (об SPI уже шла речь выше).

Поле Sequence Number было введено в RFC 2402. Значение счетчика, содержащееся в этом поле, может использоваться для защиты от атак путем повторных посылок перехваченных пакетов. Если функция защиты от повторов активирована (а это указывается в SA), отправитель последовательно наращивает значение поля для каждого пакета, передаваемого в рамках данной SA.

Сама аутентификация обеспечивается передачей данных, содержащихся в поле Authentication Data. Аутентификационные данные представляют собой хэш-функцию, вычисленную на основе содержимого пакета с использованием алгоритмов MD5 или SHA-1. Симметричный секретный ключ шифрования устанавливается вручную или по протоколу IKE.

Аутентификация производится путем создания так называемой имитовставки (MAC), для чего используется хэш-функция и секретный ключ. Во всех реализациях АН обязательно должно поддерживаться использование хэш-функций с ключом HMAC-MD5-96 (используется по умолчанию) и HMAC-SHA-1-96, представляющих собой варианты хэш-функций MD5 и SHA-1, соответственно. Но могут использоваться и другие алгоритмы хеширования. Полученное значение, называемое в описании протокола ICV (Integrity Check Value - значение контроля целостности) помещается в поле Authentication Data. Это поле переменной длины, т.к. разные алгоритмы хеширования формируют разные по длине дайджесты.



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

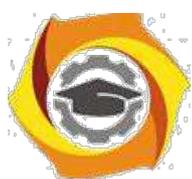
При использовании АН в транспортном режиме, ICV рассчитывается для TCP/UDP-заголовка, данных и неизменяемых полей IP-заголовка. Изменяемые поля, такие например как поле TTL, при расчете значения хэш-функции принимаются равными 0. В туннельном режиме хэшируется весь исходный IP-пакет и неизменяемые поля нового заголовка.

ESP (Encapsulation Security Payload) — протокол IPSec, предназначенный для шифрования данных. ESP предоставляет три вида сервисов безопасности:

30. обеспечение конфиденциальности (шифрование содержимого IP-пакетов,
7. также частичная защита от анализа трафика путем применения туннельного режима);
8. обеспечение целостности IP-пакетов и аутентификации источника данных;
9. обеспечение защиты от воспроизведения IP-пакетов.

Как видно, функциональность ESP шире, чем АН (добавляется шифрование). Кроме того, ESP не обязательно предоставляет все сервисы, но либо конфиденциальность, либо аутентификация должны быть задействованы.

Так же, как и АН, ESP может работать в транспортном и туннельном режимах. Исходный пакет, пакеты после обработки протоколом ESP в транспортном и туннельном режимах показаны на рисунке 4.9. На рисунке серым фоном отмечены поля пакета, подлежащие шифрованию.



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**



а) исходный пакет



б) пакет после обработки в транспортном режиме

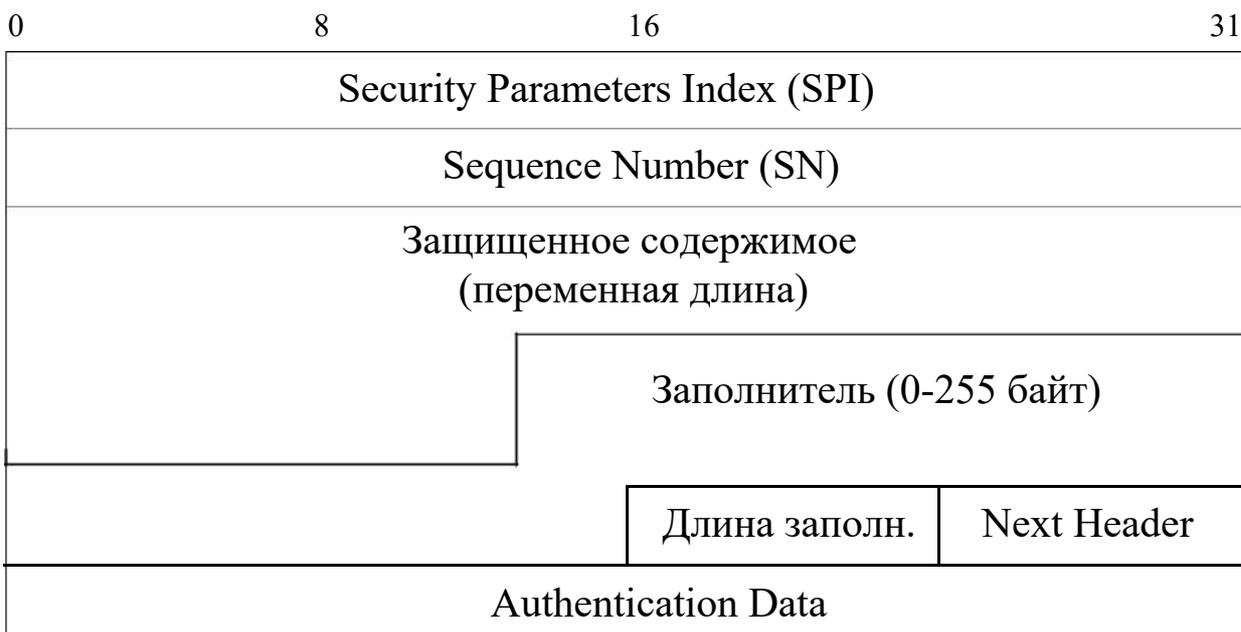


в) пакет после обработки в туннельном режиме

Рисунок 4.9 – Форматы пакетов

Так как ESP поддерживает и аутентификацию, и шифрование, для аутентификации может использоваться имитовставка с использованием хэш-функций с ключом HMAC-MD5-96 (по умолчанию) и HMAC-SHA-1-96 (как и у AH), но могут использоваться и другие алгоритмы. Для шифрования могут использоваться алгоритмы DES, 3DES, AES, и т.д.

Формат заголовка ESP выглядит в соответствии с рисунком 4.10. Это не столько заголовок, сколько обертка (инкапсулирующая оболочка, как видно из рисунка 4.10) для зашифрованного содержимого. Например, ссылку на следующий заголовок нельзя выносить в начало, в незашифрованную часть, так как она лишится конфиденциальности.





**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Рисунок 4.10 – Формат заголовка ESP

Заголовок ESP начинается с двух 32-разрядных значений - SPI и SN. Роль их такая же, как в протоколе AH - SPI идентифицирует SA, использующийся для создания данного туннеля; SN позволяет защититься от повторов пакетов. SN и SPI не шифруются.

Следующим идет поле, содержащее зашифрованные данные. После них следует поле заполнителя, который нужен для того, чтобы выровнять длину шифруемых полей до значения, кратного размеру блока алгоритма шифрования.

После заполнителя идут поля, содержащие значение длины заполнителя и указание на номер, соответствующий протоколу следующего уровня. Четыре перечисленных поля (данные, заполнитель, длина, следующий протокол) защищаются шифрованием.

Если ESP используется и для аутентификации данных, то завершает пакет поле переменной длины, содержащее ICV.

Применение протокола ESP к исходящим пакетам можно представлять себе следующим образом. Из пакета извлекается его часть – остаток, то есть та часть, которая подлежит шифрованию (закрашенные поля на рисунке 4.9). Далее следуют этапы:

13. остаток пакета копируется в буфер;

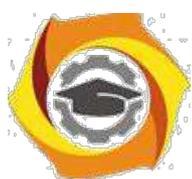
14. к остатку приписываются дополняющие байты, их число и номер (тип)

первого заголовка остатка, с тем, чтобы номер был прижат к границе 32-битного слова, а размер буфера удовлетворял требованиям алгоритма шифрования;

7. текущее содержимое буфера шифруется;

8. в начало буфера приписываются поля "Индекс параметров безопасности

(SPI)" и "Порядковый номер (SN)" с соответствующими значениями;



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

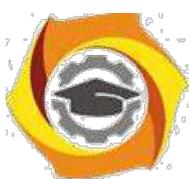
8. пополненное содержимое буфера аутентифицируется, в его конец помещается поле "Аутентификационные данные";

9. в новый пакет переписываются начальные заголовки старого пакета и конечное содержимое буфера.

Если в ESP включены и шифрование, и аутентификация, то аутентифицируется зашифрованный пакет. Для входящих пакетов действия выполняются в обратном порядке, т. е. сначала производится аутентификация. Это позволяет не тратить ресурсы на расшифровку поддельных пакетов, что в какой-то степени защищает от атак на доступность.

Из сравнения протоколов АН и ESP можно сделать следующие выводы. Если необходимо знать, что данные из идентифицированного источника передаются без нарушения целостности, а их конфиденциальность обеспечивать не требуется, можно использовать протокол АН, который защищает протоколы высших уровней и поля заголовка IP, не изменяемые в пути. Защита означает, что соответствующие значения нельзя изменить, потому что это будет обнаружено второй стороной IPSec, и любая модифицированная дейтаграмма IP будет отвергнута. Протокол АН не обеспечивает защиту от прослушивания канала и просмотра нарушителем заголовка и данных. Но поскольку заголовок и данные незаметно изменить нельзя, измененные пакеты отвергаются.

Если необходимо сохранить данные в тайне (обеспечить конфиденциальность), необходимо использовать ESP. Данный протокол предполагает шифрование протоколов высших уровней в транспортном режиме и всей исходной дейтаграммы IP в туннельном режиме, так что извлечь информацию о пакетах путем прослушивания канала передачи невозможно. Протокол ESP может также обеспечить для пакетов сервис аутентификации. Однако при использовании ESP в транспортном режиме



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

внешний оригинальный заголовок IP не защищается, а в туннельном режиме не защищается новый заголовок IP.

Таким образом, при необходимости обеспечения конфиденциальности и целостности передаваемых данных с одновременной аутентификацией их источника необходимо использовать протокол ESP в туннельном режиме. Более того, при создании ISAKMP-туннеля необходимо использовать основной режим (main mode).

При использовании оборудования Cisco для создания VPN можно либо использовать рассмотренное выше туннелирование, и отправлять по туннелю зашифрованный трафик, либо не использовать туннельные интерфейсы.

Рассмотрим сначала конфигурирование VPN с туннелированием. Часто такой режим называют IPSec-over-GRE. Будем использовать тот же пример, рисунок 4.11.

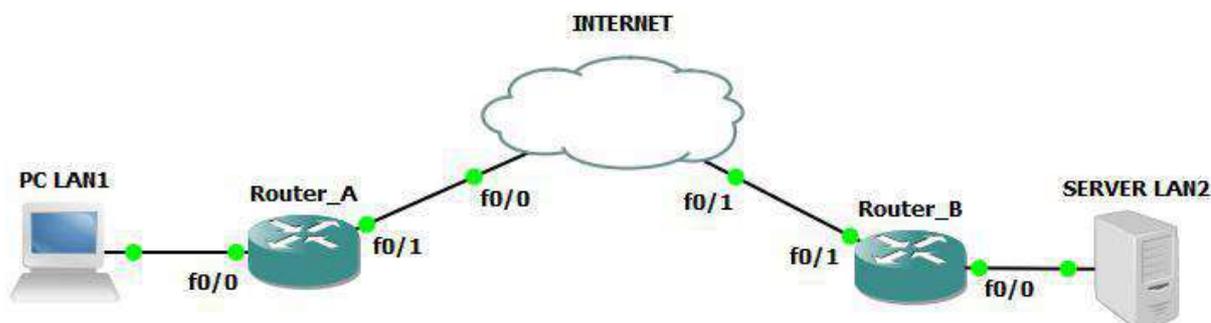


Рисунок 4.11 – Пример объединения двух частных сетей

Адресация в сети следующая:

Router_A:

f0/0 – 192.168.1.1/24

f0/1 – 11.11.11.11/8

Router_B:

f0/0 – 192.168.3.1/24

f0/1 – 33.33.33.33/8

Адрес туннеля – 192.168.2.0/30.



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

На первом этапе необходимо создать политику с номером 1 которая определит алгоритм и используемые протоколы при обмене ключами (IKE Фаза 1). Номер политики определяет ее приоритет, так как может быть создано несколько политик (чем меньше число, тем выше приоритет). Часто задаются несколько таких политик с различными комбинациями шифрования, хеша и номера группы Диффи-Хеллмана (DH). При создании `isakmp sa`, та сторона, которая инициирует соединение, отправляет все локально настроенные политики. Принимающая сторона просматривает по очереди, в порядке приоритетности свои локально настроенные политики. Первая же политика, для которой найдено совпадение, будет использоваться. Команда создания политики:

RouterA(config)#crypto isakmp policy 1

10. соответствии с логикой Cisco IOS, если мы создаем что-либо, мы сразу попадаем в конфигурирование того, что мы создали. Поэтому после создания политики 1 мы попадаем в режим ее конфигурирования.

Команды конфигурирования:

RouterA(config-isakmp)#encryption <алгоритм шифрования>

RouterA(config-isakmp)#hash <алгоритм хэширования>

RouterA(config-isakmp)#authentication <метод аутентификации>

RouterA(config-isakmp)#lifetime <время жизни ISAKMP-туннеля>

7. качестве алгоритмов шифрования могут быть указаны **des**, **3des**, **aes**. В качестве алгоритма хэширования может быть использован **sha** или **md5**. В качестве метода аутентификации может быть указан метод с открытым ключом **pre-share**, либо более сложные методы, например, аутентификация с использованием цифровой подписи **RSA Digital Signatures** **rsa-sig**.

При использовании **pre-share** может также быть задан номер группы Диффи-Хеллмана (DH):

RouterA (config-isakmp)# group 2



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Группа означает конкретный алгоритм ДН [9]:

Алгоритм ДН-1 - ключ 768 бит

Алгоритм ДН-2 - ключ 1024 бит

Алгоритм ДН-5 - ключ 1536 бит

Также в случае использования открытого ключа необходимо указать сам ключ (он должен быть одинаковым на обоих маршрутизаторах):

RouterA(config)#crypto isakmp key <№> <ключ> address <адрес соседа>

Далее указываются возможные параметры создаваемого SA (политика Фазы 2). Так как это набор протоколов, предлагаемых встречной стороне, он называется **transform-set**. Дальнейшее конфигурирование удачно иллюстрируется рисунком 4.12, заимствованным из [10].

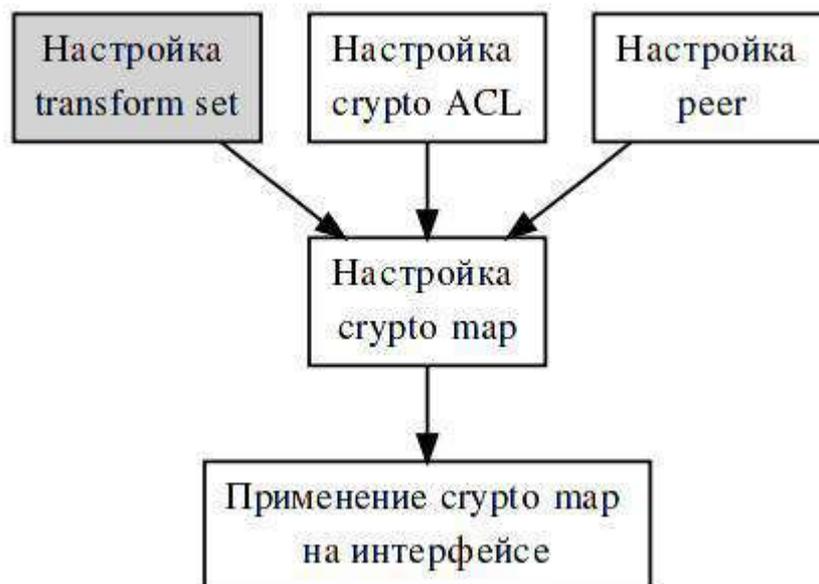
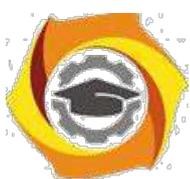


Рисунок 4.12 – Принцип конфигурирования transform-set

Из рисунка 4.12 видно, что сначала необходимо создать transform-set 9. указанием используемых протоколов (AH или ESP), способов аутентификации и шифрования. С использованием ACL фильтруется трафик, подлежащий шифрованию, с использованием peer указывается «сосед», с которым устанавливается туннель. Криптокарта (crypto map) представляет собой указание деталей шифрования – адрес «соседа» по



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

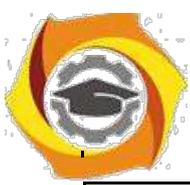
туннелю, используемый ACL, используемый transform-set.
Заключительным этапом является привязка созданной криптокарты на конкретном интерфейсе.

**RouterA(config)# crypto ipsec transform-set <имя> <аутентификация>
<шифрование>**

ESP и АН могут быть использованы одновременно, а могут и поодиночке. В таблице 4.1 представлены каждый из возможных вариантов.

Таблица 4.1 – Варианты использования протоколов для АН и ESP

Тип преобразования	Синтаксис	Описание
АН Transform (один из списка)	ah-md5-hmac	Протокол АН с алгоритмом аутентификации MD5
	ah-sha-hmac	Протокол АН с алгоритмом аутентификации SHA
	ah-gost-28147-mac	Протокол АН с алгоритмом ГОСТ 28147-89 (в режиме выработки имитовставки)
	ah-gost3411-hmac	Протокол АН с алгоритмом ГОСТ Р 34.11-94
ESP Encryption Transform (один из списка)	esp-null	Протокол ESP с алгоритмом Null.
	esp-des	Протокол ESP с 56-битным алгоритмом DES
	esp-3des	Протокол ESP с 168-битным алгоритмом 3DES
	esp-aes-128	Протокол ESP с 128-битным алгоритмом AES
	esp-aes-192	Протокол ESP с 192-битным алгоритмом AES
	esp-aes-256	Протокол ESP с 256-битным алгоритмом AES
	esp-gost28147	Протокол ESP с алгоритмом ГОСТ 28147-89 (в режиме простой замены с зацеплением)
	esp-gost28147-4m-imit	Протокол ESP с алгоритмом ГОСТ 28147-89 (в комбинированном режиме: гаммирование и вычисление имитовставки. в соответствии со спецификацией ESP_GOST-4M-IMIT)
ESP Authentication Transform (один из списка)	esp-md5-hmac	Протокол ESP с алгоритмом аутентификации MD5
	esp-sha-hmac	Протокол ESP с алгоритмом аутентификации SHA



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

	esp-gost28147-mac	Протокол ESP с алгоритмом ГОСТ 28147-89 (в режиме выработки имитовставки)
	esp-gost3411-hmac	Протокол ESP с алгоритмом ГОСТ Р 34.11-94

Таким образом, если мы предполагаем использование только ESP с алгоритмом шифрования 3DES и аутентификацией с хэш-функцией MD5, команда принимает вид:

```
RouterA(config)#crypto ipsec transform-set LAN1 esp-3des esp-md5-hmac
```

Далее с использованием команд **mode tunnel** и **mode transport** можно включить либо туннельный, либо транспортный режимы.

Далее необходимо сформировать так называемую криптокарту (crypto map), в которой указываются детали шифрования:

```
RouterA(config)#crypto map <имя> <№> ipsec-isakmp
```

```
RouterA(config-crypto-map)#set peer <адрес соседа>
```

```
RouterA(config-crypto-map)#set transform-set <имя transform-set >
```

```
RouterA(config-crypto-map)#match address <список доступа>
```

8. первой команде № - это номер набора шифрования для создаваемого туннеля. Дело в том, что в маршрутизаторе может существовать только одна криптокарта. Однако в самой карте можно создавать наборы шифрования для нескольких туннелей. Например, мы создаем второй туннель. Тогда в самой карте поменяем порядковый номер:

```
RouterA (config)# crypto map MAP <другой №> ipsec-isakmp
```

После этой команды можно указывать параметры для туннеля с номером **другой №**.

Также необходимо настроить список доступа (crypto ACL на рисунке 4.12). В данном случае шифровать необходимо те пакеты, которые передаются в туннель, то есть имеют GRE-заголовок. Поэтому создадим расширенный список доступа:



**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

```
RouterA(config)#access-list 100 permit gre host 11.11.11.11 host 33.33.33.33
```

Теперь необходимо перейти в режим конфигурирования внешнего интерфейса (у нас это fa0/1) и привязать к нему созданную криптокарту:

```
RouterA(config)#interface fa 0/1 RouterA(config-  
if)#crypto map <имя кпиптокарты>
```

Аналогичные настройки необходимо произвести на втором шлюзе безопасности (у нас это RouterB):

```
RouterB(config)#crypto isakmp policy 1
```

```
RouterB(config-isakmp)#encryption 3des
```

```
RouterB(config-isakmp)#hash md5
```

```
RouterB(config-isakmp)#authentication pre-share
```

```
RouterA (config-isakmp)# group 2
```

```
RouterB(config)#exit
```

```
RouterB(config)#crypto isakmp key 0 PASS address 11.11.11.11
```

```
RouterB(config)#crypto ipsec transform-set LAN2 esp-3des esp-md5-hmac
```

```
RouterB(cfg-crypto-trans)#mode tunnel RouterB(cfg-crypto-trans)#exit
```

```
RouterB(config)#crypto map MAP2 10 ipsec-isakmp
```

```
RouterB(config-crypto-map)#set peer 11.11.11.11
```

```
RouterB(config-crypto-map)#set transform-set
```

```
LAN2 RouterB(config-crypto-map)#match address
```

```
100 RouterB(config-crypto-map)#exit
```

```
RouterB(config)#access-list 100 permit gre host 33.33.33.33 host 11.11.11.11
```

```
RouterB(config)#interface fa 0/1
```

```
RouterB(config-if)#crypto map MAP2
```

Настроенные параметры на Router_A можно посмотреть с использованием команды **show running-config**, рисунок 4.13.



```
crypto isakmp policy 1
  encr 3des
  hash md5
  authentication pre-share
  lifetime 10000
crypto isakmp key PASS address 33.33.33.33
!
!
crypto ipsec transform-set FILIAL esp-3des esp-md5-hmac
!
crypto map MAP 10 ipsec-isakmp
  set peer 33.33.33.33
  set transform-set FILIAL
  match address 100
!
!
!
!
interface Tunnel0
  ip address 192.168.2.1 255.255.255.252
  tunnel source 11.11.11.11
  tunnel destination 33.33.33.33
!
interface FastEthernet0/0
  ip address 192.168.1.1 255.255.255.0
  duplex auto
  speed auto
!
interface FastEthernet0/1
  ip address 11.11.11.11 255.0.0.0
  duplex auto
  speed auto
  crypto map MAP
```

Рисунок 4.13 – Просмотр конфигурации

Для просмотра параметров шифрования можно использовать ряд команд:

- show crypto isakmp policy** – просмотр параметров туннеля в первой фазе;
- show crypto map** – просмотр созданных карт шифрования;
- show crypto isakmp sa** – просмотр созданных SA (фаза 1);
- show crypto ipsec sa** – просмотр созданных SA (фаза 2).

Выполнение этих команд, выполненных с использованием GNS3, иллюстрируется рисунками 4.14 – 4.17.



```
R2#show crypto isakmp policy

Global IKE policy
Protection suite of priority 1
  encryption algorithm: Three key triple DES
  hash algorithm:      Message Digest 5
  authentication method: Pre-Shared Key
  Diffie-Hellman group: #1 (768 bit)
  lifetime:           10000 seconds, no volume limit
Default protection suite
  encryption algorithm: DES - Data Encryption Standard (56 bit keys).
  hash algorithm:      Secure Hash Standard
  authentication method: Rivest-Shamir-Adleman Signature
  Diffie-Hellman group: #1 (768 bit)
  lifetime:           86400 seconds, no volume limit
R2#
```

Рисунок 4.14 – Результат выполнения команды **show crypto isakmp policy**

Из рисунка видно, что на маршрутизаторе имеются две политики – одна созданная нами (1), и вторая – по умолчанию.

```
R2#show crypto map
Crypto Map "MAP" 10 ipsec-isakmp
Peer = 33.33.33.33
Extended IP access list 100
  access-list 100 permit gre host 11.11.11.11 host 33.33.33.33
Current peer: 33.33.33.33
Security association lifetime: 4608000 kilobytes/3600 seconds
PFS (Y/N): N
Transform sets={
  FILIAL,
}
Interfaces using crypto map MAP:
  FastEthernet0/1
R2#
```

Рисунок 4.15 – Результат выполнения команды **show crypto map**

```
R2#show crypto isakmp sa
dst          src          state          conn-id slot status
33.33.33.33  11.11.11.11  QM_IDLE       1      0 ACTIVE
R2#
```

Рисунок 4.16 – Результат выполнения команды **show crypto isakmp sa**

Если в графе state указано QM_IDLE, значит первая фаза прошла успешно, в противном случае отобразится сообщение MM_NO_STATE. Также SA, созданные на первой фазе, можно посмотреть более подробно 5. использованием команды **show crypto isakmp sa detail**.



```
R2#show crypto ipsec sa

interface: FastEthernet0/1
  Crypto map tag: MAP, local addr 11.11.11.11

protected vrf: (none)
local ident (addr/mask/prot/port): (11.11.11.11/255.255.255.255/47/0)
remote ident (addr/mask/prot/port): (33.33.33.33/255.255.255.255/47/0)
current_peer 33.33.33.33 port 500
  PERMIT, flags={origin_is_acl,}
#pkts encaps: 5, #pkts encrypt: 5, #pkts digest: 5
#pkts decaps: 5, #pkts decrypt: 5, #pkts verify: 5
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 10, #recv errors 0

local crypto endpt.: 11.11.11.11, remote crypto endpt.: 33.33.33.33
path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/1
current outbound spi: 0xD0053F69(3490004841)

inbound esp sas:
  spi: 0x4345F484(1128658052)
    transform: esp-3des esp-md5-hmac ,
    in use settings = {Tunnel, }
    conn id: 2001, flow_id: SW:1, crypto map: MAP
    sa timing: remaining key lifetime (k/sec): (4596639/950)
    IV size: 8 bytes
    replay detection support: Y
    Status: ACTIVE

inbound ah sas:

inbound pcp sas:

outbound esp sas:
  spi: 0xD0053F69(3490004841)
    transform: esp-3des esp-md5-hmac ,
    in use settings = {Tunnel, }
    conn id: 2002, flow_id: SW:2, crypto map: MAP
    sa timing: remaining key lifetime (k/sec): (4596639/932)
    IV size: 8 bytes
    replay detection support: Y
    Status: ACTIVE

outbound ah sas:

outbound pcp sas:
```

Рисунок 4.17 – Результат выполнения команды **show crypto ipsec sa**

Количество зашифрованных и расшифрованных пакетов можно посмотреть с использованием команды **show crypto engine connections active**, рисунок 4.18.



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

```
R2#show crypto engine connections active
```

ID	Interface	IP-Address	State	Algorithm	Encrypt	Decrypt
1	FastEthernet0/1	11.11.11.11	set	HMAC_MD5+3DES_56_C	0	0
2001	FastEthernet0/1	11.11.11.11	set	3DES+MD5	0	5
2002	FastEthernet0/1	11.11.11.11	set	3DES+MD5	5	0

```
R2#
```

Рисунок 4.18 – Результат выполнения команды **show crypto engine connections active**

Второй способ позволяет настроить VPN без туннеля GRE, туннелирование производится самим IPSec. Для этого в списке доступа необходимо указать не протокол GRE, а в явном виде указать сети источника и получателя трафика:

RouterA(config)#

Ip access-list extended ACL_CRYPTO

permit ip 192.168.1.0 0.0.0.255 192.168.4.0 0.0.0.255

Кроме того, при настройке IPSec-туннеля можно использовать профили.

4.5 Отчет по работе

Сконфигурированный VPN-туннель на маршрутизаторах Cisco.

Методы отладки и тестирования программных продуктов

Методическое пособие

для проведения лабораторных работ

Лабораторная работа № 1. Интегрированная среда разработки и отладки приложений LAZARUS

Цель лабораторной работы

Освоить работу с интегрированной средой разработки и отладки приложений LAZARUS, изучить методы создания интерактивных приложений.

В Краткие теоретические сведения

Пользовательский интерфейс Lazarus. Интегрированная среда разработки Lazarus представляет собой многооконную систему, вид (пользовательский интерфейс) которой может различаться в зависимости от настроек. После загрузки интерфейс Lazarus представлен на рис.1.1.



Негосударственное частное образовательное учреждение высшего образования «Технический университет»

К основным окнам среды программирования Lazarus относятся:

16. главное окно;
17. окно Инспектора объектов;
18. окно Формы, или Конструктора формы;
19. окно Редактора кода;
20. окно Сообщений.

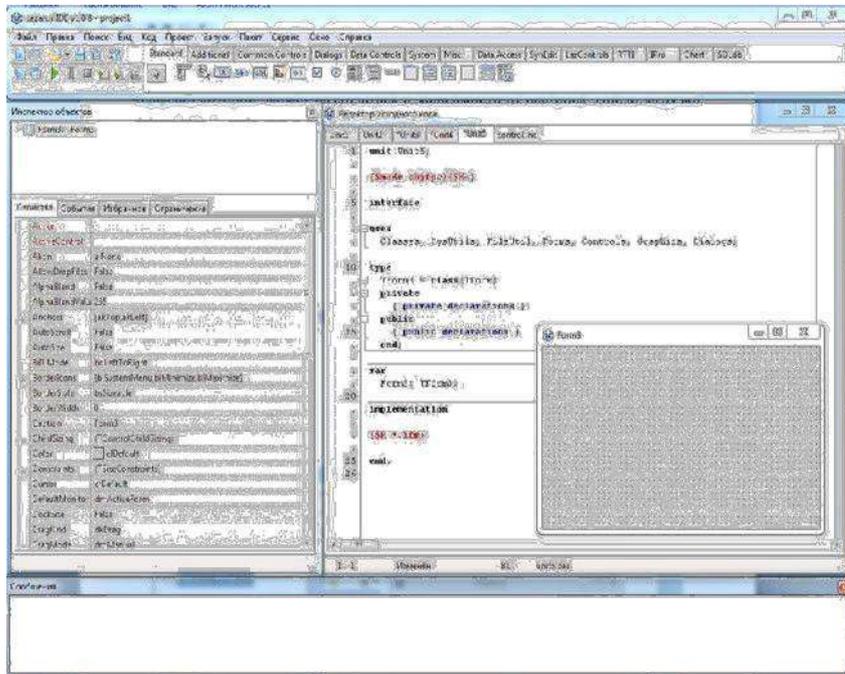
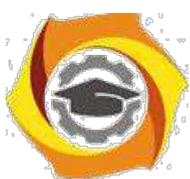


Рис. 1.1. Основные окна среды программирования Lazarus
Главное окно Lazarus содержит:

15. главное меню;
16. палитру компонентов.



Главное меню содержит обширный набор команд для доступа к функциям Lazarus.

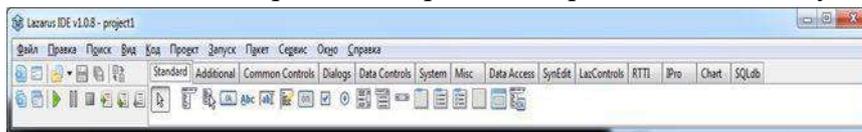


Рис. 1.2. Главное меню среды программирования Lazarus

Панели инструментов находятся под главным меню в левой части главного окна и содержат кнопки быстрого доступа к наиболее частым командам главного меню.

Палитра компонентов находится под главным меню в правой части главного окна и содержит множество компонентов, размещаемых в создаваемых формах. Компоненты являются своего рода строительными блоками, из которых конструируются формы приложения.

Окно формы (или Конструктора формы) первоначально находится в центре экрана и имеет заголовок Form1. В нем выполняется проектирование формы, путем размещения на форме различных компонентов.

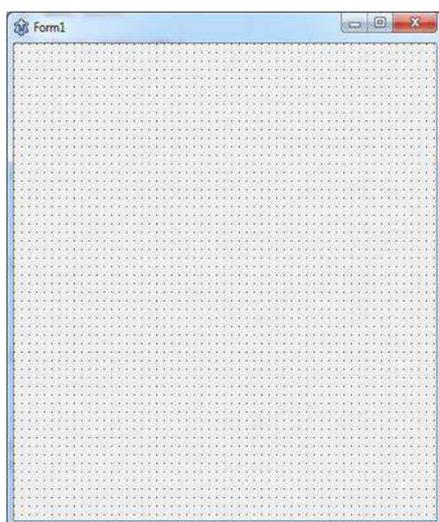


Рис. 1.3. Окно формы (или Конструктора формы)

Окно Редактора кода после запуска системы программирования находится под (или поверх) окна Формы. Редактор кода представляет собой обычный текстовый редактор, с помощью которого можно редактировать текст программы (листинг).

Окно Инспектора объектов расположено под Главным окном в левой части экрана и состоит из двух частей: в верхней части окна Инспектора объектов располагается список всех созданных объектов (форма и все компоненты, которые расположены на форме); в нижней части отображены свойства и события объектов для текущей формы или компонента. Вкладка Свойства отражает свойства выбранного компонента или формы, а вкладка События – процедуры, которые должны быть выполнены при возникновении указанного события.

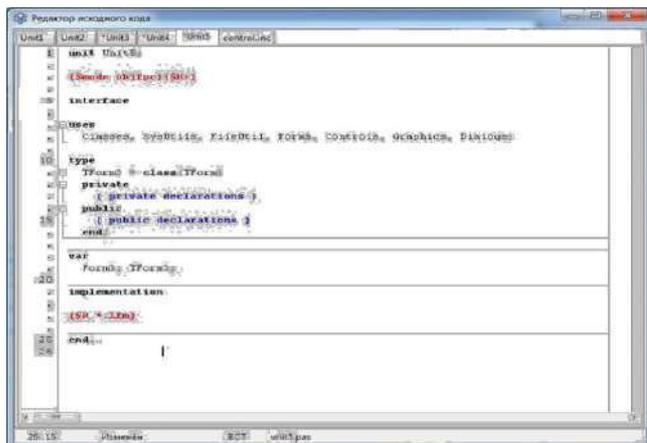
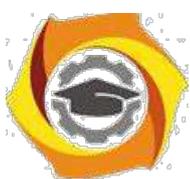


Рис. 1.4. Окно Редактора кода

Основная форма программы – содержит доступ ко всем остальным формам программы. Если закрыть подчиненное окно, то проект остается открытым. Если закрыть главную форму программы – закроется все приложение.

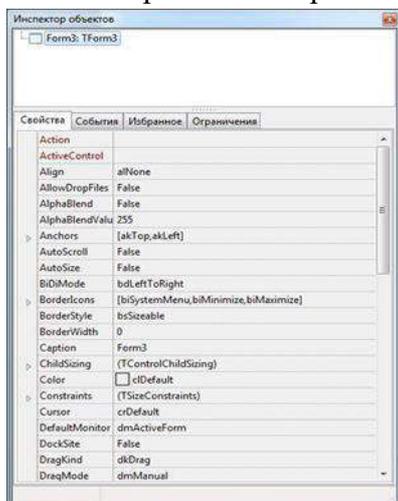


Рис. 1.5. Окно Инспектора объектов

Главное меню программы – имеет древовидную структуру и содержит доступ ко всем элементам программы от «Файл» до «Справка».

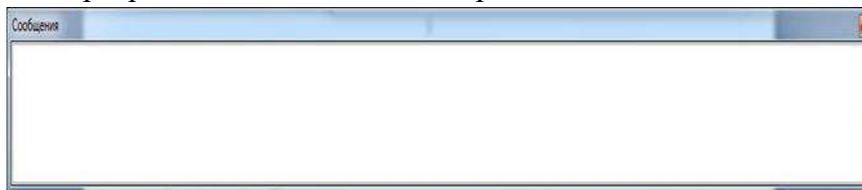


Рис. 1.6. Окно ошибок и подсказок

Управление проектом: назначение кнопок управление проектом (рис.1.7).

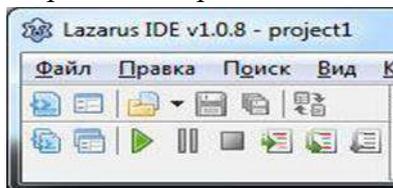
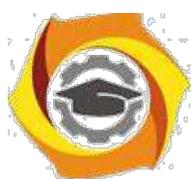


Рис. 1.7. Основные кнопки управления проектом



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

Верхний ряд: слева – направо:



- 16) Создать модуль – создает новый модуль форму.
- 17) Открыть – открывает проект, модуль или любой другой файл (понимаемый средой Lazarus).
- 18) Сохранить – сохраняет текущий модуль на диск, перед первым сохранением – запросит имя и папку, куда сохранить файл. Если текущий файл был сохранен и не редактировался, то кнопка серого цвета.
- 19) Сохранить все – Сохраняет все модули проекта, если модуль ни разу не сохранялся – предложит указать имя и папку для файлов.
- 20) Создать форму – создает новую форму программы и прикрепляет ее к проекту.
- 21) Переключить форму/модуль – осуществляет переход между окном формы и окном модуля. Можно переключаться с помощью клавиши <F12>.

Примечания:

– Среда Lazarus плохо понимает русские буквы в именах модулей (это связано с синтаксисом языка Free Pascal).

– По умолчанию при создании нового проекта к нему присоединена пустая форма. Проект может содержать несколько форм, в том числе и ни одной (консольное приложение, сервис, библиотека и т.д.). Первая форма, созданная в проекте, называется главной. Остальные подчиненные. При закрытии главной формы закрывается весь проект.

– Не все модули содержат форму проекта. Многие модули не имеют визуального представления, например модуль математики Math.

Нижний ряд кнопок слева – направо:

23. Показать модули – отображает список всех модулей проекта и позволяет переключиться на любой из них;

24. Показать формы – отображает список всех форм проекта и позволяет переключиться на любую из форм.

25. Запуск проекта – осуществляет сборку проекта и его запуск

86. Виде исполняемого файла под отладчиком, т. е. жизнью проекта управляет среда программирования Lazarus. (Если кнопка серая – проект запущен и находится в списке запущенных программ Windows).

11. Пауза – работу запущенного проекта можно приостановить

– поставить на «паузу».



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

/ Останов – принудительная остановка проекта, можно остановить проект сочетанием клавиш <Ctrl + F2>.



/ Шаг со входом – отладка проекта пошаговая с заходом во все процедуры и последовательная отладка этих процедур.

/ Шаг в обход – пошаговая отладка проекта без отладки вызываемых процедур и функций, каждая вызываемая процедура выполняется за один шаг отладки.

– Можно зайти как в процедуры своего проекта, так и библиотечные функции Lazarus, но в системные процедуры Windows (API Windows) попасть не получится.

Закладки компонентов – содержат наборы различных компонентов сгруппированные по различным признакам. В примерах использоваться компоненты из вкладок «Standard» и «Additional».

Компоненты – готовые настраиваемые блоки программ, которые можно устанавливать на форму и подключать к модулям. Для установки компонента на форму, необходимо один раз «кликнуть» на нем в панели компонентов, затем второй раз на форме –

/ том месте, где необходимо его разместить. Установленные компоненты можно выбрать «кликнув» по нему или с помощью рамки выделения.

– Выбранные компоненты имеют по краям черные прямоугольные маркеры, за которые можно перемещать объект по форме и изменять его размеры. Некоторые компоненты можно только перемещать.

– Свойства, как и типы данных, могут быть: целыми, вещественными, строковыми, логическими, множествами или сложными. События у компонентов создаются пустыми.

Инспектор объектов – форма позволяющая настроить свойства каждого компонента индивидуально. Инспектор объектов имеет несколько закладок.

Закладка «Свойство» отображает большинство свойств объекта, хотя и не все. Свойство объекта это имя и значение. Левый столбец – имя свойства, правый – значение.

Закладка «События» – позволяет посмотреть список большинства событий, на которые может реагировать компонент, а так-же процедуры привязанные к каждому из событий.

Форма программы – форма на которой разрабатывается интерфейс программы с помощью компонентов.

Модуль программы – окно в котором содержится исходный код на языке Object Pascal.

Окно ошибок – окно в котором отображаются все ошибки и подсказки при сборке проекта.

– Среда программирования Lazarus автоматически вносит изменения в код программы при добавлении компонент на форму и создании обработчиков событий. Программист создавая тело программы вносит свои изменения в код.

Ошибки могут быть:

В Синтаксические – когда исходный текст не понимает среда программирования Lazarus.

В Логические – когда код с точки зрения среды программирования написан верно, но программа выполняет не те действия, которые ожидает пользователь от программы.

Первый тип ошибок может отследить среда программирования и программист, второй тип – только программист.



Проект рекомендуется сохранять в индивидуальную папку, (проект – группа связанных файлов).

Для каждого проекта рекомендуется создавать следующую структуру: Для доступа к проекту по сети разместим в папке общие документы (Мой компьютер/ Общие документы), создадим папку с номером группы (например «ДВ-31»), далее фамилия (например «Иванов»), далее «1» для первой лабораторной, для второй «2» и т. д.

\endash Методика и порядок выполнения работы

Основным инструментом, которым пользуется программист в процессе разработки приложения, является Палитра Компонентов (рис.1.8).

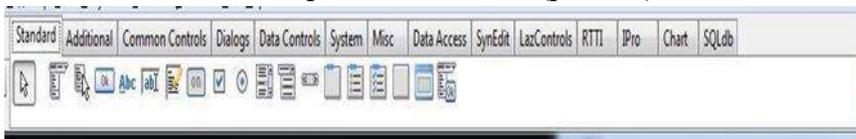


Рис. 1.8. Палитра Компонент

Палитра Компонент, позволяет выбрать нужные объекты для размещения их на Дизайнере Форм. Для использования Палитры Компонент просто первый раз щелкните мышкой на один из объектов и потом второй раз – на Дизайнере Форм. Выбранный объект появится на проектируемом окне, и им можно манипулировать с помощью мыши.

Палитра Компонент использует постраничную группировку объектов. Внизу Палитры находится набор закладок – Standard, Additional, Dialogs и т.д. Если щелкнуть мышью на одну из закладок, то можно перейти на следующую страницу Палитры Компонент. Принцип разбивки на страницы широко применяется в среде программирования Lazarus, и его легко можно использовать в своей программе.

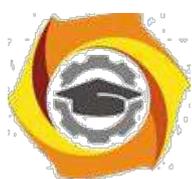
Предположим, помещаете компонент TEdit на форму; можно двигать его с места на место. Также можно использовать границу, прорисованную вокруг объекта, для изменения его размеров. Большинство других компонент можно манипулировать тем же образом. Однако, невидимые во время выполнения программы компоненты (типа TMenu или TDataBase) не меняют своей формы.

Каждый компонент является настоящим объектом, и можно менять его вид и поведение с помощью Инспектора Объектов, который состоит из двух страниц, каждую из которых можно использовать для определения поведения данного компонента.

Первая страница – это список свойств, вторая – список событий. Если нужно изменить что-нибудь, связанное с определенным компонентом, то это обычно делается это в Инспекторе Объектов.

23. к примеру, можно изменить имя и размер компонента TLabel, изменяя свойства Caption, Left, Top, Height, и Width.

Можно использовать закладки внизу Инспектора Объектов для переключения между страницами свойств и событий. Страница событий связана с Редактором; если дважды щелкнуть мышкой на правую сторону какого-нибудь пункта, то соответствующий данному-



**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

событию код автоматически запишется в Редактор, сам Редактор немедленно получит фокус, и сразу же имеете возможность добавить код обработчика данного события.



ЗАДАНИЕ

27. Изменить заголовок окна формы с Form1 на «Первая программа», используя свойство Caption.
28. Изменить цвет формы на clAqua, используя свойство Color.
29. Разместить в центре формы компоненту Label. Задать: надпись метки – «СВЕТОФОР», цвет метки – серый. Изменить свойство Font: шрифт – Times New Roman, начертание – жирный, размер – 14.
30. Расположить на форме компоненты Panel1, Panel2, Panel3, для которых поочередно задать свойство Caption пустым.
31. Расположить на форме три командные кнопки Button1, Button2, Button3. Задать надписи на этих кнопках «Красный», «Желтый», «Зеленый». В результате должна получиться форма, согласно рисунка.



16. Задать событие для первой командной кнопки Button1. Для этого необходимо выделить данный компонент и перейти в Ин-спекторе Объектов на страницу События. Затем на против события OnClick дважды щелкнуть левой кнопкой мыши. После выбора события автоматически открывается окно кода программы.
17. При нажатии на командную кнопку Button1 с надписью «Красный» цвет компоненты Panel1 будет меняться на красный, а цвет компонент Panel2 и Panel3 будет меняться на белый. Записать в процедуре следующую последовательность действий:
- ```
Procedure TForm1.Button1Click (Sender:TObject); Begin
Panel1.Color:=clRed;
Panel2.Color:=clWhite;
Panel3.Color:=clWhite; End;
```
18. Создать событие и реакцию на событие для командной кнопки Button2: цвет компоненты Panel2 будет меняться на желтый, а цвет компонент Panel1 и Panel3 будет меняться на белый. Если вы не знаете, как записать название цвета, посмотрите возможные цвета свойства Color в Инспекторе Объектов.
19. Создать событие и реакцию на событие для командной кнопки Button3: цвет компоненты Panel3 будет меняться на зеленый, а цвет компонент Panel1 и Panel2 будет меняться на белый.
20. На диске создать папку с номером группы (в скобках указать номер подгруппы). В папке с номером группы создать папку с именем Первая программа. Сохранить свою программу в папке Первая программа.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

21. Добавить появление на компоненте Panel1 при нажатии на командную кнопку Button1 информации «СТОЙТЕ» белым цветом, жирным шрифтом, размер шрифта – 12. Для



этого необходимо в имеющуюся процедуру добавить следующие действия: Procedure TForm1.Button1Click(Sender:TObject); Begin

Panel1.Color:=clRed;

Panel2.Color:=clWhite;

Panel3.Color:=clWhite;

Panel1.Caption:='СТОЙТЕ'; {задание на панели надписи} Panel1.Font.Color:=clWhite;

{задание цвета шрифта} Panel1.Font.Size:=12; {задание размера шрифта}  
Panel1.Font.Style:=[fsBold]; {задание начертания шрифта} End;

22. Добавить появление на компоненте Panel2 при нажатии на командную кнопку Button2 информации «ВНИМАНИЕ» белым цветом, жирным шрифтом, размер шрифта – 12.

23. Добавить появление на компоненте Panel3 при нажатии на командную кнопку Button3 информации «ИДИТЕ» белым цветом, жирным шрифтом, размер шрифта – 12.

24. Сохранить изменения в программе и запустить ее на ис-полнение.

25. На форме добавить командную кнопку Button4. Задать для нее надпись «Автор». При нажатии на кнопку должно выводиться сообщение об авторе программы. Для реализации данного задания задать для нее реакцию на событие:

Procedure TForm1.Button4Click(Sender:TObject); begin

ShowMessage('Программа разработана Ивановым С. '); end;

#### Контрольные вопросы

- № Основные элементы IDE LAZARUS, панели, элементы управления.
- № Отладочный инструментарий LAZARUS.
- № Порядок установки точек останова программы. Отладочный вывод информации.
- № Опишите структуру и назначение отдельных элементов головной программы приложения Lazarus.
- № Каково назначение модуля в проекте приложения Lazarus? Опишите назначение отдельных разделов модуля.
- № Как различается доступность объектов, описанных в разделах interface и implementation?
- № Как указать ссылку на свойства и методы объекта в тексте программы?
- № Какие компоненты входят в интегрированную среду разработки приложений Lazarus?
- № Перечислите основные компоненты окна среды Lazarus и укажите их назначение.
- № Понятие подпрограммы. Процедуры в Free Pascal.
- № Понятие подпрограммы. Функции в Free Pascal.
- № Составление библиотек подпрограмм. Модули.
- № Поясните понятия «Раздел объявлений», «Раздел реализации».



## Лабораторная работа № 2 . Тестирование и отладка приложений в среде LAZARUS

### Цель лабораторной работы

Освоить работу с встроенным отладчиком интегрированной среды LAZARUS, изучить категории ошибок, способы их обнаружения и устранения.

### 37. Краткие теоретические сведения

#### Тестирование и отладка программы

Чем больше опыта имеет программист, тем меньше ошибок в коде он совершает, но даже самый опытный программист всё же допускает ошибки. И любая современная *среда разработки* программ должна иметь собственные инструменты для отладки приложений, а также для своевременного обнаружения и исправления возможных ошибок. Программные ошибки на программистском сленге называют *багами* (англ. *bug* - жук), а программы отладки кода - *дебаггерами* (англ. *debugger* - отладчик). Lazarus, как современная *среда разработки* приложений, имеет собственный встроенный отладчик, работу с которым мы разберем на этой лекции.

Ошибки, которые может допустить программист, условно делятся на три группы:

37. Синтаксические
38. Времени выполнения (run-time errors)
39. Алгоритмические

Синтаксические ошибки

**Синтаксические ошибки** легче всего обнаружить и исправить - их обнаруживает *компилятор*, не давая скомпилировать и запустить программу. Причем *компилятор* устанавливает *курсор* на ошибку, или после неё, а в окне сообщений выводит соответствующее сообщение, например, такое:

```
procedure TForm1.Button1Click(Sender: TObject);
40 begin
41 i := 10;
end;
end.
45
```

41: 4 BCT C:\TEMP\unit1.pas

unit1.pas(41,4) Error: Identifier not found "i"  
unit1.pas(46) Fatal: There were 1 errors compiling module, stopping

Рис. 2.1. Найденная компилятором синтаксическая ошибка - нет объявления переменной i



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

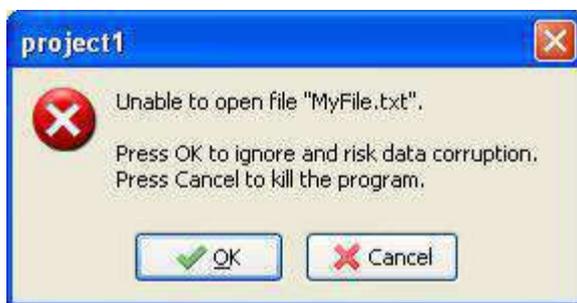
Подобные ошибки могут возникнуть при неправильном написании директивы или имени функции (процедуры); при попытке обратиться к переменной или константе, которую не объявляли (рис. 2.1); при попытке вызвать функцию (процедуру, переменную, константу)



из модуля, который не был подключен в разделе *uses*; при других аналогичных недосмотрах программиста.

Как уже говорилось, *компилятор* при нахождении подобной ошибки приостанавливает процесс компиляции, выводит сообщение о найденной ошибке и устанавливает *курсор* на допущенную ошибку, или после неё. Программисту остается только внести исправления в *код программы* и выполнить повторную компиляцию.

**Ошибки времени выполнения (run-time errors)** тоже, как правило, легко устранимы. Они обычно проявляются уже при первых запусках программы, или во *время тестирования*. Если такую программу запустить из среды Lazarus, то она скомпилируется, но при попытке загрузки, или в момент совершения ошибки, приостановит свою работу, выведя на экран соответствующее сообщение. Например, такое:



**Рис. 2.2.** Сообщение Lazarus об ошибке времени выполнения

В данном случае *программа* при загрузке должна была считать в *память* отсутствующий *текстовый файл MyFile.txt*. Поскольку *программа* вызвала ошибку, она не запустилась, но в среде Lazarus процесс отладки продолжается, о чем свидетельствует сообщение в скобках в заголовке главного *меню*, после названия проекта. Программисту в подобных случаях нужно сбросить отладчик командой *меню "Запуск -> Сбросить отладчик"*, после чего можно продолжить работу над проектом.

Ошибка времени выполнения может возникнуть не только при загрузке программы, но и во время её работы. Например, если бы попытка чтения несуществующего файла была сделана не при загрузке программы, а при нажатии на кнопку, то *программа* бы нормально запустилась и работала, пока *пользователь* не нажмет на эту кнопку.

Если программу запустить из самой *Windows*, при возникновении этой ошибки появится такое же сообщение. При этом если нажать **"ОК"**, *программа* даже может запуститься, но корректно работать все равно не будет.

**Ошибки времени выполнения** бывают не только явными, но и неявными, при которых *программа* продолжает свою работу, не выводя никаких сообщений, а программист даже не догадывается о наличии ошибки. Примером неявной ошибки может служить так называемая **утечка памяти**. Утечка памяти возникает в случаях, когда программист забывает освободить выделенную под *объект память*. Например, мы объявляем переменную типа *TStringList*, и работаем с ней:



begin

**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



```
MySL:= TStringList.Create;
MySL.Add('Новая строка');
end;
```

30.     данном примере программист допустил типичную для начинающих ошибку - не освободил *класс* TStringList. Это не приведет к сбою или аварийному завершению программы, но в итоге можно бесполезно израсходовать очень много памяти. Конечно, эта *память* будет освобождена после выгрузки программы (за этим следит *операционная система*), но утечка памяти во *время выполнения* программы тоже может привести к неприятным последствиям, потребляя все больше и больше ресурсов и излишне нагружая *процессор*. В подобных случаях после работы с объектом программисту нужно не забывать освобождать *память*:

```
begin
 MySL:= TStringList.Create;
 MySL.Add('Новая строка');
 ...; //работа с объектом
 MySL.Free; //освободили объект
end;
```

Однако *ошибки времени выполнения* могут случиться и во время работы с объектом. Если есть такой риск, программист должен не забывать про возможность обработки исключительных ситуаций. В данном случае вышеприведенный код правильной будет оформить таким образом:

```
begin
 try
 MySL:= TStringList.Create;
 MySL.Add('Новая строка');
 ...; //работа с объектом
 finally
 MySL.Free; //освободили объект, даже если была
 ошибка end;
end;
```

Итак, во избежание ошибок времени выполнения программист должен не забывать делать проверку на правильность ввода пользователем допустимых значений, заключать опасный код в блоки try...finally...end или try...except...end, делать проверку на существование открываемого файла функцией FileExists и вообще соблюдать предусмотрительность во всех слабых местах программы. Не полагайтесь на пользователя, ведь недаром говорят, что если в программе можно допустить ошибку, *пользователь* эту возможность непременно найдет.

#### Алгоритмические ошибки

Если вы не допустили ни синтаксических ошибок, ни ошибок времени выполнения, *программа* скомпилировалась, запустилась и работает нормально, то это еще не означает, что в программе нет ошибок. Убедиться в этом можно только в процессе её *тестирования*.



**Тестирование** - процесс проверки работоспособности программы путем ввода в неё различных, даже намеренно ошибочных данных, и последующей контрольной проверке выводимого результата.

Если *программа* работает правильно с одними наборами исходных данных, и неправильно с другими, то это свидетельствует о наличии алгоритмической ошибки. Алгоритмические ошибки иногда называют логическими, обычно они связаны с неверной реализацией алгоритма программы: вместо "+" ошибочно поставили "-", вместо "/" - "\*", вместо деления значения на 0,01 разделили на 0,001 и т.п. Такие ошибки обычно не обнаруживаются во *время компиляции*, *программа* нормально запускается, работает, а при анализе выводимого результата выясняется, что он неверный. При этом *компилятор* не укажет программисту на ошибку - чтобы найти и устранить её, приходится анализировать код, пошагово "прокручивать" его выполнение, следя за результатом. Такой процесс называется *отладкой*.

**Отладка** - процесс поиска и устранения ошибок, чаще алгоритмических. Хотя отладчик позволяет справиться и с ошибками времени выполнения, которые не обнаруживаются явно.

Работа с отладчиком

Давайте от теории перейдем к практике. Загрузите **Lazarus** с новым проектом, установите на форму простую кнопку и сохраните проект в папку **JP2**. Имена проекта, формы, модуля и кнопки изменять не нужно, оставьте имена, данные по умолчанию.

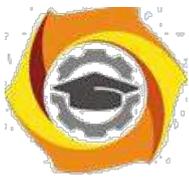
Далее, сгенерируйте событие OnClick для кнопки, в котором напишите следующий код:

```
procedure TForm1.Button1Click(Sender:
TObject); var
 i: Integer;
 st: TStringList;
begin
 //создаем список строк:
 st:= TStringList.Create;
 try
 //генерируем список:
 for i:= -3 to 3 do begin
 st.Append('10/'+IntToStr(i)+'='+FloatToStr(10/i));
 end;
 //выводим список на экран:
 ShowMessage(st.Text);
 finally
 //st будет освобождена даже в случае run-time ошибки:
 st.Free;
 end;
end;
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

23. данной программе целочисленную переменную  $i$  используем в качестве счетчика для *цикла* for. Цикл производим от -3 до 3, то есть, 7 раз, включая ноль. В теле *цикла* мы делим 10 на *значение*  $i$ , результат оформляем в виде строки и добавляем к списку строк st.



Выше говорилось, что подобные действия нужно заключать в блок обработки исключительных ситуаций `try-finally-end`, что мы и сделали.

Если вы внимательно изучали курс, то невооруженным глазом видите, что при четвертом проходе *цикла* произойдет ошибка времени выполнения - *деление 10 на ноль*. Такой очевидный пример больше всего подходит для знакомства с встроенным отладчиком, так как вы уже знаете, где будет ошибка, и сможете проанализировать работу отладчика. Поэтому притворимся, что не подозреваем об ошибке.

Итак, программу мы написали, сохранили, пора её компилировать. Нажмите кнопку "Запустить" на **Панели управления** (или `<F9>`). Программа нормально скомпилировалась и запустилась. Нажмем кнопку **Button1**. И тут же получаем ошибку:

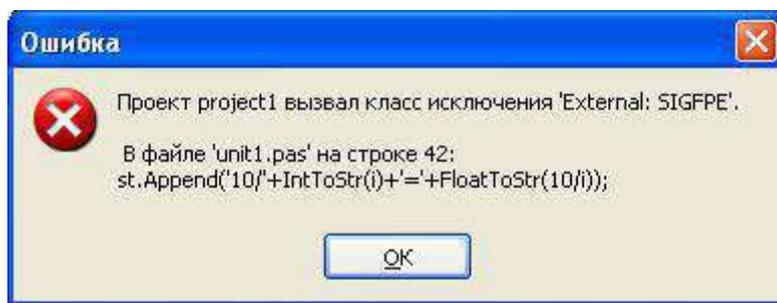


Рис. 2.3. Сообщение Lazarus об ошибке

Судя по сообщению, ошибка произошла во время выполнения кода 42-й строки. Ладно, нажмем "ОК" и командой "Запуск -> Сбросить отладчик" прекратим выполнение программы. Вернемся к коду и проанализируем 42-ю строку (если вы добавляли пустые строки, то у вас будет другой номер):

```
st.Append('10/'+IntToStr(i)+'='+FloatToStr(10/i));
```

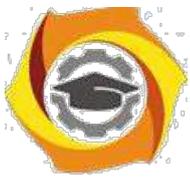
Ну что, ничего криминального тут нет, почему же произошла ошибка? Код верный и должен был нормально выполняться... Когда вы заходите в подобный *тупик*, помочь вам может *здравый смысл* и встроенный отладчик. *Здравый смысл* говорит, что ошибка произошла где-то в теле *цикла* `for`. А чтобы воспользоваться отладчиком, нужно приостановить выполнение программы на этом цикле, чтобы потом построчно его продолжить. Для остановки работы программы служат так называемые **точки останова** (англ. *breakpoints*).

**Точки останова** - это строки, перед выполнением которых отладчик приостанавливает выполнение программы, и ждет ваших дальнейших действий.

Вы можете установить одну такую точку или несколько, в различных частях кода. Поскольку ошибка возникает в 42-й строке, разумней будет приостановить выполнение на предыдущей, 41-й строке. Переведите *курсор* на эту строку, на любое её *место*.

Установить точку останова можно разными способами:

- В Командой главного меню "Запуск -> Добавить точку останова -> Точка останова в исходном коде". В открывшемся окне "Параметры точки останова" нажать "ОК".
- В Щелкнуть по строке правой кнопкой, и в всплывающем меню выбрать "Отладка -> Переключить точку останова".
- В Нажать "горячую клавишу" `<F5>`.



ООО Щелкнуть по нужной строке в левой части Редактора кода, где указаны номера строк.

Последние два способа наиболее удобны, но выбирать вам. В любом случае, строка с установленной точкой останова окрасится красным цветом:

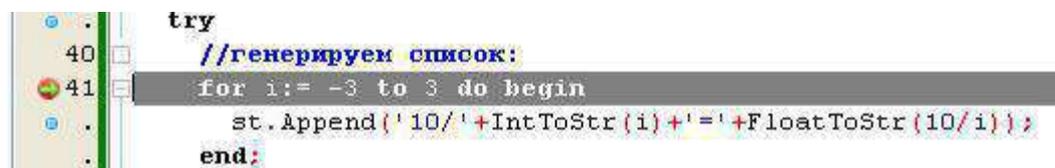


```
try
//генерируем список:
41 for i:= -3 to 3 do begin
. st.Append('10/' + IntToStr(i) + '=' + FloatToStr(10/i));
. end;
. //выводим список на экран:
45 ShowMessage(st.Text);
```

Рис. 2.4. Строка с точкой останова

Снять точку останова удобней также последними двумя способами. Точка останова у нас есть, снова нажимаем кнопку "Запустить". Программа начинает свою работу, нажимаем кнопку "Button1".

Теперь программа не вывела ошибку, а приостановила свою работу и вывела на передний план Редактор кодов с выделенной серым цветом строкой, которая в данный момент готовится к выполнению:



```
try
//генерируем список:
41 for i:= -3 to 3 do begin
. st.Append('10/' + IntToStr(i) + '=' + FloatToStr(10/i));
. end;
```

Рис. 2.5. Строка, которая будет выполнена далее

Тут очень важно понимать, что программа была остановлена ДО выполнения этой строки, а не ПОСЛЕ неё. То есть, в настоящий момент переменной *i* еще не присвоено значение -3. Далее, мы можем выполнять с отладчиком различные действия, которые собраны в разделе главного меню "Запуск". Обычно требуется пошаговое выполнение программы. Для этого можно использовать команду "Запуск -> Шаг в обход" (или <F8>), "Запуск -> Шаг с входом" (или <F7>) или "Запуск -> Шаг с выходом" (или <Shift+F8>). "Шаг в обход" означает, что если в коде будет встречен вызов какой-нибудь функции или процедуры, отладчик выполнит их и остановится на следующей после вызова строке. При выборе "Шаг со входом", отладчик также пошагово будет выполнять и вызываемые функции-процедуры. "Шаг с выходом" подразумевает, что если в строке нет вызовов функций, то остановки происходят, как при "Шаг в обход". Если в строке есть выражение, то остановка происходит вначале перед строкой, затем перед вычислением каждой функции, чтобы мы имели возможность посмотреть значения параметров, передаваемых в функцию.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

У нас вызовов функций нет, поэтому мы можем воспользоваться как <F7>, так и <F8>  
(чаще всего используют <F8> - Шаг в обход).



Итак, нажмем <F8>, и отладчик выполнит строку с точкой останова, и выделит серым следующую строку. Снова нажмем <F8>, и снова будет выделена эта строка - был выполнен шаг *цикла*. Нажав несколько раз <F8>, мы добьемся появления на экране всё той же ошибки, которая заблокирует дальнейшее выполнение программы. Становится понятно, что цикл нормально выполняется несколько проходов, после чего всё же возникает ошибка. Включаем логику: внутри *цикла* у нас изменяется только *переменная i*, значит, ошибка как-то связана с ней. А как узнать, как именно?

Здесь нам на помощь приходит еще один полезный инструмент отладчика - наблюдение за значениями переменных. Сбросьте программу командой "Запуск -> Сбросить отладчик". Теперь снова нажмите кнопку "Запустить", а потом снова кнопку "Button1". Отладчик снова приостановил выполнение программы на строчке с циклом, однако не спешите нажимать <F8>. Для начала, добавим наблюдение над переменной *i*. Делается это командой "Запуск -> Добавить наблюдение", которая была недоступна, пока *программа* не начала выполняться. В строке "Выражение" укажите переменную *i*, и нажмите "ОК":

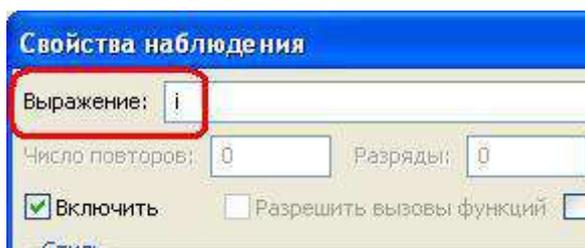


Рис. 2.6. Установка наблюдения за переменной

Теперь отладчик наблюдает за значениями переменной *i*, но нам от этого не легче - мы то не видим этих значений! Чтобы их увидеть, нужно вывести на экран окно **Списка наблюдений**. Делается это командой "Вид -> Окна отладки -> Окно наблюдений" или "горячими клавишами" <Ctrl+Alt+W>.

Расположите окно ниже **Редактор кода**, на *место* **Окна сообщений**. В этом окне вы сможете видеть *выражение* - нашу переменную *i*, и её текущее *значение*. Чтобы показать работу с отладчиком более наглядно, давайте добавим еще одно *выражение*, за которым будем наблюдать. Выберите "Запуск -> Добавить наблюдение" и в строке "Выражение" укажите не просто переменную, а *выражение* которое у нас должно вычисляться внутри *цикла*. В окне **Списка наблюдения** вы увидите и переменную *i*, и *выражение*, а также их значения:

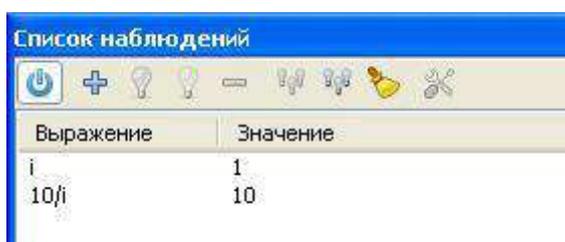




Рис. 2.7. Окно Списка наблюдений

Поскольку переменной  $i$  еще не было присвоено значения  $-3$ , то в колонке значений вы, скорее всего, увидите  $1$ , которым по умолчанию была проинициализирована наша *переменная*. Соответственное значение будет и у выражения. Теперь мы готовы двигаться дальше. Нажимаем **<F8>**. В **Списке наблюдений** сразу же изменилась картина -  $i$  теперь равно  $-3$ , а выражение  $-3,3333\dots$

Нажимаем **<F8>** ещё раз. Снова значения изменились, теперь  $i = -2$ , а *выражение* =  $-5$ . Мы понимаем, что цикл работает, и два его шага были сделаны. Нажимаем **<F8>** еще два раза. Сейчас *переменная* содержит ноль, а значение выражения указывает "inf". Однако строка с вычислением еще не была выполнена, не забываем об этом. Снова нажимаем **<F8>**,  
14. снова получаем ошибку. А в значениях переменной и выражения видим слово "evaluating", что переводится, как "оценка". Теперь мы наглядно видим, что в строке `st.Append('10/'+IntToStr(i)+'=' + FloatToStr(10/i));`

возникает ошибка, когда *переменная*  $i$  равна нулю. И тут уже несложно догадаться, почему эта ошибка возникает - потому что происходит попытка деления  $10$  на  $0$ .

Это можно проверить, пропустив выполнение вычисления, когда  $i=0$ . Закройте окно с ошибкой, сбросьте отладчик. Снова нажмите кнопку "Запустить", и кнопку "Button1". Снова выведите окно **Списка наблюдения**. Нажимайте **<F8>**, пока  $i$  не станет  $0$ ,

32. *выражение* - inf. Теперь, в **Окне наблюдений** щелкните правой кнопкой мыши по строке с переменной  $i$ , и в всплывающем меню выберите команду "Вычислить/Изменить". В открывшемся окне вы увидите строку "Выражение", где будет указана *переменная*  $i$ .

29. поле "Результат" будет указано значение  $0$ . А в строке "Новое значение" нам нужно указать значение, которое мы желаем принудительно присвоить переменной. Тут укажем  $1$  и нажмем **<Enter>** или кнопку "Изменить". В поле "Результат" значение должно смениться на  $1$ . Снова нажмем **<F8>**, значение  $i$  изменится на  $2$ , ошибки не будет. Нажав **<F8>** еще несколько раз, мы доберемся до конца программы и увидим сообщение, которое она и должна была вывести по нашему замыслу:

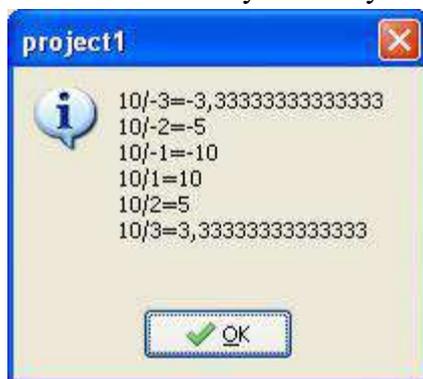


Рис. 2.8. Результирующее сообщение программы

Как видите, *вычисление*, где  $i$  равна нулю, было пропущено.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Встроенный отладчик имеет и другие инструменты, с которыми вы сами сможете со временем освоиться, экспериментируя с ними.



### Контрольные вопросы и задания

31. Покажите на примерах, как в оболочке Lazarus осуществляется: запуск и выход из оболочки, загрузка и сохранение файла, вызов справки, в т.ч. по ключевому слову, на которое указывает курсор, контекстный поиск и замена текста, компиляция и запуск программы.
32. Объясните понятия: синтаксическая ошибка, ошибка времени выполнения, логическая ошибка.
33. Используя приемы отладки, ответьте на следующие вопросы по программе `project8.lpr`:
34. Каково значение переменной `F` при вызове подпрограммы `Factorial(14)`, когда:
- 8. `i` равно 5
  - 9. `i` равно 9
  - 10. `i` равно 12 ?
10. Каково значение переменной `Sum` при вызове подпрограммы `Harmonic(25)`, когда:
- а) `i` равно 5
  - б) `i` равно 10
  - с) `i` равно 20 ?
11. 4Покажите на примерах, как в оболочке Lazarus осуществляется: добавление, редактирование и удаление переменных в окне просмотра значений переменных, пошаговое выполнение программ, в т.ч. с пошаговым выполнением операторов в вызываемых подпрограммах, выполнение программы до строки, на которую указывает курсор, завершение отладки программы, создание, редактирование и удаление точек останова программы.



### Лабораторная работа № 3. Разработка и отладка приложения «Калькулятор» в среде LAZARUS

#### Цель лабораторной работы

Освоить работу с встроенным отладчиком интегрированной среды LAZARUS в процессе самостоятельной работы над приложением в соответствии с индивидуальным заданием.

**Задание:** Используя IDE LAZARUS разработать соответствующее вашему варианту приложение и предварительно протестировать его на работоспособность. В приложении учесть ситуации не корректного ввода данных. Составить отчет о выполненной работе, в котором показать на примерах применение отладочных средств IDE LAZARUS.

#### Перечень индивидуальных заданий:

##### *Вариант 1*

Приложение – «Калькулятор»:

Приложение должно содержать два поля ввода: первое для ввода целочисленных переменных  $x$ , второе для ввода вещественных переменных  $y$ . Семь кнопок для вычисления соответствующих операций:  $x+y$ ,  $x-y$ ,  $x*y$ ,  $x/y$ ,  $x^y$ ,  $\ln y$ ,  $\log x y$ . Получаемый результат должен выводиться в отдельном новом окне мессадж бокса – «Результат».

Примерный внешний вид разрабатываемого приложения:

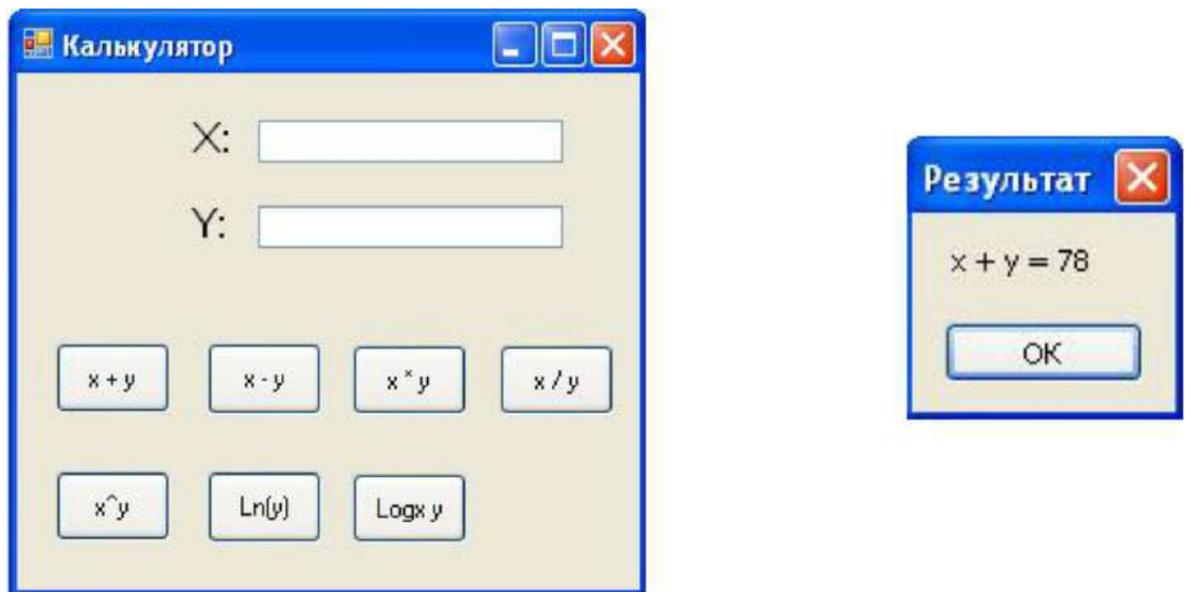


Рис. 3. 1



### Вариант 2

Приложение – «День недели»:

Приложение должно содержать три поля ввода: первое для ввода целочисленной переменной Число, второе для ввода текстовой переменной Месяц (здесь возможно использовать как альтернативу содержащий список месяцев combobox), третье для ввода целочисленной переменной Год. Так же должна быть кнопка «Вычислить». Получаемый результат необходимо выводиться в отдельном новом окне messagebox – «Результат».

Примерный внешний вид разрабатываемого приложения:

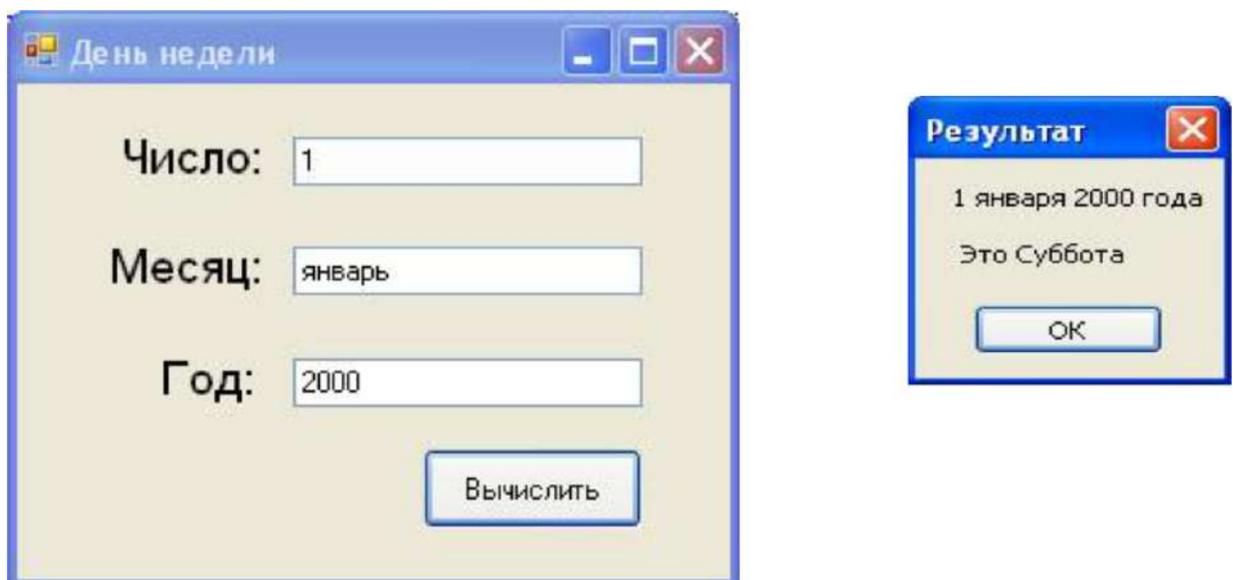
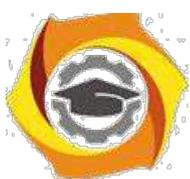


Рис. 3. 2



### Вариант 3

Приложение – «Конвектор валют»:

Приложение должно содержать два combobox: первый для выбора Исходной валюты, т.е. валюты с которой будем переводить, второй для выбора Результирующей валюты, т.е. валюты в которую будем переводить. И два поля ввода: первое для ввода целочисленной переменной Исходной суммы, второе для ввода вещественного Курса по которому осуществляется перевод. Так же должна быть кнопка «Перевод». Получаемый результат необходимо выводиться в отдельном новом окне messagebox – «Результат».

Примечание1: Оба combobox должны содержать в себе, по крайней мере, следующие четыре валюты: Рубли, Доллары, Евро, Фунты.

Примечание2: После выбора обеих валют в строке для ввода Курса, должен появиться какой либо установленный по умолчанию в программе курс перевода. Но, пользователь должен иметь возможность изменить этот появившийся «курс по умолчанию», если он его не устраивает.

Примерный внешний вид разрабатываемого приложения:

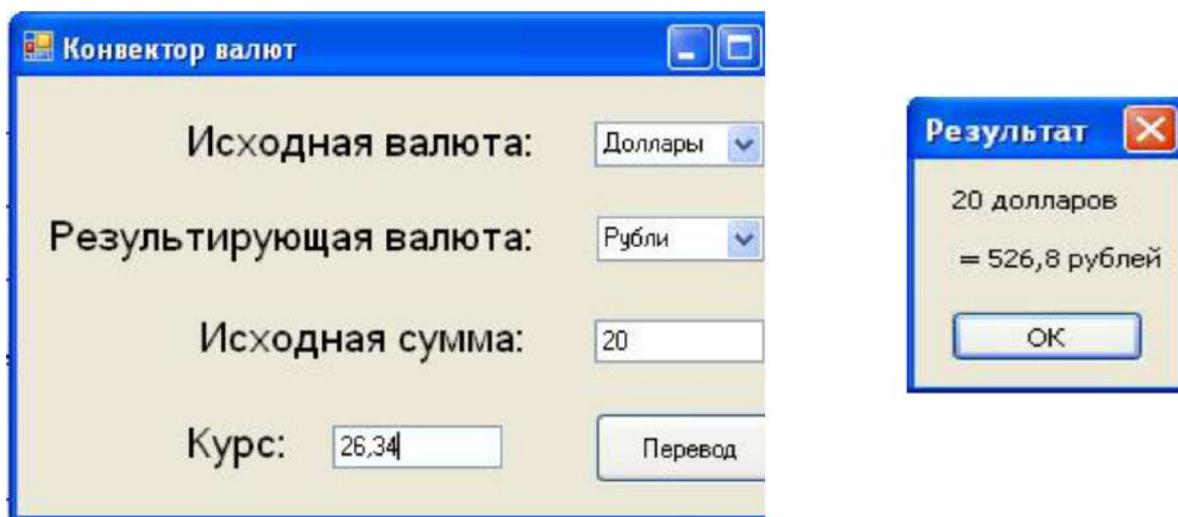


Рис. 3. 3



#### Вариант 4

Приложение – «Площади/Объёмы фигур»:

Приложение должно содержать один checkbox – 3D-объект, один combobox – Фигура и два поля для ввода. В зависимости от того нажат checkbox или нет, список фигур для выбора должен быть следующим:

15. При не нажатом checkbox: Квадрат, Круг, Треугольник;

16. При нажатом checkbox: Куб, Шар, Пирамида;

Поля для ввода должны отображаться (быть видимыми) в том количестве, которое необходимо для ввода необходимых данных для той или иной фигуры. Так, должны быть видимы: Сторона (для Квадрата), Радиус (для Круга), Основание и Высота (для Треугольника), Сторона (для Куба), Радиус (для Шара), Сторона и Высота (для Пирамиды).

Так же должна присутствовать кнопка «Вычислить», а получаемый результат необходимо выводиться в отдельном новом окне messagebox – «Результат».

Примечание1: Площади фигур вычисляются по следующим формулам:

- $S_{\text{квадрат}} = a^2$
- $S_{\text{круга}} = r^2$
- $S_{\text{треугольника}} = \frac{1}{2} h a$
- $V_{\text{куба}} = a^3$
- $V_{\text{шара}} = \frac{4}{3} r^3$
- $V_{\text{пирамиды}} = \frac{1}{3} h a^2$

Примерный внешний вид разрабатываемого приложения:

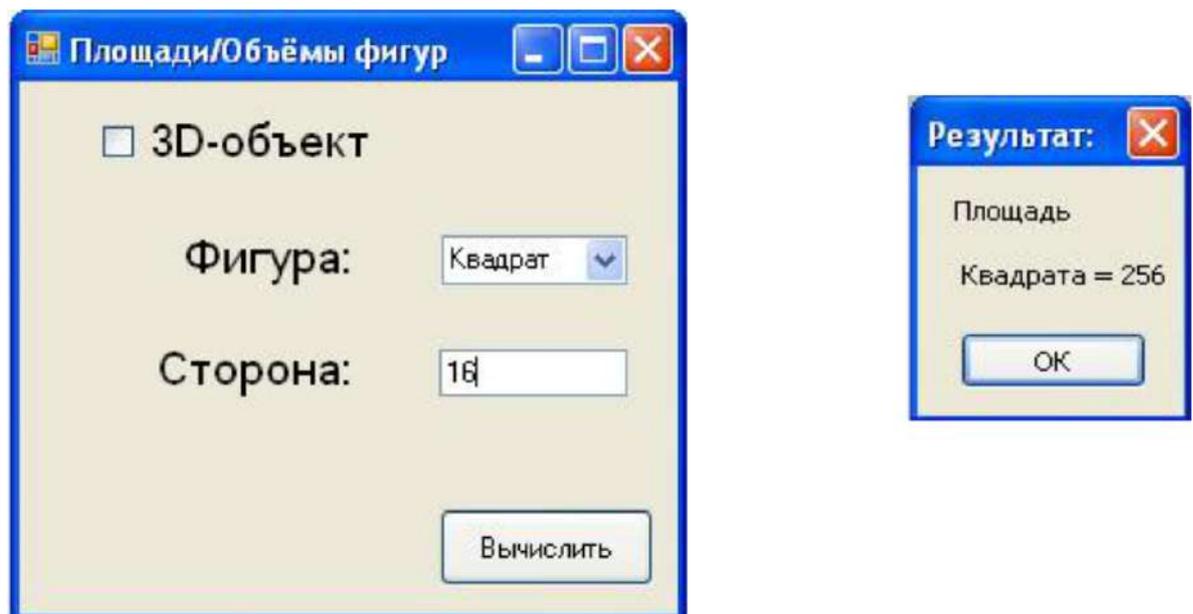


Рис. 3. 4



**Вариант 5**

Приложение – «Гороскопы»:

Приложение должно содержать три поля для ввода: Число (целое), Месяц (строковое или целое), Год (целое). Два checkbox-а для выбора Пола: Муж и Жен. Так же должна присутствовать кнопка «Определить», выводящая получаемый результат в отдельное окно messagebox – «Результат».

Примечание1: Для выбора пользователем месяца вместо поля для ввода можно использовать combobox содержащий, в себе все 12 месяцев года.

Примечание2: Если сложно реализовать программу, определяющую кроме зодиакального знака ещё и китайский год рождения человека, то от определения года можно в приложении отказаться.

Примерный внешний вид разрабатываемого приложения:

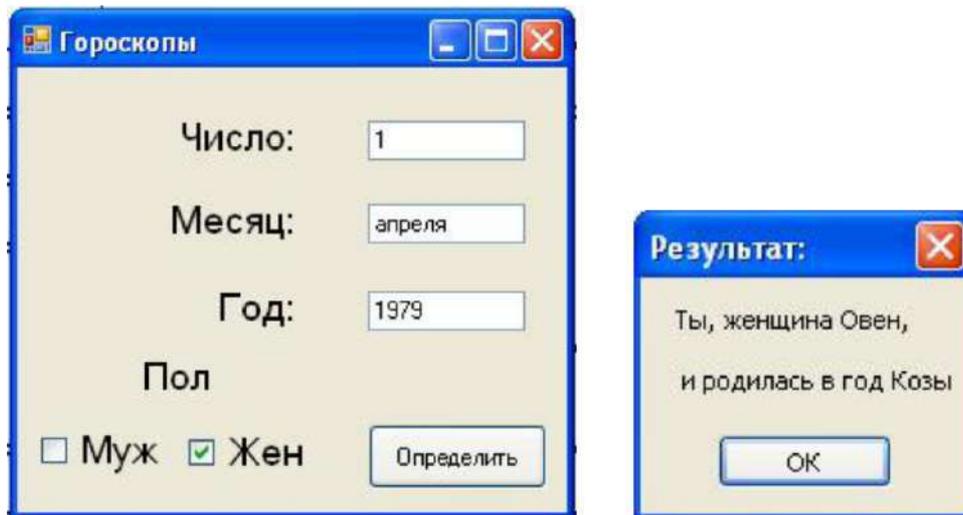
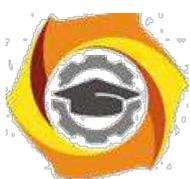


Рис. 3. 5

### Контрольные вопросы

9. Каково назначение обработчиков событий? Каким образом можно инициировать создание процедуры-обработчика событий? Что такое отладка программного средства?
10. Что такое тестирование программного средства?
11. Что такое автономная отладка программного средства?
12. Что такое комплексная отладка программного средства?
13. Что такое ведущий отладочный модуль?
14. Что такое отладочный имитатор программного модуля?
15. Проблемы и перспективы развития современной программной инженерии.
16. Различные подходы в программировании: «снизу-вверх», «сверху-вниз» (структурный подход), объектно-ориентированный



## Лабораторная работа № 4. Нарастиваемый подход к тестированию

### Цель лабораторной работы

Освоить методику нарастиваемого подхода к тестированию. Используя «Нарастиваемый подход к тестированию» создать тестовые примеры и провести тестирование разработанной на предыдущем занятии программы.

### 10. Краткие теоретические сведения

«Нарастиваемый подход к тестированию» содержит 8 стадий:

#### **Стадия 1: Изучение** (Ознакомиться с приложением);

Поработать какое-то время с тестируемым приложением, изучить его...

Например, у нас имеется приложение «Налоговый калькулятор», которое рассчитывает величину подоходного налога, взимаемого с налогоплательщика в США.

Приложение содержит много окон для ввода данных, четыре из которых показаны ниже. В выходном окне перечисляются результаты.

Налоговый калькулятор: сведения о налогоплательщике

Фамилия:  Номер карточки социального пособия:

Фамилия супруга (супруги):  Номер карточки социального пособия супруга (супруги):

Домашний адрес:

Освобождение от уплаты налогов:  Вы  Супруг (супруга)

Список иждивенцев:

Фамилия:  Номер карточки социального пособия:

Статус:

Холост

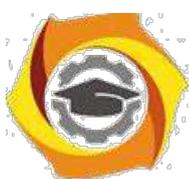
Женат (замужняя), с совместной заявкой

Женат (замужняя), с отдельной заявкой

Глава семейства

Вдовец

Рис. 4. 1 Налоговый калькулятор. Информация о налогоплательщике



Налоговый калькулятор: Информация о доходах

Доход:

|                                                                    |                      |
|--------------------------------------------------------------------|----------------------|
| Жалованья, оклады, премии                                          | <input type="text"/> |
| Доля, облагаемая налогом                                           | <input type="text"/> |
| Обычные дивиденды                                                  | <input type="text"/> |
| Коммерческие доходы или потери                                     | <input type="text"/> |
| Убытки или прибыли на капитал                                      | <input type="text"/> |
| Полученные алименты                                                | <input type="text"/> |
| Распределение по пенсионным счетам                                 | <input type="text"/> |
| Пособия и рента                                                    | <input type="text"/> |
| Арендная плата по недвижимости, лицензионные платежи, товарищества | <input type="text"/> |
| Доходы или потери от сельского хозяйства                           | <input type="text"/> |
| Пособие по безработице                                             | <input type="text"/> |
| Пособия по социальному обеспечению                                 | <input type="text"/> |
| Прочие доходы                                                      | <input type="text"/> |

<<Назад    Дальше>>    Отмена

Рис. 4. 2 Налоговый калькулятор. Информация о доходах

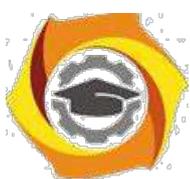
Налоговый калькулятор: статьи отчислений

Статьи отчислений:

|                                     |                      |
|-------------------------------------|----------------------|
| Расходы на медицинское обслуживание | <input type="text"/> |
| Местный подоходный налог            | <input type="text"/> |
| Налог на недвижимость               | <input type="text"/> |
| Налог на движимое имущество         | <input type="text"/> |
| Процент по закладной на дом         | <input type="text"/> |
| Благотворительные отчисления        | <input type="text"/> |
| Невозмещенные расходы служащего     | <input type="text"/> |
| Налоговые расходы                   | <input type="text"/> |
| Прочие отчисления                   | <input type="text"/> |

<<Назад    Дальше>>    Отмена

Рис. 4. 3 Налоговый калькулятор. Информация о статьях отчислений



Налоговый калькулятор: налоги и кредиты

Налоги и кредиты:

|                                                                            |                      |
|----------------------------------------------------------------------------|----------------------|
| Кредит на социальное обеспечение ребенка                                   | <input type="text"/> |
| Выходное пособие или пособие по инвалидности                               | <input type="text"/> |
| Кредит на получение образования                                            | <input type="text"/> |
| Налоговая скидка при наличии детей                                         | <input type="text"/> |
| Налог на самостоятельную предпринимательскую деятельность                  | <input type="text"/> |
| Пенсионный налог                                                           | <input type="text"/> |
| Налог на предпринимателей, работающих дома                                 | <input type="text"/> |
| Федеральный подоходный налог                                               | <input type="text"/> |
| Налоговые льготы, предоставляемые получателям заработной платы и жалованья | <input type="text"/> |

<<Назад    Дальше>>    Отмена

Рис. 4. 4 Налоговый калькулятор. Информация о налогах и кредитах

Налоговый калькулятор: итоги за 2000-й налоговый год

Отчет по 2000-му налоговому году:

Общий налог: 0\$  
Всего к уплате: 0\$  
Количество возвращаемых средств: 0\$  
Задолженность: 0\$  
Размер пени, приблизительно: 0\$

Готово

Рис. 4. 5 Налоговый калькулятор. Вычисленные результаты

**Стадия 2: Базовый тест** (Разработать и реализовать один простой тестовый пример);

Разработать при помощи MS-Excel таблицу для базового тестового примера, вбить в неё базовый тестовый пример, провести тест приложения на его основе и указать в таблице полученный при тестировании результат. Сравнить полученный результат с ожидаемым результатом, сделать вывод и отметку в таблице, о том пройден тест или нет.



Таблица 1.1. Базовый тестовый пример

| ID тестового примера | Статус         | Заруботок, (\$) | Величина отчислений, (\$) | Ожидаемые результаты: причитающийся налог, (\$) | Реальный результат |
|----------------------|----------------|-----------------|---------------------------|-------------------------------------------------|--------------------|
| налог 1              | одинокий (-ая) | 40 000          | 0                         | 5 779                                           |                    |

**Стадия 3: Анализ тенденций** (Определить, работает ли приложение так, как было задумано, когда ещё нельзя предварительно оценить реальные результаты);

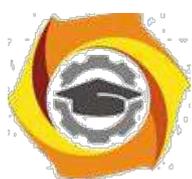
Разработать 2-3 серии из 6-10 тестов и провести анализ тенденций при работе приложения. Результаты и выводы зафиксировать в файле электронной таблицы. (Ниже приводится таблица с одной серией тестов для анализа 1 тенденции по полю «Заруботок»)

Таблица 1.2. Базовый тестовый пример

| ID тестового примера | Статус         | Заруботок, (\$) | Величина отчислений, (\$) | Ожидаемые результаты: причитающийся налог, (\$) | Реальный результат |
|----------------------|----------------|-----------------|---------------------------|-------------------------------------------------|--------------------|
| налог 1              | одинокий (-ая) | 40 000          | 0                         | 5 779                                           | 5 779              |
| тенденция 1          | одинокий (-ая) | 43 000          | 0                         | тенденция к возрастанию                         | 6 619              |
| тенденция 2          | одинокий (-ая) | 46 000          | 0                         |                                                 | 7 459              |
| тенденция 3          | одинокий (-ая) | 49 000          | 0                         |                                                 | 8 299              |
| тенденция 4          | одинокий (-ая) | 52 000          | 0                         |                                                 | 9 139              |
| тенденция 5          | одинокий (-ая) | 55 000          | 0                         |                                                 | 9 979              |
| тенденция 6          | одинокий (-ая) | 58 000          | 0                         |                                                 | 10 819             |
| тенденция 7          | одинокий (-ая) | 61 000          | 0                         |                                                 | 11 659             |
| тенденция 8          | одинокий (-ая) | 64 000          | 0                         |                                                 | 12 499             |

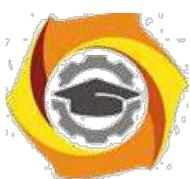
**Стадия 4: Инвентаризация** (Определить различные категории данных и создать тесты для каждого элемента категории);

Провести инвентаризацию по одному полю данных вашего приложения, т.е. составить серии тестов прорабатывающую все «ветви» для этого поля приложения. (Ниже



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

приводится таблица с серией тестов делающих *инвентаризацию* по полю «Статус» приложения «Налоговый калькулятор»)



**Таблица 1.3. Тестовый пример, использующий инвентарный список Статус**

| ID тестового примера | Статус                                             | Заработок, (\$) | Величина отчислений, (\$) | Ожидаемые результаты: причитающийся налог, (\$) | Реальный результат |
|----------------------|----------------------------------------------------|-----------------|---------------------------|-------------------------------------------------|--------------------|
| налог 1              | одинокий (-ая)                                     | 40 000          | 44 000                    | 5 779                                           |                    |
| налог 2              | женат (замужняя) с независимой декларацией доходов | 40 000          | 3 675                     | 6 537                                           |                    |
| налог 3              | женат (замужняя) с совместной декларацией доходов  | 40 000          | 7 350                     | 4 061                                           |                    |
| налог 4              | глава семьи                                        | 40 000          | 6 450                     | 4 616                                           |                    |
| налог 5              | вдовец (вдова)                                     | 40 000          | 7 350                     | 4 481                                           |                    |

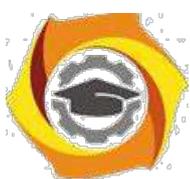
**Стадия 5: Комбинирование элементов инвентарных списков** (Скомбинировать различные входные данные по разным полям);

Разработать и затем провести тестовые примеры комбинирующих в себе различные элементы 2-х инвентарных списков из стадии 4.

**Таблица 1.5. Тестовый пример с комбинацией инвентаризации**

| ID тестового примера | Статус                                             | Входные данные  |                            |                                        |                                       |                             |                                   | Статьи отчислений | Выходные данные            |                                                    |                                                                |                                  |                         |                        |
|----------------------|----------------------------------------------------|-----------------|----------------------------|----------------------------------------|---------------------------------------|-----------------------------|-----------------------------------|-------------------|----------------------------|----------------------------------------------------|----------------------------------------------------------------|----------------------------------|-------------------------|------------------------|
|                      |                                                    | Заработок, (\$) | Материально-зависимые лица | Налог на медицину и стоматологию, (\$) | Государственный и местный налог, (\$) | Налог на недвижимость, (\$) | Налог на движимое имущество, (\$) |                   | Процент по закладной, (\$) | Пожертвования благотворительным организациям, (\$) | Неиспользованные затраты на содержание нового автомобиля, (\$) | Плата за подготовку налога, (\$) | Прочие отчисления, (\$) | Сумма отчислений, (\$) |
| налог 15             | одинокий (-ая)                                     | 40000           | 0                          | 0                                      | 2200                                  | 1890                        | 80                                | 2800              | 500                        | 100                                                | 50                                                             | 0                                | 7 470                   | 4 911                  |
| налог 16             | женат (замужняя) с независимой декларацией доходов | 60000           | 0                          | 0                                      | 2200                                  | 1890                        | 80                                | 2800              | 500                        | 100                                                | 50                                                             | 0                                | 7 470                   | 11 073                 |

**Стадия 6: Граничные оценки** (Оценить поведение приложения на границах данных и при переходе через них);



Разработать серию тестов, оценивающую границы различных данных. В качестве «границ» могут служить:

11. минимальные и максимальные значения диапазона данных;
12. минимальный и максимальный размер поля (например, минимальное и максимальное количество введённых в поле символов);
13. минимальный и максимальный размер буфера (памяти);
14. значение данных при переходе через которое приложение должно вести себя иначе.

Так же напомним, что общее правило тестирования границ – создать три тестовых примера, чтобы охватить следующие значения:

8. граничное значение (*значение на границе*);
9. граничное значение – 1 (*значение слева от границы*);
10. граничное значение + 1 (*значение справа от границы*);

(В приведённом ниже первом примере тестируются две границы, при переходе через которые происходит смена налоговой категории налогоплательщика. Первая граница – это значение заработка 139.519\$, при переходе через неё налоговая категория меняется с 31% на 36%. Вторая граница – это заработок в 292.749\$, при его увеличении, так же происходит изменение процентов собираемого налога с 36% на 39,6%.)

Таблица 1.8. Тестовый пример, использующий список налоговых ставок

| ID тестового примера | Статус         | Зарплата (\$) | Материально зависимые лица | Отчисления (\$) | Ожидаемые результаты: причитающийся налог, (\$) | Реальные результаты | Примечание: величина облагаемой налогом прибыли, (\$) | Примечание: м вычисления на |
|----------------------|----------------|---------------|----------------------------|-----------------|-------------------------------------------------|---------------------|-------------------------------------------------------|-----------------------------|
| налог 17             | одинокий (-ая) | 107 200       | 0                          | 0               | 25 681                                          |                     | 100 000                                               | налоговая кате "31%"        |
| налог 18             | одинокий (-ая) | 139 519       | 0                          | 0               | 35 787                                          |                     | 132 599                                               | налоговая кате "31%"        |
| налог 19             | одинокий (-ая) | 139 520       | 0                          | 0               | 35 787                                          |                     | 132 600                                               | налоговая кате "36%"        |
| налог 20             | одинокий (-ая) | 292 749       | 0                          | 0               | 91 857                                          |                     | 288 349                                               | налоговая кате "36%"        |
| налог 21             | одинокий (-ая) | 292 750       | 0                          | 0\$             | 91 857                                          |                     | 288 350                                               | налоговая кате "39,6%"      |

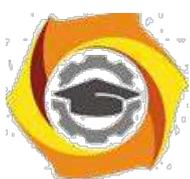


Таблица 1.12. Тест, проверяющий нулевую границу

| ID тестового примера | Статус         | Заработок, (\$) | Материально зависимые лица | Величина отчислений, (\$) | Ожидаемые результаты причитающийся налог |
|----------------------|----------------|-----------------|----------------------------|---------------------------|------------------------------------------|
| налог 26             | одиноким (-ая) | 0               | 0                          | 0                         | 0                                        |
| налог 27             | одиноким (-ая) | -1              | 0                          | 0                         | Ошибка                                   |
| налог 28             | одиноким (-ая) | 0               | 0                          | -1                        | Ошибка                                   |
| налог 29             | одиноким (-ая) | 0               | -1                         | 0                         | Ошибка                                   |
| налог 30             | одиноким (-ая) | 7 200           | 0                          | 0                         | 0                                        |

**Стадия 7: Ошибочные данные** (Оценить поведение системы при вводе неправильных данных);

Разработать и провести тесты, оценивающие поведение приложения при вводе ошибочных данных.

На этой стадии обычно создаются тесты следующих категорий:

- данные не вводятся в поля вообще для того, чтобы отследить поведение приложения: «вылетит» ли оно или выведет сообщение об ошибке;
- вводятся не верные числовые данные (отрицательные значения в полях, предназначенных по умолчанию для ввода положительных чисел или буквенно-символьные комбинации символов);
- вводятся данные какого-либо формата, который для такого типа данных считается не допустимым;
- используются не обычные комбинации данных;
- проверяется использование нулевого значения, если это ещё не сделано в предыдущих тестах.

**Стадия 8: Создание напряжений** (Попытаться вывести систему из строя);

Создайте тест, проверяющий, как работает приложение при напряжениях в среде.

9. именно, проверьте и зафиксируйте, сколько требуется приложению времени на выполнение базового теста, когда в среде работает только оно одно. И сколько требуется времени на базовый тест одному приложению, когда в среде параллельно запущено 15-20 тестируемых приложений.

Контрольные вопросы

- Перечислите основные стадии наращиваемого подхода к тестированию.
- Опишите стадию 1 «Изучение»: Цель и описание этой стадии, Основные формы изучения



- И Опишите стадию 2 «Базовое тестирование»: Цель и описание этой стадии, Основные источники определения ожидаемых результирующих данных
- И Опишите стадию 3 «Анализ тенденций»: Цель и описание этой стадии, Условия при выполнении одного из которых она проводится
- И Опишите стадию 4 «Инвентаризация»: Цель и описание этой стадии
- И Опишите стадию 5 «Комбинирование элементов инвентарных списков»: Цель и описание этой стадии, Два подхода к комбинированию.
- И Опишите стадию 6 «Граничные оценки»: Цель и описание этой стадии, Примеры пределов в зависимости от типов данных.
- И Опишите стадию 7 «Ошибочные данные»: Цель и описание этой стадии, Возможные категории тестов с ошибочными данными.
- И Опишите стадию 8 «Создание напряжений»: Цель и описание этой стадии, Используемые для создания напряжений тесты



## Лабораторная работа № 5. Тестирование программ методами белого ящика

**Цель работы:** Усвоение студентами методов тестирования логики программы, формализованного описания результатов тестирования и стандартов по составлению схем программ.

### · Краткие теоретические сведения

Тестирование программного обеспечения охватывает целый ряд видов деятельности, аналогичных последовательности процессов разработки программного обеспечения. В него входят /1/:

- а) постановка задачи для теста,
- б) проектирование теста,
- в) написание тестов,
- г) тестирование тестов,
- д) выполнение тестов,
- е) изучение результатов тестирования.

Решающую роль играет проектирование тестов. Возможны разные подходы к проектированию тестов. Первый состоит в том, что тесты проектируются на основе внешних спецификаций программ и модулей, либо спецификаций сопряжения программы или модуля. Программа при этом рассматривается как черный ящик (стратегия ‘черного ящика’). Существо такого подхода - проверить соответствует ли программа внешним спецификациям. При этом логика модуля совершенно не принимается во внимание.

Второй подход основан на анализе логики программы (стратегия ‘белого ящика’). Существо подхода - в проверке каждого пути, каждой ветви алгоритма. При этом внешняя спецификация во внимание не принимается.

Ни один из этих подходов не является оптимальным. Из анализа существа первого подхода ясно, что его реализация сводится к проверке всех возможных комбинаций значений на входе программы. Тестирование любой программы для всех значений входных данных невозможно, так как их бесконечное множество, поэтому ограничиваются меньшим. При этом исходят из максимальной отдачи теста по сравнению

· затратами на его создание. Она измеряется вероятностью того, что тест выявит ошибки, если они имеются в программе. Затраты измеряются временем и стоимостью подготовки, выполнения и проверки результатов теста.

Проанализируем теперь второй подход к тестированию. Даже если предположить, что выполнены тесты для всех путей программы, нельзя с полной уверенностью утверждать, что модуль не содержит ошибок.

Очевидное основание этого утверждения состоит в том, что выполнение всех путей не гарантирует соответствия программы ее спецификациям. Допустим, если требовалось написать программу для вычисления кубического корня, а программа фактически вычисляет корень квадратный, то программа будет совершенно неправильной, даже если



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

проверить все пути. Вторая проблема - отсутствующие пути. Если программа реализует спецификации не полностью (например, отсутствует такая специализированная функция,



как проверка на отрицательное значение входных данных программы вычисления квадратного корня), никакое тестирование существующих путей не выявит такой ошибки. И, наконец, проблема зависимости результатов тестирования от входных данных. Путь может правильно выполняться для одних данных и неправильно для других. Например, если для определения равенства 3 чисел программируется выражение вида:

$$IF (A+B+C)/3=A,$$

то оно будет верным не для всех значений А, В и С (ошибка возникает в том случае, когда из двух значений В или С одно больше, а другое на столько же меньше А). Если концентрировать внимание только на тестировании путей, нет гарантии, что эта ошибка будет выявлена.

Таким образом, полное тестирование программы невозможно. Тест для любой программы будет обязательно неполным, то есть тестирование не гарантирует полное отсутствие ошибок в программе. Стратегия проектирования тестов заключается в том, чтобы попытаться уменьшить эту неполноту насколько это возможно.

## 2.1 Методы стратегии ‘белого ящика’

Тестирование по принципу белого ящика характеризуется степенью, какой тесты выполняют или покрывают логику (исходный текст программы).

### 2.1.1 Метод покрытия операторов

Целью этого метода тестирования является выполнение каждого оператора программы хотя бы один раз.

Пример:

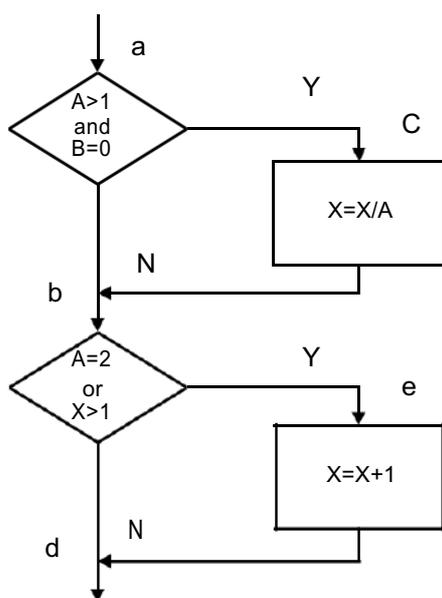


Рисунок 5.1

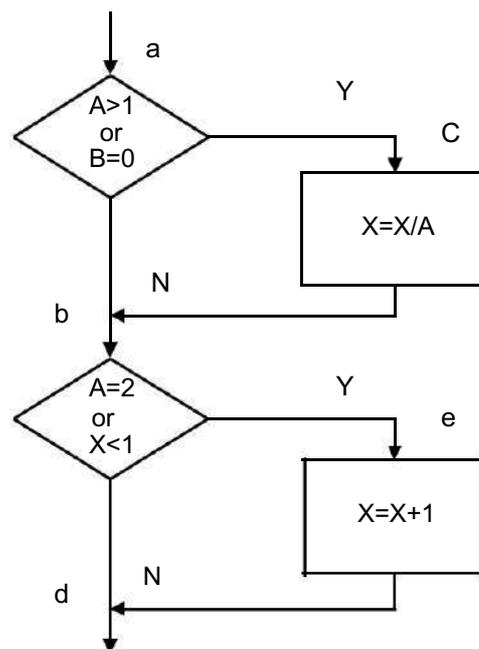


Рисунок 5.2



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

· этой программе можно выполнить каждый оператор, записав один единственный тест, который реализовал бы путь *ase*. Т.е., если бы на входе было:  $A=2$ ,  $B=0$ ,  $X=3$ ,



каждый оператор выполнялся бы один раз. Но этот критерий на самом деле хуже, чем он кажется на первый взгляд. Пусть в первом условии вместо “and” → “or” и во втором вместо “ $x > 1$ ” → “ $x < 1$ ” (блок-схема правильной программы приведена на рисунке 5.1, а неправильной - на рисунке 5.2). Результаты тестирования приведены в таблице 5.1. Обратите внимание: ожидаемый результат определяется по алгоритму на рисунке 5.1, а фактический - по алгоритму рисунка 5.2, поскольку определяется чувствительность метода тестирования к ошибкам программирования. Как видно из этой таблицы, ни одна из внесенных в алгоритм ошибок не будет обнаружена.

Таблица 5.1 - Результат тестирования методом покрытия операторов

| Тест          | Ожидаемый результат | Фактический результат | Результат тестирования |
|---------------|---------------------|-----------------------|------------------------|
| A=2, B=0, X=3 | X=2,5               | X=2,5                 | неуспешно              |

### 2.2 Метод покрытия решений (покрытия переходов)

Более сильный метод тестирования известен как покрытие решений (покрытие переходов). Согласно данному методу каждое направление перехода должно быть реализовано по крайней мере один раз.

Покрытие решений обычно удовлетворяет критерию покрытия операторов. Поскольку каждый оператор лежит на некотором пути, исходящем либо из оператора перехода, либо из точки входа программы, при выполнении каждого направления перехода каждый оператор должен быть выполнен.

Для программы приведенной на рисунке 2.2 покрытие решений может быть выполнено двумя тестами, покрывающими пути {ace, abd}, либо {acd,abe}. Пути {acd,abe} покроем, выбрав следующие исходные данные: {A=3, B=0, X=3} и {A=2, B=1, X=1} (результаты тестирования - в таблице 2.2).

Таблица 5.2 - Результат тестирования методом покрытия решений

| Тест          | Ожидаемый результат | Фактический результат | Результат тестирования |
|---------------|---------------------|-----------------------|------------------------|
| A=3, B=0, X=3 | X=1                 | X=1                   | неуспешно              |
| A=2, B=1, X=1 | X=2                 | X=1,5                 | успешно                |

### 2.3 Метод покрытия условий

Лучшие результаты по сравнению с предыдущими может дать метод покрытия условий. В этом случае записывается число тестов, достаточное для того, чтобы все возможные результаты каждого условия в решении выполнялись по крайней мере один раз.

9) предыдущем примере имеем четыре условия: {A>1, B=0}, {A=2, X>1}. Следовательно, требуется достаточное число тестов, такое, чтобы реализовать ситуации,



где  $A > 1$ ,  $A = 1$ ,  $B = 0$  и  $B = 1$  в точке  $a$  и  $A = 2$ ,  $A = 1$ ,  $X > 1$  и  $X = 1$  в точке  $b$ . Тесты, удовлетворяющие критерию покрытия условий и соответствующие им пути:

- а)  $A = 2$ ,  $B = 0$ ,  $X = 4$                       *ace*
- б)  $A = 1$ ,  $B = 1$ ,  $X = 0$                       *abd*

Таблица 5.3 - Результаты тестирования методом покрытия условий

| Тест                        | Ожидаемый результат | Фактический результат | Результат тестирования |
|-----------------------------|---------------------|-----------------------|------------------------|
| $A = 2$ , $B = 0$ , $X = 4$ | $X = 3$             | $X = 3$               | неуспешно              |
| $A = 1$ , $B = 1$ , $X = 0$ | $X = 0$             | $X = 1$               | успешно                |

#### 2.4 Критерий решений (условий)

Критерий покрытия решений/условий требует такого достаточного набора тестов, чтобы все возможные результаты каждого условия в решении выполнялись по крайней мере один раз, все результаты каждого решения выполнялись по крайней мере один раз и, кроме того, каждой точке входа передавалось управление по крайней мере один раз.

##### Два теста метода покрытия условий

- а)  $A = 2$ ,  $B = 0$ ,  $X = 4$                       *ace*
- б)  $A = 1$ ,  $B = 1$ ,  $X = 0$                       *abd*

отвечают и критерию покрытия решений/условий. Это является следствием того, что одни условия приведенных решений скрывают другие условия в этих решениях. Так, если условие  $A > 1$  будет ложным, транслятор может не проверять условия  $B = 0$ , поскольку при любом результате условия  $B = 0$ , результат решения  $((A > 1) \& (B = 0))$  примет значение *ложь*. Следовательно, недостатком критерия покрытия решений/условий является невозможность его применения для выполнения всех результатов всех условий.

Другая реализация рассматриваемого примера приведена на рисунке 5.3. Многоусловные решения исходной программы разбиты на отдельные решения и переходы. Наиболее полное покрытие тестами в этом случае выполняется так, чтобы выполнялись все возможные результаты каждого простого решения. Для этого нужно покрыть пути НЛР (тест  $A = 2, B = 0, X = 4$ ), НМКТ (тест  $A = 3, B = 1, X = 0$ ), НКТ (тест  $A = 0, B = 0, X = 0$ ), НКР (тест  $A = 0, B = 0, X = 2$ ).

Протестировав алгоритм на рисунке 5.3, нетрудно убедиться в том, что критерии покрытия условий и критерии покрытия решений/условий недостаточно чувствительны к ошибкам в логических выражениях.

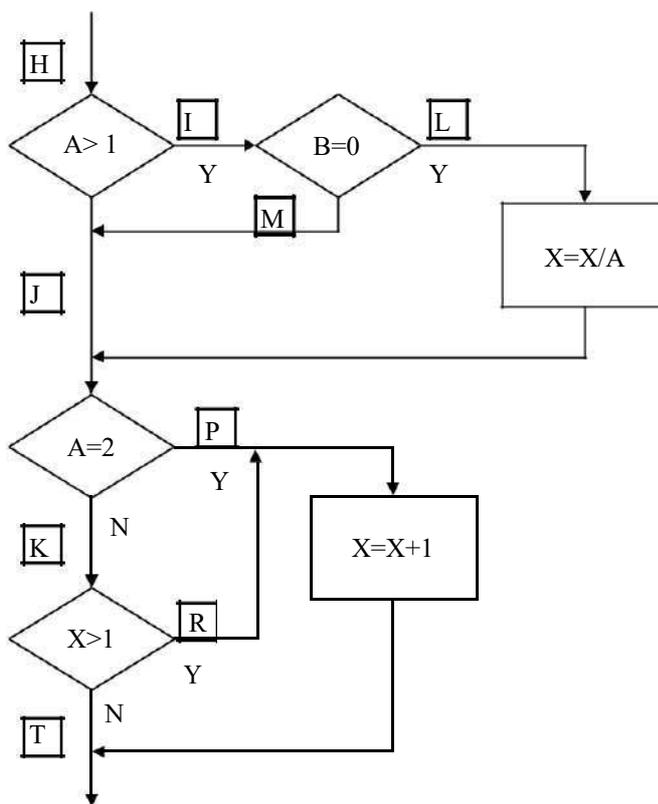
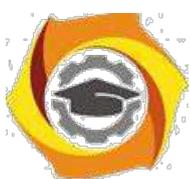


Рисунок 5.3

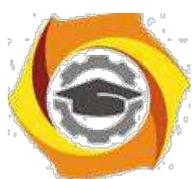
### 2.5 Метод комбинаторного покрытия условий.

Критерием, который решает эти и некоторые другие проблемы, является комбинаторное покрытие условий. Он требует создания такого числа тестов, чтобы все возможные комбинации результатов условия в каждом решении выполнялись по крайней мере один раз. Набор тестов, удовлетворяющих критерию комбинаторного покрытия условий, удовлетворяет также и критериям покрытия решений, покрытия условий и покрытия решений/условий.

По этому критерию в рассматриваемом примере должны быть покрыты тестами следующие восемь комбинаций:

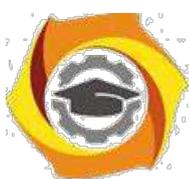
- а)  $A > 1, B = 0$ ;
- б)  $A > 1, B \neq 0$ ;
- в)  $A = 1, B = 0$ ;
- г)  $A = 1, B \neq 0$ ;
- д)  $A = 2, X > 1$ ;
- е)  $A = 2, X \leq 1$ ;
- ж)  $A \neq 2, X > 1$ ;
- з)  $A \neq 2, X \leq 1$ ;

Для того чтобы протестировать эти комбинации, необязательно использовать все 8 тестов. Фактически они могут быть покрыты четырьмя тестами:



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- A=2, B=0, X=4 {покрывает а, д};



- иA=2, B=1, X=1 {покрывает б, е};
- иA=0,5, B=0, X=2 {покрывает в, ж};
- иA=1, B=0, X=1 {покрывает г, з}.

Таблица 5.4 - Результаты тестирования методом комбинаторного покрытия условий

| Тест              | Ожидаемый результат | Фактический результат | Результат тестирования |
|-------------------|---------------------|-----------------------|------------------------|
| A=2, B=0, X=4     | X=3                 | X=3                   | неуспешно              |
| A=2, B=1, X=1     | X=2                 | X=1,5                 | успешно                |
| A=0,5 B=0,<br>X=2 | X=3                 | X=4                   | успешно                |
| A=1, B=0, X=1     | X=1                 | X=1                   | неуспешно              |

#### Методика выполнения лабораторной работы

- Написать программу, реализующую заданный преподавателем алгоритм обработки данных.
- Отобразить алгоритм решения задачи в виде схемы программы
- Обозначить буквами или цифрами ветви алгоритма
- Провести тестирование программного продукта рассмотренными методами.
- Выписать пути алгоритма, которые должны быть проверены тестами для рассматриваемого метода тестирования.
- Записать тесты, которые позволят пройти по путям алгоритма,.
- Протестировать разработанную Вами программу. Результаты оформить в виде таблиц (таблицы 2.1-2.4).
- Оформить отчет по лабораторной работе.

#### 4 Содержание отчета

- Цель работы.
  - Программа решения поставленной Вам задачи.
  - Схема программы (см. пп.2,3).
  - Таблицы тестирования программы (п.7).
- В Выводы по результатам тестирования (целью тестирования является обнаружение ошибок в программе).



### Контрольные вопросы

- Какие пункты содержит спецификация программного обеспечения?
- Что такое управляющий граф программы?
- Оценка степени тестируемости ПО.
- Критерии структурного тестирования.
- Построение управляющего графа программы.
- Метод покрытия операторов
- Метод покрытия решений (покрытия переходов)
- Метод покрытия условий
- Критерий решений (условий)
- Метод комбинаторного покрытия условий



## Лабораторная работа № 6. Нарастиваемое тестирование приложения «Калькулятор» в среде LAZARUS

### Цель лабораторной работы

Совершенствовать навыки применения методики нарастиваемого подхода к тестированию. Используя «Нарастиваемый подход к тестированию» создать тестовые примеры и провести тестирование разработанной на предыдущем занятии программы «Калькулятор».

### В Краткие теоретические сведения

«Нарастиваемый подход к тестированию» содержит 8 стадий:

#### **Стадия 1: Изучение** (Ознакомиться с приложением);

Поработать какое-то время с тестируемым приложением, изучить его...

**Стадия 2: Базовый тест** (Разработать и реализовать один простой тестовый пример);

Разработать при помощи MS-Excel таблицу для базового тестового примера, вбить в неё базовый тестовый пример, провести тест приложения на его основе и указать в таблице полученный при тестировании результат. Сравнить полученный результат с ожидаемым результатом, сделать вывод и отметку в таблице, о том пройден тест или нет.

**Стадия 3: Анализ тенденций** (Определить, работает ли приложение так, как было задумано, когда ещё нельзя предварительно оценить реальные результаты);

Разработать 2-3 серии из 6-10 тестов и провести анализ тенденций при работе приложения. Результаты и выводы зафиксировать в файле электронной таблицы. (Ниже приводится таблица с одной серией тестов для анализа 1 тенденции по полю «Зарботок»)

**Стадия 4: Инвентаризация** (Определить различные категории данных и создать тесты для каждого элемента категории);

Провести инвентаризацию по одному полю данных вашего приложения, т.е. составить серии тестов прорабатывающую все «ветви» для этого поля приложения.

**Стадия 5: Комбинирование элементов инвентарных списков** (Скомбинировать различные входные данные по разным полям);

Разработать и затем провести тестовые примеры комбинирующих в себе различные элементы 2-х инвентарных списков из стадии 4.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Стадия 6: Граничные оценки** (Оценить поведение приложения на границах данных и при переходе через них);



Разработать серию тестов, оценивающую границы различных данных. В качестве «границ» могут служить:

- В минимальные и максимальные значения диапазона данных;
- В минимальный и максимальный размер поля (например, минимальное и максимальное количество введённых в поле символов);
- В минимальный и максимальный размер буфера (памяти);
- В значение данных при переходе через которое приложение должно вести себя иначе.

Так же напомним, что общее правило тестирования границ – создать три тестовых примера, чтобы охватить следующие значения:

- к граничное значение (*значение на границе*);
- к граничное значение – 1 (*значение слева от границы*);
- к граничное значение + 1 (*значение справа от границы*);

**Стадия 7: Ошибочные данные** (Оценить поведение системы при вводе неправильных данных);

Разработать и провести тесты, оценивающие поведение приложения при вводе ошибочных данных.

На этой стадии обычно создаются тесты следующих категорий:

- О данные не вводятся в поля вообще для того, чтобы отследить поведение приложения: «вылетит» ли оно или выведет сообщение об ошибке;
- О вводятся не верные числовые данные (отрицательные значения в полях, предназначенных по умолчанию для ввода положительных чисел или буквенно-символьные комбинации символов);
- О вводятся данные какого-либо формата, который для такого типа данных считается не допустимым;
- О используются не обычные комбинации данных;
- О проверяется использование нулевого значения, если это ещё не сделано в предыдущих тестах.

**Стадия 8: Создание напряжений** (Попытаться вывести систему из строя);

Создайте тест, проверяющий, как работает приложение при напряжениях в среде. В именно, проверьте и зафиксируйте, сколько требуется приложению времени на выполнение базового теста, когда в среде работает только оно одно. И сколько требуется времени на базовый тест одному приложению, когда в среде параллельно запущено 15-20 тестируемых приложений.

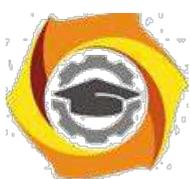
**ЗАДАНИЕ:**



Используя “Наращиваемый подход к тестированию” создать тестовые примеры и провести тестирование разработанной на предыдущем занятии программы.

Контрольные вопросы

- В Дополнительные методы тестирования:
- В Какие существуют ещё методы, кроме инвентаризации?
- В Что позволяет протестировать подход, основанный на теории графов ?
- В Какие ещё нужны дополнительные тесты?
- В Какие дополнительные дисциплины должно включать в себя качественное тестирование?
- В Основные преимущества проведения тестирования даже при нехватке времени на него,
- В Основная тактика тестирования при нехватке времени,
- В Как определить наиболее важные для тестирования области?
- В Классификация выборочных методов.
- В Основные понятия тестирования: концепция тестирования, подходы.
- В Различия тестирования и отладки.
- В Фазы и технология тестирования.
- В Проблемы тестирования.
- В Критерии выбора тестов: структурные, функциональные, стохастические.
- В Критерии выбора тестов: мутационные, оценки покрытия проекта.



## Лабораторная работа № 7. Задача Майерса. Развитие навыков интуитивного тестирования

### Цель лабораторной работы

Совершенствовать навыки применения методов тестирования на примере задачи Майерса. Изучить интуитивный подход к тестированию.

### б) Краткие теоретические сведения

повседневной жизни мы постоянно сталкиваемся с необходимостью тестирования программ, с которыми нам приходится работать. Это свежие версии текстовых и графических редакторов, почтовых клиентов и служебных утилит, новые модели мобильных телефонов и обновленный дизайн любимых сайтов. Конечно, такое персональное тестирования чаще всего проводится неформально, в процессе повседневной работы или развлечения, без оформления тест-кейсов и баг-репортов. Тем не менее, в результате практически каждый человек в сознательном возрасте приобретает определенные интуитивные навыки к тестированию программ и программно-управляемых устройств.

своей книге «Искусство тестирования программ» Гленфорд Майерс предложил простое задание, позволяющее определить интуитивные способности человека к тестированию ПО.

Исходные данные:

Имеется гипотетическая (существующая только в воображении испытуемого) программа. Программа считывает три целочисленных значения из диалогового окна ввода данных. Эти значения интерпретируются как длины сторон треугольника. Программа выводит сообщение о том, каким является данный треугольник — неравносторонним, равнобедренным или равносторонним. Примерно вот так:

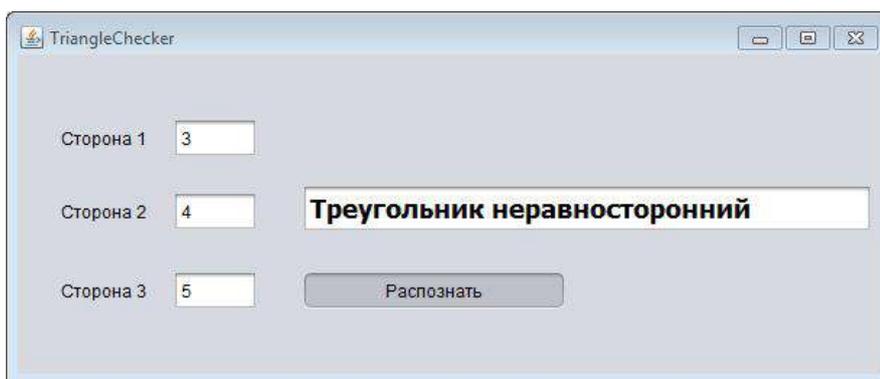


Рис. 7. 3

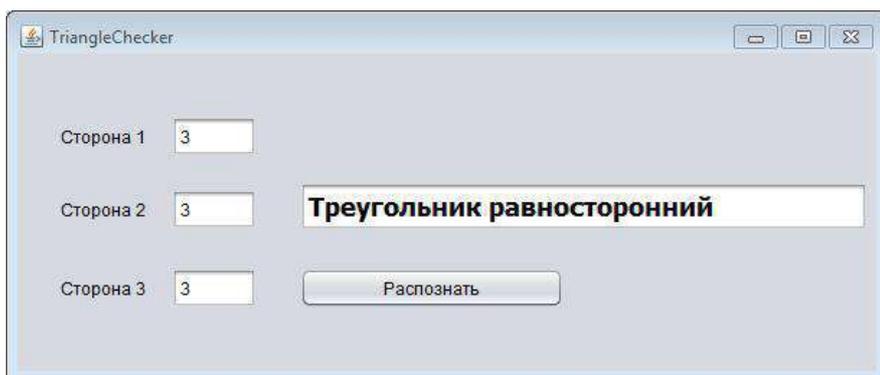
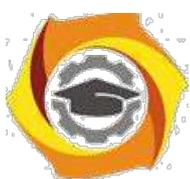


Рис. 7.1

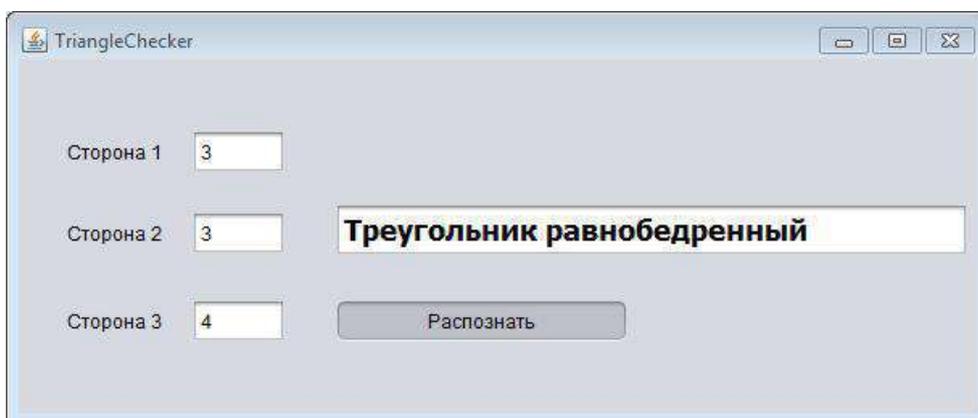


Рис. 7.2

Напомним, что треугольник называется разносторонним (неравносторонним), если все три его стороны не равны друг другу; равнобедренным — если две его стороны равны, а равносторонним — если все три его стороны равны.

## Виды треугольников



*Равносторонний*



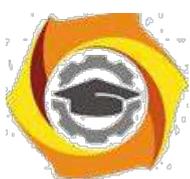
*Равнобедренный*



*Неравносторонний*

Рис. 7.4





Лабораторная работа №1. Ключ к расчету результата:

Проставьте в правом поле ответы на вопросы (да/нет).

|                                                                                                                                                                                                                                                                                                                  |  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| к Есть ли у вас тест-кейс, представляющий действительный неравносторонний треугольник? Учтите, что наличие таких наборов, как 1, 2, 3 или 2, 5, 10, еще не дает права ответить утвердительно, поскольку треугольников с такими сторонами не существует.                                                          |  |
| к Есть ли у вас тест-кейс, представляющий действительный равносторонний треугольник?                                                                                                                                                                                                                             |  |
| к Есть ли у вас тест-кейс, представляющий действительный равнобедренный треугольник? Учтите, что такие наборы значений, как 2, 2, 4, не засчитываются, поскольку представляют несуществующие треугольники.                                                                                                       |  |
| к Есть ли у вас хотя бы три тест-кейса, представляющих три действительных равнобедренных треугольника, которые соответствуют различным перестановкам двух равных сторон (например, 3, 3, 4; 3, 4, 3; 4, 3, 3)?                                                                                                   |  |
| к Есть ли у вас тест-кейс, в котором одна из сторон имеет нулевую длину?                                                                                                                                                                                                                                         |  |
| к Есть ли у вас тест-кейс, в котором длине одной из сторон соответствует отрицательное значение?                                                                                                                                                                                                                 |  |
| к Есть ли у вас тест-кейс, включающий такой набор из трех целых положительных чисел, в котором сумма двух из них равна третьему числу? Иными словами, вывод программой сообщения о том, что набор значений 1, 2, 3 представляет неравносторонний треугольник, будет означать, что в программе содержится ошибка. |  |
| к Есть ли у вас хотя бы три тест-кейса категории 7 для трех перестановок трех положительных чисел, представляющих треугольник, в котором длина одной из сторон равна сумме длин двух других (например, 1, 2, 3; 1, 3, 2,5, 1, 2)?                                                                                |  |
| к Есть ли у вас тест-кейс, включающий три положительных целых числа, причем сумма двух из них меньше третьего (например: 1, 2, 4 или 12, 15, 30)?                                                                                                                                                                |  |
| к Есть ли у вас хотя бы три тест-кейса категории 9, которые охватывают три перестановки (например, 1, 2, 4; 1, 4, 2; 4, 1, 2)?                                                                                                                                                                                   |  |
| к Есть ли у вас тест-кейс, в котором все стороны треугольника имеют нулевую длину (т.е. 0, 0, 0)?                                                                                                                                                                                                                |  |
| к Есть ли у вас хотя бы один тест-кейс, в котором заданы нецелочисленные значения (например, 2.5, 3.5, 5.5)?                                                                                                                                                                                                     |  |
| к Есть ли у вас хотя бы один тест-кейс, в котором задано неверное количество значений (например, указано два целых числа вместо трех)?                                                                                                                                                                           |  |
| к Были ли для каждого тест-кейса указаны не только данные на входе, но и ожидаемый результат на выходе программы? Именно для этого предназначалась последняя (никак не озаглавленная) колонка таблицы.                                                                                                           |  |
|                                                                                                                                                                                                                                                                                                                  |  |

Посчитайте количество ответов «да» и оцените результат:

**0 - 3.** Интуитивные навыки тестирования у вас развиты недостаточно. Но не отчаивайтесь, приложив немного усилий, вы можете значительно улучшить свой результат.

**3 - 6.** В целом неплохо. В этом диапазоне находятся результаты большинства людей, имеющих отношение к IT-технологиям.



**6 - 9.** Отлично! Такой результат с первого раза, как правило, демонстрируют только профессиональные разработчики ПО.

**9 - 14.** Супер! Вы либо тестировщик по специальности, либо выполняете задание не в первый раз.

#### Контрольные вопросы

8. Понятие тестирования ПО. Основные определения.
9. Цели и принципы тестирования.
10. Основная цель тестирования.
11. Уровень доверия, корректное поведение, реальное окружение.
12. Тестирование и качество.
13. Уровни восприятия тестирования в компании.
14. Участники тестирования, их роль, квалификация и обязанности



## Лабораторная работа №8. Функциональное тестирование программ

**Цель работы:** провести функциональное тестирование разработанного программного средства в соответствии с заданным вариантом

### в Краткие теоретические сведения

Процесс тестирования состоит из трёх этапов:

25. Проектирование тестов.
26. Исполнение тестов.
27. Анализ полученных результатов.

На первом этапе решается вопрос о выборе некоторого подмножества множества тестов, которое сможет найти наибольшее количество ошибок за наименьший промежуток времени. На этапе исполнения тестов проводят, запуск тестов и отлавливают ошибки в тестируемом программном продукте.

Виды тестов

Функциональные тесты составляются на уровне спецификации, до решения задачи. Будущий алгоритм рассматривается как «черный ящик» - функция с неизвестной (или не рассматриваемой) структурой, преобразующая входы в выходы. Суть функциональных тестов: каким бы способом ни решалась задача, при заданных входных значениях должны получиться соответствующие выходные значения.

Структурные тесты составляются для проверки логики решения, или логики работы уже готового алгоритма. Логика определяется последовательностью операций, их условным выполнением или повторением (т.е. композицией базовых конструкций). Совокупность структурных тестов должна обеспечить проверку каждой из таких конструкций.

Чаще всего совокупность тщательно составленных функциональных тестов покрывает множество структурных тестов.

Приведенные понятия различаются тем, что первое рассматривает программу только с точки зрения входов и выходов, тогда как второе относится к ее структуре; но оба понятия не касаются процесса организации тестирования.

Общая последовательность разработки тестов

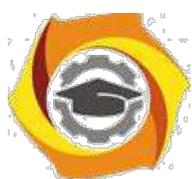
Наиболее рациональная процедура заключается в том, что сначала разрабатываются функциональные тесты, а затем – структурные.

Функциональное тестирование (тестирование «черного ящика»)

При функциональном тестировании выявляются следующие категории ошибок:

9. некорректность или отсутствие функций;
10. ошибки интерфейса;
11. ошибки в структурах данных;
12. ошибки машинных характеристик (нехватка памяти и др.);
13. ошибки инициализации и завершения.

Техника тестирования ориентирована:



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

9. на сокращение необходимого количества тестовых вариантов;
10. на выявление классов ошибок, а не отдельных ошибок.



### Способы функционального тестирования

#### Разбиение на классы эквивалентности

Это самый популярный способ. Его суть заключается в разделении области входных данных программы на классы эквивалентности и разработке для каждого класса одного тестового варианта.

Класс эквивалентности – набор данных с общими свойствами, в силу чего при обработке любого набора данных этого класса задействуется один и тот же набор операторов<sup>1</sup>.

Классы эквивалентности определяются по спецификации программы. Тесты строятся в соответствии с классами эквивалентности, а именно: выбирается вариант исходных данных некоторого класса и определяются соответствующие выходные данные.

Самыми общими классами эквивалентности являются классы допустимых и недопустимых (аномальных) исходных данных. Описание класса строится как комбинация условий, описывающих каждое входное данное.

Условия допустимости или недопустимости данных задают возможные значения данных и могут описывать:

10. некоторое конкретное значение; определяется один допустимый и два недопустимых класса эквивалентности: заданное значение, множество значений меньше заданного, множество значений больше заданного;
11. диапазон значений; определяется один допустимый и два недопустимых класса эквивалентности: множество значений в границах диапазона; множество значений, выходящих за левую границу диапазона; множество значений, выходящих за правую границу диапазона;
12. множество конкретных значений; определяется один допустимый и один недопустимый класс эквивалентности: заданное множество и множество значений, в него не входящих.

Такие классы можно описать языком логики, например, языком исчисления предикатов. Описания более сложных условий и соответствующих классов (например, элементы массива должны находиться в некотором диапазоне и при этом массив не должен содержать нулевых элементов) могут быть построены на основании приведенных выше условий.

#### Анализ граничных значений

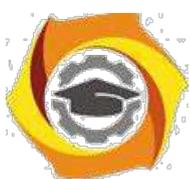
Этот способ построения тестов дополняет предыдущий и предполагает анализ значений, лежащих на границе допустимых и недопустимых данных. Построение таких тестов часто диктуется интуицией.

#### Основные правила построения тестов:

11. если условие правильности данных задает диапазон, то строятся тесты для левой и правой границы диапазона; для значений чуть левее левой и чуть правее правой границы;

---

<sup>1</sup>См. например, [1].



- У если условие правильности данных задает дискретное множество значений, то строятся тесты для минимального и максимального значений; для значений чуть меньше минимума и чуть больше максимума;
- У если используются структуры данных с переменными границами (массивы), то строятся тесты для минимального и максимального значения границ.

#### Диаграммы причин-следствий

Взаимосвязь классов эквивалентности и соответствующих им действий описывается формально в виде графа на основе автоматного подхода. Граф преобразуется

7. таблицу решений, столбцы которой в свою очередь преобразуются в тестовые варианты.

#### **Порядок выполнения работы**

По результатам лабораторных работ 1-7 разработать тестовые наборы для функционального тестирования .

Провести тестирование программы и представить результаты в виде таблицы. (Приложение 1).

Выработать рекомендации для корректировки тестируемой программы.

Представить отчет по лабораторной работе для защиты.

Защита отчета по лабораторной работе

Защита отчета по лабораторной работе заключается в предъявлении преподавателю полученных результатов, демонстрации полученных навыков в ответах на вопросы преподавателя.

#### Контрольные вопросы

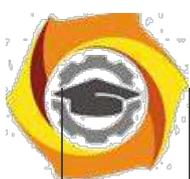
- ' Что такое тестирование ПС?
- ' Чем тестирование отличается от отладки ПС?
- ' Для чего проводится функциональное тестирование?
- ' Что такое комплексное тестирование?
- ' Каковы правила тестирования программы «как черного ящика»?
- ' Как проводится тестирования программы по принципу «белого ящика»?
- ' Что такое модульное тестирование?
- ' Как осуществляется сборка программы при модульно тестировании?

Задания и методические указания для студентов по выполнению практических работ по модулю " Методы повышения эффективности производства ".

Практические занятия являются формой аудиторных занятий. Практические работы по модулю имеют целью под руководством преподавателя на практике закрепление и углубление изученного материала и приобретение умений и навыков.

Студентам для лучшего усвоения материала рекомендуется вести запись информации, полученной во время обсуждения вопросов на практических занятиях.

#### *Тематика практических работ*



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

|    |                                                                                                                                                                                                                                                                                                                                                   |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Анализ решений (в условиях неопределенности). Анализ решений (в условиях риска).                                                                                                                                                                                                                                                                  |
| 2  | Операции над нечеткими отношениями и множествами. Определение оптимального решения при конечном горизонте планирования методом итераций по стратегиям. Метод анализа иерархий. Решение задач принятия решений на языке бинарных отношений                                                                                                         |
| 3  | Понятийный аппарат в управлении проектами. Модель «дуга – работа» (activity-on-arrow). Модель «узел – работа» (activity-on-node). Расчёт сетевой модели.                                                                                                                                                                                          |
| 4  | Проекты, ограниченные по времени. Проекты, ограниченные по ресурсам. Минимизация затрат. Формулировка задачи линейного программирования для данной модели. Имитационное моделирование. Средняя норма прибыли на инвестиции. Метод оценки по периоду окупаемости (payback method).                                                                 |
| 5  | Функции кадрового менеджмента в управлении производственной организацией. Кадровая политика и кадровые процедуры производственной организации                                                                                                                                                                                                     |
| 7. | Административные методы управления. Обеспечение безопасности труда. СОУТ. Социально-психологические методы управления. Экономические методы управления. Стили управления производственным коллективом. Профилактика производственных конфликтов. Подготовка персонала к внедрению инноваций. Управление корпоративной (организационной) культурой |

**Тема 1. Принятие инженерных решений.**

***Практическая работа №1***

Анализ решений (в условиях неопределенности). Анализ решений (в условиях риска).

**Цель занятия:** Формирование у студентов научных представлений о факторах, вызывающих стрессовые ситуации в профессиональной деятельности и о способах эмоциональной саморегуляции.

**Задание:** Провести самообследование с помощью психологических диагностических тестов, написать выводы

***Порядок выполнения:***

4. Ознакомиться с причинами стрессовых ситуаций в профессиональной деятельности.
5. Методика определения стрессоустойчивости и социальной адаптации Холмса и Раге.
6. Методика «Инвентаризация симптомов стресса»
7. Обсуждение результатов и формулирование выводов
8. Сформулировать приемы, позволяющие регулировать негативные состояния в профессиональной деятельности

***Вопросы для самоконтроля***



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Управление стрессовостью психического состояния в процессе профессиональной адаптации

Влияние стрессов на личностное и профессиональное развитие работника.

Проявление профессиональных деформаций в процессе профессиональной деятельности.

Преодоление стрессовых ситуаций в ходе ответственного выполнения производственного задания.

## ***Задания для самоподготовки.***

Подготовить презентацию по проблеме «Профессиональные деструкции».

### ***Отчет должен включать в себя***

Конспекты о причинах стресса в профессиональной деятельности  
Результаты выполнения диагностических методик.

Выводы по результатам самодиагностики и рекомендации по регулированию негативных эмоциональных состояний в профессиональной деятельности.

## ***Практическая работа №2***

Операции над нечеткими отношениями и множествами. Определение оптимального решения при конечном горизонте планирования методом итераций по стратегиям. Метод анализа иерархий. Решение задач принятия решений на языке бинарных отношений

### ***Цель работы.***

– отработка навыка решения практических задач

### ***Устные вопросы по теме занятия:***

Операции над нечеткими отношениями и множествами.

Определение оптимального решения при конечном горизонте планирования методом итераций по стратегиям.

Метод анализа иерархий.

Решение задач принятия решений на языке бинарных отношений

### ***Практическое задание:*** выполнение заданий по теме занятия

***Результатом успешного выполнения практического задания*** считается представленный в электронном виде комплексный документ, оформленный в соответствии с предъявленными требованиями.

## **Тема 2. Проектное управление.**

### ***Практическая работа №3***

Понятийный аппарат в управлении проектами. Модель «дуга – работа» (activity-on-arrow). Модель «узел – работа» (activity-on-node). Расчёт сетевой модели.

### ***Цель работы.***

– отработка навыка решения практических задач

### ***Устные вопросы по теме занятия:*** -

Автоматизация управления проектами. -

Функции персональных компьютерных систем.

-Основные инструменты распределенных интегрированных систем.

-Модель «дуга – работа» (activity-on-arrow)

-Модель «узел – работа» (activity-on-node)

-Адаптация правил построения сетей к реальности.

-Построение календарного плана и распределение ресурсов -

Проекты, ограниченные по ресурсам  
***Практическое задание:*** выполнение заданий по теме занятия.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Результатом успешного выполнения практического задания считается представленный в электронном виде комплексный документ, оформленный в соответствии с предъявленными требованиями.

## **Практическая работа №4**

Проекты, ограниченные по времени. Проекты, ограниченные по ресурсам. Минимизация затрат. Формулировка задачи линейного программирования для данной модели. Имитационное моделирование. Средняя норма прибыли на инвестиции. Метод оценки по периоду окупаемости (payback method).

*Цель работы.*

– отработка навыка решения практических задач

*Устные вопросы по теме занятия:*

Проекты, ограниченные по времени.

Проекты, ограниченные по ресурсам.

Минимизация затрат.

Формулировка задачи линейного программирования для данной модели.

Имитационное моделирование.

Средняя норма прибыли на инвестиции.

Метод оценки по периоду окупаемости (payback method).

*Практическое задание:* выполнение заданий по теме занятия

Результатом успешного выполнения практического задания считается представленный в электронном виде комплексный документ, оформленный в соответствии с предъявленными требованиями.

## **Тема 3. Управление производственным коллективом.**

### **Практическая работа №5**

Функции кадрового менеджмента в управлении производственной организацией. Кадровая политика и кадровые процедуры производственной организации

**Функции кадрового менеджмента в управлении производственной организацией.**

**Устные вопросы по теме практического занятия:**

6. Задачи кадрового менеджмента на современном этапе.

7. Универсальные функции управления персоналом: планирование, организация, координация, регулирование, контроль.

8. Конкретно-специфические функции управления производственным персоналом: административная, планирования, социальная, повышения качества деятельности, воспитательная, мотивационная, информационно-аналитическая.

**Содержание практического занятия**

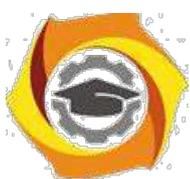
**1 этап.** Знакомство с теоретическими основами кадрового менеджмента.

**1.1.** Самостоятельное изучение устных вопросов по теме практического занятия. Использование доступа к электронным библиотечным ресурсам. Индивидуальная работа.

**1.2.** Обсуждение теоретического материала, пояснение спорных позиций. Дискуссия под руководством преподавателя.

**2 этап.** Анализ организационных структур управления персоналом производственных организаций. Работа в группах.

**2.1.** На основании анализа оргструктуры АО «УКБТМ» (схема 1) выделить какие структурные подразделения и какие функции в области управления персоналом могут выполнять. Свою позицию обосновать.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

## 2.2. Проверочная письменная работа по итогам занятия. Ответить на вопросы:

Какие задачи в области кадрового менеджмента данная структура позволяет реализовать  
б. почему?

В реализации каких задачи в области кадрового менеджмента в данной организации  
могут возникнуть проблемы и почему?

Сформулируйте предложения по исследованию указанных выше проблем. **Результатом  
практического занятия** считается умение магистратов соотносить имеющиеся  
структурные подразделения с выполняемыми управленческими функциями, выделять  
организационные проблемы в структуре управления производственным персоналом, выделять  
цели и задачи организации и координации процесса управления персоналом.

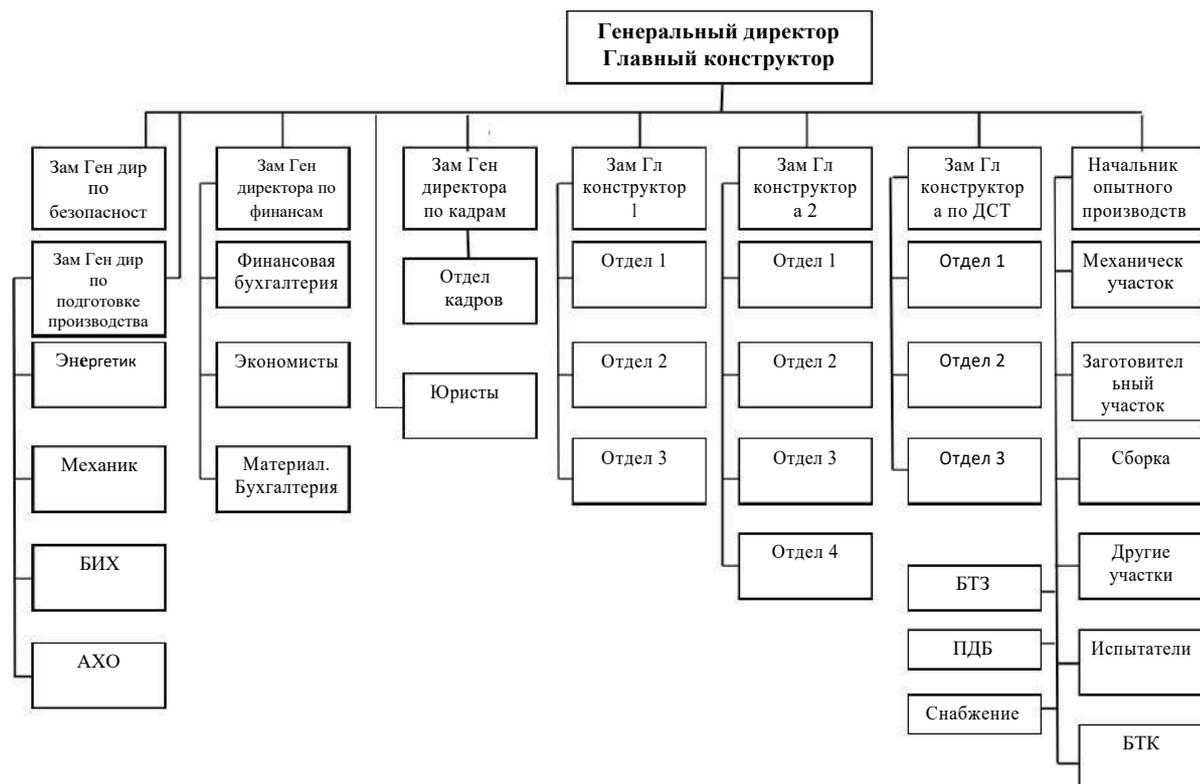


Схема 1- Организационная структура АО «УКБТМ»

### **Кадровая политика и кадровые процедуры производственной организации.**

#### **Устные вопросы по теме практического занятия:**

6. Основные группы кадровых процедур: приемы на работу; переводы на другую работу; увольнения работников; предоставление отпусков работникам; привлечение к ответственности и др.

7. Общероссийский классификатор управленческой документации (ОКУД) ОК 011-93.

8. Формы первичной учетной документации по учету труда и его оплаты.

#### **Содержание практического занятия**

**Этап 1.** Изучить теоретическую часть работы. Ознакомиться с формами первичной учетной документации по учету труда и его оплаты. Работа с правовой поисковой системой «КонсультантПлюс».

#### **Устные вопросы:**

19. Для чего предназначены унифицированные формы первичной учетной документации по учету труда и его оплаты?

20. Можно ли вносить изменения в унифицированные формы, и какие именно изменения?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

6. Что такое приказ по личному составу?
7. Какие процедуры движения кадров документируют приказами по личному составу?
8. Для чего составляется штатное расписание? Что оно включает в себя?
9. Какие документы нужны, чтобы составить штатное расписание?
10. Как и когда происходит утверждение штатного расписания?

**Этап 2.** На основании структуры организации (схема 2) создать список сотрудников в виде базы данных. Подготовить штатное расписание (Унифицированная форма № Т-3 Утверждена постановлением Госкомстата России от 05.01.2004 № 1).

**Этап 3.** Проверочная работа. Заполнить следующие унифицированные формы:

6. Унифицированная форма N Т-1. Утверждена Постановлением Госкомстата России от 05.01.2004 №1.
7. Унифицированная форма № Т-5. Утверждена постановлением Госкомстата России от 05.01.2004 № 1.
8. Унифицированная форма № Т-5а. Утверждена постановлением Госкомстата России от 05.01.2004 № 1.
9. Унифицированная форма № Т-6. Утверждена постановлением Госкомстата России от 05.01.2004 № 1.
10. Унифицированная форма № Т-7. Утверждена постановлением Госкомстата России от 05.01.2004 № 1.

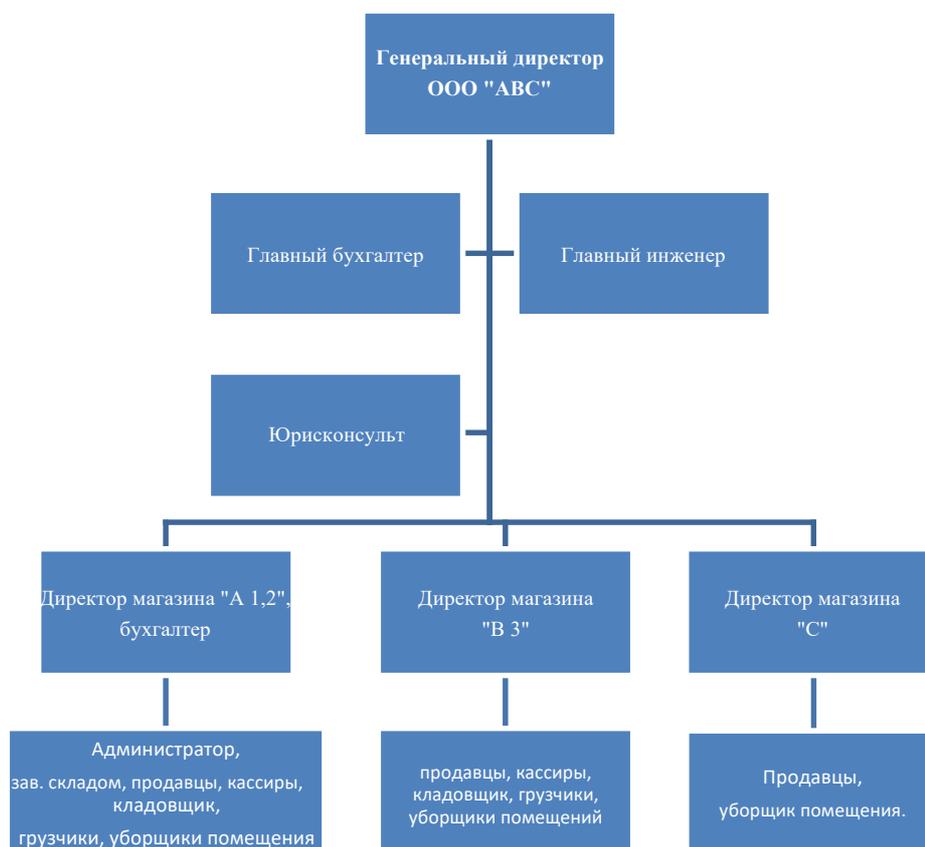


Схема 2 - Организационная структура ООО «АВС».

**Результатом практического занятия** считается умение магистрантов правильно проводить кадровые процедуры и заполнять указанные формы.

**Практическая работа №6**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Административные методы управления. Обеспечение безопасности труда. СОУТ. Социально-психологические методы управления. Экономические методы управления. Стили управления производственным коллективом. Профилактика производственных конфликтов. Подготовка персонала к внедрению инноваций. Управление корпоративной (организационной) культурой

***Административные методы управления производственным коллективом***

**Устные вопросы по теме практического занятия:**

- \* Власть, дисциплина, ответственность как основа организационно-административных методов управления.
- \* Виды организационно-административных методов управления: указания, правила, рекомендации, контроль.
- \* Нормативно-правовые акты как основания организационно-административных методов управления.

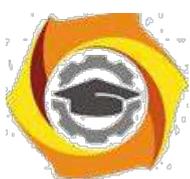
**Содержание практического занятия**

**1 этап.** Обсуждение с преподавателем устных вопросов по теме практического занятия.

**2 этап.** С помощью правовой поисковой системой «КонсультантПлюс» заполнение третьей колонки таблицы 1.

**Таблица 1. – Перечень необходимых документов, касающихся трудовых отношений**

| Документ                                                                         | Ссылка на законодательство                                             | В каких случаях оформляется |
|----------------------------------------------------------------------------------|------------------------------------------------------------------------|-----------------------------|
| Правила внутреннего трудового распорядка                                         | Статья 189 ТК РФ                                                       |                             |
| Штатное расписание                                                               | Постановление Госкомстата России от 05.01.2004 N 1                     |                             |
| Книга учета движения трудовых книжек и вкладышей в них                           | Постановление Минтруда России от 10.10.2003 N 69                       |                             |
| Должностные инструкции по каждой должности в соответствии со штатным расписанием | Единый тарифно-квалификационный справочник, профессиональные стандарты |                             |
| Положение о персональных данных работников                                       | Статья 86 ТК РФ                                                        |                             |
| Положение об оплате труда работников                                             | Раздел VI ТК РФ                                                        |                             |
| Положение о премировании и материальном стимулировании работников                | Раздел VI ТК РФ                                                        |                             |
| Положение об охране труда                                                        | Раздел X ТК РФ                                                         |                             |
| Инструкции по охране труда по профессиям                                         | Раздел X ТК РФ                                                         |                             |
| Журнал инструктажа (ознакомление с инструкциями)                                 | Раздел X ТК РФ                                                         |                             |
| Журнал прохождения работниками обязательного                                     | Статья 69 ТК РФ                                                        |                             |



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

|                                                                              |                                                                                                                                               |  |
|------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|--|
| медицинского освидетельствования                                             |                                                                                                                                               |  |
| График отпусков                                                              | Статья 123 ТК РФ, Постановление Госкомстата России от 05.01.2004 N 1                                                                          |  |
| Положение о коммерческой тайне                                               | Федеральный закон от 29.07.2004 N 98-ФЗ «О коммерческой тайне»                                                                                |  |
| Коллективный договор                                                         | Глава 7 ТК РФ                                                                                                                                 |  |
| Трудовой договор                                                             | Статьи 16, 56, 67 ТК РФ                                                                                                                       |  |
| Приказ о приеме на работу                                                    | Статья 68 ТК РФ, Постановление Госкомстата России от 05.01.2004 N 1                                                                           |  |
| Личная карточка работника                                                    | Постановление Госкомстата России от 05.01.2004 N 1                                                                                            |  |
| Трудовая книжка                                                              | Статья 66 ТК РФ, Постановление Правительства РФ от 16.04.2003 N 225 «О трудовых книжках»,<br>Постановление Минтруда России от 10.10.2003 N 69 |  |
| Договоры о полной материальной ответственности                               | Приложения N 2 и N 4 к Постановлению Минтруда России от 31.12.2002 N 85                                                                       |  |
| Приказ о предоставлении отпуска работнику                                    | Постановление Госкомстата России от 05.01.2004 N 1                                                                                            |  |
| Заявление работника о предоставлении отпуска без сохранения заработной платы | Статья 128 ТК РФ                                                                                                                              |  |
| Табель учета рабочего времени и расчета оплаты труда                         | Постановление Госкомстата России от 05.01.2004 N 1                                                                                            |  |
| График сменности                                                             | Статья 103 ТК РФ                                                                                                                              |  |

**3 этап.** Обсуждение результатов работы магистрантов с преподавателем.

**Результатом практического занятия** считается умение магистрантов найти необходимую информацию в области нормативно-правового регулирования трудовых отношений для конструктивного взаимодействия с подчиненными, руководством подразделения, персоналом других структурных подразделений и руководством организации Компании.

**Обеспечение безопасности труда. Специальная оценка условий труда.**

**Устные вопросы по теме практического занятия:**

- в Каково значение создания благоприятной производственной среды (экономическое и социальное)?
- в Что понимается под вредными и опасными производственными факторами?
- в Какие имеются основные группы вредных и опасных производственных факторов?
- в Какие частные показатели могут характеризовать состояние условий труда и его безопасности?
- в Каковы основные методы интегральной оценки условий труда?
- в Какие нормативные документы используются в качестве критериев оценки?
- в В чем суть СОУТ и для каких целей используются ее результаты?



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- и Что такое условия труда?
- и Какие существуют виды производственных опасностей?
- и От чего зависят быстрота и качество действий человека?
- и Каковы основные показатели тяжести физического труда?
- и Что такое рабочее место?
- и По каким параметрам оценивается фактическое состояние рабочих мест?
- и С какой целью производится СОУТ?

## **Содержание практического занятия**

**1 этап.** Обсуждение с преподавателем устных вопросов по теме практического занятия.

**2 этап.** С помощью правовой поисковой системы «КонсультантПлюс» решите следующие практические задачи:

- \* Выделите на примере одного из производственных цехов металлургического предприятия вредные и опасные факторы производства.
- \* Классифицируйте рабочие места на примере одного из производственных цехов металлургического предприятия.
- \* Заполните карту СОУТ.
- \* Составьте список нормативных актов, используемых при спецоценке рабочих мест, используя одну из справочно-правовых систем «КонсультантПлюс».
- \* Составьте список инструкций по технике безопасности для одного из производственных цехов металлургического предприятия.

**Результатом практического занятия** считается умение магистрантов найти необходимую информацию в области нормативно-правового регулирования трудовых отношений для конструктивного взаимодействия с подчиненными, руководством подразделения, персоналом других структурных подразделений и руководством организации Компании, оценить риск и определить меры по обеспечению безопасных условий труда.

## ***Социально-психологические методы управления производственным коллективом***

### **Устные вопросы по теме практического занятия:**

- В Социально-психологические методы управления как основа повышения социальной активности людей.
- В Современные системы морального стимулирования производственного персонала.

## **Содержание практического занятия**

**1 этап.** Обсуждение с преподавателем устных вопросов по теме практического занятия.

**2 этап.** Групповая работа по формированию комплекса методов морального стимулирования производственного коллектива для решения конкретных организационных проблем.

Варианты организационных проблем:

- увеличение количества организационных конфликтов между руководителями производственных подразделений (мастерами участков, начальниками отделов);
  - нежелание производственных работников участвовать в крупных социальных акциях корпорации (спортивных соревнованиях, гражданских акциях, представительских акциях т.п.);
- 2 возрастание конфликтных ситуаций и организационных споров между руководителями крупных подразделений корпорации (начальниками цехов, начальниками управлений, служб и т.п.);
- 2 повышение текучести высококвалифицированных производственных кадров в корпорации;
- 2 отсутствие положительной динамики в повышении квалификации производственных рабочих;
- 2 повышение количества больничных листов рабочих и служащего персонала цеха в течение календарного года.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**3 этап.** Защита группой разработанного пакета предложений по решению конкретной организационной проблемы с помощью социально-психологических методов управления производственным коллективом.

**Результатом практического занятия** считается умение разрабатывать комплекс методов морального стимулирования производственного коллектива в четком соответствии с конкретной организационной проблемой.

*Экономические методы управления производственным коллективом*

**Устные вопросы по теме практического занятия:**

- Экономические методы управления как основа достижения экономических целей производственной организации.
- Современные методы экономического (материального) стимулирования производственного персонала: заработная плата, доплата за условия труда, надбавки, внутрифирменные льготы, участие в прибылях и в акционерном капитале, премиальные выплаты (бонусы) и др.

**Содержание практического занятия**

**1 этап.** Обсуждение с преподавателем устных вопросов по теме практического занятия.

**и этап.** Выполнение практической задачи. Групповая работа.

**Задание:** На основании данных таблицы 2 сформировать комплекс методов экономического (материального) стимулирования производственного коллектива для решения конкретных организационных проблем, а именно:

- 4) Корпорация купила небольшое работающее производственное предприятие в небольшом городе другого региона. Стоит задача переоборудования предприятия, освоение новых производственных технологий, выпуск высоко технологичной продукции, оптимизации ресурсов, выход на новые рынки сбыта.
- 5) Стабильно работающее предприятие. Выигран тендер на производство дополнительной продукции на 3 года. Имеющихся кадровых ресурсов недостаточно для реализации проекта.

Произошло снижение конкуренции предприятия на имеющемся рынке продукции (цена, качество, условия поставок и др.). Предприятие пока не работает в убыток, но прогнозы на ближайший год – снижение доходности на 10%.

Методические указания  
к лабораторным и практическим занятиям

**Микроконтроллеры**  
**ИЗУЧЕНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ**  
**СПЕЦКОНТРОЛЛЕРА**

**Цель**

В Углубить и закрепить теоретические знания по принципам построения специализированных контроллеров;

В Приобрести практические навыки самостоятельного изучения и анализа схем контроллеров;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

В Совершенствовать навыки анализа, обобщения и систематизации полученных результатов, навыки составления и оформления отчетных материалов, навыки точного и лаконичного представления докладов на вопросы технического характера.

**Учебные вопросы**

- 1.1. Назначение, основные технические характеристики микроконтроллера AVR;
- 1.2. Структурная схема и назначение периферийных устройств;
- 1.3. Программная модель микроконтроллера.

**Литература для подготовки к занятию**

- и Голубцов М.С. Микроконтроллеры AVR: от простого к сложному. М.: СОЛОН-Пресс, 2003. 288 с.
- и Редькин П.П. Микроконтроллеры Atmel архитектуры AVR32 семейства AT32UC3. Руководство пользователя. М.: Техносфера, 2010. - 784 с.
- и Белов А.В. Самоучитель разработчика устройств на микроконтроллерах AVR. - СПб.: Наука и техника, 2008. -544 с.
- и Ревич Ю.В. Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. СПб.: БХВ-Петербург, 2011. -352с.
- и Евстифеев А.В. Микроконтроллеры AVR семейства Tiny. Руководство пользователя. - М.: Издательский дом Додэка-XXI, 2007 . - 432с.

**Содержание отчета**

- Название работы.
- Название каждого учебного вопроса и краткий конспект в объеме практических заданий по вопросу.

**Вопросы для подготовки к занятию**

- Алгоритм работы центрального процессора.
- Принципы работы оперативной памяти и Flash-памяти.
- Характеристика гарвардской и фон-Неймановской архитектур.
- В чем отличие CISC и RISC структур системы команд.

5

**Актуальность занятия**

Спецконтроллер являются основными устройствами, осуществляющим управление и обработку данных в системах управления. Поэтому очень важно знать его характеристики, технические возможности взаимодействия с периферией и программную модель.

***Задание 1.1. Назначение, основные технические характеристики микроконтроллера AVR***



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Микропроцессоры (МП) и микроконтроллеры (МК) для встраиваемых приложений [*embedded applications*] ориентированы на построение законченных систем управления и их узлов. В частности 8-разрядные МК в общем объеме процессорных микросхем составляет чуть меньше 30%. AVR, пожалуй, одно из самых интересных направлений, развиваемых корпорацией Atmel. Они представляют собой мощный инструмент для создания современных высокопроизводительных и экономичных многоцелевых контроллеров. На настоящий момент соотношение "цена - производительность - энергопотребление" для AVR является одним из лучших на мировом рынке 8-разрядных микроконтроллеров. Объемы продаж AVR в мире удваиваются ежегодно. В геометрической прогрессии растет число сторонних фирм, разрабатывающих и выпускающих разнообразные программные и аппаратные средства поддержки разработок для них. Можно считать, что AVR постепенно становится еще одним индустриальным стандартом среди 8-разрядных микроконтроллеров общего назначения.

МК отличают от других процессорных устройств такие существенные черты:

5. развитые встроенные аппаратные узлы ввода/вывода, разгружающие процессор от рутинных действий и поддерживающие интерфейсы различных типов - дискретный, параллельный, последовательный, аналоговый;

6. интегрированная на кристалл память программ и данных, в том числе, энергонезависимая;

7. наличие большого числа разновидностей кристаллов, различающихся числом выводов и типом корпуса, объемами памяти, составом блоков вво-да/вывода и пр.;

8. встроенные узлы загрузки кодов программ и данных и узлы внутри-схемной отладки, работающие по последовательному интерфейсу;

9. гибкая система тактирования и управления энергопотреблением, что актуально для мобильных устройств.

Основные различия между МК данной группы сводятся к производительности процессора, определяемой сочетанием следующих факторов: состав системы команд и режимов адресации, разрядность процессора (8/16/32), частота тактирования (десятки - сотни мегагерц). Рост разрядности сочетается с ростом частоты и снижением напряжения питания (с 5 В до 3.3 В, а ядра - до 1.2 В).



На отечественном рынке 8-разрядные МК представлены прежде всего «клонами» и «потомками» модели *MCS-8051*, семействами *PIC12/14/16/18* (*MicroChip*), семейством *AVR* (фирма *Atmel*) и пр. Эти МК ориентированы на решение задач управления с преобладанием логических и несложных арифметических законов управления. Их максимальные тактовые частоты доходят сегодня до 16...25 МГц при питании 5 В. Уровень входных и выходных сигналов при этом имеет достаточный запас помехоустойчивости – примерно одна треть от напряжения питания, то есть не менее 1.7 В.

9. частности, в данном курсе наибольшее внимание будет уделено семейству *AVR* фирмы *Atmel*, как одному из динамично развивающихся и доступному на отечественном рынке.

*Семейством (или серией) МК* называется ряд микросхем однородной разработки, имеющих идентичную архитектуру, но различающихся количеством ячеек памяти, выводов корпуса, составом и количественными характеристиками встроенных периферийных модулей. Различают в зависимости от аппаратного обеспечения семейства "AVR-tiny", "AVR-classic" и "AVR-mega". Выпуск МК для встраиваемых применений в виде семейства позволяет производителям снизить затраты на создание богатой номенклатуры микросхем (при разработке новых кристаллов используются готовые проверенные в производстве модули кристалла), а пользователям МК сокращает затраты времени на освоение моделей, наиболее подходящих для решения новых задач.

#### *История и особенности семейства AVR-8*

10. середине 90-х годов (20 века) фирма *Atmel* выпустила клон МК *MCS-8051* по технологии К-МОП – семейство *AT89xx*, в котором реализовала два существенных на тот момент новшества:

- **интегрированная на кристалле память программ** типа *FlashROM* и энергонезависимая память данных типа *EEPROM*,

- **встроенный механизм загрузки/выгрузки** содержимого памяти программ (*FlashROM*) и данных (*EEPROM*) с использованием последовательного интерфейса (типа *SPI*).

Сочетание этих новаций обеспечило существенное удешевление и упрощение - отпала необходимость в использовании дорого керамического корпуса с кварцевым окном для ультрафиолетового стирания, и отпала необходимость использования не дешевого (не менее 300\$) параллельного программатора, в который при каждой операции записи приходилось помещать корпус с микросхемой памяти программ. Новая технология стала называться внутрисистемным программированием [*In-System Programming - ISP*], существенно упростила и удешевила процедуру редактирования содержимого памяти программ (ПП) и данных (ПД) - теперь стало возможным выполнять перепрограммирование МК прямо "в системе", то есть на целевой плате.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Следующим шагом фирмы *Atmel* в конце 90-х годов стала разработка нового семейства 8-разрядных МК *AVR-8 (AT90xxxx)*, в котором кроме новшеств, внедренных при разработке *AT89xx*, добавились следующие:



- **интегрированные на кристалл узлы ввода аналоговых сигналов** - аналоговый компаратор и аналогово-цифровой преобразователь;
- новая система команд [RISC]** сразу разрабатывалась с расчетом на использование языка C/C++,
- процессор поддерживает **4-шаговый конвейер**, что обеспечивает выполнение большинства команд (кроме команд перехода) за один машинный цикл,
- отказ от аккумулятора и использование достаточно **большого файла регистров общего назначения** (32\*8 бит) снижает необходимость частых обращений к ПД (ОЗУ) с более долгими или длинными режимами адресации;
- этом же файле разместились 16-разрядные указатели на ПД и ПП;
- отказ от использования внешней ПП при наличии функции внутри-системного программирования позволил использовать корпуса с малым числом ног (менее 40), что еще более снижает стоимость микросхемы. Дополнительным преимуществом отказа от внешней ПП стала повышенная степень защиты содержимого ПП от несанкционированного копирования.
- Последовательность разработки подсемейств AVR-8: AT90, ATmega, [FPSLIC], ATtiny, ATxmega.

В отчете представить:

- Основные технические характеристики МКК ATtiny 2313;
- Отличие МКК от микропроцессора.

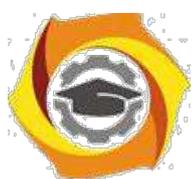
### **Задание 1.2. Структурная схема и назначение периферийных устройств**

*Архитектурой* называют образ, видимый пользователю, это концептуальное представление для программиста или схемотехника, работающего с данным МК. Основными элементами архитектуры являются:

- система команд и способы адресации;
- организация памяти и регистров;
- формат разрядной сетки
- организация встроенных периферийных устройств, механизм обращения к ним и системы прерываний;
- типовые схемы включения с описанием назначения и расположения выводов и особенностей внутренней схемотехники выводов.

Знание архитектуры является минимально достаточным для понимания - практической работы с микроконтроллером. Сравнение однотипных черт и параметров архитектуры разных МК позволяет грамотно выбирать МК для решения конкретной задачи.

*Структура* МК отражает состав и связи его внутренних блоков в графическом виде. Знание структуры МК не является обязательным для работы с



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

МК, но облегчает понимание и сравнение МК между собой за счет дополнительного визуального восприятия.



### *Архитектура AVR*

**Система команд** 8-разрядных МК серии AVR содержит от 90 до 133 инструкций, число которых зависит от модели. Как и для большинства других МК, этот набор можно условно разделить на три основные группы:

- С арифметико-логические, включая сдвиги (собственно вычисления),
- С передачи данных между ячейками памяти МК,
- Светлений (переходов) в программе (формируют структуру программы).

По формату обрабатываемых данных различают команды, оперирующие с байтами (8-битное слово) и с битами, отдельные команды работают с двухбайтными числами. Нередко группу команд работы с битами выделяют в самостоятельную четвертую группу.

**Формат команды** описывает структуру машинного слова команды. Часть бит выделяют под *код операции*, это обязательная часть любой команды. Остальные биты указывают *операнды*, то есть данное, адрес ячейки данных или адрес перехода. Различают команды безоперандные, одно- и двухоперандные. В двухоперандных командах обработки данных один из операндов называется источником [*source*], его содержимое не изменяется, другой – приемником [*destination*], его содержимое содержит результат операции.

**Способы адресации** определяют способы кодирования операндов, для кодирования используется двоичный код. Различают следующие виды адресации:

- *прямая* – адрес в виде константы, число бит  $n$  определяет число адресуемых ячеек  $N = 2n$ ,

- в *непосредственная* (или абсолютная) – данное в виде константы,

- в несколько разновидностей *косвенной* – указывается адрес регистра-указателя ( $X, Y, Z, PC, SP$ ), содержимое которого содержит адрес ячейки данных или перехода:

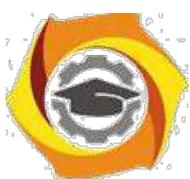
- в *простая косвенная*,

- в с *преддекрементом* – значение регистра указателя увеличивается до использования адреса,

- в с *постинкрементом* – значение регистра указателя уменьшается после использования адреса,

- в *индексная* – значение адреса вычисляется как сумма содержимого регистра указателя и константы,

- в *относительная* – разновидность индексной с использованием счетчика команд как регистра указателя перехода.



### Структура AVR

Микроконтроллер AVR содержит: быстрый RISC-процессор (Reduced Instruction Set Computer), два типа энергонезависимой памяти (Flash-память программ и память данных EEPROM), оперативную память RAM, порты ввода/вывода и различные периферийные интерфейсные схемы (рис.1.1).

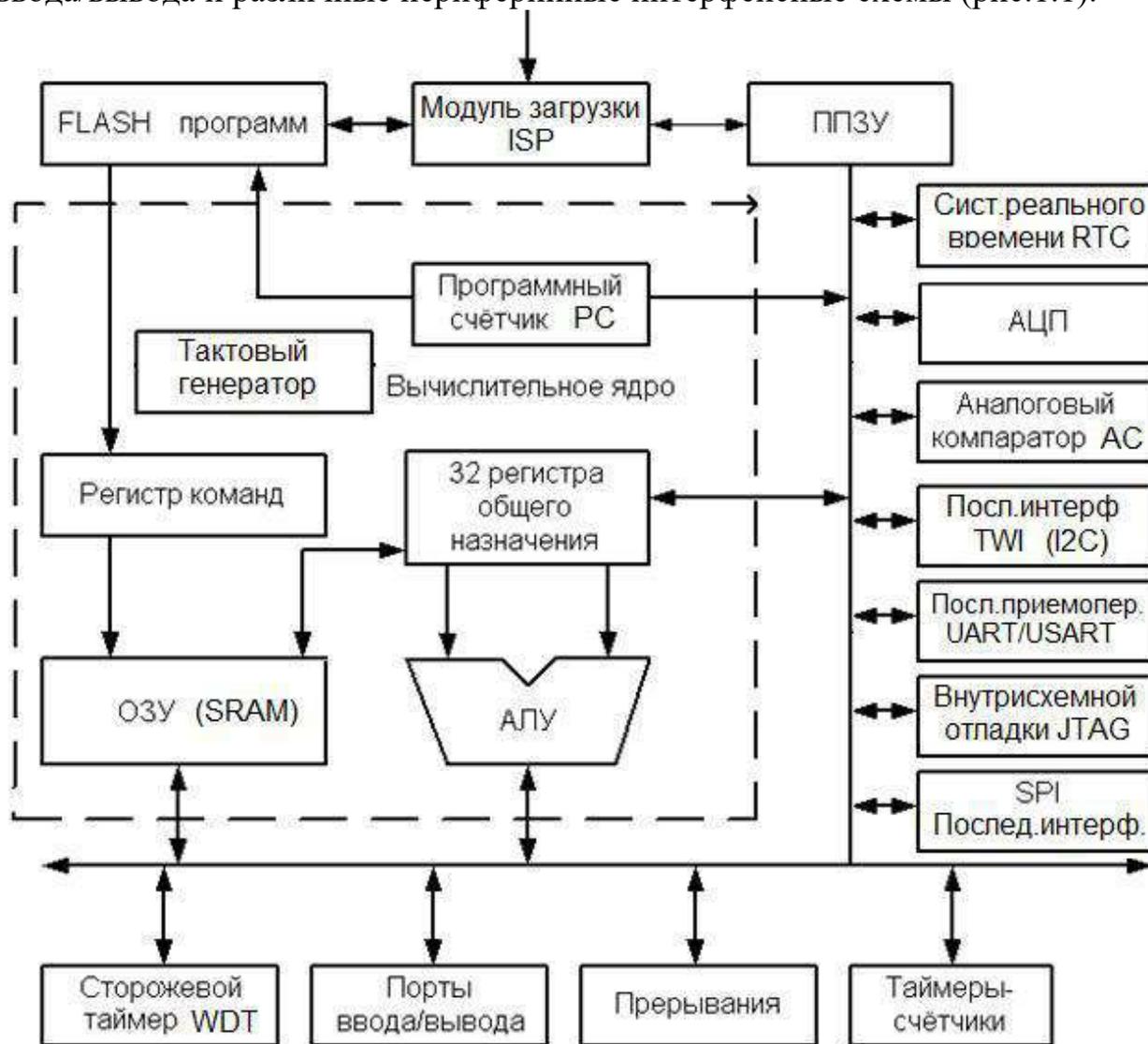


Рис.1.1. Структурная схема МК AVR

#### Вычислительное ядро

Сердцем микроконтроллеров AVR является 8-битное вычислительное ядро или центральное процессорное устройство (ЦПУ), построенное на принципах RISC-архитектуры. Основой этого блока служит арифметико-логическое устройство (АЛУ). По системному тактовому сигналу из памяти программ в соответствии с содержимым счетчика команд PC (Program Counter) выбирается очередная команда и выполняется в АЛУ. Во время вы-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

бора команды из памяти программ происходит выполнение предыдущей выбранной команды, что и позволяет достичь быстродействия 1 MIPS на 1 МГц.



### **Память программ (Flash-память)**

Память программ предназначена для хранения последовательности команд, управляющих функционированием микроконтроллера, и имеет 16-ти битную организацию. Все AVR имеют Flash-память программ, которая может быть различного размера – от 1 до 256 КБайт. Ее главное достоинство в том, что она построена на принципе электрической перепрограммируемости, т. е. допускает многократное стирание и запись информации. Программа заносится во Flash-память AVR как с помощью обычного программатора, так и с помощью SPI-интерфейса (последовательный периферийный интерфейс - Serial Peripheral Interface), в том числе непосредственно на собранной плате. Возможностью внутрисхемного программирования (функция ISP) через коммуникационный интерфейс SPI обладают все микроконтроллеры AVR, кроме Tiny11 и Tiny28.

Все микроконтроллеры семейства Mega имеют возможность *самопрограммирования*, т. е. самостоятельного изменения содержимого своей памяти программ. Эта особенность позволяет создавать на их основе очень гибкие системы, алгоритм работы которых будет меняться самим микроконтроллером в зависимости от каких-либо внутренних условий или внешних событий. Гарантированное число циклов перезаписи Flash-памяти у микроконтроллеров AVR второго поколения составляет не менее 10 тыс. циклов при типовом значении 100 тыс. циклов. (В официальной технической документации Atmel Corp. указывается значение 10 тыс. циклов.)

### **Память данных**

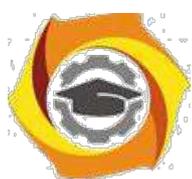
Память данных разделена на три части:

7. регистровая память,
8. оперативная память (ОЗУ – оперативное запоминающее устройство или RAM)
9. энергонезависимая память (ЭСППЗУ или EEPROM).

### **Регистровая память (РОН и РВВ)**

*Регистров общего назначения (РОН - General Purpose Registers – GPR)* всего 32, они имеют байтовый формат и объединены в файл. РОН находятся в начале адресного пространства оперативной памяти, но физически не являются ее частью. Поэтому к ним можно обращаться двумя способами (как к регистрам и как к памяти). Такое решение является особенностью AVR и повышает эффективность работы и производительность микроконтроллера.

*Служебные регистры ввода/вывода (РВВ)* объединяют регистры управления микроконтроллером, регистры состояния и т. п., а также регистры управления периферийными устройствами, входящими в состав микро-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

контроллера. По сути, управление микроконтроллером заключается в управлении этими регистрами.



РОН и РВВ расположены в адресном пространстве ОЗУ, но не являются его частью. Отличие между регистрами и оперативной памятью состоит в том, что с регистрами можно производить любые операции (арифметические, логические, битовые), а в оперативную память можно лишь записывать данные из регистров.

### **Энергонезависимая память данных (EEPROM)**

Для долговременного хранения различной информации, которая может изменяться в процессе функционирования микроконтроллерной системы, используется EEPROM-память - перепрограммируемое ПЗУ (ППЗУ). Все AVR имеют блок энергонезависимой электрически перезаписываемой памяти данных EEPROM от 64 Байт до 4 КБайт. Этот тип памяти, доступный программе микроконтроллера непосредственно в ходе ее выполнения, удобен для хранения промежуточных данных, различных констант, коэффициентов, серийных номеров, ключей и т.п. EEPROM может быть загружена извне как через SPI интерфейс, так и с помощью обычного программатора. Число циклов стирание/запись – не менее 100 тыс.

Два программируемых бита секретности позволяют защитить память программ и энергонезависимую память данных EEPROM от несанкционированного считывания

### **Оперативная память (ОЗУ или RAM)**

Внутренняя оперативная статическая память Static RAM (SRAM) имеет байтовый формат и используется для оперативного хранения данных. Размер оперативной памяти может варьироваться у различных чипов от 64 Байт до 4 КБайт. Число циклов чтения и записи в RAM не ограничено, но при отключении питающего напряжения вся информация теряется. Для некоторых микроконтроллеров возможна организация подключения внешнего статического ОЗУ объемом до 64К.

микроконтроллерах AVR реализована Гарвардская архитектура, в соответствии с которой разделены не только адресные пространства памяти программ и памяти данных, но и шины доступа к ним.

Периферия микроконтроллеров AVR включает: порты ввода и вывода, поддержку внешних прерываний, таймеры-счетчики, сторожевой таймер, аналоговые компараторы, 10-разрядный 8-канальный АЦП, интерфейсы UART, JTAG и SPI, устройство сброса по понижению питания, широтно-импульсные модуляторы.

### **Порты ввода/вывода (I/O)**

Порты ввода/вывода AVR имеют число независимых линий —вход/выход|| от 3 до 53. Каждая линия порта может быть запрограммирована на вход или на



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

выход. Мощные выходные драйверы обеспечивают токовую нагрузочную способность 20 мА на линию порта (втекающий ток) при мак-



симальном значении 40 мА, что позволяет, например, непосредственно подключать к микроконтроллеру светодиоды и биполярные транзисторы. Общая токовая нагрузка на все линии одного порта не должна превышать 80 мА (все значения приведены для напряжения питания 5 В).

Архитектурная особенность построения портов ввода/вывода у AVR заключается в том, что для каждого физического вывода (пина) существует 3 бита контроля/управления, а не 2, как у распространенных 8-разрядных микроконтроллеров (Intel, Microchip, Motorola и т.д.). Это позволяет избежать необходимости иметь копию содержимого порта в памяти для безопасности. В повышает скорость работы микроконтроллера при работе с внешними устройствами, особенно в условиях внешних электрических помех.

### **Прерывания (INTERRUPTS)**

Система прерываний – одна из важнейших частей микроконтроллера. Все микроконтроллеры AVR имеют многоуровневую систему прерываний. Прерывание прекращает нормальный ход программы для выполнения приоритетной задачи, определяемой внутренним или внешним событием.

Для каждого такого события разрабатывается отдельная программа, которую называют подпрограммой обработки запроса на прерывание (для краткости – подпрограммой прерывания), и размещается в памяти программ. При возникновении события, вызывающего прерывание, микроконтроллер сохраняет содержимое счетчика команд, прерывает выполнение центральным процессором текущей программы и переходит к выполнению подпрограммы обработки прерывания.

После выполнения подпрограммы прерывания осуществляется восстановление предварительно сохраненного счетчика команд РС и процессор возвращается к выполнению прерванной программы.

Для каждого события может быть установлен приоритет. Понятие приоритет означает, что выполняемая подпрограмма прерывания может быть прервана другим событием только при условии, что оно имеет более высокий приоритет, чем текущее. В противном случае центральный процессор перейдет к обработке нового события только после окончания обработки предыдущего.

### **Таймеры/счетчики (TIMER/COUNTERS)**

Микроконтроллеры AVR имеют в своем составе от 1 до 4 таймеров/счетчиков с разрядностью 8 или 16 бит, которые могут работать и как таймеры от внутреннего источника тактовой частоты, и как счетчики внешних событий.

Их можно использовать для точного формирования временных интервалов, подсчета импульсов на выводах микроконтроллера, формирования последовательности импульсов, тактирования приемопередатчика последовательного канала связи. В режиме широтно-импульсной модуляции ШИМ



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

(PWM) таймер/счетчик может представлять собой широтно-импульсный модулятор и используется для генерирования сигнала с программируемыми



частотой и скважностью. Таймеры/счетчики способны вырабатывать запросы прерываний, переключая процессор на их обслуживание по событиям и освобождая его от необходимости периодического опроса состояния таймеров. Поскольку основное применение микроконтроллеры находят в системах реального времени, таймеры/счетчики являются одним из наиболее важных элементов.

### **Сторожевой таймер (WDT)**

Сторожевой таймер (WatchDog Timer) предназначен для предотвращения катастрофических последствий от случайных сбоев программы. Он имеет свой собственный RC-генератор, работающий на частоте 1 МГц. Как и для основного внутреннего RC-генератора, значение 1 МГц является приближенным и зависит прежде всего от величины напряжения питания микроконтроллера и от температуры. Таймер снабжен своим собственным предделителем входной частоты с программируемым коэффициентом деления, что позволяет подстраивать временной интервал переполнения таймера и сброса микроконтроллера.

Идея использования сторожевого таймера предельно проста и состоит в регулярном его сбрасывании под управлением программы или внешнего воздействия до того, как закончится его выдержка времени и не произойдет сброс процессора. Если программа работает нормально, то команда сброса сторожевого таймера должна регулярно выполняться, предохраняя процессор от сброса. Если же микропроцессор случайно вышел за пределы программы (например, от сильной помехи по цепи питания) либо зациклился на каком-либо участке программы, команда сброса сторожевого таймера скорее всего не будет выполнена в течение достаточного времени и произойдет полный сброс процессора, инициализирующий все регистры и приводящий систему в рабочее состояние.

### **Аналоговый компаратор (АС)**

Аналоговый компаратор (Analog Comparator) сравнивает напряжения на двух выводах (пинах) микроконтроллера. Результат сравнения будет логическое значение, которое может быть прочитано из программы.

Выход аналогового компаратора можно включить на прерывание от аналогового компаратора. Пользователь может установить срабатывание прерывания по нарастающему или спадающему фронту или по переключению. Присутствует у всех современных AVR, кроме Mega8515.

**Аналого-цифровой преобразователь (A/D CONVERTER)** Аналого-цифровой преобразователь (АЦП) служит для получения числового значения напряжения, поданного на его вход. Этот результат сохраняется в регистре данных АЦП. Какой из выводов (пинов) микроконтроллера



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

будет являться входом АЦП, определяется числом, занесенным в соответствующий регистр.



### **Универсальный последовательный приемопередатчик (UART или USART)**

Универсальный асинхронный или универсальный синхронно/асинхронный приемопередатчик (Universal Synchronous/Asynchronous Receiver and Transmitter – UART или USART) – удобный и простой последовательный интерфейс для организации информационного канала обмена микроконтроллера с внешним миром. Способен работать в дуплексном режиме (одновременная передача и прием данных). Он поддерживает протокол стандарта RS-232, что обеспечивает возможность организации связи с персональным компьютером. (Для стыковки МК и компьютера обязательно понадобится схема сопряжения уровней сигналов. Для этого существуют специальные микросхемы, например MAX232.)

### **Последовательный периферийный интерфейс SPI**

Последовательный периферийный трехпроводный интерфейс SPI (Serial Peripheral Interface) предназначен для организации синхронного обмена данными между двумя устройствами. С его помощью может осуществляться обмен данными между микроконтроллером и различными устройствами, такими, как цифровые потенциометры, ЦАП/АЦП, FLASH-ПЗУ и др. С помощью этого интерфейса удобно производить обмен данными между несколькими микроконтроллерами AVR. Кроме того, через интерфейс SPI может осуществляться программирование микроконтроллера.

### **Двухпроводной последовательный интерфейс TWI**

Двухпроводной последовательный интерфейс TWI (Two-wire Serial Interface) является полным аналогом базовой версии интерфейса I2C (двухпроводная двунаправленная шина) фирмы Philips. Этот интерфейс позволяет объединить вместе до 128 различных устройств с помощью двунаправленной шины, состоящей из линии тактового сигнала (SCL) и линии данных (SDA).

### **Интерфейс JTAG**

Joint Test Action Group — специализированный аппаратный интерфейс, разработанный для тестирования собранных печатных плат. Разработан группой ведущих специалистов по проблемам тестирования электронных компонентов. Из-за широкой функциональности JTAG стал использоваться для отладки и программирования. На данный момент JTAG-интерфейс применяется при периферийном сканировании (тестировании печатных плат с установленными на них процессорами на наличие в цепях коротких замыканий, непропаек, западаний на 0 или 1). Управление JTAG-интерфейсом описывается в т. н. BSDL-файле, который предоставляется разработчиком каждой конкретной микросхемы. В большинстве случаев эти файлы находятся на сайте производителя микросхемы. Интерфейс JTAG был зарегистрирован в качестве промышленного стандарта IEEE Std 1149.1-1990. Четырехпровод-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

ный интерфейс JTAG используется для тестирования печатных плат, внутрисхемной отладки, программирования микроконтроллеров. Многие микрокон-



троллеры семейства Mega имеют совместимый с IEEE Std 1149.1 интерфейс JTAG или debugWIRE для встроенной отладки. Кроме того, все микроконтроллеры Mega с флэш-памятью емкостью 16 кбайт и более могут программироваться через интерфейс JTAG.

### **Модуль загрузки ISP**

Модуль загрузки или модуль внутрисхемного программирования (ISP-In-System Programmability) является основным способом программирования микроконтроллеров и программируемых логических интегральных схем и других устройств с памятью Flash или EEPROM. Модуль использует какой-либо стандартный интерфейс, встроенный в программируемую микросхему для последовательного занесения в нее кодов программ или требуемых данных.

### **Внутренний тактовый генератор**

Тактовый генератор вырабатывает импульсы для синхронизации работы всех узлов микроконтроллера. Внутренний тактовый генератор AVR может запускаться от нескольких источников опорной частоты (внешний генератор, внешний кварцевый резонатор, внутренняя или внешняя RC-цепочка). Минимальная допустимая частота ничем не ограничена (вплоть до пошагового режима). Максимальная рабочая частота определяется конкретным типом микроконтроллера и указывается Atmel в его характеристиках, хотя практически любой AVR-микроконтроллер с заявленной рабочей частотой, например, в 10 МГц при комнатной температуре легко может быть —разогнан|| до 12 МГц и выше. Микроконтроллер ATtiny15L. Он содержит блок PLL для аппаратного умножения основной тактовой частоты в 16 раз. При номинальном значении последней 1,6 МГц получаемая вспомогательная периферийная частота равна 25,6 МГц. Эта частота может служить источником для одного из таймеров/счетчиков микроконтроллера, значительно повышая временное разрешение его работы.

### **Система реального времени (RTC)**

RTC (Real Time Clock) реализована во всех микроконтроллерах Mega и в некоторых кристаллах —classic|. Таймер/счетчик RTC имеет отдельный делитель, который может быть программным способом подключен или к источнику основной тактовой частоты, или к дополнительному асинхронному источнику опорной частоты (кварцевый резонатор или внешний синхросигнал). Для этой цели зарезервированы два вывода микросхемы. Внутренний осциллятор оптимизирован для работы с внешним —часовым|| кварцевым резонатором 32,768 кГц.

с последней новации семейства AVR-8 – 8/16-разрядном подсемействе ATxmega, при наличии того же ядра и общей архитектуре, кроме усовершен-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

ствованных таймеров /счетчиков и устройств аналогового интерфейса, встречаются принципиально новые периферийные блоки:



– Система управления событиями для разгрузки процессора от пере-сылки сигналов между блоками МК.

– Контроллер прямого доступа к памяти - ориентирован на пересылку пакетов данных между блоками МК без участия процессора.

AVR функционируют при напряжениях питания от 1,8 до 6,0 Вольт. Ток потребления в активном режиме зависит от величины напряжения пита-ния и частоты, на которой работает микроконтроллер, и составляет менее 1 мА для 500 кГц, 5 ... 6 мА для 5 МГц и 8 ... 9 мА для частоты 12 МГц. AVR могут быть переведены программным путем в один из трех режимов пони-женного энергопотребления.

**Режим холостого хода (IDLE).** Прекращает работу только процессор и фиксируется содержимое памяти данных, а внутренний генератор синхро-сигналов, таймеры, система прерываний и сторожевой таймер продолжают функционировать. Ток потребления не превышает 2,5 мА на частоте 12 МГц.

**Стоповый режим (POWER DOWN).** Сохраняется содержимое реги-стрового файла, но останавливается внутренний генератор синхросигналов, и, следовательно, останавливаются все функции, пока не поступит сигнал внешнего прерывания или аппаратного сброса. При включенном сторожевом таймере ток потребления в этом режиме составляет около 80 мкА, а при вы-ключенном – менее 1 мкА. (Все приведенные значения справедливы для на-пряжения питания 5 В).

**Экономичный режим (POWER SAVE).** Продолжает работать только генератор таймера, что обеспечивает сохранность временной базы. Все ос-тальные функции отключены.

**Сброс при снижении напряжения питания (BOD).** Схема BOD (Brown-Out Detection) отслеживает напряжение источника питания. Если схема включена, то при снижении питания ниже некоторого значения она переводит микроконтроллер в состояние сброса. Когда напряжение питания вновь увеличится до порогового значения, запускается таймер задержки сброса. После формирования задержки внутренний сигнал сброса снимается и происходит запуск микроконтроллера.

В отчете представить:

34. Структурную схему МКК ATTiny 2313;

35. Назначение основных элементов структурной схемы.

### *Задание 1.3. Программная модель микроконтроллера*

Программная модель микропроцессора представляет собой совокуп-ность программно доступных ресурсов. В программную модель микрокон-троллеров семейства AVR входят РОН, регистры ввода-вывода, память про-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

грамм, оперативная память данных и энергонезависимая память данных, которые показаны на рис.1.2.

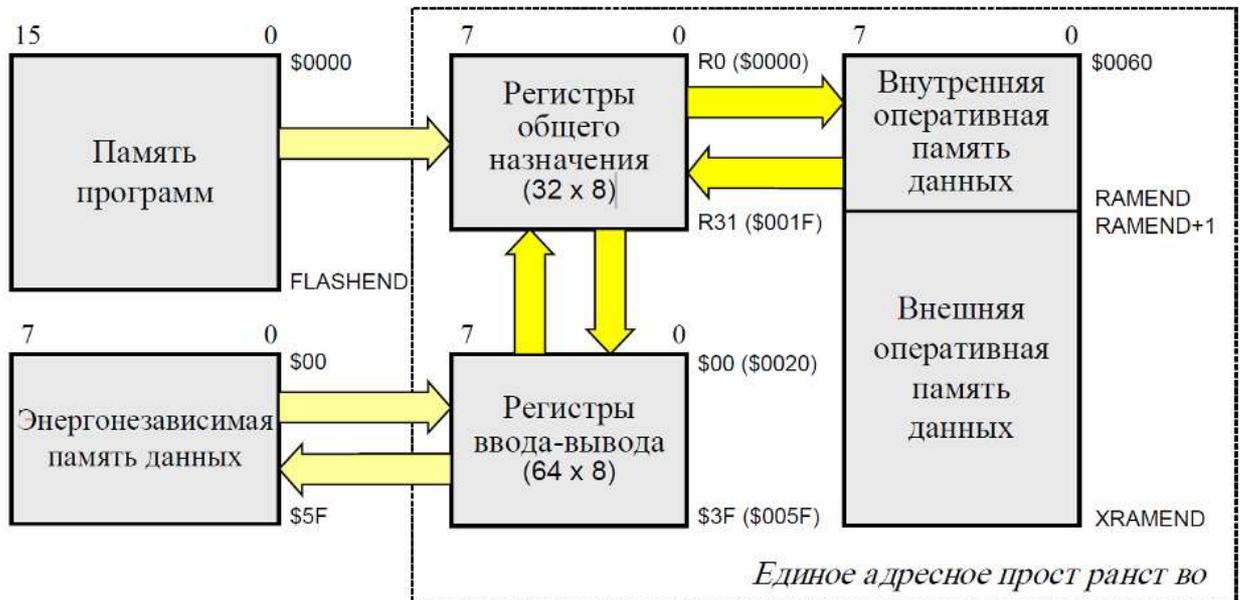


Рис.1.2. Программная модель AVR контроллера

РОН (R0...R31) могут использоваться в программе для хранения данных, адресов, констант и другой информации. Шесть старших регистров объединены попарно и составляют три 16-разрядных регистра X[R27:R26], Y[R29:R28] и Z[R31:R30] (рис.1.3).

РОН, регистры ввода-вывода и оперативная память данных образуют единое адресное пространство. Адресное пространство – это множество доступных ячеек памяти, различимых по адресам; адресом называется число, однозначно идентифицирующее ячейку памяти (регистр). Адреса ячеек памяти традиционно записываются в шестнадцатеричной системе счисления, на что указывает знак \$ в

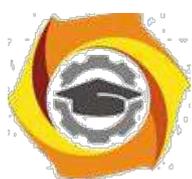


Рис.1.3. Регистры микроконтроллера AVR

обозначении адреса.

Существует две конфигурации единого адресного пространства памяти AVR-микроконтроллеров. В конфигурации "А" младшие 32 адреса (\$0000...\$001F) соответствуют РОН, следующие 64 адреса

(\$0020...\$005F) занимают регистры ввода-вывода, внутренняя оперативная память данных начинается с адреса \$0060. В конфигурации "В" начиная с адреса \$0060 размещаются 160 дополнительных регистров ввода-вывода; внутренняя оперативная память данных начинается с адреса \$0100. Конфигурация А используется в младших моделях микроконтроллеров и в некоторых старших моделях в режиме совместимости с моделями, снятыми с производства; конфигурация В – в старших моделях.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

В память программ, кроме собственно программы, могут быть записаны постоянные данные, которые не изменяются в процессе работы микро-



процессорной системы (константы, таблицы линеаризации датчиков и т. п.). Выполнение программы при включении питания или после сброса микроконтроллера начинается с команды, находящейся по адресу \$0000 (т. е. в первой ячейке) памяти программ.

Энергонезависимая память данных предназначена для хранения информации, которая может изменяться непосредственно в процессе работы микропроцессорной системы (калибровочные коэффициенты, конфигурационные параметры и т. п.). Энергонезависимая память данных имеет отдельное адресное пространство и может быть считана и записана программным путём.

В отчете представить:

- 4) Программную модель МКК ATTiny 2313;
- 5) Уточненные параметры элементов программной модели: размеры типов памяти, разрядность регистров, счетчика команд, указателя стека и т.д..

#### ***Вопросы для самоконтроля***

- в Чем отличается "структура" от "архитектуры"?
- в Какие внешние устройства могут быть в составе микроконтроллера?
- в Какие типы памяти есть в микроконтроллере и каково их назначение?
- в Каково назначение интерфейса SPI?
- в Для чего создается модуль загрузки ISP?
- в Как используются регистры общего назначения?
- в Каково назначение аналогового компаратора?
- в Каково назначение последовательного приемопередатчика?
- в Каково назначение последовательного интерфейса TWI?
- в Что входит в состав программной модели микроконтроллера?



## 7) УПРАВЛЕНИЕ ПОРТАМИ МИКРОКОНТРОЛЛЕРА

### Цель:

3. Углубить и закрепить теоретические знания по принципам построения и работы специализированных контроллеров;
4. Приобрести практические навыки самостоятельного управления портами контроллера;
5. Совершенствовать навыки анализа, обобщения и систематизации полученных результатов, навыки составления и оформления отчетных материалов, навыки точного и лаконичного представления докладов на вопросы технического характера.

### Учебные вопросы:

- 2.1. Программная модель портов микроконтроллера;
- 2.2. Способы адресации операндов;
- 2.3. Разработка программы управления портами.

### Литература для подготовки к занятию

7. Белов А.В. Создаем устройства на микроконтроллерах. - СПб.: Наука и техника, 2007. - 304 с.
8. Евстифеев А.В. Микроконтроллеры AVR семейства Tiny. Руководство пользователя. - М.: Издательский дом Додэка-XXI, 2007. - 432с.
9. Ревич Ю.В. Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. СПб.: БХВ-Петербург, 2011. - 352с.
10. Егоров А.А. и др. Программирование микроконтроллеров семейства AVR. Учебно-методическое пособие. - М.: Издательство МАИ, 2004. - 72с.

### Содержание отчета

19. Название работы.
20. Название каждого учебного вопроса и краткий конспект в объеме практических заданий по вопросу.

### Вопросы для подготовки к занятию

20. Алгоритм работы центрального процессора;
21. Принципы работы оперативной памяти и Flash-памяти;
22. Характеристика гарвардской и фон-Неймановской архитектур;
23. В чем отличие CISC и RISC структур системы команд;
24. Чем для процессора является порт?
25. В чем отличие обращений процессора к порту и памяти?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Актуальность занятия**



Спецконтроллер являются основными устройствами, осуществляющим управление и обработку данных в системах управления. Работа его в режиме включения и выключения нагрузки является базовой, такие алгоритмы используются практически во всех системах.

### ***Задание 2.1. Программная модель портов микроконтроллера;***

Изучить программную модель параллельных портов МКК ATtiny2313 с целью практического их программирования.

В отчете представить:

- Программную модель портов;
- Адреса портов;
- Назначение регистров портов.

### **Назначение и возможности портов AVR**

Порты ввода/вывода (I/O) AVR имеют число независимых линий —вход/выход от 3 до 53. Каждая линия порта может быть запрограммирована на вход или на выход. Мощные выходные драйверы обеспечивают токовую нагрузочную способность 20 мА на линию порта (втекающий ток) при максимальном значении 40 мА, что позволяет, например, непосредственно подключать к микроконтроллеру светодиоды и биполярные транзисторы. Общая токовая нагрузка на все линии одного порта не должна превышать 80 мА (все значения приведены для напряжения питания 5 В).

Архитектурная особенность построения портов ввода/вывода у AVR заключается в том, что для каждого физического вывода (пина) существует 3 бита контроля/управления, а не 2, как у распространенных 8-разрядных микроконтроллеров (Intel, Microchip, Motorola и т.д.). Это позволяет избежать необходимости иметь копию содержимого порта в памяти для безопасности 23. повышает скорость работы микроконтроллера при работе с внешними устройствами, особенно в условиях внешних электрических помех.

Порты AVR имеют следующие особенности:

Прием и выдачу данных (для Tiny 6-20 выводов, для Mega - 23-86 выводов);

Программного изменения направления передачи;

Наличие триггеров Шмитта на входах, что обеспечивает восстановление прямоугольной формы сигналов после искажения 5) линиях связи (рис.2.1);

Возможность программного подключения подтягивающего резистора, что упрощает схему



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

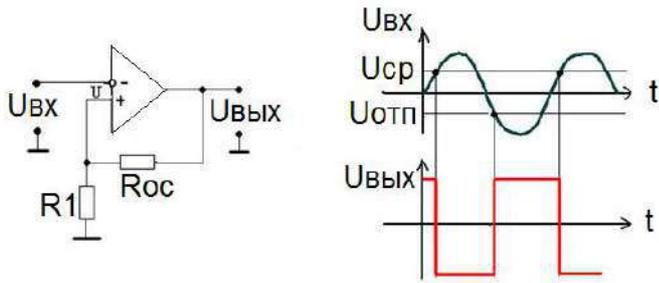
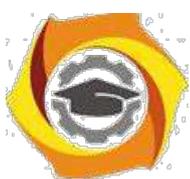


Рис.2.1. Диаграммы работы  
триггера Шмитта



4.увеличивает гибкость управления;

Программное управление отдельными битами портов;

Совмещение возможности ввода-вывода и функций управления для выводов ИМС, что также организуется программно.

### Программная модель портов AVR

AVR-микроконтроллерах в зависимости от типа имеется от двух до шести универсальных двунаправленных портов ввода-вывода (порты A...F), содержащих семь или восемь сигнальных линий, которые соединены с соответствующими выводами БИС микроконтроллера. Например, в микроконтроллере ATtiny2313 имеется три порта A, B, и D. Сигнальные линии порта A обозначаются PA0...PA7, порта B – PB0...PB7 и т. д. Для МКК ATtiny2313 форматы параллельных портов показаны на рис.2.2.

С каждым портом связаны три регистра ввода-вывода: - регистр направления передачи данных DDRx. Он задает направление передачи данных по каждой сигнальной линии порта (0 – разряд используется для ввода данных, 1 – для вывода данных);

- регистр данных порта PORTx. Он позволяет задавать состояние сигнальных линий порта при выводе информации;

- регистр выводов порта PINx. Он позволяет считывать состояние сигнальных линий порта при вводе информации.

Для МКК ATtiny2313 имена и адреса таких портов следующие:

|            |            |            |
|------------|------------|------------|
| PORTA=\$1D | PORTB=\$18 | PORTD=\$12 |
| DDRA=\$1A  | DDRB=\$17  | DDRD=\$11  |
| PINA=\$19  | PINB=\$16  | PIND=\$10  |

11. помощью указанных регистров реализуется требуемый протокол обмена данными. Здесь под словом протокол понимается совокупность правил передачи или приема кодированной информации. Основными процедурами при реализации некоторого протокола являются операции инициализации порта, вывода данных в порт, ввода данных из порта, а также формирования временных задержек.

Для обращения к отдельным разрядам портов зарезервированы имена этих разрядов (x - это одно из имен портов: A, B, D):

- для регистра PINx: PINx0 ... PINx7;
- для регистра DDRx: DDx0 ... DDx7;

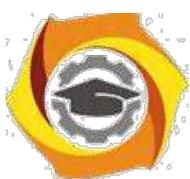


Рис.2.2. Форматы регистров



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- для регистра PORTx: PORTx0 ... PORTx7 или Px0 ... Px7.



*Инициализация порта* состоит в назначении направления обмена данными по сигнальным линиям порта. Например, если сигнальные линии **PA0**, **PA2**, порта **A** служат для вывода данных, а сигнальные линии **PA1** – для ввода данных, инициализация порта может быть произведена следующим образом:

```
ldi R16, 0b00000101
out DDRA, R16
```

*Вывод данных в порт* производится путём записи значений в регистр **PORTx**, например:

```
ldi R16, 0b10011101
out PORTA, R16 ; вывод содержимого регистра R16 в порт A.
```

Для изменения состояния только одного разряда порта ввода-вывода удобно использовать команды установки **SBI** (*Set Bit in I/O Register*) и сброса **CBI** (*Clear Bit in I/O Register*) бита в регистре ввода-вывода.

*Ввод данных из порта* производится путём чтения содержимого регистра **PINx** выводов порта, например:

```
in R16, PINA ; считывание состояния порта A в регистр R16.
```

Регистр данных **PORTx** в зависимости от режима работы порта выполняет различные функции:

4. если порт запрограммирован на вывод, то в нем хранятся выдаваемые данные;
5. если порт запрограммирован на ввод, то
  7. - подключает подтягивающий резистор;
  9. - отключает подтягивающий резистор

Варианты конфигурирования портов AVR можно свести в таблицу 2.1.

Таблица 2.1

Состояние элементов портов в различных режимах

| Направление DDRB | Данные PORTB | Подключен. R PUD | Rподтяг   | Вывод       |
|------------------|--------------|------------------|-----------|-------------|
| 1=вых            | D            | *                | Откл      | Вых=D       |
| 0 = вход         | 1            | 0=вкл            | Подключен | Ист. тока   |
|                  |              | 1=откл           | Откл      | Z-состояние |
|                  | 0            | *                | Откл      | Z-состояние |



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

По умолчанию разряд **PUD** (*Pull Up Disable*) равен **0** (R – подтягивающий резистор подключен). **PUD** – 7-й разряд регистра MCUCR=35 (*MCU Control Register*) - Общий регистр управления микроконтроллера.



### **Исходное состояние портов (устройств) по умолчанию**

После формирования команды RESTART исходное положение аппаратуры следующее:

10. устройства выключены;
11. порты запрограммированы на ввод ( $DDRx=0$ );
12. подтягивающие резисторы подключены;
13. значения разрядов обнулены.

### ***Задание 2.2. Способы адресации операндов***

Изучить команды пересылки МКК, уточнить особенности выполнения каждой команды в среде AVR Studio и сделать выводы об используемых способах адресации операндов в МКК. При необходимости воспользоваться директивами ассемблера, упомянутыми в тексте и изученными ранее. Список команд ассемблера представлен в приложении.

В отчете представить:

11. Примеры записи групп команд пересылки;
12. Диапазон доступных адресов в команде;
13. Указание способа адресации операндов в команде.

2. зависимости от количества используемых операндов возможны три типа команд AVR-микроконтроллера: безадресные, одноадресные и двухадресные [4]. В первом типе команд присутствует только код операции (КОП), определяющий выполняемую командой функцию. В командах второго и третьего типов помимо кода операции содержится адресная часть, устанавливающая способ доступа соответственно к одному или двум участвующим в команде операндам (аргументам команды). Способ формирования адреса операнда, указание на который содержится в команде, называется адресацией (addressing). С помощью того или иного способа адресации вычисляется физический адрес, который поступает на шину адреса процессора для выбора ячейки памяти или регистра, используемых в команде.

3. соответствии с типом адресуемой памяти способы адресации в AVR-микроконтроллерах можно разделить на способы адресации РОИ и регистров ввода-вывода, способы адресации оперативной памяти данных (ОЗУ) и способы адресации памяти программ. Возможность использования различных способов адресации позволяет сократить размер и время выполнения программ.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Для адресации РОН и регистров ввода-вывода предусмотрен всего один режим – прямая регистровая адресация.**



При *прямой регистровой адресации РОН* операндом является содержимое регистра общего назначения, указанного в команде. Команды с прямой регистровой адресацией могут адресовать один (**Rd**) или два (**Rr** и **Rd**) РОН. Во втором случае результат выполнения команды сохраняется в регистре **Rd**. Прямая регистровая адресация РОН применяется во всех арифметических и логических командах, а также в некоторых командах работы с битами, т. к. эти команды выполняются в АЛУ только над содержимым РОН. Команды, вторым операндом которых является константа, могут использовать в качестве первого операнда только регистры из старшей половины РОН (**R16...R31**).

При *прямой регистровой адресации регистра ввода-вывода* операнд содержится в регистре ввода-вывода, указанном в команде. Адрес регистра ввода-вывода хранится в шести разрядах слова команды. Прямая регистровая адресация регистров ввода-вывода используется в командах чтения **IN** и записи **OUT** регистра ввода-вывода, а также в ряде других команд работы с регистрами ввода-вывода.

**Для адресации оперативной памяти данных** используются пять способов адресации: непосредственная, косвенная, косвенная со смещением, косвенная с преддекрементом и косвенная с постинкрементом.

При *непосредственной адресации оперативной памяти данных* операндом является содержимое ячейки ОЗУ, адрес которой указан в команде. Адрес операнда содержится в 16 младших разрядах 32-разрядной команды. Непосредственная адресация используется в команде **LDS** (*Load Direct from Data Space*) загрузки из ОЗУ и в команде **STS** (*Store Direct to Data Space*) загрузки в ОЗУ. Зарезервировать байты в ОЗУ позволяет директива **.byte**. Для того чтобы на выделенную область памяти можно было сослаться, директиве **.byte** должна предшествовать метка. Директива **.byte** имеет один параметр – количество выделяемых байт и может использоваться только в сегменте данных, определяемом с помощью директивы **.dseg** (*data segment*). Задать требуемое размещение выделяемой области памяти позволяет директива **.org** (*origin* – смещение). Начало программного сегмента указывается с помощью директивы **.cseg** (*code segment*). Директивы **.dseg** и **.cseg** не имеют параметров. Выделенные в оперативной памяти данных байты не инициализируются.

При *косвенной адресации оперативной памяти данных* операндом является содержимое ячейки ОЗУ, адрес которой находится в регистре **X**, **Y** или **Z**. Косвенная адресация используется в команде **LD** (*Load Indirect*) косвенной загрузки из ОЗУ и команде **ST** (*Store Indirect*) косвенной загрузки в ОЗУ.

#### 4. При *косвенной адресации оперативной памяти данных со смещением*



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*ем* адрес операнда в оперативной памяти данных вычисляется путём прибав-



ления к содержимому регистра **Y** или **Z** смещения, указанного в команде. Смещение содержится в шести разрядах слова команды. Косвенная адресация со смещением используется в команде **LDD** (*Load Indirect with Displacement*) косвенной загрузки из ОЗУ со смещением и в команде **STD** (*Store Indirect with Displacement*) косвенной загрузки в ОЗУ со смещением.

и При *косвенной адресации оперативной памяти данных с предекрементом* (лат. *decrementum* – уменьшение, убыль) перед выполнением команды содержимое указанного в команде регистра **X**, **Y** или **Z** декрементируется (уменьшается на единицу); декрементированное содержимое регистра **X**, **Y** или **Z** является адресом операнда в оперативной памяти данных. Косвенная адресация с предекрементом используется в команде **LD** косвенной загрузки из ОЗУ и команде **ST** косвенной загрузки в ОЗУ.

и При *косвенной адресации оперативной памяти данных с постинкрементом* (лат. *incrementum* – увеличение, рост) адресом операнда в оперативной памяти данных является содержимое регистра **X**, **Y** или **Z**, указанного в команде; после выполнения команды содержимое регистра **X**, **Y** или **Z** инкрементируется, т. е. увеличивается на единицу. Косвенная адресация с постинкрементом используется в команде **LD** косвенной загрузки из ОЗУ и команде **ST** косвенной загрузки в ОЗУ.

**Для адресации памяти программ** используется непосредственная адресация, косвенная адресация, относительная адресация, адресация константы и адресация константы с постинкрементом.

При *непосредственной адресации памяти программ* выполнение программы продолжается с адреса, указанного в команде. Непосредственная адресация памяти программ используется в командах **JMP** и **CALL**.

При *косвенной адресации памяти программ* выполнение программы продолжается с адреса, содержащегося в регистре **Z**, т. е. в программный счётчик загружается содержимое регистра **Z**. Косвенная адресация памяти программ используется в командах **IJMP** и **ICALL**.

При *относительной адресации памяти программ* выполнение программы продолжается с адреса (**PC** + **k** + 1), где **PC** – содержимое программного счётчика; **k** – указанный в команде относительный адрес, который может принимать значения от –2048 до 2047. Относительная адресация памяти программ используется в командах **RJMP** и **RCALL**.

При *адресации константы* адрес байта константы содержится в регистре **Z**. Адресация константы в памяти программ используется в командах **LPM** (*Load Program Memory*), которая загружает адресованный регистром **Z** байт в указанный в команде регистр. Если регистр-приёмник не указан (команда используется без операндов), байт загружается в регистр **R0**. Команда **ELPM** (*Extended Load Program Memory*) служит для загрузки константы из



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

памяти программ объёмом более 64К слов. При этом для расширения регистра-указателя **Z** используется регистр **RAMPZ**, связанный с регистром **Z**.



Задать данные в память программ позволяет директива *.db* (*define bytes*). Для того, чтобы на заданные ячейки памяти можно было ссылаться, директиве должна предшествовать метка. Параметры директивы – последовательность выражений, разделённых запятыми; каждое выражение должно быть числом в диапазоне –128...255 или в результате вычисления давать результат в этом же диапазоне, в противном случае число усекается до байта. Директива *.db* размещается в программном сегменте и может использоваться совместно с директивой *.org*. Задавать положение данных в памяти программ следует таким образом, чтобы была исключена возможность непреднамеренного перехода к выполнению их как команд микроконтроллера.

При адресации константы в памяти программ с постинкрементом адрес байта константы содержится в регистре **Z** и формируется так же, как и при адресации константы. Адресованный регистром **Z** байт загружается в указанный регистр; после выполнения команды содержимое регистра **Z** инкрементируется. Адресация константы в памяти программ с постинкрементом используется в командах *LPM* и *ELPM*.

в диалоговом окне *AVR Simulator Options* в разделе *Device Selection* установить тактовую частоту моделирования работы микроконтроллера, равную 8,0 МГц (поле *Frequency*).

Выполнить трансляцию и отладку созданных программ. По данным, выводимым после трансляции на закладке *Build* окна *Output*, проанализировать использование памяти программ (*Program memory usage*) под код программы (*Code*) и константы (*Constants*), оценить объём неиспользованной (*Unused*) и общей занятой (*Total*) памяти.

При отладке программ использовать средства наблюдения за содержимым регистров и ячеек памяти. По полю *Cycle Counter* объекта *Processor* закладки *I/O* окна *Workspace* определить число тактов выполнения программы, по полю *Stop Watch* – время выполнения программы (до выполнения команды, организующей бесконечный цикл). Зафиксировать эти сведения в отчёте. По результатам выполнения программ сделать выводы о правильности выполнения команд.

### ***Задание 2.3. Разработка программы управления портами***

Разработать схему алгоритма и программу в соответствии с последующими заданиями. Выполнить трансляцию и отладку программы. Продемонстрировать работу программы на эмуляторе и в программе Proteus. Представить данные в отчете в соответствии с каждым заданием.



### Задание 2.3.1. Управление светодиодом от кнопки

Разработать устройство управления одним светодиодом при помощи одной кнопки. При нажатии кнопки светодиод должен зажегаться, при отпуске - погаснуть.

Разработка алгоритма управления опирается на конкретную схему аппаратных средств. Поэтому решение должно начинаться с разработки схемы.

Управление светодиодом и работа с кнопкой осуществляется с помощью портов. Желательно порты назначить разные (рис.2.3). Это позволит упростить настройку портов: светодиод подключается к порту, настроенному на вывод, а кнопка к порту, настроенному на ввод.

Светодиод VD1 включен через токоограничивающий резистор R3 и зажигается при наличии нуля на выходе контакта PB.0. При появлении на выводе логической единицы потенциалы на светодиоде выравниваются, и он не светится. При необходимости зажигать светодиод высоким потенциалом, его следовало подключить к общему проводу.

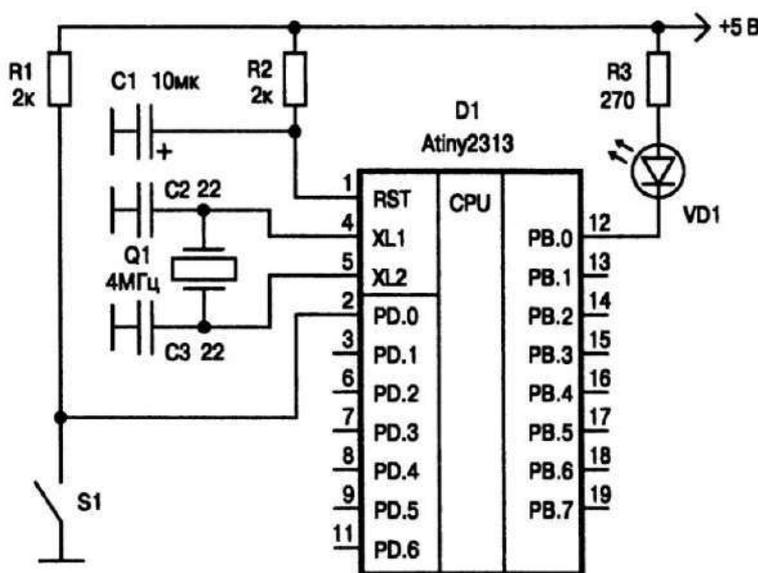


Рис.2.3. Схема подключения светодиода и кнопки

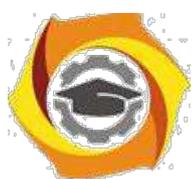
к исходному состоянию контакты кнопки S1 разомкнуты и через резистор R1 на вход PD.0 подается плюс источника питания, что соответствует логической единице. При нажатии кнопки потенциал контакта падает до нуля, что соответствует логическому нулю. Это классическое подключение кнопки.

Для контроллеров AVR схему можно упростить, подключив вместо внешнего резистора резистор встроенный (подтя-

гивающий).

Кроме того, схема начального сброса C1R2 также может быть исключена, потому что в контроллерах AVR имеется внутренняя схема сброса, которая обеспечивает стабильный сброс, а высоких помех и нестабильного питания не предполагается.

Можно обойтись и без внешнего кварца, воспользовавшись встроенным генератором. Но его можно оставить в расчете на будущие задачи. Все настройки внешних цепей можно обеспечить fuse-переключателями. Поэтому



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

схема должна получиться как показана на рис.2.4. Квар для упрощения следует также исключить.



Алгоритм управления светодиодом.

Выполнить операции начальной настройки. Далее в непрерывном цикле опрашивать кнопку и по ее состоянию управлять светодиодом.

Операции начальной настройки:

1. Инициализировать стек;
2. Настроить порт В на вывод;
3. Подать на выход PB.0 единицу (потушить светодиод);
4. Сконфигурировать порт D на ввод;
5. Подключить подтягивающие резисторы;

Операции тела цикла:

В Прочитать состояние кнопки: разряда PD.0;

В Если состояние разряда равно единице (кнопка не нажата), выключить светодиод;

В Если значение разряда PD.0 равно нулю (кнопка нажата), включить светодиод;

В Перейти на начало цикла.

Структура программы управления светодиодом на Ассемблере показана на рис.2.5.

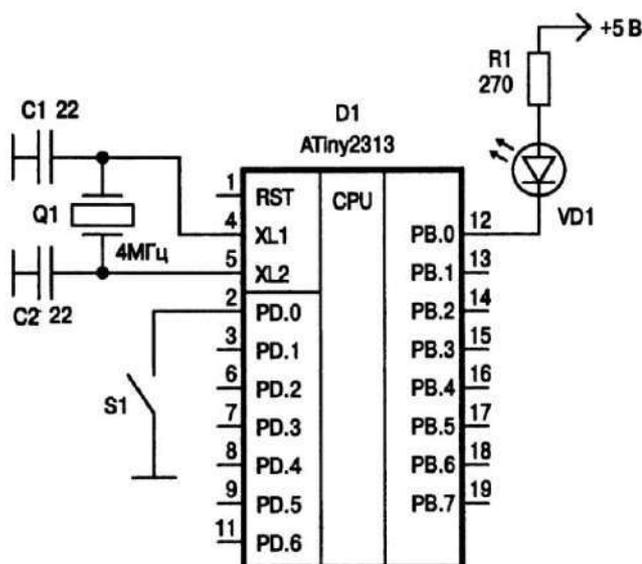


Рис.2.4. Упрощенная схема подключения кнопки и светодиода



```

Пример 1 ##
Программа управления светодиодом ##

----- Команды управления

1 .include "tn2313def.inc" ; Присоединение файла описаний
2 .list ; Включение листинга

3 .def temp = R16 ; Определение главного рабочего регистра

;----- Начало программного кода

4 .cseg ; Выбор сегмента программного кода
5 .org 0 ; Установка текущего адреса на ноль

;----- Инициализация стека

6 ldi temp, RAMEND ; Выбор адреса вершины стека
7 out SPL, temp ; Запись его в регистр стека

;----- Инициализация портов ВВ

8 ldi temp, 0 ; Записываем 0 в регистр temp
9 out DDRD, temp ; Записываем этот ноль в DDRD (порт PD на ввод)

10 ldi temp, 0xFF ; Записываем число $FF в регистр temp
11 out DDRB, temp ; Записываем temp в DDRB (порт PB на вывод)
12 out PORTB, temp ; Записываем temp в PORTB (потушить светодиод)
13 out PORTD, temp ; Записываем temp в PORTD (включаем внутр. резист.)

;----- Инициализация компаратора

14 ldi temp, 0x80 ; Выключение компаратора
15 out ACSR, temp

;----- Основной цикл

16 main: in temp, PIND ; Читаем содержимое порта PD
17 out PORTB, temp ; Пересылаем в порт PB
18 rjmp main ; К началу цикла
```

Рис.2.5. Структура программы на Ассемблере

**.include "tn2313def.inc"** - псевдооператор (директива ассемблера) присоединения текста. Текст вставляется в том месте, где установлен оператор. Параметром является файл. В данном случае это файл описаний микроконтроллера. Стандартное место его размещения **C:\Program Files\Atmel\AVR Tools\AvrAssembler\Appnotes\**. В частности из указанного файла транслятор узнает имена и параметры конкретного микроконтроллера:

```
RAMEND EQU 0x00DF
DDRD EQU $11
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

SPL

EQU

\$3D и т.д.



**.list** - включение генерации листинга;

**.def temp = R16** - макроопределение. Присваивает регистрам содержательные имена для увеличения наглядности и читабельности текста программ.

**.cseg** - выбор сегмента кодов в памяти программ Flash. До объявления другого сегмента все коды будут располагаться в выбранной памяти.

Остальные строки программы необходимо разобрать по справочнику команд и уяснить логику работы программы.

### ***Практическое задание***

Набрать программу в среде разработки, запустить и устранить выявленные ошибки.

Промоделировать схему с помощью эмулятора AVR Studio.

Набрать схему в среде Proteus без внешнего кварца и промоделировать работу программы.

### ***В отчете представить:***

К Схему подключения кнопки и светодиода (желательно в среде Proteus).

К Программу на языке ассемблера с необходимыми пояснениями.

К Выбранные типы элементов схемы в программе Proteus.

### ***Структура программы на языке C***

Вариант программы

```
/******
```

```
ПРИМЕР МИГАНИЯ СВЕТОДИОДОМ 10 РАЗ
```

```
*****/
```

```
#define F_CPU 1000000UL // указываем частоту в герцах
#include<avr/io.h> // подключение вв-выв по типу МКК
#include<util/delay.h> // подкл. функций циклов задержки
```

В ОПРЕДЕЛЕНИЕ ГЛОБАЛЬНЫХ ПЕРЕМЕННЫХ

```
int main(void) // начало основной программы
{
 CLKPR = 0x80; //Коэфделен.частоты сист. генератора
 CLKPR = 0x00; //Коэф=

 PORTA = 0x00; //Инициализация порта ВВ А(на вход)
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*DDRA = 0x00;*

*//R подтягивающие ОТКЛ (Z состояние)*



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

31

```
//Инициализация порта B
//
PORTD = 0xFF; //Инициализация порта ВВ D (на выход)
DDRD = 0x00; //Все единицы на выход

TCCR0A = 0x00; //Инициализация таймера/счетчика 0
TCCR0B = 0x00; //Источник – системный генератор
TCNT0 = 0x00; //Значение частоты – таймер 0 остановл.
//Режим Normal макс значение FFh

OCR0A = 0x00; //Выход OCR0A - отключен
OCR0B = 0x00; //Выход OCR0B–отключен

TCCR1A = 0x00; //Инициализация таймера/счетчика1
TCCR1B = 0x00; //Источник – системный генератор
TCNT1H = 0x00; //Значение частоты – таймер 1остановл.
TCNT1L = 0x00; //Режим Normal макс значение FFFFh
ICR1H = 0x00; //Noise Canceler: Off
ICR1L = 0x00; //Input Capture on Falling Edge
OCR1AH = 0x00; //Выход OCR0A –отключен
OCR1AL = 0x00; //
OCR1BH = 0x00; //ВыходOCR0B – отключен
OCR1BL = 0x00; //

GIMSK = 0x00; //Инициализация внешних прерываний
MCUCR = 0x00; //INT00 – выкл; INTO1 – выкл

TIMSK = 0x00; //Инициализация прерываний от таймеров

USICR = 0x00; //Инициализация последов интерф USI
//режим - выкл
//Clock source: Register&Counter=no clk
//USI Counter OverflowInterrupt: off

ACSR = 0x80; //Инициализация аналогового компаратора
//Компаратор выключен
//Вход от Сч 1 выключен

int i; //объявляем переменную i
DDRD = 0xff; //все выводы порта D на выход
```



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

Цикл: **for** ( $i = 1; i \leq 10; i++$ ) // цикл "для", повтор 10 раз



```
{
 PORTD |= _BV(PD1); ///"1" на PD1 (зажечь светодиод)
 _delay_ms(250); ///задержка 0.25 сек.
 PORTD &= ~_BV(PD1); ///"0" на PD1 (погасить светодиод)
 _delay_ms(250); ///задержка 0.25 сек.
}
 // goto Cikli;
} ///закрывающая скобка осн.Прогр.
```

После присоединения файла описаний размещается описание глобальных переменных.

Настройка системы по умолчанию производится автоматически после системного сброса. Однако включение команд начальной настройки обеспечивает правильную работу программы в случае ошибки программы и передачи управления на начало программы, а также при срабатывании сторожевого таймера.

Настройка производится всех портов, аналогично тому, как это делалось на ассемблере.

Далее задается непрерывный цикл. Это обязательный элемент программы любого микроконтроллера.

Функция `_BV()` выполняет поразрядный сдвиг на число разрядов, указанное в скобках. Например:

```
PORTD |= _BV(PD3); // установить "1" в разряде 3 порта D;
PORTD &= ~_BV(PD4); // установить "0" на линии 4 порта D.
Эта запись аналогична операндам:
PORTD |= 1<<3;
PORTD &= ~(1<<4).
```

### ***Практическое задание:***

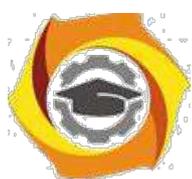
- Реализовать вариант программы на языке C, убедиться в правильности компиляции. Использовать только строки инициализации требуемых портов и стека.

- Устранить выявленные ошибки (при необходимости).

- Проверить работу программы на эмуляторе AVR Studio. Добиться ее работоспособности.

12. Проверить работу программы на языке C с использованием команды ***goto Cikli*** и без нее. Сделать вывод о работоспособности обоих вариантов. Указать возможную причину получившегося результата;

13. Проверить инициализацию стека без использования явных команд инициализации.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

14. Промоделировать работу программы в среде Proteus.  
Зафиксировать использованные элементы схемы.



**10. отчете представить:**

1. Схему алгоритма задачи;
2. Текст программы на языке C (с указанием только строк инициализации портов);  
В Схему подключения внешних дискретных элементов на эмуляторе. Типы используемых компонентов.  
В Значение указателя стека в программе на языке C.

**Задание 2.3.2. Управление переключением светодиода от кнопки**

Используя предыдущую схему, обеспечить при каждом нажатии кнопки переключение светодиода в противоположное состояние.

Алгоритм управления от кнопки:

- Прочитать состояние младшего разряда PD.0;
- Если значение разряда равно единице (кнопка не нажата), то перейти / началу цикла;
- Если значение разряда равно нулю (кнопка нажата), то изменить состояние выхода PB.0 на противоположное. Для переключения светодиода предварительно необходимо проверить его текущее состояние;
- Перейти на начало цикла.

**Практическое задание**

- 8) Составить схему алгоритма.
- 9) Разработать программу и добиться правильной компиляции.
- 10) Проверить работоспособность программы на эмуляторе AVR Studio.
- 11) Промоделировать работу программы в среде Proteus. Зафиксировать использованные элементы схемы. В качестве кнопки можно использовать компонент BUTTON ACTIVE.

Если светодиод мигает часто и устойчивое его состояние назначается хаотически, то, вероятно, это происходит потому, что за время нажатия он успевает многократно переключиться, а момент отпускания является случайным. Для устранения этого явления следует после переключения светодиода добавить цикл ожидания отпускания кнопки (когда PD.0 будет равен единице). Добавить в программу этот фрагмент и проверить работоспособность устройства.

**8) отчете представить:**

1. Схему алгоритма задачи;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

2. Текст программы на языке программирования (только основной цикл);



13. Схему подключения внешних дискретных элементов на эмуляторе, если она изменилась по отношению к предыдущему заданию.

***Задание 2.3.3. Управление переключением светодиода от кнопки с защитой отдребезга***

Используя предыдущую схему, обеспечить при каждом нажатии кнопки переключение светодиода в противоположное состояние. Алгоритм должен предусмотреть исключение дребезга контактов кнопки.

Дребезг это эффект выдачи от элемента коммутации при одном нажатии серии импульсов за счет упругих свойств механических элементов конструкции. Это приводит к тому, что на вход порта поступает целая серия переключений вместо одного. Самый простой программный способ борьбы с этим явлением - введение задержек. Обнаружив первое переключение, программа должна выдержать задержку не менее времени дребезга контактов.

В отчете представить:

14. Схему алгоритма задачи;
15. Текст программы на языке программирования (только основного цикла);
16. Уточнить схему подключения внешних дискретных элементов на эмуляторе, если она изменилась по отношению к предыдущему заданию.

***Задание 2.3.4. Управление переключением мигающего светодиода***

Используя исходную схему, нажатием кнопки необходимо включать и выключать мигание светодиода. Частота мигания светодиода 5 Гц (для обеспечения возможности наблюдения мигания частоту допускается изменить, но с указанием ее реального значения).

Алгоритм:

- Прочитать порт PD;
  - Если младший разряд равен нулю, включить алгоритм мигания;
  - Если младший разряд равен единицы, выключить алгоритм мигания
- 12) выключить светодиод;
- Перейти к началу основного цикла (п.1).

Алгоритм мигания:

8. Зажечь светодиод;
9. Выдержать паузу;
10. Потушить светодиод;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

11. Выдержать паузу;



5 Перейти к п.1.

В отчете представить:

9. Схему алгоритма задачи;
10. Текст программы на языке программирования (с указанием только основного блока);
11. Уточнить схему подключения внешних дискретных элементов на эмуляторе, если она изменилась по отношению к предыдущему заданию.

### ***Задание 2.3.5. Бегущие огни***

Обеспечить бегущий огонь из одного светодиода по гирлянде из восьми светодиодов. Направление движения гирлянды должно изменяться нажатием кнопки. Частота мигания светодиода 1 Гц.

Задача решается в том же порядке, как и предыдущие.

В отчете представить:

21. Схему алгоритма задачи;
22. Текст программы на заданном языке программирования (с указанием только основного блока);
23. Уточнить схему подключения внешних дискретных элементов на эмуляторе, если она изменилась по отношению к предыдущему заданию.

### ***Вопросы для самоконтроля***

- В Какие адреса занимают порты ввода-вывода контроллера?
- В Какие регистры управляют портами?
- В Какие режимы можно задать портам?
- В Какова типовая программная модель порта?
- В Какие способы адресация РОН и регистров ввода-вывода AVR-микроконтроллеров существуют?
- В Какие способы адресации памяти данных AVR-микроконтроллеров используются на практике?
- В Какие используются способы адресации памяти программ AVR-микроконтроллеров?
- В Особенности выполнения арифметических и логических операций в AVR-микроконтроллерах.
- В В чем отличие команд безусловного перехода ассемблера КР580 и микроконтроллера?
- 10. Для чего предназначена директива .def ?
- 9. Назначение и использование регистров X, Y и Z.
- 10. Как осуществляется инициализация стека на языке ассемблера и языке C?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

11. Какова структура программы на языке ассемблера и языке С.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

36

- Как записать на ассемблере команду или команды, эквивалентные команде присваивания на языке С?
- Как записать на ассемблере команду или команды, эквивалентные командам цикла на языке С?
- Просмотрите машинные коды, получающиеся после компиляции с языка С.
- Какое состояние имеют порты регистров РОН и РВВ по умолчанию?
- Каково назначение функции `_BV()`?
- Каково назначение функции `_delay()` ?



## 10. УПРАВЛЕНИЕ СЧЕТЧИКАМИ МИКРОКОНТРОЛЛЕРА

### Цель:

- 7) Углубить и закрепить теоретические знания по принципам построения и работы специализированных контроллеров;
- 8) Приобрести практические навыки самостоятельного управления счетчиками контроллера;
- 9) Совершенствовать навыки анализа, обобщения и систематизации полученных результатов, навыки составления и оформления отчетных материалов, навыки точного и лаконичного представления докладов на вопросы технического характера.

### Учебные вопросы:

- 3.1. Программная модель таймеров-счетчиков микроконтроллера;
- 3.2. Задание параметров счетного сигнала;
- 3.3. Разработка программы управления счетчиками в режиме Normal;
- 3.4. Разработка программы управления счетчиками в режиме СТС;
- 3.5. Разработка программы управления счетчиками в режиме PWM.

### Литература для подготовки к занятию

- 9) Белов А.В. Создаем устройства на микроконтроллерах. - СПб.: Наука и техника, 2007. - 304 с.
- 10) Евстифеев А.В. Микроконтроллеры AVR семейства Tiny. Руководство пользователя. - М.: Издательский дом Додэка-XXI, 2007. - 432с.
- 11) Ревич Ю.В. Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. СПб.: БХВ-Петербург, 2011. - 352с.
- 12) Егоров А.А. и др. Программирование микроконтроллеров семейства AVR. Учебно-методическое пособие. - М.: Издательство МАИ, 2004. - 72с.

### Содержание отчета

- В Название работы.
- В Название каждого учебного вопроса и краткий конспект в объеме практических заданий по вопросу.

### Вопросы для подготовки к занятию

11. Как работает счетчик? Какие режимы реализует счетчик?
12. Какие существуют методы построения делителей?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

13. Какие сигналы выдает схема сравнения?
14. На что влияет разрядность счетчика?
15. Как посчитать максимальный код счетчика?



и Что такое прерывание? Как оно обрабатывается процессором?

### **Актуальность занятия**

Спецконтроллер являются основными устройствами, осуществляющим управление и обработку данных в системах управления. Контроль временных интервалов, реакция на окончание времени таймеров, формирование широтно-импульсных последовательностей для управления объектами являются основными управляющими функциями микроконтроллера. Поэтому понимание этих процессов и умение их программировать является обязательной для специалиста.

### ***Задание 3.1. Программная модель таймеров-счетчиков микроконтроллера***

Изучить программную модель счетчиков МКК ATtiny2313 с целью практического их программирования.

#### ***В отчете представить:***

6. Программную модель счетчиков с указанием всех портов и их адресов;
8. Структуру регистра управления и таблицы задания режимов работы счетчиков;
9. Назначение внешних выводов счетчиков.

### **Назначение и возможности счетчиков AVR**

Счетчики контроллера решают различные задачи при выполнении управленческих функций. Все они связаны в основном с необходимостью формирования точных временных интервалов, причем без дополнительной нагрузки на процессор, а также с подсчетом числа событий:

9. измерение и задание временных интервалов;
  10. подсчет количества внешних событий (импульсов);
  11. формирование одиночных и периодических импульсов с постоянными и переменными параметрами (частота, длительность импульса, количество импульсов, фазы);
  12. преобразование временного интервала в код для АЦП;
  13. фиксация состояния счетчика по внешнему сигналу (захват) и т.д.
10. микроконтроллерах бывает до десяти счетчиков. В ATtiny2313 имеется два счетчика: однобайтный T0 и двухбайтный T1.



### Программная модель счетчиков AVR

5. AVR-микроконтроллере ATtiny2313 имеется два счетчика: 8-разрядный T0 и 16-разрядный T1. Структура их почти совпадает, за исключением разрядности и средств реализации захвата счетчика для T1. На рис.3.1. представлена внутренняя структура счетчика T1.

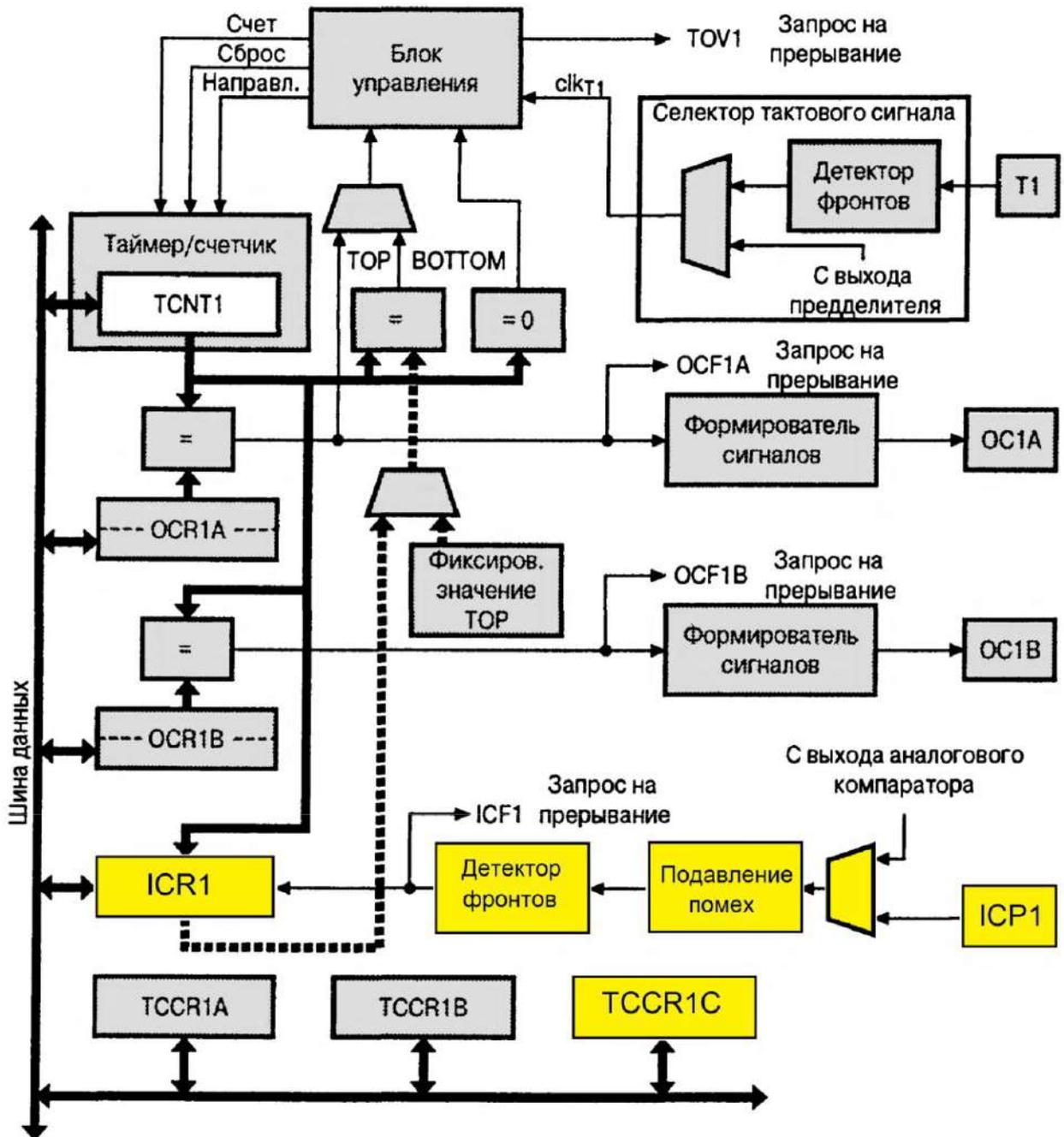
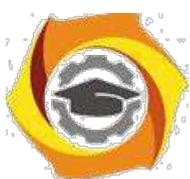


Рис.3.1. Структурная схема счетчика T1 ATtiny2313



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Назначение внешних выводов таймера T1 (в скобках указаны аналогичные выводы для таймера T0):**



T1 (T0) - вход внешнего сигнала таймера. На этот вход подаются внешние подсчитываемые сигналы;

OC1A (OC0A), OC1B (OC0B) - выход схемы сравнения регистра A и B таймера соответственно. Сигнал формируется при совпадении кода счетчика и кода в регистре A или B;

ICP1 - вход захвата таймера T1. По этому сигналу код счетчика запоминается в регистре ICR1 и может быть прочитан при выполнении обработчика прерывания ICP. В счетчике T0 такого входа и режима не предусмотрено.

Расположение выводов таймеров микроконтроллера показано на рис.3.2. Выводы таймеров совпадают с выводами цифровых портов контроллера и подключаются к счетчикам при установке единиц, настраивающих поведение этих выводов, в регистрах управления счетчиками (см. табл.3.3)

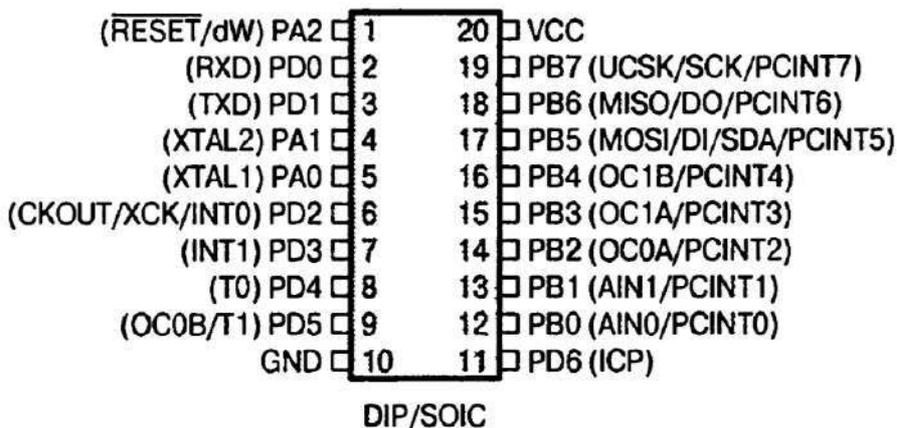


Рис.3.2. Выводы микроконтроллера ATtiny2313

Количество регистров ввода-вывода, имеющих в конкретном таймере-счетчике, зависит от сложности, разрядности и возможностей последнего, количества генерируемых прерываний.

Таймер-счетчик T1 (в скобках приведены данные для счетчика-таймера T0) имеет следующие порты, их адреса и назначение (см. рис.3.1).

### Счетчик таймера TCNT1 (TCNT0)

TCNT1 (TCNT0=\$32) - реверсивный 16-разрядный (8-разрядный) счетчик таймера T1 (T0). В исходном положении (после подачи питания) счетчик обнулен. Может быть сброшен, выполнять инкремент или декремент по каждому тактовому сигналу. Доступен для чтения и записи в любой момент времени. В зависимости от режима счета счетчик при достижении максимального или минимального значения устанавливается флаг TOV1 (TOV0) и генерируется прерывание. 16-разрядный счетчик TCNT1 представлен двумя 8-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

разрядными портами: старший байт - TCNT1H=\$2D и младший байт TCNT1L=\$2C. Такое деление обусловлено 8-разрядной шиной данных. За-



пись данных в счетчик должно осуществляться двумя последовательными посылками.

### Регистры совпадения OCR1A, OCR1B

**OCR1A** и **OCR1B** ( $OCR0B = \$3C$ ) - 16-разрядные (восьмиразрядный) регистры совпадения. В каждом такте счета происходит непрерывное сравнение их значений со счетчиком таймера TCNT1 (TCNT0). При совпадении - > следующем такте устанавливается флаг OC1A (OC0A) или OC1B (OC0B) и генерируется прерывание. Кроме того, при совпадении может меняться состояние выводов OC1A (OC0A) и OC1B (OC0B), если они сконфигурированы как выходы (этот бит в регистре DDRx должен быть равен 1). Аналогично счетчику таймера T1, 16-разрядные регистры совпадения счетчика T1 адресуются побайтно: регистр OCR1A представлен старшим байтом  $OCR1AH = \$2B$  и младшим байтом  $OCR1AL = \$2A$ , регистр OCR1B соответственно -  $OCR1BH = \$29$  и  $OCR1BL = \$28$ .

### Регистр захвата таймера ICR1

**ICR1** - *Input Capture Registers* - регистр захвата таймера T1. Представлен двумя портами: старший байт -  $ICR1H = \$25$  и младший байт -  $ICR1L = \$24$ . Регистр предназначен для сохранения состояния счетчика в режиме захвата при поступлении активного фронта (тип фронта может быть настроен) от входа ICP1, либо по сигналу аналогового компаратора. Одновременно с записью в регистр захвата устанавливается флаг ICF1 - *Input Capture Flag* - и формируется запрос на прерывание. Для захвата по сигналу ICP1 этот вывод должен быть сконфигурирован на вход (бит регистра DDRx должен иметь 0). Копирование счетчика в регистр захвата происходит с задержкой от 2,5 до 3,5 тактов от момента поступления первого фронта. Задержку вносят синхронизатор и схема подавления помех. Если же включена схема подавления помех, которая контролирует еще 4 выборки с частотой тактового генератора, то задержка увеличивается еще на 4 такта.

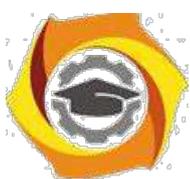
### Доступ к 16-разрядным регистрам

Для чтения и записи в 16-разрядные регистры через 8-разрядную шину данных используется регистр временного хранения в автоматическом режиме. Первая пересылка осуществляется с временным регистром, а вторая завершает обмен всех 16 разрядов.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Поэтому для записи в 16-разрядный регистр необходимо сначала записать старший байт, а затем младший. При чтении 16-разрядного регистра сначала нужно прочитать младший байт, а затем старший.



### Регистры управления TCCR1A, TCCR1B, TCCR1C

TCCR1A=\$2F (TCCR0A=\$30), TCCR1B=\$2E (TCCR0B=\$33), TCCR1C=\$22 - регистры управления счетчиком-таймером. Они задают возможные режимы работы таймеров и осуществляют все другие настройки счетчиков-таймеров: форму сигналов на выходе, частоту счета, источник тактового сигнала, особенности работы блока сравнения и т.д.

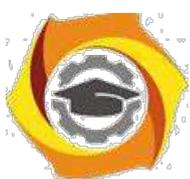
Регистры управления для таймера T0 и назначение разрядов показаны на рис.3.3. Аналогичные регистры существуют для T1 - на рис.3.4.

| TCCR0A=\$30 Регистр A управления счетчика T0<br><i>Timer/Counter Control Register</i> |        |                                             |        |   |   |                               |       |
|---------------------------------------------------------------------------------------|--------|---------------------------------------------|--------|---|---|-------------------------------|-------|
| 7                                                                                     | 6      | 5                                           | 4      | 3 | 2 | 1                             | 0     |
| COM0A1                                                                                | COM0A0 | COM0B1                                      | COM0B0 | - | - | WGM01                         | WGM00 |
| Режим работы блока сравнения A (вывод OC0A)                                           |        | Режим работы блока сравнения B (вывод OC0B) |        |   |   | Режим работы таймера/счетчика |       |

| TCCR0B=\$33 Регистр B управления счетчика T0<br><i>Timer/Counter Control Register</i> |                                                |   |   |                               |                                                 |      |      |
|---------------------------------------------------------------------------------------|------------------------------------------------|---|---|-------------------------------|-------------------------------------------------|------|------|
| 7                                                                                     | 6                                              | 5 | 4 | 3                             | 2                                               | 1    | 0    |
| FOC0A                                                                                 | FOC0B                                          | - | - | WGM02                         | CS02                                            | CS01 | CS00 |
| Принудительное изменение состояния вывода OC0A                                        | Принудительное изменение состояния вывода OC0B |   |   | Режим работы таймера/счетчика | Управление тактовым сигналом (источник сигнала) |      |      |

Рис.3.3. Регистры управления таймером T0



|                                                                                                            |        |                                             |        |   |   |                                  |       |
|------------------------------------------------------------------------------------------------------------|--------|---------------------------------------------|--------|---|---|----------------------------------|-------|
| TCCR1A=\$2F Регистр А управления счетчика Т1 (аналог TCCR0A=\$30)<br><i>Timer/Counter Control Register</i> |        |                                             |        |   |   |                                  |       |
| 7                                                                                                          | 6      | 5                                           | 4      | 3 | 2 | 1                                | 0     |
| COM1A1                                                                                                     | COM1A0 | COM1B1                                      | COM1B0 | - | - | WGM11                            | WGM10 |
| Режим работы блока сравнения А (вывод ОС1А)                                                                |        | Режим работы блока сравнения В (вывод ОС1В) |        |   |   | Режим работы таймера/счетчика Т1 |       |

|                                                                                                                     |                              |   |                               |       |                                                 |      |      |
|---------------------------------------------------------------------------------------------------------------------|------------------------------|---|-------------------------------|-------|-------------------------------------------------|------|------|
| TCCR1B=\$2E Регистр В управления счетчика Т1 (частично аналог TCCR0B=\$33)<br><i>Timer/Counter Control Register</i> |                              |   |                               |       |                                                 |      |      |
| 7                                                                                                                   | 6                            | 5 | 4                             | 3     | 2                                               | 1    | 0    |
| ICNC1                                                                                                               | ICES1                        | - | WGM13                         | WGM12 | CS12                                            | CS11 | CS10 |
| Включение схемы подавления помех                                                                                    | Выбор фронта сигнала захвата |   | Режим работы таймера/счетчика |       | Управление тактовым сигналом (источник сигнала) |      |      |

|                                                                                       |       |                                                         |   |   |   |   |   |
|---------------------------------------------------------------------------------------|-------|---------------------------------------------------------|---|---|---|---|---|
| TCCR1C=\$22 Регистр С управления счетчика Т1<br><i>Timer/Counter Control Register</i> |       |                                                         |   |   |   |   |   |
| 7                                                                                     | 6     | 5                                                       | 4 | 3 | 2 | 1 | 0 |
| FOC1A                                                                                 | FOC1B | -                                                       | - | - | - | - | - |
| Принудительное изменение вывода ОС1А (по COM1A1:COM1A0)                               |       | Принудительное изменение вывода ОС1В (по COM1B1:COM1B0) |   |   |   |   |   |

Рис.3.4. Регистры управления счетчиком Т1

Таймер Т0 может работать в шести различных режимах, которые можно установить битами WGM0-WGM2 (Wave Generator Mode). Комбинации этих битов и соответствующие им режимы представлены в таблице 3.1.

Таблица 3.1

Режимы работы таймера Т0

| Номер режима | WGM02 | WGM01 | WGM00 | Режим работы таймера/счетчика Т0 | Модуль счета (TOP) | Обновление регистров OCR0x | Момент установки флага TOV0 |
|--------------|-------|-------|-------|----------------------------------|--------------------|----------------------------|-----------------------------|
| 0            | 0     | 0     | 0     | Normal                           | \$FF               | Немедленно                 | \$FF                        |
| 1            | 0     | 0     | 1     | Phase correct PWM                | \$FF               | При TOP                    | \$00                        |
| 2            | 0     | 1     | 0     | СТС (сброс при совпадении)       | OCR0A              | Немедленно                 | \$FF                        |
| 3            | 0     | 1     | 1     | Fast PWM                         | \$FF               | При TOP                    | \$FF                        |
| 4            | 1     | 0     | 0     | Зарезервировано                  | —                  | —                          | —                           |
| 5            | 1     | 0     | 1     | Phase correct PWM                | OCR0A              | При TOP                    | \$00                        |
| 6            | 1     | 1     | 0     | Зарезервировано                  | —                  | —                          | —                           |
| 7            | 1     | 1     | 1     | Fast PWM                         | OCR0A              | При TOP                    | TOP                         |



Счетчик T1 является 16-разрядным. В отличие от T0 счетчик реализует дополнительно режим захвата. Элементы структуры, обеспечивающие этот режим, показаны желтым цветом (тонкими линиями) на рис.3.1.

Возможные режимы счетчика T1 представлены в таблице 3.2.

### Режим Normal (таймера)

Это наиболее простой режим, который имеет место в любом контроллере:

- счетчик работает как суммирующий от тактового сигнала до максимального значения;

- при переполнении счетчика T0 (код \$FF меняется на \$00) устанавливается флаг переполнения TOV0=1. Аналогично происходит в счетчике T1 (код \$FFFF меняется на код \$0000, устанавливается флаг TOV1);

- при совпадении кода счетчика с регистрами сравнения устанавливается соответствующий флаг OCF0A или OCF0B (OCF1A или OCF1B) и генерируется прерывание (если есть маска и общее разрешение прерывания);

- 7. при подключении выводов OC0A или OC0B (OC1A или OC1B) на них формируется сигнал, форма которого программируется битами COM (Com-pare Match Output Mode) в соответствии с таблицей 3.3.

***Если хоть один из этих битов установлен в 1, то вывод OC0 перестает функционировать как обычный вывод общего назначения и подключается к схеме сравнения таймера счетчика. Однако при этом он должен быть еще настроен как выход. Настройка OC0A и OC0B (OC1A и OC1B) должна быть выполнена до конфигурирования выводов на выход.***

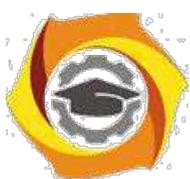


Таблица 3.2

Режимы работы счетчика T1

| Номер режима | WGM13 | WGM12 | WGM11 | WGM10 | Режим работы таймера/счетчика T1 | Модуль счета (TOP) | Обновление регистров OCR1x | Момент установки флага TOV1 |
|--------------|-------|-------|-------|-------|----------------------------------|--------------------|----------------------------|-----------------------------|
| 0            | 0     | 0     | 0     | 0     | Normal                           | \$FFFF             | Немедленно                 | \$FFFF                      |
| 1            | 0     | 0     | 0     | 1     | Phase correct PWM, 8-битный      | \$00FF             | При TOP                    | \$0000                      |
| 2            | 0     | 0     | 1     | 0     | Phase correct PWM, 9-битный      | \$01FF             | При TOP                    | \$0000                      |
| 3            | 0     | 0     | 1     | 1     | Phase correct PWM, 10-битный     | \$03FF             | При TOP                    | \$0000                      |
| 4            | 0     | 1     | 0     | 0     | CTC (сброс при совпадении)       | OCR1A              | Немедленно                 | \$FFFF                      |
| 5            | 0     | 1     | 0     | 1     | Fast PWM, 8-битный               | \$00FF             | При TOP                    | При TOP                     |
| 6            | 0     | 1     | 1     | 0     | Fast PWM, 9-битный               | \$01FF             | При TOP                    | При TOP                     |
| 7            | 0     | 1     | 1     | 1     | Fast PWM, 10-битный              | \$03FF             | При TOP                    | При TOP                     |
| 8            | 1     | 0     | 0     | 0     | Phase and Frequency Correct PWM  | ICR1               | \$0000                     | \$0000                      |
| 9            | 1     | 0     | 0     | 1     | Phase and Frequency Correct PWM  | OCR1A              | \$0000                     | \$0000                      |
| 10           | 1     | 0     | 1     | 0     | Phase correct PWM                | ICR1               | При TOP                    | \$0000                      |
| 11           | 1     | 0     | 1     | 1     | Phase correct PWM                | OCR1A              | При TOP                    | \$0000                      |
| 12           | 1     | 1     | 0     | 0     | CTC (сброс при совпадении)       | ICR1               | Немедленно                 | \$FFFF                      |
| 13           | 1     | 1     | 0     | 1     | Зарезервировано                  | —                  | —                          | —                           |
| 14           | 1     | 1     | 1     | 0     | Fast PWM                         | ICR1               | При TOP                    | При TOP                     |
| 15           | 1     | 1     | 1     | 1     | Fast PWM                         | OCR1A              | При TOP                    | При TOP                     |

Таблица 3.3

Управление выводами OC0A и OC0B в режиме Normal

| COM0A1 (COM0B1) | COM0A0 (COM0B0) | Описание                                         |
|-----------------|-----------------|--------------------------------------------------|
| 0               | 0               | Таймер/счетчик T0 отключен от вывода OC0A (OC0B) |
| 0               | 1               | Состояние вывода меняется на противоположное     |
| 1               | 0               | Вывод сбрасывается в 0                           |
| 1               | 1               | Вывод устанавливается в 1                        |



При необходимости состояние выводов OC0A и OC0B (OC1A и OC1B) могут быть изменены в соответствии с табл.5.3 принудительно (Force Output Compare) записью 1 в бит FOC0A и FOC0B (FOC1A и FOC1B) регистра TCCR0B (TCCR1B). Прерывание при этом не генерируется и в режиме CTC не сбрасывается счетчик. При чтении этот вывод будет нулем.

### Режим CTC

Режим CTC (*Clear Timer on Compare* - сброс при совпадении) используется для генерации сигналов заданной частоты. Режим характеризуется следующими процессами:

- >счетчик работает как суммирующий по тактовому сигналу;
- >предельное значение счетчика определяется регистром сравнения A, затем сброс и опять счет начинается с нуля;
- >после совпадения с регистром устанавливается флаг OCF0A (OCF1A), что может при разрешении сопровождаться прерыванием;
- >одновременно с установкой флага может изменяться и состояние выводов OC0A (OC1A) микроконтроллера. Характер изменения зависит от бит COM0A1- COM0A0 (COM1A1- COM1A0). Управление выводами в этом режиме аналогично режиму Normal (см. табл.3.3);
- >совпадение с регистром сравнения B, если его значение не превышает значения регистра A, также сопровождается выработкой прерывания и изменением состояния выхода OC0B (OC1B), но счетчик не сбрасывается.

Диаграммы режима CTC представлены на рис.3.5. Такие диаграммы получаются при задании режима переключения сигналов на выводах (биты управления выводами соответствуют 01). Легко видеть, что период генерации увеличивается по мере увеличения кода в регистрах сравнения. Наклонная линия иллюстрирует увеличение кода счетчика.

Для расчета частоты генерации  $F$  используется выражение

$$F = \frac{f_{CLK}}{2 \cdot N \cdot (1 + OCR0A)},$$

- где  $f_{CLK}$  - тактовая частота микроконтроллера;  
 $N$  - коэффициент деления предделителя;  
 $OCR0A$  - код в регистре сравнения.

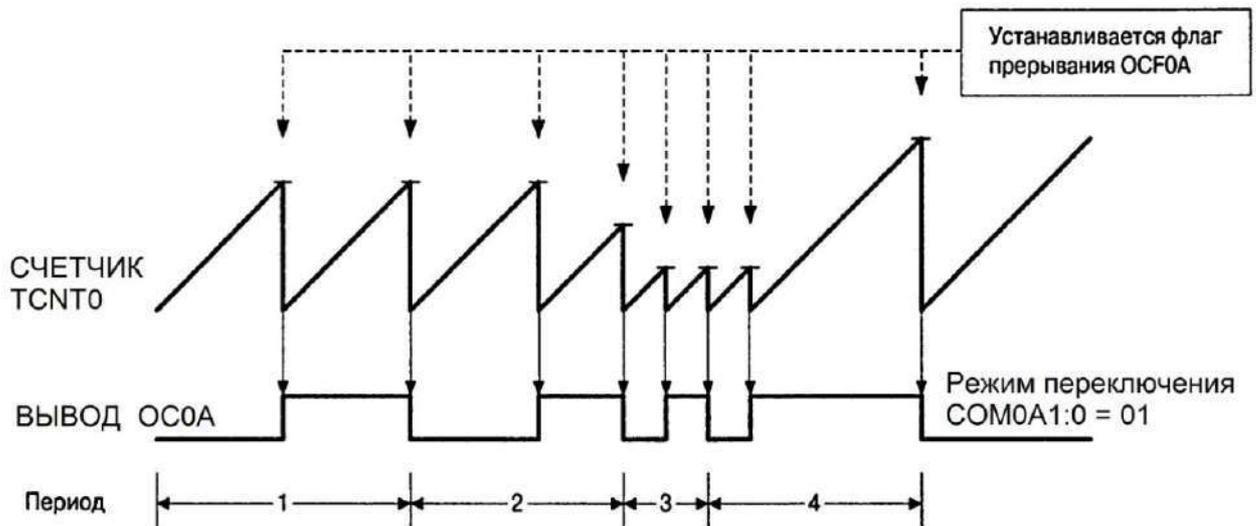


Рис.3.5. Временные диаграммы работы в режиме CTC

### Режим Fast PWM

Режим Fast PWM (*Fast Puls-Wedth Modulation* - быстродействующей широтно-импульсной модуляции - ШИМ) используется для регулирования мощности, выпрямления, цифро-аналогового преобразования и т.д.

Режим характеризуется следующими процессами:

6. счетчик работает в режиме суммирования по тактовому сигналу до максимального значения  $\$FF$ . В альтернативном варианте при режиме №7 счетчика T0 (при  $WGM02=1$ ) - до значения регистра сравнения OCR0A. В этом режиме запись в регистр сравнения происходит только при достижении счетчиком состояния  $\$FF$ . Это называется двойной буферизацией записи. В противном случае появились бы несимметричные импульсы на выходе модулятора (формирователя сигналов, см. рис.3.1);

7. при переполнении (переход через  $\$FF$ ) устанавливается флаг переполнения TOV0;

8. при равенстве содержимого счетчика и регистра сравнения устанавливается флаг совпадения OCF0A или OCF0B;

9. при равенстве содержимого счетчика и регистра сравнения еще происходит управление выводами в соответствии с таблицей 3.4.

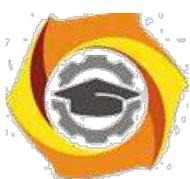


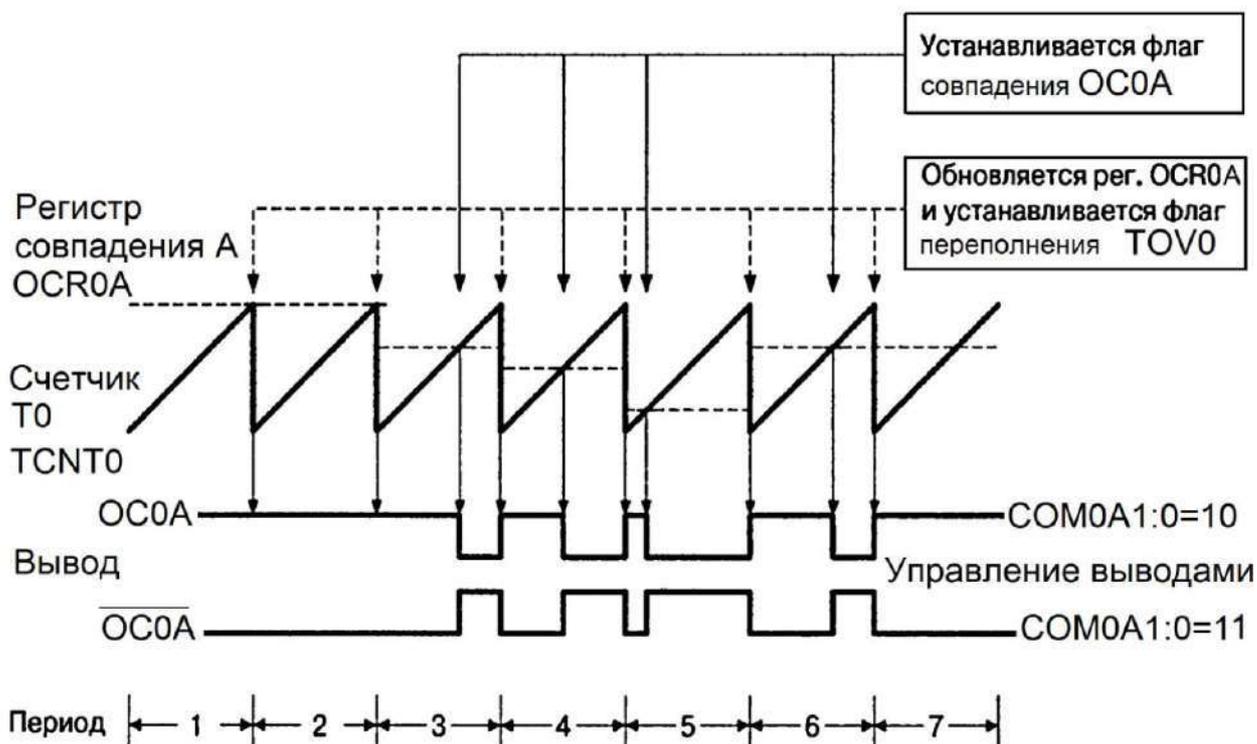
Таблица 3.4

Управление выводами OC0A и OC0B в режиме Fast PWM

| COM0A1 (COM0B1) | COM0A0 (COM0B0) | Описание                                                                                                                                                                                                     |
|-----------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0               | 0               | Таймер/счетчик T0 отключен от вывода OC0A (OC0B)                                                                                                                                                             |
| 0               | 1               | <b>OC0A:</b><br>WGM02 = 0 — таймер/счетчик T0 отключен от вывода OC0A;<br>WGM02 = 1 — состояние вывода меняется на противоположное при равенстве регистров TCNT0 и OCR0A.<br><b>OC0B:</b><br>Зарезервировано |
| 1               | 0               | Сбрасывается в 0 при равенстве регистров TCNT0 и OCR0x. Устанавливается в 1 при достижении счетчиком максимального значения (неинвертированный ШИМ-сигнал)                                                   |
| 1               | 1               | Устанавливается в 1 при равенстве регистров TCNT0 и OCR0x. Сбрасывается в 0 при достижении счетчиком максимального значения (инвертированный ШИМ-сигнал)                                                     |

Диаграммы сигналов в режиме Fast PWM представлены на рис.3.6.

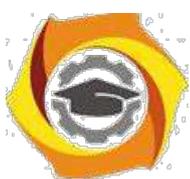
Легко видеть, что при счете до \$FF увеличением кода в регистре сравнения увеличивается степень заполнения единицами периода генерации. При необходимости сигнал на выходе можно проинвертировать, поменяв биты COM0A1- COM0A0 с 10 на 11.





**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Рис.3.6. Диаграммы сигналов в режиме Fast PWM



Частота генерируемого сигнала определяется выражением:

$$F_{OC0A} = \frac{f_{CLK}}{N \cdot 256},$$

где N - коэффициент деления предделителя;  
f<sub>CLK</sub> - тактовая частота микроконтроллера.

### Режим Fast Correct PWM

Режим Fast Correct PWM (ШИМ с точной фазой) предпочтительней для управления двигателями благодаря симметричности изменения состояния счетчика. В этом режиме реализуемые процессы очень похожи на режим Fast PWM, но с учетом выполнения реверсивного счета:

- счетчик суммирует от нуля до максимального значения, затем вычитает до нуля. Возможность ограничения максимального кода счетчика регистром сравнения также присутствует;

С при достижении кода \$00 формируется флаг переполнения TOV0;

С при совпадении с регистром сравнения формируется флаг совпадения

OC0A или OC0B;

С при совпадении с регистром сравнения происходит изменение сигнала на выходе в соответствии с таблицей 3.5.

Таблица 3.5

Управление выводами, OC0A, OC0B в режиме Fast Correct PWM

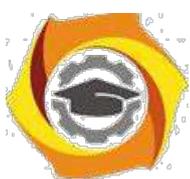
| COM0A1<br>(COM0B1) | COM0A0<br>(COM0B0) | Описание                                                                                                                                                                                                     |
|--------------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0                  | 0                  | Таймер/счетчик T0 отключен от вывода OC0A (OC0B)                                                                                                                                                             |
| 0                  | 1                  | <b>OC0A:</b><br>WGM02 = 0 — таймер/счетчик T0 отключен от вывода OC0A;<br>WGM02 = 1 — состояние вывода меняется на противоположное при равенстве регистров TCNT0 и OCR0A.<br><b>OC0B:</b><br>Зарезервировано |
| 1                  | 0                  | Сбрасывается в 0 при прямом счете и устанавливается в 1 при обратном счете (неинвертированный ШИМ-сигнал)                                                                                                    |
| 1                  | 1                  | Устанавливается в 1 при прямом счете и сбрасывается в 0 при обратном счете (инвертированный ШИМ-сигнал)                                                                                                      |

h) учетом реверсивного счета (за период счетчик два раза перебирает свои состояния) частота выходного сигнала будет в 2 раза меньше частоты в режиме Fast PWM:



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

$$FOC_{0A} = \frac{f_{clk}}{N \cdot 512}$$



Биты FOC0A и FOC0B принудительно изменяют состояние вывода OC0A и OC0B (в режимах Normal и CTC). При записи 1 в этот бит состояние вывода изменяется в соответствии с установками битов COM0A1-COM0A0. Прерывание при этом не генерируется и сброс таймера (в режиме CTC) не производится.

Диаграмма сигналов в режиме Fast Correct PWM показана на рис.3.7.

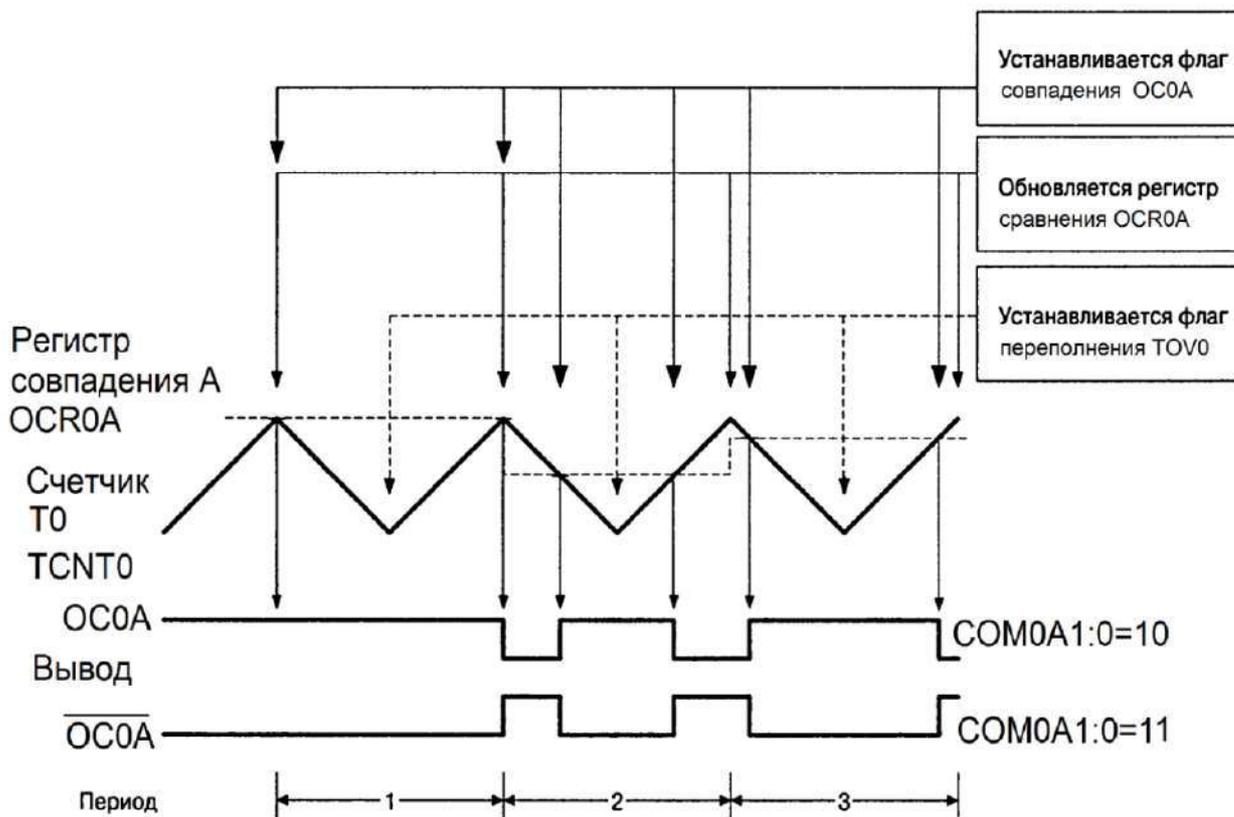


Рис.3.7. Диаграмма сигналов в режиме Fast Correct PWM

Работа счетчика T1 ATtiny2313 аналогична счетчику T0. Но в нем реализован дополнительно режим захвата и используются двухбайтные порты, что увеличивает диапазон используемых кодов. Структурные дополнения T1 по сравнению со счетчиком T0 выделены желтым цветом на рис.3.1.

Запись в двухбайтные регистры счетчика T1 осуществляется за два обращения. Сначала записывается старший байт, который сохраняется в буфере. При записи младшего байта происходит размещение двухбайтного кода в регистр таймера. Содержимое буфера, содержащего старший байт, можно использовать многократно при адресации к младшему байту выбранного двухбайтного регистра.

При чтении 16-битного регистра процесс должен начинаться с младшего байта. При этом старший байт записывается в буфер, из которого и происходит чтение старшего байта.

При обращении к 16-битным регистрам прерывания необходимо запретить во избежание изменения состояния регистров до завершения обращения.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Уточнить детали работы таймера Т1 можно по литературе [2].



### Исходное состояние счетчиков по умолчанию

После формирования команды RESTART исходное положение аппаратуры следующее:

- С устройства выключены;
- С предделитель выключен (счетчик не работает);
- С внешние выходы отключены.

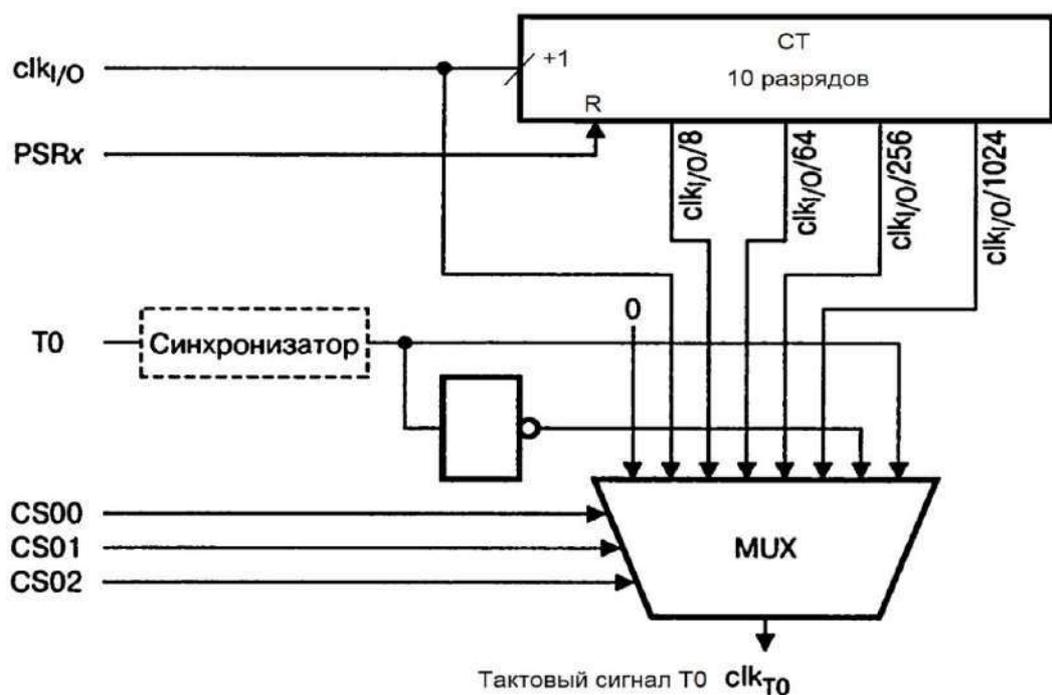
### Задание 3.2. Задание параметров счетного сигнала

Изучить программную модель управления частой работы таймеров-счетчиков, научиться программировать требуемые параметры тактовых сигналов счетчиков-таймеров.

#### В отчете представить:

- 12) Адреса и имена регистров ввода-вывода, обеспечивающих управление предделителем;
- 13) Диапазон коэффициентов деления предделителя;
- 14) Назначение битов регистров управления предделителя.

Предделитель таймеров предназначен для формирования тактовых сигналов таймеров-счетчиков. Функциональная схема предделителя представлена на рис.3.8. Оба таймера используют один предделитель, но управляются тактовые сигналы индивидуально.





**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Рис.3.8. Функциональная схема предделителя таймеров T0 и T1



Управление предделителем заключается в задании коэффициента деления тактовой частоты контроллера, остановке таймеров, переключении на счет внешних сигналов. Возможные источники сигналов для таймера T0 назначаются битами регистра TCCR0B=\$33 (рис.3.3) в соответствии с таблицей 3.6. Для счетчика T1 таблица аналогичная, но управление осуществляется битами в регистре TCCR1B=\$2E (рис.3.4).

Таблица 3.6

Выбор источника тактового сигнала счетчиков T0 и T1

| CS02 | CS01 | CS00 | Источник тактового сигнала T0                                  |
|------|------|------|----------------------------------------------------------------|
| 0    | 0    | 0    | Таймер/счетчик остановлен                                      |
| 0    | 0    | 1    | clk <sub>I/O</sub>                                             |
| 0    | 1    | 0    | clk <sub>I/O</sub> /8                                          |
| 0    | 1    | 1    | clk <sub>I/O</sub> /64                                         |
| 1    | 0    | 0    | clk <sub>I/O</sub> /256                                        |
| 1    | 0    | 1    | clk <sub>I/O</sub> /1024                                       |
| 1    | 1    | 0    | Вывод T0, счет осуществляется по спадающему фронту импульсов   |
| 1    | 1    | 1    | Вывод T0, счет осуществляется по нарастающему фронту импульсов |

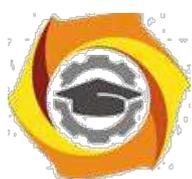
Установкой в 1 бита PSR10 (Prescaler Reset Timer/Counter) Общего регистра управления таймерами/счетчиками GTCCR (General Timer/Counter Control Register) можно сбросить предделитель в ноль.

### ***Задание 3.3. Разработка программы управления счетчиками в режиме Normal***

Разработать схему алгоритма и программу в соответствии с последующими заданиями. Выполнить трансляцию и отладку программы. Продемонстрировать работу программы на эмуляторе и в программе Proteus. Представить данные в отчете в соответствии с каждым заданием.

9) самом общем случае инициализация таймера в режиме Normal (в остальных режимах аналогично) состоит из следующих шагов:

- остановка таймера в TCCR0B (обнуление разрядов CS00 - CS02):  
 $TCCR0B (1 \ll CS02) | (1 \ll CS01) | (1 \ll CS00);$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



- задание режима работы Normal в TCCR0A и TCCR0B без старта (обнуление разрядов WGM02 . . . WGM00):

```
TCCR0A & = ~ (1<<WGM01)|(1<<WGM00); //По умолчанию
TCCR0B & = ~ (1<<WGM02);
```

16. задание режима работы выводов счетчика в TCCR0A (выводы COM0A и COM0B), например, в режим "изменение состояния на противоположное":

```
TCCR0A & = ~ (1<<COM0A0)|(1<< COM0B0);
```

17. установка цифровых портов "на выход" для подключения выводов счетчика OC0A и OC0B (регистры DDRB, DDRD):

```
DDRB |= (1<<PB2) //для вывода OC0A
DDRD |= (1<<PD5) //для вывода OC0B;
```

- установка начального значения счетчика TCNT0 (TCNT1) или регистров сравнения OCR0 (OCR1):

```
TCNT0 = const; //Или OCR0A=... OCR0B= ...;
```

в сброс флагов счетчика-таймера T0 в регистре TIFR:

```
TIFR |= (1<<OCF0B)|(1<<TOV0)|(1<<OCF0A);
```

в разрешение прерываний счетчика-таймера T0 в TIMSK:

```
TIMSK |= (1<<TOIE0)|(1<<OCIE0B)|(1<<OCIE0A);
```

в установка предделителя счетчика-таймера T0 в TCCR0B, то есть старт таймера:

```
TCCR0B |= (1<<CS02)|(0<<CS01)|(1<<CS00).
```

Такие шаги могут быть выполнены в полном объеме или частично, исходя из текущих решаемых задач.

### ***Задание 3.3.1. Генератор сигналов на основе таймера T0***

Запрограммировать счетчик T0 в режиме Normal с максимальной частотой генерации с выхода OC0A. Режим выхода - изменение сигнала на противоположный. Тактовая частота процессора - 4 МГц.

#### ***В отчете представить:***

- В Выражения для расчета частоты и периода генератора;
- В Программу на языке C. Структуру данных управляющих регистров таймера T0;
- В Схему моделирования с использованием осциллографа;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

4. Расчеты частоты генерации по данным AVR Studio;



5. Результаты измерения частоты и периода генерации с помощью осциллографа. Вывод о соответствии данных расчета и измерения.

**Задача 3.3.2.** Изменить программу для генерации сигналов на выходах OC0A и OC0B со сдвигом на четверть периода. Сохранить в программе прежние настройки регистров управления с помощью знаков комментариев.

**В отчете представить:**

- и Изменения в программу задачи 3.3.1.
- и Выражения для задания режимов работы счетчика.
- и Вывод о фазовом сдвиге сигналов.

**Задача 3.3.3.** Уменьшить в 8 раз частоту генерации на выходах OC0A и OC0B. Доказать факт изменения частоты по сигналам на осциллографе и параметрам AVR Studio.

**В отчете представить:**

- В Изменения в программу задачи 3.3.2.
- В Данные расчета частоты генерации по данным AVR Studio и настройкам осциллографа. Вывод об их соответствии.

**Задание 3.4. Разработка программы управления счетчиками в режиме СТС**

Сигнал должен генерироваться на выходах OC0A или OC1A. Целесообразно использовать режим СТС, специально предназначенный для генерации сигналов настраиваемой частоты. Используя конкретный выбранный счетчик контроллера, обосновать возможность и запрограммировать контроллер на выдачу сигнала на выходе регистра А с частотой 1 Гц. Тактовая частота контроллера 4 МГц.

Пример расчета параметров настройки проектируемого генератора на счетчике T1 для частоты задающего генератора  $f_{CLK} = 8$  МГц для выработки сигнала прерывания по переполнению с частотой 1 Гц.

Для расчета следует использовать следующие этапы:

В Количество заполнений счетчика за 1 сек:

$$8\,000\,000 / 65536 = 122 \text{ раз};$$

- Если генератор должен переключается всякий раз по переполнению, то переключений (миганий, периодов за секунду) получится:

$$122 / 2 = 61 \text{ раз};$$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

В Для приближения к 1 Гц необходимо уменьшить исходную тактовую частоту процессора в 64 раза. Это можно выполнить за счет коэффициента делителя  $K=64$ . Поэтому получится частота:



$$61 / 64 = 0,954 \text{ Гц};$$

5. Соответственно, для получения точно 1 Гц следует уменьшить количество отсчетов счетчика в 0,954 раза:

$$65536 * 0,954 = 62521;$$

7. Следовательно, если до переполнения счетчик должен получить 62521 фронт тактового сигнала, то код предустановки:

$$65536 - 62521 = 3015 = 0BC7H.$$

При переключении генератора по сигналу прерывания по переполнению для получения частоты переключения 1 Гц необходимо будет этот код (0BC7H) записывать в счетчик T1 как код предустановки. В обработчике можно с этой же частотой управлять любым внешним устройством (за 1с будет выдаваться два сигнала переполнения). Кроме того на выходе OC1A при установке кода в регистре сравнения OCR1A, например, 0000H также будут происходить переключения с той же частотой в момент сброса счетчика.

Для расчета данных для режима CTC в рамках исходного задания необходимо учесть, что число тактов до сброса определяется кодом в регистре сравнения. Поэтому полученный при расчете код необходимо просто занести в регистр сравнения. Заполнение счетчика будет происходить от нуля до значения регистра сравнения, затем снова с нуля.

Кроме того при инициализации 16-разрядных регистров следует учитывать порядок записи и чтения младшего и старшего байта.

Проверку расчетов следует проводить по данным процессора в AVR Studio. Измерения частоты по данным осциллографа в системе моделирования будут искажены из-за чрезмерной нагрузки процессора.

***В отчете представить:***

- и Расчеты для получения необходимой частоты;
- и Программу для микроконтроллера на языке C;
- и Доказательства правильности расчетов.

***Задание 3.5. Разработка программы управления счетчиками в режиме PWM.***

Запрограммировать счетчик T0 в режиме ***Fast PWM*** с максимальной частотой генерации с выхода OC0A. Длительность единичного сигнала - 50% периода. Тактовая частота процессора - 4 МГц. После отладки и подтверждения результатов на выходе OC0A запрограммировать формирование сигнала на выходе OC0B с заполнением 20% и сравнить результаты в среде Proteus.



**В отчете представить:**

- Выращения для расчета частоты и периода генератора для двух выходов;
- Программу на языке C. Структуру данных управляющих регистров таймера T0;
- Расчеты частоты генерации по данным AVR Studio.
- Диаграмму моделирования с использованием осциллографа;
- Результаты измерения частоты и периода генерации с помощью осциллографа. Вывод о соответствии данных расчета и измерения.

**Задание 3.6. Бегущие огни (задержка таймером без использования прерываний)**

Проанализировать представленную программу. При отсутствии замечаний получить код для программирования микроконтроллера. Смоделировать работу бегущих огней в программе Proteus. При невозможности наблюдать эффект бегущих огней выявить причину и предложить способы устранения этого недостатка.

```
#include <avr/io.h>
void wait1 (void) // Функция задержки
{
 TCNT1=0;
 while (TCNT1<780) {};
}
int main(void) // Главная функция программы
{
 unsigned char rab; // Вводим переменную rab

 PORTB=0xFF; // Инициализация порта B
 DDRB=0xFF;

 PORTD=0x7F; // Инициализация порта D
 DDRD=0x00;

 TCCR1A=0x00; // Инициализация таймера/счетчика 1
 TCCR1B=0x05;

 ACSR=0x80; // Инициализация аналогового компаратора
```



*while (1)*

**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



```
{
 {
 if (rab = 0b100000000; ((PIND &(1<<PD0))) // Проверка состояния
 while (rab!=0) // переключателя
 {
 // Сдвиг вправо
 // Запись начального значения

 PORTB=rab^0xFF; // Запись а порт с инверсией
 rab = rab >> 1; // Сдвиг разрядов
 wait1 (); // Задержка в 200 мСек
 }
 }
 else
 {
 // Сдвиг влево
 // Запись начального значения
 rab = 0b00000001;
 while (rab!=0)
 {
 PORTB=rab^0xFF; // Запись в порт с инверсией
 rab = rab << 1; // Сдвиг разрядов
 wait1 (); // Задержка в 200 мСек
 }
 }
 }
 }
}
```

**В отчете представить:**

- Алгоритм решения задачи;
- Предложения по усовершенствованию схемы;
- Схему устройства в программе Proteus.

**Вопросы для самоконтроля**

- и Какое назначение имеют таймеры/счетчики?
- и Какие адреса занимают порты счетчиков контроллера?
- и Какие регистры управляют счетчиками?
- и Какие режимы можно задать счетчикам?
- и Какова типовая программная модель счетчиков?
- и Как осуществляется инициализация стека на языке ассемблера и языке C?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- и Какое состояние имеют регистры ввода-вывода счетчиков?
- и Какие режимы и каково исходное состояние счетчиков после команды RESET?



- Почему вместо программного формирования временного интервала используют автономный счетчик?
- 10. Какими способами можно изменить частоту переполнения счетчиков?
- 11. Докажите, что на счетчике T0 построить генератор 1 Гц невозможно.
- 12. В каком случае вырабатывается сигнал сравнения с регистром В счет-чика в режиме СТС?



## • ИССЛЕДОВАНИЕ ПРЕРЫВАНИЙ МИКРОКОНТРОЛЛЕРА

### Цель:

С Углубить и закрепить теоретические знания по принципам организации прерываний специализированных контроллеров;

С Приобрести практические навыки самостоятельного использования прерываний для решения задач;

С Совершенствовать навыки анализа, обобщения и систематизации полученных результатов, навыки составления и оформления отчетных материалов, навыки точного и лаконичного представления докладов на вопросы технического характера.

### Учебные вопросы:

- 4.1. Организация прерываний в AVR микроконтроллерах;
- 4.2. Использование внешних прерываний;
- 4.3. Использование прерываний счетчиков.

### Литература для подготовки к занятию

- С Белов А.В. Создаем устройства на микроконтроллерах. - СПб.: Наука и техника, 2007. - 304 с.
- С Евстифеев А.В. Микроконтроллеры AVR семейства Tiny. Руководство пользователя. - М.: Издательский дом Додэка-XXI, 2007. - 432с.
- С Ревич Ю.В. Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. СПб.: БХВ-Петербург, 2011. - 352с.
- С Егоров А.А. и др. Программирование микроконтроллеров семейства AVR. Учебно-методическое пособие. - М.: Издательство МАИ, 2004. - 72с.

### Содержание отчета

- Название работы.
- Название каждого учебного вопроса и краткий конспект в объеме практических заданий по вопросу.

### Вопросы для подготовки к занятию

- Что называется прерыванием?
- Источники прерываний?
- Что называется контекстом при осуществлении прерывания?
- Что такое маска прерывания? С какой целью используется?
- Что называется приоритетом прерываний?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Какие прерывания используются в современных ЭВМ?
- Этапы механизма прерывания.



### Актуальность занятия

Микроконтроллер работает в режиме реального времени. Это требует реализации точных временных интервалов, при этом реализация их должна осуществляться без дополнительной загрузки процессора. Кроме того актуальными являются задачи генерации сигналов произвольной частоты, сигналов широтно-импульсной модуляции, осуществления АЦП и ЦАП преобразований и т.д. Все эти задачи решают встроенные таймеры-счетчики.

### **Задание 4.1. Организация прерываний в AVR микроконтроллерах**

Изучить организацию прерываний в микроконтроллерах AVR, состав программных средств для управления прерываниями МКК ATtiny2313, способы получения справочной информации.

В отчете представить:

- Размещение и структуру таблицы векторов прерываний;
- Способы организации прерываний на языке ассемблера и C в виде примеров программ;
- Состав регистров для управления прерываниями в МКК ATtiny2313;
- Путь с макрорасширением с описанием векторов прерываний МКК.

### **Назначение и механизм реализации прерываний AVR**

Прерывание (Interrupt) – сигнал, свидетельствующий о наступлении какого-либо события. При этом выполнение текущей последовательности команд приостанавливается, и управление передается процедуре обработки прерывания (обработчику), соответствующей данному событию. После завершения обработчика исполнение кода продолжается с команды, которая является следующей после той, на которой произошло прерывание.

Прерывания позволяют организовать и синхронизировать параллельно (одновременно) протекающие процессы.

Для реализации данной логики работы процессор выполняет следующие действия:

- По сигналу прерывания процессор аппаратно запрещает все прерывания (сбрасывает глобальный флаг I в регистре состояния SREG) и сохраняет в стеке адрес следующей команды. Адрес следующей команды определяется как сумма текущего состоянию счетчика команд плюс длина последней команды, на которой произошло прерывание. Локальный флаг, вызвавший прерывание, должен быть сброшен программно в обработчике прерывания;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- В счетчик команд загружается адрес вектора прерывания, соответствующий данному событию. Все события, вызывающие прерывания, имеют свой уникальный и неизменный адрес в таблице векторов. ***Вектором преры-***

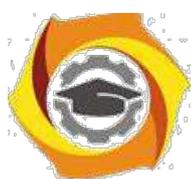


**вания** называют код в таблице векторов, который закреплен за данным событием (это фиксированная строка программной области, определяющая, куда переходит программа в случае возникновения прерывания). По этому адресу, как правило, находится команда **безусловного перехода к подпрограмме** обработки прерывания (обработчику). Адреса векторов прерываний для микроконтроллера ATtiny2313 приведены в таблице 4.1. Всего в этом контроллере 19 прерываний. Прерывание с номером 0 особое, оно не имеет флага, не может быть маскировано. Уровень приоритета уменьшается по мере увеличения адреса размещения вектора прерывания.

Таблица 4.1

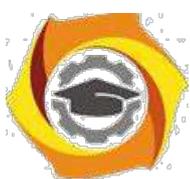
Прерывания микроконтроллера ATtiny2313

| Номер вектора | Адрес перехода | Источник     | Описание прерывания                                                                 |
|---------------|----------------|--------------|-------------------------------------------------------------------------------------|
| 0             | 0x0000         | RESET        | Внешний сброс, сброс при включении питания, сброс по срабатыванию охранного таймера |
| 1             | 0x0001         | INT0         | Внешний запрос на прерывание по входу INT0                                          |
| 2             | 0x0002         | INT1         | Внешний запрос на прерывание по входу INT1                                          |
| 3             | 0x0003         | TIMER1_CAPT1 | Прерывание по захвату таймера/счетчика 1                                            |
| 4             | 0x0004         | TIMER1_COMPA | Прерывание по совпадению таймера/счетчика 1, канал A(1=? в f)                       |
| 5             | 0x0005         | T1MER1_OVF1  | Прерывание по переполнению таймера/счетчика 1                                       |
| 6             | 0x0006         | TIMER0_OVF0  | Прерывание по переполнению таймера/счетчика 0                                       |
| 7             | 0x0007         | UART_RX      | USART, прием завершен                                                               |
| 8             | 0x0008         | UART_UDRE    | USART, буфер данных пуст                                                            |
| 9             | 0x0009         | UART_TX      | USART, передача завершена                                                           |
| 10            | 0x000A         | ANA_COMP     | Прерывание от аналогового компаратора                                               |
| 11            | 0x000B         | PCINT        | Прерывание по изменению на любом из выводов*                                        |
| 12            | 0x000C         | TIMER1_COMPB | Прерывание по совпадению таймера/счетчика1, канал B*                                |
| 13            | 0x000D         | TIMER0_COMPA | Прерывание по совпадению таймера/счетчика0, канал A*                                |
| 14            | 0x000E         | TIMER0_COMPB | Прерывание по совпадению таймера/счетчика 0, канал B*                               |



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

|    |        |           |                                 |
|----|--------|-----------|---------------------------------|
| 15 | 0x000F | USI_START | Прерывание по USI. Готовность к |
|----|--------|-----------|---------------------------------|



|    |        |              |                                  |
|----|--------|--------------|----------------------------------|
|    |        |              | старту*                          |
| 16 | 0x0010 | USI_OVERFLOW | Прерывание по USI, Переполнение* |
| 17 | 0x0011 | EE_READY     | Готовность EEPROM                |
| 18 | 0x0012 | WDT_OVERFLOW | Переполение охранного таймера    |

Имена прерываний и различных констант можно уточнить в заголовочном файле данного микроконтроллера (*C:\Program Files\Atmel\AVR Tools\AVR Toolchain\avr\include\avr\io2313.h*).

Имен векторов может быть несколько, но сделано это исключительно для совместимости в разных версиях компиляторов.

- Занесением адреса вектора прерывания в счетчик команд фактически заканчивается выполнение аппаратных действий процессора на сигнал прерывания. На деле это означает, что следующей командой будет выполняться команда, содержащаяся в таблице векторов по адресу, закрепленному за этим событием (номером): безусловный переход к подпрограмме обработки данного прерывания;

- Подпрограмма обработки прерывания (обработчик) выполняет все необходимые команды для свершившегося события и заканчивается командой выхода из подпрограммы обработчика прерывания *RETI - RETurn from Interrupt*. По этой команде восстанавливается общее разрешение на прерывания (флаг I регистра SREG устанавливается в 1), и верхние данные стека заносятся в счетчик команд. Если стек не нарушен, то там должен находиться адрес возврата из подпрограммы обработки прерывания. Поэтому эти действия эквивалентны передаче управления на команду, которая является следующей после той, на которой прерывание произошло. Выполнение этой команды занимает четыре такта. После выхода из прерывания процессор всегда выполняет одну команду основной программы, прежде чем обслужить любое отложенное прерывание.

### Особенности конструирования обработчиков

Наступление прерывания в произвольный момент времени определяет, что любой обработчик не может принимать и передавать каких-либо параметров. Просто потому, что момент прерывания случаен и событие, вызвавшее прерывание, также неизвестно заранее.

Внутри обработчика прерывания можно программно разрешить или запретить общее прерывание. Для этого соответственно командой *SEI (Set Enable Interrupt)* установить 1 во флаг общего прерывания I регистра состояния SREG или программно запретить прерывания командой *CLI (Clear Interrupt)* - очистить, т.е. сбросить в 0 флаг I. На языке C эти команды



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

соответственно записываются в виде макроса *sei()* или *cli()*. Макросы просто подставляют в исходный текст соответствующую ассемблерную команду. Но



при этом, разрешив прерывания, можно получить вложенные прерывания с непредсказуемыми последствиями или разрушение стека. Поэтому пользоваться этим следует с ясным пониманием последствий. В частности, понимая, что запрещая прерывания, закладываем запаздывание в обработке возможных прерываний, по крайней мере, до окончания выполняющегося обработчика.

Другие прерывания не потеряются. Все последующие прерывания, которые могут поступить во время выполнения обработчика и глобального запрета обработки прерываний, установят свои флаги. После разрешения общего прерывания они сгенерируют свои прерывания и будут обработаны. Обработка осуществляется по приоритету. Первым обрабатывается то прерывание, чей вектор меньше по адресу, ближе к началу памяти. Если за время запрета поступило несколько прерываний от одного источника, то обработка произойдет, естественно, только один раз. Очереди одинаковых прерываний не предусмотрены.

При возникновении прерывания устанавливается флаг, соответствующий этому событию (его бит устанавливается в 1). При запуске подпрограммы обработки прерывания он аппаратно сбрасывается 0 для фиксации последующих событий. ***Программно сбросить флаг в ноль можно, записав в него 1. Именно 1 !!!***

Если прерывание произойдет в спящем режиме микроконтроллера (sleep mode), то время реакции помимо обычных четырех тактов увеличивается еще на четыре такта, плюс время, заложенное во фьюз битах SUT1 и SUT0 (Start-Up Time).

Момент поступления прерывания неизвестен. Поэтому все, что использовалось в прерванном процессе должно быть обязательно сохранено после завершения обработчика. Для этого в начале обработчика последовательно сохраняют в стеке (команда PUSH) регистр состояния SREG и все те регистры, которые задействованы в обработчике. Перед завершением обработчика в обратном порядке их восстанавливают (команда POP).

Прерывания отвлекают процессор и блокируют другие прерывания. Поэтому обработчики должны быть максимально короткими и быстрыми. Все другие длинные процедуры должны выполняться в фоновом режиме. Поэтому при необходимости обработки массивов данных следует обработчиком сохранить банные в буфере, а обработку осуществлять за пределами обработчика.

Прерывания происходят в произвольный момент времени и зависят от автономно протекающих процессов, а результат их работы еще и от момента возникновения в основном коде. По этой причине некорректно организован-



ные прерывания могут носить характер сбоев и очень трудно поддаваться диагностированию. Поэтому все, что связано с прерываниями должно быть изучено глубоко и очень конкретно.

### Таблица векторов прерываний

Таблица векторов прерываний (упорядоченный по приоритету список векторов прерываний) располагается с адреса 0x00 области памяти программ. Он не стирается при пропадании питания.

Адресуемая строка в программной области представлена 2-байтным словом. В семействе ATtiny строка вектора прерывания занимает 2 байта, а в семействе Mega занимают 4 байта. С учетом этого для ATtiny2313 программный код может выглядеть следующим образом:

```
.include "tn2313def.inc"
.cseg ;Выбор сегмента кодов (FLASH)
.org 0 ;Указатель текущего адреса PC

rjmp Reset ; Внешний сброс
rjmp INT_0 ; Внешний запрос по входу INT0
rjmp INT_1 ; Внешний запрос по входу INT1
rjmp Timer1_capt1 ; Захват таймера/счетчика 1
rjmp Timer1_compA ; Совпадению таймера/счетчика 1 канал A
rjmp Timer1_OVF1 ; Переполнение таймера/счетчика 1
rjmp Timer0_OVF0 ; Переполнение таймера/счетчика 0
rjmp UART_RX ; USART, прием завершен
rjmp UART_UDRE ; USART, буфер данных пуст
rjmp UART_TX ; USART, передача завершена
rjmp ANA_COMP ; Аналоговый компаратор
rjmp PCINT ; Изменение на любом из выводов*
rjmp Timer1_compB ; Совпадению T1, канал B*
rjmp Timer0_compA ; Совпадению T0, канал A*
rjmp Timer0_compB ; Совпадению T0, канал B*
rjmp USI_START ; USI. Готовность к старту*
rjmp USI_OVERFLOW ; USI, Переполнение*
rjmp EE_READY ; Готовность EEPROM
rjmp WDT_OVERFLOW ; Переполнение сторожевого таймера
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Прерывание Reset (Сброс), в отличие от всех остальных, нельзя запретить. Такие прерывания еще называют немаскируемыми NMI (Non-maskable Interrupts). Это прерывание вызывается пятью событиями:



- включение питания (Power-on Reset). Микроконтроллер находится в состоянии сброса, пока напряжение питания находится ниже порогового значения VPOT;
- внешний сброс (External Reset). Микроконтроллер находится в состоянии сброса, пока на вывод RESET подан низкий уровень;
- переполнение сторожевого таймера (Watchdog Reset). Микроконтроллер сбрасывается через некоторый период времени, заданный в сторожевом таймере;
- понижение напряжения (Brown-Out Reset). Микроконтроллер сбрасывается, когда напряжение питания VCC опускается ниже порогового значения VBOT и когда Brown-Out детектор включен;
- JTAG сброс. Микроконтроллер находится в состоянии сброса, пока в однобитном регистре Reset Register находится значение логической единицы. Используется при программировании.

Если используются не все прерывания, то в таблице векторов вместо команд относительного перехода *rjmp* **МЕТКА** для всех неиспользуемых прерываний записывается команда выхода из обработчика *reti*. В этом случае создаются так называемые заглушки: обращение к таблице векторов происходит, а там сразу выполняется выход из обработчика, что обеспечивает реализацию всего механизма обработки прерывания, но по факту ничего не происходит (выполняется пустой обработчик).

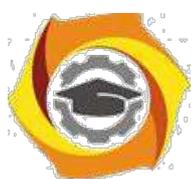
Более компактная запись может получиться при использовании псевдокоманды указателя текущего адреса **ORG**. С его помощью задается адрес в таблице векторов для используемого прерывания и определяется начало адресов кодов основной программы:

```
.cseg ; Выбор сегмента кодов (FLASH)
.org 0 ; Указатель текущего адреса PC

rjmp Reset ; Внешний сброс

.org 2 ; Указание адреса INT_1
; Альтернативный вариант указания адреса
; .org INT1addr Адрес внешнего INT_1
; В файле описаний: .equ INT1addr=$002
rjmp INT_1 ; Внешний запрос по входу INT1
...
.org INT_VECTORS_SIZE ; Конец таблицы прерываний
; Для ATtuny2313 = 0x0013

INT_1: ; Обработчик INT_1
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

...



...  
*reti*

```
RESET: ldi R16, RAMEND ;Инициализация стека =0xDF
 out SPL, R16
 ...
```

```
В инициализация внешнего прерывания INT1 по переднему
фронту ldi R17, (1<<ISC011)|(1<<ISC10) ; Код 0x0C
 out MCUCR, R17
 ldi R16, (1<<INTF1) ; Очистка флагов внеш. прерываний
 out EIFR, R16
 ldi R16, 1<<INT1 ; разрешение внеш. прер.INT1
 out GIMSK, R16 ; Код 0x80

 sei ; Глобальное разрешение прерыв.
```

```
forever: nop ; Пустая команда (no operation)
 rjmp forever ; Бесконечный цикл
```

Легко заметить, что обработчики прерываний располагаются за таблицей векторов. В самом низу размещается секция инициализации и основное тело программы.

8. исходном состоянии МКК (после RESET) все прерывания запрещены (бит I регистра SREG равен 0) - глобальный запрет. Кроме того, запрещены все прерывания на локальном уровне - сброшены в 0 все биты маски каждого прерывания индивидуально. Разрешение прерываний осуществляется индивидуально. Такие разрешения на глобальном и локальном уровнях следует делать программно. Пример такого разрешения для INT1 представлен в предыдущем варианте программы.

## Организация прерываний на языке C

Для использования прерываний необходимо подключить библиотеку прерываний с помощью макросов, в которых описаны порты и все векторы прерываний:

```
#include <avr/io.h>
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
#include <avr/interrupt.h>
```



Процедура обработки прерывания не может ничего принимать в качестве аргумента, а также не может ничего возвращать. Это связано с тем, что все прерывания в AVR асинхронные, а источники и приемники данных не известны. Кроме того, это сокращает время обработки прерывания.

Процедуры обработчиков прерываний в AVR Studio (AVR GCC), объявляются перед основной программой с использованием макроопределения *ISR* (*Interrupt Service Routine* - обработчик прерываний):

```
ISR (<имя вектора прерывания_vect>)
{
 /*код обработчика прерывания*/
}
```

Каждому прерыванию соответствует вектор прерывания. Например, обработчик прерывания по совпадению значения таймера со значением регистра OCR1A будет иметь следующий вид:

```
ISR (TIMER1_COMPA_vect)
{
 /*код обработчика*/
}
```

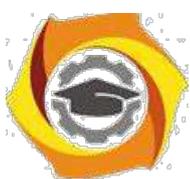
Если используется несколько обработчиков с одним кодом, то можно сослаться на одну и ту же процедуру обработки прерывания:

```
ISR(INT0_vect)
{
 14. Обработчик
}
ISR(INT1_vect, ISR_ALIASOF(INT0_vect));
```

**Таблица векторов строится компилятором самостоятельно.** Неиспользуемые векторы часто заполняются по адресу вектора RESET. Поэтому случайные вызовы несуществующего прерывания приводит к сбросу МКК.

Большинство Си-компиляторов самостоятельно вычисляют занятые в обработчике регистры и перед началом и завершением выполнения тела процедуры обработки прерывания вставляют дополнительные "подпрограммы сохранения/восстановления регистров" общего назначения (РОН) и регистра состояния (SREG).

Программно управление глобальным прерыванием осуществляется с помощью макросов *sei()* - разрешить и *cli()* - запретить. Локальные прерыва-



ния управляются, как и на ассемблере, с помощью записи битов в регистры маски прерываний соответствующих источников.

### Программная модель контроллера для управления прерываниями

Разрешение/запрещение прерываний в контроллере осуществляется глобально путем установки в 1/0 флага общего разрешения прерываний - бит *I* - *Interrupt* Регистра состояния SREG (рис.4.1). Манипуляции выполняются командами *sei/cli*, а также локальным (индивидуальным) разрешением/запрещением путем установки 1/0 в бит маски соответствующего источника прерываний.

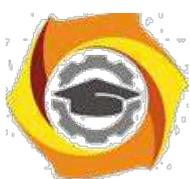
Кроме того, сброс конкретного локального и глобального разрешений осуществляется аппаратно при вызове конкретного прерывания, а затем аппаратно разрешается глобальное прерывание при выходе из обработчика.

| SREG=\$3F Регистр состояния |                           |                    |            |                                   |                        |           |               |
|-----------------------------|---------------------------|--------------------|------------|-----------------------------------|------------------------|-----------|---------------|
| 7                           | 6                         | 5                  | 4          | 3                                 | 2                      | 1         | 0             |
| I                           | T                         | H                  | S          | V                                 | N                      | Z         | C             |
| Общее разрешение прерываний | Хранение копируемого бита | Половинный перенос | Флаг знака | Переполнение дополнительного кода | Отрицательное значение | Флаг нуля | Флаг переноса |

Рис.4.1. Регистр состояния *SREG* - *Status Register*

Для управления прерываниями таймеров используются регистры:

- регистр маски прерываний от счетчиков TIMSK и
- регистр флагов прерываний от счетчиков TIFR (рис.4.2).



| TIMSK=\$39 Регистр МАСКИ прерываний от счетчика<br><i>Timer/Counter Interrupt Mask Register</i> |                                  |                                  |   |                                 |                                  |                                  |                                  |
|-------------------------------------------------------------------------------------------------|----------------------------------|----------------------------------|---|---------------------------------|----------------------------------|----------------------------------|----------------------------------|
| 7                                                                                               | 6                                | 5                                | 4 | 3                               | 2                                | 1                                | 0                                |
| TOIE1                                                                                           | OCIE1A                           | OCIE1B                           | - | TICIE1                          | OCIE0B                           | TOIE0                            | OCIE0A                           |
| Разрешен прерыван по переполн T1                                                                | Разрешен прерыван по совпад A T1 | Разрешен прерыван по совпад B T1 |   | Разрешен прерыван по Захвату T1 | Разрешен прерыван по совпад B T0 | Разрешен прерыван по переполн T0 | Разрешен прерыван по совпад A T0 |

| TIFR=\$38 Регистр ФЛАГОВ прерываний от счетчика<br><i>Timer/Counter Interrupt Flag Register</i> |                                 |                                  |   |                                |                                  |                                 |                                  |
|-------------------------------------------------------------------------------------------------|---------------------------------|----------------------------------|---|--------------------------------|----------------------------------|---------------------------------|----------------------------------|
| 7                                                                                               | 6                               | 5                                | 4 | 3                              | 2                                | 1                               | 0                                |
| TOV1                                                                                            | OCF1A                           | OCF1B                            | - | TICF1                          | OCF0B                            | TOV0                            | OCF0A                            |
| Флаг прерыван по переполн T1 №5                                                                 | Флаг прерыван по совпад A T1 №4 | Флаг прерыван по совпад B T1 №12 |   | Флаг прерыван по Захвату T1 №3 | Флаг прерыван по совпад B T0 №14 | Флаг прерыван по переполн T0 №6 | Флаг прерыван по совпад A T0 №13 |

Рис.4.2. Регистры управления прерываниями счетчиков T0 и T1

Источниками прерываний от счетчиков могут быть:  
и переполнение счетчика T0 и T1 (флаг **TOV** - *Timer/Counter Overflow Flag* и разрешение прерывания **TOIE** - *Timer/Counter Overflow Interrupt Enable*);

совпадение кода счетчика с регистром A и B счетчиков T0 и T1 (флаг **OCF** - *Output Compare Flag* и разрешение прерывания **OCIE** - *Output Compare Interrupt Enable*);

изхват таймера счетчика T1 (флаг **TICF** - *Timer/Counter Input Compare Flag* и разрешение прерывания **TICIE** - *Timer/Counter Input Compare Interrupt Enable*). Осуществляется это прерывание по сигналу внутреннего компаратора или внешнего сигнала счетчика **ICP** - *Input Capture Pin*.

**Для управления внешними прерываниями** используются регистры:

- о общий регистр маски прерываний GIMSK и
- о регистр флагов внешних прерываний EIFR (рис.4.3);
- о регистр состояния микроконтроллера MCUCR (рис.4.4).

Источниками внешних прерываний могут быть:

- сигналы на входах внешних прерываний **INT0** и **INT1** (флаг **INTF** - *Interrupt Flag* и разрешение прерывания **INT** - *Interrupt*). Тип активного сигнала на этих входах определяется битами регистра MCUCR (см. рис.4.4), комбинации которых приведены в таблице 4.2;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- сигналы изменения состояния выводов (флаг *PCIF* - *Pin Change Interrupt Flag* и разрешение прерывания *PCIE* - *Pin Change Interrupt Enable*).



| GIMSK=\$3B Общий регистр МАСКИ прерываний<br><i>General Interrupt Mask Register</i> |                                     |                                              |   |   |   |   |   |
|-------------------------------------------------------------------------------------|-------------------------------------|----------------------------------------------|---|---|---|---|---|
| 7                                                                                   | 6                                   | 5                                            | 4 | 3 | 2 | 1 | 0 |
| INT1                                                                                | INT0                                | PCIE                                         |   |   |   |   |   |
| Разрешение внешнего прерывания INT1                                                 | Разрешение внешнего прерывания INT0 | Разрешение прерывания по изменению выводов 0 |   |   |   |   |   |

| EIFR=\$3A Регистр ФЛАГОВ внешних прерываний<br><i>External Interrupt Flag Register</i> |                               |                                      |   |   |   |   |   |
|----------------------------------------------------------------------------------------|-------------------------------|--------------------------------------|---|---|---|---|---|
| 7                                                                                      | 6                             | 5                                    | 4 | 3 | 2 | 1 | 0 |
| INTF1                                                                                  | INTF0                         | PCIF                                 |   |   |   |   |   |
| Флаг внешнего прерывания INT1                                                          | Флаг внешнего прерывания INT0 | Флаг прерывания по изменению выводов |   |   |   |   |   |

Рис.4.3. Регистры управление внешними прерываниями

| MCUCR=\$35 Общий регистр управления микроконтроллера<br><i>MCU Control Register</i> |   |   |   |                                            |       |                                            |       |
|-------------------------------------------------------------------------------------|---|---|---|--------------------------------------------|-------|--------------------------------------------|-------|
| 7                                                                                   | 6 | 5 | 4 | 3                                          | 2     | 1                                          | 0     |
| -                                                                                   | - | - | - | ISC11                                      | ISC10 | ISC01                                      | ISC00 |
|                                                                                     |   |   |   | Условия генерации внешнего прерывания INT1 |       | Условия генерации внешнего прерывания INT0 |       |

Рис.4.4. Общий регистр управления микроконтроллером

Таблица 4.2

Типы активных сигналов для генерации внешних прерываний

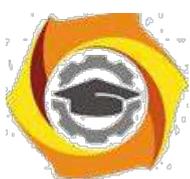
| Бит                        | Описание                                                                       |                                               |                                             |
|----------------------------|--------------------------------------------------------------------------------|-----------------------------------------------|---------------------------------------------|
| ISC01:ISC00<br>ISC11:ISC10 | Определяют условие генерации внешних прерываний INT0 и INT1 следующим образом: |                                               |                                             |
|                            | <b>ISCn1</b>                                                                   | <b>ISCn0</b>                                  | <b>Условие</b>                              |
|                            | 0                                                                              | 0                                             | По НИЗКОМУ уровню на выводе INTn            |
|                            | 0                                                                              | 1                                             | При любом изменении сигнала на выводе INTn  |
|                            | 1                                                                              | 0                                             | По спадающему фронту сигнала на выводе INTn |
| 1                          | 1                                                                              | По нарастающему фронту сигнала на выводе INTn |                                             |

*ISC* - *Interrupt Sense Control* - управление чувствительностью.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Изменение сигналов для формирования прерываний фиксируется на входах, совпадающих с портом В микроконтроллера. Эти сигналы могут



маскироваться с помощью Регистра маски прерываний по изменению состояний выводов PCMSK (рис.4.5), и не формировать прерывания.

| PCMSK=\$20 Регистр МАСКИ прерываний по изменению состояний выводов<br><i>Pin Change Mask Register</i> |                                          |                                          |                                          |                                          |                                          |                                          |                                          |
|-------------------------------------------------------------------------------------------------------|------------------------------------------|------------------------------------------|------------------------------------------|------------------------------------------|------------------------------------------|------------------------------------------|------------------------------------------|
| 7                                                                                                     | 6                                        | 5                                        | 4                                        | 3                                        | 2                                        | 1                                        | 0                                        |
| PCINT7                                                                                                | PCINT6                                   | PCINT5                                   | PCINT4                                   | PCINT3                                   | PCINT2                                   | PCINT1                                   | PCINT0                                   |
| Разрешение прерывания по изменению сост.                                                              | Разрешение прерывания по изменению сост. | Разрешение прерывания по изменению сост. | Разрешение прерывания по изменению сост. | Разрешение прерывания по изменению сост. | Разрешение прерывания по изменению сост. | Разрешение прерывания по изменению сост. | Разрешение прерывания по изменению сост. |

Рис.4.5. Регистр маски прерываний по изменению состояний выводов

При наличии единицы в разряде PCINTx (*Pin Change Interrupt*) изменение состояния на входе для этого бита вызовет прерывание.

Внешние прерывания генерируются даже при конфигурировании соответствующих выводов на выход. Это дает возможность генерировать прерывания программно. Выводы микроконтроллера представлены на рис.4.6.

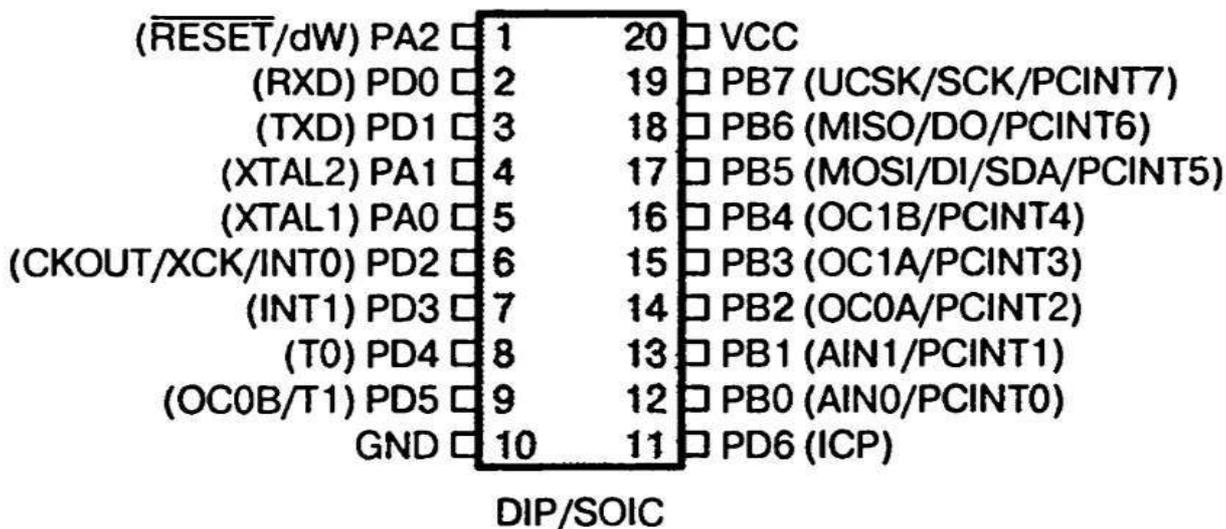


Рис.4.6. Выводы микроконтроллера ATtiny2313

### ***Задание 4.2. Использование внешних прерываний***

#### ***Задание 4.2.1. Зажигание и гашение светодиода кнопкой***

Нажатием кнопки необходимо зажигать, а следующим нажатием гасить светодиод. Одиночным нажатием считается нажатие кнопки с последующим



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

отпускаяем. Для фиксации нажатия кнопки использовать прерывание INT1. Оно должно изменить флаг, по значениям которого должен включаться и вы-



ключаться светодиод. Флаг должен быть задан глобально. Разряд для управления светодиодом выбрать самостоятельно.

Проанализировать представленную программу. При отсутствии замечаний получить код для программирования микроконтроллера и проверить его работу в отладчике. Смоделировать работу кнопки в программе Proteus.

Если схема работает неправильно, то доработать так, чтобы зажигание и погасание светодиода происходило по заднему фронту, формируемому кнопкой.

Вариант программы:

```
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

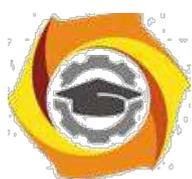
int num = 1; // Глобальная переменная
ISR(INT1_vect){ //Обработчик INT1
 if(num == 1)
 num = 0;
 else
 num = 1;
}

int main (void) //Основная процедура
{
 DDRB = 0xFF; // Порт В - выход
 MCUCR = (0<<ISC11) | (1<<ISC10); //Настройка на любое изменение
 GIMSK |= (1<<INT1); // Разрешение локальное, INT1

 sei(); // Разрешение прерыв. глобальное

 while(1){
 if(num == 1)
 PORTB |= (1<<PB0); // Включаем PB0
 else
 PORTB &= ~(1<<PB0); // Выключаем PB0

 _delay_ms(100); // Ждём 100мс
 }
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

}



***В отчете представить:***

- В Алгоритм и программу решения задачи;
- В Предложения по усовершенствованию схемы;
- В Электрическую схему в Proteus.

***Задание 4.2.2. Зажигание и гашение светодиода разными кнопками***

Разработать и отладить программу, которая позволяет включать светодиод одной кнопкой, а гасить - другой. Для подключения кнопок использовать внешние прерывания INT0, INT1. Работу программы проверить в программе Proteus.

***В отчете представить:***

- с Алгоритм и программу решения задачи;
- с Предложения по усовершенствованию схемы;
- с Электрическую схему в Proteus (доработать схему из задания 4.2.1).

***Задание 4.3. Использование прерываний счетчиков***

***Задание 4.3.1. Прерывание счетчика по совпадению***

Написать программу, обеспечивающую мигание светодиодом с частотой 1Гц с использованием прерываний счетчика по совпадению регистра А. Режим счетчика - Normal. Собрать схему в Proteus и убедиться в ее работоспособности. Тактовая частота процессора 4 МГц.

Если коэффициент делителя выбрать 64, то частота тактирования счетчика TCNT1 вычисляется:  $4\text{МГц}/64=62,5\text{КГц}$ , что соответствует периоду  $16\text{ мкс}$ .

Для периода прерывания по совпадению 500 мс таймер получит значение:  $500\text{мс}/16\text{мкс}=31250$ , или  $0x7A12$ . Это число вполне допустимо для 16-битного регистра.

Отладить схему. Увеличить период мигания до 1с. Замерить период с помощью осциллографа и сравнить с расчетными данными AVR Studio. Проверить работу программы с обработчиком:

Первичный вариант программы:

```
#include <avr/io.h>
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
#include <avr/interrupt.h>
```



```
ISR //Обработчик прерывания
(TIMER1_COMPA_vect); {
 TCNT1=0; //Обнуление счетчика

 if(PORTB &(1<<PB0))
 PORTB&=~(1<<PB0);
 else
 PORTB|=(1<<PB0);
}

int main(void) //Основная программа
{
 DDRB=0xFF;
 PORTB=0;

 OCR1A=0x7A12; //Код совпадения в регистре А
 TIMSK1=(1<<OCIE1A); // Разрешить по совпадению
 TCCR1B=(1<<CS11)|(1<<CS10) ;Предделитель: K=64

 sei(); //Глобальное разрешение прерываний

 while(1);
}
```

***В отчете представить:***

6. Схему подключения светодиода;
7. Расчетные соотношения для периода 1 сек;
8. Схему алгоритма программы;
9. Откорректированный текст программы с комментариями.

***Задание 4.3.2. Генерация ШИМ с помощью таймера***

Разработать и отладить программу и схему, обеспечивающих плавное изменение яркости горения светодиода. Использовать режим широтно-импульсной модуляции (ШИМ) Fast PWM.

Проверить работу программы в отладчике и программе Proteus.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Учсть, что ШИМ характеризуется коэффициентом заполнения  $D$  (величина обратная  $D$  называется скважностью):

$$D = \frac{t}{T}$$



В выражении  $t$  - время заполнения в границе периода  $T$ .  
Действующее напряжение (или среднее)  $U_{CP}$  сигнала прямо пропорционально коэффициенту заполнения и амплитудному значению  $U_{AM}$

$$U_{CP}=D*U_{AM}.$$

Если такой сигнал подать на фильтр низких частот, то получится аналоговый сигнал, величина которого пропорциональна величине заполнения. Это простой способ получения аналогового сигнала, простейший ЦАП.

Вариант программы:

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdint.h>

volatile uint8_t pwm_value=0,dn_count=0;

ISR (TIMER1_COMPA_vect)
{
 TCNT1=0;

 if(dn_count) //плавно меняем яркость диода, по шагу за раз
 pwm_value--;
 else
 pwm_value++;

 if(pwm_value==0) //проверка границ, переключение
 //разгорание/затухание

 dn_count=0;
 if(pwm_value==0xFF)
 dn_count=1;

 OCR0A=pwm_value; //устанавливаем новый коэфф. заполнения
}

void main(void)
{
 DDRD=0xFF; //настройка порта на выход
 PORTD=0;
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*OCR1A=0x7A12; //конст., определяющая частоту прерываний  
TIMSK1=(1<<OCIE1A); //разрешаем прерыв. по совпадению канала A*



```
TCCR1B=(1<<CS11)|(1<<CS10); //запускаем таймер 1

TCCR0A=(1<<COM0A1)|(1<<WGM01)|(1<<WGM00); //таймер 0 будет
//генерировать ШИМ
OCR0A=128; //начальное значение ШИМ
TCCR0B=(1<<CS00); //запускаем таймер 0

sei(); //разрешаем прерывания

while(1); //далее процесс идет на прерываниях и
//аппаратном ШИМе
}
```

С этой программе используется макрос *stdint.h*. В нем объявлены некоторые типы данных с явно выраженной разрядностью:

```
uint8_t — unsigned 8-bit integer type;
uint16_t — unsigned 16-bit integer type;
uint32_t — unsigned 32-bit integer type;
int8_t — signed 8-bit integer type.
```

Эти описания повышают читаемость кода, гарантируют переносимость кода для других микроконтроллеров, иногда сокращают написание при объявлении переменных. Но применение этого макрорасширения не обязательно.

Модификатор *volatile* означает, что компилятору запрещается оптимизировать данную переменную. Это делает работу компилятора предсказуемой. Для предотвращения работы оптимизатора по отношению ко всей программе можно поставить уровень оптимизации `-Os` в свойствах проекта.

#### **С отчете представить:**

- В Схему подключения светодиода (если схема не совпадает с предыдущими схемами);
- В Расчетные соотношения для периода 1 сек;
- В Откорректированный текст программы с комментариями.
- В Расчеты для определения времени полного загорания и погасания.

#### **Вопросы для самоконтроля**

- В Какие адреса занимают порты ввода-вывода управления прерываниями?
- В Назовите основные источники прерываний для МКК.
- В Какие прерывания имеются для счетчиков в МКК ATtiny2313?
- В Назначение маски прерываний счетчиков?
- В Какими командами разрешаются и запрещаются прерывания?
- В С какой целью запрещаются прерывания?



- В Что составляет контекст процесса?
- В Как сохранить контекст? С какой целью его сохраняют?
- В Предложите алгоритм формирования длительной задержки с помощью таймеров?
10. Почему таймеры самостоятельно не могут организовать минутной за-держки?
- и Какое макрорасширение используется для работы с прерываниями на языке C?
12. Как написать обработчик на языке C?
- 8 Что представляет собой реакция микроконтроллера на сигнал преры-вания?
14. Каким образом процессор завершает выполнение обработчика? Что при этом выполняется?
- 19 Каким образом с помощью таймера организовать задержку большой длительности (например, минуты)?
- 20 Какие настройки выполняются для организации прерываний?
- 21 Какую роль выполняют флаги прерываний?
- 22 Как определяется приоритет прерываний микроконтроллера?
- 23 Как можно уточнить зарезервированные имена обработчиков;
- 24 Что нужно сделать, чтобы обработчик не выполнялся?
- 25 Сколько ячеек памяти занимает таблица векторов?
22. Как долго существует флаг прерывания?
- В Какова реакция контроллера на несколько прерываний одного объекта до их обработки?
- В При каких условиях могут образоваться вложенные прерывания?
- В Как можно организовать программные прерывания?
- В С какой целью предусмотрен захват таймера?
27. Как осуществляется инициализация таблицы векторов прерываний счетчиков?
28. Настройку каких регистров необходимо произвести для разрешения прерываний от таймер/счетчиков?
29. Можно ли отменить очередь прерываний?
30. Исходя из чего определяется очередность обработки прерываний при возникновении нескольких прерываний одновременно?
31. Какие прерывания могут быть утрачены после разрешения появления?
32. Какой регистр предназначен для глобального разрешения/запрещения прерывания?



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

33. Какие регистры используются для управления прерываниями счетчиков?
34. Какие регистры используются для управления внешними прерываниями?
35. Чем различаются между собой внешние прерывания?

МУ к Производственной(проектно-технологической) практике

## Общие положения

### Цели производственной(проектно-технологической) практики

Целями практики являются систематизация теоретических знаний и совершенствование профессиональных умений и навыков в области профессиональной деятельности, а также получение необходимой информации для качественного выполнения выпускной квалификационной работы по выбранной теме.

Производственная (проектно-технологическая) практика студентов является обязательной составной частью ООП и представляет собой вид учебных занятий, непосредственно ориентированных на профессионально-практическую подготовку обучающихся.

Практика проводится в соответствии с учебным планом и графиком учебного процесса.

Задолженность по практике является академической задолженностью.

Практика может проходить на базе кафедр и лабораторий ТУ или в сторонних организациях. Последнее возможно, если организации располагают необходимой материальной базой и квалифицированными специалистами.

Прохождение практики в сторонних организациях осуществляется на основании двустороннего договора ТУ УГМК со сторонней организацией о приеме студентов ТУ УГМК на неоплачиваемую практику.

Для руководства практикой приказом директора ТУ назначается руководитель практики от университета.

В случае прохождения практики в сторонней организации эта организация также назначает руководителя практики – от сторонней организации.

Места прохождения практик определяются для каждого студента в соответствии с целями и задачами практики. При этом, как правило, должны выбираться предприятия, учреждения и организации связи. Также практика

5

может проходить в организациях, которые занимаются другими видами деятельности, однако при этом организация должна содержать развитую внутреннюю телекоммуникационную сеть.

В общем случае место прохождения практики должно:

- соответствовать направлению подготовки и виду практики;
- иметь необходимую структуру и сферы деятельности, предусмотренные программой практики;
- располагать квалифицированными кадрами для руководства практикой.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Место прохождения практики должно соответствовать теме выпускной квалификационной работы (далее ВКР). Оно определяется с учетом мнения руководителя ВКР каждого конкретного студента.

Для прохождения практики студент должен написать заявление с просьбой направить его на практику в конкретную организацию и представить гарантийное письмо от этой организации о возможности прохождения в ней практики. Разрешение на индивидуальное прохождение практики утверждается приказом директора ТУ.

Производственная (проектно-технологическая) практика проходит в период между зимней и летней сессиями.

Студенты заочной формы обучения, работающие по профилю подготовки, проходят практику по месту работы по согласованию с администрацией предприятия. Студенты, не работающие по направлению подготовки, проходят практику по месту жительства на предприятиях (организациях), профиль работы которых соответствует профилю подготовки в ТУ, и с которыми у ТУ есть заключенные договоры о прохождении на них практики студентами ТУ УГМК.

Выполненные задания по практике студент привозит в ТУ УГМК, где в установленном порядке проходит защита итогов практики.

Студентам, работающим не по профилю подготовки или не работающим по уважительной причине (декретный отпуск, сокращение штатов, служба супруга в ВС РФ и т.п.), место практики подбирает заведующий выпускающей кафедрой. Он согласовывает со студентом место прохождения практики на одном из предприятий, с которыми у ТУ имеются соответствующие договоры и которое находится ближе к месту его проживания. **Права и обязанности студентов при прохождении практики**

#### **В обязанности студента входит:**

- присутствие на организационном собрании для ознакомления с задачами практики, знакомства с руководителем практики от ТУ и получения индивидуальных заданий;
- участие в инструктаже с руководителем практики от сторонней организации по технике безопасности, правилам внутреннего трудового распорядка и для получения пропуска в данную организацию при наличии пропускной системы;
- подчинение действующим в сторонней организации правилам внутреннего трудового распорядка и выполнение указаний руководителя практики от организации;
- осуществление работы в соответствии с программой практики, календарным планом и индивидуальным заданием, определенным руководителем практики от ТУ УГМК;
- ведение дневника, в котором отражается вся практическая деятельность согласно программе практики и календарному плану;
- консультации с руководителем практики при составлении отчета, выполнении индивидуального задания, подборе материалов для ВКР на преддипломной практике;
- представление отчета и дневника практики:
  - руководителю практики от сторонней организации – за 3 дня до окончания практики для проверки и составления отзыва,



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- руководителю практики от ТУ УГМК – в недельный срок после окончания практики для проверки и выставления предварительной оценки,

7

- руководителю ВКР – в трехдневный срок после окончания практики для проверки и выставления рекомендуемой оценки.

## Студент имеет право:

- предлагать свое место прохождения практики на согласование заведующему выпускающей кафедрой;
- требовать предоставления условий для выполнения заданий на практику на месте прохождения практики;
- требовать выполнения условий для безопасного прохождения практики.

Продолжительность рабочего дня студентов при прохождении практики в организациях составляет для студентов в возрасте от 16 до 18 лет не более 36 часов в неделю (ст.92 ТК РФ), в возрасте от 18 лет и старше не более 40 часов в неделю (ст.91 ТК РФ). С момента зачисления студентов в период практики в качестве практикантов на рабочие места на них распространяются правила охраны труда и правила внутреннего распорядка, действующие в организации.

## Подготовка к практике

Перед убытием на практику с группой проводится собрание, на котором студентам доводятся сроки практики, последовательность её прохождения и порядок отчетности по ней. Проводит собрание руководитель практики от института. На нём также присутствуют декан и заведующий выпускной кафедры.

На собрании производится оформление первой части дневника практики.

В частности, заполняются:

- титульный лист (в соответствии с приложением 1);
- направление на практику;
- план практики;
- индивидуальное задание.

После оформления дневника в разделе «Направление на практику» ставится подпись декана, которая утверждается печатью деканата.

Типовой план проведения практики должен содержать следующие пункты:

1. Прохождение инструктажа по правилам и мерам безопасности.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

2. Изучение требований безопасности на рабочем месте.
3. Изучение структуры организации и должностных обязанностей, соответствующих должности, на которой проходит практика.
4. Изучение правил организации рабочего места.
5. Изучение назначения, состава, характеристик и правил эксплуатации оборудования (специального программного обеспечения), эксплуатируемого должностным лицом.
6. Рассмотрение вопросов проведения технического обслуживания оборудования, поиска неисправностей и методов их устранения, подготовки заявок на ремонт имеющегося и закупку нового оборудования.
7. Выполнение обязанностей, соответствующих должности, на которой проходит практика.
8. Оформление дневника и отчёта по производственной (проектнотехнологической) практике.
9. Отчёт о прохождении практики на предприятии и получение отзыва от руководителя.

Все пункты плана практики устанавливаются руководителем практики. Данные пункты являются основными, но могут быть дополнены или изменены, исходя из особенностей того места, в котором будет проходить практика.

**Типовой вариант индивидуального задания** для прохождения практики.

1. Пройти инструктаж по правилам и мерам безопасности, установленным в организации.
2. Изучить требования безопасности на рабочем месте.
3. Рассмотреть штатную структуру организации. Уяснить должность, на которой будет проходить практика. Изучить должностные обязанности.
4. Изучить правила организации рабочего места.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

5. Изучить назначение, состав, характеристики и правила эксплуатации оборудования (специального программного обеспечения), эксплуатируемого должностным лицом.

6. Изучить реализуемые на предприятии правила проведения технического обслуживания оборудования, поиска неисправностей и методов

10

их устранения, подготовки заявок на ремонт имеющегося и закупку нового оборудования.

7. Выполнять обязанности должностного лица. Получить умения и навыки выполнения обязанностей, соответствующих должности, на которой проходит практика.

8. Оформить дневник и отчёт по практике.

9. Отчитаться о прохождении практики перед руководителем от предприятия и получить от него отзыв о работе.

## Прохождение практики

При прохождении практики студент должен выполнить все намеченные планом практики мероприятия и полностью выполнить индивидуальное задание на практику.

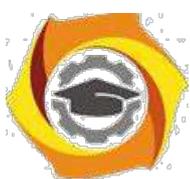
В процессе практики должны быть в окончательном виде оформлены отчётные документы по практике. Это дневник и отчёт по практике.

В дневнике по практике студенту необходимо заполнить раздел «Отчёт о прохождении практики».

После этого, в последний день практики, когда будут подводиться итоги практики на предприятии, студенту необходимо представить дневник руководителю практики от предприятия. В дневнике практики руководитель практики от предприятия должен написать отзыв о работе студента во время практики. Отзыв пишется по приведённому в дневнике образцу.

В отзыве руководителя обязательно должна присутствовать:

- ФИО студента;
- период прохождения практики;
- должность, в которой проходил практику студент (одна из штатных должностей организации, непосредственно связанная с направлением и профилем подготовки студента);



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

- характеристика работы студента;
- оценка студенту за прохождения практики  
выставляется по «двухбальной системе» «зачтено»/ «не зачтено»;
- подпись руководителя практики; - печать предприятия, подтверждающая подлинность подписи руководителя практики от предприятия.

По убытию из предприятия студент в обязательном порядке должен получить отметку у сотрудника отдела кадров об убытии с предприятия.

**Состав отчёта по практике.**

Отчёт по практике формируется в следующем порядке:

1. Титульный лист.
2. Содержание.
3. Текст пояснительной записки.

В отчёте обязательно должно присутствовать описание технологии передачи и обработки данных, которое используется в оборудовании компании.

Также должны быть рассмотрены альтернативные технологии.

Отчет должен содержать следующую информацию: общее описание организации (вид деятельности, краткая история), должностные обязанности, требования безопасности на рабочем месте, перечень используемого оборудования и аппаратуры, используемой практикантом при выполнении обязанностей должностного лица. После чего должны быть раскрыты вопросы: предназначение и состав оборудования, его технические характеристики, правила эксплуатации и технического обслуживания.

4. Список сокращений.
5. Список используемой литературы (документация на оборудование или внутренние инструкции компании по правилам его эксплуатации).

Общий объём отчета по преддипломной практике должен быть не менее 15 листов, при этом должен раскрывать содержание по всем выше приведённым вопросам эксплуатации оборудования и его характеристик.

### 3.1 Требования к оформлению текста пояснительной записки

#### 3.1.1 Общие требования

Текст должен быть набран на персональном компьютере (ПК), распечатан на бумаге формата А4 на одной стороне листа. Текст, включая все заголовки, печатается через полтора интервала. Используется шрифт «Times New Roman», размер шрифта – 14 пт. Исключением является заголовок раздела, он оформляется размером шрифта – 16 пт.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Расстояние от верхней или нижней строки текста до верхней или нижней рамки формы должно быть не менее 5 мм. Абзацы в тексте начинают отступом, равным 1,25 см (пяти знакам). При оформлении работы необходимо соблюдать следующие размеры полей на листе:

- правое 10 мм;
- левое 30 мм;
- верхнее 20 мм;
- нижнее 30 мм.

## 3.1.2 Оформление формул

В формулах в качестве символов следует применять обозначения, установленные соответствующими государственными стандартами. Желательно для набора формул использовать встроенный в текстовый редактор «Word» редактор формул «Equations» или отдельную программу, встраиваемую в текстовый редактор, например, «MathType». В формулах буквы русского и греческого алфавитов и цифры пишутся прямым шрифтом, а буквы латинского алфавита – курсивом. Размер основного шрифта в формулах должен быть, как и во всем тексте пояснительной записки, равен 14. Пояснения символов и числовых коэффициентов в формулах, если они не пояснены ранее, должны быть приведены непосредственно под формулой. Пояснение каждого символа даётся с новой строки в той последовательности, в которой символы приведены в

формуле. Первая строка пояснения начинается со слова "где" без двоеточия после него.

Пример 1. Плотность образца,  $\rho$  кг/м<sup>3</sup>, вычисляются по формуле:

$$\rho = \frac{m}{V}, \quad (1)$$

где  $m$  – масса образца, кг,

$V$  – объем образца, м<sup>3</sup>.

Формулы, следующие одна за другой и не разделенные текстом, разделяют запятой.

Выравнивание нумерации формул должно производиться по правому краю страницы.

Пример 2. Реакция фильтра нижних частот соответственно на положительный и отрицательный скачки напряжения на его входе описывается соотношениями:

$$U_{\text{ВЫХ}}(t) = U_{\text{ВХ}}(1 - e^{-t/RC}), \quad (2)$$

$$U_{\text{ВЫХ}}(t) = U_{\text{ВХ}} e^{-t/RC}, \quad (3)$$



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

где  $t$  – время, с,

$U_{\text{вх}}$  – изменение (скачок) напряжения на входе фильтра, В,

$R$  – сопротивление, Ом,  $C$   
– емкость, Ф.

Переносить формулы на следующую строку допускается только на знаках выполняемых операций, причем знак в начале следующей строки повторяют. При переносе формулы на знаке умножения применяют знак умножения в виде "х".

Формулы нумеруются сквозной нумерацией арабскими цифрами, которые записываются на уровне формулы справа в конце строки в круглых скобках. Одну формулу (единственную) тоже обозначают – (1).

Ссылки в тексте на формулы дают по их порядковым номерам, указываемым в скобках, например, в формуле (1).

Допускается нумерация формул в пределах раздела. В этом случае номер формулы состоит из номера раздела и порядкового номера формулы, разделенных точкой, например, (1.1). Если применяется нумерация с учётом номера раздела, то желательно такой же подход применять также к рисункам и таблицам. Это обеспечит единый вид представления информации.

### 3.1.3 Оформление иллюстраций

Все иллюстрации (чертежи, графики, диаграммы, фотографии), включаемые в текстовый материал записки, именуется рисунками.

Иллюстрации нумеруются арабскими цифрами сквозной нумерацией.

Если рисунок один, то он обозначается так: Рисунок 1.

Допускается нумеровать иллюстрации в пределах раздела. В этом случае номер иллюстрации состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой, например, Рисунок 1.1.

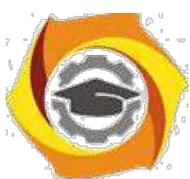
При ссылках на иллюстрации следует писать: «...в соответствии с рисунком 2» при их сквозной нумерации и «... в соответствии с рисунком 1.2» при нумерации в пределах раздела.

Иллюстрации могут иметь наименование и пояснительные данные (подрисуночный текст). Точка после подрисуночного текста не ставится. Слово «Рисунок» и наименование помещают после рисунка и пояснительных данных (если они есть), выравнивают по центру и располагают следующим образом:



Рисунок 1.1 – Общий вид коммутатора D-link DGS-1510-28

Допускается размещение рисунка в альбомном расположении страницы.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

В этом случае подрисовочная надпись должна быть ориентирована таким

16

образом, чтобы читалась при повороте раскрытой книги ПЗ по часовой стрелке на 90 градусов.

### 3.1.4 Оформление таблиц

Таблицу, в зависимости от ее размера, помещают под текстом, в котором впервые дана ссылка на нее, или на следующей странице, а, при необходимости, в приложении к документу.

Начало и конец таблицы отделяются от текста одним пробелом.

Цифровой материал для лучшей наглядности и удобства сравнения показателей оформляют в виде таблиц. Название таблицы должно отражать её содержание, быть точным, кратким.

Название помещают над таблицей (выравнивание по левому краю), начинают с заглавной буквы, не подчеркивают.

Таблицы необходимо нумеровать арабскими цифрами сквозной нумерацией. Если таблица одна, то она должна быть обозначена "Таблица 3".

Допускается нумеровать таблицы в пределах раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы, разделенных точкой. В тексте пояснительной записки на все таблицы должны быть приведены ссылки, при ссылке следует писать слово "таблица 3.1" с указанием её номера. Высота строк таблицы должна быть не менее 8 мм.

Таблицу, в зависимости от её размера, помещают непосредственно после первой ссылки на неё или (при отсутствии места) сразу на следующей странице. Например:

Таблица 3.1 – Сравнительная таблица стандартов беспроводной связи

| Технология | Стандарт | Использование | Пропускная способность | Радиус действия | Частоты |
|------------|----------|---------------|------------------------|-----------------|---------|
| 1          | 2        | 3             | 4                      | 5               | 6       |
| Wi-Fi      | 802.11a  | WLAN          | до 54 Мбит/с           | до 300 метров   | 5,0 ГГц |

17

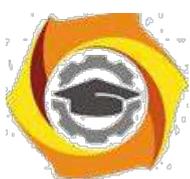
|       |         |      |              |               |         |
|-------|---------|------|--------------|---------------|---------|
| Wi-Fi | 802.11b | WLAN | до 11 Мбит/с | до 300 метров | 2,4 ГГц |
|-------|---------|------|--------------|---------------|---------|

Продолжение таблицы 3.1

| 1     | 2       | 3    | 4            | 5             | 6       |
|-------|---------|------|--------------|---------------|---------|
| Wi-Fi | 802.11g | WLAN | до 54 Мбит/с | до 300 метров | 2,4 ГГц |

Графу "номер по порядку" в таблицу включать не допускается. (ГОСТ 2.105-95).

Допускается помещать таблицу вдоль длинной стороны листа пояснительной записки (т.е. вертикально).



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Таблицу с большим количеством строк допускается переносить на другой лист, при этом в первой части таблицы нижнюю горизонтальную линию, ограничивающую таблицу, не проводят. При переносе продолжения таблицы на другую сторону листа заголовок помещают только над её первой частью.

Слово "Таблица" указывают один раз слева над первой частью таблицы; над другими её частями, если они переносятся на следующую страницу, пишут слова "Продолжение таблицы" с указанием номера (обозначения) таблицы, как показано выше.

Если повторяющийся в графе таблицы текст состоит только из одного слова, его допускается заменять кавычками; если он состоит из двух и более слов, при первом повторении его заменяют словами "То же", а далее кавычками. Если предыдущая фраза является частью последующей, то допускается заменить её словами "То же" и добавить дополнительные сведения.

При наличии в таблице горизонтальных линий текст необходимо повторять.

Не допускается заменять кавычками повторяющиеся в таблице цифры, математические знаки, знаки процента и номера, типоразмеров изделий, обозначения нормативных документов.

Если цифровые или иные данные в какой-либо строке таблицы не приводятся, то в ней ставят прочерк (тире).

При указании в таблице последовательных интервалов чисел, охватывающих все числа ряда, их следует записывать "От...до...включ.":

В интервале, охватывающем числа ряда, между крайними числами ряда в таблице допускается ставить тире.

Цифры в графах таблиц должны проставляться так, чтобы разряды чисел во всей графе были расположены один под другим, если они относятся к одному показателю. При этом количество десятичных знаков для всех значений должно быть одинаково.

При необходимости пояснения отдельных данных, приведенных в таблице, эти данные следует обозначать надстрочными знаками сноски.

Знак сноски ставят непосредственно после того слова, числа или символа, к которому даётся пояснение, и такой же знак ставят перед текстом пояснения к этой сноске.

Знак сноски выполняют арабскими цифрами со скобкой и помещают на уровне верхнего обреза шрифта. Нумерация сносок – отдельная для каждой таблицы.

В случае необходимости допускается в таблицах уменьшать размер шрифта до 12 или 10 пунктов, а также сокращать межстрочный интервал с полуторного до одинарного.

Допускается вместо цифр выполнять сноски звёздочками. При этом не рекомендуется применять более четырех звёздочек. Сноски располагают в конце таблицы с абзацного отступа.

## Отчетность по практике

Зачёт по преддипломной практике производится в СКФ МУТУСИ. По прибытии в институт, в назначенное для проведения зачёта время, студент обязан представить руководителю практики от института оформленные по установленным требованиям дневник и отчет по преддипломной практике.

К зачёту допускаются только те студенты, которые представили правильно оформленные дневник и отчёт.

Дневник по практике считается оформленным правильно и принимается, если:

- все поля дневника содержат информацию, в достаточной степени раскрывающую рассматриваемые вопросы;
- записи выполнены читаемым почерком, без серьёзных ошибок, допускаются незначительные поправки;



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- все подписи представителей организации и печати организации в наличии, читаемы и соответствуют месту проведения практики, указанному в приказе о прохождении практики.

Отчёт по преддипломной практике считается оформленным правильно и принимается, если:

- содержание отчёта позволяет судить о выполняемых студентом действиях при проведении практики;
- отчёт содержит анализ научно-технических источников и эксплуатационной документации, характеризующий различные технологии передачи и обработки данных и их сравнение;
- отчёт содержит актуальную информацию и оригинальные обобщения и выводы;
- оформление соответствует установленным требованиям.

Оценка за преддипломную практику определяется по уровню усвоения студентом всех компетенций. Определение знаний, умений и навыков производится путём получения ответов на задаваемые контрольные вопросы, а, при необходимости, и по выполнению практических задач. Вопросы и задачи выбираются, исходя из специфики места и выполняемых обязанностей студентом при прохождении практики.

МУ к Производственной(эксплуатационной) практике

## Общие положения

Целями производственной практики являются систематизация теоретических знаний, полученных в процессе обучения, приобретение и совершенствование профессиональных умений и навыков в области профессиональной деятельности.

Производственная практика студентов является обязательной составной частью ООП и представляет собой вид учебных занятий, непосредственно ориентированных на профессионально-практическую подготовку обучающихся.

Практика проводится в соответствии с учебным планом и графиком учебного процесса.

Задолженность по практике является академической задолженностью.

Практика может проходить на базе кафедр и лабораторий ТУ или в сторонних организациях. Последнее возможно, если организации располагают необходимой материальной базой и квалифицированными специалистами.

Прохождение практики в сторонних организациях осуществляется на основании двустороннего договора ТУ УГМК со сторонней организацией о приеме студентов ТУ УГМК на неоплачиваемую практику.

Для руководства практикой приказом директора ТУ назначается руководитель практики от университета.

В случае прохождения практики в сторонней организации эта организация также назначает руководителя практики – от сторонней организации.

Места прохождения практик определяются для каждого студента в соответствии с целями и задачами практики. При этом, как правило, должны выбираться предприятия, учреждения и



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

организации связи. Также практика может проходить в организациях, которые занимаются другими видами деятельности, однако, при этом организация должна иметь развитую внутреннюю телекоммуникационную сеть.

В общем случае место прохождения практики должно:

- соответствовать направлению подготовки и виду практики;
- иметь необходимую структуру и сферы деятельности, предусмотренные программой практики;
- располагать квалифицированными кадрами для руководства практикой.

## **Права и обязанности студентов при прохождении практики**

### **В обязанности студента входит:**

- участие в инструктаже с руководителем практики от сторонней организации по технике безопасности, правилам внутреннего трудового распорядка и для получения пропуска в данную организацию при наличии пропускной системы;
- подчинение действующим в сторонней организации правилам внутреннего трудового распорядка и выполнение указаний руководителя практики от организации;
- осуществление работы в соответствии с программой практики, календарным планом и индивидуальным заданием, определенным руководителем практики от ТУ УГМК;
- ведение дневника, в котором отражается вся практическая деятельность согласно программе практики и календарному плану;
- консультации с руководителем практики при составлении отчета, выполнении индивидуального задания, подборе материалов для ВКР; – представление отчета и дневника практики:
  - руководителю практики от сторонней организации – за 3 дня до окончания практики для проверки и составления отзыва,
  - руководителю практики от ТУ УГМК – в недельный срок после окончания практики для проверки и выставления предварительной оценки,
  - руководителю ВКР – в трехдневный срок после окончания практики для проверки и выставления рекомендуемой оценки.

### **Студент имеет право:**

- предлагать свое место прохождения практики на согласование заведующему выпускающей кафедрой;
- требовать предоставления условий для выполнения заданий на практику на месте прохождения практики;
- требовать выполнения условий для безопасного прохождения практики.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Продолжительность рабочего дня студентов при прохождении практики в организациях составляет для студентов в возрасте от 16 до 18 лет не более 36 часов в неделю (ст.92 ТК РФ), в возрасте от 18 лет и старше не более 40 часов в неделю (ст.91 ТК РФ). С момента зачисления студентов в период практики в качестве практикантов на рабочие места на них распространяются правила охраны труда и правила внутреннего распорядка, действующие в организации.

## Подготовка к практике

Перед убытием на практику с группой проводится собрание, на котором студентам доводятся сроки практики, последовательность её прохождения и порядок отчётности по ней. Проводит собрание руководитель практики от института. На нём также присутствуют декан и заведующий выпускной кафедры.

На собрании производится оформление первой части дневника практики. В частности, заполняются:

- титульный лист (в соответствии с приложением 1);
- направление на практику;
- план практики;
- индивидуальное задание.

После оформления дневника в разделе «Направление на практику» ставится подпись декана, которая утверждается печатью деканата.

**Типовой план проведения практики** должен содержать следующие пункты:

1. Прохождение инструктажа по правилам и мерам безопасности.
2. Изучение требований безопасности на рабочем месте.
3. Изучение структуры организации и должностных обязанностей, соответствующих должности, на которой проходит практика.
4. Изучение правил организации рабочего места.
5. Изучение назначения, состава, характеристик и правил эксплуатации оборудования (специального программного обеспечения), эксплуатируемого должностным лицом.
6. Рассмотрение вопросов проведения технического обслуживания оборудования, поиска неисправностей и методов их устранения, подготовки заявок на ремонт имеющегося и закупку нового оборудования.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

7. Выполнение обязанностей, соответствующих должности, на которой проходит практика.
8. Оформление дневника и отчёта по производственной практике.
9. Отчёт о прохождении производственной практики на предприятии и получение отзыва от руководителя.

Все пункты плана практики устанавливаются руководителем практики. Данные пункты являются основными, но могут быть дополнены или изменены, исходя из особенностей того места, в котором будет проходить практика.

**Типовой вариант индивидуального задания** для прохождения производственной практики.

1. Пройти инструктаж по правилам и мерам безопасности, установленным в организации.
2. Изучить требования безопасности на рабочем месте.
3. Рассмотреть штатную структуру организации. Уяснить должность, на которой будет проходить практика. Изучить должностные обязанности.
4. Изучить правила организации рабочего места.
5. Изучить назначение, состав, характеристики и правила эксплуатации оборудования (специального программного обеспечения), эксплуатируемого должностным лицом.
6. Изучить реализуемые на предприятии правила проведения технического обслуживания оборудования, поиска неисправностей и методов их устранения, подготовки заявок на ремонт имеющегося и закупку нового оборудования.
7. Выполнять обязанности должностного лица. Получить умения и навыки выполнения обязанностей, соответствующих должности, на которой проходит практика.
8. Оформить дневник и отчёт по производственной практике.
9. Отчитаться о прохождении производственной практики перед руководителем от предприятия и получить от него отзыв о работе.

Прохождение практики



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

При прохождении практики студент должен выполнить все намеченные планом практики мероприятия и полностью выполнить индивидуальное задание на практику.

В процессе практики должны быть в окончательном виде оформлены отчётные документы по практике. Это дневник и отчёт по практике.

В дневнике по практике студенту необходимо заполнить раздел «Отчёт о прохождении практики».

После этого, в последний день практики, когда будут подводиться итоги практики на предприятии, студенту необходимо представить дневник руководителю практики от предприятия. В дневнике практики руководитель практики от предприятия должен написать отзыв о работе студента во время практики. Отзыв пишется по приведённому в дневнике образцу.

В отзыве руководителя обязательно должна присутствовать:

- ФИО студента;
- период прохождения практики;
- должность, в которой проходил практику студент (одна из штатных должностей организации, непосредственно связанная с направлением и профилем подготовки студента);
- характеристика работы студента;
- оценка студенту за прохождение практики (по стандартной четырёхбалльной системе);
- подпись руководителя практики;
- печать предприятия, подтверждающая подлинность подписи руководителя практики от предприятия.

По убытию из предприятия, студент в обязательном порядке должен получить отметку у сотрудника отдела кадров об убытии с предприятия.

### **Состав отчёта по производственной практике.**

Отчёт по практике формируется в следующем порядке:

1. Титульный лист.
2. Содержание.
3. Текст пояснительной записки.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Должен содержать следующую информацию: общее описание организации (вид деятельности, краткая история), должностные обязанности, требования безопасности на рабочем месте, перечень используемого оборудования и аппаратуры, используемой практикантом при выполнении обязанностей должностного лица. После чего должны быть раскрыты вопросы: предназначение и состав оборудования, его технические характеристики, правила эксплуатации и технического обслуживания.

4. Список сокращений.

5. Список используемой литературы (документация на оборудование или внутренние инструкции компании по правилам его эксплуатации).

Общий объём отчета по производственной практике должен быть не менее 15 листов, при этом отчет должен раскрывать содержание по всем выше приведённым вопросам эксплуатации оборудования и его характеристик.

## 3.1 Требования к оформлению текста пояснительной записки

### 3.1.1 Общие требования

Текст должен быть набран на персональном компьютере (ПК), распечатан на бумаге формата А4 на одной стороне листа. Текст, включая все заголовки, печатается через полтора интервала. Используется шрифт «Times New Roman», размер шрифта – 14 пт. Исключением является заголовок раздела, он оформляется размером шрифта – 16 пт.

Расстояние от верхней или нижней строки текста до верхней или нижней рамки формы должно быть не менее 5 мм. Абзацы в тексте начинают отступом, равным 1,25 см (пяти знакам).

При оформлении работы необходимо соблюдать следующие размеры полей на листе:

- правое 10 мм;
- левое 30 мм;
- верхнее 20 мм;
- нижнее 30 мм.

### 3.1.2 Оформление формул

В формулах в качестве символов следует применять обозначения, установленные соответствующими государственными стандартами. Желательно для набора формул использовать встроенный в текстовый редактор «Word» редактор формул «Equations» или отдельную программу, встраиваемую в текстовый редактор, например, «MathType».

В формулах буквы русского и греческого алфавитов и цифры пишутся прямым шрифтом, а буквы латинского алфавита – курсивом. Размер основного шрифта в формулах должен быть, как и во всем тексте пояснительной записки, равен 14.

Пояснения символов и числовых коэффициентов в формулах, если они не пояснены ранее, должны быть приведены непосредственно под формулой. Пояснение каждого символа даётся с новой строки в той последовательности, в которой символы приведены в формуле. Первая строка пояснения начинается со слова "где" без двоеточия после него.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Пример 1. Плотность образца,  $\rho$  кг/м<sup>3</sup>, вычисляют по формуле:

$$\rho = \frac{m}{V}, \quad (1)$$

где  $m$  – масса образца, кг,

$V$  – объем образца, м<sup>3</sup>.

Формулы, следующие одна за другой и не разделенные текстом, разделяют запятой. Выравнивание нумерации формул должно производиться по правому краю страницы.

Пример 2. Реакция фильтра нижних частот соответственно на положительный и отрицательный скачки напряжения на его входе описывается соотношениями:

$$U_{\text{ВЫХ}}(t) = U_{\text{ВХ}}(1 - e^{-t/RC}), \quad (2)$$

$$U_{\text{ВЫХ}}(t) = U_{\text{ВХ}} e^{-t/RC}, \quad (3)$$

где  $t$  – время, с,

$U_{\text{вх}}$  – изменение (скачок) напряжения на входе фильтра, В,

$R$  – сопротивление,

Ом,  $C$  – емкость, Ф.

Переносить формулы на следующую строку допускается только на знаках выполняемых операций, причем знак в начале следующей строки повторяют. При переносе формулы на знаке умножения применяют знак умножения в виде "x".

Формулы нумеруются сквозной нумерацией арабскими цифрами, которые записываются на уровне формулы справа в конце строки в круглых скобках.

Одну формулу (единственную) тоже обозначают – (1).

Ссылки в тексте на формулы дают по их порядковым номерам, указываемым в скобках, например, в формуле (1).

Допускается нумерация формул в пределах раздела. В этом случае номер формулы состоит из номера раздела и порядкового номера формулы, разделенных точкой, например, (1.1). Если применяется нумерация с учётом номера раздела, то желательно такой же подход применять также к рисункам и таблицам. Это обеспечит единый вид представления информации.

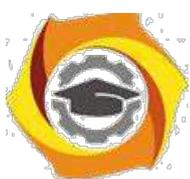
### 3.1.3 Оформление иллюстраций

Все иллюстрации (чертежи, графики, диаграммы, фотографии), включаемые в текстовый материал записки, именуется рисунками.

Иллюстрации нумеруются арабскими цифрами сквозной нумерацией. Если рисунок один, то он обозначается так: Рисунок 1.

Допускается нумеровать иллюстрации в пределах раздела. В этом случае номер иллюстрации состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой, например, Рисунок 1.1.

При ссылках на иллюстрации следует писать: «...в соответствии с рисунком 2» при их сквозной нумерации и «... в соответствии с рисунком



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

1.2» при нумерации в пределах раздела.

Иллюстрации могут иметь наименование и пояснительные данные (подрисующий текст). Точка после подрисующего текста не ставится. Слово «Рисунок» и наименование помещают после рисунка и пояснительных данных (если они есть), выравнивают по центру и располагают следующим образом:



Рисунок 1.1 – Общий вид коммутатора D-link DGS-1510-28

Допускается размещение рисунка в альбомном расположении страницы. В этом случае подрисующая надпись должна быть ориентирована таким образом, чтобы читалась при повороте раскрытой книги ПЗ по часовой стрелке на 90 градусов.

### 3.1.4 Оформление таблиц

Таблицу, в зависимости от ее размера, помещают под текстом, в котором впервые дана ссылка на нее, или на следующей странице, а при необходимости, в приложении к документу.

Начало и конец таблицы отделяются от текста одним пробелом.

Цифровой материал для лучшей наглядности и удобства сравнения показателей оформляют в виде таблиц. Название таблицы должно отражать её содержание, быть точным, кратким. Название помещают над таблицей (выравнивание по левому краю), начинают с заглавной буквы, не подчеркивают.

Таблицы необходимо нумеровать арабскими цифрами сквозной нумерацией. Если таблица одна, то она должна быть обозначена "Таблица 3".

Допускается нумеровать таблицы в пределах раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы, разделенных точкой. В тексте пояснительной записки на все таблицы должны быть приведены ссылки, при ссылке следует писать слово "таблица 3.1" с указанием её номера.

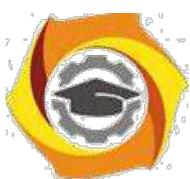
Высота строк таблицы должна быть не менее 8 мм.

Таблицу, в зависимости от её размера, помещают непосредственно после первой ссылки на неё или (при отсутствии места) сразу на следующей странице.

Например:

Таблица 3.1 – Сравнительная таблица стандартов беспроводной связи

| Технология | Стандарт | Использование | Пропускная способность | Радиус действия | Частоты |
|------------|----------|---------------|------------------------|-----------------|---------|
| 1          | 2        | 3             | 4                      | 5               | 6       |
| Wi-Fi      | 802.11a  | WLAN          | до 54 Мбит/с           | до 300 метров   | 5,0 ГГц |



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

|       |         |      |              |               |         |
|-------|---------|------|--------------|---------------|---------|
| Wi-Fi | 802.11b | WLAN | до 11 Мбит/с | до 300 метров | 2,4 ГГц |
|-------|---------|------|--------------|---------------|---------|

Продолжение таблицы 3.1

| 1     | 2       | 3    | 4            | 5             | 6       |
|-------|---------|------|--------------|---------------|---------|
| Wi-Fi | 802.11g | WLAN | до 54 Мбит/с | до 300 метров | 2,4 ГГц |

Графу "номер по порядку" в таблицу включать не допускается. (ГОСТ 2.105-95).

Допускается помещать таблицу вдоль длинной стороны листа пояснительной записки (т.е. вертикально).

Таблицу с большим количеством строк допускается переносить на другой лист, при этом в первой части таблицы нижнюю горизонтальную линию, ограничивающую таблицу, не проводят. При переносе продолжения таблицы на другую сторону листа заголовок помещают только над её первой частью.

Слово "Таблица" указывают один раз слева над первой частью таблицы; над другими её частями, если они переносятся на следующую страницу, пишут слова "Продолжение таблицы" с указанием номера (обозначения) таблицы, как показано выше.

Если повторяющийся в графе таблицы текст состоит только из одного слова, его допускается заменять кавычками; если он состоит из двух и более слов, при первом повторении его заменяют словами "То же", а далее кавычками. Если предыдущая фраза является частью последующей, то допускается заменить её словами "То же" и добавить дополнительные сведения. При наличии в таблице горизонтальных линий текст необходимо повторять.

Не допускается заменять кавычками повторяющиеся в таблице цифры, математические знаки, знаки процента и номера, типоразмеров изделий, обозначения нормативных документов.

Если цифровые или иные данные в какой-либо строке таблицы не приводятся, то в ней ставят прочерк (тире).

При указании в таблице последовательных интервалов чисел, охватывающих все числа ряда, их следует записывать "От...до...включ."

В интервале, охватывающем числа ряда, между крайними числами ряда в таблице допускается ставить тире.

Цифры в графах таблиц должны проставляться так, чтобы разряды чисел во всей графе были расположены один под другим, если они относятся к одному показателю. При этом количество десятичных знаков для всех значений должно быть одинаково.

При необходимости пояснения отдельных данных, приведенных в таблице, эти данные следует обозначать надстрочными знаками сноски.

Знак сноски ставят непосредственно после того слова, числа или символа, к которому даётся пояснение, и такой же знак ставят перед текстом пояснения к этой сноске.

Знак сноски выполняют арабскими цифрами со скобкой и помещают на уровне верхнего обреза шрифта. Нумерация сносок – отдельная для каждой таблицы.

В случае необходимости допускается в таблицах уменьшать размер шрифта до 12 или 10 пунктов, а также сокращать межстрочный интервал с полуторного до одинарного.

Допускается вместо цифр выполнять сноски звёздочками. При этом не рекомендуется применять более четырех звёздочек. Сноски располагают в конце таблицы с абзацного отступа.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

## Отчетность по практике

Зачёт по производственной практике производится в ТУ УГМК. По прибытии в институт, в назначенное для проведения зачёта время, студент обязан представить руководителю практики от института оформленные по установленным требованиям дневник и отчет по производственной практике.

К зачёту допускаются только те студенты, которые представили правильно оформленные дневник и отчёт.

Дневник по производственной практике считается оформленным правильно и принимается, если:

- все поля дневника содержат информацию, в достаточной степени раскрывающую рассматриваемые вопросы;
- записи выполнены читаемым почерком, без серьёзных ошибок, допускаются незначительные помарки;
- все подписи представителей организации и печати организации в наличии, читаемы и соответствуют месту проведения практики, указанному в приказе о прохождении практики.

Отчёт по производственной практике считается оформленным правильно и принимается, если:

- содержание отчёта позволяет судить о выполняемых студентом действиях при проведении практики;
- отчёт содержит анализ научно-технических источников и эксплуатационной документации, характеризующий различные технологии передачи и обработки данных и их сравнение;
- отчёт содержит актуальную информацию и оригинальные обобщения, и выводы;
- оформление соответствует установленным требованиям.

Оценка за производственную практику определяется по уровню усвоения студентом всех компетенций. Определение знаний умений и навыков производится путём получения ответов на задаваемые контрольные вопросы, а при необходимости, и по выполнению практических задач. Вопросы и задачи выбираются, исходя из специфики места и выполняемых обязанностей студентом при прохождении практики.

# Мультимедиа технологии и протоколы

**Методическое пособие для практических занятий**



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

**Практическое занятие 1. Аппаратное обеспечение**

мультимедиа

**Устройство и работа мультимедийного проектора**

**1 Цель работы**

10. Ознакомиться с технологией работы мультимедийных проекторов
11. Ознакомиться со способами подключения к компьютеру

**2 Пояснение к работе**

2.1 Краткие теоретические сведения

**Мультимедийный проектор**

**Мультимедийные проекторы** - универсальные приборы, которые дают нам возможность смотреть видеокассеты и DVD-диски, работать с любыми другими источниками изображения, например с телевидением, в том числе с кабельным и спутниковым. К проектору также можно подключить компьютер, документ-камеру или целую систему видеонаблюдения

Среди разработанных на сегодняшний день технологий проецирования цветного изображения на внешний экран можно выделить четыре основные, получившие наиболее широкое применение в коммерческих продуктах ведущих производителей и различающиеся в первую очередь типом элемента, используемого для формирования изображения:

|              |         |         |             |
|--------------|---------|---------|-------------|
| <b>CRT -</b> | Cathode | Ray     | Tube;       |
| <b>LCD -</b> | Liquid  | Crystal | Display;    |
| <b>DLP -</b> | Digital | Light   | Processing; |



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

**D-ILA** - Direct Drive Image Light Amplifier.  
**CRT-технология**

Мультимедийные проекторы на базе электронно-лучевых трубок (CRT) выпускаются в течение уже нескольких десятилетий. Но, несмотря на появление более современных технологий, по качеству воспроизведения изображения (разрешение, четкость, точность цветопередачи), уровню акустического шума и длительности непрерывной работы они до сих пор не имеют себе равных. Ни одна другая технология пока не обеспечивает столь же глубокий уровень черного и столь же широкий динамический диапазон яркости изображения, благодаря которым CRT-проекторы позволяют различать детали даже при демонстрации затемненных сцен.

Обладая несомненными достоинствами, особенно при демонстрации видео, CRT-проекторы имеют и ряд существенных недостатков, ограничивающих сферу их применения. При значительных габаритах и массе в несколько десятков килограмм они проигрывают современным портативным мультимедиа-проекторам в яркости.

**LCD-технология**

8. мультимедийных проекторах, выполненных по технологии LCD (Liquid Crystal Display), функции формирователя изображения выполняет LCD-матрица просветного типа. По принципу действия такие аппараты напоминают обычные диапроекторы (см. Устройство LCD-проектора) с той разницей, что проецируемое на внешний экран изображение формируется при прохождении излучаемого лампой светового потока не через слайд, а через жидкокристаллическую панель, состоящую из множества электрически управляемых элементов - пикселей. В зависимости от величины приложенного к каждому такому элементу переменного напряжения меняется его прозрачность, а, следовательно, и уровень освещенности участка экрана, на который проецируется данный пиксел.

LCD-технология позволила существенно удешевить проекционные аппараты,  
6

уменьшить их габариты и одновременно увеличить излучаемый ими световой поток. Она естественным образом адаптирована к воспроизведению видеосигналов от компьютерных источников, а также сохраненных в цифровом формате видеофайлов. LCD-проекторы просты в обращении и настройке и сохраняют свои параметры после транспортировки. Именно поэтому они широко применяются в бизнес-сфере для проведения презентаций и демонстрации шоу-программ.

**DLP-технология**



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Лежащая в основе любого DLP-проектора технология цифровой обработки света (DLP) базируется на разработках корпорации Texas Instruments, создавшей новый тип формирователя изображения - цифровое микрзеркальное устройство DMD (Digital Micromirror Device). DMD-формирователь представляет собой кремниевую пластину, на поверхности которой размещены сотни тысяч управляемых микрзеркал. Главное его преимущество по сравнению с формирователями иного типа заключается в высокой световой эффективности, обусловленной двумя факторами: более эффективным использованием рабочей поверхности формирователя (коэффициент использования - до 90%) и меньшим поглощением световой энергии работающими "на отражение" микрзеркалами, которые к тому же не требуют применения поляризаторов. В силу этих причин, а также относительно простого решения проблемы отвода тепла, DLP-технология позволяет создавать как мощные проекционные аппараты с большим световым потоком, так и сверхминиатюрные проекторы для мобильных пользователей. Именно в этих классах продуктов DLP-технология сегодня доминирует.

Современные DLP-проекторы строятся по схеме с одним, двумя и тремя DMD-кристаллами. Как и LCD-аппараты, они характеризуются собственным оптическим разрешением, определяемым числом микрзеркал в DMD-матрице, и наилучшим образом приспособлены для воспроизведения графической и видеоинформации, хранящейся в цифровом формате.

Используемый в них принцип формирования полутонов основывается на свойстве человеческого глаза усреднять визуальную информацию за короткий промежуток времени и требует применения сложных алгоритмов пересчета входных данных в управляющие микрзеркалами ШИМ-последовательности. Качество алгоритмов во многом определяет достигаемую точность цветопередачи.

### **D-ILA-технология**

Относительно недавно разработанная компанией Huges-JVC технология D-ILA (Direct Drive Image Light Amplifier) фактически является первым коммерческим воплощением так называемой технологии LCOS, представляющей, по мнению большинства экспертов, одно из наиболее перспективных направлений в области создания проекционного оборудования. Подобно LCD-технологии она базируется на свойствах жидких кристаллов, однако, вместо обычных просветных матриц на основе аморфного или поликристаллического кремния, предполагает использование

И качестве формирователей изображения приборов отражающего типа (см. Устройство D-ILA-проекторов). В матрице D-ILA светомодулирующий жидкокристаллический слой располагается поверх подложки из монокристаллического кремния, на которой фотолитографическим способом



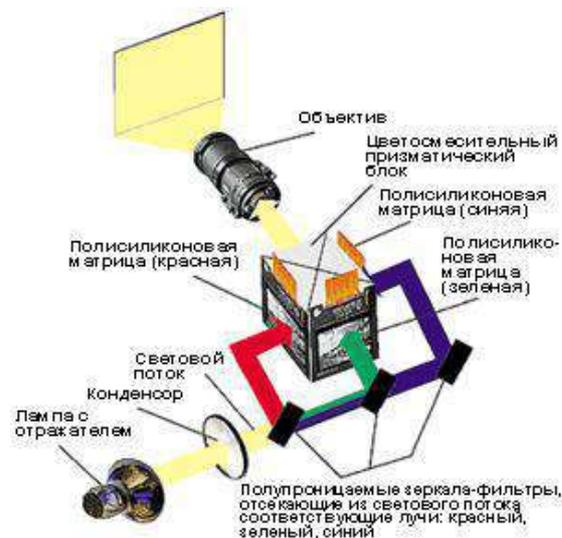
# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

сформированы управляющие пикселями электроды, одновременно выполняющие функции отражающих элементов. Почти вся схема управления матрицей размещается непосредственно в подложке, что обеспечивает данной технологии ряд существенных преимуществ по сравнению с LCD-панелями. Матрицы D-ILA проще в изготовлении и при меньших размерах могут иметь существенно более высокое разрешение. Эффективность использования площади кристалла в них достигает 93%, что практически исключает проявление сеточной структуры на экране. 7

Большинство выпущенных к настоящему времени D-ILA-проекторов базируются

на матрицах с разрешением SXGA (1365x1024 пикселей) и, обладая световым потоком в пределах от 1000 до 7000 ANSI-лм, характеризуются сравнительно большой массой и высокой ценой. Кроме того, существуют и матрицы повышенного разрешения QXGA (2048x1536 пикселей) размером 1.3 дюйма по диагонали. Последние обеспечивают полноценное воспроизведение видеосигналов стандарта HDTV (1080i).

## Устройство LCD-проектора



Современные LCD-проекторы выполняются на базе трех полисиликоновых жидкокристаллических матриц, размером, в основном, от 0.7 до 1.8 дюймов по диагонали. Структурная схема такого проектора представлена на рисунке.

Световое излучение лампы с помощью конденсора преобразуется в равномерный световой поток, из которого дихроичные зеркала-фильтры выделяют три цветные составляющие (красную, синюю и зеленую) и направляют их на соответствующие LCD-матрицы. Сформированные ими цветные изображения объединяются в цветосмесительном призматическом блоке в одно полноцветное, которое затем через объектив проецируется на внешний экран.

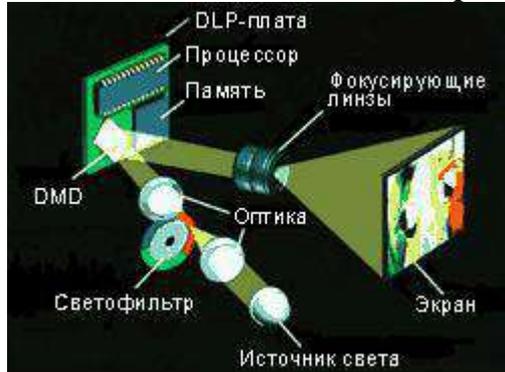
## Устройство DLP-проектора



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Современные DLP-проекторы строятся по схеме с одним, двумя и тремя DMD-матрицами.

### Оптическая схема одноматричного DLP-проектора

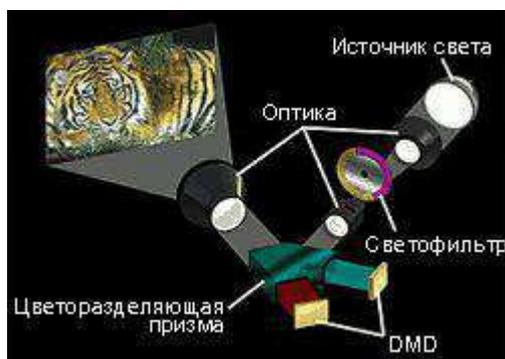


· одноматричном DLP-проекторе световой поток лампы пропускается через вращающийся фильтр с тремя секторами, окрашенными в цвета составляющих пространства RGB. В зависимости от угла поворота фильтра DMD-кристалл формирует на экране синюю, красную или зеленую картинку, которые последовательно сменяют одна другую за короткий интервал времени. Усредняя

отражаемый экраном световой поток, человеческий глаз воспринимает изображение как полноцветное.

По схеме с одним DMD-кристаллом в настоящее время строятся наиболее миниатюрные DLP-проекторы. Они применяются для проведения мобильных бизнес-презентаций, а также для демонстрации цветного видео. Следует, однако, учитывать, что в последнем случае световой поток проектора с четырехсекторным цветным фильтром оказывается ниже указанного в техническом паспорте, т. к. в этом режиме прозрачный сектор не работает, и эффективность использования светового потока лампы снижается.

### Оптическая схема двухматричного DLP-проектора

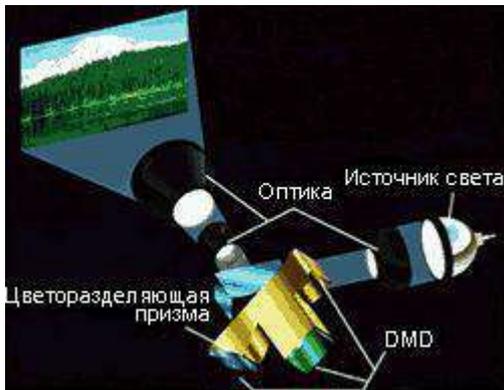


В двухматричных DLP-проекторах вращающийся цветной фильтр имеет два сектора пурпурного (смесь красного с синим) и желтого (смесь красного и зеленого) цветов. Дихроичные призмы разделяют световой поток на составляющие, при этом поток красного цвета в каждом случае направляется на одну из DMD-матриц. На вторую в

зависимости от положения фильтра направляется поток либо синего, либо зеленого цвета. Таким образом, двухматричные проекторы, в отличие от одноматричных, проецируют на экран картинку красного цвета постоянно, что позволяет компенсировать недостаточную интенсивность красной части спектра излучения некоторых ламп.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**



**Оптическая схема трехматричного DLP-проектора**

В трехматричных DLP-проекторах световой поток лампы с помощью дихроичных призм расщепляется на три составляющих (RGB), каждая из которых направляется на свою DMD-матрицу, формирующую картинку одного цвета. Объектив аппарата проецирует на экран одновременно три цветных картинки, формируя таким образом полноцветное

изображение.

Благодаря высокой эффективности использования светового излучения лампы, трехматричные DLP-проекторы, как правило, характеризуются повышенным световым потоком, достигающим у наиболее мощных аппаратов 18000 ANSI-лм.

**Порты входов и выходов мультимедиа проекторов**

- Порт HD 15 female (15 штырьковый трехрядный разъем ("мама"))
- DVI порт female ("мама")
- 5-BNC порт
- RCA порт для видео (желтый)
- S-видео порт
- IEEE-1394 (FireWire)
- SDI порт (соединитель BNC)
- USB порт
- RCA порт для звука (белый, красный / звук) (AUDIO IN)



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

- Мини-джек (порт входа звукового сигнала) (AUDIO IN, COMPUTER AUDIO IN)
- Мини-джек (порт выхода звукового сигнала) (MONITOR)
- D-Sub (9 pin) male ("папа")
- Мини DIN 8 порт (RS 232, Mouse, PS/2)

**Порт HD 15 female (15 штырьковый трехрядный разъем - "мама")**



Служит для подключения компьютера. Это - наиболее распространенный способ присоединения компьютеров и мониторов к видео проектору. RGB-вход, чаще всего, имеет указательную

надпись "computer" (см. рис.), поскольку, почти всегда источником RGB-сигнала является компьютер. RGB-выход, там где он существует, чаще всего помечается как "monitor". Он может быть использован для пересылки уже полученного сигнала на монитор или другое внешнее устройство, например еще один проектор. Иногда RGB-интерфейс именуют VGA или SVGA.

**DVI порт female ("мама")**



Интерфейс Цифрового Видео (Digital Visual Interface), является относительно новым для передачи графического сигнала. Он был разработан, как альтернатива RGB, для связи источников сигнала с отображающими устройствами. Этот интерфейс использует полностью цифровой стандарт передачи данных, благодаря которому можно избежать ухудшения качества, возникающего при аналогово-цифровых преобразованиях изображения.

**S-видео порт**





**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Буква S в названии этого порта, соответствующего сигнала и типа соединителя означает separate (раздельный), так как в этом сигнале информация о яркости и цвете передается по двум раздельным проводам. S-видео сигнал обеспечивает заметно лучшее качество изображения, по сравнению с композитным. S-видео называют также Y/C-video.

### **IEEE-1394 (FireWire)**



Это порт, в ближайшее время, может появиться на многих моделях проекторов, хотя сегодня он имеется пока только на некоторых профессиональных. IEEE-1394 (торговая марка компании Apple) используется, например, в цифровых видеокамерах mini-DV. Передача сигнала в проектор непосредственно в цифровом виде также будет обеспечивать очень высокое качество изображения.

### **USB порт**



Universal Serial Bus (универсальная шина для передачи данных) все чаще встречается на разных моделях проекторов. Служит пока, главным образом, для осуществления управления проектором при помощи компьютера, и для управления компьютером с пульта дистанционного управления проектором. В дальнейшем не исключено использование этого порта и для передачи изображения и звука.

### **RCA порт для видео (желтый)**





**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

· проекторах, особенно компактных, это прежде всего вход композитного видео сигнала. Это - стандартный видеопорт и он может быть использован для воспроизведения сигналов видеомагнитофонов, DVD-плееров и других стандартных видеоустройств. Чаще всего для этого порта используется соединитель типа RCA (известный еще как "тольпан", из за схожей с цветком формы соединителей старых выпусков). Выходы композитного видео сигнала присутствуют на всех, за редким исключением, видео устройствах магнитофонах, камерах, проигрывателях DVD и т.п.



**RCA порт для звука (белый, красный / звук) (AUDIO IN)** Это порт для подключения звука. Чаще всего он включается вместе с входами от видеоисточников. Согласно принятому стандарту белым цветом обозначается разъем левого, а красным правого канала

стерео системы.

· **Задание**

3.1 Рассмотрите мультимедийный проектор. Запишите определение мультимедийного проектора, укажите, сколько типов проецирования существует и какие; запишите с помощью каких портов подключаются мультимедийные проекторы.

3.2 Зарисуйте схему устройства одного из видов проектора.

3.3 Подключите мультимедийный проектор разными способами и настройте его. Убедитесь, что изображение проецируется на мультимедийный экран.

3.4 Зарисуйте схему подключения мультимедийного проектора к компьютеру  
10) указанием портов и соединительных портов.

**4 Работа в кабинете (лаборатории)**

4.1 Ознакомиться с теоретическим материалом по лабораторной работе.

4.2 Выполнить предложенные задания.

4.3 Продемонстрировать результаты выполнения предложенных заданий.



## 5 Контрольные вопросы

- 5.1 Что такое мультимедийный проектор? Какие устройства можно подключать к нему?
- 5.2 Какие технологии проецирования цветного изображения на внешний экран на сегодняшний день существуют?
- 5.3 Какие типы портов входов и выходов мультимедиа проекторов существуют?
- 5.4 Какой наиболее распространенный способ присоединения компьютеров и мониторов к мультимедийному проектору?
- 5.5 Какой тип порта используется в цифровых видеокамерах mini-DV?

## 6 Содержание отчета

- 6.1 название работы;
- 6.2 цель работы;
- 6.3 краткий конспект теоретической части;
- 6.4 ответы на контрольные вопросы;
- 6.5 выводы к работе.

## 7 Список литературы

и Гребенюк Е.И. Технические средства информатизации: Учебник для сред. проф. образования / Е.И.Гребенюк, Н.А. Гребенюк. – 2-е изд., стер. – М.: Издательский центр «Академия», 2005. – 272 с.

и <http://www.polymedia.ru>



Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»

Практическое занятие 2 Технологии работы с аудио, фото и  
видео информации

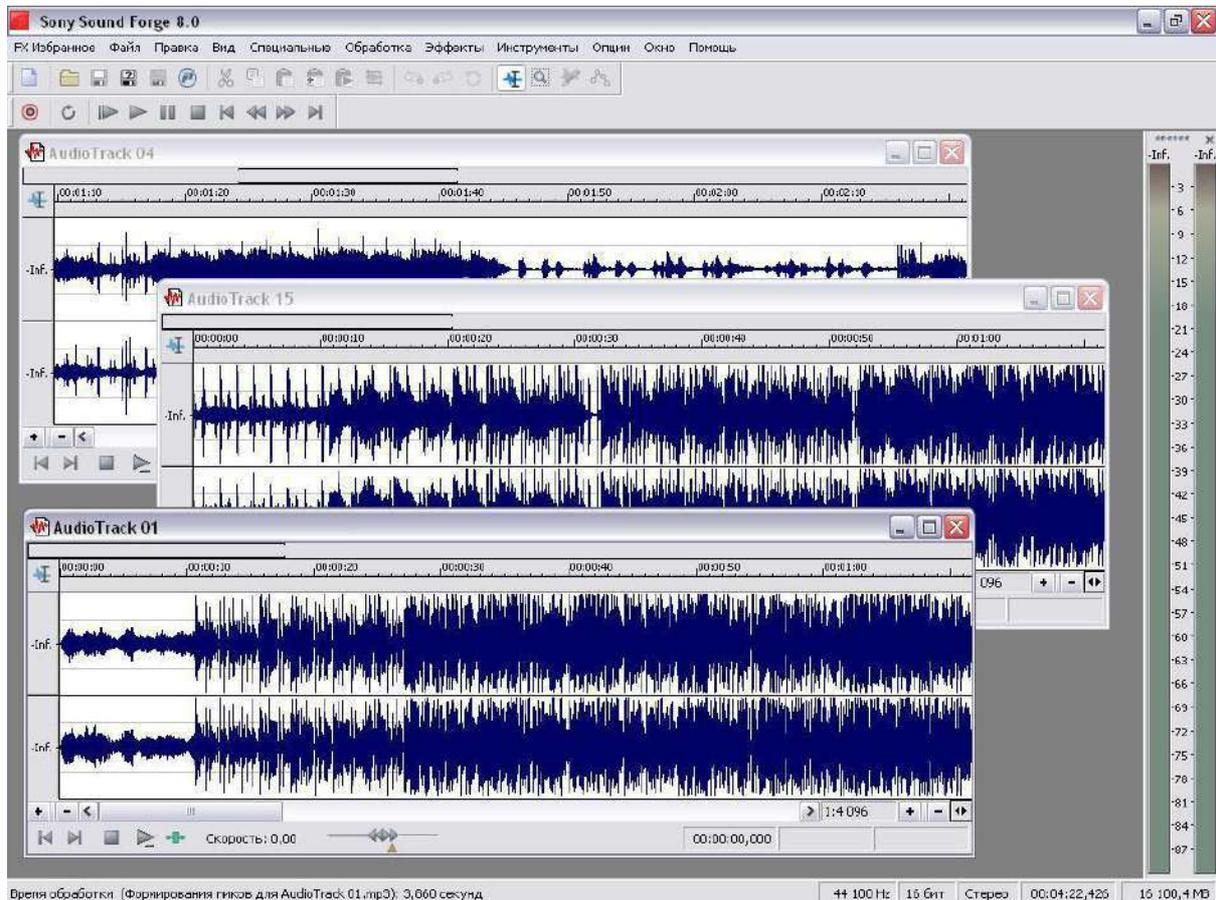
· Цель работы

1.1 Ознакомиться с программой Sound Forge 8.0.

1.2 Научиться записывать и обрабатывать звуковые файлы.

**2 Пояснение к работе**

2.1 Краткие теоретические сведения



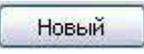
Sound Forge 8.0 фирмы Sonic Foundry, Inc - это одна из самых популярных и действительно полезных программ, предназначенных для обработки аудио на профессиональном уровне. С ее помощью можно обрабатывать аудиосигнал, изменяя его до неузнаваемости или же редактировать неудачно записанную партию какого-либо музыкального инструмента. Sound Forge 8.0 успешно



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

объединяет в себе практически полный набор современных звуковых эффектов и мощные средства редактирования звуков.

Для того чтобы записать звуковой фрагмент необходимо открыть новый файл, нажав на кнопку  или выполнив команду Файл/Новый. Контроль над записью, воспроизведением и перемоткой к началу и концу файла осуществляется управляющими кнопками, расположенными в правой верхней части экрана и напоминающими соответствующие кнопки бытового магнитофона. Для записи используем микрофон или любой другой источник сигнала, например, магнитофон или CD-проигрыватель, подключенный ко входу звуковой платы. Далее, нажимаем на кнопку записи  и видим перед собой вспомогательное меню. В этом меню выбирается формат

записываемого файла [  : Частоту, глубину, каналы Моно/Стерео], режим записи

(замена предыдущего дубля последующим, последовательная запись дублей), тут же находятся и индикаторы уровня входного сигнала. Итак, чтобы начать запись, достаточно нажать кнопку  и приступить непосредственно к звукоизвлечению. Для прекращения записи нажмите кнопку , появившуюся на месте кнопки . Затем закройте вспомогательное меню и приступайте к редактированию записанной фонограммы.

Сначала необходимо удалить заведомо лишние паузы в начале и в конце файла, появившиеся в результате задержки между моментом нажатия кнопки Запись/Остановить и началом/окончанием звукоизвлечения. Для этого нужно отметить подлежащую удалению область файла и выполнить команду Правка/Вырезать, либо нажать Delete на клавиатуре. Области файла отмечаются очень просто - точно так же, как вы делаете это в текстовом редакторе: ставите мышь в выбранное место на экране и двигаете ее вправо или влево, предварительно нажав левую кнопку, затем отпускаете кнопку. Отмеченная область окрашивается в черный цвет. Заметьте, что если вы работаете со стереофайлом и хотите отметить оба канала сразу, то курсор нужно ставить недалеко от горизонтальной линии, разделяющей стереоканалы. Если же вам нужно отметить область только на одном из каналов, то курсор ставьте ниже среднего уровня правого канала или выше среднего уровня левого. Сам курсор изменит в этом случае форму на  или .



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Иногда паузу перед началом файла нужно оставить. В этом случае следует избавиться от всех шумов, присутствующих в паузе, выделив ее и выполнив команду (Обработка/Тишина).

Если сигнал слабо изменяется по громкости, то полезно слегка ограничить его по уровню для того, чтобы избежать возможных искажений в процессе дальнейших преобразований. Для этого нажмите Эффекты/Динамика/Графическая и в появившемся меню нормализуйте звуковой выходи нажмите на ОК.

Далее иногда полезно пропустить файл через эквалайзер, чтобы придать звуку желаемый тембр. В Sound Forge имеется три основных типа эквалайзера: параметрический (  ) или Обработка/Эквалайзер/Параметрический), графический (  ) или Обработка/Эквалайзер/Графический) и параграфический. Параметрический эквалайзер позволяет усилить или подавить выбранную вами полосу частот, графический же 10-полосный эквалайзер дает более гибкий контроль над звуком и представлен в виде панели обычного аналогового эквалайзера, с которым многие из вас наверняка знакомы.

Теперь самое время придать звуку новые краски и оттенки, применив эффект под названием Хор. Хор создает впечатление, что звучат два или несколько источников сигнала одновременно. Это достигается путем добавления к исходному необработанному сигналу его копии, задержанной во времени и слегка измененной по высоте. Для активизации хора нажмите  (Эффекты/Хор) и в появившемся меню выберите какую-либо предустановку или регулируйте параметры самостоятельно, изменяя параметры уровня сигнала на входе, уровня необработанного сигнала на выходе, уровня обработанного сигнала на выходе, задержка обработанного сигнала относительно исходного, частота модулирования обрабатываемого сигнала, глубина модуляции, в процентах указывается, какая часть обработанного сигнала подвергается повторной обработке, количество обработок исходного сигнала эффектом.

Одним из наиболее часто применяемых звуковых эффектов, подходящих и для нашего случая, является реверберация, которая используется для придания звуку "объема" и для имитации акустических условий различных типов помещений. Практически ни одна современная музыкальная запись не обходится без использования реверберации в той или иной мере и звуковым эффектом сейчас можно считать даже не присутствие реверберации, а ее отсутствие. Эффект реверберации основан на задержке многочисленных копий исходного сигнала во времени. Sound Forge позволяет работать с восемью копиями, для каждой из которых можно независимо задавать время задержки, амплитуду и размещение в стереоспектре. Задержанные во



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

времени сигналы имитируют первое отражение звуковой волны от стен воображаемого помещения и далее подвергаются повторному преобразованию с параметрами фильтра высоких частот. Повторное преобразование служит для имитации тысяч переотражений звука, в результате чего становится невозможным услышать какое-либо отдельное эхо в определенный момент времени. В итоге после реверберации мы имеем очень естественный и насыщенный звук. Так как реверберация использует сложный алгоритм обработки, настройки этого эффекта являются одними из самых сложных в Sound Forge. Если вы не пользуетесь заданными предустановками, то вам придется потратить довольно много времени для того, чтобы найти нужное звучание. Для вызова эффекта необходимо нажать

кнопку  (Эффекты/Реверберация) и выбрать подходящую предустановку, либо заняться регулировками самостоятельно, при этом в нижней части меню реверберация вы можете видеть эхограмму, отражающую процесс регулировки.

Далее можно применить к записанному звуковому файлу такой часто используемый эффект, как Задержка/Эхо, который вызывается нажатием  (Эффекты/Задержка/Эхо). Задержка позволяет имитировать эхо и создает впечатление, что звук, который вы слышите, раздается в большом помещении, в горах или же наоборот, в чрезвычайно маленькой комнате.

На этом основную обработку аудиофайла можно закончить, если только вы не хотите добиться чего-нибудь экстраординарного.

Следует отметить, что три описанных выше эффекта (Хор, Реверберация, Задержка/Эхо) являются родственными, в их основу положен принцип задержки одной или нескольких копий исходного сигнала во времени. Поэтому применять все эти эффекты сразу к одному звуковому файлу далеко не всегда целесообразно. Часто хватает или одного из них или сочетания двух, например, Хор-Реверберация. Звук не должен в итоге оказаться забитым многочисленными эхо и непонятными шумами. Внимательно прослушивайте звуковой файл после каждой обработки и добивайтесь наилучшего результата.

· помощью имеющихся в Sound Forge средств обработки можно легко реализовать эффект постепенного нарастания громкости звучания в начале файла или затухания громкости в его конце. Достаточно лишь выделить требуемую область звуковой волны и нажать  (Обработка/Постепенное изменение/Нарастание) для получения эффекта нарастания громкости или  (Обработка/Постепенное изменение/Затухание) для ее затухания. Изменения амплитуды сигнала на протяжении всего файла можно задавать и графически ( или Обработка/Постепенное изменение/графический). Графическое



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

представление очень наглядно и предоставляет вам определенную свободу творчества.

Описанные выше звуковые эффекты используются при обработке чуть ли не каждого звукового файла. Кроме них, Sound Forge предлагает и множество других широко распространенных инструментов, которые очень интересны, но используются далеко не всегда. Это такие специфические эффекты, как флэнжер ( **К**, Эффекты/Флэнжер), интересным образом изменяющий тембр сигнала, Реверс ( **Я**, Обработка/Реверс), позволяющий воспроизвести звуковой фрагмент задом наперед.

## **ВЗадание**

3.1 Запишите звуковой фрагмент и проведите обработку наиболее часто используемыми эффектами, обогатив тем самым звуковой спектр аудиосигнала.

3.2 Запишите какие звуковые эффекты вы применили при обработке аудиосигнала.

**- Работа в кабинете (лаборатории)**

4.1 Ознакомиться с теоретическим материалом по лабораторной работе.

4.2 Выполнить предложенные задания.

4.3 Продемонстрировать результаты выполнения предложенных заданий.

## **5 Контрольные вопросы**

5.1 Какие программы, предназначенные для обработки аудио вы знаете?

5.2 Сколько и какие типы эквалайзера имеется в Sound Forge 8.0?

5.3 Какие звуковые эффекты вы знаете?

5.4 На чем основан эффект реверберации?

## **6 Содержание отчета**

6.1 название работы;

6.2 цель работы;

6.3 краткий конспект теоретической части;

6.4 ответы на контрольные вопросы;



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

6.5 выводы к работе.

## **7 Список литературы**

7.1 Гребенюк Е.И. Технические средства информатизации: Учебник для сред. проф. образования / Е.И.Гребенюк, Н.А. Гребенюк. – 2-е изд., стер. – М.: Издательский центр «Академия», 2005. – 272 с.

7.2 <http://computers.plib.ru/multimedia/SoundForge/>



## Практическое занятие 3 Компьютерная анимация

### Задание 3.1 Покадровая анимация

Для создания покадровой анимации требуется предварительно подготовить (или, по крайней мере продумать) каждый кадр фильма. При этом необходимо учитывать следующее обстоятельство: плавность перехода от одного кадра к другому и, соответственно, плавность в естественность движений персонажей зависят от того, насколько отличается следующий кадр от предыдущего (а не от скорости смены кадров, как иногда полагают). Другими словами, чем больше кадров содержит "мультик", тем ближе движения персонажей к естественным. А частота смены кадров влияет на скорость воспроизведения анимации.

Мультик, созданный с применением покадровой анимации, представляет собой последовательность ключевых кадров, с каждым из которых связано некоторое изображение (картинка) на столе.

Создадим простейшую покадровую анимацию, применяя на практике знания о создании различных типов кадров.

Наш первый мультик будет прост и примитивен. Предположим, что "героем" фильма будет шар, который падает и разбивается на три части. Будем считать, что для раскрытия столь "сложного" сюжета достаточно пяти кадров:

- в Шар в исходном положении.
- в Шар падает (в середине кадра).
- в Шар упал, но пока цел.
- в От шара откололись два кусочка.
- в На столе лежит нечто, в предыдущей жизни бывшее шариком.

учетом описанного сюжета последовательность работы должна быть следующей.

- к На временной линейке в ячейке первого кадра щелкните правой кнопкой мыши и в контекстном меню выберите команду "Вставить ключевой кадр", если пустого ключевого кадра нет.
- к Инструментом "Овал" нарисуйте круг (с заливкой) где-нибудь в верхней части стола; при этом первый кадр на временной линейке будет помечен как ключевой (черной точкой).
- к Щелкните правой кнопкой мыши в ячейке второго кадра и в контекстном меню выберите команду "Вставить ключевой кадр". На линейке появился второй ключевой кадр, при этом кадр полностью повторил предыдущий. Обратите внимание: при выделенном кадре в ячейке на временной линейке (темным цветом) все изображение у нас сразу выделено (можно не выделять наш объект заново). Переместите изображение шара в среднюю часть монтажного стола - шар в процессе падения.
- к Щелкните правой кнопкой мыши в ячейке третьего кадра и в контекстном меню вновь выберите команду "Вставить ключевой кадр" или нажмите клавишу F6. Новый кадр также будет помечен как ключевой, и для него сохранится предыдущее изображение. Переместите шар в нижнюю часть стола.
- к Создайте четвертый кадр, выбирая в контекстном меню команду "Вставить ключевой кадр". Инструментом "Лассо" выделите поочередно "кусочки шара" и перенесите их в сторону (можно к кусочкам применить трансформацию объекта, например поворот).
- к Создайте пятый кадр, выбирая в контекстном меню команду "Вставить кадр".

После завершения работы нужно проверить, все ли получилось так, как вы хотели. Чтобы воспроизвести фильм, достаточно просто нажать клавишу Enter (альтернативный вариант - выбрать в меню "Управление" команду "Воспроизвести"). При этом считывающая головка панели временной диаграммы автоматически перемещается от текущего к



## Негосударственное частное образовательное учреждение высшего образования «Технический университет»

последнему ключевому кадру. Вы можете принудительно выбрать любой кадр для просмотра, щелкнув в соответствующей ячейке временной диаграммы левой кнопкой мыши.

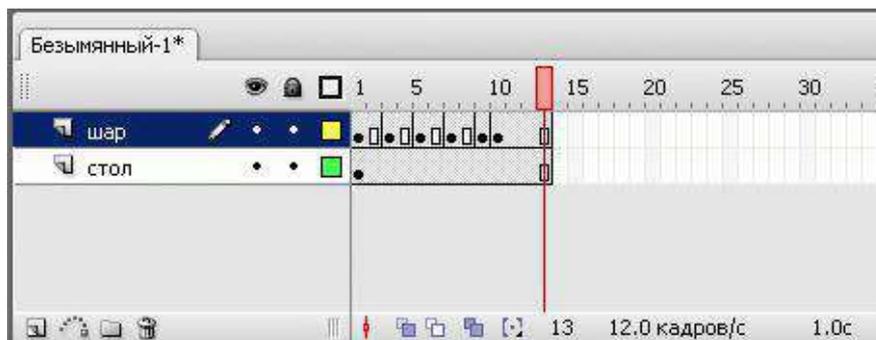
Чтобы просмотреть анимацию "глазами пользователя", т. е. как она будет выглядеть в итоге, нажмите комбинацию клавиш CTRL+Enter или Shift+CTRL+Enter (в меню "Управление" - "Тестировать ролик" или "Тестировать сцену").

Просмотрев полученный мультик, мы имеем два вывода:

- О Наконец-то! У меня на экране что-то движется.
- О Можно ли исправить? У меня получились резкие изменения, и сам мультик слишком короткий и быстрый.

Исправим некоторые недочеты. Для этого:

- в вставим еще парочку ключевых кадров для движения шара - выберем первый ключевой кадр, в контекстном меню вновь выберем команду "Вставить ключевой кадр" или нажмем клавишу F6 и передвинем изображение шара в промежуточное положение между начальным и "средним". Так же добавим ключевой кадр с промежуточным положением - теперь между третьим и четвертым кадром;
- в если на ваш взгляд анимация происходит слишком быстро, добавим к каждому ключевому кадру по дополнительному промежуточному. Для этого выбираем ключевой кадр, в контекстном меню выбираем команду "Вставить кадр" или нажимаем клавишу F5;



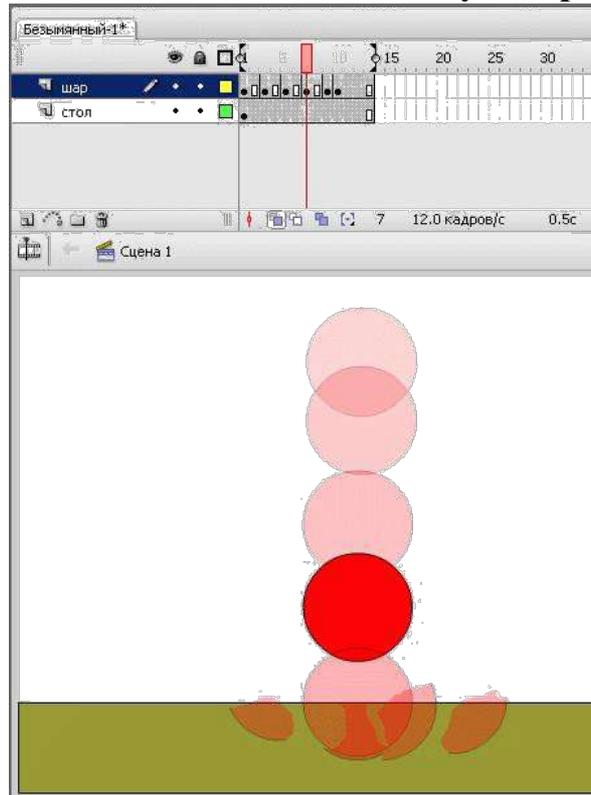
- В добавим к анимации фон - поверхность, о которую разбивается наш шар. Для этого создадим еще один слой в панели временной линейки, перетащим его вниз (наш "стол" должен быть под шариком). По умолчанию при создании нового слоя сразу создается пустой ключевой кадр и промежуточные по длине уже готовой анимации. Проверим, что находимся в ключевом кадре нового слоя, и внизу монтажного стола нарисуем прямоугольник, обозначающий "стол". Убедитесь, что изображение прямоугольника есть на всех кадрах. В этом слое нам больше не потребуются ключевые кадры, так как фон статичен и не предполагает движения.

Рассмотрение возможностей временной шкалы нельзя считать полным без упоминания

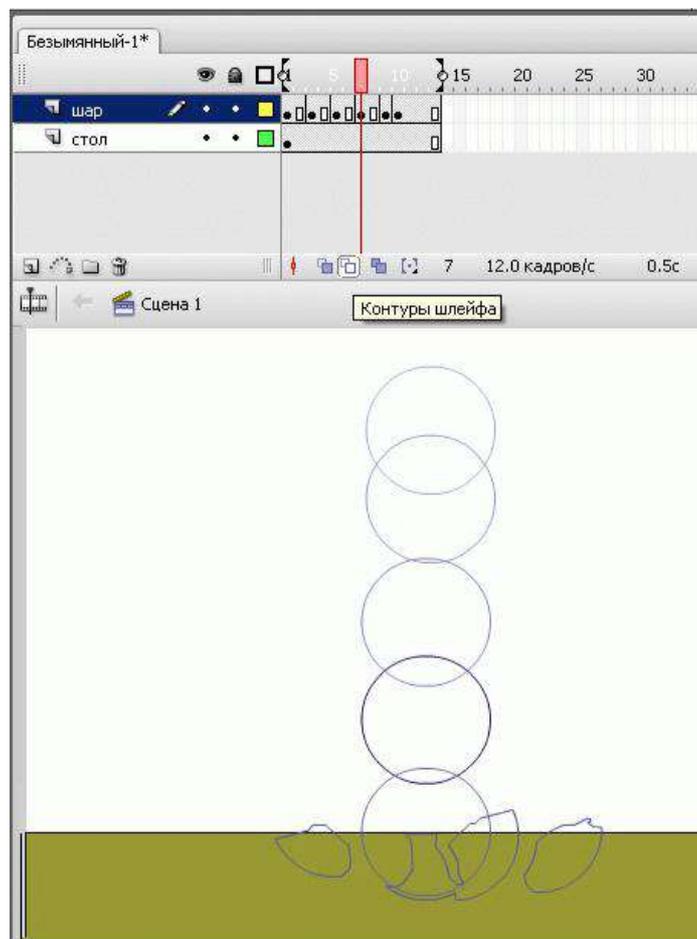
- 7) пяти кнопок, расположенных в низу линейки кадров. а)

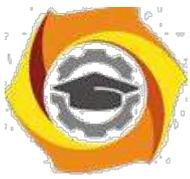


Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»



b)





## Негосударственное частное образовательное учреждение высшего образования «Технический университет»

к помощью кнопок "Шлейф" или "Контур шлейфа" можно проконтролировать анимационное преобразование. При нажатии на одну из них на рабочее поле выводится содержание не только текущего кадра, но и нескольких соседних. "Шлейф" - выводит содержание соседних кадров в полупрозрачном виде, а "Контур шлейфа" - в виде контуров. Предыдущий рисунок иллюстрирует возможность увидеть на рабочем поле сразу несколько кадров созданной нами анимации.

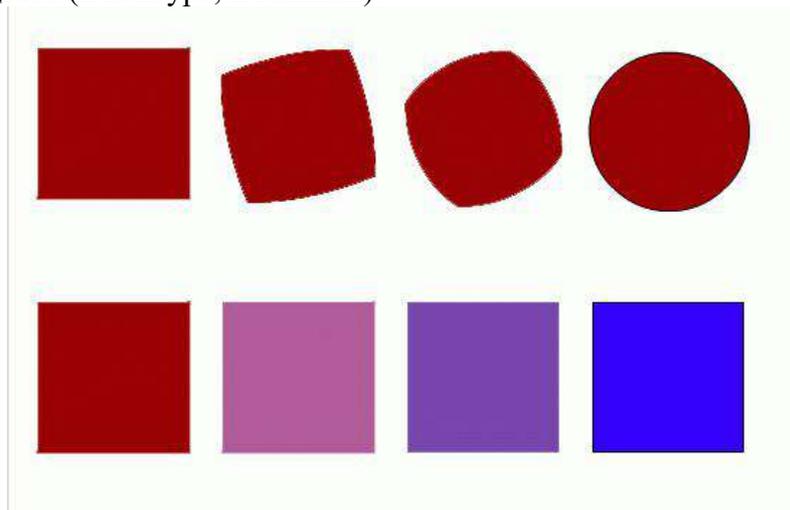
Итак, первая анимация готова. У вас в руках есть все возможности ее редактирования (кадры, слои, объекты).

Сохраняйте промежуточный результат! Сохраненный файл будет иметь расширение .fla. Это формат редактируемого файла во Flash.

Чтобы сохранились только результаты нашей работы, законченный файл должен пройти процесс компиляции. Такая компиляция происходит, когда мы нажимаем комбинацию клавиш CTRL+Enter. При этом создается файл с расширением .swf. Если файл .fla уже был ранее сохранен в какой-нибудь папке, то в той же папке сохраняется и вновь созданный SWF-файл. Его можно открыть для просмотра двойным щелчком в "Проводнике". На любом этапе создания анимации, в том числе и на любом этапе рисования, можно посмотреть на свой труд глазами "пользователя". Для этого нужно всего лишь нажать комбинацию клавиш CTRL+Enter. Чтобы продолжить работу над роликом, закрываем окно тестирования и возвращаемся в среду разработки.

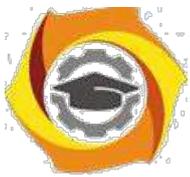
Уже по первому опыту становится ясно, что создание покадровой анимации - длительный и трудоемкий процесс. Для уменьшения трудоемкости применяется автоматическая анимация. Учтите: автоматизация не решит всех проблем, иногда не даст желаемого результата - для создания хорошего мультфильма "только компьютера" недостаточно.

**Задание 3.2 Автоматическая анимация трансформации объекта - анимация формы** Используя анимацию трансформации, вы можете создавать эффект плавного "перетекания" объекта из одной формы в другую. Причем результирующая форма может не иметь абсолютно ничего общего с исходной. Кроме изменения формы можно получить эффект изменения цвета (и контура, и заливки).



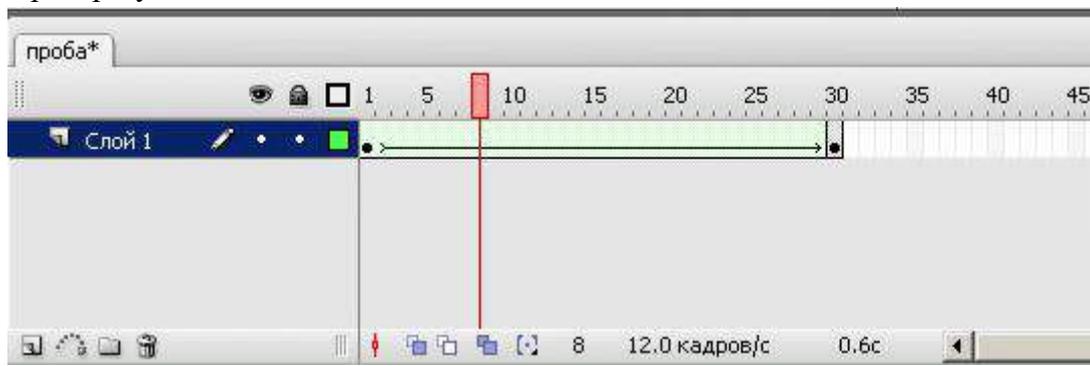
Рассмотрим анимацию формы на примере.

15. Инструментом "Овал" нарисуем круг в первом ключевом кадре. Анимация формы применяется только для несгруппированных объектов (голубой рамки вокруг нашего изображения не должно быть). В противном случае или выполняем команду "Разделить", или рисуем заново.



## Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- в Щелчком правой кнопкой мыши в ячейке того кадра, который хотим сделать последним в преобразовании, например 24-го (по умолчанию скорость нашей анимации 12 кадров в секунду, значит мы будем любоваться преобразованием 2 секунды). В контекстном меню выберем команду "Вставить ключевой кадр" или нажмем клавишу F6. В результате получим следующую "цепочку": ключевой кадр, 22 промежуточных, идентичных первому ключевому, ключевой кадр.
- в Во втором ключевом кадре заменим изображение круга на квадрат другого цвета. (Можно было вставить пустой ключевой кадр и нарисовать фигуру).
- в Выберем первый ключевой кадр, щелкнув в ячейке первого кадра (не на линейке кадров!) На столе появилось изображение первого кадра, а в панели свойств отобразились настройки для кадра. В раскрывающемся списке "Анимация" выберем "Форма - анимация формы" (по умолчанию там стоит "Нет"). В результате первый и последний ключевые кадры на временной линейке будут соединены стрелкой на светло-зеленом фоне (рис. 5.6). Это говорит о том, что создание анимации успешно завершено.
- в Проверим результат.



Ключевые кадры для анимации формы (трансформации) обозначаются как черные точки, соединенные линией со стрелкой, на светло-зеленом фоне (линия со стрелкой заменяет все промежуточные кадры). Если у вас между кадрами появилась пунктирная линия, значит, или конечный (заключительный) ключевой кадр отсутствует, или в настройках анимации что-то сделано не так.

Выбор пункта "Форма" в списке "Анимация" приводит к изменению формата панели свойств. С ее помощью могут быть установлены следующие дополнительные параметры анимации трансформирования объекта:

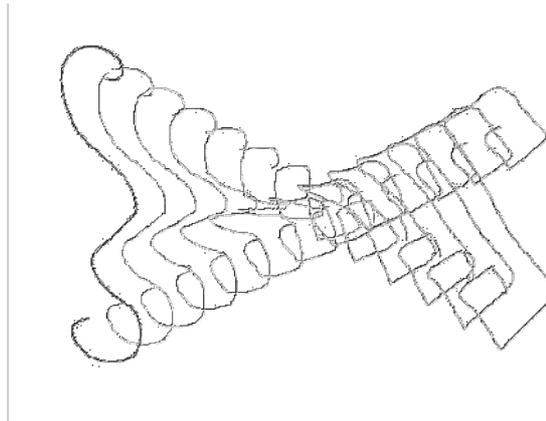
- 28. скорость изменений; по умолчанию изменения протекают с постоянной скоростью, однако вы можете управлять ею, изменяя значения параметра "Замедлить": отрицательные значения этого параметра (от -1 до -100) означают, что изменения будут постепенно ускоряться, а положительные значения (от 1 до 100), наоборот, означают постепенное замедление изменений;
- 29. способ трансформации; он определяется значением, выбранным в списке "Смешать":
- 30. "Дистрибутив" ("Размазанный") - Flash создает анимацию, в которой промежуточные формы являются более сглаженными;
- 31. "Угловой" - Flash создает анимацию, при которой в промежуточных формах сохраняются очевидные углы и прямые линии.

По умолчанию Flash пытается произвести переход от одной формы к другой "кратчайшим путем". Поэтому промежуточные кадры могут оказаться весьма неожиданными для вас - приходится редактировать изображения или искать другие способы анимации.

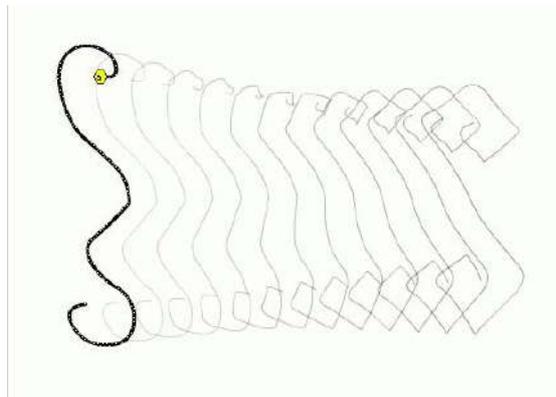
а)



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»



b)

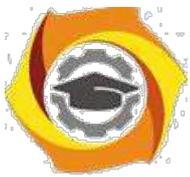


Для управления более сложными изменениями формы используются так называемые контрольные точки. Они определяют те точки исходной формы, взаимное расположение которых требуется сохранить при переносе в новую форму.

Контрольные точки обозначаются на изображении небольшими кружками с буквами. Каждой метке-точке на исходном изображении должна соответствовать точка с тем же именем на результирующем изображении. Всего для одной фигуры может быть использовано не более 26 контрольных точек (по числу букв латинского алфавита). Метки на исходном изображении окрашены в желтый цвет, на результирующем - зеленым.

Для расстановки контрольных точек требуется выполнить следующие действия:

14. Создайте анимацию формы между двумя ключевыми кадрами.
15. Щелкните левой кнопкой мыши в ячейке ключевого кадра, соответствующего исходному изображению (первого ключевого кадра).
16. В меню "Изменить" выберите каскадное меню "Форма", а в нем - команду "Добавить контрольную точку"; в результате на изображении появится "заготовка" первой метки - кружок красного цвета с буквой "a".
17. Переместите его мышью на ту точку изображения, которую вы хотите пометить как контрольную (Расстановку меток следует выполнять с помощью инструмента "Стрелка" при включенном модификаторе "Привязка к объектам"). Контрольные точки должны находиться на контурах и границах объектов.
18. Щелкните левой кнопкой мыши в ячейке кадра, соответствующего результирующему изображению (второго ключевого кадра в анимации формы). В центре будет присутствовать красный кружок с той же буквой, что и в исходном кадре.
19. Переместите кружок в точку изображения, которая должна соответствовать отмеченной в исходном кадре; после перемещения кружок изменит цвет на зеленый.



## Негосударственное частное образовательное учреждение высшего образования «Технический университет»

11. Вернитесь на первый ключевой кадр анимированной последовательности и убедитесь, что цвет метки изменился на желтый.
12. Если требуется продолжить расстановку контрольных точек, повторите описанную процедуру для каждой из них. Если вы используете при трансформации несколько контрольных точек, то выполняйте расстановку меток по часовой стрелке, начиная с левого верхнего угла изображения. Это обусловлено тем, что Flash обрабатывает контрольные точки в алфавитном порядке.

После расстановки контрольных точек целесообразно провести тестовый запуск фильма и при необходимости скорректировать их расстановку и выбор. Всегда можно удалить лишние метки или добавить новые (в первом ключевом кадре анимации). После того как результат вас удовлетворит, вы можете "спрятать" контрольные точки. Перечисленные операции удобнее всего выполнять с помощью контекстного меню. Чтобы его открыть, следует щелкнуть на одной из меток правой кнопкой мыши.

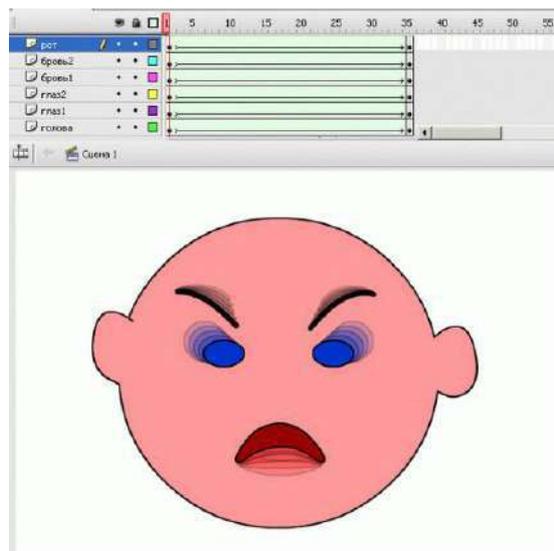
Анимация формы просто завораживает подростков - они могут по несколько минут любоваться самыми простыми преобразованиями. При этом, как показывает опыт, они в большинстве своем не видят нелогичности и "неправильности" преобразований, когда все находится в движении.

Особенность применения автоматической анимации состоит в том, что с ее помощью можно анимировать только один объект на слое. Для создания сцены, в которой предполагается наличие нескольких анимированных объектов, требуется разместить каждый из них на отдельном слое.

Выполним мини-упражнение: нарисуем рожицу и анимируем ее мимику.

При выполнении этого задания можно долго биться над преобразованием, если всю рожицу нарисовать в ключевом кадре одного слоя, а в другом ключевом кадре этого же слоя нарисовать преобразованную рожицу. В принципе задание может получиться, но слишком много факторов при этом надо учесть. Ведь придется расставлять контрольные точки для всех основных элементов.

13. можно выполнить эту работу быстро, "полушутя": посчитаем, сколько "объектов" должны двигаться в мимике рожицы - например, само "лицо" (смена цвета и формы), рот (смена формы), глаза - два, брови две (!) - всего 6 объектов, значит должно использоваться 6 слоев.



Чем проще объекты в ключевых кадрах, тем проще программе создать "правильные" промежуточные кадры при просчитывании анимации формы. Осталось только не перепутать слои при рисовании (будьте внимательнее!).



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

**Практическое занятие 4 Подготовка презентаций с  
помощью Microsoft Office PowerPoint**

**12. Цель работы**

- 1.1 научиться размещать текст и рисунки на слайдах
- 1.2 научиться использовать шаблоны оформления
- 1.3 применять анимационные эффекты
- 1.4 подготавливать презентацию к показу и печати

**2 Пояснение к работе**

**2.1 Краткие теоретические сведения**

Назначение Microsoft Office PowerPoint состоит в подготовке электронной презентации. Презентация – это связанные между собой слайды. Слайд – это законченный по смыслу фрагмент презентации. В оформлении презентации применяются: текст, графика, эффекты анимации, звук, видеоклипы.

Можно создавать два вида презентаций: *автоматическую* и *управляемую докладчиком*. В автоматической презентации смена слайдов происходит по времени, в управляемой же презентации нужно щелкать мышью, чтобы перейти на следующий слайд.

При создании большой презентации используют сортировщик слайдов, оглавление и управляющие кнопки. *Сортировщик слайдов* — это режим просмотра, в котором удобно копировать, перемещать, вставлять и удалять слайды. На *итоговом слайде* размещают гипертекстовые ссылки на слайды разделов, поэтому с итогового слайда можно быстро перейти на нужный слайд. *Управляющие кнопки* (вперед, назад, выход и т. п.) вставляются в презентацию, управляемую докладчиком, они помогают просматривать слайды в ручном режиме.

После создания презентации необходимо выполнить следующие действия:

- У просмотреть презентацию в режиме показа слайдов, в этом режиме слайды занимают весь экран компьютера, именно так они выглядят и во время демонстрации на проекторе;
- У если нужно, изменить текст и оформление, добавить эффекты анимации, звук, видео;



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

8. упаковать файл презентации и записать на компакт-диск чтобы показывать на другом компьютере;
9. распечатать раздаточный материал.

### **Основные приемы работы с презентацией Создание новой презентации**

Существует три способа создания новой презентации.

*1-й способ:* при запуске программы PowerPoint автоматически создается новая презентация

*2-й способ:* **Файл -> Создать ->** <в области задач **Создание презентации** выберите пункт **Из шаблона оформления**> - выберите нужный шаблон>;

*3-й способ:* **Файл -> Создать ->** <в области задач **Создание презентации** выберите пункт **Из мастера автосодержания**> -><выполните указания мастера> ->• <нажмите кнопку **Готово**>.

### **Открытие существующей презентации**

Для открытия существующей презентации можно воспользоваться одним из трех способов.

*1-й способ:* <запустить программу PowerPoint -> **Файл -> Открыть**>;

*2-й способ:* <в Проводнике открыть диск (папку), где находится презентация> -> <сделать двойной щелчок по значку презентации>

*3-й способ:* **Пуск -> Документы ->** <щелкнуть по названию презентации >;

' разделе **Документы** меню **Пуск** находятся названия не всех, а только недавно открывавшихся документов.

### **Сохранение презентации**

В программе PowerPoint презентацию можно сохранить в двух типах файла:

PPT — используется для редактирования презентации;

PPS — используется для демонстрации презентации.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

*1-й способ:* сохранение новой презентации для редактирования: **Файл -> Сохранить как** -> <выбрать диск и папку, в поле **Имя файла** набрать название презентации, проверить тип файла: **Презентация (\*.ppt)**> -> **Сохранить;**

*2-й способ:* сохранение изменений в документе: **Файл -> Сохранить;**

*3-й способ:* сохранение презентации для просмотра: **Файл -> Сохранить как** -> выбрать диск и папку, в поле **Имя файла** набрать название презентации> -> <из списка **Тип файла** выбрать **Демонстрация PowerPoint (\*.pps)**> -> **Сохранить. Добавление слайда**

Новый слайд вставляется всегда после текущего (выделенного) слайда одним из способов:

*1-й способ:* создание пустого слайда — **Вставка -> Создать;**

*2-й способ:* копия текущего слайда — **Вставка -> Дублировать слайд. Удаление слайда**

Для удаления слайда существуют два способа.

*1-й способ:* <выделите эскиз слайда в области структуры> -> <нажмите клавишу **Delete**>.

*2-й способ:* <щелкните по эскизу слайда в области эскизов правой кнопкой мыши> -> **Удалить слайд.**

**Изменение порядка слайдов**

Для изменения порядка слайдов в презентации удобно пользоваться режимом отображения **Сортировщик слайдов**. Для перехода в этот режим выполните команду **Вид -> Сортировщик слайдов**. Порядок слайдов изменяется перетаскиванием эскизов слайдов.

**Вставка текстового поля**



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Для вставки текстового поля выполните команду **Вставка -> Надпись ->** <щелкните мышью на слайде в месте вставки текстового поля>. На слайде появится небольшое поле, в котором мигает текстовый курсор.

### **Вставка рисунка или других объектов**

Вставка рисунка или нужного объекта возможна двумя способами:

*1-й способ:* с помощью значков макета — <в области задач **Макет слайда** примените макет с объектом (найдите заполнитель со значками)> -> <на слайде щелкните значок нужного объекта>;

*2-й способ:* с помощью меню **Вставка** — <примените нужный макет с объектом> -> <щелкните в заполнителе, в который предполагается вставить объект> -> **Вставка** -> <выберите нужный тип вставляемого объекта>

### **Применение шаблонов оформления**

Существуют три способа применения шаблонов оформления.

*1-й способ:* **Формат -> Дизайн слайда** (или нажмите кнопку **Дизайн слайда** Ц на панели инструментов **Форматирование**);

*2-й способ:* если требуется применить шаблон оформления ко всем слайдам презентации, выберите один эскиз слайда и щелкните по нужному шаблону оформления;

*3-й способ:* если требуется применить шаблон оформления только к выбранным слайдам презентации, выберите эти слайды (выбирайте слайды, удерживая клавишу **Ctrl**), а затем щелкните по нужному шаблону оформления.

**Применение макета слайда **Формат -> Макет слайда**** (или щелкните правой кнопкой мыши эскиз слайда на вкладке **Слайды** и выберите в контекстном меню команду **Макет слайда**) -> <убедитесь, что выбран эскиз того слайда, к которому требуется применить макет, и щелкните по макету>.

Для того чтобы вставить новый слайд с заданным макетом, выберите нужный макет и выберите в списке команду **Добавить новый слайд**.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

**Применение эффектов анимации**  
Для применения эффектов анимации выполните действия: **Формат -\* Оформление слайда** -> <в области задач **Дизайн слайда** щелкните по ссылке **Эффекты анимации**, после чего в области задач появится список эффектов> -> <щелчком мыши выберите желаемый эффект>.

Для того чтобы увидеть результат, в области задач **Дизайн слайда** нажмите кнопку **Просмотр**.

Эффект анимации по умолчанию применяется не ко всей презентации, а только к выделенным слайдам. Для того чтобы применить выбранный эффект ко всем слайдам презентации, в области задач **Дизайн слайда** нажмите кнопку **Применить ко всем слайдам**. **Просмотр и редактирование страниц заметок**  
Для просмотра и редактирования страниц заметок выполните команду **Вид - Страницы заметок** --> <добавьте текст и примените дополнительное форматирование>.

**Настройка презентации** Настройка презентации осуществляется при выполнении команд **Показ слайдов -> Настройка презентации -> Автоматический (полный экран) - ОК**.

*Обратите внимание*, что в нижней группе **Смена слайдов** переключатель должен стоять на опции **По времени**. Таким образом, вы задаете режим непрерывной смены слайдов, пока не будет нажата клавиша Esc.

**Переход слайда** Для перехода слайда выполните команду **Показ слайдов - Переход слайда**.

- В В группе **Продвижение** обязательно снимите флажок с команды **По щелчку** (иначе следующий слайд не покажется, пока зритель не догадается щелкнуть мышкой по экрану).
- В Поставьте флажок на команду **Автоматически после** и задайте через сколько секунд появится следующий слайд (5-10 с).
- В В списке **Эффект** (изначально в списке выбрана команда **Нет перехода**) выберите вид перехода и укажите скорость перехода (**Медленно** или **Средне**).
- В Можно также выбрать звук, с которым будет появляться следующий слайд.
- В Нажмите **Применить**.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

### **Показ слайдов презентации**

Показ слайдов можно выполнить тремя способами:

*1-й способ:* Вид -> **Показ слайдов**;

*2-й способ:* нажмите клавишу **F5** (чтобы начать с первого слайда)  
или **Shift+F5** (чтобы начать с текущего слайда);

*3-й способ:* щелкните по слайду, с которого требуется начать просмотр презентации, а затем нажмите кнопку **Показ слайдов W** в нижнем левом углу окна.

**Выход из режима показа презентации:** нажмите клавишу **Esc**.

### **Запуск показа слайдов из файла**

Для показа слайдов выполните действия: <в окне **Мой компьютер** или **Проводнике** выберите файл презентации, который требуется открыть для показа слайдов> -> <щелкните правой кнопкой мыши по значку или имени файла презентации> - <выберите команду контекстного меню **Показать X**

## 2.2 Перечень используемого оборудования

Персональный компьютер



### **- Задание**

3.1 Откройте папку Учебные материалы/Картинки/Зоопарк

3.2 Создайте превенцию «Московский зоопарк» как на рисунке со следующими слайдами:

•  
Слайд 1 – Московский зоопарк;

•  
Слайд 2 – Рыбы;

•  
Слайд 3 – Рептилии;

•  
Слайд 4 – Птицы

•  
Слайд 5 – Остров зверей;



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Слайд 6 – Адрес зоопарка.

***Текст для слайда «Московский зоопарк»:***

Добро пожаловать в Московский зоопарк! Московский зоопарк, старейший и крупнейший зоопарк нашей страны, был основан в 1864 году.

Сейчас коллекция зоопарка насчитывает 6094 экземпляров 100 \ видов животных.

Площадь зоопарка — 21,4676 га. Количество посетителей — 1 290 385 чел.

***Текст для слайда «Адрес зоопарка»:***

Зоопарк расположен по адресу: Москва, Б. Грузинская, 1. Зоопарк работает ежедневно, кроме понедельника:

- летом: с 10.00 до 20.00;
- зимой: с 10.00 до 17.00

**- Работа в кабинете (лаборатории)**

4.1 Ознакомиться с теоретическим материалом по лабораторной работе.

4.2 Выполнить предложенные задания.

4.3 Продемонстрировать результаты выполнения предложенных заданий.

**10. Контрольные вопросы**

5.1 Какие два вида презентации можно создать в PowerPoint?

5.2 Как изменить шаблон оформления?

5.3 Зачем используются эффекты анимации?

5.4 Как задать эффект при переходе к новому слайду?



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

5.5 Как установить непрерывный показ слайдов (автоматическая презентация)?

5.6 Что такое презентация?

5.7 Что такое слайд?

## **6 Содержание отчета**

6.1 название работы;

6.2 цель работы;

6.3 краткий конспект теоретической части;

6.4 ответы на контрольные вопросы;

6.5 выводы к работе.

## **11. Литература**

7.1 Пакеты прикладных программ: Учеб. пособие для сред. проф. образования / Э.В. Фуфаев, Л.И. Фуфаева. – М.: Издательский центр «Академия», 2004. – 352 с.

7.2 Практикум по информатике: учеб. пособие / Под ред. Л.Г. Гагариной. Ч. - – М.: ИД «Форум»: Инфра – М, 2006. – 320 с.: ил. – (Профессиональное обр



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

## Практическое занятие 5 Работа с цифровыми устройствами ввода изображений

### - Цель работы

Ознакомиться с технологией работы с цифровыми устройствами ввода изображений,  
Ознакомиться со способами подключения к компьютеру.

## 2 Пояснение к работе

### 2.1 Краткие теоретические сведения

#### Устройство цифровых фотоаппаратов

#### Конструкция фотоаппаратов



Любой фотоаппарат, в том числе и цифровой, можно условно разделить на три части.



1.  
оптическая система, состоящая из объектива (иногда с насадками) и затвора;
2.  
регистратор изображения;
3.  
устройство хранения отснятых кадров.

- обычном фотоаппарате функции второй и третьей частей выполняет плёнка, в цифровом для этого используется два разных устройства.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Для регистрации изображения используется **электронно-оптический преобразователь (ЭОП)**, а для хранения – флэш-память

### **Оптическая система**

Одной из составляющих фотоаппарата является его объектив. Чем больше фокусное расстояние, тем меньше угол зрения - предметов попадает в кадр меньше, но их размер в кадре больше. И, наоборот, при уменьшении фокусного расстояния объекты съемки становятся меньше, но в кадр их попадает больше.

Часто фокусное расстояние для цифровой фотокамеры указывается двумя цифрами, например, 6-15 мм (28-72 мм).

Применяются объективы:

С объективы с **переменным фокусным расстоянием zoom** (вариообъективы). Под кратностью объектива подразумевают отношение максимального фокусного расстояния к минимальному, например, **105/35 = 3** – кратность объектива равна 3.

в объективы с **постоянным фокусным расстоянием fixed focus**. Постоянное фокусное расстояние несколько ограничивает возможности фотографа, в то же время конструкция таких устройств предельно проста.

Для портретной, пейзажной и спортивной съемки используются разные объективы, каждый с наиболее подходящим фокусным расстоянием. Они закрепляются на камере с использованием резьбового либо байонетное соединения.

Сменная оптика сохранилась в зеркальных камерах.

**Зеркальной (SLR - single lens reflex)** называется камера, в которой изображение, попадающее в объектив, с помощью специальной оптической системы проецируется на поверхность экрана фокусировки. Это изображение пользователь наблюдает в видоискателе и визуально контролирует кадрирование и фокусировку.

Для точного определения дистанции съемки применяются разнообразные оптические устройства. Одним из них является **микрорастр**, система микроскопических пирамидок, нанесенных на поверхность экрана фокусировки.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Чтобы изображение попало на видеоискатель, используется либо зеркальце, убирающееся в момент съемки, либо полупрозрачная призма.

Последнее время многие модели этого класса используют принцип видеокамер - вместо оптического видеоискателя в них установлен цветной ЖК-дисплей с хорошим разрешением - порядка 130 тысяч элементов. При этом на дисплей выводится дополнительная информация - значения диафрагмы, выдержки, количество кадров и т. д.

Важнейшим оптическим определением является **экспозиция**.

нашем случае светочувствительным элементом является ПЗС-матрица. От экспозиции, сообщенной матрице, во многом зависит качество снимка - недостаточная экспозиция (называемая фотографами недодержкой) приводит к плохой проработке деталей в тенях, избыточная экспозиция (передержка) - к плохой проработке светлых участков. Для управления экспозицией используются **диафрагма** и **выдержка**, для расширения их диапазона применяют материалы с более высокой светочувствительностью.

Диафрагма представляет собой светонепроницаемую преграду с центральным отверстием изменяемого диаметра. Наиболее распространена ирисовая диафрагма, у которой световое отверстие образуется несколькими дугообразными лепестками (ламелями), соединенными с подвижным кольцом-коронкой. При повороте кольца лепестки сходятся (или расходятся), плавно уменьшая (или увеличивая) отверстие диафрагмы. Величина действующего отверстия диафрагмы изменяется в зависимости от условий съемки (освещенности фотографируемого объекта и чувствительности ПЗС-матрицы), а также выдержки. От величины отверстия диафрагмы зависит диапазон резко изображаемого пространства - чем меньше отверстие, тем больше глубина, резкости, и наоборот.

Следующей важной характеристикой ПЗС-матрицы является **светочувствительность**.

Чем выше светочувствительность, тем меньшее количество света требуется для реакции материала.

После прохождения оптики световой поток попадает на регистрирующий элемент.

В качестве ЭОП используется два типа устройств – ПЗС- матрицы (матрицы приборов с зарядовой связью) и КМОП-матрицы. По конкуренции данные устройства примерно похожи основное различие заключается в разрешении. Эта же характеристика является одной из основных при



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

описании цифровой фотокамеры. Именно дороговизна ПЗС-матрицы с большим количеством элементов сдерживала развитие цифровой фотографии. И если для любительской видеокамеры достаточно матрицы из 300 тысяч элементов, то для фотографии размером 9x12 см необходимо наличие как минимум 1 мегапикселя. Мегапиксельные ПЗС-матрицы с разрешением 1280x960 появились в цифровых фотокамерах в конце 1997 года. Подавляющее большинство любительских фотоаппаратов оснащены ПЗС-матрицами.

Чем больше мегапикселей имеет светочувствительная матрица фотоаппарата, тем больше разрешение будет у фотографий. Чем выше разрешение, тем большего размера фотографию можно распечатать.

10. для фотографии 10 x 15 достаточно камеры 2 мегапикселя
11. для фотографии 12 x 17,5 достаточно камеры 3 мегапикселя
12. для фотографии 20 x 25 достаточно камеры 4 мегапикселя
13. для фотографии 27,5 x 35 достаточно камеры 5 мегапикселя
14. для фотографии 40 x 50 достаточно камеры 7 мегапикселя

### **Устройства хранения информации**

Световой поток проходит через оптическую подсистему на электронно-оптический преобразователь (ЭОП). Полученное аналоговое изображение посредством преобразователей стало цифровым. Скорость считывания кадра

ПЗС-матрицы значительно выше скорости записи на устройство долговременного хранения информации, какого бы типа оно ни было. Для промежуточного хранения и обработки изображения используется буферная память.

**Буферная память**, аналогична ОЗУ, используемому в персональных компьютерах. Основное отличие в том, что при выборе тех или иных микросхем основное внимание уделяется не столько быстродействию (хотя и оно немаловажно), сколько надежности и малому энергопотреблению.

27. буфер может помещаться и обрабатываться не один, а несколько кадров. Таким образом, значительно сокращается интервал, необходимый для подготовки камеры к следующей съемке, практически время затрачивается только на зарядку вспышки. Если не использовать вспышку, то становится доступным режим непрерывной съёмки, когда камера делает несколько (до



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

кадров с высокой скоростью (до 3 кадров в секунду) и помещает их в буфер, где они обрабатываются и потом записываются в долговременную память. Если АЦП матрицы обеспечивает высокую пропускную способность, то возможен режим видеосъемки - затвор при этом остается открытым на все время съемки.

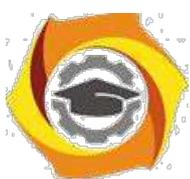
После того как программное обеспечение камеры создало на основе данных с ПЗС-матрицы полноцветное изображение, возникает задача его сохранения. Графические файлы очень велики и поэтому требуется их дополнительная обработка – сжатие в форматы JPEG, TIFF, RAW.

В устройствам долговременного хранения предъявляется ряд жестких требований. Во-первых, необходима возможность продолжительного хранения без источников питания. Во-вторых, требуется минимальное энергопотребление при операциях записи/считывания/стирания. В-третьих, время записи/считывания/стирания должно быть как можно меньше. В-четвертых, габариты должны быть минимальными. И, наконец, самое главное – устройство обязано быть стопроцентно надежным. Перечисленным требованиям в наиболее полной мере удовлетворяют конструкции, использующие так называемую **флэш-память**.

Этот тип памяти является промежуточным между ПЗУ (постоянное запоминающее устройство, в литературе - АОМ, read-only memory), которое хранит информацию без источников питания, но не позволяет ее модифицировать, и ОЗУ, которое допускает информацию модифицировать, но хранить ее не может. Флэш-память использует питание только при считывании данных и их модификации, причем для считывания необходимо менее высокое напряжение, а для записи – повышенное.

Разработкой цифровой техники независимо друг от друга занимались различные компании, что привело к созданию таких устройств хранения информации:

- с Карты PCMCIA
- с CompactFlash
- с SmartMedia
- с MultiMedia Card
- с Memory Stick
- с xD – Picture Card
- с Secure Digital)



Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»



**Подключение цифрового фотоаппарата к компьютеру**



После того как фотографии с помощью цифрового фотоаппарата сделаны, их можно либо распечатать сразу, либо отредактировать на компьютере.

**Минимальные системные требования:**

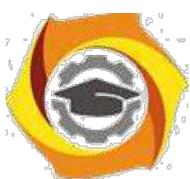
- 36. операционная система Microsoft Windows XP и выше;
- 37. процессор Pentium II 1,2 ГГц;
- 38. 128 Мб оперативной памяти;
- 201 Мб места на жестком диске;



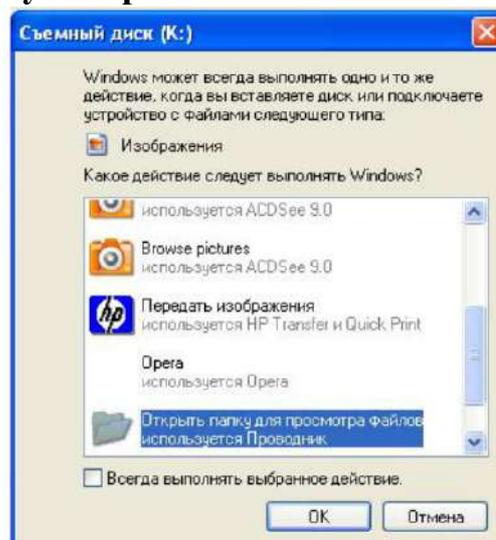
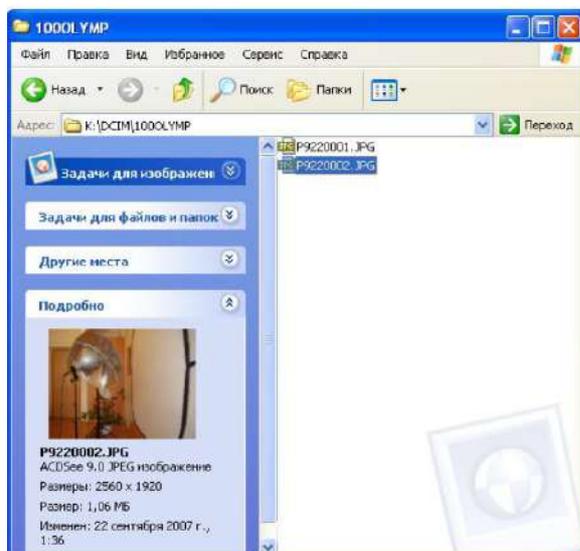
разрешение экрана от 1024x768 с глубиной цвета 32 бит.

Соединение камеры с компьютером осуществляется при помощи специального кабеля (современные камеры обычно используют интерфейс USB). В некоторых моделях предусмотрена базовая станция, в которую устанавливается камера. Базовая станция подсоединяется к компьютеру и источнику питания для зарядки аккумуляторов.

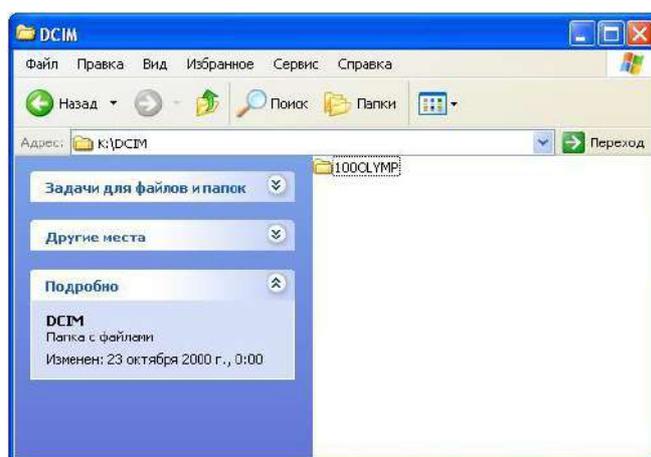
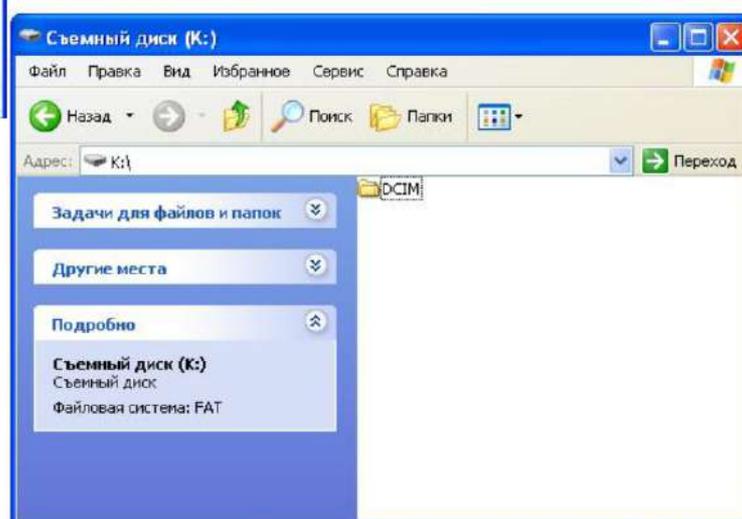
При подключении к компьютеру фотоаппарат распознается как внешний накопитель. Если используется Windows XP, то проблем с драйверами не возникает и все работает через 10-30 секунд после подключения.



Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»



Также используется кардридер -  
устройство чтения карт памяти.  
Обычно кардридеры универсальны  
В читают несколько типов  
флэш-карт.  
Далее фотоизображения можно  
скопировать на компьютере и  
обработать с помощью любого



графического редактора.

Возможности графического  
редактора Adobe Photoshop,  
наиболее часто используемые  
при обработке фотографий:

- открытие и сохранение файлов  
в различных форматах;
- работа с цветом, коррекция  
оттенков, яркости и контраста

изображения;

- 6) исправление эффекта «красных глаз»;
- 7) удаление случайно попавших в кадр объектов;



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

в исправление искажений перспективы (например, выпрямление падающих зданий), коррекция искажений, вносимых линзами объектива;

в добавление к изображению поясняющих надписей;

в организация коллекции изображений и подготовка фотографий к отправке по электронной почте или размещению в Интернете;

в подготовка изображений к печати и печать на домашнем принтере или в цифровой фотолаборатории.

## 2.2 Перечень используемого оборудования

Персональный компьютер

### 8) Задание

3.1 Рассмотрите предложенный цифровой фотоаппарат. Зарисуйте схематично его и укажите составные части. Узнайте и запишите, сколько мегапикселей имеет светочувствительная матрица фотоаппарата, и какого максимального разрешения с его помощью можно сделать фотографии.

3.2 Сфотографируйте что-нибудь на фотоаппарат. Подключите камеру к компьютеру и скопируйте с него изображения в свою рабочую папку.

### 6. Работа в кабинете (лаборатории)

4.1 Ознакомиться с теоретическим материалом по лабораторной работе.

4.2 Выполнить предложенные задания.

4.3 Продемонстрировать результаты выполнения предложенных заданий.

## 5 Контрольные вопросы

5.1 На какие части можно условно разделить цифровой фотоаппарат?

5.2 Какие бывают объективы? Каковы их характеристики?

5.3 От чего зависит качество цифровой фотографии?

5.4 Что используется для хранения цифровых фотографий?



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

5.5 Как подключить цифровой фотоаппарат к компьютеру? Для чего это делается?

## **6 Содержание отчета**

6.1 название работы;

6.2 цель работы;

6.3 краткий конспект теоретической части;

6.4 ответы на контрольные вопросы;

6.5 выводы к работе.

## **7 Список литературы**

7.1 Скотт Келби Цифровая фотография = The Digital Photography Book. — М.: «Вильямс», 2007

7.2 Джули Адэр Кинг Цифровая фотография для "чайников" = Digital Photography For Dummies. — 5-е изд. — М.: «Диалектика», 2007



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

## Практическое занятие 6 Работа с векторной графикой

### 1 Цель работы

- 1.1 Ознакомиться методами работы с векторной графикой.
- 1.2 Ознакомиться с векторными объектами и способами их обработки

### 2 Пояснение к работе

#### 2.1 Краткие теоретические сведения

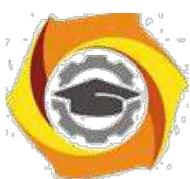
##### Понятие объекта в CorelDRAW

Любое изображение в векторном формате состоит из множества составляющих частей, которые редактируются независимо друг от друга. Главными кирпичиками, из которых составляется изображение, являются, так называемые, объекты. Понятие объекта является основным понятием в редакторе CorelDRAW.

Объектом называется элемент изображения: прямая, круг, прямоугольник, кривая, замкнутая кривая, многоугольник и другие.

Так как с помощью комбинации нескольких объектов можно создавать новый объект, то объекты могут иметь довольно замысловатый вид. Кроме того, CorelDRAW 10 может создавать группы объектов для дальнейшего редактирования группы как единого объекта.

Вне зависимости от внешнего вида, любой векторный объект CorelDRAW имеет ряд общих характеристик. Поясним это на простом примере (Рис. 1.1). Любой объект имеет некоторое количество точек или узлов, соединенных прямыми или кривыми линиями - сегментами. Координаты узлов и параметры сегментов определяют внешний вид объекта. Область внутри объекта можно закрасить или залить одним цветом, смесью цветов или узором. Эту область принято называть заливкой. Сегменты объекта образуют контур, который также имеет свой цвет. Толщину контура можно изменять. Различают замкнутые и разомкнутые контуры. У одного объекта не может быть различных заливок или соединительных линий различной толщины и разных цветов. Для создания сложных изображений требуется использовать множество объектов.



Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»

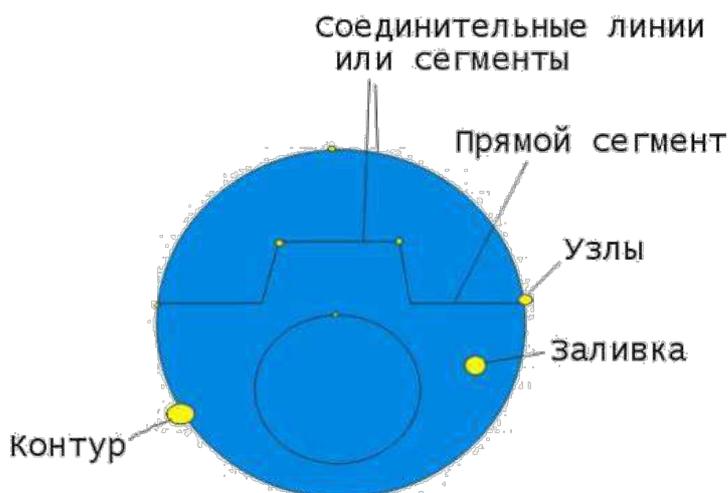


Рис. 1.1. Пример объекта

Одним из важных объектов CorelDRAW являются плавно изогнутые кривые, с помощью которых можно построить любой произвольный контур. Эти кривые называются кривыми Безье. Кроме положения начальной и конечной точки (то есть узлов кривой), внешний вид кривой определяется кривизной, то есть ее изогнутостью между двумя узлами. Кривизна определяется двумя параметрами кривой в каждом узле, которые графически представлены с помощью отрезков, выходящих из узлов. Эти отрезки называются манипуляторами кривизны (Рис. 1.2).

30

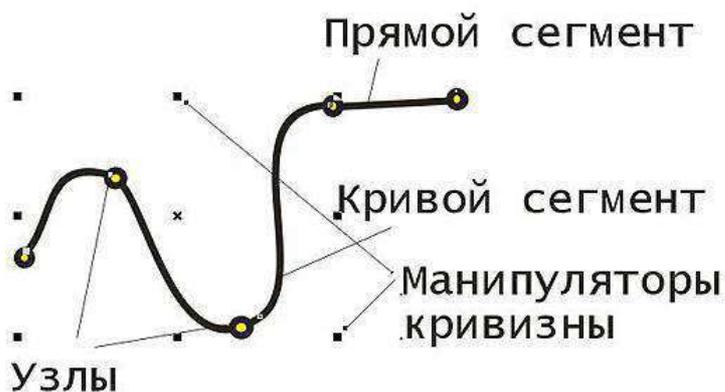


Рис. 1.2. Кривая Безье

Первым параметром, определяющим кривизну, является наклон кривой при входе в узел. Наклон манипулятора кривизны и показывает наклон кривой. Кривая как магнитом притягивается к манипуляторам кривизны. Вторым параметром является степень кривизны, то есть, то, как быстро при удалении от узла кривая расходуется с прямой проведенной через узел с тем же



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

наклоном. Степень кривизны определяется длиной манипулятора кривизны. Таким образом, координаты узлов, наклон и длина манипуляторов кривизны определяют внешний вид кривой Безье. Из множества кривых Безье можно составить любую кривую.

Работа над любой иллюстрацией в CorelDRAW заключается в создании объектов, их редактировании и расположении в нужных местах. При этом сначала создается приблизительная форма объектов, после чего форма уточняется путем добавления, удаления и перемещения узлов контура. После создания необходимой формы объекта задается цвет контура и выбирается заливка объекта. Создать в редакторе можно как стандартные фигуры: прямоугольники, эллипсы, многоугольники, автофигуры, спирали и решетки, так и произвольные фигуры, состоящие из прямых и кривых линий. Среди стандартных фигур есть достаточно сложные рисунки.

Богатые возможности форматирования позволяют прямо в редакторе создавать небольшие текстовые документы, оформленные рисунками. Применение оригинальных эффектов поможет создать красивый рисунок из простых объектов. Каждый рисунок, созданный в редакторе, состоит из одного или нескольких объектов, которые могут накладываться и полностью или частично закрывать друг друга. В качестве объектов могут использоваться растровые рисунки, подготовленные ранее с помощью любого редактора растровой графики и импортированные в CorelDRAW.

### **Основные приемы работы с CorelDRAW:**

21. Создание простых геометрических фигур или произвольных кривых и ломаных, замкнутых и разомкнутых. Вставка и форматирование текста.
22. Редактирование любого объекта, изменение цвета контура и заливки, изменение формы объекта.
23. Вставка готовых картинок или ранее созданных вами иллюстраций в документ. • Применение разнообразных художественных эффектов.
24. Размещение всех объектов в нужных местах, определение порядка взаимного перекрытия объектов.

Рабочее окно программы CorelDRAW не сильно отличается от окон других графических редакторов. Запустите CorelDRAW. Выберите вариант **New Graphic (Создать)**, щелкнув мышью на соответствующем рисунке (Создать) чтобы создать новый документ. Начальный диалог закроется, и вы сможете начать работу над иллюстрациями, но вначале уделите немного



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

времени настройке программы. Если рабочее окно занимает не весь экран, нажмите кнопку закрыть расположенную в заголовке



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

программы. Теперь, когда рабочее окно занимает весь экран, работать с CorelDRAW станет удобнее.

## 2.2 Перечень используемого оборудования

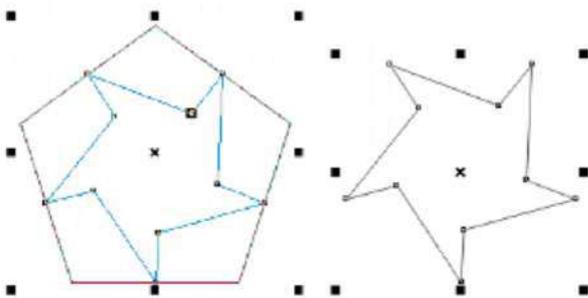
Персональный компьютер

### 26. Задание

3.1 При помощи секторов, дуг и эллипса, самостоятельно постройте веселую рожицу, изображенную ниже.



3.2 Нарисуйте показанные на рисунках многоугольники.



3.3 Нарисуйте автомобиль, с помощью инструмента "Кривая Безье".

3.4 Напишите произвольное слово с помощью инструмента "Фигурный текст", измените форму всех букв, перемещая узлы в разные стороны. На панели атрибутов можно добавлять новые узлы или удалять ненужные.

3.5 1. Нарисуйте эллипс. Заполните его красным цветом.

Нарисуйте на странице прямоугольник. Заполните его желтым цветом.

Выберите инструмент "Пипетка".

Щелкните на заливке красного эллипса.

На панели инструмента "Пипетка" выберите инструмент "Заполнитель".



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

24. Щелкните на желтом прямоугольнике. Прямоугольник закрасится красным цветом.

**6) Работа в кабинете (лаборатории)**

4.1 Ознакомиться с теоретическим материалом по лабораторной работе.

4.2 Выполнить предложенные задания.

4.3 Продемонстрировать результаты выполнения предложенных заданий.

**5 Контрольные вопросы**

5.1 Опишите основные приемы работы с CorelDRAW

5.2 Что называют объектом в CorelDRAW?

5.3 Зарисуйте и подпишите «Кривая Безье»

**6 Содержание отчета**

6.1 название работы;

6.2 цель работы;

6.3 краткий конспект теоретической части;

6.4 ответы на контрольные вопросы;

6.5 выводы к работе.

**7 Список литературы**

7.1 Скотт Келби Цифровая фотография = The Digital Photography Book. — М.: «Вильямс», 2007



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

## Практическое занятие 7 Работа с пиксельной графикой

### С Цель работы

5. Ознакомиться с основными понятиями - пиксел и разрешение
6. Научиться изменять размер изображения
7. Изучить понятие слоя

## 2 Пояснение к работе

### 2.1 Краткие теоретические сведения



#### 6. Пиксел и разрешение

**Цифровая фотография** - это изображение, состоящее из множества маленьких точек - "пикселей". Этих точек не видно на фотографии потому, что они очень малы и сливаются в линии.

Но если вы откроете цифровую фотографию на компьютере с помощью Adobe Photoshop или ACDSee (или другой программы просмотра изображений), и максимально увеличите картинку инструментом "Лупа" то перед вами окажется мозаика из разноцветных квадратиков.

Эти квадратик и называют "пикселями". Из них как из кирпичиков складывается оцифрованное изображение.

Чем "пиксели" мельче, а их количество больше, тем лучше качество изображения. Это свойство картинке называется "разрешением" и измеряется количеством "пикселей" на дюйм (dpi или ppi) или в "мегапикселах" - количестве точек (пикселей) во всей фотографии. Стандартным разрешением считается 300 dpi (пикселей на дюйм).

Если у Вас двух-трехмегапиксельный фотоаппарат, то Вы сможете получить только небольшие (10x15) фотографии в хорошем качестве.

Чтобы открыть в Adobe Photoshop файл с фотографией нужно выбрать меню "Файл"-**"Открыть"**, выбрать файл и нажать **"Открыть"**. Для удобства можно растянуть окно с фотографией шире.



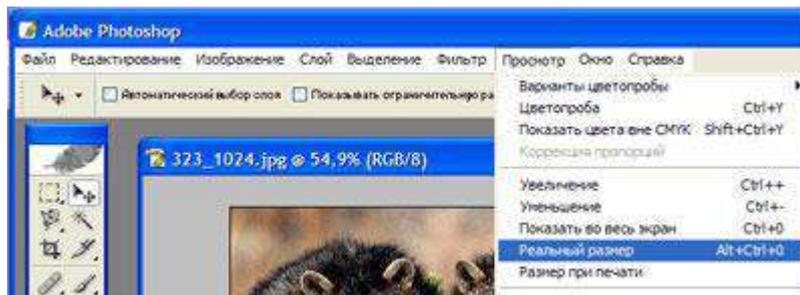
**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

### Размер изображения

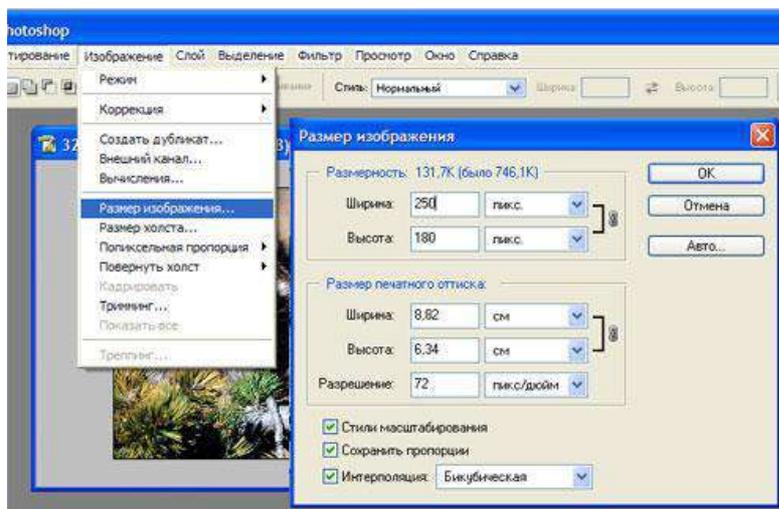
Когда говорят о размере фотографии, имеют в виду либо размер файла фотографии, либо размер самого изображения.

б. размер изображения может измеряться в сантиметрах, если речь идет о печати фотографий, либо в "пикселах", если обсуждается отображение фотографии на мониторе компьютера (при пересылке по электронной почте или публикации на сайт в интернете и т.п.).

Чтобы поменять масштаб отображения фотографии в Adobe Photoshop нужно выбрать меню "**Просмотр**", а затем "**Реальный размер**" - это фактический размер в пикселах, или "**Показать во весь экран**" - это масштаб, удобный для просмотра картинка полностью (если картинка очень большая - то ее масштаб уменьшается по размеру экрана), или "**Размер при печати**" - это размер, приближенный к тому, каким будет изображение после распечатки на принтере или в фотолаборатории.



Но меняя масштаб отображения фотографии Вы не меняете ее фактический размер. Регулировка масштаба служит только для удобства редактирования.



Изменение размера изображения осуществляется в окошке "Размер изображения": "**Изображение**" -- "**Размер изображения**". В этом окошке важны две строки: "ширина" и "высота". В них вводятся числа в пикселах, сантиметрах и т.д. Обратите внимание на пункт "**Сохранять пропорции**" - при редактировании фотографии он должен быть активирован ("галочка").

### Вращение изображения



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

8. сожалению, большинство фотографий неидеальны: одни слишком темные, в другие попали нежелательные объекты и т.д. Такие снимки можно подправить, отредактировать. Для этого используется программа Adobe Photoshop.

Очень часто фотограф не обращает внимание на линию горизонта и фотографии получаются с наклоном. Этот недостаток легко исправить - нужно немного повернуть изображение в обратную сторону.

**Панель инструментов**

|                         |                      |
|-------------------------|----------------------|
| Прямоугольное выделение | Перемещение          |
| Лассо                   | Волшебная палочка    |
| Кадрирование            | Раскройка            |
| Восстанавливающая кисть | Кисть                |
| Штамп                   | Архивная кисть       |
| Ластик                  | Градиент             |
| Размытие                | Осветлитель          |
| Выделение контура       | Горизонтальный текст |
| Перо                    | Прямоугольник        |
| Комментарий             | Пиветка              |
| Рука                    | Масштаб              |
| Основной цвет           | Фоновый цвет         |

помощью инструмента **"Прямоугольное выделение"** (пунктирный квадратик на панели инструментов) выделяется вся картинка. Затем щелчком правой кнопкой мыши внутри выделенного фрагмента в появившемся меню выбирается **"Свободное трансформирование"**. Тогда пунктирное выделение превратится в прямоугольник с "квадратиками". За эти квадратики можно "потянуть" и фотография исказится.

Поводите курсором рядом с угловыми "квадратиками" пока курсор не превратится в загнутую двустороннюю стрелочку. Затем нажмите левую кнопку мыши и потихоньку покрутите рисунок, пока он не выровняется так как нужно, и нажмите на клавиатуре кнопку "Enter". Чтобы убрать выделение щелкните курсором на невыделенной области рисунка или по свободному полю внутри окна с фотографией.

### **Кадрирование**

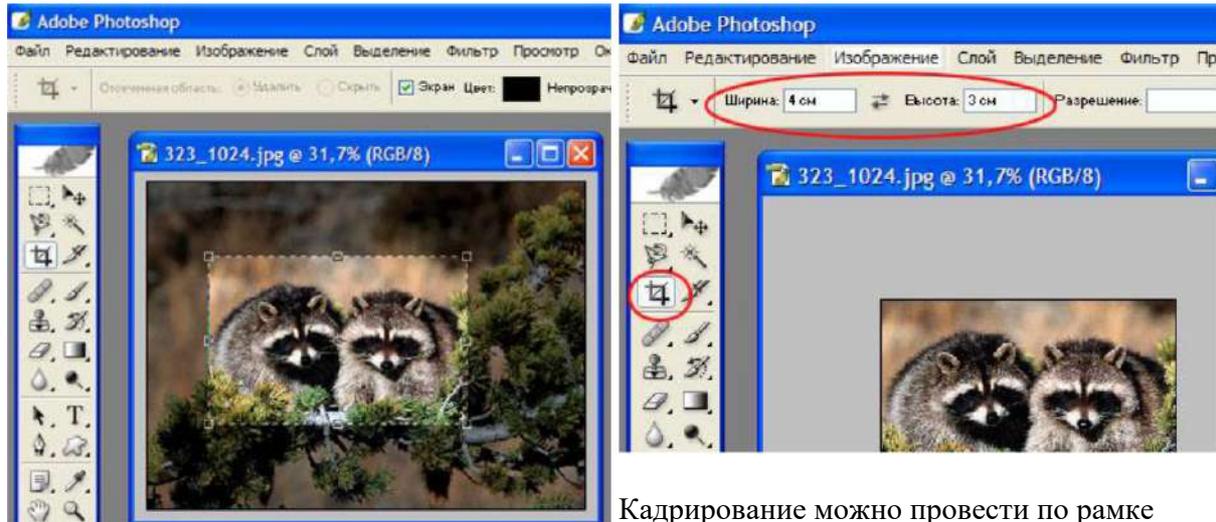
Иногда требуется кадрировать изображение, т.е. выделить фрагмент (Кадр), который нужен, а остальное - отрезать (например, после вращения остаются белые поля по углам).

Для этого используется инструмент **"Кадрирование"** (на панели инструментов). При выборе данного инструмента можно выбрать область



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

кадрирования на глаз, нажав левую кнопку мыши в одном из углов изображения и ведя курсор (не отпуская кнопки) по диагонали в противоположный угол. После этого можно подправить кадрируемую область. Потяните за один из квадратных маркеров, а после того как необходимая область выбрана, нажмите "Enter" или сделайте двойной щелчок мыши.



Кадрирование можно провести по рамке конкретных размеров, для этого в свойствах данного инструмента нужно установить значение ширины и высоты рамки кадра.

Для того, чтобы инструменты вращения, кадрирования и изменения размеров фотографии были лучше освоены, экспериментируйте с ними в разных сочетаниях.

Чтобы сохранить отредактированный файл - выберите пункт "**Файл**", затем "**Сохранить как**". При сохранении документа кроме имени можно выбрать и формат графического файла, нажав на выпадающее меню.

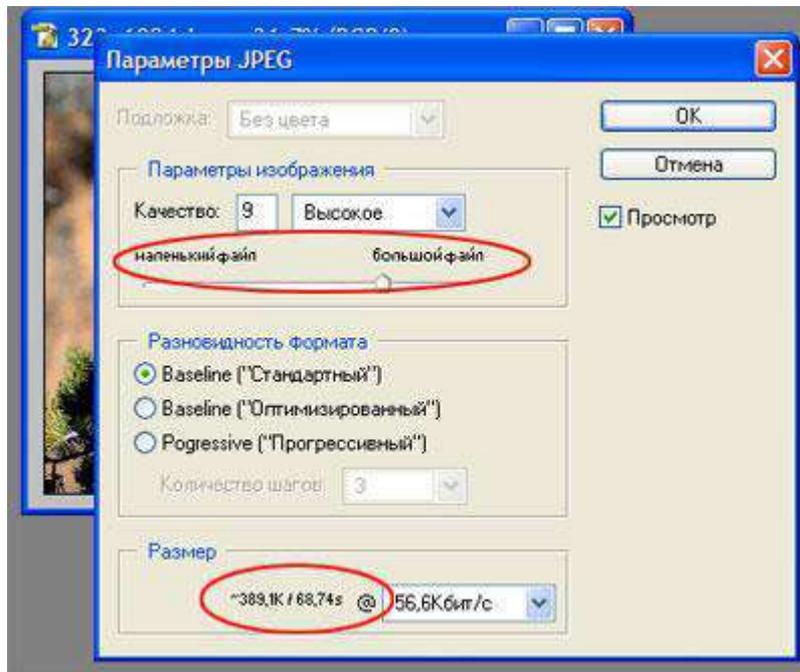
**JPEG** - Это формат, позволяющий создать минимальный по размерам файл с наименьшей потерей качества изображения. Поэтому он получил большую популярность - это стандарт для изображений в Интернете, в нем сохраняют многие фотоаппараты и фотографии в JPEG принимаются на печать в лабораториях. Минус у этого формата в том, что он при каждом сохранении документа немного искажает изображение.

Поэтому сохраняйте в нем фотографии только после всех этапов редактирования, перед тем как нести цифровой носитель в фотолабораторию. При сохранении в JPEG появляется окошко, где нужно задать качество



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

передачи изображения.



"Просмотр" должен быть активирован "галочкой". Регулируя "ползунком" качество изображения, изменяется размер изображения в поле "Размер". После достижения оптимального баланса нажмите "Ok".

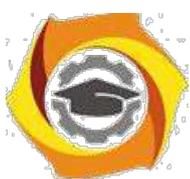
**TIFF** - Этот формат позволяет максимально точно передать изображение, он не "компрессует" файл, но вес такого файла очень велик и занимает много места на жестком диске, флэш-носителе и в памяти фотоаппарата. Это стандарт в полиграфии, где требуется четкая картинка с максимальным разрешением.

**PSD** - Это "родной" формат программы Adobe Photoshop. В нем сохраняется не только изображение, но и служебная информация программы (слои из которых состоит документ, скрытые фрагменты и т.д.). Используйте его для "промежуточного" сохранения работ.

**Слой** в программе Adobe Photoshop - это часть плоскости, на которой находится полное изображение или его отдельные составляющие. Эти плоскости могут быть прозрачными, непрозрачными или с переменными значениями прозрачности, их можно удалять, добавлять, перемещать сверху вниз и снизу вверх, объединять в единое целое все или некоторые.

Преимущество слоевой структуры изображения состоит в том, что можно редактировать изображение на каждом отдельном слое, не затрагивая его части на других слоях.

Для того, чтобы создать слой, необходимо выбрать меню **Слой - Новый - Слой**.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Палитра слоев (Рис. 2.1) является центром управления слоями. Для того, чтобы вывести палитру Слои на экран, следует нажать клавишу < F7 > или меню **Окно – Слои**.

10. палитре **Layers (Слои)** все слои размещаются по порядку, верхний слой в палитре является верхним слоем изображения. В каждый момент времени редактируется только один слой – активный.

**Активный слой** выделен в палитре слоев подсветкой и пиктограммой, изображающей кисть. Чтобы сделать слой активным, необходимо щелкнуть кнопкой мыши на его имени.

Слева на слое нарисован глаз, который показывает, что слой включен, т.е. то, что изображено на слое, входит в состав общего изображения. Если щелкнуть по глазу – изображение слоя не будет отображаться на общем изображении.

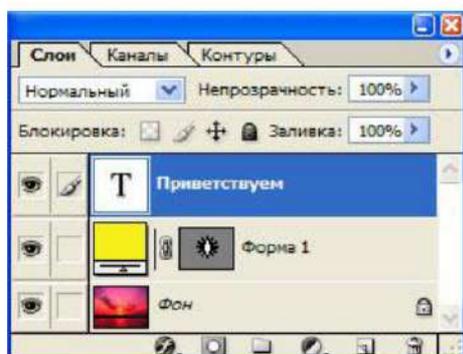


Рис. 2.1 Палитра Слои

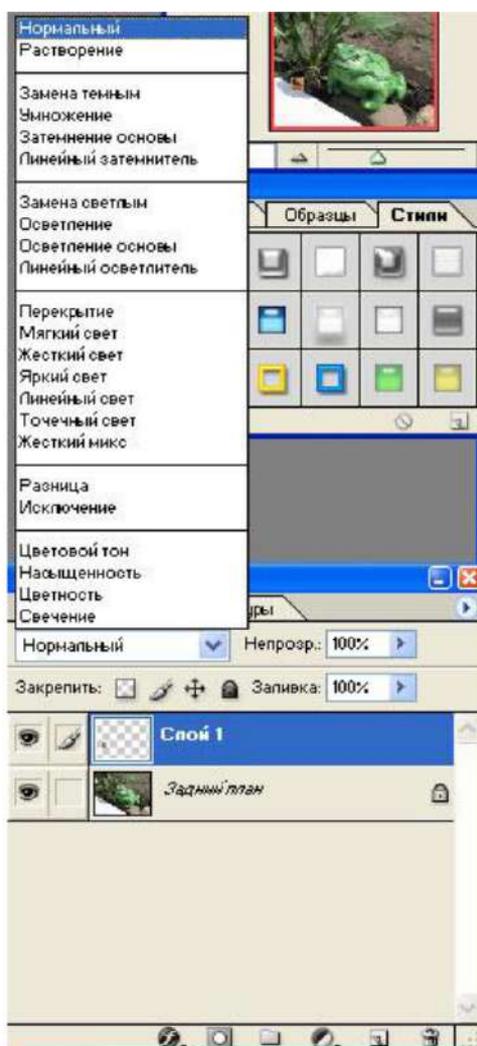


Рис. 2.2 Режимы наложения



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

**Слой можно перемещать** относительно друг друга, для этого необходимо перетащить на одну или несколько позиций вниз или вверх. Чтобы удалить слой – перетащить его в корзину, изображенную в нижнем правом углу палитры.

Нажав 2 раза на слове Слой... , можно переименовать его.

Параметр "**Непрозрачность**" на палитре Слои позволяет установить прозрачность слоя от 0 до 100.

**Режим наложения** (Рис. 2.2)(по умолчанию установлен **Нормальный**) позволяет устанавливать режим наложения одного слоя на другой, для этого нужно выбрать подходящий режим.

После того как редактирование изображения со слоями завершено, можно объединить слои. Меню **Слой - Объединить с предыдущим** (или **Объединить видимые** или **Выполнить сведение**)

Палитра **История** (меню **Окно - История**) используется для возврата действия на один шаг назад. Передвиньте ползунок на одну или более позиций вверх.

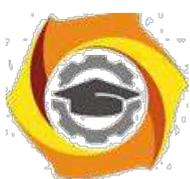
### **Инструменты выделения**

На панели инструментов существуют некоторые инструменты, которые необходимы для выделения объектов на рисунке, для того, чтобы их, например, вставить в другой файл.

14. **Инструмент "Прямоугольное выделение"** изучалось ранее.

15. **Инструмент "Лассо"**  (Рис. 2.3) имеет несколько видов: "Лассо", "Прямоугольное (полигональное) лассо", "Магнитное лассо".

Выделение с помощью Лассо образуется с помощью узловых точек (Рис. 2.4). Удаляются неправильно установленные точки кнопкой **Delit** на клавиатуре. Если нужно поставить дополнительную точку – нажмите левую кнопку мыши. Дойдя до начальной точки выделяемого контура, появится значок **лассо с ноликом**, после нажатия кнопки мыши контур замкнется (Рис.2.5).



Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»

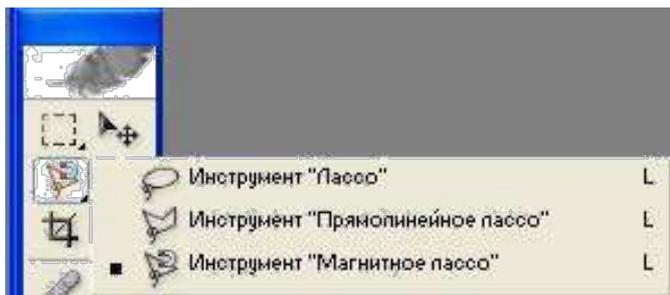


Рис. 2.3 Инструмент Лассо

Чтобы скопировать выделенный фрагмент используется меню **Редактирование – Копировать** . Чтобы вставить – меню **Редактирование – Вставить** .

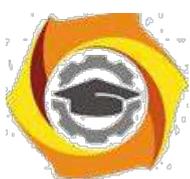
Перемещение готового рисунка на другой документ (или перемещение внутри одного документа) осуществляется с помощью инструмента **Перемещение**  Панели инструментов, нажав левой кнопкой мыши на рисунке и перетащив его.

14. Инструмент **"Волшебная палочка"**  используется для выделения одинаковых по цветовому тону частей изображения.



Рис. 2.4 Пример выделения лепестков ромашки с помощью

**Магнитного лассо.**



Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»

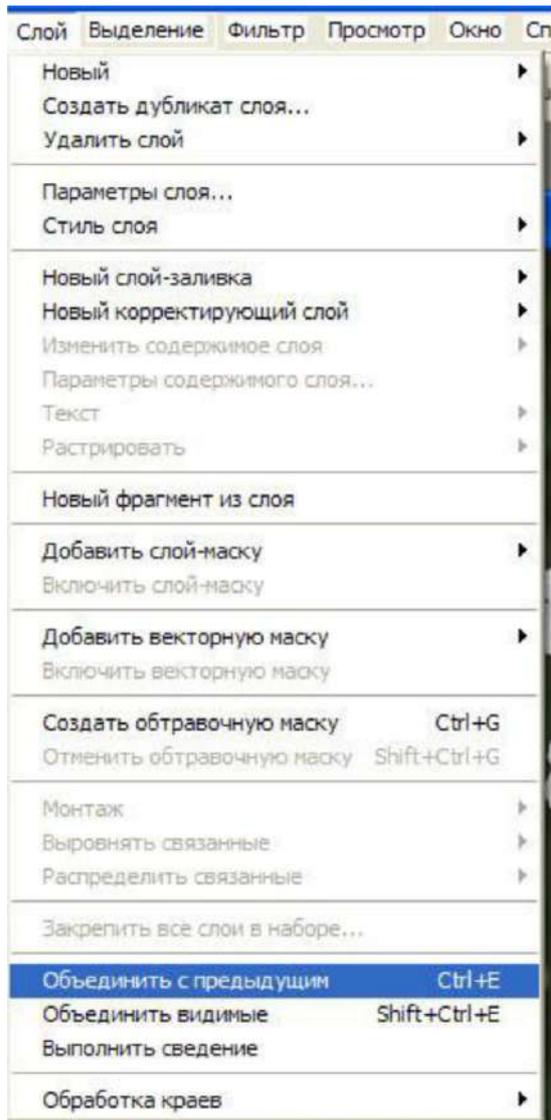


Рис. 2.5 Замкнутый контур выделения

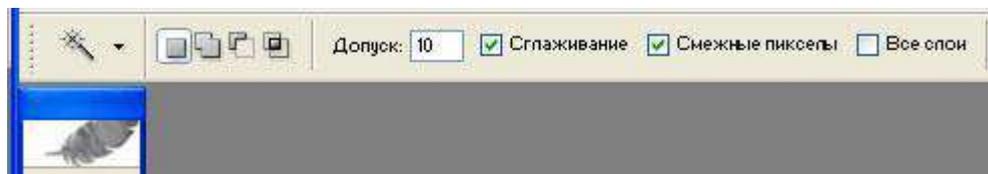


Рис. 2.6 Панель свойств инструмента "Волшебная палочка"

**Допуск** - определяет допустимы отклонения от взятого образца.

**Сглаживание** - Сглаживает переход на краях.

**Смежные пиксели** - выделяет олько смежные пиксели

**Все слои** - определяет цвет пиксела на основе данных со всех слоев



Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»



Рис. 2.7 Пример выделения голубого фона с допуском 10.

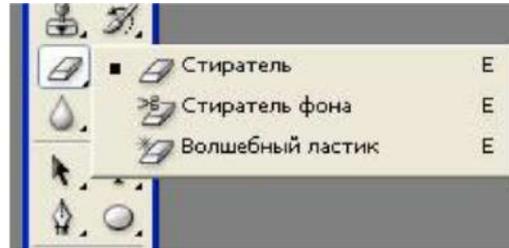


Рис.2.8 Инструмент Ластик

**2.3. Инструмент Ластик** используется для стирания ненужных частей, а ее производная "**Волшебный ластик**" (Рис.2.8) для удаления фона вокруг объекта. Используйте различную чувствительность на панели свойств ластика.

#### 2.4. Фильтры.

К каждому слою или целому изображению можно применить различные фильтры (рис. 2.9), которые имитируют, искажают, размывают, изменяют резкость, стилизуют, и другое. Для примера выберем изображение и добавим к нему фильтр (**Меню Фильтр** (рис. 2.9)), например **Имитация - Масляная живопись** (Рис. 2.10). Возможно изменить параметры фильтра, например размер кисти, резкость и т.д. для каждого свои.





Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»

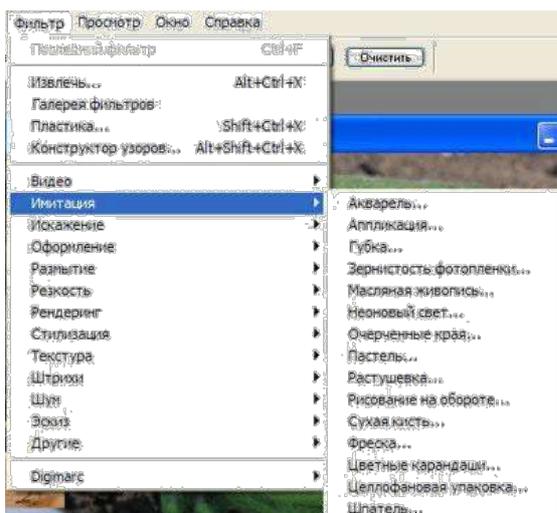


Рис. 2.10 Окно фильтра

Рис. 2.9 Меню Фильтр



Изображение без  
применения  
фильтров



Фильтр "Масляная  
живопись"



Фильтр "Целлофановая  
упаковка"



Фильтр "Рассеянный  
свет"

Рис. 2.11 Примеры применения различных фильтров

## 2.2 Перечень используемого оборудования

Персональный компьютер

## 4. Задание

3.1 Откройте один из файлов с фотографией, который имеется на компьютере. Измените размер изображения, установив значение ширины 500 пикселей. С помощью свободного трансформирования разверните изображение на 180 градусов. Обрежьте данную картинку с параметрами 4x3,



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

оставив в кадре какой-нибудь крупный элемент рисунка. Сохраните получившийся файл в формате JPEG в папке на рабочем столе.

3.2 Выберите три-четыре изображения, которые имеются на вашем компьютере и создайте коллаж. Используйте как целые изображения в качестве фона, так и вырезанные из других фрагменты. Также воспользуйтесь режимами наложения слоев и различными фильтрами.

### **5. Работа в кабинете (лаборатории)**

4.1 Ознакомиться с теоретическим материалом по лабораторной работе.

4.2 Выполнить предложенные задания.

4.3 Продемонстрировать результаты выполнения предложенных заданий.

### **5 Контрольные вопросы**

5.1 Что такое цифровая фотография?

5.2 Что значит кадрировать изображение?

5.3 Что такое слои в Adobe Photoshop?

5.4 Какие параметры у инструмента "**Волшебная палочка**"?

5.5 Какие фильтры вы использовали при выполнении второго задания?

### **6 Содержание отчета**

6.1 название работы;

6.2 цель работы;

6.3 краткий конспект теоретической части;

6.4 ответы на контрольные вопросы;

6.5 выводы к работе.

### **7 Список литературы**



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

7.1 Скотт Келби Цифровая фотография = The Digital Photography Book. — М.: «Вильямс», 2007

7.2 Джули Адэр Кинг Цифровая фотография для "чайников" = Digital Photography For Dummies. — 5-е изд. — М.: «Диалектика», 2007

7.3 Дик Мак-Келланд, Барбара Обермайер Photoshop 7 для «чайников».: Пер. с англ. — М.: Издательский дом «Вильямс», 2002. — 448 с.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

**Практическое занятие 8 Подготовка публикаций с помощью  
Microsoft Office Publisher**

### **1 Цель работы**

- 1.1 Научиться использовать заготовки публикаций, изменять заготовку для создания собственной публикации;
- 1.2 Научиться добавлять текст в публикацию, создавать колонки и продолжать статью на другой странице;
- 1.3 Научиться добавлять рисунки, управлять обтеканием текста вокруг рисунков.

### **2 Пояснение к работе**

#### 2.1 Краткие теоретические сведения

Microsoft Office Publisher 2003 – это программа, с помощью которой можно создавать публикации и WEB-сайты.

#### **Основные приемы работы по созданию публикации**

##### *5. Создание новой публикации на основе заготовки*

Запустите Microsoft Office Publisher 2003. Слева в области задач **Новая публикация** выберете пункт в группе **Создать из макета** категорию публикации и затем выберете тип публикации.

##### *2. Изменение цветовой схемы*

и области задач **Настройка публикации** выберете пункт **Цветовые схемы**. В списке **Применить цветовую схему** выберете цветовую схему.

##### *в Изменение шрифтовой схемы*

к области задач **Настройка публикации** выберете пункт **Шрифтовые схемы**. В списке **Применить шрифтовую схему** выберете требуемую схему шрифтов.

##### *6. Поиск конкретной области задач*



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Щелкните вверху области задач по значку со стрелкой, направленной вниз, затем выберите область задач. Если область задач не видна, выберите в меню **Вид** пункт **Область задач**.

***В Добавление нового текста***

Нажмите кнопку **Текстовое поле** на панели инструментов **Объекты**. Чтобы создать прямоугольник на странице, перетащите указатель. Наберите свой текст в пустом поле, окруженном круглыми маркерами.

***Перемещение текстового поля***

Установите указатель на любой части границы текстового поля. Когда указатель примет вид крестообразной стрелки, перетащите текстовое поле.

***К Изменение размеров текстового поля***

Щелкните по текстовому полю. Установите указатель на один из круглых маркеров. Когда указатель примет вид двунаправленной стрелки  $\phi$  В перетащите маркер.

***- Поворот текстового поля***

Щелкните по текстовому полю. Установите указатель на зеленый маркер поворота. Когда указатель примет вид кружка (&, перетащите маркер в требуемом направлении поворота текста.

***15. Изменение внешнего вида текста***

Выберите текст. В меню **Формат** выберите пункт:

11. **Шрифт**, чтобы изменить шрифт или его размер, цвет и стиль начертания;
12. **Абзац**, чтобы изменить выравнивание, отступы, междустрочный интервал, а также разрывы строк и страниц;



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

**Список**, чтобы установить или изменить характеристики оформления списка.

***В Изменение внешнего вида текстового поля***

Щелкните по текстовому полю. В меню **Формат** выберите пункт **Текстовое поле**, а затем в зависимости от требуемых изменений выполните следующие действия:

чтобы добавить цветную или фигурную границу,  
откройте вкладку **Цвета и линии**;  
чтобы добавить фоновую заливку, откройте вкладку **Цвета и линии**;  
чтобы изменить зазоры между текстом и границей текстового поля,  
откройте вкладку **Текстовое поле**.

***в Разбиение текстового поля на колонки***

Щелкните по текстовому полю. В меню **Формат** выберите пункт **Текстовое поле**, откройте вкладку **Текстовое поле** и установите параметр **Колонки**.

***12) Добавление в текстовые поля уведомлений «Продолжение»***

Щелкните текстовое поле. В меню **Формат** выберите пункт **Текстовое поле**, откройте вкладку **Текстовое поле** и затем установите один из флажков: **Включить «Продолжение на странице...»** или **Включить «Начало на странице...»**.

***9) Добавление нового рисунка или рамки***

Чтобы заменить существующий рисунок-заполнитель, щелкните правой кнопкой мыши по рисунку и выберите команду **Изменить рисунок**.

Если команда **Изменить рисунок** не видна, рисунок является частью группы. Повторно щелкайте по рисунку, пока вокруг него не появятся серые кружочки, после этого щелкните по рисунку правой кнопкой мыши.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Чтобы создать новую рамку и поместить в нее рисунок, нажмите кнопку **Рамка рисунка** на панели инструментов **Объекты**.

Чтобы добавить пустую рамку рисунка, выберите в меню **Вставка** пункт **Рисунок**, а затем выберите пункт **Пустая рамка ри-сунка**.

#### ***14. Перемещение рисунка***

Установите указатель на рисунке. Когда указатель примет вид крестообразной стрелки перетащите рисунок.

#### ***17. Изменение размеров рисунка***

Чтобы изменить размеры рисунка и сохранить первоначальные пропорции, щелкните рисунок, нажмите клавишу **Shift** и перетащите угловой маркер.

Чтобы изменить рисунок посредством удаления его ненуж-

ных частей, нажмите кнопку **Обрезка НФ** на панели инструментов **Рисунок**. Установите указатель на черный маркер обрезки и перетащите маркер.

#### ***13) Поворот рисунка***

Щелкните по рисунку. Установите указатель на зеленый

маркер поворота. Когда указатель примет вид кружка (С£, поверните рисунок.

- ***Управление режимами обтекания текстом***

Нажмите кнопку **Обтекание** на панели инструментов **Текст** **Рисунок**.

#### ***12. Вывод на экран панели инструментов Рисунок***

6 меню **Вид** выберите пункт **Панели инструментов**, затем выберите пункт **Рисунок**.

#### ***12. Добавление границы***

Нажмите кнопку **Формат рисунка *ty*** на панели инструмен-



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

тов **Рисунок** и откройте вкладку **Цвета и линии**.

**24. Перекрашивание рисунка**

Нажмите кнопку **Формат рисунка** на панели инструментов **Рисунок** и в зависимости от требуемых изменений выполните следующие действия:

В чтобы добавить цветной фон, откройте вкладку **Цвета и линии**.

**12.** группе **Заливка** выберите цвет и значение прозрачности (фон будет виден только на рисунках с прозрачными областями, в противном случае не закрашивайте всю рамку);

2) чтобы перекрасить рисунок, откройте вкладку **Рисунок**. Нажмите кнопку **Перекрасить** и выберите требуемый цвет.

- **Преобразование рисунка в подложку**

Нажмите кнопку **Формат рисунка** на панели инструментов **Рисунок** и откройте вкладку **Рисунок**. В группе **Рамка рисунка** выберите пункт **Подложка** в списке **Цвет**, затем нажмите кнопку **Перекрасить**. Выберите цвет.

**11.** меню **Расположение** выберите пункт **Порядок** и пункт **На задний план**, чтобы поместить подложку под текстом и рисунками, которые будут отображаться на ней.

**22. Использование кнопки Формат рисунка**

Если при нажатии кнопки **Формат рисунка** открывается диалоговое окно **Формат объекта**, значит, выбранный рисунок является частью группы. Чтобы отформатировать рисунок, входящий в группу, щелкните по рисунку 1 раз и далее повторно щелкайте

**2.2 Перечень используемого оборудования**

Персональный компьютер

**10) Задание**

**3.1** Создайте информационный буклет на любую тему, подберите свой текст и рисунки.

**3.2** Пример буклета «Замки Луары». Материалы для этой работы можно найти в папке Учебные материалы/ Картинки/Замки Луары

**13) Работа в кабинете (лаборатории)**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- 4.1 Ознакомиться с теоретическим материалом по лабораторной работе.
- 4.2 Выполнить предложенные задания.
- 4.3 Продемонстрировать результаты выполнения предложенных заданий.

## **5 Контрольные вопросы**

- 5.1 Какие виды документов можно подготовить в программе Microsoft Publisher? 5.2 Что такое таблица?
- 5.2 Чем текст в Publisher отличается от текста в текстовом процессоре?
- 5.3 Как добавить текст?
- 5.4 Какие в Publisher есть возможности по изменению внешнего вида текста?
- 5.5 Какие есть возможности по изменению вида текстовых полей?
- 5.6 Как разместить длинную статью на нескольких страницах?

## **6 Содержание отчета**

- 6.1 название работы;
- 6.2 цель работы;
- 6.3 краткий конспект теоретической части;
- 6.4 ответы на контрольные вопросы;
- 6.5 выводы к работе.

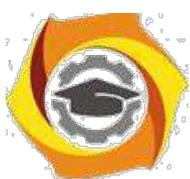
## **7 Список литературы**

- 7.1 Пакеты прикладных программ: Учеб. пособие для сред. проф. образования / Э.В. Фуфаев, Л.И. Фуфаева. – М.: Издательский центр «Академия», 2004. – 352 с.
- 7.2 Практикум по информатике: учеб. пособие / Под ред. Л.Г. Гагариной. Ч. В – М.: ИД «Форум»: Инфра – М, 2006. – 320 с.: ил. – (Профессиональное образование).

Методические рекомендации для магистров по организации и выполнению самостоятельной работы практике " Научно-исследовательская работа".

### *Тематика самостоятельной работы*

| № | Наименование работы |
|---|---------------------|
|   |                     |



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

|   |                                                                                                                                                                                                                         |
|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   |                                                                                                                                                                                                                         |
| 1 | Составление отчета по темам практики: Анализ методов исследования по теме диссертации.<br>Обоснование выбора методов исследования.                                                                                      |
| 2 | Составление отчета по темам практики: Проведение экспериментальной части исследований.<br>Обработка результатов экспериментов.                                                                                          |
| 3 | Составление отчета по темам практики: Анализ результатов теоретической части исследования.<br>Анализ результатов практической части исследования.<br>Оформление научных результатов в формате магистерской диссертации. |

*Самостоятельная работа № 1,2,3*

Составление отчета по НИР

ОГЛАВЛЕНИЕ

|                                                           |           |
|-----------------------------------------------------------|-----------|
| <b>ОГЛАВЛЕНИЕ .....</b>                                   | <b>2</b>  |
| <b>1. ОБЩИЕ ПОЛОЖЕНИЯ .....</b>                           | <b>3</b>  |
| <b>2. СТРУКТУРА ОТЧЕТА .....</b>                          | <b>3</b>  |
| <b>3. ТРЕБОВАНИЯ К СТРУКТУРНЫМ ЭЛЕМЕНТАМ ОТЧЕТА .....</b> | <b>3</b>  |
| <b>4. ПРАВИЛА ОФОРМЛЕНИЯ ОТЧЕТА .....</b>                 | <b>7</b>  |
| <b>ЗАДАНИЕ .....</b>                                      | <b>14</b> |



#### 4. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Отчет о НИР - научно-технический документ, который содержит систематизированные данные о научно-исследовательской работе, описывающий процесс или результаты научно-технического исследования, или состояние научно-технической проблемы.

1.2. При выполнении НИР, кроме заключительного отчета о работе в целом, могут быть составлены промежуточные отчеты по отдельным этапам НИР.

1.3. Отчет о НИР подлежит обязательному нормоконтролю в организации-исполнителе. При проведении нормоконтроля рекомендуется руководствоваться ГОСТ 2.111.

1.4. Ответственность за достоверность данных, содержащихся в отчете, и за соответствие его требованиям настоящего стандарта несет организация-исполнитель.

#### 6. СТРУКТУРА ОТЧЕТА

Структурными элементами отчета о НИР являются:

9. **титульный лист;**
10. список исполнителей;
11. **реферат;**
12. **содержание;**
13. перечень сокращений, условных обозначений, символов, единиц и тер-минов;
14. **введение;**
15. **основная часть;**
16. **заключение;**
17. список использованных источников;
18. приложения.

Обязательные структурные элементы выделены полужирным шрифтом. Остальные структурные элементы включают в отчет по смотрению исполнителя НИР с учетом требований разд.3 и 4.

#### 7. ТРЕБОВАНИЯ К СТРУКТУРНЫМ ЭЛЕМЕНТАМ ОТЧЕТА

##### 3.1. Титульный лист

3.1.1. Титульный лист является первой страницей отчета о НИР, служит источником информации, необходимой для обработки и поиска документа.

3.1.2. На титульном листе приводят следующие сведения:

4. наименование организации - исполнителя НИР;
5. индекс УДК;
6. коды Высших классификационных группировок Общесоюзного классификатора промышленной и сельскохозяйственной продукции для НИР, предшествующих постановке продукции на производство;
7. номера, идентифицирующие отчет;
8. грифы согласования и утверждения;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

9. наименование работы;
10. наименование отчета;
11. вид отчета (заключительный, промежуточный);
12. номер (шифр) темы;
13. должности, ученые степени, ученые звания, фамилии и инициалы руководи-телей организации - исполнителя НИР, руководителей НИР;
14. место и дата составления отчета.

3.1.3. Если отчет о НИР состоит из двух и более книг, каждая книга должна иметь свой титульный лист, соответствующий титульному листу первой книги и содержащий сведения, относящиеся к данной книге.

### 3.2. Список исполнителей

3.2.1. В список исполнителей должны быть включены фамилии и инициалы, должности, ученые степени, ученые звания руководителей НИР, ответственных исполнителей, исполнителей и соисполнителей, принимавших творческое участие в выполнении работы. В конце списка помещают подпись нормоконтролера.

3.2.2. Если отчет выполнен одним исполнителем, его должность, ученую степень, ученое звание, фамилию и инициалы, а также подпись нормоконтролера следует указывать на титульном листе отчета.

### 3.3. Реферат

3.3.1. Общие требования к реферату на отчет о НИР - по ГОСТ 7.9.

3.3.2. Реферат должен содержать:

21. сведения об объеме отчета, количестве иллюстраций, таблиц, приложений, количестве книг отчета, количестве использованных источников;
22. перечень ключевых слов;
23. текст реферата.

3.3.2.1. Перечень ключевых слов должен включать от 5 до 15 слов или словосочетаний из текста отчета, которые в наибольшей мере характеризуют его содержание и обеспечивают возможность информационного поиска. Ключевые слова приводятся в именительном падеже и печатаются прописными буквами в строку через запяты.

3.3.2.2. Текст реферата должен отражать;

11. объект исследования или разработки;
12. цель работы;
13. метод исследования и аппаратуру;
14. полученные результаты и их новизну;
15. основные конструктивные, технологические и технико-эксплуатационные характеристики;
16. степень внедрения;
17. рекомендации по внедрению или итоги внедрения результатов НИР;
18. область применения;
19. экономическую эффективность или значимость работы;
20. прогнозные предположения о развитии объекта исследования.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Если отчет не содержит сведений по какой-либо из перечисленных структурных частей реферата, то в тексте реферата она опускается, при этом последовательность изложения сохраняется.

#### 3.4. Содержание

3.4.1 Содержание включает введение, наименование всех разделов, подразделов, пунктов (если они имеют наименование) и заключение с указанием номеров страниц, с которых начинаются эти элементы отчета о НИР.

3.4.2. При составлении отчета, состоящего из двух и более книг, в каждой из них должно быть свое содержание. При этом в первой книге помещают содержание всего отчета с указанием номеров книг, в последующих - только содержание соответствующей книги. Допускается в первой книге вместо содержания последующих книг указывать только их наименование.

3.4.3. В отчете о НИР объемом не более 10 страниц содержание допускается не составлять.

3.5. Перечень сокращений, условных обозначений, символов, единиц и терминов

3.5.1. Принятые в отчете малораспространенные сокращения, условные обозначения, символы, единицы и специфические термины должны быть представлены в виде отдельного списка.

3.5.2. Если сокращения, условные обозначения, символы, единицы и термины повторяются в отчете менее трех раз, отдельный список не составляют, а расшифровку дают непосредственно в тексте отчета при первом упоминании.

#### 3.6. Введение

3.6.1. Введение должно содержать оценку современного состояния решаемой научно-технической проблемы, основание и исходные данные для разработки темы, обоснование необходимости проведения НИР, сведения о планируемом научно-техническом уровне разработки, о патентных исследованиях и выводы из них, сведения о метрологическом обеспечении НИР. Во введении должны быть показаны актуальность и новизна темы, связь данной работы с другими научно-исследовательскими работами.

3.6.2. Во введении промежуточного отчета по этапу НИР должны быть приведены цели и задачи исследований на этапе, их место в выполнении НИР в целом и обоснование выделения этапа.

3.6.3. Во введении заключительного отчета о НИР помещают перечень наименований всех подготовленных промежуточных отчетов по этапам и их инвентарные номера.

#### 3.7. Основная часть

3.7.1. Основная часть отчета должна содержать данные, отражающие существо, методику и основные результаты выполненной НИР.

3.7.2. Основная часть должна содержать:

11. выбор направления исследований, включающий обоснование выбора принятого направления исследования, методы решения задач и их сравнительную оценку, разработку общей методики проведения НИР;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- \* теоретические и (или) экспериментальные исследования, включающие определение характера и содержания теоретических исследований, методы исследований, методы расчета, обоснование необходимости проведения экспериментальных работ, принципы действия разработанных объектов, их характеристики, обоснование выбранного метрологического обеспечения работ, данные об объектах измерений, измеряемых величинах и средствах измерений, их метрологические характеристики, оценку правильности и экономичности выбора средств измерений (в том числе и нестандартизируемых) и методик выполнения измерений, сведения об их аттестации, оценку погрешности измерений, полученные экспериментальные данные;
- \* обобщение и оценку результатов исследований, включающие оценку полноты решения поставленной задачи и предложения по дальнейшим направлениям работ, оценку достоверности полученных результатов и их сравнение с аналогичными результатами отечественных и зарубежных работ, обоснование необходимости проведения дополнительных исследований, отрицательные результаты, приводящие к необходимости прекращения дальнейших исследований.

3.7.3. В зависимости от особенностей выполненной НИР основную часть излагают в виде текста, таблицы, сочетания иллюстраций и таблиц или сочетания текста, иллюстраций и таблиц.

Основную часть отчета следует делить на разделы и пункты. Разделы основной части могут делиться на пункты или на подразделы и пункты. Пункты, при необходимости, могут делиться на подпункты. Каждый пункт должен содержать законченную информацию.

3.7.4. Представление в отчете данных о свойствах веществ и материалов - по ГОСТ 7.54.

3.7.5. Единицы физических величин в отчете - по ГОСТ 8.417.

3.8. Заключение

Заключение должно содержать:

в краткие выводы по результатам выполненной НИР или отдельных ее этапов, оценку полноты решений поставленных задач, разработку рекомендаций и исходных данных по конкретному использованию результатов НИР, оценку технико-экономической эффективности внедрения. Если определение технико-экономической эффективности невозможно, необходимо указать народнохозяйственную, научную, социальную значимость работы;

воценку научно-технического уровня выполненной НИР в сравнении с лучшими достижениями в данной области.

3.9. Список использованных источников

Список должен содержать сведения об источниках, использованных при составлении отчета. Сведения об источниках приводятся в соответствии с требованиями ГОСТ 7.1.

3.10. Приложения

3.10.1. В приложения рекомендуется включать материалы, связанные с выполненной НИР, которые по каким-либо причинам не могут быть включены в основную часть.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

и приложения могут быть включены:

- 1) материалы, дополняющие отчет;
- 2) промежуточные математические доказательства, формулы и расчеты;
- 3) таблицы вспомогательных цифровых данных;
- 4) протоколы испытаний;
- 5) описание аппаратуры и приборов, применяемых при проведении экспериментов, измерений и испытаний;
- 6) заключение метрологической экспертизы;
- 7) инструкции, методики, описания алгоритмов и программ задач, решаемых ЭВМ, разработанных в процессе выполнения НИР; 8) иллюстрации вспомогательного характера;
- 9) копию технического задания на НИР, программы работ, договора или другого исходного документа для выполнения НИР;
- 10) протокол рассмотрения выполненной НИР на научно-техническом совете;
- 11) акты внедрения результатов НИР и др.

3.10.2. В приложения следует включать иллюстрации, таблицы, распечатки с ЭВМ, выполненные на листах формата А3.

3.10.3. В приложения к отчету о НИР, предшествующей постановке продукции на производство, должен быть включен проект технического задания на разработку (модернизацию) продукции или документ (заявку, протокол, контракт и др.), содержащий обоснованные технико-экономические требования к продукции.

3.10.4. В приложения к отчету о НИР, в составе которой предусмотрено проведение патентных исследований, должен быть включен отчет о патентных исследованиях, оформленный по ГОСТ 15.011, библиографический список публикаций и патентных документов, полученных в результате выполнения НИР, - по ГОСТ 7.1.

#### \* ПРАВИЛА ОФОРМЛЕНИЯ ОТЧЕТА

##### 4.1. Общие требования

4.1.1. Страницы текста отчета о НИР и включенные в отчет иллюстрации, таблицы и распечатки с ЭВМ должны соответствовать формату А4 по ГОСТ 9327. Допускается представлять иллюстрации, таблицы и распечатки с ЭВМ на листах формата А3.

4.1.2. Отчет о НИР должен быть выполнен машинописным способом или с применением печатающих и графических устройств вывода ЭВМ на одной стороне листа белой бумаги через полтора интервала. Для отчетов, выполненных на печатающих и графических устройствах вывода ЭВМ, высота букв и цифр должна быть не менее 1.8 мм.

Текст отчета следует печатать, соблюдая следующие размеры полей: левое - не менее 30 мм, правое - не менее 10 мм, верхнее - не менее 15 мм, нижнее - не менее 20 мм.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

4.1.3. Вне зависимости от способа выполнения отчета качество напечатанного текста и оформление иллюстраций, таблиц, распечаток с ЭВМ должно удовлетворять требованию их четкого воспроизведения (электрографическое копирование, микрофильмирование).

4.1.4. При выполнении отчета необходимо соблюдать равномерную плотность, контрастность и четкость изображения по всему отчету. В отчете должны быть четкие, не расплывшиеся линии, буквы, цифры и знаки. Все линии, буквы, цифры и знаки должны быть одинаково черными по всему отчету.

4.1.5. Вписывать в отпечатанный текст отчета отдельные слова, формулы, знаки допускается только черными чернилами или черной тушью, при этом плотность вписанного текста должна быть максимально приближена к плотности основного изображения.

4.1.6. Опечатки, опiski и графические неточности допускается исправлять подчисткой или закрашиванием белой краской и нанесением на том же месте исправленного изображения машинописным способом или от руки черными чернилами или черной тушью.

4.1.7. Фамилии, названия учреждений, организаций, фирм, название изделий и другие имена собственные в отчете приводят на языке оригинала. Допускается транслитерировать имена собственные и приводить названия организаций В переводе на языке отчета с добавлением (при первом упоминании) оригинального названия.

4.1.8. Сокращение русских слов и словосочетаний в отчете - по ГОСТ 7.12.

4.1.9. Наименования структурных элементов отчета «СПИСОК ИСПОЛНИТЕЛЕЙ», «РЕФЕРАТ», «СОДЕРЖАНИЕ», «ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ И ТЕРМИНОВ», «ВВЕДЕНИЕ», «ЗАКЛЮЧЕНИЕ», «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ» служат заголовками структурных элементов отчета.

4.1.10. Заголовки структурных элементов отчета и разделов основной части следует располагать в середине строки без точки в конце и печатать прописными буквами, не подчеркивая.

4.1.11. Заголовки подразделов и пунктов следует начинать с абзацного отступа и печатать с прописной буквы вразрядку, не подчеркивая, без точки в конце.

4.1.12. Если заголовок включает несколько предложений, их разделяют точками. Переносы слов в заголовках не допускаются.

4.1.13. Расстояние между заголовками структурных элементов отчета и разделов основной части и текстом должно быть не менее 3 - 4 интервалов.

4.1.14. Пункты и подпункты основной части следует начинать печатать с абзацного отступа.

#### 4.2. Нумерация страниц отчета

4.2.1. Страницы отчета следует нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всему тексту отчета. Номер страницы проставляют в правом верхнем углу без точки в конце.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

4.2.2. Титульный лист включают в общую нумерацию страниц отчета. Номер страницы на титульном листе не проставляют.

4.2.3. Иллюстрации и таблицы, расположенные на отдельных листах, и распечатки с ЭВМ включают в общую нумерацию страниц отчета. Иллюстрации, таблицы и распечатки с ЭВМ на листе формата А3 учитывают как одну страницу.

4.3. Нумерация разделов, подразделов, пунктов, подпунктов и книг отчета

4.3.1. Разделы, подразделы, пункты, подпункты и книги отчета следует нумеровать арабскими цифрами.

4.3.2. Разделы отчета должны иметь порядковую нумерацию в пределах основной части отчета и обозначаться арабскими цифрами с точкой, например, 1., 2., 3. и т.д.

4.3.3. Пункты должны иметь порядковую нумерацию в пределах каждого раздела или подраздела. Номер пункта включает номер раздела и порядковый номер подраздела или пункта, разделенные точкой, например 1.1., 1.2., 1.3. или 1.1.1., 1.1.2., 1.1.3. и т.д.

4.3.4. Номер подпункта включает номер раздела, подраздела, пункта и порядковый номер подпункта, разделенные точкой, например 1.1.1.1., 1.1.1.2. и т.д.

4.3.5. Если раздел или подраздел имеет только один пункт или пункт имеет один подпункт, то нумеровать пункт (подпункт) не следует.

4.3.6. Книги отчета должны иметь порядковую нумерацию в пределах всего отчета. Номер каждой книги следует проставлять арабскими цифрами на титульном листе под указанием вида отчета, например, «Книга 3».

4.4. Иллюстрации

4.4.1. Иллюстрации (чертежи, графики, схемы, диаграммы, фотоснимки) следует располагать в отчете непосредственно после текста, в котором они упоминаются впервые, или на следующей странице.

На все иллюстрации должны быть даны ссылки в отчете.

4.4.2. Чертежи, графики, диаграммы, схемы, помещаемые в отчете, должны соответствовать требованиям государственных стандартов ЕСКД.

4.4.3. Фотоснимки размером меньше формата А4 должны быть наклеены на стандартные листы белой бумаги.

4.4.4. Иллюстрации должны иметь название, которое помещают над иллюстрацией. При необходимости под иллюстрацией помещают поясняющие данные (подрисуночный текст).

Иллюстрация обозначается словом «Рис.», которое помещают после поясняющих данных.

4.4.5. Иллюстрации следует нумеровать арабскими цифрами порядковой нумерацией в пределах всего отчета.

4.4.6. Если в отчете только одна иллюстрация, ее нумеровать не следует и слово «Рис.» под ней не пишут.

4.4.7. Иллюстрацию следует выполнять на одной странице. Если иллюстрация не уместится на одной странице, можно переносить ее на другие страницы, при этом название иллюстрации помещают на первой странице, поясняющие данные - к каждой странице и под ними указывают «Рис. , лист ».



#### 4.5. Таблицы

4.5.1. Цифровой материал должен оформляться в виде таблиц.

4.5.2. Таблицу следует располагать в отчете непосредственно после текста, в котором она упоминается впервые, или на следующей странице. На все таблицы должны быть ссылки в отчете.

4.5.3. Таблицы следует нумеровать арабскими цифрами порядковой нумерацией в пределах всего отчета. Номер следует размещать в правом верхнем углу над заголовком таблицы после слова «Таблица».

4.5.4. Если в отчете одна таблица, ее не нумеруют и слово «Таблица» не пишут.

4.5.5. Оформление таблиц в отчете - по ГОСТ 2.105.

#### 4.6. Перечисления и примечания

4.6.1. Перечисления, при необходимости, могут быть приведены внутри пунктов или подпунктов. Перечисления следует нумеровать порядковой нумерацией арабскими цифрами со скобкой, например 1), 2), 3) и т.д., и печатать строчными буквами с абзацного отступа.

В пределах одного пункта или подпункта не допускается более одной группы перечислений.

4.6.2. Примечания следует помещать в отчете при необходимости пояснения содержания текста, таблицы или иллюстрации. Примечания размещают непосредственно после пункта, подпункта, таблицы, иллюстрации, к которым они относятся, и печатают с прописной буквы с абзацного отступа.

Слово «Примечание» следует печатать с прописной буквы с абзацного отступа вразрядку и не подчеркивать.

4.6.3. Одно примечание не нумеруют. Несколько примечаний следует нумеровать порядковой нумерацией арабскими цифрами точкой, например:  
Примечание.

.....

.....

Примечания:

1.....

2.....

#### 4.7. Формулы и уравнения

4.7.1. Пояснение значений символов и числовых коэффициентов следует приводить непосредственно под формулой в той же последовательности, в которой они даны в формуле. Значение каждого символа и числового коэффициента следует давать с новой строки. Первую строку пояснения начинают со слова «где» без двоеточия.

4.7.2. Уравнения и формулы следует выделять из текста в отдельную строку. Выше и ниже каждой формулы или уравнения должно быть оставлено не менее одной свободной строки. Если уравнение не умещается в одну строку, оно должно быть перенесено после знака равенства (=) или после знаков плюс (+), минус (-), умножения (x), деления (:), или других математических знаков.



4.7.3. Формулы в отчете следует нумеровать порядковой нумерацией в пределах всего отчета арабскими цифрами в круглых скобках в крайнем правом положении на строке.

Если в отчете только одна формула или уравнение, их не нумеруют. 4.8. Ссылки

4.8.1. Ссылки на источники следует указывать порядковым номером по списку источников, выделенным двумя косыми чертами.

Наряду с общим списком допускается приводить ссылки на источники в подстрочном примечании. Оформление ссылок - по ГОСТ 7.1.

4.8.2. Ссылки на разделы, подразделы, пункты, подпункты, иллюстрации, таблицы, формулы, уравнения, перечисления, приложения следует указывать их порядковым номером, например: «... в разд. 4», «... по п.3.3.4», «... в подпункте 2.3.4.1, перечисление 3», «... по формуле (3)», «... в уравнении (2)», «... на рис.8», «... в приложении б».

Если в отчете одна иллюстрация, одна таблица, одна формула, одно уравнение, одно приложение, следует при ссылках писать «на рисунке», «в таблице», «по формуле», «в уравнении», «в приложении».

4.9. Титульный лист

4.9.1. Титульный лист содержит реквизиты:

– наименование министерства (ведомства) или другого структурного образования, в систему которого входит организация-исполнитель, наименование организации (в том числе и сокращенное);

– индекс УДК, код ВКГ ОКП (для отчетов о НИР, предшествующих разработке и модернизации продукции) и номер государственной регистрации, предоставляемые организацией-исполнителем, а также надпись «Инв. №» - эти данные размещаются одно под другим;

– специальные отметки (при наличии в отчете численных данных о свойствах веществ и материалов в этой части проставляется аббревиатура ГСССД - государственная служба стандартных справочных данных);

– гриф согласования, гриф утверждения.

Гриф согласования состоит из слова «СОГЛАСОВАНО», должности с указанием наименования организации, ученой степени, ученого звания лица, согласовавшего отчет, его личной подписи, ее расшифровки, даты согласования, печати согласующей организации.

Если согласование проводилось письмом, следует указывать сокращенное наименование согласующей организации, исходящий номер и дату письма.

Гриф утверждения состоит из слова «УТВЕРЖДАЮ», должности с указанием наименования организации, ученой степени, ученого звания лица, утвердившего отчет, личной подписи, ее расшифровки и даты утверждения отчета. Здесь же проставляется печать организации, утвердившей отчет.

Подписи и даты подписания должны быть выполнены только черными чернилами или тушью.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Элементы даты приводят арабскими цифрами в одной строке в следующей последовательности: день месяца, месяц, год, например: дату 1 октября 2016 г.. следует оформлять 01.10.16;

и вид документа, строчными буквами с первой прописной - наименование зарегистрированной НИР, прописными буквами - наименование отчета, строчными буквами в круглых скобках - вид отчета (промежуточный или заключительный), строчными буквами с первой прописной номер книги отчета (если наименование зарегистрированной НИР совпадает с наименованием отчета, его печатают прописными буквами);

и шифр государственной научно-технической программы, шифр работы, присвоенный организацией-исполнителем;

и должности, ученые степени, ученые звания руководителей организации-исполнителя НИР, руководителей НИР, затем оставляют свободное поле для личных подписей и помещают инициалы и фамилии лиц, подписавших отчет, ниже личных подписей проставляют даты подписания (если на титульном листе не размещаются все необходимые подписи, то допускается переносить их на следующую страницу;

и город и год выпуска отчета.

#### 4.10. Список исполнителей

4.10.1. Фамилии и инициалы, должности, ученые степени, ученые звания в списке следует располагать столбцом. Слева указывают должности, ученые степени, ученые звания исполнителей и соисполнителей, затем оставляют свободное поле для подлинных подписей, справа указывают инициалы и фамилии исполнителей и соисполнителей. Возле каждой фамилии в скобках следует указывать номер раздела (подраздела) и фактическую часть работы, подготовленную конкретным исполнителем. Для соисполнителей следует также указывать наименование организации-соисполнителя.

4.11. Перечень сокращений, условных обозначений, символов, единиц и терминов

Перечень должен располагаться столбцом. Слева в алфавитном порядке приводят сокращения, условные обозначения, символы, единицы и термины, справа - их детальную расшифровку.

#### 4.12. Список использованных источников

Сведения об источниках следует располагать в порядке появления ссылок на источники в тексте отчета и нумеровать арабскими цифрами с

точкой. 4.13. Приложения

4.13.1. Приложения следует оформлять как продолжение отчета на его последующих страницах или в виде отдельной книги, располагая приложения в порядке появления на них ссылок в тексте отчета.

4.13.2. Каждое приложение должно начинаться с новой страницы и иметь содержательный заголовок, напечатанный прописными буквами. В правом верхнем углу над заголовком прописными буквами должно быть напечатано слово «ПРИЛОЖЕНИЕ».



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Если приложений в отчете более одного, их следует нумеровать арабскими цифрами порядковой нумерацией.

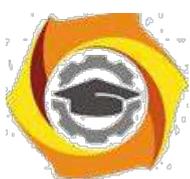
4.13.3. При оформлении приложений отдельной книгой отчета на титуль-ном листе под номером книги следует писать прописными буквами слово «ПРИЛОЖЕНИЯ».

4.13.4. При необходимости текст приложений может быть разделен на разделы, подразделы, пункты, которые следует нумеровать в пределах каждого приложения в соответствии с требованиями подраздела 4.3.

4.13.5. Имеющиеся в тексте приложения иллюстрации, таблицы, формулы б) уравнения следует нумеровать в пределах каждого приложения в соответствии с требованиями подразделов 4.4, 4.5 и 4.7.

4.13.6. Перечисления и примечания в тексте приложения оформляют и нумеруют в соответствии с требованиями подраздела 4.6.

4.13.7. Если в качестве приложения в отчете используется документ, имеющий самостоятельное значение и оформляемый согласно требованиям к документу данного вида, его вкладывают в отчет без изменений в оригинале. На титульном листе документа в правом верхнем углу печатают слово «ПРИЛОЖЕНИЕ» и проставляют его номер, а страницы, на которых размещен документ, включают в общую нумерацию страниц отчета.



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

ПРИЛОЖЕНИЕ  
*Пакет документов по практике*



Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет УГМК»  
Кафедра информационных технологий

ЗАДАНИЕ  
на практику

Студент: \_\_\_\_\_

Группа: \_\_\_\_\_

Вид практики: Производственная практика: Научно-исследовательская работа

Срок практики: \_\_\_\_\_

Место практики: \_\_\_\_\_

г. Верхняя Пышма  
2023 г.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**с СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИКИ**

Практика проводится на предприятии в соответствии с заданием на практику. По результатам практики студент предоставляет отчет по практике.

**Содержание практики**

| <b>№ п.п.</b> | <b>Разделы (этапы) практики</b>                                                                                                                         | <b>Всего дней</b> |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <b>1</b>      | <b>Выездная практика на предприятии</b>                                                                                                                 | <b>10</b>         |
| <b>1.1</b>    | Инструктаж по соблюдению правил внутреннего трудового распорядка, требований охраны труда, пожарной безопасности и ТБ                                   | 1                 |
| <b>1.2</b>    | Проведение научно-исследовательской работы на предприятии, направленной на управление и устойчивое развитие электрохозяйства в соответствии с ТЗ на ВКР | 7                 |
| <b>1.3</b>    | Подготовка отчета по практике                                                                                                                           | 2                 |

Задание на практику получил(а) \_\_\_\_\_

Подпись

Дата

ФИО

«Согласовано»

Заведующий кафедрой ИТ

Подпись

**Горбатов С.В.**

ФИО



## ИНСТРУКЦИЯ ПО ОФОРМЛЕНИЮ ДОКУМЕНТОВ ПО ПРАКТИКЕ

отчете по практике (приложение 2) должны содержаться следующие документы, подписанные руководителем практики и заверенные печатью организации:

- 6) формуляр отчета по практике (приложение 3);
- 7) отзыв руководителя практики от предприятия (приложение 4);
- 8) справка о проведении инструктажей обучающихся по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, правилами внутреннего трудового распорядка (приложение 1);

Приложение 1

### СПРАВКА

**7.проведении инструктажа обучающихся по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, правилами внутреннего трудового распорядка**

| Дата и время прохождения инструктажа | Наименование инструктажа (краткое содержание, характер требований) | Подпись, /расшифровка подписи, должность инструктора// | Подпись, /расшифровка подписи инструктируемого/ |
|--------------------------------------|--------------------------------------------------------------------|--------------------------------------------------------|-------------------------------------------------|
| 03.07.2017 г.<br>с 9-00 до 9-30      | Требования охраны труда:<br>1.<br>2.                               | Иванов И.А., мастер<br>АО «__»                         | Сидоров А.А.                                    |
| 03.07.2017 г.<br>с 9-30 до 10-00     | Требования техники безопасности:<br>1.<br>2.                       | Иванов И.А., мастер<br>АО «__»                         | Сидоров А.А.                                    |
| 03.07.2017 г.<br>с 10-00 до 10-30    | Требования пожарной безопасности:<br>1.<br>2.                      | Иванов И.А., мастер<br>АО «__»                         | Сидоров А.А.                                    |
| 03.07.2017 г.<br>с 10-30 до 11-00    | Правилами внутреннего трудового распорядка:<br>1.<br>2..           | Петров А.С. специалист кадрового управления АО<br>«__» | Сидоров А.А.                                    |

Руководитель практики от предприятия: \_\_\_\_\_ И.О.Фамилия  
Руководитель по персоналу  
организации УГМК: \_\_\_\_\_ И.О.Фамилия

М.П.



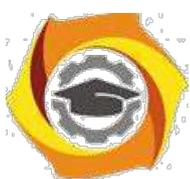
### Структура отчета по практике

8. *Титульный лист*
9. *Содержание* (включает названия разделов с указанием страниц, с которых они начинаются)
10. *Введение* (включает цели, задачи, сроки практики, краткую характеристику предприятия, технологических процессов, оборудования в соответствии с направлением подготовки)
11. *Основная часть* (включает описание выполненной на практике работы, с подробным раскрытием видов работ, оборудования, приборов, инструментов, технологий и т.д.)
12. *Заключение* (включает выводы, рациональные предложения, инновации сделанные во время прохождения практики)
13. *Список использованной литературы и нормативных документов*
14. *Приложения* (содержат вспомогательный материал, не включенный в основную часть отчета: таблицы, схемы, заполненные формы отчетности и т.д.)

*Все разделы Отчета по практике должны быть логически связаны между собой.*

*Оформление отчета должно быть в соответствии с:*

ГОСТ 7.32-2017 - «Система стандартов по информации, библиотечному и издательскому делу. Отчёт о научно-исследовательской работе. Структура и правила оформления».



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

ТИТУЛЬНЫЙ ЛИСТ



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет УГМК»  
Кафедра информационных технологий

**Отчет  
по практике**

---

Наименование практики

**Выполнил:**

|         |        |       |         |
|---------|--------|-------|---------|
| _____   | Ф.И.О. | _____ | Подпись |
| Студент | _____  | _____ | _____   |
| Группа  | _____  | _____ | _____   |

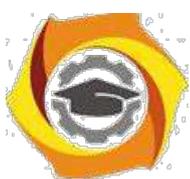
**Руководитель практики:**

|       |        |       |         |
|-------|--------|-------|---------|
| _____ | Ф.И.О. | _____ | Подпись |
|-------|--------|-------|---------|

**Руководитель практики от кафедры:**

|               |        |       |         |
|---------------|--------|-------|---------|
| Горбатов С.В. | _____  | _____ | Подпись |
|               | Ф.И.О. |       |         |

**Оценка:** \_\_\_\_\_



**ФОРМУЛЯР ОТЧЕТА ПО ПРАКТИКЕ**

\_\_\_\_\_  
*Название практики*

1. Студент \_\_\_\_\_

*ФИО*

проходил(а) практику с «\_\_» \_\_\_\_\_ 20\_\_ по «\_\_» \_\_\_\_\_ 2017  
в \_\_\_\_\_

\_\_\_\_\_  
*Название организации, адрес*

ФИО руководителя организации (объекта практики) \_\_\_\_\_

2. За время \_\_\_\_\_

\_\_\_\_\_ были выполнены следующие вид работ: \_\_\_\_\_ (в соответствии с заданием на  
практику)

3. Заключение руководителя практики от организации:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Руководитель практики от предприятия: \_\_\_\_\_

*ФИО*

*Подпись*

Заключение руководителя практики от ТУ УГМК:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Общая оценка по практике: \_\_\_\_\_

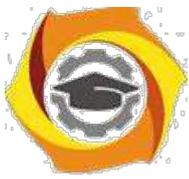
Руководитель практики от ТУ УГМК: \_\_\_\_\_

Горбатов С.В.

*ФИО*

*Подпись*





## Объектно-ориентированное программирование Методические указания к практическим занятиям

### Практическое занятие №1

#### Составление программ с инкапсуляцией полей и с перегрузкой (полиморфизмом) методов

##### 1. Цель занятия:

Выработать умения и навыки по составлению программ с инкапсулированными данными (инкапсуляция) и перегруженными методами (полиморфизм).

##### 2. Рекомендации:

Изучить материалы лекций №№4,5.

#### Краткая теория

##### Классы. «Перегруженные» функции.

ООП позволяет разбить задачу на ряд самостоятельных связанных между собой подзадач. Каждая подзадача содержит коды и данные, относящиеся к объекту, что упрощает решение задачи в целом и позволяет решать задачи большего объема.

Понятие объекта тесно связано с понятием класса.

Класс – это дальнейшее развитие понятия структуры. Он позволяет создавать новые типы и определять функции, манипулирующие с этими типами. По сути, класс это тип.

Объект - это представитель определенного класса. В общем, виде объект можно рассматривать как переменную, определенную программистом.

ООП использует механизмы инкапсуляции, полиморфизма и наследования.

Инкапсуляция позволяет создавать объекты - данные, процедуры и функции, манипулирующие с этими данными. Данные, доступные для использования внутри объекта - `private`, данные доступные извне - `public`.

Полиморфизм позволяет одно имя функции использовать для решения разных задач (общих для класса действий). В зависимости от данных выполняются те или иные действия.

Наследование позволяет одному объекту наследовать свойства другого объекта, т.е. порожденный класс наследует свойства родительского класса и добавляет собственные свойства.

Класс используется для создания объектов. Основная форма имеет вид:

```
class имя класса
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

{  
закрытые функции и переменные



public:

открытые функции, функции-члены и переменные

}

список объектов;//не является обязательным

Закрытые функции и переменные это члены (members) класса, которые до-ступны только для других членов этого класса. Открытые функции и перемен-ные доступны для любой части программы, в которой находится класс. Функ-ции, объявленные внутри описания класса, называются функциями членами (member functions).

Для определения функций-членов используется форма:

тип имя класса :: имя функции-члена (параметры)

{

тело функции

}

Два двоеточия :: после имени класса называются операцией расширения области видимости (scope resolution operator).

Класс только определяет тип объектов, а сами объекты не задает (па-мять для их хранения не выделяется). Для создания объектов имя класса ис-пользуется как спецификатор типа данных. После создания объекта к откры-тым членам класса можно обращаться, используя операцию «точка».

Пример.

```
#include <iostream.h>
class class1 //объявлен класс с именем class1 {

 int a; //доступна для функций членов class1
 public:
 int kwadrat(int b); //функция член класса class1
};
int class1::kwadrat(int b) //определение функции «kwadrat()»-члена класса
class1
{
 a=b*b;
 return a;
}
void main()
{
 class1 c; int z,x;
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

x=3;

//создается объект «С» типа class1



```
z=c.kwadrat(x);
```

```
вывод квадрата числа 3
}
```

```
cout<<"\n"<<z<<"\n";//вычисление и
```

### «Перегруженные» функции

Две или более функции называются перегруженными, если они имеют одно и то же имя.

Обычно функции отличаются количеством и типом аргументов. Транслятор автоматически на основании количества или типов аргументов выберет правильный вариант.

Пример.

```
#include <iostream.h>
```

```
void k(int a); //прототип первой функции void
k(int a, float b); //прототип второй функции
```

```
void k(int a) //описание первой функции
{
 cout << a <<"\n";
}
```

```
void k(int a, float b) //описание второй функции {

 cout <<a<<"\n"<< b <<"\n";
}
```

```
main()
{
 k(4); //вызов первой функции
 k(5, 10.2); //вызов второй функции
 return 0;
}
```

## 12. Порядок выполнения задания



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первая задача; №по журналу +3 – вторая



задача; и № по журналу +5 – третья задача (если достигнуто окончание списка вариантов заданий, то перейти в его начало).

3.2. Составить программу по заданию.

3.3. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.

3.4. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.

## 20. Варианты заданий:

**Составить методы с одинаковыми названиями для решения следующих задач**

17. Найти среднее арифметическое положительных чисел, введенных с клавиатуры. Всего ввести  $N$  различных чисел.

18. Ввести с клавиатуры  $N$  чисел. Найти сумму тех из них, которые принадлежат интервалу  $(2;9)$ .

19. Для  $N$  введенных с клавиатуры чисел найти сумму положительных кратных 3.

20. Для арифметической прогрессии 4, 9, 14, 19... найти первые  $n$  членов этой прогрессии.

21. Найти сумму отрицательных значений функции  $Z = \sin(5-x)/\cos(x-2)$  для  $x$ , изменяющегося на отрезке

$[-5,12]$  с шагом 1.

- Найти среднее арифметическое отрицательных чисел, введенных с клавиатуры. Всего ввести  $N$  различных чисел.

- Найти среднее арифметическое чисел, принадлежащих отрезку  $[2,184]$ , кратных 2 и введенных с клавиатуры. Всего ввести  $N$  различных чисел.

8. Найти сумму значений функции, больших 2  $Z = \sin(1/x) + 5\cos(1/(x-3)) + x$  для  $x$ , изменяющегося на отрезке  $[-3,8]$  с шагом 1.

9. Найти  $n$  членов последовательности  $x_1 = x_2 = x_3 = 1; x_k = x_{k-1} + x_{k-3}$ .

- Вычислить последовательность  $N$  чисел  $A_0 = x, A_1 = 2, A_k = A_{k-1} - A_{k-2}$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Для  $x_1 = 0,3$  и  $x_2 = -0,3$  найти  $x_k = k + \sin(x_k - 2)$  для  $k$ , изменяющегося следующим образом:  $k = 3, 4, \dots, 14$ .



и Составить таблицу перевода дюймов в сантиметры для расстояний от 1 до 13 дюймов с шагом 1.

и Вывести на печать значения функции, меньшие 2,  $Z = \sin(1/x) + 5\cos(x-3) + x$  для  $x$ , изменяющегося на отрезке  $[-7,4]$  с шагом 1.

14. Напечатать таблицу значений функции  $Y = \operatorname{tg}(x/b) + x/(b-2)$  для  $x$ , изменяющегося от 0 до 10 с шагом 1 ( $b$  - произвольное число).

её Вычислить  $N$ -ый член последовательности  $x_k = x_{k-2} - x_{k-1}$ ,  $x_0 = 2,4$ ,  $x_1 = 3,8$ .

Составить таблицу перевода суток (от 1 до 7) в часы, минуты, секунды.

Вычислить  $N$ -ый член последовательности

$$x_k = x_{k-1} + (2/3)x_{k-2} + 1, \quad x_1 = -1, \quad x_2 = 1,38.$$

В Напечатать значения функции

21.  $= 1/(x-2) + 1/(x-5) + \ln(12,8-X)$  для  $x$ , изменяющегося на отрезке  $[-4,14]$  с шагом 1.

Вывести на печать отрицательные значения функции

$z = \sin(5-x)/\cos(x-2)$  для  $x$ , изменяющегося на отрезке  $[-6,13]$  с шагом 1 (учесть область допустимых значений функции).

17. Из  $N$  введенных с клавиатуры чисел напечатать кратные 3 и меньшие 58.

18. Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(1,11)$  и являются четными.

19. Из  $N$  введенных с клавиатуры чисел напечатать положительные, кратные 3.

20. Вывести на печать значения функции

$z = \sin(x/(x-2))$ , находящиеся в интервале  $(-0,4;0,8)$  для  $x$ , изменяющегося от 8 до -6 с шагом 1.

22) Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(2;9)$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

23) Для геометрической прогрессии 2, 6, 18, 54, 162 ... определить первые  $n$  членов этой прогрессии.



В Ввести с клавиатуры N чисел. Напечатать те из них, которые не принадлежат интервалу (1;5).

27. Найти n членов последовательности  $x_1 = x_2 = x_3 = 1$ ;  $x_k = x_{k-1} - 2x_{k-3}$ .

Вычислить последовательность N чисел  $A_0 = x$ ,

$A_1 = 2$ ,  $A_k = A_{k-1} + A_{k-2}$

Составить таблицу перевода килограммов (от 1 до 13) в граммы с шагом

87 Найти сумму значений функции  $Y = \cos(x/A) + x/(A-2)$  для x, изменяющегося от 2 до 13 с шагом 1 (A - произвольное число).

## Практическое занятие №2

### Создание приложений с различными видами наследования

#### 1. Цель занятия:

Выработать умения и навыки по составлению программ, содержащий классы, наследующие другие классы.

#### 2. Рекомендации:

Изучить материалы лекций №№4-6.

### Краткая теория

#### Наследование. Виртуальные функции. Указатели на объекты

Наследование - это механизм посредством которого один класс (производный или потомок) может наследовать свойства другого класса (базового или предка).

Базовый класс определяет все качества, которые являются общими для всех производных классов.

Пример:

```
//Базовый класс
```

```
class B {
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
int i;
```



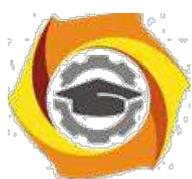
```
public:
 void set_i(int n);
 int get_i();
};
//Производный класс D
class D : public B
{
 int j;
public:
 void set_j(int n);
 int mul();
};
```

После имени класса D стоит двоеточие, за которым стоит ключевое слово public и имя класса B. Это означает, что класс D будет наследовать все компоненты класса B. Само ключевое слово public информирует компилятор о том, что т.к. класс B будет наследоваться, то все открытые элементы базового класса будут открытыми элементами производного класса. Однако все закрытые элементы базового класса остаются закрытыми.

Пример:

12. Простой пример наследования.

```
#include <iostream.h>
/ Задание базового класса
class base {
 int i;
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

public:

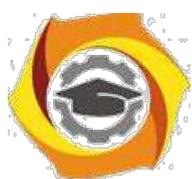


```
void set_i(int n);
int get_i();
};

/ Задание производного класса
class derived : public base
{
 int j;
 public:
 void set_j(int n);
 int mul();
};

/ Установка значения i в базовом классе
void base::set_i(int n)
{
 i = n;
}

/ Возврат значения i в базовом классе
int base::get_i()
{
 return i;
}
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

}



В Установка значения  $j$  в производном классе

```
void derived::set_j(int n)
{
 j = n;
}
```

\endash Возврат значения  $i$  из `base` и, одновременно,  $j$  из `derived`

```
int derived::mul()
{
 24. производный класс может вызывать функции-члены базового
 класса
 return j * get_i();
}
```

```
main()
{
 derived ob;

 ob.set_i(10); // загрузка i в base
 ob.set_j(4); // загрузка j в derived

 cout << ob.mul(); // вывод числа 40
}
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

```
return 0;
```

```
}
```



При определении `mul()` вызывается функция `get_i()`- базового класса `B`, а не производного `D`, что указывает на то, что открытые члены базового класса становятся открытыми членами производного. Но в функции `mul()` вместо прямого доступа к `i`, необходимо вызывать `get_i()`, потому что закрытые члены базового класса(`i`) остаются закрытыми для производных классов.

Виртуальные функции.

Проблема: как будет вызываться функция производного класса, имеющая та-кое же название, что функция базового класса.

Пример.

```
#include <stdio.h>

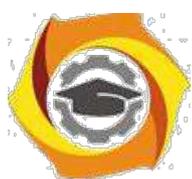
class base
{
public:
 int i;

 base(int x); //конструктор

 void func()
 {
 printf("Базовая функция %d",i);

 return;
 };
};

//текст конструктора
base::base(int x)
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

{



Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»

13

```
i=x;

return;

};

class der1: public base
{
public:
 der1(int x) :base(x) {};//конструктор
 void func()
 {
 printf("Функция из производного класса %d",
 i*i); return;
 }
};

main()
{
 base * pc; pc->func(); return 0;
 base ob(2);
 der1 ob1(2);
 pc=&ob;
 pc->func();
 pc=&ob1;
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

```
//указатель на базовый класс
//создать экземпляр объекта производного класса
//указатель на объект базового класса
//вызов функции базового класса
//указатель на объект производного класса
//попытка вызова функции производного класса
}
```



На первый взгляд кажется, что в первом случае будет вызываться функция базового класса, а во втором функция производного. Однако при проверке Вы легко убедитесь, что и в том и в другом случае будет вызвана функция базового класса. При этом, компилятору трудно «понять», какую реально функцию мы имеем в виду и он на стадии компилирования подставляет во всех тех случаях, где встречается имя `func()` адрес функции базового класса. Такой процесс установки адресов называется "ранним связыванием" или "статическим связыванием". Если же мы хотим, чтобы во втором случае, т.е. когда указатель `rc` указывал на производный класс вызывалась функция этого класса, ее еще в базовом классе следует указать как виртуальную. В нашем случае вместо строки `void func()` следует написать `virtual void func()`.

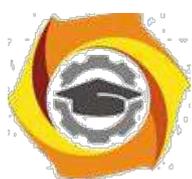
Это напоминает «перегрузку» функций, но «перегружаемые» функции отличаются друг от друга типом или аргументами, здесь же функции должны быть идентичны.

32. случае использования виртуальных функций адрес вызываемой функции будет определяться в процессе выполнения кода программы. Такой процесс называется "поздним связыванием" или "динамическим связыванием".

Пример:

```
#include <stdio.h>

class base {
public:
 int i;
 base(int x); //конструктор
 virtual void func()
 {
 printf("Базовая функция %d\n",i);
 return;
 };
};
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

};



```
//текст конструктора
base::base(int x)
{
 i=x;
 return;
};
class der1: public base {
public:
 der1(int x) :base(x) {}; //конструктор
 void func()
 {
 printf("Функция из производного класса %d\n", i*i);
 return;
 }
};
class der2: public base {
public:
 der2(int x) :base(x) {}; //конструктор
};

main()
{
 base * pc; base ob(2); der1 ob1(2);
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

//указатель  
на базовый  
класс

//создать  
экземпляр  
объекта  
базового  
класса

//создать  
экземпляр  
объекта  
производног  
о класса 1



```
der2 ob2(2); //создать экземпляр объекта производного класса 2

pc=&ob; //указатель на объект базового класса

pc->func(); //вызов функции базового класса

pc=&ob1; //указатель на объект производного класса 1

pc->func(); //попытка вызова функции производного класса

pc=&ob2; //указатель на объект производного класса 2

pc->func(); //попытка вызова функции производного класса

return 0;

}
```

Введен еще один производный класс. В нем функция `func()` не определена. В этом случае будет вызываться функция класса родителя. Т.е. появится строка: «Базовая функция 2». При этом, принцип прост: если Вы хотите, чтобы вызывалась функция родительского класса, не определяйте ее в производном. Еще один вопрос может возникнуть в связи с данным примером: как быть, если мы хотим, чтобы для класса объектов `der2` вызывалась функция класса `der1`. Решение очень просто - сделайте класс `der2` наследником не класса `base`, а класса `der1`.

22. производных классах функция, определенная в базовом классе как виртуальная может определяться, а может и нет. Если Вы хотите, чтобы во всех производных классах обязательно была определена виртуальная функция, то в базовом классе ее надо определить следующим образом:

```
virtual void func() = 0;
```

26. этом случае базовый класс называется агрегатным и от него нельзя будет создавать экземпляры объектов, зато во всех производных классах



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

компилятор «потребуется» определить данную виртуальную функцию и, тем самым, уменьшить вероятность ошибок.



### № Порядок выполнения задания

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: № по журналу - первая задачи, решение которой производится в методе класса №1; № по журналу +3 – вторая задача, решение которой производится в методе класса №2; и № по журналу +5 – третья задача, решение которой производится в методе класса №3 (если достигнуто окончание списка вариантов заданий, то перейти в его начало).

3.2. Исследовать как меняется код программы в зависимости от последовательности наследования:

38.1,2,3;

39.2,1,3;

40.2,3,1;

41.3,2,1;

42.3,1,2;

43.1,3,2.

Отразить выводы в отчете.

3.3. Составить программу по заданию.

3.4. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.

3.5. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.

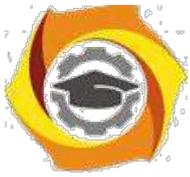
### 40. Варианты заданий:

Составить функции пользователя для следующих задач

31. Составить программу для перевода длины в метрах в длину в сантиметрах, определив функцию, выполняющую это преобразование и передав длину в метрах в качестве параметра.

32. Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив функцию, выполняющую это действие, и передавая массивы в качестве параметра.

33. Даны числа S, T. Получить с использованием функции пользователя  $F(T, -2S, 1.17) + F(2.2, T, S - T)$  где  $F(A, B, C) = (2A - B - \sin(C)) / (5 + C)$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

34. Составить программу перевода двоичной записи натурального числа в десятичную, описав соответствующую функцию с параметром. Перевод осуществлять для чисел, вводимых с клавиатуры. Признак конца ввода - число 0.

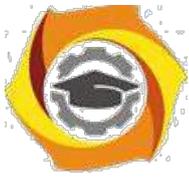


5. Даны числа  $S, T$ . Получить с использованием функции пользователя с параметрами  $G(1, \sin(S)) + 2G(T * S, 24) - G(5, -S)$ , где  $G(A, B) = (2A + B * B) / (A * B * 2 + B * 5)$ .
24. Составить программу для расчета значений гипотенузы треугольника, определив функцию, выполняющую этот расчет. Катеты передаются в качестве параметров.
25. Найти периметр десятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, которые передаются функции в качестве параметров из основной программы.
26. Найти периметр шестиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами. Координаты передаются функции в качестве параметров из основной программы.
27. Найти площадь пятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, и процедуру вычисления площади треугольника по трем сторонам. Описать функции с соответствующими формальными параметрами.
28. Составить программу вывода на экран всех натуральных чисел, не превосходящих  $N$  и делящихся на каждую из своих цифр. Описать соответствующую функцию, получающую из основной программы в качестве параметра натуральное число и возвращающую TRUE, если оно удовлетворяет указанному условию.
29. Используя подпрограмму - функцию, составить программу для нахождения максимального из трех чисел. Числа передаются функции в качестве параметров.
30. Используя подпрограмму - функцию, составить программу для печати знаков трех чисел, введенных с клавиатуры и передаваемых функции в качестве параметра.
31. Используя подпрограмму - функцию, составить программу для возведения чисел в целую положительную степень. Число передаются функции в качестве параметра из основной программы. Расчет вести для чисел, пока не будет введено число, равное 0.
32. Используя подпрограмму - функцию, составить программу для вычисления функции  $Z = (X1 + Y1) / (X1 * Y1)$ , где  $X1$  - первый корень уравнения  $X^2 - 4 * X - 1 = 0$ ;  $Y1$  - первый корень уравнения  $2 * Y^2 + A * Y - A^2 = 0$  ( $A$  - произвольное).
33. Задав функцию, вывести на печать средние арифметические двух массивов, введенных с клавиатуры. Массив передается функции в качестве параметра.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

34. Задав функцию, рассчитать и вывести на печать максимальные значения в трех парах чисел, вводимых с клавиатуры. Пара чисел передается функции в качестве параметра.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

19

- В Найти периметр восьмиугольника, координаты вершин которого заданы. Определить функцию вычисления расстояния между двумя точками, заданными своими координатами. Координаты передать функции в качестве параметров.
- В Даны четыре пары чисел. Получить с использованием функции пользователя наибольший общий делитель для каждой пары.
- В Даны числа A, B, C. Получить с использованием функции пользователя наименьшее значение. Числа передаются функции из основной программы в качестве параметров.
- В Даны числа  $x = 1, 2, \dots, N$ . Получить с использованием функции пользователя значения  $3 * P(x+3) * P(x)$  для заданных  $x$ , где  $P(x) = 10 * x^3 - 14 * x^2 + 12 * x - 2$ .
- В Составить программу для расчета значений катета треугольника, определив функцию, выполняющую этот расчет. Гипотенуза и второй катет передаются в качестве параметров.
- В Даны целые числа a, b, c, d. Проверить с использованием функции пользователя их четность. Число для проверки передается в функцию в качестве параметра из основной программы.
- В Для каждого из 10 введенных с клавиатуры чисел напечатать сообщение: является ли оно простым или нет, описав функцию логического типа, возвращающую значение “ИСТИНА”, если число, переданное ей в качестве параметра, является простым.
24. Даны числа S, T. Получить с использованием функции пользователя  $Y(T, S) = G(12, S) + G(T, S) - G(2S - 1, S * T)$ , где  $G(A, B) = (2 * A + B * B) / (A * B * 2 + B * 5)$ .
- OOO Определите функцию, определяющую, какой целой степенью числа 2 является ее аргумент (если число не является степенью двойки - выдать соответствующее сообщение).
- OOO Определите функцию, подсчитывающую сумму N первых элементов целочисленного массива A. N и массив A передать в качестве параметров.
- OOO Вычислить количество простых чисел, не превосходящих заданного N. Описать функцию логического типа, возвращающую значение true, если число простое и false в противном случае.
- OOO Используя подпрограмму - функцию с параметрами, составить программу для вычисления функции  $F(X, Y) = (2X^3 - 4 * X^2 + X + 1) / (9 * Y^3 + Y + 4) + 3 * Y^2 + 5 * Y$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

ООО Составить программу для перевода веса в граммах в вес в килограммах, определив функцию, выполняющую это преобразование. Вес в граммах передается функции в качестве параметра.



15. Даны числа S, T. Получить с использованием функции пользователя  $G(12, S)+G(T, S)-G(2S-1, S*T)$  где  $G(A, B) = (2*A+B*B)/(A*B*2+B*5)$ .

### Практическое занятие №3

#### Составление программ с инкапсуляцией полей и с перегрузкой (полиморфизмом) методов в Java

##### 1. Цель занятия:

Выработать умения и навыки по составлению программ с инкапсулированными данными (инкапсуляция) и перегруженными методами (полиморфизм).

##### 2. Рекомендации:

Изучить материалы лекций №№4,5.

#### Краткая теория

##### Совмещение методов на примере Java-программы

Язык Java позволяет создавать несколько методов с одинаковыми именами, но с разными списками параметров. Такая техника называется совмещением методов (**method overloading**).

33. в качестве примера приведена версия класса Pt, в которой совмещение методов использовано для определения альтернативного конструктора, который инициализирует координаты x и y значениями по умолчанию (-1).

```
class Pt
{
 int x, y;
 Pt(int x, int y)
 {
 this.x = x;
 this.y = y;
 }

 Pt()
 {
```



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

30. =  
-1; y=-  
1;  
}  
}



```
class PtCrAlt
{
 public static void main(String args[])
 {
 Pt ob = new Pt();
 System.out.println("x = " + ob.x + " y = " + ob.y);
 }
}
```

этом примере объект класса Pt создается не при вызове первого конструктора, как это было раньше, а с помощью второго конструктора без параметров. Вот результат работы этой программы:

$x = -1 y = -1$

Решение о том, какой конструктор нужно вызвать в том или ином случае, принимается в соответствии с количеством и типом параметров, указанных в операторе `new`. **Недопустимо объявлять в классе методы с одинаковыми именами и сигнатурами.** В сигнатуре метода не учитываются имена формальных параметров учитываются лишь их типы и количество.

### **this в конструкторах**

Очередной вариант класса Pt показывает, как, используя **this** и совмещение методов, можно строить одни конструкторы на основе других.

```
class Pt
{ int x, y;

 Pt(int x, int y)
 {
 this.x = x;
 this.y = y;
 }

 Pt()
 {
 this(-1, -1);
 }
}
```

35. этом примере второй конструктор для завершения инициализации объекта обращается к первому конструктору.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Методы, использующие совмещение имен, не обязательно должны быть конструкторами. В следующем примере в класс Pt добавлены два метода distance. Функция distance возвращает расстояние между двумя точками. Одно-



му из совмещенных методов в качестве параметров передаются координаты точки  $x$  и  $y$ , другому же эта информация передается в виде параметра-объекта класса  $Pt$ .

```
class Pt
{ double x, y;
 Pt(double x, double y)
 {
 this.x = x;
 this.y = y;
 }

 double distance (double x, double y)
 { double dx, dy;
 dx = this.x - x;
 dy = this.y - y;
 return Math.sqrt(dx*dx + dy*dy);
 }

 double distance(Pt ob)
 {
 return distance(ob.x, ob.y);
 }
}

class PtDist
{
 public static void main(String args[])
 {
 Pt ob1 = new Pt(0, 0);
 Pt ob2 = new Pt(30, 40);
 System.out.println("ob1 = " + ob1.x + ", " + ob1.y);
 System.out.println("ob2 = " + ob2.x + ", " + ob2.y);
 System.out.println("ob1.distance(p2) = " + ob1.distance(p2));
 System.out.println("ob1.distance(60, 80) = " + ob1.distance(60, 80));
 }
}
```

Во второй форме метода **distance** для получения результата вызывается его первая форма. Ниже приведен результат работы этой программы:

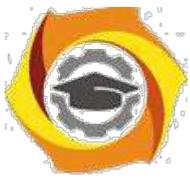
```
ob1 = 0, 0
ob2 = 30, 40
```



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

*ob1.distance(ob2) = 50.0*

*ob1.distance(60, 80) = 100.0*



## Инкапсулирование полей на примере Java-программы

Данные инкапсулируются в класс путем объявления переменных между открывающей и закрывающей фигурными скобками, выделяющими в определении класса его тело.

Эти переменные объявляются точно так же, как локальные переменные, которые надо объявлять вне методов, в том числе вне метода main.

```
class Pt
{
int x, y;
}
```

### Оператор new

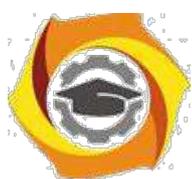
Оператор new создает экземпляр указанного класса и возвращает ссылку на вновь созданный объект.

```
Pt ob = new Pt();
```

Вы можете создать несколько ссылок на один и тот же объект.

Оператор точка используется для доступа к переменным и методам объекта.

```
class PtDva
{
 public static void main(String args[])
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

}



```
Pt ob1 = new Pt();
Pt ob2 = new Pt();
ob1.x = 10;
ob1.y = 20;
ob2.x = 42;
ob2.y = 99;
System.out.println("x = " + ob1.x + " y = " + ob1.y);
System.out.println("x = " + ob2.x + " y = " + ob2.y);
}
}
```

12.     этом примере снова использовался класс Pt, было создано два объекта этого класса, и их переменным x и y присвоены различные значения. Таким образом мы продемонстрировали, что переменные различных объектов независимы на самом деле.

Ниже приведен результат, полученный при выполнении этой программы.

=10y=20

=42y=99

## 17.    Порядок выполнения задания

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первая задача; №по журналу +3 – вторая задача; и №по журналу +5 – третья задача (если достигнуто окончание списка вариантов заданий, то перейти в его начало).

3.2. Составить программу по заданию.

3.3. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

3.4. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.



**17. Варианты заданий:**

**Составить методы с одинаковыми названиями для решения следующих задач**

11. Найти среднее арифметическое положительных чисел, введенных с клавиатуры. Всего ввести  $N$  различных чисел.

12. Ввести с клавиатуры  $N$  чисел. Найти сумму тех из них, которые принадлежат интервалу  $(2;9)$ .

13. Для  $N$  введенных с клавиатуры чисел найти сумму положительных кратных 3.

14. Для арифметической прогрессии 4, 9, 14, 19... найти первые  $n$  членов этой прогрессии.

15. Найти сумму отрицательных значений функции  $Z=\sin(5-x)/\cos(x-2)$  для  $x$ , изменяющегося на отрезке

$[-5,12]$  с шагом 1.

15. Найти среднее арифметическое отрицательных чисел, введенных с клавиатуры. Всего ввести  $N$  различных чисел.

16. Найти среднее арифметическое чисел, принадлежащих отрезку  $[2,184]$ , кратных 2 и введенных с клавиатуры. Всего ввести  $N$  различных чисел.

8. Найти сумму значений функции, больших 2  $Z=\sin(1/x)+5\cos(1/(x-3))+x$  для  $x$ , изменяющегося на отрезке  $[-3,8]$  с шагом 1.

9. Найти  $n$  членов последовательности  $x_1 = x_2 = x_3 = 1; x_k = x_{k-1} + x_{k-3}$ .

11. Вычислить последовательность  $N$  чисел  $A_0 = x, A_1 = 2, A_k = A_{k-1} - A_{k-2}$ .

12. Для  $x_1 = 0,3$  и  $x_2 = -0,3$  найти  $x_k = k + \sin(x_{k-2})$  для  $k$ , изменяющегося следующим образом:  $k = 3,4,\dots,14$ .

13. Составить таблицу перевода дюймов в сантиметры для расстояний от 1 до 13 дюймов с шагом 1.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

14. Вывести на печать значения функции, меньшие 2,  $Z = \sin(1/x) + 5\cos(x-3) + x$  для  $x$ , изменяющегося на отрезке  $[-7, 4]$  с шагом 1.

14. Напечатать таблицу значений функции  $Y = \operatorname{tg}(x/b) + x/(b-2)$  для  $x$ , изменяющегося от 0 до 10 с шагом 1 ( $b$  - произвольное число).



15. Вычислить  $N$ -ый член последовательности  $x_k = x_{k-2} - x_{k-1}$ ,  $x_0 = 2,4$   
 $x_1 = 3,8$ .

Составить таблицу перевода суток (от 1 до 7) в часы, минуты, секунды.

Вычислить  $N$ -ый член последовательности

$$x_k = x_{k-1} + (2/3)x_{k-2} + 1, \quad x_1 = -1, \quad x_2 = 1,38.$$

12. Напечатать значения функции

$9. = 1/(x-2) + 1/(x-5) + \ln(12,8-X)$  для  $x$ , изменяющегося на отрезке  $[-4,14]$  с шагом 1.

Вывести на печать отрицательные значения функции

$z = \sin(5-x)/\cos(x-2)$  для  $x$ , изменяющегося на отрезке  $[-6,13]$  с шагом 1 (учесть область допустимых значений функции).

И Из  $N$  введенных с клавиатуры чисел напечатать кратные 3 и меньшие 58.

И Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(1,11)$  и являются четными.

И Из  $N$  введенных с клавиатуры чисел напечатать положительные, кратные 3.

И Вывести на печать значения функции

$z = \sin(x/(x-2))$ , находящиеся в интервале  $(-0,4;0,8)$  для  $x$ , изменяющегося от 8 до -6 с шагом 1.

· Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(2;9)$ .

· Для геометрической прогрессии 2, 6, 18, 54, 162 ... определить первые  $n$  членов этой прогрессии.

· Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые не принадлежат интервалу  $(1;5)$ .

27. Найти  $n$  членов последовательности  $x_1 = x_2 = x_3 = 1$ ;  $x_k = x_{k-1} - 2x_{k-3}$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Вычислить последовательность  $N$  чисел  $A_0 = x$ ,  
 $A_1 = 2$ ,  $A_k = A_{k-1} + A_{k-2}$



Составить таблицу перевода килограммов (от 1 до 13) в граммы с шагом

11) Найти сумму значений функции  $Y = \cos(x/A) + x/(A-2)$  для  $x$ , изменяющегося от 2 до 13 с шагом 1 ( $A$  - произвольное число).

## Практическое занятие №4

### Составление программ с наследованием классов в Java

#### 1. Цель занятия:

Выработать умения и навыки по составлению программ, содержащих классы, наследующие другие классы.

#### 2. Рекомендации:

Изучить материалы лекций №№4-6.

### Краткая теория

#### Реализация множественного наследования на примере Java - программ

Класс может быть производным не только от одного базового класса, а и от многих. Этот случай называется множественным наследованием. Синтаксис описания множественного наследования похож на синтаксис простого наследования:

```
class A {

};
class B {

};
class C : public A, public B {

};
```

Базовые классы класса C перечислены после двоеточия в строке описания класса и разделены запятыми.



## Методы (функции) классов и множественное наследование

Рассмотрим пример множественного наследования. Пусть нам для некоторых служащих необходимо указать их образование в программе EMPLOY. Теперь предположим, что в другой программе у нас существует класс student, в котором указывается образование каждого студента. Тогда вместо изменения класса employee мы воспользуемся данными класса student с помощью множественного наследования.

и классе student содержатся сведения о школе или университете, которые закончил студент, и об уровне полученного им образования. Эти данные хранятся в строковом формате. Методы getedu() и putedu() позволяют нам ввести данные о студенте и просмотреть их.

Информация об образовании нужна нам не для всех служащих. Предположим, что нам не нужны записи об образовании рабочих, а необходимы только записи об ученых и менеджерах. Поэтому мы модифицируем классы manager и scientist так, что они будут яЭта маленькая программа показывает только взаимосвязь между классами:

```
class student
{
};

class employee
{
};

class manager : public employee, private student
{
};

class scientist : private employee, private student
{
};

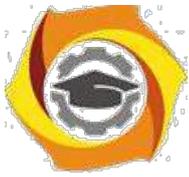
class laborer : public employee
{
};
```

Теперь мы рассмотрим эти классы более детально в листинге EMPMULT.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

// empmult.cpp



- множественное наследование

```
#include <iostream>

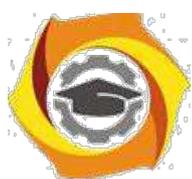
using namespace std;

const int LEN = 80; // максимальная длина имени
////////////////////////////////////

class student // сведения об образовании
{
private:
char school[LEN]; // название учебного
заведения char degree[LEN]; // уровень
образования public:
void getedu()
{
cout << " Введите название учебного заведения: ";
cin >> school;

cout << " Введите степень высшего образования\n"; // вляться производными клас-
сов employee и student.
cout << " (неполное высшее, бакалавр, магистр, кандидат наук): ";
cin >> degree;
}

void putedu() const
{
cout << "\n Учебное заведение: " << school;
cout << "\n Степень: " << degree;
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

};



Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»

30

////////////////////////////////////

```
class employee // некий сотрудник
{
private:
char name[LEN]; // имя сотрудника
unsigned long number; // номер
сотрудника public:
void getdata()
{
cout << "\n Введите фамилию: "; cin >> name;
cout << " Введите номер: "; cin >> number;
}
void putdata() const
{
cout << "\n Фамилия: " << name;
cout << "\n Номер: " << number;
}
};
```

////////////////////////////////////

```
class manager : private employee, private student // менеджер
{
private:
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

char title[LEN]; // должность сотрудника

double dues; // сумма взносов в гольф-

клуб public:



```
void getdata()
{
employee::getdata();
cout << " Введите должность: "; cin >> title;
cout << " Введите сумму взносов в гольф-клуб: "; cin >> dues;
student::getedu();
}

void putdata() const
{
employee::putdata();
cout << "\n Должность: " << title;
cout << "\n Сумма взносов в гольф-клуб: " << dues;
student::putedu();
}

};

////////////////////////////////////

class scientist : private employee, private student // ученый
{
private:
int pubs; // количество публикаций
public:
void getdata()
{
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

```
employee::getdata();
```

```
cout << " Введите количество публикаций: "; cin >> pubs;
```



```
student::getedu();
}

void putdata() const
{
employee::putdata();
cout << "\n Количество публикаций: " << pubs;
student::putedu();
}
};

////////////////////////////////////

class laborer : public employee // рабочий
{
};

////////////////////////////////////

int main()
{
manager m1;
scientist s1, s2;
laborer l1;
· введем информацию о нескольких сотрудниках
cout << endl;
cout << "\nВвод информации о первом менеджере";
m1.getdata();
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

```
cout << "\nВвод информации о первом
ученом"; s1.getdata();
```



```
cout << "\nВвод информации о втором ученом";
s2.getdata();
cout << "\nВвод информации о первом рабочем";
l1.getdata();
// выведем полученную информацию на экран
cout << "\nИнформация о первом менеджере";
m1.putdata();
cout << "\nИнформация о первом ученом";
s1.putdata();
cout << "\nИнформация о втором ученом";
s2.putdata();
cout << "\nИнформация о первом рабочем";
l1.putdata();
cout << endl;
return 0;
}
```

Функции `getdata()` и `putdata()` классов `manager` и `scientist` включают в себя такие вызовы функций класса `student`, как

```
student::getedu(); и
student::putedu();
```

Эти методы доступны классам `manager` и `scientist`, поскольку названные классы наследуются от класса `student`.

**В Порядок выполнения задания**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первая задачи, решение которой производится в методе класса №1; №по журналу +3 – вторая задача, решение которой



производится в методе класса №2; и №по журналу +5 – третья задача, решение которой производится в методе класса №3 (если достигнуто окончание списка вариантов зданий, то перейти в его начало).

3.2. Исследовать как меняется код программы в зависимости от последовательности наследования:

- 1,2,3;
- 2,1,3;
- 2,3,1;
- 3,2,1;
- 3,1,2;
- 1,3,2.

Отразить выводы в отчете.

3.3. Составить программу по заданию.

3.4. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.

3.5. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.

### **В Варианты заданий:**

**Составить функции пользователя для следующих задач**

- в Составить программу для перевода длины в метрах в длину в сантиметрах, определив функцию, выполняющую это преобразование и передав длину в метрах в качестве параметра.
- в Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив функцию, выполняющую это действие, и передавая массивы в качестве параметра.
- в Даны числа S, T. Получить с использованием функции пользователя  $F(T, -2S, 1.17) + F(2.2, T, S - T)$  где  $F(A, B, C) = (2A - B - \sin(C)) / (5 + C)$
- в Составить программу перевода двоичной записи натурального числа в десятичную, описав соответствующую функцию с параметром. Перевод осуществлять для чисел, вводимых с клавиатуры. Признак конца ввода - число 0.

35. Даны числа S, T. Получить с использованием функции пользователя с параметрами  $G(1, \sin(S)) + 2G(T * S, 24) - G(5, -S)$ , где  $G(A, B) = (2A + B * B) / (A * B^2 + B * 5)$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

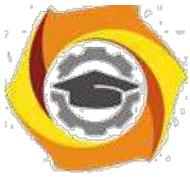
- к Составить программу для расчета значений гипотенузы треугольника, определив функцию, выполняющую этот расчет. Катеты передаются в качестве параметров.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

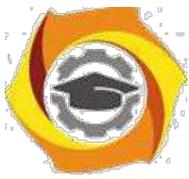
35

- Найти периметр десятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, которые передаются функции в качестве параметров из основной программы.
- Найти периметр шестиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами. Координаты передаются функции в качестве параметров из основной программы.
- Найти площадь пятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, и процедуру вычисления площади треугольника по трем сторонам. Описать функции с соответствующими формальными параметрами.
- Составить программу вывода на экран всех натуральных чисел, не превосходящих N и делящихся на каждую из своих цифр. Описать соответствующую функцию, получающую из основной программы в качестве параметра натуральное число и возвращающую TRUE, если оно удовлетворяет указанному условию.
- Используя подпрограмму - функцию, составить программу для нахождения максимального из трех чисел. Числа передаются функции в качестве параметров.
- Используя подпрограмму - функцию, составить программу для печати знаков трех чисел, введенных с клавиатуры и передаваемых функции в качестве параметра.
- Используя подпрограмму - функцию, составить программу для возведения чисел в целую положительную степень. Число передается функции в качестве параметра из основной программы. Расчет вести для чисел, пока не будет введено число, равное 0.
- Используя подпрограмму - функцию, составить программу для вычисления функции  $Z=(X1+Y1)/(X1*Y1)$ , где X1 - первый корень уравнения  $X^2-4*X-1=0$ ; Y1 - первый корень уравнения  $2*Y^2 + A*Y - A^2 = 0$  (A - произвольное).
- Задав функцию, вывести на печать средние арифметические двух массивов, введенных с клавиатуры. Массив передается функции в качестве параметра.
- Задав функцию, рассчитать и вывести на печать максимальные значения в трех парах чисел, вводимых с клавиатуры. Пара чисел передается функции в качестве параметра.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Найти периметр восьмиугольника, координаты вершин которого заданы. Определить функцию вычисления расстояния между двумя точками, заданными своими координатами. Координаты передать функции в качестве параметров.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

36

- в Даны четыре пары чисел. Получить с использованием функции пользователя наибольший общий делитель для каждой пары.
- в Даны числа A, B, C. Получить с использованием функции пользователя наименьшее значение. Числа передаются функции из основной программы в качестве параметров.
- в Даны числа  $x = 1, 2, \dots, N$ . Получить с использованием функции пользователя значения  $3 * P(x+3) * P(x)$  для заданных  $x$ , где  $P(x) = 10 * x^3 - 14 * x^2 + 12 * x - 2$ .
- в Составить программу для расчета значений катета треугольника, определив функцию, выполняющую этот расчет. Гипотенуза и второй катет передаются в качестве параметров.
- в Даны целые числа a,b,c,d. Проверить с использованием функции пользователя их четность. Число для проверки передается в функцию в качестве параметра из основной программы.
- в Для каждого из 10 введенных с клавиатуры чисел напечатать сообщение: является ли оно простым или нет, описав функцию логического типа, возвращающую значение “ИСТИНА”, если число, переданное ей в качестве параметра, является простым.
54. Даны числа S, T. Получить с использованием функции пользователя  $Y(T,S) = G(12,S) + G(T,S) - G(2S-1, S*T)$ , где  $G(A,B) = (2*A + B*B) / (A*B*2 + B*5)$ .
- В Определите функцию, определяющую, какой целой степенью числа 2 является ее аргумент (если число не является степенью двойки - выдать соответствующее сообщение).
- В Определите функцию, подсчитывающую сумму N первых элементов целочисленного массива A. N и массив A передать в качестве параметров.
- В Вычислить количество простых чисел, не превосходящих заданного N. Описать функцию логического типа, возвращающую значение true, если число простое и false в противном случае.
- В Используя подпрограмму - функцию с параметрами, составить программу для вычисления функции  $F(X,Y) = (2X^3 - 4 * X^2 + X + 1) / (9 * Y^3 + Y + 4) + 3 * Y^2 + 5 * Y$ .
- В Составить программу для перевода веса в граммах в вес в килограммах, определив функцию, выполняющую это преобразование. Вес в граммах передается функции в качестве параметра.
- В Даны числа S, T. Получить с использованием функции пользователя  $G(12, S) + G(T, S) - G(2S-1, S*T)$  где  $G(A, B) = (2 * A + B * B) / (A * B * 2 + B * 5)$ .



## Объектно-ориентированное программирование Методические указания к практическим занятиям Практическое занятие №1

### Составление программ с инкапсуляцией полей и с перегрузкой (полиморфизмом) методов

#### 1. Цель занятия:

Выработать умения и навыки по составлению программ с инкапсулированными данными (инкапсуляция) и перегруженными методами (полиморфизм).

#### 2. Рекомендации:

Изучить материалы лекций №№4,5.

### Краткая теория

#### Классы. «Перегруженные» функции.

ООП позволяет разбить задачу на ряд самостоятельных связанных между собой подзадач. Каждая подзадача содержит коды и данные, относящиеся к объекту, что упрощает решение задачи в целом и позволяет решать задачи большего объема.

Понятие объекта тесно связано с понятием класса.

Класс – это дальнейшее развитие понятия структуры. Он позволяет создавать новые типы и определять функции, манипулирующие с этими типами. По сути, класс это тип.

Объект - это представитель определенного класса. В общем, виде объект можно рассматривать как переменную, определенную программистом.

ООП использует механизмы инкапсуляции, полиморфизма и наследования.

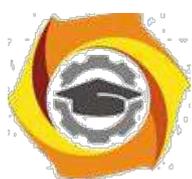
Инкапсуляция позволяет создавать объекты - данные, процедуры и функции, манипулирующие с этими данными. Данные, доступные для использования внутри объекта - private, данные доступные извне - public.

Полиморфизм позволяет одно имя функции использовать для решения разных задач (общих для класса действий). В зависимости от данных выполняются те или иные действия.

Наследование позволяет одному объекту наследовать свойства другого объекта, т.е. порожденный класс наследует свойства родительского класса и добавляет собственные свойства.

Класс используется для создания объектов. Основная форма имеет вид:

```
class имя класса
{
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

закрытые функции и переменные



public:

открытые функции, функции-члены и переменные

}

список объектов;//не является обязательным

Закрытые функции и переменные это члены (members) класса, которые до-ступны только для других членов этого класса. Открытые функции и перемен-ные доступны для любой части программы, в которой находится класс. Функ-ции, объявленные внутри описания класса, называются функциями членами (member functions).

Для определения функций-членов используется форма:

тип имя класса :: имя функции-члена (параметры)

{

тело функции

}

Два двоеточия :: после имени класса называются операцией расширения области видимости (scope resolution operator).

Класс только определяет тип объектов, а сами объекты не задает (па-мять для их хранения не выделяется). Для создания объектов имя класса ис-пользуется как спецификатор типа данных. После создания объекта к откры-тым членам класса можно обращаться, используя операцию «точка».

Пример.

```
#include <iostream.h>
class class1 //объявлен класс с именем class1 {

 int a; //доступна для функций членов class1
 public:
 int kwadrat(int b); //функция член класса class1
};
int class1::kwadrat(int b) //определение функции «kwadrat()»-члена класса
class1
{
 a=b*b;
 return a;
}
void main()
{
 class1 c; int z,x;
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

x=3;

//создается объект «C» типа class1



```
z=c.kwadrat(x);
```

```
вывод квадрата числа 3
}
```

```
cout<<"\n"<<z<<"\n";//вычисление и
```

### «Перегруженные» функции

Две или более функции называются перегруженными, если они имеют одно и то же имя.

Обычно функции отличаются количеством и типом аргументов. Транслятор автоматически на основании количества или типов аргументов выберет правильный вариант.

Пример.

```
#include <iostream.h>
```

```
void k(int a); //прототип первой функции void
k(int a, float b); //прототип второй функции
```

```
void k(int a) //описание первой функции
{
 cout << a <<"\n";
}
```

```
void k(int a, float b) //описание второй функции {

 cout <<a<<"\n"<< b <<"\n";
}
```

```
main()
{
 k(4); //вызов первой функции
 k(5, 10.2); //вызов второй функции
 return 0;
}
```

### 13. Порядок выполнения задания



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первая задача; №по журналу +3 – вторая



задача; и № по журналу +5 – третья задача (если достигнуто окончание списка вариантов заданий, то перейти в его начало).

3.2. Составить программу по заданию.

3.3. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.

3.4. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.

## 21. Варианты заданий:

**Составить методы с одинаковыми названиями для решения следующих задач**

22. Найти среднее арифметическое положительных чисел, введенных с клавиатуры. Всего ввести  $N$  различных чисел.

23. Ввести с клавиатуры  $N$  чисел. Найти сумму тех из них, которые принадлежат интервалу  $(2;9)$ .

24. Для  $N$  введенных с клавиатуры чисел найти сумму положительных кратных 3.

25. Для арифметической прогрессии 4, 9, 14, 19... найти первые  $n$  членов этой прогрессии.

26. Найти сумму отрицательных значений функции  $Z = \sin(5-x)/\cos(x-2)$  для  $x$ , изменяющегося на отрезке

$[-5,12]$  с шагом 1.

- Найти среднее арифметическое отрицательных чисел, введенных с клавиатуры. Всего ввести  $N$  различных чисел.

- Найти среднее арифметическое чисел, принадлежащих отрезку  $[2,184]$ , кратных 2 и введенных с клавиатуры. Всего ввести  $N$  различных чисел.

8. Найти сумму значений функции, больших 2  $Z = \sin(1/x) + 5\cos(1/(x-3)) + x$  для  $x$ , изменяющегося на отрезке  $[-3,8]$  с шагом 1.

9. Найти  $n$  членов последовательности  $x_1 = x_2 = x_3 = 1; x_k = x_{k-1} + x_{k-3}$ .

- Вычислить последовательность  $N$  чисел  $A_0 = x, A_1 = 2, A_k = A_{k-1} - A_{k-2}$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Для  $x_1 = 0,3$  и  $x_2 = -0,3$  найти  $x_k = k + \sin(x_k - 2)$  для  $k$ , изменяющегося следующим образом:  $k = 3, 4, \dots, 14$ .



и Составить таблицу перевода дюймов в сантиметры для расстояний от 1 до 13 дюймов с шагом 1.

и Вывести на печать значения функции, меньшие 2,  $Z = \sin(1/x) + 5\cos(x-3) + x$  для  $x$ , изменяющегося на отрезке  $[-7,4]$  с шагом 1.

14. Напечатать таблицу значений функции  $Y = \operatorname{tg}(x/b) + x/(b-2)$  для  $x$ , изменяющегося от 0 до 10 с шагом 1 ( $b$  - произвольное число).

её Вычислить  $N$ -ый член последовательности  $x_k = x_{k-2} - x_{k-1}$ ,  $x_0 = 2,4$ ,  $x_1 = 3,8$ .

Составить таблицу перевода суток (от 1 до 7) в часы, минуты, секунды.

Вычислить  $N$ -ый член последовательности

$$x_k = x_{k-1} + (2/3)x_{k-2} + 1, \quad x_1 = -1, \quad x_2 = 1,38.$$

В Напечатать значения функции

22.  $= 1/(x-2) + 1/(x-5) + \ln(12,8-X)$  для  $x$ , изменяющегося на отрезке  $[-4,14]$  с шагом 1.

Вывести на печать отрицательные значения функции

$z = \sin(5-x)/\cos(x-2)$  для  $x$ , изменяющегося на отрезке  $[-6,13]$  с шагом 1 (учесть область допустимых значений функции).

21. Из  $N$  введенных с клавиатуры чисел напечатать кратные 3 и меньшие 58.

22. Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(1,11)$  и являются четными.

23. Из  $N$  введенных с клавиатуры чисел напечатать положительные, кратные 3.

24. Вывести на печать значения функции

$z = \sin(x/(x-2))$ , находящиеся в интервале  $(-0,4;0,8)$  для  $x$ , изменяющегося от 8 до -6 с шагом 1.

24) Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(2;9)$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

25) Для геометрической прогрессии 2, 6, 18, 54, 162 ... определить первые  $n$  членов этой прогрессии.



В Ввести с клавиатуры N чисел. Напечатать те из них, которые не принадлежат интервалу (1;5).

27. Найти n членов последовательности  $x_1 = x_2 = x_3 = 1$ ;  $x_k = x_{k-1} - 2x_{k-3}$ .

Вычислить последовательность N чисел  $A_0 = x$ ,

$A_1 = 2$ ,  $A_k = A_{k-1} + A_{k-2}$

Составить таблицу перевода килограммов (от 1 до 13) в граммы с шагом

88 Найти сумму значений функции  $Y = \cos(x/A) + x/(A-2)$  для x, изменяющегося от 2 до 13 с шагом 1 (A - произвольное число).

## Практическое занятие №2

### Создание приложений с различными видами наследования

#### 1. Цель занятия:

Выработать умения и навыки по составлению программ, содержащий классы, наследующие другие классы.

#### 2. Рекомендации:

Изучить материалы лекций №№4-6.

### Краткая теория

#### Наследование. Виртуальные функции. Указатели на объекты

Наследование - это механизм посредством которого один класс (производный или потомок) может наследовать свойства другого класса (базового или предка).

Базовый класс определяет все качества, которые являются общими для всех производных классов.

Пример:

```
//Базовый класс
```

```
class B {
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

`int i;`



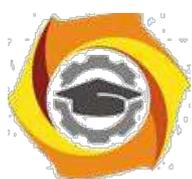
```
public:
 void set_i(int n);
 int get_i();
};
//Производный класс D
class D : public B
{
 int j;
public:
 void set_j(int n);
 int mul();
};
```

После имени класса D стоит двоеточие, за которым стоит ключевое слово public и имя класса B. Это означает, что класс D будет наследовать все компоненты класса B. Само ключевое слово public информирует компилятор о том, что т.к. класс B будет наследоваться, то все открытые элементы базового класса будут открытыми элементами производного класса. Однако все закрытые элементы базового класса остаются закрытыми.

Пример:

13. Простой пример наследования.

```
#include <iostream.h>
/ Задание базового класса
class base {
 int i;
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

public:

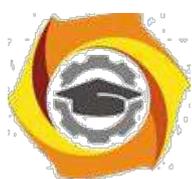


```
void set_i(int n);
int get_i();
};

/ Задание производного класса
class derived : public base
{
 int j;
 public:
 void set_j(int n);
 int mul();
};

/ Установка значения i в базовом классе
void base::set_i(int n)
{
 i = n;
}

/ Возврат значения i в базовом классе
int base::get_i()
{
 return i;
}
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

}



В Установка значения  $j$  в производном классе

```
void derived::set_j(int n)
{
 j = n;
}
```

\endash Возврат значения  $i$  из `base` и, одновременно,  $j$  из `derived`

```
int derived::mul()
{
 25. производный класс может вызывать функции-члены базового
 класса
 return j * get_i();
}
```

```
main()
{
 derived ob;

 ob.set_i(10); // загрузка i в base
 ob.set_j(4); // загрузка j в derived

 cout << ob.mul(); // вывод числа 40
}
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

```
return 0;
```

```
}
```



При определении `mul()` вызывается функция `get_i()`- базового класса `B`, а не производного `D`, что указывает на то, что открытые члены базового класса становятся открытыми членами производного. Но в функции `mul()` вместо прямого доступа к `i`, необходимо вызывать `get_i()`, потому что закрытые члены базового класса(`i`) остаются закрытыми для производных классов.

Виртуальные функции.

Проблема: как будет вызываться функция производного класса, имеющая та-кое же название, что функция базового класса.

Пример.

```
#include <stdio.h>

class base
{
public:
 int i;

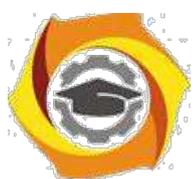
 base(int x); //конструктор

 void func()
 {
 printf("Базовая функция %d",i);

 return;
 };
};

//текст конструктора

base::base(int x)
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

{



```
i=x;

return;

};

class der1: public base
{
public:
 der1(int x) :base(x) {};//конструктор
 void func()
 {
 printf("Функция из производного класса %d",
 i*i); return;
 }
};

main()
{
 base * pc; pc->func(); return 0;
 base ob(2);
 der1 ob1(2);
 pc=&ob;
 pc->func();
 pc=&ob1;
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

```
//указатель на базовый класс
//создать экземпляр объекта производного класса
//указатель на объект базового класса
//вызов функции базового класса
//указатель на объект производного класса
//попытка вызова функции производного класса
}
```



На первый взгляд кажется, что в первом случае будет вызываться функция базового класса, а во втором функция производного. Однако при проверке Вы легко убедитесь, что и в том и в другом случае будет вызвана функция базового класса. При этом, компилятору трудно «понять», какую реально функцию мы имеем в виду и он на стадии компилирования подставляет во всех тех случаях, где встречается имя `func()` адрес функции базового класса. Такой процесс установки адресов называется "ранним связыванием" или "статическим связыванием". Если же мы хотим, чтобы во втором случае, т.е. когда указатель `rc` указывал на производный класс вызывалась функция этого класса, ее еще в базовом классе следует указать как виртуальную. В нашем случае вместо строки `void func()` следует написать `virtual void func()`.

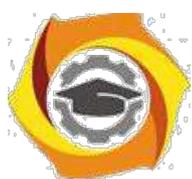
Это напоминает «перегрузку» функций, но «перегружаемые» функции отличаются друг от друга типом или аргументами, здесь же функции должны быть идентичны.

33. случае использования виртуальных функций адрес вызываемой функции будет определяться в процессе выполнения кода программы. Такой процесс называется "поздним связыванием" или "динамическим связыванием".

Пример:

```
#include <stdio.h>

class base {
public:
 int i;
 base(int x); //конструктор
 virtual void func()
 {
 printf("Базовая функция %d\n",i);
 return;
 };
};
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

};



```
//текст конструктора
base::base(int x)
{
 i=x;
 return;
};
class der1: public base {
public:
 der1(int x) :base(x) {}; //конструктор
 void func()
 {
 printf("Функция из производного класса %d\n", i*i);
 return;
 }
};
class der2: public base {
public:
 der2(int x) :base(x) {}; //конструктор
};

main()
{
 base * pc; base ob(2); der1 ob1(2);
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

//указатель  
на базовый  
класс

//создать  
экземпляр  
объекта  
базового  
класса

//создать  
экземпляр  
объекта  
производног  
о класса 1



```
der2 ob2(2); //создать экземпляр объекта производного класса 2

pc=&ob; //указатель на объект базового класса

pc->func(); //вызов функции базового класса

pc=&ob1; //указатель на объект производного класса 1

pc->func(); //попытка вызова функции производного класса

pc=&ob2; //указатель на объект производного класса 2

pc->func(); //попытка вызова функции производного класса

return 0;

}
```

Введен еще один производный класс. В нем функция `func()` не определена. В этом случае будет вызываться функция класса родителя. Т.е. появится строка: «Базовая функция 2». При этом, принцип прост: если Вы хотите, чтобы вызывалась функция родительского класса, не определяйте ее в производном. Еще один вопрос может возникнуть в связи с данным примером: как быть, если мы хотим, чтобы для класса объектов `der2` вызывалась функция класса `der1`. Решение очень просто - сделайте класс `der2` наследником не класса `base`, а класса `der1`.

23. производных классах функция, определенная в базовом классе как виртуальная может определяться, а может и нет. Если Вы хотите, чтобы во всех производных классах обязательно была определена виртуальная функция, то в базовом классе ее надо определить следующим образом:

```
virtual void func() = 0;
```

27. этом случае базовый класс называется агрегатным и от него нельзя будет создавать экземпляры объектов, зато во всех производных классах



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

компилятор «потребуется» определить данную виртуальную функцию и, тем самым, уменьшить вероятность ошибок.



### № Порядок выполнения задания

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первая задачи, решение которой производится в методе класса №1; №по журналу +3 – вторая задача, решение которой производится в методе класса №2; и №по журналу +5 – третья задача, решение которой производится в методе класса №3 (если достигнуто окончание списка вариантов заданий, то перейти в его начало).

3.2. Исследовать как меняется код программы в зависимости от последовательности наследования:

44.1,2,3;

45.2,1,3;

46.2,3,1;

47.3,2,1;

48.3,1,2;

49.1,3,2.

Отразить выводы в отчете.

3.3. Составить программу по заданию.

3.4. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.

3.5. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.

### 41. Варианты заданий:

Составить функции пользователя для следующих задач

35. Составить программу для перевода длины в метрах в длину в сантиметрах, определив функцию, выполняющую это преобразование и передав длину в метрах в качестве параметра.

36. Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив функцию, выполняющую это действие, и передавая массивы в качестве параметра.

37. Даны числа S, T. Получить с использованием функции пользователя  $F(T, -2S, 1.17) + F(2.2, T, S - T)$  где  $F(A, B, C) = (2A - B - \sin(C)) / (5 + C)$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

38. Составить программу перевода двоичной записи натурального числа в десятичную, описав соответствующую функцию с параметром. Перевод осуществлять для чисел, вводимых с клавиатуры. Признак конца ввода - число 0.

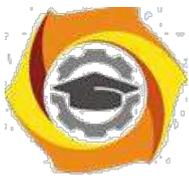


5. Даны числа  $S$ ,  $T$ . Получить с использованием функции пользователя с параметрами  $G(1, \sin(S)) + 2G(T * S, 24) - G(5, -S)$ , где  $G(A, B) = (2A + B * B) / (A * B * 2 + B * 5)$ .
35. Составить программу для расчета значений гипотенузы треугольника, определив функцию, выполняющую этот расчет. Катеты передаются в качестве параметров.
36. Найти периметр десятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, которые передаются функции в качестве параметров из основной программы.
37. Найти периметр шестиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами. Координаты передаются функции в качестве параметров из основной программы.
38. Найти площадь пятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, и процедуру вычисления площади треугольника по трем сторонам. Описать функции с соответствующими формальными параметрами.
39. Составить программу вывода на экран всех натуральных чисел, не превосходящих  $N$  и делящихся на каждую из своих цифр. Описать соответствующую функцию, получающую из основной программы в качестве параметра натуральное число и возвращающую TRUE, если оно удовлетворяет указанному условию.
40. Используя подпрограмму - функцию, составить программу для нахождения максимального из трех чисел. Числа передаются функции в качестве параметров.
41. Используя подпрограмму - функцию, составить программу для печати знаков трех чисел, введенных с клавиатуры и передаваемых функции в качестве параметра.
42. Используя подпрограмму - функцию, составить программу для возведения чисел в целую положительную степень. Число передается функции в качестве параметра из основной программы. Расчет вести для чисел, пока не будет введено число, равное 0.
43. Используя подпрограмму - функцию, составить программу для вычисления функции  $Z = (X1 + Y1) / (X1 * Y1)$ , где  $X1$  - первый корень уравнения  $X^2 - 4 * X - 1 = 0$ ;  $Y1$  - первый корень уравнения  $2 * Y^2 + A * Y - A^2 = 0$  ( $A$  - произвольное).
44. Задав функцию, вывести на печать средние арифметические двух массивов, введенных с клавиатуры. Массив передается функции в качестве параметра.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

45. Задав функцию, рассчитать и вывести на печать максимальные значения в трех парах чисел, вводимых с клавиатуры. Пара чисел передается функции в качестве параметра.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

19

- В Найти периметр восьмиугольника, координаты вершин которого заданы. Определить функцию вычисления расстояния между двумя точками, заданными своими координатами. Координаты передать функции в качестве параметров.
- В Даны четыре пары чисел. Получить с использованием функции пользователя наибольший общий делитель для каждой пары.
- В Даны числа A, B, C. Получить с использованием функции пользователя наименьшее значение. Числа передаются функции из основной программы в качестве параметров.
- В Даны числа  $x = 1, 2, \dots, N$ . Получить с использованием функции пользователя значения  $3 * P(x+3) * P(x)$  для заданных  $x$ , где  $P(x) = 10 * x^3 - 14 * x^2 + 12 * x - 2$ .
- В Составить программу для расчета значений катета треугольника, определив функцию, выполняющую этот расчет. Гипотенуза и второй катет передаются в качестве параметров.
- В Даны целые числа a, b, c, d. Проверить с использованием функции пользователя их четность. Число для проверки передается в функцию в качестве параметра из основной программы.
- В Для каждого из 10 введенных с клавиатуры чисел напечатать сообщение: является ли оно простым или нет, описав функцию логического типа, возвращающую значение “ИСТИНА”, если число, переданное ей в качестве параметра, является простым.
24. Даны числа S, T. Получить с использованием функции пользователя  $Y(T, S) = G(12, S) + G(T, S) - G(2S - 1, S * T)$ , где  $G(A, B) = (2 * A + B * B) / (A * B * 2 + B * 5)$ .
- OOO Определите функцию, определяющую, какой целой степенью числа 2 является ее аргумент (если число не является степенью двойки - выдать соответствующее сообщение).
- OOO Определите функцию, подсчитывающую сумму N первых элементов целочисленного массива A. N и массив A передать в качестве параметров.
- OOO Вычислить количество простых чисел, не превосходящих заданного N. Описать функцию логического типа, возвращающую значение true, если число простое и false в противном случае.
- OOO Используя подпрограмму - функцию с параметрами, составить программу для вычисления функции  $F(x, y) = (2x^3 - 4x^2 + x + 1) / (9y^3 + y + 4) + 3y^2 + 5y$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

ООО Составить программу для перевода веса в граммах в вес в килограммах, определив функцию, выполняющую это преобразование. Вес в граммах передается функции в качестве параметра.



16. Даны числа S, T. Получить с использованием функции пользователя  $G(12, S)+G(T, S)-G(2S-1, S*T)$  где  $G(A, B) = (2*A+B*B)/(A*B*2+B*5)$ .

### Практическое занятие №3

#### Составление программ с инкапсуляцией полей и с перегрузкой (полиморфизмом) методов в Java

##### 1. Цель занятия:

Выработать умения и навыки по составлению программ с инкапсулированными данными (инкапсуляция) и перегруженными методами (полиморфизм).

##### 2. Рекомендации:

Изучить материалы лекций №№4,5.

#### Краткая теория

##### Совмещение методов на примере Java-программы

Язык Java позволяет создавать несколько методов с одинаковыми именами, но с разными списками параметров. Такая техника называется совмещением методов (**method overloading**).

34. в качестве примера приведена версия класса Pt, в которой совмещение методов использовано для определения альтернативного конструктора, который инициализирует координаты x и y значениями по умолчанию (-1).

```
class Pt
{
 int x, y;
 Pt(int x, int y)
 {
 this.x = x;
 this.y = y;
 }

 Pt()
 {
```



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

31. =  
-1; y=-  
1;  
}  
}



```
class PtCrAlt
{
 public static void main(String args[])
 {
 Pt ob = new Pt();
 System.out.println("x = " + ob.x + " y = " + ob.y);
 }
}
```

этом примере объект класса Pt создается не при вызове первого конструктора, как это было раньше, а с помощью второго конструктора без параметров. Вот результат работы этой программы:

$x = -1 y = -1$

Решение о том, какой конструктор нужно вызвать в том или ином случае, принимается в соответствии с количеством и типом параметров, указанных в операторе `new`. **Недопустимо объявлять в классе методы с одинаковыми именами и сигнатурами.** В сигнатуре метода не учитываются имена формальных параметров учитываются лишь их типы и количество.

### **this в конструкторах**

Очередной вариант класса Pt показывает, как, используя **this** и совмещение методов, можно строить одни конструкторы на основе других.

```
class Pt
{ int x, y;

 Pt(int x, int y)
 {
 this.x = x;
 this.y = y;
 }

 Pt()
 {
 this(-1, -1);
 }
}
```

36. этом примере второй конструктор для завершения инициализации объекта обращается к первому конструктору.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Методы, использующие совмещение имен, не обязательно должны быть конструкторами. В следующем примере в класс Pt добавлены два метода distance. Функция distance возвращает расстояние между двумя точками. Одно-



му из совмещенных методов в качестве параметров передаются координаты точки  $x$  и  $y$ , другому же эта информация передается в виде параметра-объекта класса  $Pt$ .

```
class Pt
{ double x, y;
 Pt(double x, double y)
 {
 this.x = x;
 this.y = y;
 }

 double distance (double x, double y)
 { double dx, dy;
 dx = this.x - x;
 dy = this.y - y;
 return Math.sqrt(dx*dx + dy*dy);
 }

 double distance(Pt ob)
 {
 return distance(ob.x, ob.y);
 }
}

class PtDist
{
 public static void main(String args[])
 {
 Pt ob1 = new Pt(0, 0);
 Pt ob2 = new Pt(30, 40);
 System.out.println("ob1 = " + ob1.x + ", " + ob1.y);
 System.out.println("ob2 = " + ob2.x + ", " + ob2.y);
 System.out.println("ob1.distance(p2) = " + ob1.distance(p2));
 System.out.println("ob1.distance(60, 80) = " + ob1.distance(60, 80));
 }
}
```

Во второй форме метода **distance** для получения результата вызывается его первая форма. Ниже приведен результат работы этой программы:

```
ob1 = 0, 0
ob2 = 30, 40
```



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

*ob1.distance(ob2) = 50.0*

*ob1.distance(60, 80) = 100.0*



## Инкапсулирование полей на примере Java-программы

Данные инкапсулируются в класс путем объявления переменных между открывающей и закрывающей фигурными скобками, выделяющими в определении класса его тело.

Эти переменные объявляются точно так же, как локальные переменные, которые надо объявлять вне методов, в том числе вне метода main.

```
class Pt
{
int x, y;
}
```

### Оператор new

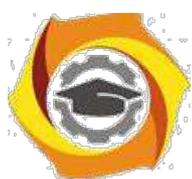
Оператор new создает экземпляр указанного класса и возвращает ссылку на вновь созданный объект.

```
Pt ob = new Pt();
```

Вы можете создать несколько ссылок на один и тот же объект.

Оператор точка используется для доступа к переменным и методам объекта.

```
class PtDva
{
 public static void main(String args[])
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

}



```
Pt ob1 = new Pt();
Pt ob2 = new Pt();
ob1.x = 10;
ob1.y = 20;
ob2.x = 42;
ob2.y = 99;
System.out.println("x = " + ob1.x + " y = " + ob1.y);
System.out.println("x = " + ob2.x + " y = " + ob2.y);
}
}
```

13.     этом примере снова использовался класс Pt, было создано два объекта этого класса, и их переменным x и y присвоены различные значения. Таким образом мы продемонстрировали, что переменные различных объектов независимы на самом деле.

Ниже приведен результат, полученный при выполнении этой программы.

=10y=20

=42y=99

## 18.    Порядок выполнения задания

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первая задача; №по журналу +3 – вторая задача; и №по журналу +5 – третья задача (если достигнуто окончание списка вариантов заданий, то перейти в его начало).

3.2. Составить программу по заданию.

3.3. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

3.4. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.



**18. Варианты заданий:**

**Составить методы с одинаковыми названиями для решения следующих задач**

16. Найти среднее арифметическое положительных чисел, введенных с клавиатуры. Всего ввести  $N$  различных чисел.

17. Ввести с клавиатуры  $N$  чисел. Найти сумму тех из них, которые принадлежат интервалу  $(2;9)$ .

18. Для  $N$  введенных с клавиатуры чисел найти сумму положительных кратных 3.

19. Для арифметической прогрессии 4, 9, 14, 19... найти первые  $n$  членов этой прогрессии.

20. Найти сумму отрицательных значений функции  $Z=\sin(5-x)/\cos(x-2)$  для  $x$ , изменяющегося на отрезке

$[-5,12]$  с шагом 1.

17. Найти среднее арифметическое отрицательных чисел, введенных с клавиатуры. Всего ввести  $N$  различных чисел.

18. Найти среднее арифметическое чисел, принадлежащих отрезку  $[2,184]$ , кратных 2 и введенных с клавиатуры. Всего ввести  $N$  различных чисел.

8. Найти сумму значений функции, больших 2  $Z=\sin(1/x)+5\cos(1/(x-3))+x$  для  $x$ , изменяющегося на отрезке  $[-3,8]$  с шагом 1.

9. Найти  $n$  членов последовательности  $x_1 = x_2 = x_3 = 1; x_k = x_{k-1} + x_{k-3}$ .

15. Вычислить последовательность  $N$  чисел  $A_0 = x, A_1 = 2, A_k = A_{k-1} - A_{k-2}$ .

16. Для  $x_1 = 0,3$  и  $x_2 = -0,3$  найти  $x_k = k + \sin(x_{k-2})$  для  $k$ , изменяющегося следующим образом:  $k = 3,4,\dots,14$ .

17. Составить таблицу перевода дюймов в сантиметры для расстояний от 1 до 13 дюймов с шагом 1.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

18. Вывести на печать значения функции, меньшие 2,  $Z = \sin(1/x) + 5\cos(x-3) + x$  для  $x$ , изменяющегося на отрезке  $[-7, 4]$  с шагом 1.

14. Напечатать таблицу значений функции  $Y = \operatorname{tg}(x/b) + x/(b-2)$  для  $x$ , изменяющегося от 0 до 10 с шагом 1 ( $b$  - произвольное число).



16. Вычислить  $N$ -ый член последовательности  $x_k = x_{k-2} - x_{k-1}$ ,  $x_0 = 2,4$   
 $x_1 = 3,8$ .

Составить таблицу перевода суток (от 1 до 7) в часы, минуты, секунды.

Вычислить  $N$ -ый член последовательности

$$x_k = x_{k-1} + (2/3)x_{k-2} + 1, \quad x_1 = -1, \quad x_2 = 1,38.$$

13. Напечатать значения функции

10.  $y = 1/(x-2) + 1/(x-5) + \ln(12,8-X)$  для  $x$ , изменяющегося на отрезке  $[-4,14]$   
с шагом 1.

Вывести на печать отрицательные значения функции

$z = \sin(5-x)/\cos(x-2)$  для  $x$ , изменяющегося на отрезке  $[-6,13]$  с шагом 1  
(учесть область допустимых значений функции).

И Из  $N$  введенных с клавиатуры чисел напечатать кратные 3 и меньшие 58.

И Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(1,11)$  и являются четными.

И Из  $N$  введенных с клавиатуры чисел напечатать положительные, кратные 3.

И Вывести на печать значения функции

$z = \sin(x/(x-2))$ , находящиеся в интервале  $(-0,4;0,8)$  для  $x$ , изменяющегося от 8 до -6 с шагом 1.

· Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(2;9)$ .

· Для геометрической прогрессии 2, 6, 18, 54, 162 ... определить первые  $n$  членов этой прогрессии.

· Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые не принадлежат интервалу  $(1;5)$ .

27. Найти  $n$  членов последовательности  $x_1 = x_2 = x_3 = 1$ ;  $x_k = x_{k-1} - 2x_{k-3}$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Вычислить последовательность  $N$  чисел  $A_0 = x$ ,  
 $A_1 = 2$ ,  $A_k = A_{k-1} + A_{k-2}$



Составить таблицу перевода килограммов (от 1 до 13) в граммы с шагом

12) Найти сумму значений функции  $Y = \cos(x/A) + x/(A-2)$  для  $x$ , изменяющегося от 2 до 13 с шагом 1 ( $A$  - произвольное число).

## Практическое занятие №4

### Составление программ с наследованием классов в Java

#### 1. Цель занятия:

Выработать умения и навыки по составлению программ, содержащих классы, наследующие другие классы.

#### 2. Рекомендации:

Изучить материалы лекций №№4-6.

### Краткая теория

#### Реализация множественного наследования на примере Java - программ

Класс может быть производным не только от одного базового класса, а и от многих. Этот случай называется множественным наследованием. Синтаксис описания множественного наследования похож на синтаксис простого наследования:

```
class A {

};
class B {

};
class C : public A, public B {

};
```

Базовые классы класса C перечислены после двоеточия в строке описания класса и разделены запятыми.



## Методы (функции) классов и множественное наследование

Рассмотрим пример множественного наследования. Пусть нам для некоторых служащих необходимо указать их образование в программе EMPLOY. Теперь предположим, что в другой программе у нас существует класс student, в котором указывается образование каждого студента. Тогда вместо изменения класса employee мы воспользуемся данными класса student с помощью множественного наследования.

и классе student содержатся сведения о школе или университете, которые закончил студент, и об уровне полученного им образования. Эти данные хранятся в строковом формате. Методы getedu() и putedu() позволяют нам ввести данные о студенте и просмотреть их.

Информация об образовании нужна нам не для всех служащих. Предположим, что нам не нужны записи об образовании рабочих, а необходимы только записи об ученых и менеджерах. Поэтому мы модифицируем классы manager и scientist так, что они будут яЭта маленькая программа показывает только взаимосвязь между классами:

```
class student
{
};

class employee
{
};

class manager : public employee, private student
{
};

class scientist : private employee, private student
{
};

class laborer : public employee
{
};
```

Теперь мы рассмотрим эти классы более детально в листинге EMPMULT.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

// empmult.cpp



- множественное наследование

```
#include <iostream>

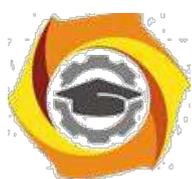
using namespace std;

const int LEN = 80; // максимальная длина имени
////////////////////

class student // сведения об образовании
{
private:
char school[LEN]; // название учебного
заведения char degree[LEN]; // уровень
образования public:
void getedu()
{
cout << " Введите название учебного заведения: ";
cin >> school;

cout << " Введите степень высшего образования\n"; // вляться производными клас-
сов employee и student.
cout << " (неполное высшее, бакалавр, магистр, кандидат наук): ";
cin >> degree;
}

void putedu() const
{
cout << "\n Учебное заведение: " << school;
cout << "\n Степень: " << degree;
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

};



Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»

30

////////////////////////////////////

```
class employee // некий сотрудник
{
private:
char name[LEN]; // имя сотрудника
unsigned long number; // номер
сотрудника public:
void getdata()
{
cout << "\n Введите фамилию: "; cin >> name;
cout << " Введите номер: "; cin >> number;
}
void putdata() const
{
cout << "\n Фамилия: " << name;
cout << "\n Номер: " << number;
}
};
```

////////////////////////////////////

```
class manager : private employee, private student // менеджер
{
private:
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

char title[LEN]; // должность сотрудника

double dues; // сумма взносов в гольф-

клуб public:



```
void getdata()
{
employee::getdata();
cout << " Введите должность: "; cin >> title;
cout << " Введите сумму взносов в гольф-клуб: "; cin >> dues;
student::getedu();
}

void putdata() const
{
employee::putdata();
cout << "\n Должность: " << title;
cout << "\n Сумма взносов в гольф-клуб: " << dues;
student::putedu();
}

};

////////////////////////////////////

class scientist : private employee, private student // ученый
{
private:
int pubs; // количество публикаций
public:
void getdata()
{
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

```
employee::getdata();
```

```
cout << " Введите количество публикаций: "; cin >> pubs;
```



```
student::getedu();
}

void putdata() const
{
employee::putdata();

cout << "\n Количество публикаций: " << pubs;

student::putedu();
}
};

////////////////////////////////////

class laborer : public employee // рабочий
{
};

////////////////////////////////////

int main()
{
manager m1;

scientist s1, s2;

laborer l1;

· введем информацию о нескольких сотрудниках

cout << endl;

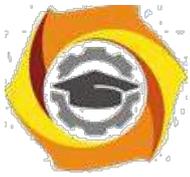
cout << "\nВвод информации о первом менеджере";

m1.getdata();
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

```
cout << "\nВвод информации о первом
ученом"; s1.getdata();
```



```
cout << "\nВвод информации о втором ученом";
s2.getdata();
cout << "\nВвод информации о первом рабочем";
l1.getdata();
// выведем полученную информацию на экран
cout << "\nИнформация о первом менеджере";
m1.putdata();
cout << "\nИнформация о первом ученом";
s1.putdata();
cout << "\nИнформация о втором ученом";
s2.putdata();
cout << "\nИнформация о первом рабочем";
l1.putdata();
cout << endl;
return 0;
}
```

Функции `getdata()` и `putdata()` классов `manager` и `scientist` включают в себя такие вызовы функций класса `student`, как

```
student::getedu(); и
student::putedu();
```

Эти методы доступны классам `manager` и `scientist`, поскольку названные классы наследуются от класса `student`.

**В Порядок выполнения задания**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первая задачи, решение которой производится в методе класса №1; №по журналу +3 – вторая задача, решение которой



производится в методе класса №2; и №по журналу +5 – третья задача, решение которой производится в методе класса №3 (если достигнуто окончание списка вариантов зданий, то перейти в его начало).

3.2. Исследовать как меняется код программы в зависимости от последовательности наследования:

- 1,2,3;
- 2,1,3;
- 2,3,1;
- 3,2,1;
- 3,1,2;
- 1,3,2.

Отразить выводы в отчете.

3.3. Составить программу по заданию.

3.4. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.

3.5. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.

### **В Варианты заданий:**

**Составить функции пользователя для следующих задач**

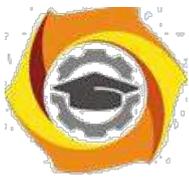
- в Составить программу для перевода длины в метрах в длину в сантиметрах, определив функцию, выполняющую это преобразование и передав длину в метрах в качестве параметра.
- в Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив функцию, выполняющую это действие, и передавая массивы в качестве параметра.
- в Даны числа S, T. Получить с использованием функции пользователя  $F(T, -2S, 1.17) + F(2.2, T, S - T)$  где  $F(A, B, C) = (2A - B - \sin(C)) / (5 + C)$
- в Составить программу перевода двоичной записи натурального числа в десятичную, описав соответствующую функцию с параметром. Перевод осуществлять для чисел, вводимых с клавиатуры. Признак конца ввода - число 0.

35. Даны числа S, T. Получить с использованием функции пользователя с параметрами  $G(1, \sin(S)) + 2G(T * S, 24) - G(5, -S)$ , где  $G(A, B) = (2A + B * B) / (A * B^2 + B * 5)$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- к Составить программу для расчета значений гипотенузы треугольника, определив функцию, выполняющую этот расчет. Катеты передаются в качестве параметров.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

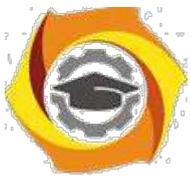
35

- Найти периметр десятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, которые передаются функции в качестве параметров из основной программы.
- Найти периметр шестиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами. Координаты передаются функции в качестве параметров из основной программы.
- Найти площадь пятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, и процедуру вычисления площади треугольника по трем сторонам. Описать функции с соответствующими формальными параметрами.
- Составить программу вывода на экран всех натуральных чисел, не превосходящих N и делящихся на каждую из своих цифр. Описать соответствующую функцию, получающую из основной программы в качестве параметра натуральное число и возвращающую TRUE, если оно удовлетворяет указанному условию.
- Используя подпрограмму - функцию, составить программу для нахождения максимального из трех чисел. Числа передаются функции в качестве параметров.
- Используя подпрограмму - функцию, составить программу для печати знаков трех чисел, введенных с клавиатуры и передаваемых функции в качестве параметра.
- Используя подпрограмму - функцию, составить программу для возведения чисел в целую положительную степень. Число передается функции в качестве параметра из основной программы. Расчет вести для чисел, пока не будет введено число, равное 0.
- Используя подпрограмму - функцию, составить программу для вычисления функции  $Z=(X1+Y1)/(X1*Y1)$ , где X1 - первый корень уравнения  $X^2-4*X-1=0$ ; Y1 - первый корень уравнения  $2*Y^2 + A*Y - A^2 = 0$  (A - произвольное).
- Задав функцию, вывести на печать средние арифметические двух массивов, введенных с клавиатуры. Массив передается функции в качестве параметра.
- Задав функцию, рассчитать и вывести на печать максимальные значения в трех парах чисел, вводимых с клавиатуры. Пара чисел передается функции в качестве параметра.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Найти периметр восьмиугольника, координаты вершин которого заданы. Определить функцию вычисления расстояния между двумя точками, заданными своими координатами. Координаты передать функции в качестве параметров.



- в Даны четыре пары чисел. Получить с использованием функции пользователя наибольший общий делитель для каждой пары.
- в Даны числа A, B, C. Получить с использованием функции пользователя наименьшее значение. Числа передаются функции из основной программы в качестве параметров.
- в Даны числа  $x = 1, 2, \dots, N$ . Получить с использованием функции пользователя значения  $3 * P(x+3) * P(x)$  для заданных  $x$ , где  $P(x) = 10 * x^3 - 14 * x^2 + 12 * x - 2$ .
- в Составить программу для расчета значений катета треугольника, определив функцию, выполняющую этот расчет. Гипотенуза и второй катет передаются в качестве параметров.
- в Даны целые числа a, b, c, d. Проверить с использованием функции пользователя их четность. Число для проверки передается в функцию в качестве параметра из основной программы.
- в Для каждого из 10 введенных с клавиатуры чисел напечатать сообщение: является ли оно простым или нет, описав функцию логического типа, возвращающую значение “ИСТИНА”, если число, переданное ей в качестве параметра, является простым.
54. Даны числа S, T. Получить с использованием функции пользователя  $Y(T, S) = G(12, S) + G(T, S) - G(2S - 1, S * T)$ , где  $G(A, B) = (2 * A + B * B) / (A * B * 2 + B * 5)$ .
- В Определите функцию, определяющую, какой целой степенью числа 2 является ее аргумент (если число не является степенью двойки - выдать соответствующее сообщение).
- В Определите функцию, подсчитывающую сумму N первых элементов целочисленного массива A. N и массив A передать в качестве параметров.
- В Вычислить количество простых чисел, не превосходящих заданного N. Описать функцию логического типа, возвращающую значение true, если число простое и false в противном случае.
- В Используя подпрограмму - функцию с параметрами, составить программу для вычисления функции  $F(X, Y) = (2X^3 - 4 * X^2 + X + 1) / (9 * Y^3 + Y + 4) + 3 * Y^2 + 5 * Y$ .
- В Составить программу для перевода веса в граммах в вес в килограммах, определив функцию, выполняющую это преобразование. Вес в граммах передается функции в качестве параметра.
- В Даны числа S, T. Получить с использованием функции пользователя  $G(12, S) + G(T, S) - G(2S - 1, S * T)$  где  $G(A, B) = (2 * A + B * B) / (A * B * 2 + B * 5)$ .



**Методическое пособие**  
**для проведения лабораторных работ и практических занятий**  
**С Практическое занятие № 1 Установка ОС Ubuntu на Oracle VM**  
**VirtualBox**

Цель работы:

- i) Приобретение навыков установки и создания виртуальных машин в Oracle VM VirtualBox
- j) Приобретение навыков установки и начальной настройки операционной системы Ubuntu

**1.1. Задание:**

- С Скачать или получить у преподавателя образ установочного диска ОС Ubuntu
- С Создать виртуальную машину в VirtualBox.
- С Установить ОС Ubuntu на созданную виртуальную машину.
- С В установленной операционной системе: -  
установить дополнения гостевой ОС -настроить  
рабочие столы (эффекты, изображения); -изменить  
раскладку клавиатуры по умолчанию;  
-определить тип сеанса загружаемую по умолчанию.
- С Ознакомиться и описать панель инструментов Ubuntu.

**1.2. Краткие теоретические сведения**

Занятия по дисциплине «Операционные системы проводятся с использованием технологии виртуализации.

Виртуализация — предоставление набора вычислительных ресурсов или их логического объединения, абстрагированное от аппаратной реализации, и обеспечивающее при этом логическую изоляцию друг от друга вычислительных процессов, выполняемых на одном физическом ресурсе.

Примером использования виртуализации является возможность запуска нескольких операционных систем на одном компьютере: при том каждый из экземпляров таких гостевых операционных систем работает со своим набором логических ресурсов (процессорных, оперативной памяти, устройств хранения), предоставлением которых из общего пула, доступного на уровне оборудования, управляет хостовая операционная система — гипервизор.

Благодаря использованию данной технологии у студента появляется возможность безопасно работать с различными гостевыми операционными системами, не подвергая опасности настройку хостовой операционной системы компьютера учебного класса.

- 15) качества средства виртуализации будем использовать программное средство  
Oracle VM VirtualBox.

VirtualBox (Oracle VM VirtualBox) — программный продукт виртуализации для операционных систем Microsoft Windows, Linux, FreeBSD, macOS, Solaris/OpenSolaris, ReactOS, DOS и других. Использование виртуальной машины (VM) на домашнем ПК, прежде всего, позволит одновременно запускать несколько операционных систем (гостевые ОС).



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

К примеру, в данный момент на компьютере или ноутбуке установлен один из выпусков операционной системы Microsoft Windows (хостовая ОС). Установка же



виртуальной машины, в данном случае VirtualBox, позволяет использовать в среде хостовой ОС любые другие системы (гостевые), включая macOS, Linux, Android, Windows и так далее, вариантов здесь может быть очень много.

Программа была создана компанией Innotek с использованием исходного кода Qemu. Первая публично доступная версия VirtualBox появилась 15 января 2007 года.

10) в феврале 2008 года Innotek был приобретён компанией Sun Microsystems, модель распространения VirtualBox при этом не изменилась. В январе 2010 года Sun Microsystems была поглощена корпорацией Oracle, модель распространения осталась прежней.

### **Ключевые возможности**

Кроссплатформенность

Модульность

Поддержка USB 2.0, когда устройства хост-машины становятся доступными для гостевых операционных систем (только в проприетарной версии)

Поддержка 64-битных гостевых систем (начиная с версии 2.0), даже на 32-битных хост-системах (начиная с версии 2.1, для этого обязательна поддержка технологии виртуализации процессором)

Поддержка SMP на стороне гостевой системы (начиная с версии 3.0, для этого обязательна поддержка технологии виртуализации процессором)

Встроенный RDP-сервер, а также поддержка клиентских USB-устройств поверх протокола RDP (только в проприетарной версии)

Экспериментальная поддержка аппаратного 3D-ускорения (OpenGL, DirectX 8/9 (с использованием кода wine) (только в 32-битных Windows XP, Vista, 7 и 8, для гостевых DOS / Windows 3.x / 95 / 98 / ME поддержка аппаратного 3D-ускорения не предусмотрена)

Поддержка образов жёстких дисков VMDK (VMware) и VHD (Microsoft Virtual PC), включая snapshots (начиная с версии 2.1)

Поддержка iSCSI (только в проприетарной версии)

Поддержка виртуализации аудиоустройств (эмуляция AC97 или SoundBlaster 16 или Intel HD Audio на выбор)

Поддержка различных видов сетевого взаимодействия (NAT, Host Networking via Bridged, Internal)

Поддержка цепочки сохранённых состояний виртуальной машины (snapshots), 18. которым может быть произведён откат из любого состояния гостевой системы

Поддержка Shared Folders для простого обмена файлами между хостовой и гостевой системами (для гостевых систем Windows 2000 и новее, Linux и Solaris)[6]

Поддержка интеграции рабочих столов (seamless mode) хостовой и гостевой операционной системой

Поддержка формата OVF/OVA

Есть возможность выбора языка интерфейса (поддерживается и русскоязычный интерфейс)

Базовая версия полностью открыта по лицензии GNU GPL, соответственно нет ограничений в использовании

### **Пакет дополнений**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

VirtualBox Guest Additions — комплект программного обеспечения,  
устанавливаемый в гостевую операционную систему и расширяющий её возможности



по взаимодействию с системой виртуализации и хост-системой. Например, после установки специального драйвера «виртуальной видеокарты» становится возможным изменять разрешение рабочего стола гостевой ОС произвольным образом вслед за размером окна VirtualBox, в котором запущена виртуальная машина.

До версии 4.0.0 существовало две версии, различавшиеся по лицензии и функциональности. Начиная с 4.0.0 закрытые компоненты вынесены в отдельный пакет дополнений (Extension Pack):

Пакет дополнений содержит закрытые компоненты и распространяется под проприетарной лицензией PUEL (бесплатно только в персональных целях или для ознакомления):

RDP сервер — позволяет подключаться к виртуальной системе удалённо с помощью любого RDP-совместимого клиента;

Поддержка USB — позволяет передавать виртуальной машине USB устройства;

Intel PXE boot ROM — загрузка операционной системы по сети. Используется для создания тонких клиентов/бездисковых рабочих станций.

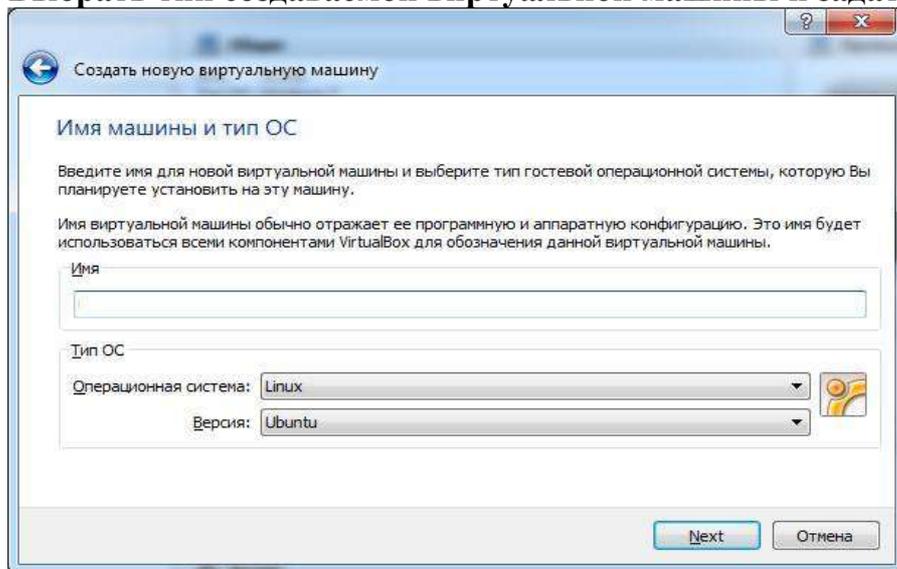
### 1.3.Ход работы:

Установка операционной системы Ubuntu производится на виртуальную машину Oracle VM VirtualBox. Сначала необходимо установить Oracle VM VirtualBox и создать виртуальную машину.

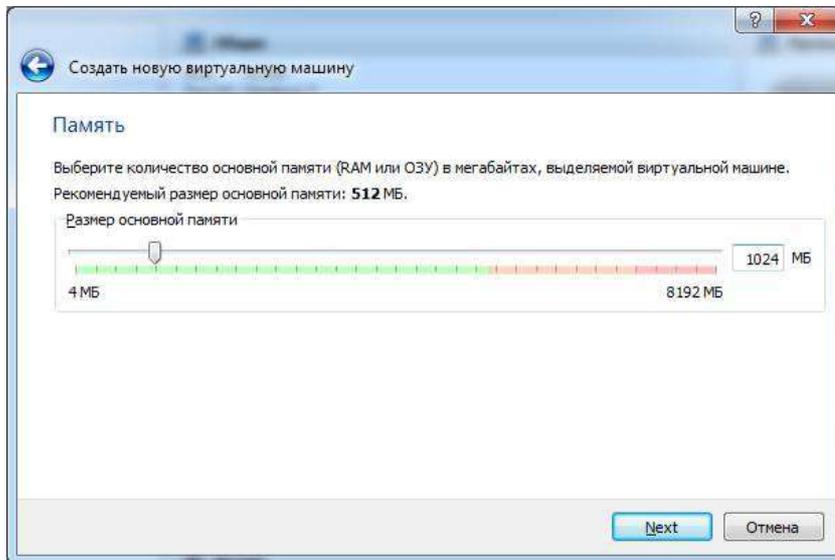
Установка Oracle VM VirtualBox производится с настройками по умолчанию. После установки виртуальной машины установить пакет расширения (Oracle\_VM\_VirtualBox\_Extension\_Pack).

#### 1)Создание виртуальной машины:

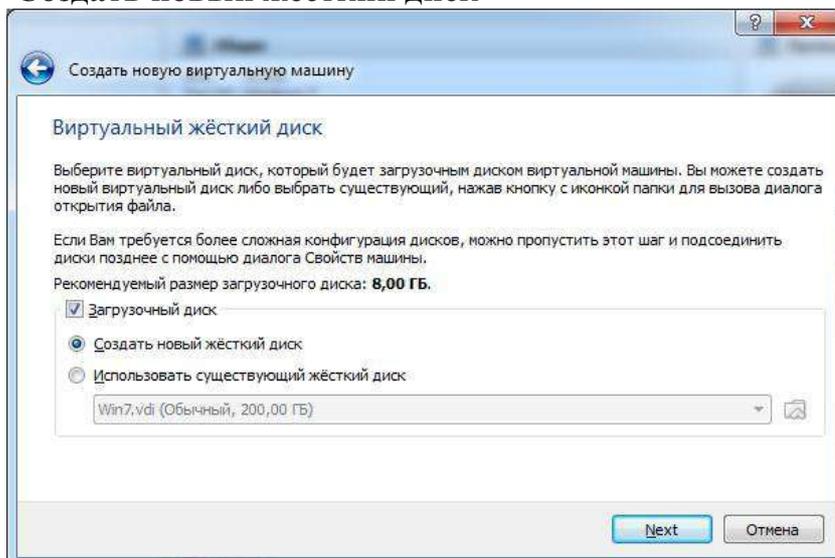
##### Выбрать тип создаваемой виртуальной машины и задать имя.



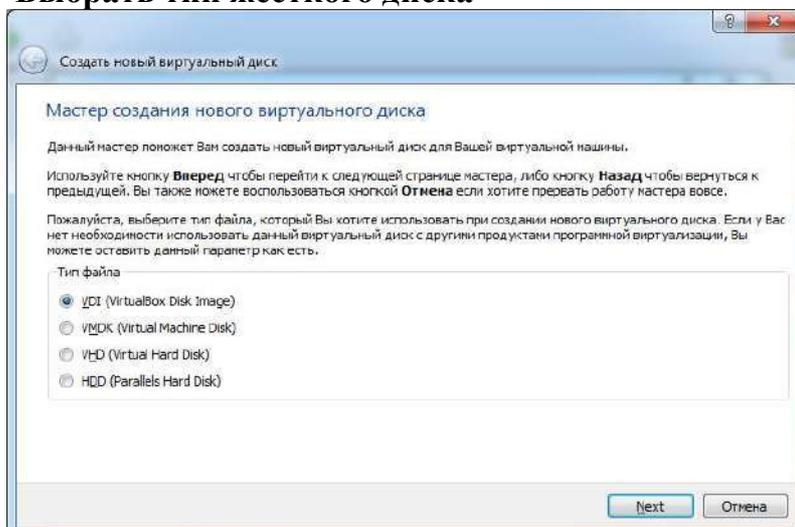
Установить объем памяти 1024 Мб



### Создать новый жесткий диск



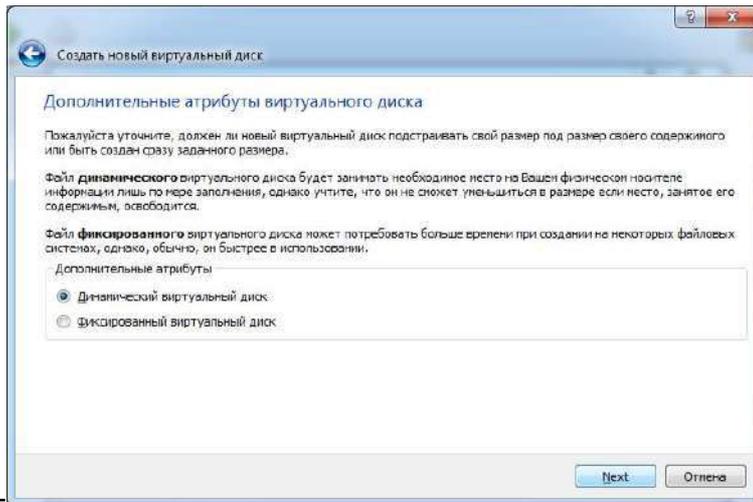
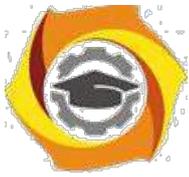
### Выбрать тип жесткого диска



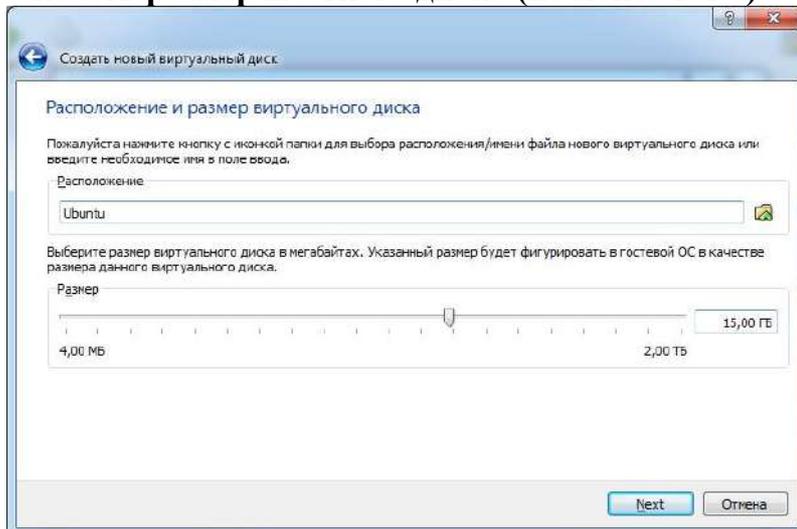


**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

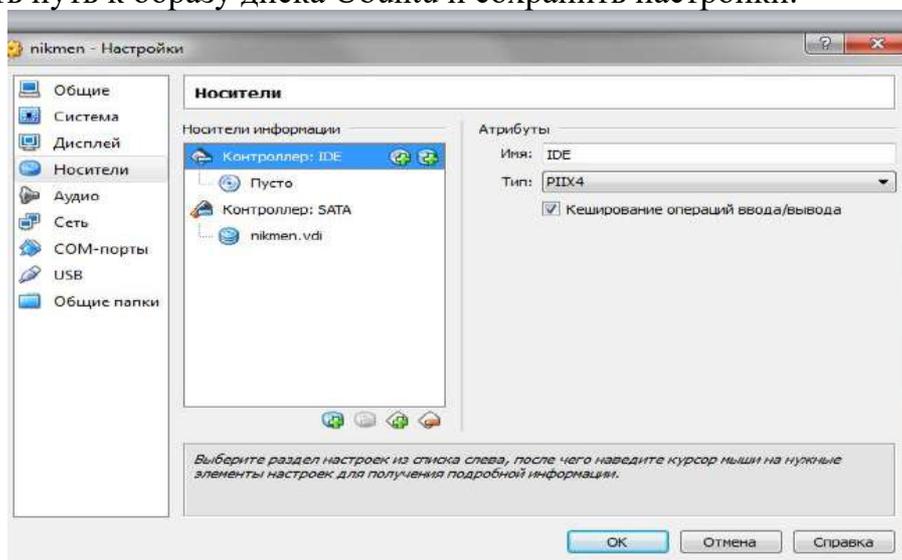
**Выбрать атрибуты жесткого диска**



### Указать размер жесткого диска (не менее 10Гб)



Дождаться установки виртуальной машины. Открыть окно свойств виртуальной машины и перейти в раздел носители. Выбрать IDE контроллер – пусто. Кликнуть по иконки диска в правой панели и выполнить «Выбрать образ загрузочного диска». Указать путь к образу диска Ubuntu и сохранить настройки.

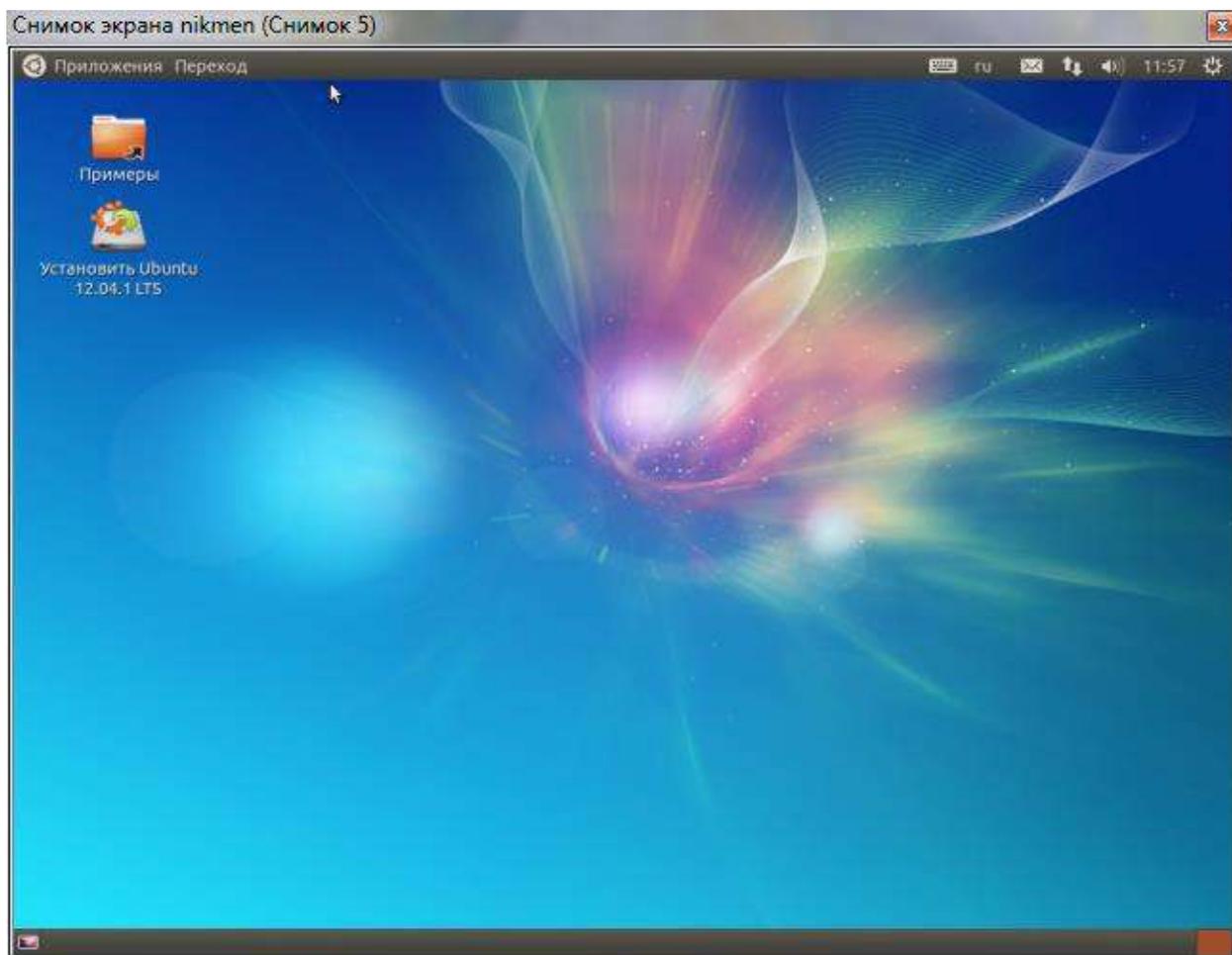
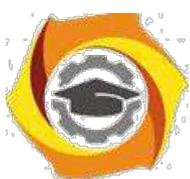




## 2) Запустить виртуальную машину. Выбрать в списке русский язык.



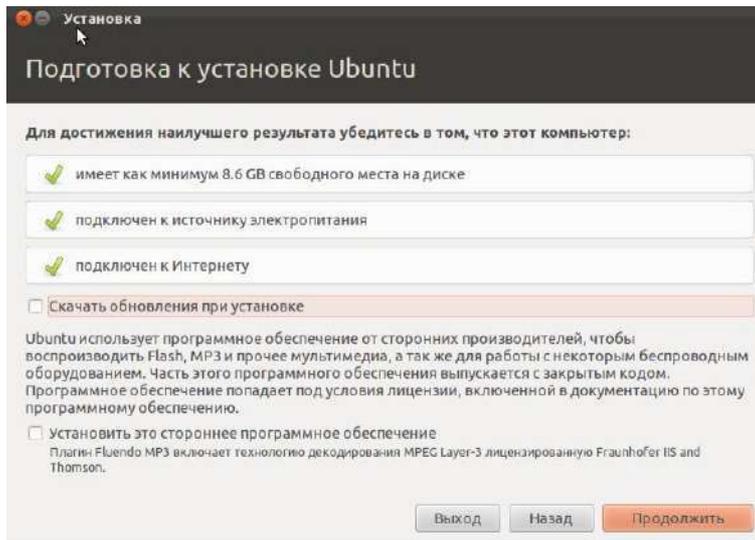
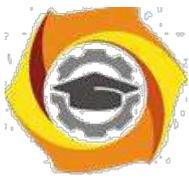
Из предложенных вариантов выбрать «Запустить Ubuntu без установки»  
Дождаться загрузки операционной системы. Запустить установка кликнув по ярлыку «Установить Ubuntu»



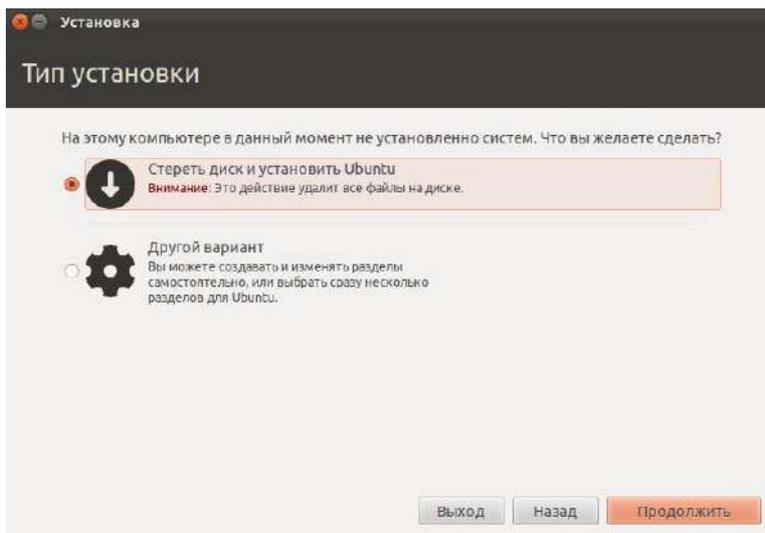
### Выбрать язык установки



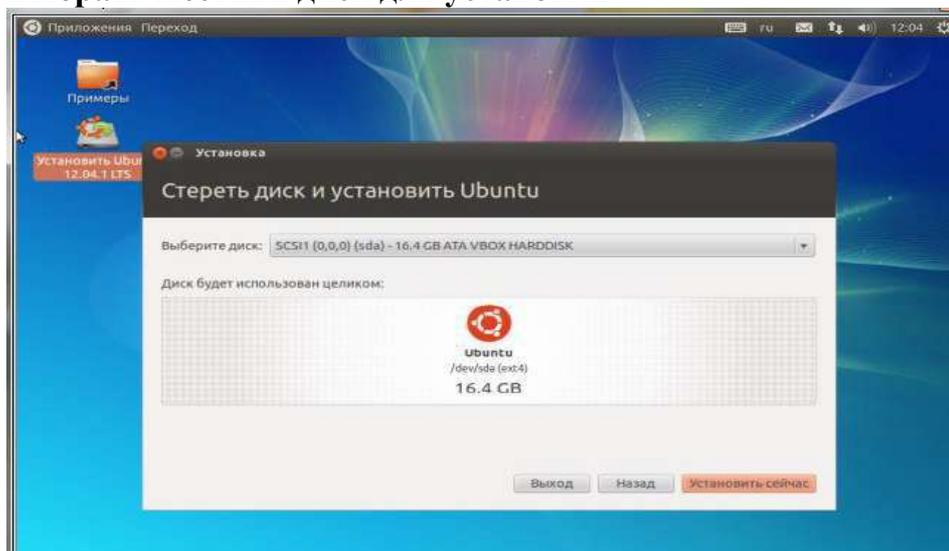
Подтвердить требования к установке



### Выбрать тип установки



### Выбрать жесткий диск для установки





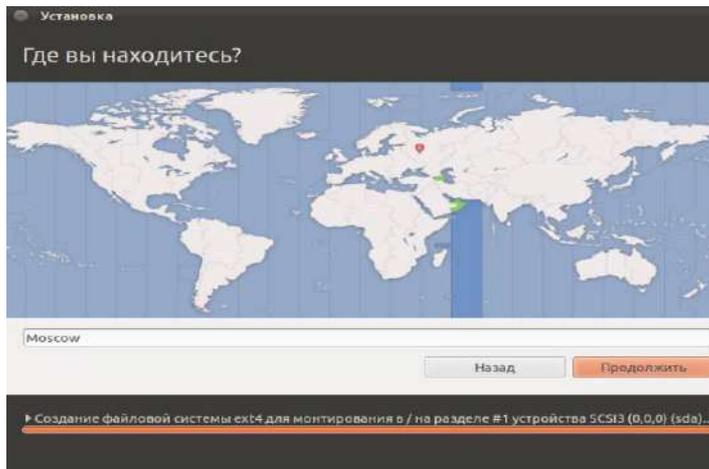
**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

**Указать региональные параметры**

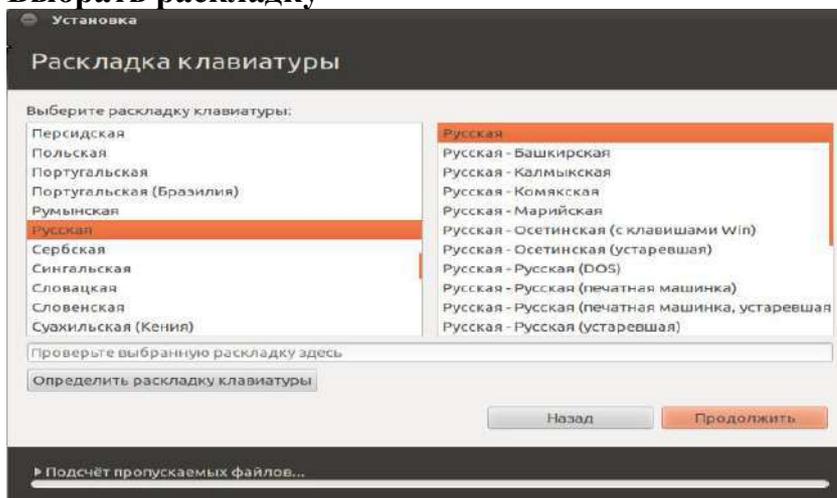


Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»

13

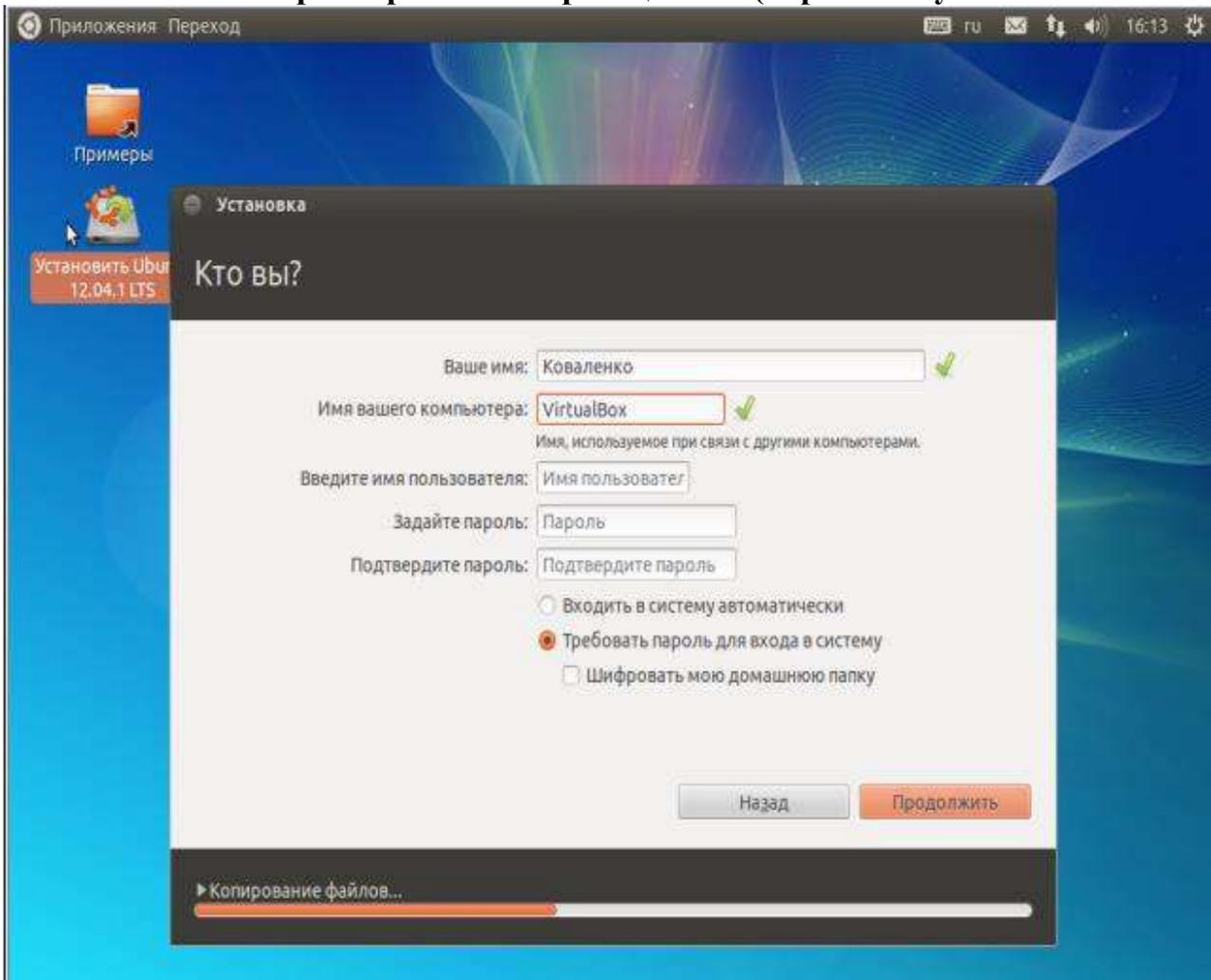


**Выбрать раскладку**





Ввести параметры авторизации (пароль установить 1234)

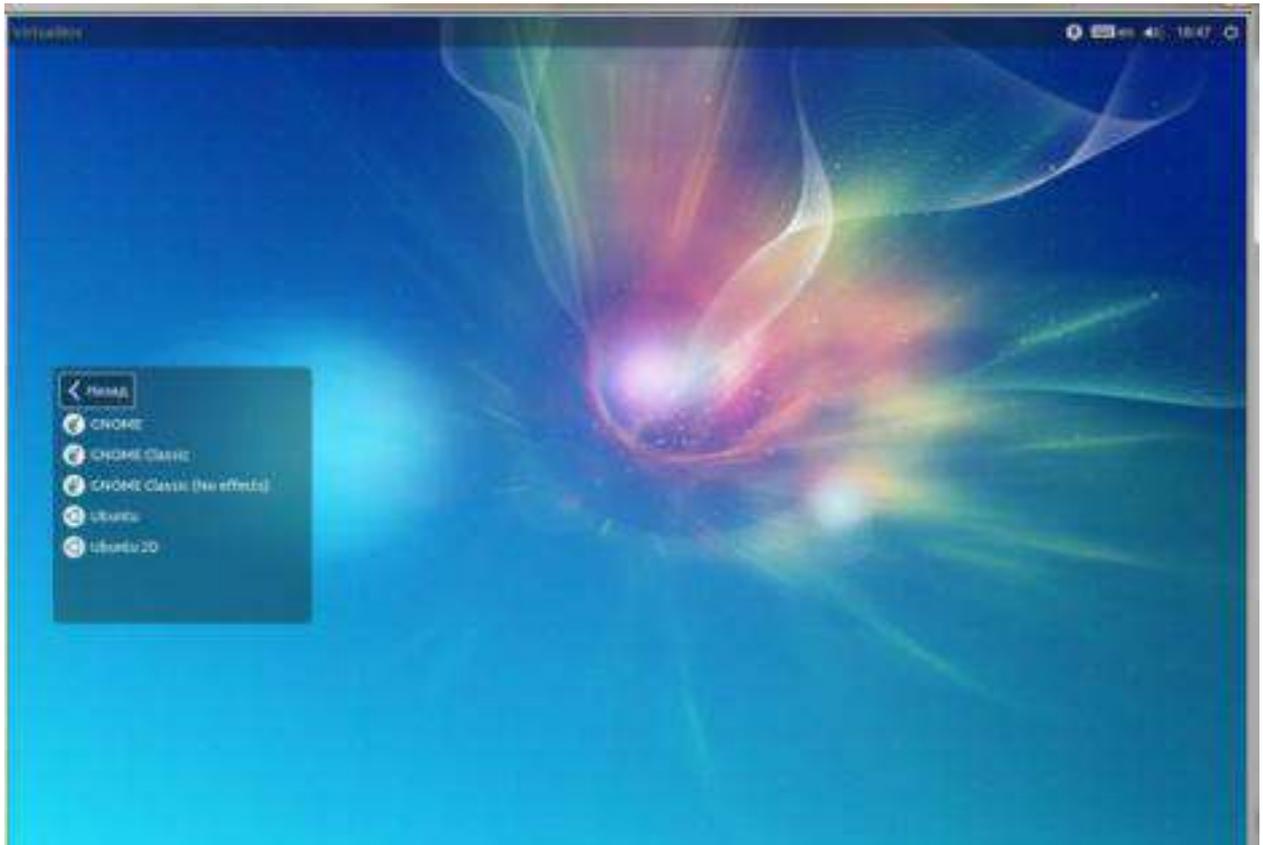


Дождаться установки операционной системы и перезагрузить виртуальную машину для входа в установленную операционную систему.

**3) Установить дополнения гостевой операционной системы:**

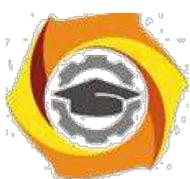
выбрав в меню VirtualBox «Устройства» - «Установить дополнения гостевой ОС». Дождаться установки и перезагрузить операционную систему.

При загрузке определить тип загружаемого сеанса (по умолчанию Ubuntu или Ubuntu 2D)



**Выводы:**

Получен опыт в создании виртуальных машин в **Oracle VM VirtualBox** знакомство с операционной системой **Ubuntu**.



## в Лабораторная работа № 1 Диспетчер задач Windows

Цель занятия: изучение диспетчера задач, его функций и возможностей.

### 2.1. Задание:

- В Изучите теорию, представленную в методичке.
- В Выполните практические задания и ответьте на вопросы.
- В Получите у преподавателя вариант задания.
- В Выполните контрольные задания в соответствии с полученным вариантом.

### 2.2. Краткие теоретические сведения

#### Утилита Диспетчер задач

Диспетчер задач в операционных системах семейства Microsoft Windows – утилита для вывода на экран списка запущенных процессов и потребляемых ими ресурсов (в частности статус, процессорное время и потребляемая оперативная память). Также есть возможность некоторой манипуляции процессами. Другими словами **Диспетчер задач** – это такая специальная программа, которая показывает нам много разной информации о том, что происходит за компьютером. Что именно показывает диспетчер?

Во-первых он показывает какие программы запущены на компьютере в данный момент.

Во-вторых он показывает какие процессы сейчас протекают на компьютере.

В-третьих – сколько системных ресурсов занимает каждый запущенный процесс и какие ресурсы компьютера процесс занимает.

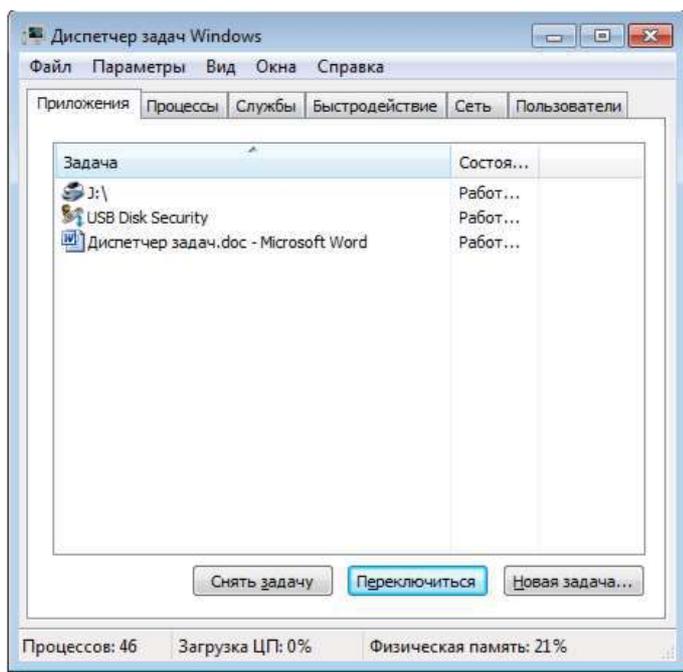
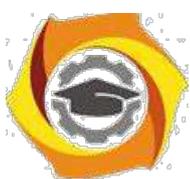


Рисунок 1 – Окно программы Диспетчер задач Windows

## Способы запуска диспетчера задач

Для запуска диспетчера задач выполните любое из следующих действий:

- и Щелкните правой кнопкой мыши по пустой области на панели задач и выберите команду Запустить диспетчер задач.

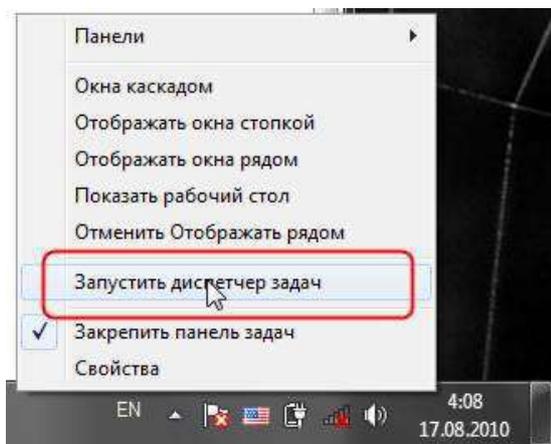


Рисунок 2 – Окно запуска диспетчера задач



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**В** Нажмите клавиши Ctrl+Shift+Esc.



В Нажмите клавиши Ctrl+Alt+Delete и выберите опцию запустить Диспетчер задач.

После нажатия этих кнопок система выдаст вам довольно обширное меню команд:

- Блокировка компьютера;
- Смена пользователя;
- Завершение сеанса;
- Смена пароля;
- Ссылка же на Диспетчер Задач находится в самом низу этого списка.

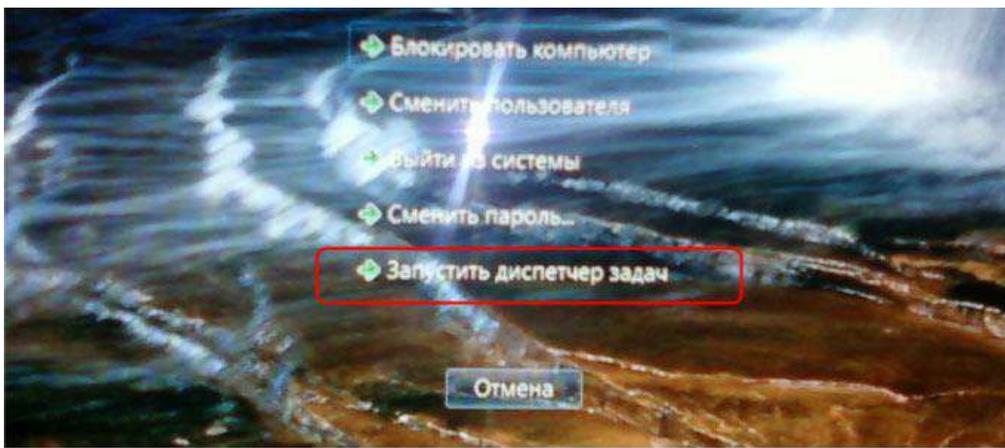
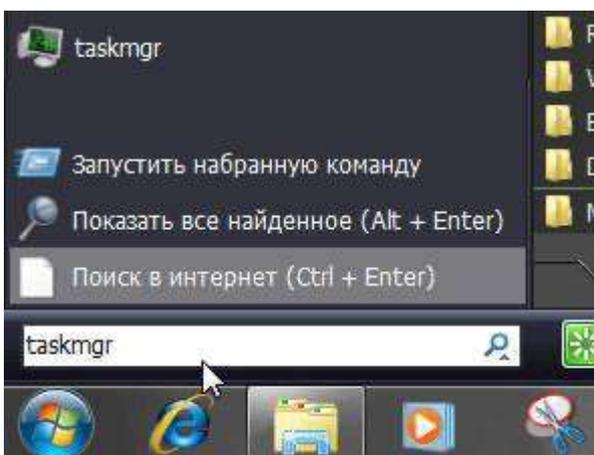


Рисунок 3 – Меню команд

В Нажмите кнопку Пуск, введите в поле Найти слово taskmgr и нажмите клавишу Enter.





**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*Рисунок 4 – Поле «Найти»*



Окно Диспетчер задач Windows под основным меню (**Файл, Параметры, Вид, Окна, Завершение работы, Справка**) содержит 6 вкладок, каждая из которых представляет полезную информацию:

6. Вкладка *Приложения*. Здесь можно найти список запущенных приложений и их состояние.
7. Вкладка *Процессы*. Перечислены все программы и процессы, запущенные в системе, – это основное окно для прекращения работы «зависших» программ или процессов.
8. Вкладка *Службы*. Содержит список программ, которые работают в фоновом режиме.
9. Вкладка *Быстродействие*. Основная вкладка для оценки производительности операционной системы.
10. Вкладка *Сеть*. Отображает объём передаваемых по локальной сети данных.
11. Вкладка *Пользователи*. С помощью этой вкладки можно увидеть список всех пользователей, подключенных к вашему компьютеру по локальной сети. В противном случае будет указана только одна учётная запись пользователя.

### **Вкладка «Приложения»**

Переходим на неё, нажимая левой кнопкой мыши. Перед нами список запущенных приложений и два столбца: столбец «Задача» и столбец «Состояние».

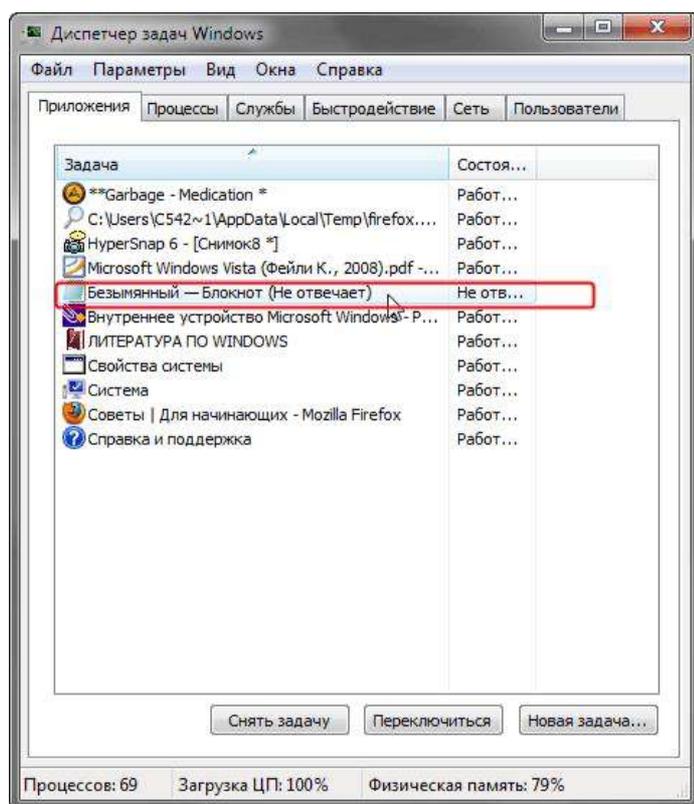
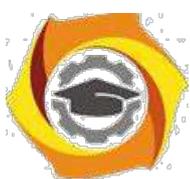


Рисунок 5 – Вкладка «Приложения»

Ищем приложение с состоянием «Не отвечает», либо «Не работает», зависит от Windows. Выделяем приложение с таким состоянием, щёлкая левой кнопкой мыши по названию в столбце «Задача».

Нажимаем на кнопку «Снять задачу». Если из списка программа сразу исчезла, то это означает, что она завершилась.

В основном зависшая программа работает некорректно и её придётся завершить вынужденно. В этом случае появится окошко с вопросом: «Завершить сейчас?», нажимаем на кнопку «Завершить сейчас».

Все несохранённые изменения, выполненные в этом приложении, будут потеряны. Если несохранённые данные важны для пользователя, есть смысл дождаться отклика программы.

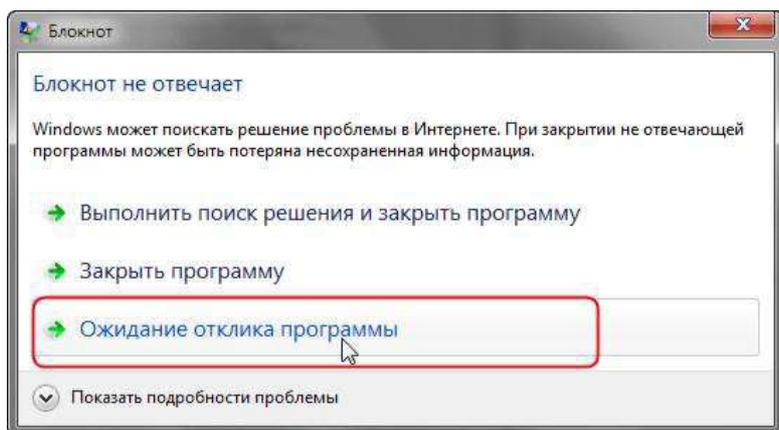
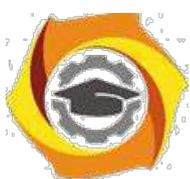


Рисунок 6 – Ожидание отклика программы

Программы, которые зависли/дали сбой, в среде Windows называются «неотвечающими»; пользователь может поместить курсор мыши в окно программы, но она не будет реагировать на щелчки мыши или нажатия клавиш. Если программа не отвечает, это не значит, что пользователь должен перезагружать компьютер. Воспользуйтесь Диспетчером задач и закройте зависшую программу. Перед тем как закрывать программу, убедитесь в том, что она действительно не отвечает. Подождите некоторое время; возможно, операционная система Windows пытается выделить для программы дополнительные ресурсы памяти. Например, если пользователь запускает макрос Visual Basic в программе Microsoft Excel или Word, ему может показаться, что программа «зависла». Сложное изменение форматирования или выполнение операций по поиску и замене в объёмном документе могут создать впечатление того, что текстовый редактор «не отвечает». Открытое диалоговое окно или окно сообщения могут не позволить пользователю выполнять никакие действия в определенной программе; поищите их под текущим окном.

Если какая-либо программа часто «зависает», то можно попробовать найти решение этой проблемы в интернете, создав отчет об ошибке.

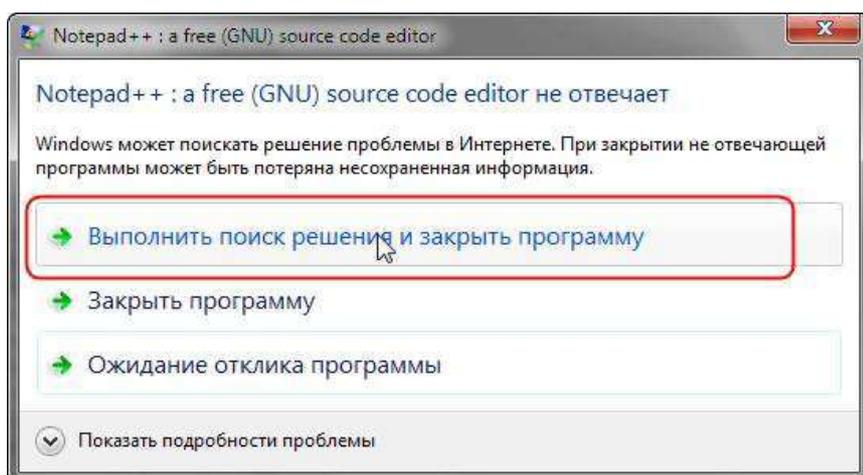




Рисунок 7 – Выполнить поиск решения и закрыть программу

Отчеты об ошибках Windows можно использовать, чтобы сообщать в корпорацию Майкрософт о проблемах на компьютере. Корпорация Майкрософт использует отчеты о проблемах для поиска решений, соответствующих описаниям проблем. В ОС Windows отображается уведомление при наличии возможных решений проблемы, также о возможности поиска дополнительных сведений в Центре поддержки. Если же решения нет, то сведения, отсылаемые в сообщении о проблеме, могут помочь Microsoft найти или создать новое решение.

Если решение проблемы, о которой сообщил компьютер, существует, оно появится в Центре поддержки.

Кнопка Переключиться – откроет выделенную программу.

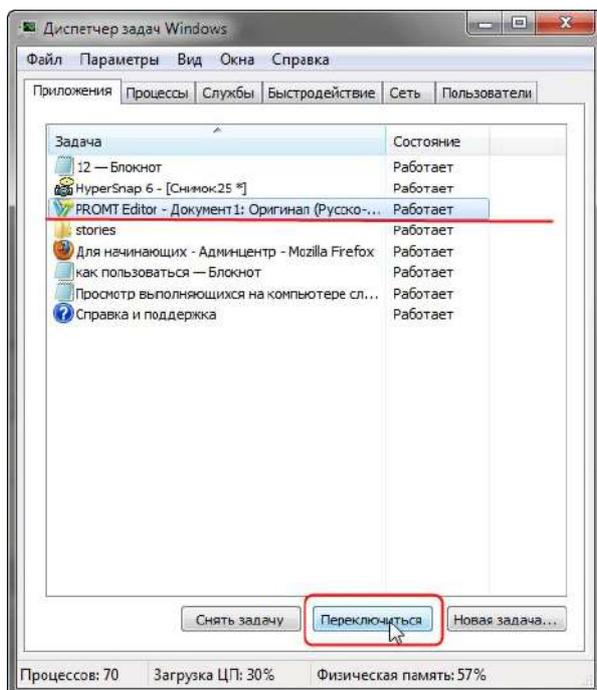
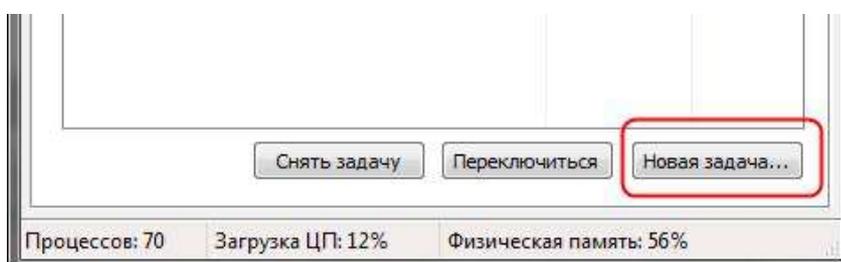


Рисунок 8 – Кнопка переключиться

Кнопка Новая задача – откроет диалоговое окно Выполнить. С помощью команды Выполнить можно быстро запускать программы, открывать файлы и папки, а также переходить на веб-сайты, если компьютер подключен к Интернету.



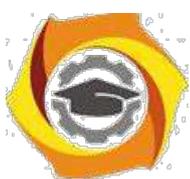


Рисунок 9 – Новая задача

и каждой программой, выполняемой на компьютере, связан определенный процесс, который запускает эту программу. Если программа перестает отвечать (или «зависает»), определение связанного с этой программой процесса может помочь в устранении возникших неполадок. Например, если известно, какой процесс используется для запуска данной программы, то чтобы закрыть зависшую программу, необходимо завершить этот процесс.

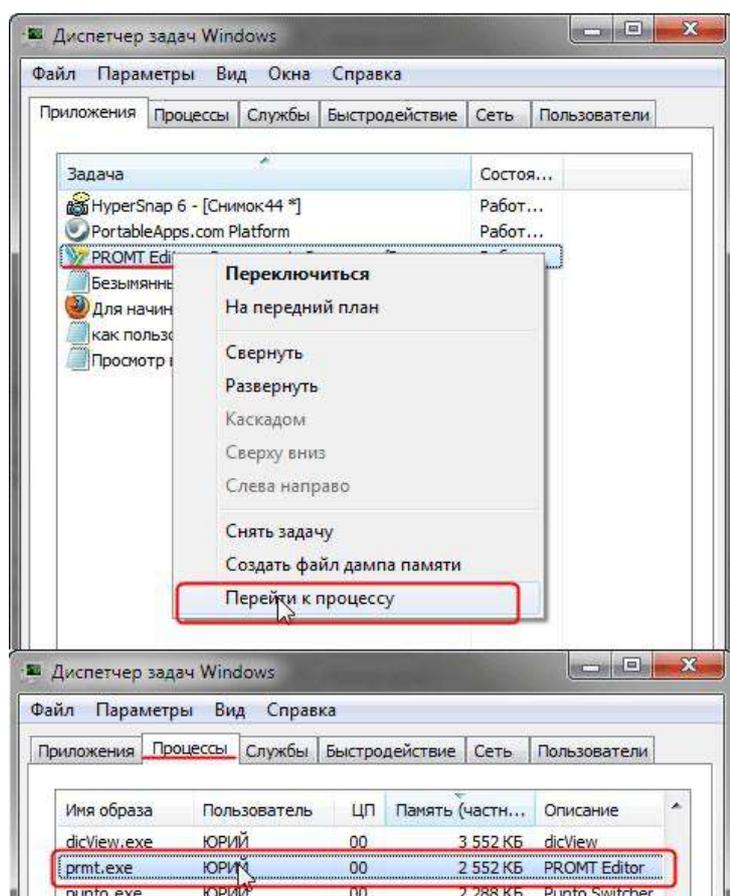


Рисунок 10 – Перейти к процессу

Чтобы определить, какой процесс используется программой, щелкните правой кнопкой мыши нужную программу и выберите команду Перейти к процессу. Связанный с данной программой процесс будет выделен на вкладке Процессы.

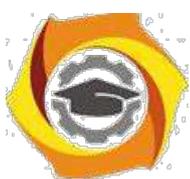
### Вкладка «Процессы»

Для просмотра сведений о процессах, выполняющихся в данный момент на компьютере, можно использовать **Диспетчер задач**.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Процесс** – это файл, например исполняемый файл, имя которого заканчивается расширением EXE. Этот файл используется компьютером для непосредственного запуска программ и других служб в специально выделенной для него области



оперативной памяти. Каждое запущенное приложение имеет соответствующий ему процесс.

Перейдите на вкладку **Процессы**. В диспетчере задач отображаются процессы, выполняющиеся в данный момент под текущей учётной записью пользователя.

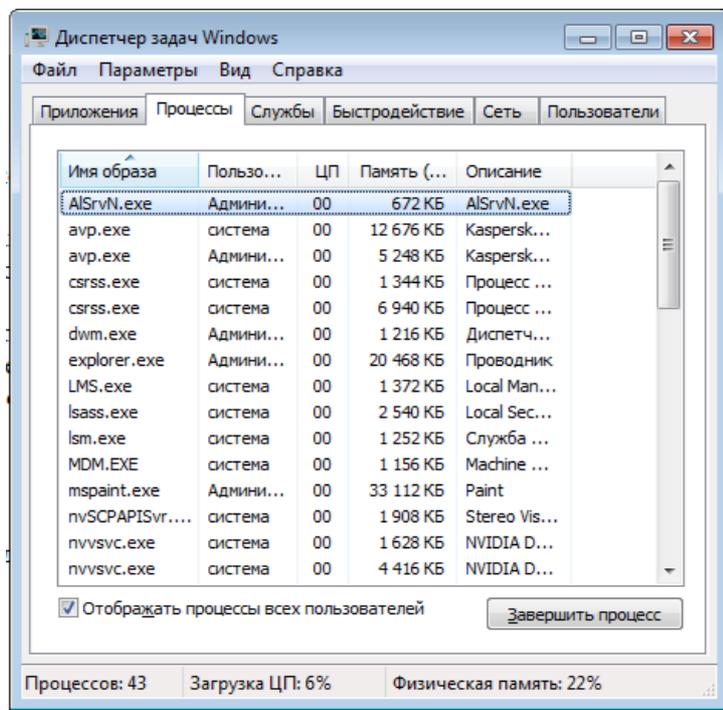


Рисунок 11 – Вкладка «Процессы»

Сколько процессов выполняется на данный момент времени, можно увидеть в нижней части диспетчера.

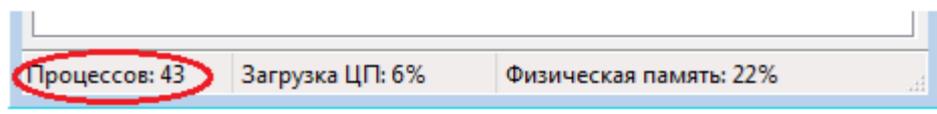
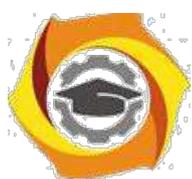


Рисунок 12 – Наличие процессов

- действительности процессов может быть намного больше. Где остальные процессы и почему они скрыты? Всё дело в том, что по умолчанию отображаются лишь процессы, на которые может влиять пользователь. Стоит лишь установить флажок *Отображать процессы всех пользователей*, как в основном окне появятся все процессы, поскольку будут добавлены процессы с атрибутами SYSTEM (системные процессы), NETWORK (сетевые процессы) и LOCAL SERVICE (локальные службы). Останавливать выполнение этих процессов не рекомендуется, поэтому не устанавливайте данный флажок без необходимости.

По умолчанию для каждого процесса отображается следующая информация:



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- *Имя образа* – название процесса;



- и *Пользователь* – имя пользователя, запустившего процесс;
- и *ЦП* – процент мощности процессора, используемый процессом;
- и *Память (частный рабочий набор)* – объём памяти, используемый процессом в ходе работы.

### Распространённые процессы:

**svchost.exe** – это главный системный процесс для тех служб, которые запускаются из динамически загружаемых библиотек (DLL-файлов). И действительно несколько экземпляров процесса svchost.exe могут быть запущены одновременно (но с разными PID\* (PID — это идентификатор пакета.)). Так как каждый из таких экземпляров представляет собой определённую преимущественно системную службу или же группу служб. Эти группы определены в следующем разделе реестра:

**HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\NTCurrentVersion\SvcHost**

Каждое значение в этом разделе представляет отдельную группу Svchost и отображается при просмотре активных процессов как отдельный экземпляр. Каждое из этих значений имеет тип REG\_MULTI\_SZ и содержит службы, выполняемые в этой группе Svchost. Каждая группа Svchost может содержать одно или несколько имен служб, извлекаемых из следующего раздела реестра, в котором подраздел Parameters содержит значение ServiceDLL:

**HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\Служба**

Чтобы просмотреть список служб, работающих в процессе Svchost, выполните описанные ниже действия.

- Нажмите на панели задач Windows кнопку Пуск и выберите пункт Все программы, затем Стандартные, затем Монитор.

2. Введите команду Tasklist /SVC и нажмите клавишу ENTER.

Команда Tasklist выводит список активных процессов



```
C:\WINDOWS\system32\CMD.exe
AvastUI.exe 1320 H/П
Bonus_ScreenshotReader.exe 1328 H/П
igfxsrvc.exe 1356 H/П
SSMMgr.exe 1780 H/П
2GISTrayNotifier.exe 1844 H/П
ctfmon.exe 1216 H/П
AtomicAlarmClock.exe 1012 H/П
LouderIt.exe 1904 H/П
URSAver.exe 1968 H/П
spoolsv.exe 1220 Spooler
svchost.exe 1432 WebClient
NetworkLicenseServer.exe 2064 ABBYY.Licensing.FineReader.Professional.10.0
PnkBstrA.exe 2188 PnkBstrA
svchost.exe 2260 stisvc
UTSCSI.EXE 2280 UTSCSI
WINWORD.EXE 3948 H/П
opera.exe 3564 H/П
cmd.exe 548 H/П
tasklist.exe 388 H/П
wmiprvse.exe 2512 H/П

C:\Documents and Settings\Administrator>
```

Рисунок 13 – Список активных процессов

Параметр /SVC используется для вывода списка активных служб в каждом процессе. Для получения дополнительных сведений о процессе введите следующую команду и нажмите клавишу ENTER:

***Tasklist /FI «PID eq идентификатор\_процесса» (кавычки обязательны).***

**DLL** (англ. Dynamic-link library – динамически подключаемая библиотека) – понятие операционных систем Microsoft Windows и IBM OS/2; динамическая библиотека, позволяющая многократное применение различными программными приложениями. К DLL относятся также элементы управления ActiveX и драйверы. В мире UNIX аналогичные функции выполняют т. н. shared objects («разделяемые объекты»).

### Цели введения DLL

Первоначально предполагалось, что введение DLL позволит эффективно организовать память и дисковое пространство, используя только один экземпляр библиотечного модуля для различных приложений. Это было особенно важно для ранних версий Microsoft Windows с жёсткими ограничениями по памяти. Далее, предполагалось улучшить эффективность разработок и использования системных средств за счёт модульности. Замена DLL–программ с одной версии на другую должна была позволить независимо наращивать систему, не затрагивая приложений. Кроме того, библиотеки DLL могли использоваться разнотипными приложениями — например, Microsoft Office, Microsoft Visual Studio и т. п.

- дальнейшем идея модульности выросла в концепцию COM (Component Object Model – объектная модель компонентов, компьютерная технология, разработанная компанией Microsoft.)



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Фактически, полных преимуществ от внедрения DLL (Dynamic-link library – библиотека динамической компоновки) получить не удалось по причине явления, называемого DLL hell («ад DLL»). DLL Hell возникает, когда несколько приложений



требуют одновременно различные, не полностью совместимые, версии DLL–библиотек, что приводит к сбоям в этих приложениях. Когда система выросла до определённых размеров, количество DLL стало превышать многие тысячи килобайт не все из них обладали полной надёжностью и совместимостью, и конфликты типа DLL Hell стали возникать очень часто, резко понижая общую надёжность системы. Поздние версии Microsoft Windows стали разрешать параллельное использование разных версий DLL, что уничтожило преимущества изначального принципа модульности.

**Службы Windows** (англ. Windows Service, сервисы) – приложения, автоматически запускаемые системой при запуске Windows и выполняющиеся вне зависимости от статуса пользователя. Имеет общие черты с концепцией. В процессе загрузки на основании записей в реестре Svchost.exe составляет список служб, которые необходимо запустить. Одновременно могут быть запущены несколько экземпляров процесса Svchost.exe. Каждый сеанс Svchost.exe содержит группу служб, следовательно, отдельные службы могут выполняться в зависимости от того, как и когда был запущен Svchost.exe. Таким образом улучшается контроль и упрощается отладка. Файл Svchost.exe расположен в папке %SystemRoot%\System32. В других каталогах под именем Svchost.exe может скрываться вирусы (Троянская программа, вирус или сетевой червь). Наиболее известные злонамеренные программы, скрываются под именем системного процесса Svchost.exe – W32.Welchia.Worm, W32/Jeefo и W32.Assarm@mm. В этом случае злонамеренный процесс должен быть немедленно завершён.

**csrss.exe** – процесс управляет отображением окон в Windows. Если пользователь откроет новое окно, доступ к нему обеспечит csrss.exe. Одновременно csrss.exe отвечает и за управление другими процессами. Если пользователь обнаружил более двух записей csrss.exe в списке Диспетчера задач и эти файлы сильно загружают процессор, это означает что за этим скрывается вредоносная программа. В этом случае выполните проверку вашего компьютера с помощью программ Антивирус Касперского или Dr.Web.

**smss.exe** – процесс отвечает за запуск сеансов пользователей. Данный процесс запускается системным потоком и отвечает за различные действия, в частности, за запуск процессов winlogon и win32 (csrss.exe) и за установку системных переменных. После того как указанные процессы запущены, процесс smss ожидает завершения работы winlogon или csrss. Если это происходит «нормально», система может завершить свою работу; если же это происходит неожиданным образом, smss.exe вызывает зависание системы (система перестаёт реагировать на запросы).



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**lsass.exe** – процесс отвечает за программы и настройки безопасности Windows и поэтому очень важен. lsass.exe очень часто является целью хакерских атак. Поэтому следует внимательно следить за этим процессом в Диспетчере задач. Если lsass.exe



заражён, не стоит самостоятельно пытаться завершить этот процесс или удалить файл. С этой задачей прекрасно справится Антивирус Касперского, он в состоянии вылечить файлы. Многие вирусы используют для маскировки маленькую хитрость – в Списке Диспетчера задач в их имени вместо прописной буквы «i» используется заглавная буква «I».

**explorer.exe** – процесс отвечает, в частности, за отображение Рабочего стола и содержания дисков. Очень часто можно встретить вирусы или «программы-шпионы», которые используют имя explorer.exe. Подобные вредители и «настоящий» файл, как правило, находятся в разных папках.

**ctfmon.exe** – процесс управляет технологиями альтернативного ввода данных. Он запускает языковую панель в системе при старте операционной системы, и работает в фоновом режиме даже после закрытия всех программ пакета Microsoft Office, независимо от того, запускались ли программы Office XP.

**avp.exe** – процесс, который зарегистрирован в качестве Kaspersky Anti-

Virus **vmsrvc.exe** – Virtual Machine Additions Services Application

**vmacthlp.exe** – Virtual Machine Additions службы общего доступа к папкам. Работа в Microsoft Virtual PC или Virtual Server и установлена виртуальная машина дополнений, которые обеспечивают интеграцию виртуальный ПК / сервер с операционной системой. Эта услуга специально реализует Virtual Machine Additions функция, которая позволяет Virtual PC, / Server, чтобы использовать папки с хост-компьютера (локальные и сетевые папки) для совместного использования и обмена файлами между ПК хозяина и Virtual PC, / Server.

**spoolsv.exe** – процесс отвечает за обработку процессов печати на локальном компьютере в операционных системах Microsoft Windows. Служба spooler ответственна за управление заданиями на печать и передачу факсимильных сообщений.

**taskmgr.exe** – собственно менеджер задач.

**services.exe** – служба Plug & Play – Позволяет компьютеру распознавать изменения в установленном оборудовании и подстраиваться под них, либо не требуя вмешательства пользователя, либо сводя его к минимуму. Остановка или отключение этой службы может привести к нестабильной работе системы. Этот процесс запускает первый svchost.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**system** – большинство системных потоков режима ядра исполняются от имени процесса system.



**winlogon.exe** – процесс управляет входом пользователей в систему и выходом из неё. Winlogon активируется только при нажатии клавиш CTRL+ALT+DEL, после чего появляется окно «Безопасность Windows».

**wmiprvse.exe** – инструментарий Windows, обеспечивает обмен управляющей информацией с устройствами.

**Бездействие системы** – процесс имеет по одному потоку на каждом процессоре и его единственная задача – учитывать время, в течение которого система не занята другими потоками. В диспетчере задач можно видеть, что этому процессу, как правило, соответствует большая часть процессорного времени.

**Все процессы, кроме указанных выше, можно удалять, не опасаясь за работу системы. Однако некоторые из них возражают против такого действия (вирусы, вредоносные программы и некоторые антивирусы), в этом случае удаляем из автозапуска лишние программа с помощью AnVir Task Manager и перезапускаем систему.**

### Вкладка «Службы»

Диспетчер задач можно использовать для просмотра сведений о службах, выполняющихся в данный момент на компьютере. Перейдите на вкладку Службы, чтобы увидеть службы, выполняющиеся в данный момент под текущей учетной записью пользователя.

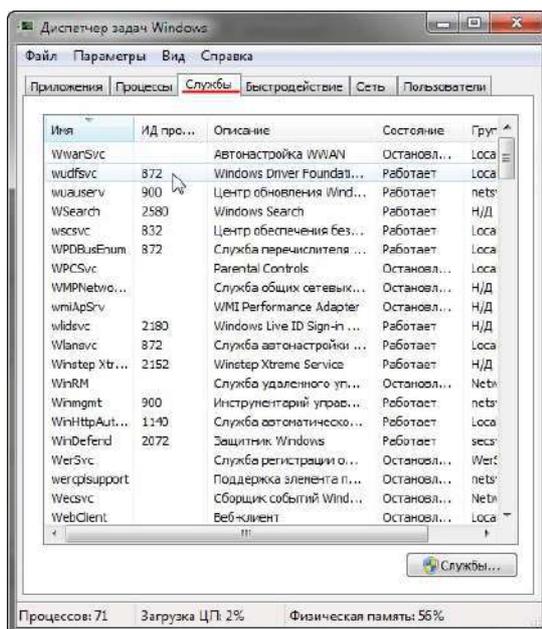
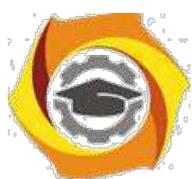


Рисунок 14 – Вкладка «Службы»



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Можно также просматривать сведения о процессах, связанных с конкретной службой.



Чтобы просмотреть процесс, связанный со службой, щелкните правой кнопкой мыши требуемую службу и выберите команду Перейти к процессу. Если команда Перейти к процессу отображается затененной, значит выбранная служба в данный момент остановлена. Состояние службы (выполняется или остановлена) отображается в столбце Статус.

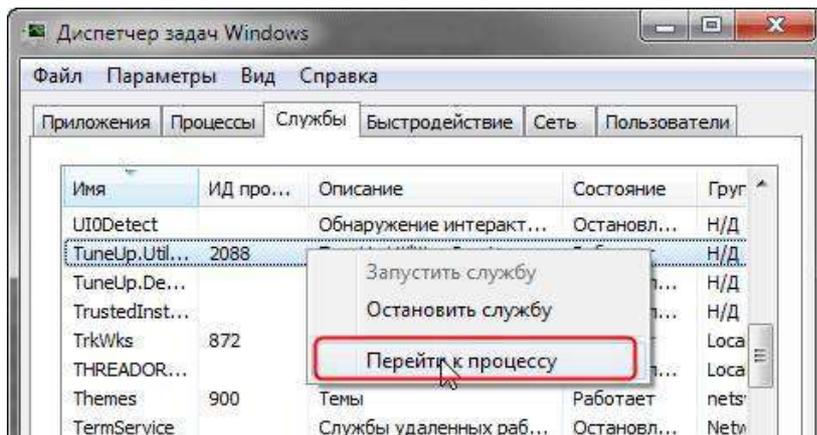


Рисунок 15 – Процесс связанный со службой

Если после выбора команды Перейти к процессу процесс не выделен на вкладке Процессы, значит он не выполняется под текущей учетной записью пользователя. Чтобы просмотреть все процессы, перейдите на вкладку Процессы и установите флажок Отображать процессы всех пользователей. При появлении запроса пароля администратора или подтверждения введите пароль или предоставьте подтверждение. Перейдите на вкладку Службы и повторите попытку просмотреть сведения о процессе.

При нажатии кнопки Службы в нижней части вкладки Службы откроется оснастка консоли управления (MMC), где опытные пользователи могут просматривать более подробные сведения о службах и настраивать дополнительные параметры.

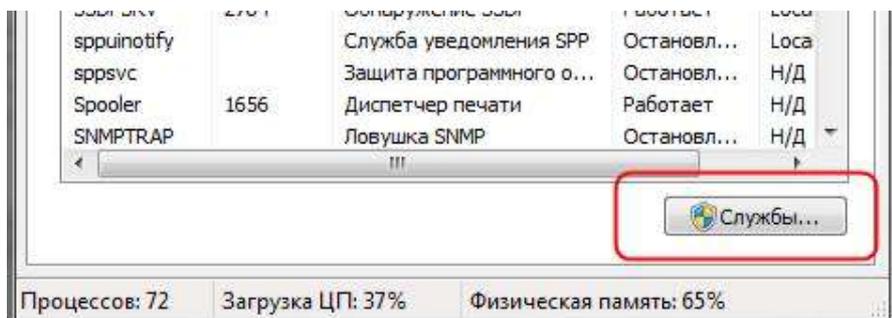


Рисунок 16 – Кнопка Службы

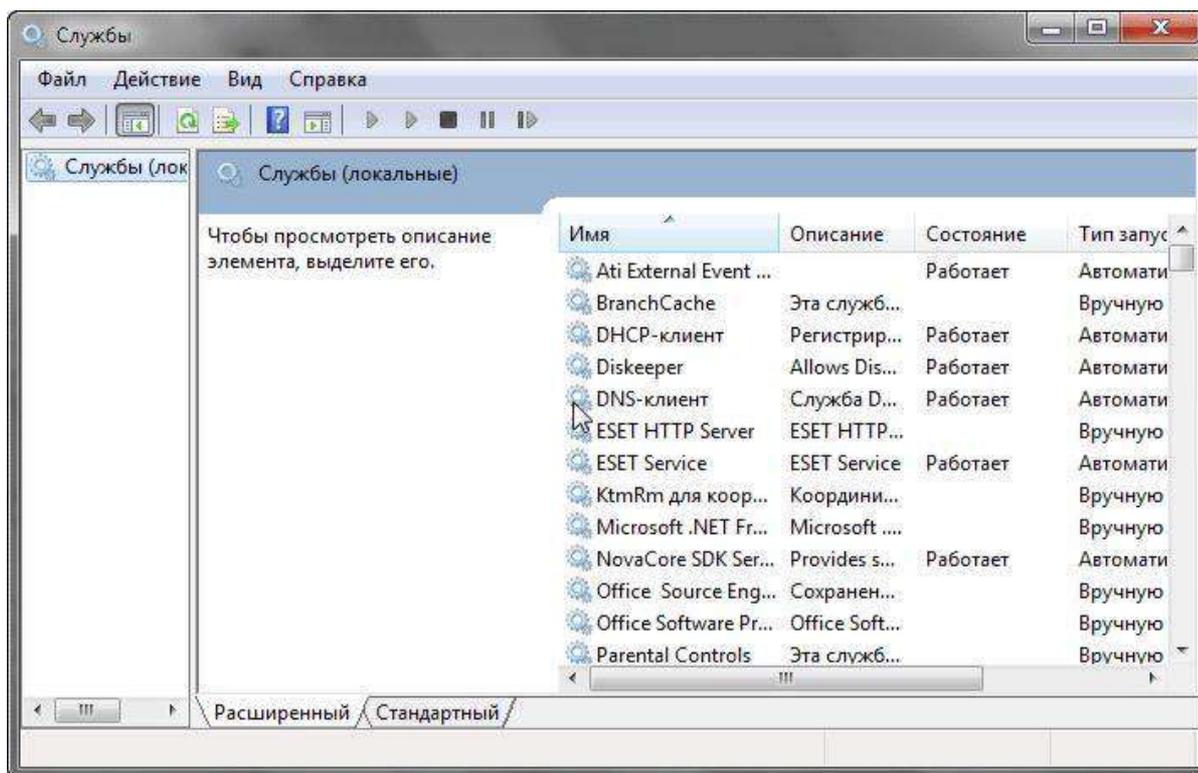
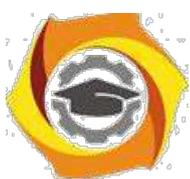


Рисунок 17 – Подробные сведения о  
службах Вкладка «Быстродействие»

На вкладке Быстродействие диспетчера задач приведены дополнительные сведения об использовании компьютером системных ресурсов, например памяти (ОЗУ) и ресурсов центрального процессора (ЦП).

На этой вкладке отображаются четыре графика. Два графика вверху показывают загрузку ЦП как в текущий момент, так и за несколько последних минут.

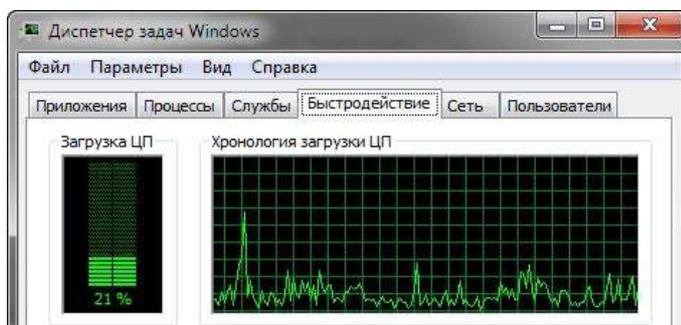
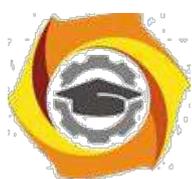


Рисунок 18 – Вкладка «Быстродействие»

Высокое процентное значение показывает, что программы или процессы используют большой объем ресурсов процессора, что может замедлить работу



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

компьютера. Приложение может не отвечать, когда процентный показатель фиксируется или приближается к значению 100%.



Процентное значение загрузки процессора показано так же в нижней части диспетчера задач.

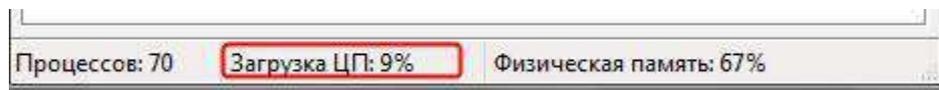


Рисунок 19 – Загрузка ЦП

Два графика внизу показывают объём используемого ОЗУ, или физической памяти, в мегабайтах (МБ) как в текущий момент, так и за несколько последних минут.



Рисунок 20 – Файл подкачки и хронология использования файла подкачки

Самая частая причина замедления работы системы под управлением Windows – заполнение физической памяти. При этом Windows начинает так называемую «подкачку» (paging) – перемещение блоков кода и данных программ (каждый такой блок называется страницей – page) из физической памяти на жесткий диск. Обращение к файлу подкачки время от времени – нормальное явление, не ухудшающее производительность системы, но частые запросы данных из файла на диске могут заметно снизить общую скорость работы системы. Эта проблема становится особенно заметной при переключении между несколькими программами, активно использующими память, на компьютере, который не содержит достаточного количества физической памяти. В результате диск почти постоянно находится в работе, потому что система пытается «перекачать» данные с него в память и обратно.

Самый быстрый способ получить информацию об использовании памяти в данный момент – запустить Диспетчер задач Windows и взглянуть на строку состояния внизу любой вкладки. Статистика использования памяти приведена в правой части вкладки С в виде двух чисел, а точнее дроби. Первое число (числитель) представляет собой текущий объём выделенной памяти – количество физической и виртуальной памяти, используемой всеми выполняемыми процессами. (*Виртуальная память, собственно, и есть файл подкачки.*) Знаменатель – общее количество доступной памяти (физической и виртуальной). Само по себе это число способно лишь предупредить вас о том, что память скоро закончится совсем, – другими словами, выделенная память примерно совпадает с доступной.



Рисунок 21 – Физическая память

Для того чтобы узнать об использовании памяти подробнее, переключитесь на вкладку Быстродействие и взгляните на таблицы в нижней части диалогового окна. Учтите: числа и подписи к ним могут означать абсолютно не то, что видно изначально.

- трех дополнительных таблиц под графиками содержатся сведения об использовании памяти и ресурсов.

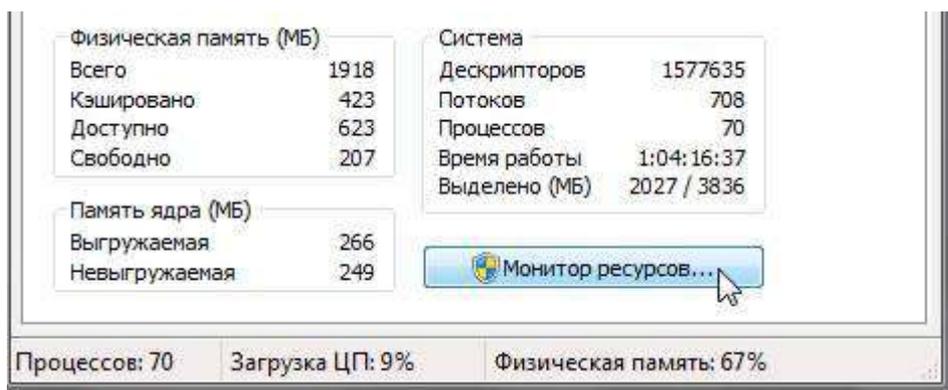


Рисунок 22 – Сведения об использовании памяти и ресурсов

Под заголовком Физическая память (МБ) значение:

- **Всего** — то объём в мегабайтах (Мбайт) оперативной памяти, установленной на компьютере.
- **Кэшировано** — это объём физической памяти, использованной за последнее время для системных ресурсов.
- **Доступно** — это объём памяти, непосредственно доступный для использования процессами, драйверами и операционной системой.
- **Свободно** — это то количество памяти, которое в данный момент не используется или не содержит полезной информации (в отличие от кэшированных файлов, которые содержат полезную информацию).

Под заголовком Память ядра (МБ):



- **Выгружаемая** — это объём виртуальной памяти, используемый основной частью Windows, называемой ядром.
- **Невыгружаемая** — это объём оперативной памяти, используемый ядром.

Под заголовком Система:

- **Дескрипторы.** Число уникальных идентификаторов объектов, используемых процессами. Это значение представляет интерес главным образом для ИТ-специалистов и программистов.
- **Потоки.** Число объектов или процессов, выполняющихся внутри более крупных процессов или программ. Это значение представляет интерес главным образом для ИТ-специалистов и программистов.
- **Процессы.** Число отдельных процессов, исполняемых на компьютере (эти сведения можно видеть и на вкладке «Процессы»).
- **Время работы.** Время, прошедшее после перезагрузки компьютера.
- **Выделено (МБ).** Описание использования виртуальной памяти (также называемой файлом подкачки). Страничный файл — это место на жестком диске, используемое Windows в дополнение к ОЗУ. Первое число представляет собой объём используемой в данное время оперативной и виртуальной памяти, а второе — объём доступной оперативной и виртуальной памяти.

Для того чтобы узнать об использовании памяти подробнее, переключитесь на вкладку Быстродействие и взгляните на таблицы в нижней части диалогового окна. Учтите: числа и подписи к ним могут означать абсолютно не то, что видно изначально.

Ниже представлены некоторые замечания по поводу значений, которые помогут выполнять мониторинг параметров памяти.

- если значение *Физическая память (КБ): Доступно* приближается к нулю, это означает, что данной системе не хватает ресурсов памяти. Причина этого может заключаться в следующем: в системе одновременно работает довольно большое количество программ или одна большая программа использует практически все ресурсы памяти.
- если значение *Физическая память (КБ): Системный кэш* более чем в половину меньше значения *Физическая память (КБ): Всего*, это означает, что данная система функционирует не настолько эффективно, насколько могла бы,



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

потому что Windows XP не удаётся сохранять достаточное количество недавно использовавшихся данных в памяти. Поскольку Windows XP не использует



*системный кэш*, когда ей требуется физическая память, закройте программы, которые не нужны.

- если значение *Выделение памяти (КБ): Всего* превышает значение *Физическая память (КБ): Всего* (и остаётся таковым), это означает, что Windows XP постоянно выгружает данные в страничный файл (page file, или paging file) и загружает их из него, а это существенно снижает степень производительности.
- если значение *Выделение памяти (КБ): Пик* выше значения *Физическая память (КБ): Всего*, это означает, что в какой-то момент во время текущего сеанса Windows XP пришлось прибегнуть к помощи страничного файла. Если на текущий момент времени значение *Выделение памяти (КБ): Всего* меньше значения *Физическая память (КБ): Всего*, критически высокое значение, скорее всего, было лишь временным событием, однако, чтобы быть абсолютно уверенным в этом, за данным значением лучше понаблюдать.

Чуть подробнее о «файле подкачки»:

Как уже говорилось ранее, Windows работает не одними только чипами ОЗУ. Помимо физической памяти для хранения программ и рабочих данных, Windows использует скрытый файл на жестком диске. При необходимости освободить ОЗУ данные перекачиваются в этот файл. Файл подкачки (page file) также называют свопом (swap file, paging file), причем в Windows XP в разных местах используются разные термины. Для единообразия мы будем повсеместно пользоваться только термином файл подкачки.

В процессе установки Windows XP файл подкачки автоматически создается в корневой папке на том же диске, где расположены системные файлы Windows. Размер файла подкачки определяется, исходя из объема физической памяти в вашей системе. По умолчанию минимальный размер файла подкачки в 1,5 раза больше, чем объем физической памяти, а максимальный размер – в 3 раза больше. Файл подкачки можно увидеть в окне Проводника, если включить режим отображения скрытых и системных файлов. Файл pagefile.sys можно найти в корневой папке системного диска.

Открыв окно Панель управления Система, вы сможете наблюдать параметры файла подкачки и изменить их по своему желанию: изменить размер файла подкачки, перенести его на другой диск или разделить на несколько физических дисков для увеличения производительности.

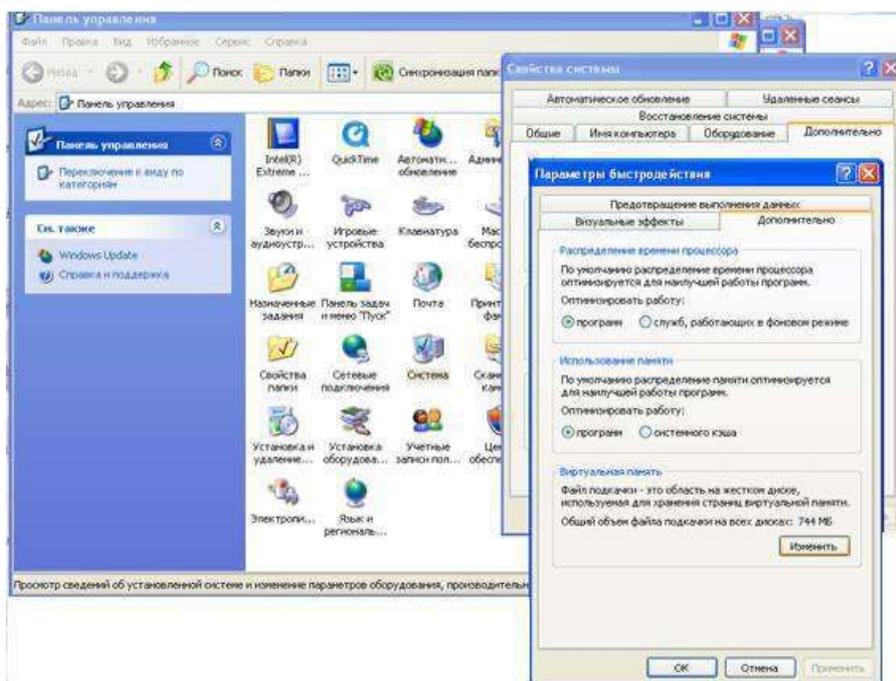
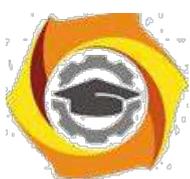
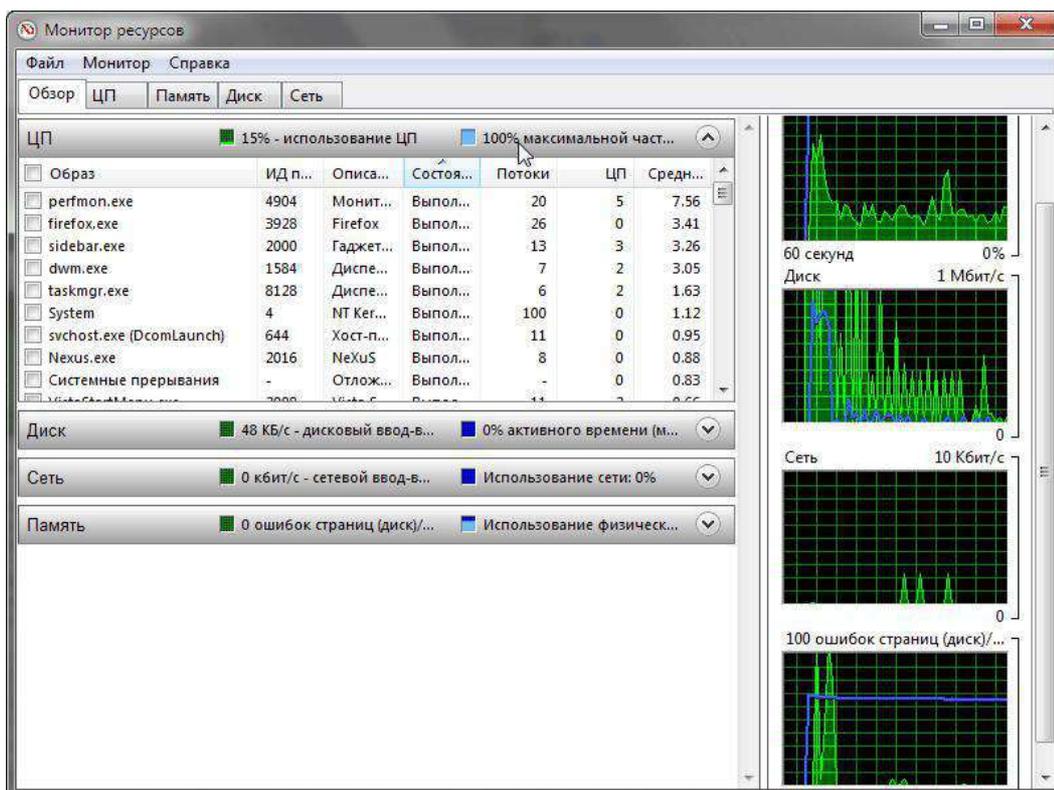


Рисунок 23 – изменение Файла подкачки

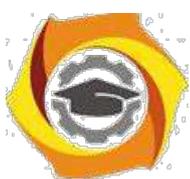
Кнопка Монитор ресурсов предназначена для просмотра дополнительных сведений об используемой памяти и ресурсах ЦПУ. Монитор ресурсов предоставляет графические сведения подобно тем, которые отображаются в диспетчере задач, но в более подробном виде. Здесь также приводятся дополнительные сведения о ресурсах, такие как использование дисков и сети.





**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*Рисунок 24 – Кнопка Монитор*



## Вкладка «Сеть»

При наличии сетевого оборудования на этой вкладке вы увидите список имеющихся сетевых подключений и графики их активности. С их помощью можно определить интенсивность использования сети и её пропускную способность.

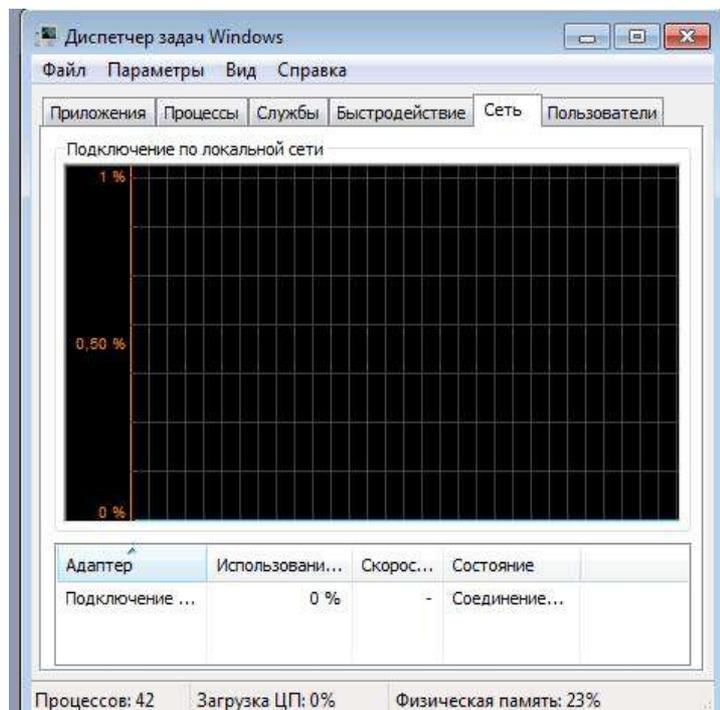
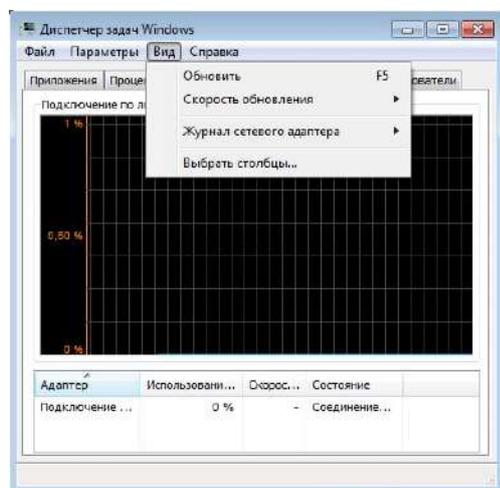
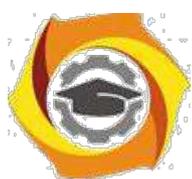


Рисунок 25 – Вкладка «Сеть»

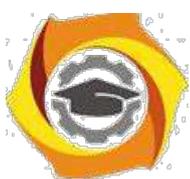
В нижней части вкладки присутствует таблица с текущими параметрами сетевых подключений. Для детального анализа работы сетевого адаптера пользователь может использовать более двух десятков дополнительных сетевых параметров, которые можно отобразить в таблице, выполнив команду **Вид** → **Выбрать столбцы**.





**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*Рисунок 26 – Выбрать столбцы*



15. с помощью команды **Вид** → **Журнал сетевого адаптера** можно включать или выключать отображение дополнительных графиков.

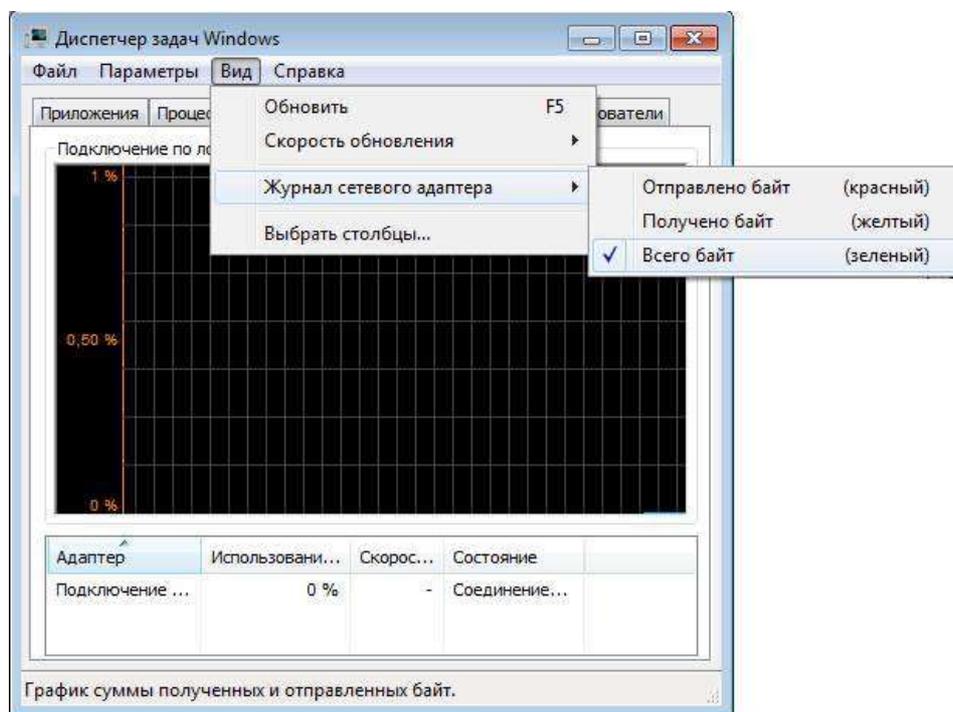


Рисунок 27 – Журнал сетевого адаптера

Существуют способы наложения запрета на автозагрузку программ через записи в реестре, указанные выше. Используются параметры типа DWORD. Все параметры должны храниться в разделе  
KLMSoftware\Microsoft\Windows\CurrentVersion\Policies\Explorer

Для запрета запуска программ, прописанных в подразделе Run раздела LOCAL MACHINE используется параметр DisableLocalMachineRun со значением 1. В этом случае система игнорирует содержимое списка Run, находящегося в LOCAL MACHINE. Аналогично действует запрет списка Run Once для LOCAL MACHINE. За состояние этой политики отвечает параметр DisableLocalMachineRunOnce. Система игнорирует содержимое RunOnce в LOCAL MACHINE.

### Вкладка «Пользователи»

Данная вкладка позволяет просмотреть список активных пользователей и при необходимости выполнить для выбранного пользователя операции отключения или выхода из системы с помощью кнопок в нижней части окна.

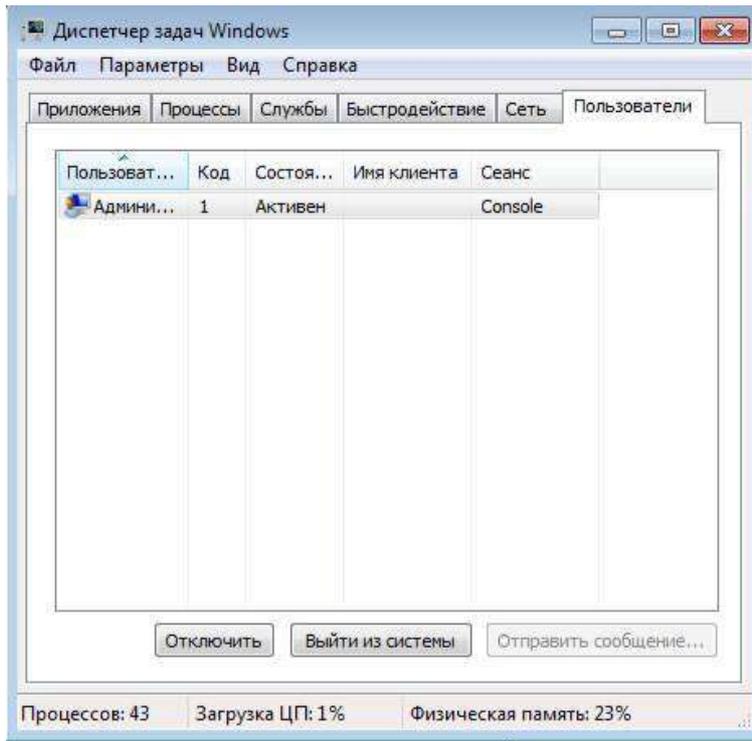


Рисунок 28 – Вкладка «Пользователь»

Закрывать окно Диспетчера задач Windows.



**Контрольные вопросы:**

- Способы запуска Диспетчера задач?
- Сколько экземпляров svchost.exe может быть запущено на компьютере? Почему?
- Что такое spoolsv.exe?
- Чем отличается файл подкачки от виртуальной памяти? Где можно увидеть?
- Дискиптор это-...
- Что такое службы? Для чего предназначены?
- Что необходимо сделать, чтобы запустить настройку системы?
- Что обеспечивает обычный запуск операционной системы?
- В каком случае рекомендуется использовать диагностический запуск?
- Что такое system.ini? Для чего он нужен?
- Что обеспечивает раздел boot?
- Какие функции возложены на файл boot.ini?
- Что такое Application Layer Gateway Service?
- Что такое timeout, rdisk(0), disk(0), default?
- Что делает Computer Browser?
- Расскажите про Cryptographic Services?



## и Практическое занятие № 2 Файловая система NTFS.

### Цель работы

Научиться устанавливать разрешения NTFS для файлов и для папок для отдельных пользователей и групп в операционной системе Windows 7, а также устранять проблемы доступа к ресурсам.

**Оборудование:** персональный компьютер (монитор, системный блок, клавиатура, мышь), ОС Windows, программа для виртуализации Oracle VM VirtualBox, образ виртуальной машины (Win7\_01).

### 3.1. Теоретические сведения

Разрешения NTFS позволяют явно указать, какие пользователи и группы имеют доступ к файлам и папкам и какие операции с содержимым этих файлов или папок им разрешено выполнять. Разрешения NTFS применимы только к томам, отформатированным с использованием файловой системы NTFS. Они не предусмотрены для томов, использующих файловые системы FAT или FAT32. Система безопасности NTFS эффективна независимо от того, обращается ли пользователь к файлу или папке, размещенными на локальном компьютере или в сети.

Разрешения, устанавливаемые для папок, отличаются от разрешений, устанавливаемых для файлов. Администраторы, владельцы файлов или папок и пользователи с разрешением «Полный доступ» имеют право назначать разрешения NTFS пользователям и группам для управления доступом к этим файлам и папкам.

#### Список управления доступом

о NTFS хранится *список управления доступом* (access control list — **ACL**) для каждого файла и папки на томе NTFS. В этом списке перечислены пользователи и группы, для которых установлены разрешения для файла или папки, а также сами назначенные разрешения. Чтобы пользователь получил доступ к ресурсу, в ACL должна быть запись, называемая *элемент списка управления доступом* (access control entry — **ACE**) для этого пользователя или группы, к которой он принадлежит. Эта запись назначит запрашиваемый

тип доступа (например, **Чтение**) пользователю. Если в ACL нет соответствующей ACE, то пользователь не получит доступ к ресурсу.

#### Множественные разрешения NTFS



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Вы можете установить несколько разрешений пользователю и всем группам, членом которых он является. Для этого вы должны иметь представление о правилах и



приоритетах, по которым в NTFS назначаются и объединяются множественные разрешения и о наследовании разрешений NTFS.

**Эффективные разрешения.** Эффективные разрешения пользователя для ресурса — это совокупность разрешений NTFS, которые вы назначаете отдельному пользователю и всем группам, к которым он принадлежит. Если у пользователя есть разрешение «Чтение» для папки, и он входит в группу, у которой есть разрешение «Запись» для той же папки, значит, у этого пользователя есть оба разрешения.

### **Установка разрешений NTFS и особых разрешений**

Вы должны руководствоваться определенными принципами при установке разрешений NTFS. Устанавливайте разрешения согласно потребностям групп и пользователей, что включает в себя разрешение или предотвращение наследования разрешений родительской папки подпапками и файлами, содержащимися в родительской папке.

Если вы уделите немного времени на планирование ваших разрешений NTFS и будете соблюдать при планировании несколько принципов, то обнаружите, что разрешениями легко управлять.

- В Для упрощения процесса администрирования сгруппируйте файлы по папкам следующих типов: папки с приложениями, папки с данными, личные папки. Централизуйте общедоступные и личные папки на отдельном томе, не содержащем файлов операционной системы и других приложений. Действуя таким образом, вы получите следующие преимущества:
  - В — сможете устанавливать разрешения только папкам, а не отдельным файлам;
  - В — упростите процесс резервного копирования, так как вам не придется делать резервные копии файлов приложений, а все общедоступные и личные папки находятся в одном месте.
- В Устанавливайте для пользователей только необходимый уровень доступа. Если необходимо чтение файла, установите пользователю разрешение Чтение для этого файла. Это уменьшит вероятность случайного изменения файла или удаления важных документов и файлов приложений пользователем.
- В Создавайте группы согласно необходимому членам группы типу доступа, затем установите соответствующие разрешения для группы. Назначайте разрешения отдельным пользователям только в тех случаях, когда это необходимо.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**В** При установке разрешений для работы с данными или файлами приложений установите разрешение Чтение и выполнение для групп **Пользователи** и



**Администраторы.** Это предотвратит случайное удаление файлов приложений или их повреждение вирусами или пользователями.

- В При установке разрешений для папок с общими данными назначьте разрешения Чтение и выполнение и Запись группе Пользователи и разрешение Полный доступ для группы Создатель-владелец. По умолчанию пользователь, создавший документ, также является его владельцем. Владелец файла может дать другому пользователю разрешение на владение файлом. Пользователь, который принимает такие права, в этом случае становится владельцем файла. Если вы установите разрешение Чтение и выполнение и Запись группе Пользователи и разрешение Полный доступ группе Создатель-владелец, то пользователи получают возможность читать и изменять документы, созданные другими пользователями, а также читать, изменять и удалять файлы и папки, создаваемые ими.
- В Запрещайте разрешения, только если необходимо запретить отдельный тип доступа определенному пользователю или группе.
- В Поощряйте пользователей в установке разрешений для файлов и папок, которые они создают, и научите их это делать самостоятельно.

Администраторы, пользователи с разрешением Полный доступ и владельцы файлов и папок могут устанавливать разрешения для отдельных пользователей и групп.

Дополнительно Позволяет получить доступ к дополнительным возможностям поиска, включая возможность поиска удаленных учетных записей пользователей, учетных записей с не устаревшими паролями и учетных записей, по которым не подключались определенное количество дней.

### Назначение или запрещение особых разрешений

Щелкните кнопку **Дополнительно**, чтобы открыть диалоговое окно **Дополнительные параметры безопасности**, где перечислены группы и пользователи и установленные для них разрешения для этого объекта. В поле **Элементы разрешений** также указано, от какого объекта разрешения унаследованы

с к каким объектам применимы. Вы можете воспользоваться диалоговым окном **Дополнительные параметры безопасности** для изменения разрешений, установленных для пользователя или группы. Для изменения разрешений, установленных для пользователя или группы, выделите пользователя и щелкните кнопку **Изменить**. Откроется диалоговое окно **Элемент разрешения для**. Затем выделите или отмените определенные разрешения, которые вы хотите изменить.



### 3.2. Задание для самостоятельной работы

**Задание 1.** Открыть Oracle VM VirtualBox

**Задание 2.** Загрузить виртуальную машину Windows 7 (Win7\_01) и создать новую учетную запись uir.

**Задание 3.** Загрузить виртуальную машину Windows 7 с учетной записью uir.

**Задание 4.** Определение разрешений NTFS по умолчанию для только что созданной папки.

Запустить **Проводник**, создать папки **C:\Folder1** и **C:\Folder1\Folder2**. Просмотреть разрешения, установленные для созданных папок, щелкнув по вкладке **Безопасность** диалогового окна свойств папки. Обратит внимание на наследование разрешений папкой **Folder2** от родительской папки **Folder 1**.

Если на экране не видна вкладка **Безопасность**, вам следует уточнить два вопроса:

В Раздел вашего диска отформатирован как NTFS или как FAT? Только на разделах NTFS используются разрешения NTFS, и, таким образом, только на разделах NTFS видна вкладка

**Безопасность.**

10. Используете вы простой общий доступ к файлам или нет? Щелкните кнопку **Отмена**, чтобы закрыть диалоговое окно свойств папки. В пункте меню **Сервис** выберите пункт **Свойства папки**. В диалоговом окне **Свойства папки** перейдите на вкладку **Вид**. В списке **Дополнительные параметры** снимите флажок **Использовать простой общий доступ к файлам (рекомендуется)** и щелкните **ОК**.

Определить для какой группы установлены особые разрешения. Щелкнуть кнопку **Дополнительно**, выделить эту группу и просмотреть установленные разрешения.

Закрывать диалоговое окно свойств папки. Закрывать окно **Проводник** и завершить сеанс.

**Задание 5.** Создать новую учетную запись **uir-1**.

**Задание 6.** Войти в систему с учетной записью **uir-1**. Запустить **Проводник**, войти в папку **C:\Folder1**. Создать два текстовых документа, присвоив им имена **file 1** и **file 2**. Попытаться выполнить следующие операции с **файлом file1**: открыть файл; изменить файл; удалить файл. Какие действия вы смогли успешно совершить и почему?



Завершить сеанс работы и войти в систему, используя учетную запись **uir-2**. Запустить **Проводник**, войти в папку **C:\Folder1**. Попытаться выполнить следующие операции с файлом **fiie2**: открыть файл; изменить файл; удалить файл. Какие действия вы смогли успешно совершить и почему? В настоящее время ваша регистрационная запись — **uir-2**. Можете ли вы изменить разрешения, установленные для пользователя, пока вы подключены как **uir-2**? Почему? Завершить сеанс.

**Задание 7.** Установить разрешения NTFS для папки **C:\Folder1**. При этом необходимо соблюдать следующие правила:

все пользователи должны иметь возможность читать документы и файлы  
Спапке **Folder1**;

все пользователи должны иметь возможность создавать документы  
в папке **Folder1**;

все пользователи должны иметь возможность изменять содержание,  
свойства и разрешения для создаваемых ими документов в папке **Folder1**;

пользователь **uir-2** несет ответственность за содержимое папки **Folder1**  
и должен иметь возможность изменять и удалять все файлы в папке **Folder1**.

Основываясь на полученной информации, определить, как следует  
изменить разрешения для соответствия этим четырем критериям?

Войти в систему, используя учетную запись **uir**. Открыть Проводник. Открыть папку **Folder1**. Щелкнуть правой кнопкой мыши значок папки **Folder1**, затем выбрать пункт меню Свойства. Перейти на вкладку **Безопасность диалогового окна** свойств папки. На вкладке **Безопасность** щелкнуть кнопку **Добавить**. Откроется диалоговое окно **Выбор: Пользователи или Группы**.

В текстовом поле **Введите имена выбираемых объектов** ввести **uir-2**, затем щелкнуть кнопку **Проверить имена**. В текстовом поле **Введите имена выбираемых объектов** должна появиться надпись **<имя компьютера>\тг-2**. Это свидетельствует, что Windows XP Professional обнаружила пользователя **uir-2** на компьютере **<имя компьютера>** и что это действительная учетная запись пользователя. Щелкнуть **ОК**, чтобы закрыть диалоговое окно **Выбор: Пользователи или Группы**. Теперь пользователь **uir-2** включен в список Группы или пользователи диалогового окна свойств папки **Folder1**. Какие разрешения установлены для пользователя **uir-2**?

Щелкнуть кнопку **Дополнительно**. Откроется диалоговое окно **Дополнительные параметры безопасности** для **Folder1**, и вы увидите, что пользователь **uir-2** включен



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

В список **Элементы разрешений**. Убедиться, что строка **uir-2** выделена, и щелкнуть кнопку **Изменить**. Откроется диалоговое окно **Элемент разрешения** для **Folder1**, и вы увидите в текстовом поле **Имя** учетную запись пользователя **uir-2**.



и колонке **Разрешить** щелкнуть **Полный доступ**. Теперь в колонке **Разрешить** установлены все флажки. Щелкнуть **ОК**, чтобы закрыть диалоговое окно **Элемент разрешения для Folder!** и щелкнуть **ОК**, чтобы закрыть диалоговое окно **Дополнительные параметры безопасности для Folder 1**. Щелкнуть **ОК**, чтобы закрыть диалоговое окно свойств папки **Folder 1**.

Закрывать **Проводник** и завершить сеанс Windows XP Professional.

Войти в систему, используя учетную запись **uir-2**. Запустить **Проводник**, войти в папку **C:\Folder1**. Попытаться выполнить следующие операции с файлом **file2**: изменить файл; удалить файл. Какие действия вы смогли успешно совершить и почему? Завершить сеанс Windows 7.

**Задание 8.** Проверить, как разрешения NTFS наследуются в иерархии папок.

Войти в систему, используя учетную запись **uir-1**. Запустить **Проводник**, войти в папку **C:\Folder1\Folder2**. Создать текстовый файл с именем **Uir3** в папке. Завершить сеанс Windows

Войти в систему, используя учетную запись **uir-2**. Запустить **Проводник**, войти в папку **C:\Folder1\Folder2**. Попытаться выполнить следующие операции с файлом **Uir3**: открыть файл; изменить файл; удалить файл. Какие действия вы смогли совершить и почему? Завершить сеанс Windows 7.

**Задание 9.** Изучить результаты смены владельца файла.

Войти в систему, используя учетную запись **uir**. В папке **C:\Folder1** создать текстовый файл **file4**.

Щелкнуть правой кнопкой мыши значок документа **file4**, затем выбрать пункт меню **Свойства**. Откроется диалоговое окно **Свойства: file4** с активной вкладкой **Общие**. Перейти на вкладку **Безопасность** для просмотра разрешений, установленных для файла **file4**. Щелкнуть кнопку **Дополнительно**. Откроется диалоговое окно **Дополнительные параметры безопасности для file4** с активной вкладкой **Разрешения**. Перейти на вкладку **Владелец**. Кто является текущим владельцем файла **file4**?

Установка разрешения, позволяющего пользователю сменить владельца.

9 диалоговом окне **Дополнительные параметры безопасности для file4** перейти на вкладку **Разрешения**. Щелкнуть кнопку **Добавить**. Откроется диалоговое окно **Выбор: Пользователи или Группы**. Убедиться, что в текстовом поле **Размещение**, которое расположено вверху диалогового окна, выбрано имя



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

вашего компьютера. В текстовом поле Введите **имена выбираемых объектов** ввести **uir-3**, затем щелкните кнопку **Проверить имена**. Щелкнуть **ОК**.



Станет активным диалоговое окно **Элемент разрешения для file4**. Обратить внимание на то, что все элементы разрешений для пользователя **uir-3** не отмечены. В колонке **Разрешения** установить флажок **Разрешить** для разрешения **Сменить владельца**. Щелкнуть **ОК**. Щелкнуть **ОК** для того, чтобы вернуться к диалоговому окну свойств файла **file4**. Щелкнуть **ОК** для сохранения изменений и закрыть диалоговое окно свойств файла **file4**. Закрыть **Проводник** и выйти из системы. Смена владельца файла.

Войти в систему, используя учетную запись **uir-3**. Запустить **Проводник**, войти в папку **C:\Folder1**. Щелкнуть правой кнопкой мыши значок файла **file4** и выбрать пункт меню **Свойства**. Перейти на вкладку **Безопасность** для просмотра разрешений для файла. Щелкнуть **Дополнительно** и перейти на вкладку **Владелец**. В колонке **Изменить владельца на** выбрать **uir-3**, затем щелкнуть кнопку **Применить**. Кто теперь является владельцем файла **Uir4**?

Щелкнуть **ОК**, чтобы закрыть диалоговое окно **Дополнительные параметры безопасности для file4**.

Проверка разрешений для файла в качестве владельца.

Щелкнуть кнопку **Дополнительно** и снять флажок **Наследовать от родительского объекта применимые к дочерним объектам разрешения, добавляя их к явно заданным в этом окне**. Установить разрешение **Полный доступ** к текстовому документу **file4** и нажать кнопку **Применить**. Щелкнуть **ОК**, чтобы закрыть диалоговое окно **Дополнительные параметры безопасности для file4**. Щелкнуть **ОК**, чтобы закрыть диалоговое окно свойств файла **file4**.

**Задание 10.** Изучить изменение разрешений и прав владельца при копировании и перемещении папок. Создание папки при подключении с учетной записью пользователя.

Пока вы зарегистрированы в системе под учетной записью **uir-3** создать папку с именем **Temp1** в корневой папке диска **C:\**. Какие разрешения установлены для этой папки? Кто является владельцем папки?

Создание папок при подключении с учетной записью члена группы **Администраторы**.

Подключитесь с учетной записью **uir** и создайте папки **Temp2** и **Temp 3** в корневой папке диска **C:\**.

Каковы разрешения для папок, которые вы только что создали? Кто является владельцем папок **Temp 2** и **Temp 3**?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Установить разрешения для папок **Temp 2** и **Temp 3**.



Снять флажок **Наследовать от родительского объекта применимые к дочерним объектам разрешения**, добавляя их к явно заданным в этом окне. В открывшемся диалоговом окне щелкните **Удалить** для удаления всех разрешений, кроме указанных ниже.

Папка **Temp 2: Администраторы** — **Полный доступ**; **Пользователи** — **Чтение и выполнение**.

Папка **Temp 3: Администраторы** — **Полный доступ**; **Операторы архива** — **Чтение и выполнение**; **Пользователи** — **Полный доступ**.

Копирование папки в другую папку на одном и том же томе NTFS.

Пока вы находитесь в системе под учетной записью **uir**, скопировать папку **C:\Temp2** папку **C:\Temp1**. Для этого выделить значок папки **C:\Temp2** и, удерживая нажатой клавишу **CTRL**, перетащить мышью **C:\Temp2** в **C:\Temp1**.

Выделив **C:\Temp1\Temp2**, просмотреть разрешения и права владельца, затем сравнить разрешения и права владельца с папкой **C:\Temp2**.

Перемещение папки на одном и том же томе.

Войти в систему с учетной записью **uir-3**. В **Проводнике** выделить значок папки **C:\Temp3**, затем переместить ее в папку **C:\Temp1**. Что произошло с разрешениями и владельцем для папки **C:\Temp1\Temp3**?

**Задание 11.** Самостоятельно определить? как предотвратить удаление пользователями, имеющими разрешение **Полный доступ** к папке, файла в этой папке, для которого установлен запрет на разрешение **Полный доступ**?

### 3.3. Контрольные вопросы

В Что такое эффективные разрешения пользователя для ресурса?

В Какие объекты по умолчанию наследуют разрешения, установленные для родительской папки?

В Чем отличается разрешение «Удаление» от разрешения «Удаление подпапок и файлов»?

В Какое разрешение NTFS для файлов следует установить для файла, если вы позволяете пользователям удалять файл, но не позволяете становиться владельцами файла?

В Если вы хотите, чтобы пользователь или группа не имела доступ к определенной папке или файлу, следует ли запретить разрешения для этой папки или файла?



## 11. Лабораторная работа № 2 Работа с системным реестром

### Цель работы

Получение основных сведений о структуре и функциях системного реестра операционной системы Windows 7

**Оборудование:** персональный компьютер (монитор, системный блок, клавиатура, мышь), ОС Windows, программа для виртуализации Oracle VM VirtualBox, образ виртуальной машины (Win7\_01).

### 4.1. Задание:

- В Изучить теоретическую часть;
- В Запустить редактор реестра.
- В Перейти в раздел реестра HKEY\_CURRENT\_USER;
- В Найти ключ, отвечающий за настройки Рабочего стола;
- В Ознакомиться со списком вложенных ключей;
- В Для произвольно выбранных из списка 5 ключей исследовать, аналогом каких

\quote Перейти в раздел реестра HKEY\_CLASSES\_ROOT;

\quote Выбрать из списка 5 ключей и описать, для файлов с какими расширениями они используются, и какие параметры для них установлены;

\quote Результаты внести в отчет.

### 4.2. Теоретические сведения:

На смену ini-файлам, имеющим ряд концептуальных ограничений, еще в Windows 3.1 было введено понятие реестра – регистрационной базы данных, хранящей различные настройки ОС и приложений. Изначально реестр был предназначен только для хранения сведений об объектах OLE (Object Linking and Embedding — связь и внедрение объектов) и сопоставлений приложений расширениям имен файлов, однако позже его структура и границы использования



расширились. Реестры разных версий Windows имеют различия; это нужно помнить при импорте reg-файлов. В Windows 7 в архитектуру реестра были введены важные новшества, улучшающие функциональность данного компонента ОС. Реестр хранится в бинарном (двоичном) виде, поэтому для ручной работы с ним необходима специальная программа — редактор реестра. В 7 это Regedit.exe, в других версиях NT ими являются Regedit.exe и Regedt32.exe, имеющий дополнительные возможности работы с реестром (Regedt32.exe есть и в 7, но на самом деле он всего лишь вызывает Regedit.exe). Есть и другие программы, в том числе и консольные (Reg.exe). Ручным модифицированием параметров реестра мы займемся чуть позже, а сейчас рассмотрим основные группы сведений, хранящихся в этой базе данных.

**Программы установки.** Любая грамотно написанная программа под Windows должна иметь свой инсталлятор-установщик. Это может быть встроенный в ОС Microsoft Installer либо любой другой. В любом случае инсталлятор использует реестр для хранения своих настроек, позволяя правильно устанавливать и удалять приложения, не трогая совместно используемые файлы.

**Распознаватель.** При каждом запуске компьютера программа NTDETECT.COM и ядро Windows распознает оборудование и сохраняет эту информацию в реестре.

**Ядро ОС.** Хранит много сведений в реестре о своей конфигурации, в том числе и данные о порядке загрузки драйверов устройств.

**Диспетчер PnP (Plug and Play).** Абсолютно необходимая вещь для большинства пользователей, которая избавляет их от мук по установке нового оборудования (не всегда, правда:)). Неудивительно, что он хранит свою информацию в реестре.

**Драйверы устройств.** Хранят здесь свои параметры.

**Административные средства.** Например, такие, как Панель управления, MMC (Micro-soft Management Console) и др.

**Пользовательские профили.** Это целая группа параметров, уникальная для каждого пользователя: настройки графической оболочки, сетевых соединений, программ и многое другое.

**Аппаратные профили.** Позволяют создавать несколько конфигураций с различным оборудованием.



**Общие настройки программ.** Почему общие? Потому, что у каждого пользователя есть профиль, где хранятся его настройки для соответствующей программы.

Таким образом, выше приведены данные о предназначении реестра. Теперь обратим внимание на логическую структуру реестра. Для лучшего понимания материала рекомендуется запустить Regedit.exe.

### Структура реестра

Реестр Windows имеет древовидную структуру, схожую со структурой файловой системы. Папкам здесь соответствуют ключи (keys) или разделы (ветви), а файлам — параметры (values). Разделы могут содержать как вложенные разделы (sub keys), так и параметры. На верхнем уровне этой иерархии находятся корневые разделы (root keys).

#### HKKEY\_LOCAL\_MACHINE

Содержит глобальную информацию о компьютерной системе, включая такие данные об аппаратных средствах и операционной системе, в том числе: тип шины, системная память, драйверы устройств и управляющие данные, используемые при запуске системы. Информация, содержащаяся в этом разделе, действует применительно ко всем пользователям, регистрирующимся в системе Windows. На верхнем уровне иерархии реестра для этого раздела имеются три псевдонима:

HKKEY\_CLASSES\_ROOT, HKKEY\_CURRENT\_CONFIG и  
HKKEY\_DYN\_DATA HKKEY\_CLASSES\_ROOT

Содержит ассоциации между приложениями и типами файлов (по расширениям имени файла). Кроме того, этот раздел содержит информацию OLE (Object Linking and Embedding), ассоциированную с объектами COM, а также данные по ассоциациям файлов и классов (эквивалент реестра ранних версий Windows, служивших настройкой над MS-DOS). Параметры этого раздела совпадают с параметрами, расположенными в разделе HKKEY\_LOCAL\_MACHINE\Software\Classes. Подробную информацию о разделе HKKEY\_CLASSES\_ROOT можно найти в руководстве OLE Programmer's Reference, входящем в состав продукта Windows NT 4.0 Software Development Kit (SDK)



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

HKY\_CURRENT\_CONFIG



Содержит конфигурационные данные для текущего аппаратного профиля. Аппаратные профили представляют собой наборы изменений, внесенных в стандартную конфигурацию сервисов и устройств, установленную данными разделов Software и System корневого раздела HKEY\_LOCAL\_MACHINE. В разделе HKEY\_CURRENT\_CONFIG отражаются только изменения. Кроме того, параметры этого раздела появляются также в разделе HKEY\_LOCAL\_MACHINE\System\CurentControlSet\HardwareProfites\CuiTent

#### HKEY\_CURRENT\_USER

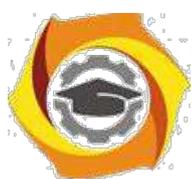
Содержит, профиль пользователя, на данный момент . зарегистрировавшегося в системе, включая переменные окружения, настройку рабочего стола, параметры настройки сети, принтеров и приложений. Этот раздел представляет собой ссылку на раздел HKEY\_USERS\username, где username — имя пользователя, зарегистрировавшегося в системе на текущий момент HKEY\_USERS Содержит все активно загруженные пользовательские профили, включая HKEY\_CURRENT\_USER, также профиль по умолчанию. Пользователи, получающие удаленный доступ к серверу, не имеют профилей, содержащихся в этом разделе; их профили загружаются в реестры на их собственных компьютерах. Windows NT/2000 требует наличия учетных записей для каждого пользователя, регистрирующегося в системе.

Раздел HKEY\_USERS содержит вложенный раздел \Default, а также другие разделы, определяемые идентификатором безопасности (Security ID) каждого пользователя

#### Типы данных

Все параметры реестра имеют фиксированный тип. В таблице 2 приводится полный список используемых типов. Не все из них используются в разных версиях NT

REG\_QWORD явно предназначен для 64-битной версии 7. Следует учесть, что ряд типов используется только системой в некоторых разделах, и создать свой параметр такого типа с помощью редактора реестра не получится.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

REG\_BINARY



Двоичные данные. Большинство сведений об аппаратных компонентах хранится в виде двоичных данных и выводится в редакторе реестра в шестнадцатеричном формате

#### REG\_DWORD

Данные, представленные целым числом (4 байта). Многие параметры служб и драйверов устройств имеют этот тип и отображаются в двоичном, шестнадцатеричном или десятичном форматах

#### REG\_EXPAND\_SZ

Строка Unicode переменной длины. Этот тип данных включает переменные, обрабатываемые программой или службой

#### REG\_MULTI\_SZ

Многострочный текст Unicode. Этот тип, как правило, имеют списки и другие записи в формате, удобном для чтения. Записи разделяются пробелами, запятыми или другими символами

#### REG\_SZ

Текстовая Unicode строка фиксированной длины

#### REG\_DWORD\_LITTLE\_ENDIAN

32-разрядное число в формате “остроконечников” — младший байт хранится первым в памяти. Эквивалент REG\_DWORD

#### REG\_DWORD\_BIG\_ENDIAN

32-разрядное число в формате “тупоконечников” — старший байт хранится первым в памяти  
REG\_LINK Символическая ссылка Unicode. Только для внутреннего использования (некоторые корневые разделы являются такой ссылкой на другие подразделы)



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

REG\_NONE



Параметр не имеет определенного типа данных

REG\_QWORD

64-разрядное число

REG\_QWORD\_LITTLE\_ENDIAN

64-разрядное число в формате “остроконечников”. Эквивалент REG\_QWORD

REG\_RESOURCE\_LIST

Список аппаратных ресурсов. Используется только в разделе  
HKLM\HARDWARE

REG\_FULL\_RESOURCE\_DESCRIPTOR

Дескриптор (описатель) аппаратного ресурса. Применяется только в  
HKLM\HARDWARE.

REG\_RESOURCE\_REQUIREMENTS\_LIST

Список необходимых аппаратных ресурсов. Используется только в  
HKLM\HARDWARE.

### **Хранение реестра**

Элементы реестра хранятся в виде атомарной структуры. Реестр разделяется на составные части, называемые ульями (hives), или кустами. Ульи хранятся на диске в виде файлов. Некоторые ульи, такие, как HKLM\HARDWARE, не сохраняются в файлах, а создаются при каждой загрузке, то есть являются изменяемыми (volatile). При запуске системы реестр собирается из ульев в единую древовидную структуру с корневыми разделами. Перечислим ульи реестра и их местоположение на диске (для NT старше версии 4.0)



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

HKLM\SYSTEM %SystemRoot%\system32\config\system



HKLM\SAM      %SystemRoot%\system32\config\SAM

HKLM\SECURITY      %SystemRoot%\system32\config\SECURITY

HKLM\SOFTWARE      %SystemRoot%\system32\config\software

HKLM\HARDWARE      Изменяемый улей

HKLM\SYSTEM\Clone      Изменяемый улей

HKU\<SID\_пользователя> %USERPROFILE%\ntuser.dat

HKU\<SID\_пользователя>\_Classes %USERPROFILE%\Local

Settings\Application

Data\Microsoft\Windows\UsrClass.dat

HKU\DEFAULT %SystemRoot%\system32\config\default

Кроме этих файлов, есть ряд вспомогательных, со следующими расширениями:

ALT – резервная копия улья HKLM\SYSTEM (отсутствует в 7).

LOG – журнал транзакций, в котором регистрируются все изменения реестра.

SAV – копии ульев в том виде, в котором они были после завершения текстовой фазы установки.

### **Дополнительные сведения**

Реестр является настоящей базой данных, поэтому в нем используется технология восстановления, похожая на оную в NTFS. Уже упомянутые LOG-файлы содержат журнал транзакций, который хранит все изменения. Благодаря этому реализуется атомарность реестра – то есть в данный момент времени в реестре могут быть либо старые значения, либо новые, даже после сбоя. Как видим, в отличие от NTFS, здесь обеспечивается сохранность не только структуры реестра, но и данных.

- тому же, реестр поддерживает такие фишки NTFS, как управление избирательным доступом и аудит событий – система безопасности пронизывает всю NT снизу доверху. Да, эти функции доступны только из Regedt32.exe или Regedit.exe для 7. А



еще весь реестр или его отдельные части можно экспортировать в текстовые reg-файлы (Unicode для Windows 2000 и старше), редактировать их в блокноте, а затем экспортировать обратно. Во многих редакторах реестра можно подключать любые доступные ульи реестра, в том числе и на удаленных машинах (при соответствующих полномочиях). Есть возможность делать резервные копии с помощью программы NTBackup.

### 4.3. Ход работы:

Для запуска системного реестра Windows 7 необходимо нажать кнопку <Пуск>, <Выполнить>, ввести команду <Regedit> и нажать <ОК>. Запуститься программа Редактор реестра .

Для перехода по разделам реестра необходимо выбрать соответствующий раздел

нажать кнопку раскрывающегося списка, находящуюся слева от названия раздела. Ключ, отвечающий за настройки рабочего стола находится по адресу

<HKEY\_CURRENT\_USER\Control Panel\Desktop>

Размеры элементов экрана в Windows (иконки, шрифты, рамки, меню, полосы прокрутки) хранятся в разделе HKEY\_CURRENT\_USER\Control Panel\desktop\WindowMetrics реестра

Ниже приведены некоторые параметры, содержащиеся в этом разделе.

BorderWidth      Ширина рамки окна

CaptionFont      Шрифт заголовка

CaptionHeight    Высота шрифта заголовка

CaptionWidth     Ширина заголовка

IconFont          Шрифт названия иконки

IconSpacing       Горизонтальный интервал между иконками

IconSpacingFactor Фактор, используемый для вычисления положения иконок

IconVerticalSpacing      Вертикальный интервал между значками

MenuFont          Параметры шрифта (гарнитура, имя шрифта, и т.д.), используемого в строках меню



MenuHeight Высота ячейки символа, используемого в строке меню  
MenuWidth Ширина ячейки символа, используемого в строке меню  
MessageFont Шрифт, используемый в сообщениях  
ScrollHeight Высота горизонтальной полосы прокрутки  
ScrollWidth Ширина вертикальной полосы прокрутки  
ShellIconBPP Число цветов (битов на точку), используемых для иконок  
ShellIconSize Размер иконок на Рабочем столе (и в проводнике в режиме

"Крупные значки")

SmCaptionFont Шрифт в маленьких заголовках  
SmCaptionHeight Высота ячейки символа в маленьком заголовке  
SmCaptionWidth Ширина ячейки символа в маленьком заголовке

Каждый ключ, содержащий данные для шрифта, состоит из последовательности байтов, соответствующих имени шрифта и нескольким флагам, определяющим тип шрифта, типы начертания (полужирный, курсив) и т.д. Эти параметры можно изменять на вкладке «Оформление» диалога «Свойства: Экран».

#### **Некоторые параметры настройки элементов экрана:**

`HKEY_CURRENT_USER\Control Panel\Desktop\WindowMetrics\ShellIconSize` – управляет размером отображения значков рабочего стола. Значение 48 указывает, что значки рабочего стола будут отображаться размером 48x48 точек. Аналог <Свойства: Экран> / <Оформление> / <Эффекты> / <Применять крупные значки>.

`HKEY_CURRENT_USER\Control Panel\Desktop\FontSmoothing` – управляет сглаживанием неровностей экранных шрифтов. Аналог <Свойства: Экран> / <Оформление> / <Эффекты> / <Применять следующий метод сглаживания экранных шрифтов>.

`HKEY_CURRENT_USER\Control Panel\Desktop\DragFullWindows` – управляет отображением содержимого окна при его перетаскивании. Аналог <Свойства: Экран> / <Оформление>/<Эффекты>/<Отображать содержимое окна при его перетаскивании>.



HKEY\_CURRENT\_USER\Control Panel\Desktop\Wallpaper – содержит путь к файлу рисунка обоев, , аналог <Свойства: Экран> / <Рабочий стол>.

HKEY\_CURRENT\_USER\Control Panel\Desktop\SCRNSAVE.EXE – содержит путь к файлу с заставкой, аналог <Свойства: Экран> / <Заставка>.

Далее рассмотрим ключ реестра HKEY\_CLASSES\_ROOT

Корневой ключ реестра HKEY\_CLASSES\_ROOT содержит информацию обо всех ассоциациях (связях) расширений имен файлов, с приложениями, поддерживающими эти типы файлов, и о данных, ассоциированных с объектами COM. Эти данные совпадают с информацией, которая содержится в ключе classes, расположенной в иерархии ниже ключа HKEY\_LOCAL\_MACHINE\SOFTWARE.

Некоторые ключи раздела HKEY\_CLASSES\_ROOT:

HKEY\_CLASSES\_ROOT\.ico – определяет параметры файлов с расширением ico (значков, иконок);

HKEY\_CLASSES\_ROOT\.xls\Excel.Sheet.8\ShellNew – определяет параметры открытия файлов с расширением XLS (параметр Filename=excel9.xls);

HKEY\_CLASSES\_ROOT\.zip\ShellNew – определяет параметры открытия файлов

расширением ZIP(параметр Filename= C:\Program Files\WinRAR\zipnew.dat);

HKEY\_CLASSES\_ROOT\Excel.Template\shell\Print\command – определяет команды печати для шаблонов электронных таблиц Excel

HKEY\_CLASSES\_ROOT\jpg – определяет программу с которой ассоциированы файлы с расширением JPG



## В Практическое занятие № 3 Консоль администрирования Windows 7

Цель работы:

Практическое освоение инструмента управления операционной системы Windows 7 – графической консоли администрирования MMC.

Оборудование: персональный компьютер (монитор, системный блок, клавиатура, мышь), ОС Windows, программа для виртуализации Oracle VM VirtualBox, образ виртуальной машины (Win7\_01).

### 5.1. Теоретическая часть

Консоль управления Microsoft Management Console (MMC) группирует средства администрирования, которые используются для администрирования сетей, компьютеров, служб и других системных компонентов.

Консоль MMC непосредственно не выполняет административные функции, однако предоставляет возможности интеграции в нее компонентов или системных приложений, выполняющие эти функции. Основной тип интегрируемых на консоль компонентов называется оснасткой, которые не могут выполняться отдельно без консоли. Среди других добавляемых элементов могут быть элементы управления ActiveX, ссылки на Web-страницы, папки, видов панели задач и собственно задачи для выполнения. Дополнительные теоретические сведения об оснастках и других используемых для интеграции на консоль элементах будут добавлены в дальнейшем, в соответствующих разделах настоящей лабораторной работы.

Базовое окно консоли MMC представляет собой графическую форму с контекстными меню, реализующие дружественный пользовательский интерфейс. Имеется панель инструментов с командами создания, открытия и сохранения консолей и, кроме того, область описания и строка состояния в нижней части окна. Чтобы увидеть базовое окно, а также непосредственно саму консоль MMC, необходимо выполнить следующие действия:

- нажмите Пуск | Выполнить,
- наберите в появившемся окне MMC.exe (или просто mmc),
- нажмите Enter для ввода.

Новая консоль MMC представляет собой отдельное окно, разделенное на две вертикальные области, в левой из которых отображается дерево консоли с его корнем. Дерево консоли оказывает доступные элементы и компоненты консоли. Правая



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

область является областью сведений, которая содержит описания элементов и выполняемых ими функций. Содержание области сведений соответствует



выбранному элементу в дереве консоли и может включать Web-страницы, графики, диаграммы, таблицы и столбцы.

Создавая надежные средства управления компьютерами сети, можно собрать и настроить собственную консоль ММС, выполняющую заданные функции администрирования. После того как добавлены все необходимые элементы и компоненты консоли, панель главного меню, панель инструментов, а также область описания и строка состояния могут быть скрыты для предотвращения в дальнейшем нежелательных изменений. Созданные таким образом управляющие системы сохраняются в файлах с расширением .msc (Management Saved Console, сохраненная консоль управления) и могут быть, в частности, распространены в пределах всей системы посредством задания к ним доступа с помощью ярлыков или элементов меню Пуск.

Чтобы увидеть консоль управления локальным компьютером в качестве примера готовой и отлаженной консоли ММС, необходимо выполнить:

8. нажмите Пуск | Выполнить,
9. наберите в появившемся окне `compmgmt.msc` (или `compmgmt`),
10. нажмите Enter для ввода.

Существует два основных режима доступа консоли администрирования, задающиеся непосредственно при ее создании:

пользовательский, в котором можно администрировать систему, работая  
10. уже существующими консолями,

авторский, в котором можно создавать новые консоли или изменять существующие.

• свою очередь, имеется три уровня режима пользователя, что обуславливает всего четыре варианта предустановленного режима доступа:

- авторский режим;
- режим пользователя — полный доступ;
- режим пользователя — ограниченный доступ, многооконный;
- режим пользователя — ограниченный доступ, однооконный.

Консоль ММС, инициализированная в авторском режиме, предоставляет полный доступ ко всем ее возможностям, включая добавление и удаление оснасток,



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

создание новых окон и панелей задач, а также просмотр любых частей дерева консоли и другие. Однако при выборе одного из трех режимов пользователя авторские



возможности исключаются. В частности, если для консоли установлен параметр «пользовательский режим — полный доступ», то предоставляются все команды управления окном консоли и полный доступ к ее дереву, но запрещается добавление, удаление оснасток и изменение свойств консоли администрирования.

Изменения консоли ММС в авторском и пользовательском режимах сохраняются по-разному. При закрытии консоли в авторском режиме выводится диалоговое окно с предложением сохранить изменения. Однако в пользовательском режиме и снятом флажке «Не сохранять изменения для этой консоли» изменения будут сохранены автоматически при закрытии.

Если консоль открыта при соблюдении одного из следующих условий:

- в базовом окне при загрузке,
- с помощью команды контекстного меню Автор,
- в командной строке с параметром /a,

то предустановленный режим игнорируется, а открытие консоли осуществляется в авторском режиме.

Очевидно, что загрузка консоли ММС в авторском режиме не требуется рядовым пользователям. Системный администратор может настроить профили пользователей так, чтобы запретить им переход в авторский режим, как из командной строки, так и через контекстное меню. Кроме того, запрет перехода в авторский режим может быть организован при использовании возможностей групповой политики, при которой, в частности, осуществляется ограничение доступа к определенным оснасткам. Рассмотрению базовых возможностей оснастки групповой политики будет посвящена вторая часть настоящей лабораторной работы.

Прежде чем создавать новую консоль ММС, необходимо определить действия, для которых предназначена эта консоль, список администрируемых компонентов, оснасток и других элементов, которые потребуются для выполнения поставленных задач. Следует также рассмотреть необходимость создания видов панели задач. После принятия этих решений можно открыть новую консоль и начать добавлять элементы

- дереву консоли. Полное руководство по созданию и настройке консолей ММС находится на Web-узле корпорации Майкрософт (<http://www.microsoft.com>).

лабораторной работе предполагается ознакомление с основными принципами организации и построения консоли администрирования ММС, а также с базовыми возможностями основных инструментов системного администратора — оснасток



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

«Локальные пользователи и группы» и «Редактор объекта групповой политики»  
(«Групповая политика»).



Перед началом выполнения заданий в среде ОС Windows 7 необходимо выполнить следующее:

- запустить виртуальную машину с ОС Windows 7 и активировать справочное меню (Пуск | Справка и поддержка);
- ознакомиться с описанием и возможностями запуска и применения консоли администрирования MMC;
- ознакомиться возможностью получения сведений пункта 2 из альтернативного источника информации, доступного непосредственно в справке консоли администрирования MMC (Справка | Вызов справки);
- ознакомиться с описанием и возможностями оснасток «Локальные пользователи и группы» и «Редактор объекта групповой политики» («Групповая политика»).

## 5.2. Практическая часть

### Задание 1. Изменение параметров и способов настройки консоли администрирования MMC

Порядок выполнения:

I. Создание консоли администрирования MMC в авторском режиме требует выполнения следующих действий:

- и нажмите Пуск | Выполнить,
- и наберите в появившемся окне MMC.exe (или просто mmc),
- и нажмите Enter для ввода.

Возможны следующие альтернативные варианты авторского запуска консоли администрирования:

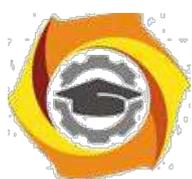
A. запуск из командной строки, используя синтаксис:

Mmc путь\имя\_файла.msc /a,

где параметр:

путь\имя\_файла.msc — запускает консоль MMC с одновременным открытием файла сохраненной консоли с именем имя\_файла.msc (Приложение 2). Если файл консоли не указан, будет открыта новая консоль MMC.

/a — открывает консоль MMC в авторском режиме.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Дополнительными параметрами команды могут быть:



/64 — открывает 64-разрядную версию консоли MMC (MMC64). Этот параметр используется только при работе в ОС Windows 7 64-Bit Edition.

/32 — открывает 32-разрядную версию консоли MMC (MMC32). При работе в ОС Windows 7 64-Bit Edition в окне консоли MMC, запущенной с этим параметром, открываются 32-разрядные оснастки.

В. запуск из файлового менеджера Проводник ОС Windows 7:

наведите манипулятор мышь на файл с расширением .msc, находящийся  
системной папке ОС (%systemroot%\system32\),

кликните правой кнопкой мыши на файле и из контекстного меню  
выберите Автор.

К Настройка параметров консоли администрирования MMC предназначена для ее конфигурирования с целью придания ей уникального вида.

Содержание задания

Для придания уникального вида сохраненной (новой) консоли администрирования MMC в авторском режиме выполните следующие действия:

В В меню Консоль выберите команду Параметры.

В На вкладке Консоль в поле названия введите новый заголовок.

В На вкладке Консоль выполните следующие действия:

нажмите кнопку Сменить значок,

в поле Имя файла введите путь к файлу, содержащему значки  
(например,

%systemroot%\system32\shell32.dll),

4. в поле Текущий значок выберите необходимый значок,

5. кликните ОК для ввода и Применить для подтверждения.

4. На вкладке Консоль из списка Режим консоли выберите пользовательский режим с полным доступом, в котором будет открываться консоль MMC при ее непосредственном запуске,

7. Для установленного в предыдущем пункте режима выполните указанные ниже действия:



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

5. запретите изменение консоли ММС при ее непосредственном запуске, установив флажок «Не сохранять изменения для этой консоли», сделайте активным



диалоговое окно Вид | Настройка вида консоли ММС при запуске, установив флажок «Позволить пользователю настраивать вид консоли»,

6. Если необходимо удалить файлы, содержащие параметры отображения файлов консоли, на вкладке Очистка диска нажмите кнопку Удалить файлы.

7. Сохраните окончательно сконфигурированную консоль администрирования ММС, выбрав самостоятельно ее имя и путь к месту расположения в меню Консоль | Сохранить как... При сохранении обратите внимание на то, что файлы консоли по умолчанию

размещаются в папке «Администрирование», имеющей полный путь %Pathname%\Главное меню\Программы\Админи-стрирование\.

8. Закройте сконфигурированную и сохраненную консоль администрирования ММС, выполнив соответствующие необходимые действия.

5. файловом менеджере Проводник ОС Windows 7 выполните следующие инструкции:

9. наведите манипулятором мышь на сохраненный файл консоли администрирования ММС и, дважды кликнув на нем, запустите консоль,

10. откройте диалоговое окно Вид | Настройка вида и, изменяя положение флажков, обратите внимание на получаемый результат,

11. изменив вид консоли ММС приемлемым образом, кликните ОК для подтверждения полученного результата,

12. в контекстном меню Консоль кликните Выход,

13. снова запустите консоль администрирования ММС, кликнув манипулятором мышь на сохраненном файле консоли,

14. изучите полученный результат,

15. сделайте вывод о проделанной работе и запишите его в отчет.

## **Задание 2. Добавление различных элементов и компонентов к дереву консоли администрирования ММС**

Основным, интегрируемым на консоль компонентом является оснастка. Оснастки существуют двух видов:

7. изолированные,



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

8. расширения.



Изолированная оснастка (или просто оснастка) добавляется к дереву консоли ММС без предварительного добавления других элементов, то есть непосредственно в корень дерева консоли.

Оснастка расширения (или просто расширение) всегда добавляется к другой изолированной оснастке или расширению, которые уже имеются в дереве консоли ММС. Если для определенной оснастки разрешены расширения, то, как правило, они работают с объектами, управляемыми непосредственно этой оснасткой, например, с компьютером, принтером, модемом или другим внешним устройством.

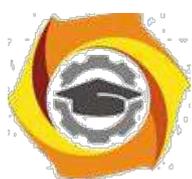
8. В дереве консоли оснастки и расширения располагаются для удобства иерархически или по группам. При добавлении новой оснастки или расширения, они появляются в виде нового элемента в дереве консоли ММС или в виде нового пункта контекстного меню, дополнительной панели инструментов, страницы свойств, а также возможно мастера, организующего определенную последовательность действий, к уже установленной оснастке.

Другими элементами, по необходимости применимыми для интеграции на консоль администрирования ММС, являются виды панели задач и собственно задачи, которые могут включать в себя команды меню для элементов консоли и команды, запускаемые из командной строки. Кроме того, могут быть созданы команды, действующие как часть дерева консоли или открывающие другой компонент.

Прежде всего, перед добавлением указанных элементов к консоли ММС, необходимо определить их число. Если, в частности, требуется добавить несколько видов панели задач, то наряду с этим необходимо определить тип каждой панели (для отображения списка и задач или только задач), а также разделить задачи по интегрированным видам панели. Добавление видов панели задач и собственно задач осуществляется посредством работы мастера создания этих элементов. При этом важно помнить, что консоль ММС должна содержать, по крайней мере, одну оснастку, чтобы возможность интеграции появилась в принципе.

Отдельной возможностью, иногда необходимой при администрировании сетей, является добавление элементов и компонентов дерева консоли администрирования ММС в виде списка ярлыков в меню «Избранное».

Дополнительные сведения о добавлении различных элементов в дерево консоли администрирования ММС можно получить, воспользовавшись справкой ОС Windows 7 (Пуск | Справка и поддержка) в разделе Общее представление о ММС \ Консоль ММС в авторском режиме \ Оснастки \ Создание консолей.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Содержание задания



Первым необходимым компонентом, добавляемым к дереву консоли администрирования ММС при ее организации и построении, является оснастка. Для добавления оснастки в авторском режиме выполните следующие действия:

10. Создайте новую Консоль управления ММС одним из описанных в пункте I текущего учебного задания способов.
11. В меню Консоль выберите команду Добавить или удалить оснастку.
12. В диалоговом окне Добавить/удалить оснастку нажмите кнопку Добавить вкладки Изолированная оснастка. Список Оснастки в диалоговом окне Добавить/удалить оснастку определяет элемент дерева консоли, к которому выполняется добавление элементов. В этом списке можно найти любой элемент дерева консоли. Обратите внимание на то, что по умолчанию это Корень консоли.
13. В диалоговом окне Добавить изолированную оснастку, выберите оснастки Службы из списка доступных в системе, кликнув на ней манипулятором мышь и нажав кнопку Добавить. Для добавления другой оснастки из списка, повторите указанные действия настоящего пункта повторно.
14. Для некоторых оснасток в процессе их инсталляции выводится диалоговое окно Выбор целевого компьютера, определяющее чем, устанавливаемая оснастка, будет управлять в дальнейшем — локальным или сетевым компьютером. Выберите Локальный компьютер, установив переключатель в соответствующее положение.
15. Нажмите Готово, Закрывать и затем кликните ОК для подтверждения ввода.
16. Скройте меню и панель инструментов оснастки Службы, выполнив действия указанные ниже:
  - у В меню Вид выберите команду Настроить,
  - у В группе Оснастка снимите флажок Меню,
  - у В группе Оснастка снимите флажок Панели инструментов.

При устанавливании или снятии флажков, соответствующие им меню и панели инструментов отображаются или скрываются, причем, для всех оснасток консоли, включая текущую. Если переключение флажков не приводит к изменению вида



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

консоли, тогда текущая оснастка не имеет специальных меню или панелей инструментов.



8. Не закрывая консоль администрирования ММС, сохраните ее, выбрав команду Сохранить в меню Консоль.

Для добавления расширений к уже установленной в предыдущем задании оснастке Службы выполните следующее:

В В меню Консоль выберите команду Добавить или удалить оснастку.

4. В диалоговом окне Добавить/удалить оснастку выберите вкладку Расширение. На этой вкладке можно выбрать любой элемент дерева консоли из списка Оснастки, которые могут быть расширены, и просмотреть Доступные расширения, которые могут быть включены или отключены. После подключения расширение автоматически размещается в дереве консоли под оснасткой, к которой оно относится. Если дерево консоли содержит больше одного экземпляра оснастки, к которой подключено расширение, все остальные экземпляры автоматически получают это расширение. Среди Доступных расширений оснастки Службы удалите флажок с расширения Расширенный вид и отметьте к чему привело это действие. Повторите аналогичные действия с другими расширениями данной оснастки и изучите получаемый результат.

5. Не закрывая консоль администрирования ММС, сохраните ее. В окне консоли администрирования выполните следующие инструкции:

4. последовательно перебирая доступные в системе оснастки, найдите те из них, которые обладают дополнительным меню, панелью инструментов или расширениями,

5. изучите полученный результат и сделайте вывод о проделанной работе,



## 6. Лабораторная работа № 3 Знакомство с операционной системой UBUNTU server 16.04

### 6.1. Теоретический материал

**Ubuntu** ([\[uˈbʊntuː\]](#); от [зулу](#) *ubuntu* — *человечность* [\[6\]](#) ; «Убунту») — [операционная система](#), основанная на [Debian GNU/Linux](#). Основным разработчиком - спонсором является компания [Canonical](#). В настоящее время проект активно развивается и поддерживается [свободным сообществом](#)<sup>[7]</sup>.

Ubuntu используется примерно 20 миллионами пользователей<sup>[8]</sup>. Он является 1-м в списке самых популярных дистрибутивов Linux для веб-серверов. По количеству пользователей, посетивших сайт [DistroWatch.com](#) (на апрель 2016 года) занимает 3-е место<sup>[9]</sup>.

Обычно новые версии дистрибутива выходят каждые полгода и поддерживаются обновлениями безопасности в течение 9 месяцев (начиная с версии 13.04, до этого поддержка осуществлялась в течение полутора лет).

Версии LTS,<sup>[10]</sup> выпускаемые раз в 2 года, поддерживаются в течение 5 лет — как [серверные](#), так и десктопные варианты<sup>[11][12][13]</sup>. (До версии 12.04 LTS срок поддержки для десктопных LTS-версий составлял 3 года.) На другие дистрибутивы LTS семейства Ubuntu действует полная поддержка в 3 года,<sup>[14]</sup> а для основы системы ([ядро](#), [Xorg](#) и прочие компоненты) — 5 лет.

Ubuntu поставляется с подборкой программного обеспечения для [серверов](#) и [рабочих станций](#). Она устанавливается на [настольные персональные компьютеры](#) с помощью [LiveCD](#) (версия *Desktop*), [LiveUSB](#) или текстового установщика (версия *Alternate*, предоставлялась до версии Ubuntu 12.04.2)<sup>[15][16]</sup>. В версии [LiveDVD](#) присутствуют несколько бóльшие возможности — начиная от установки не только в графическом,<sup>[17]</sup> но и в [текстовом](#)<sup>[18]</sup> режимах, загрузки в режиме восстановления системы и заканчивая полной [локализацией](#) и бóльшим количеством [пакетов](#) на диске. Есть версии для официально поддерживаемых архитектур, таких как [i386](#), [AMD64](#), [ARM](#). Кроме того, с 2013 года начата разработка [специальной версии Ubuntu](#) для смартфонов на архитектуре ARM и x86<sup>[1]</sup>



Текущим LTS-релизом является Ubuntu 16.04.2 LTS. Начиная с версии 13.04 Raring Ringtail, поддержка для рядовых (не LTS) релизов составит 9 месяцев вместо 18<sup>[112]</sup>.

Логотип версии Ubuntu 16.04 представляет собой схематично нарисованное животное:



Языки программирования

### **Go 1.6**

Ubuntu 16.04 включает в себя релиз Go 1.6, который состоялся в феврале этого года.

### **PHP 7**

Репозиторий Ubuntu 16.04 по умолчанию содержит PHP версии v7.0. Одной из существенных особенностей PHP 7 является значительно возросшая по сравнению со старыми версиями производительность. Также в PHP 7 есть изменения, касающиеся синтаксиса, поэтому если вы разрабатываете приложение на базе PHP 5, вам нужно выполнить миграцию согласно [инструкции на официальном сайте](#).

### **Python 3.5**

Репозиторий Ubuntu 16.04 по умолчанию укомплектован Python версии 3.5.1. Если вам необходима предыдущая версия Python, то ее можно установить следующей командой:

```
sudo apt-get install python
```

Тем, кто использует Vim как основной элемент разработки на Python, необходимо иметь в виду, что теперь этот редактор использует Python 3 в качестве стандартной версии Python, а значит, плагины, написанные на Python 2, могут перестать работать.



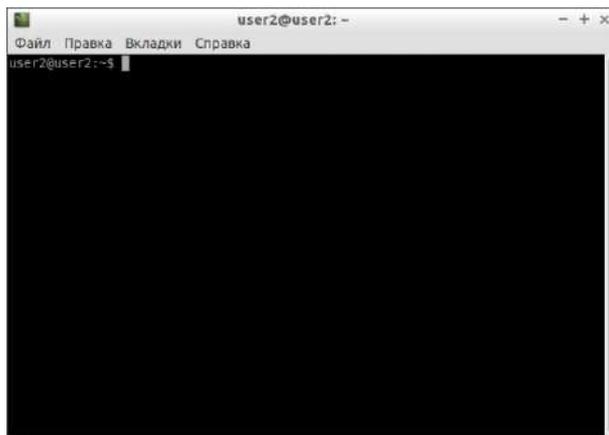
**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Управление Ubuntu server 16.04**



Управление сервером осуществляется как правило с терминала, в консольном режиме. Графическая оболочка здесь не нужна, так как она требует дополнительных ресурсов сервера, а пользователем здесь является опытный системный администратор. Кроме того, лоступ к серверу требуктся достаточно редко.

Для входа в терминал необходимо с консоли ввести логин и пароль пользователя, обычно это данные системного администратора, они вводятся при установке системы. Следует обратить внимание, что прри вводе пароля вводимые символы не отображаются ни в каком виде, просто в терминале ничего не происходит, пока вы не нажмёте клавишу enter. После этого выводятся сведения о системе и приглашение для ввода команд с указанием текущего директория (папки пользователя).



Знакомьтесь, это и есть терминал. Он создан для того, чтобы выполнять текстовые команды, поэтому отложите свою мышку в сторону и пододвиньте поближе клавиатуру<sup>1)</sup>.

### Управление терминалом

Давайте выполним какую-нибудь команду, например:

```
lsb_release -a 2> /dev/null | grep -P "(?<=Codename:)(.*)"
```

На выходе получим кодовое имя нашего дистрибутива:



```
user2@user2: ~
Файл Правка Вкладки Справка
user2@user2:~$ lsb_release -a 2> /dev/null | grep -P "(?<=Codename:)(.*)"
Codename: trusty
user2@user2:~$
```

Набирать такие команды с клавиатуры посимвольно немного неудобно, поэтому давайте сразу разберёмся с основами управления терминалом. Начнём с копирования/вставки. Стандартные сочетания клавиш Ctrl+C и Ctrl+V в терминале не работают, вместо них используется старая добрая пара Ctrl+Insert с Shift+Insert или же сочетания с Shift: Ctrl+Shift+C для копирования и Ctrl+Shift+V для вставки. Что ж, теперь вы умеете целиком копировать команды из руководств.

Кстати, в большинстве руководств и инструкций вы встретите именно терминальные команды. Это связано с тем, что, графических оболочек очень много, и объяснить, как выполнить какое-то действие для каждой из них бывает очень непросто. А терминал - один для всех, и одна и та же команда работает во всех оболочках (естественно, кроме команд по настройке самой оболочки). К тому же, намного проще дать одну команду, чем объяснить где и как 10 раз нажать мышкой.

Однако, часто всё-таки команды приходится набирать вручную, а не вставлять откуда-то. И вот тут на помощь приходит великолепное свойство терминала, называемое автодополнением. Наберите в терминале символы apt-g, а потом нажмите клавишу Tab. Терминал автоматически дополнит за вас команду. Кстати, apt-get - это основная консольная утилита управления пакетами, но об этом позже.

- теперь попробуйте набрать только apt и нажать Tab. Ничего не происходит?

14. теперь нажмите Tab два раза подряд. Видите, терминал выдал вам список всех команд, начинающихся с apt.



```
user2@user2: ~
Файл Правка Вкладки Справка
user2@user2:~$ apt
apt aptd apt-key
apt-add-repository aptdcon apt-mark
apt-cache apt-extracttemplates apt-sortpkgs
apt-cdrom apt-ftpparchive apturl
apt-config apt-get apturl-gtk
user2@user2:~$ apt
```

Удобно, не правда ли? Особенно, если привыкнуть.

Автодополнение в терминале работает практически везде, и не только для команд, но так же для их аргументов и имён файлов. Поэкспериментируйте с ним, оно значительно сокращает время набора, да и вообще, терминал без автодополнения

В это не терминал.

Еще одна хитрость. Откройте терминал нажмите сочетание клавиш Ctrl+R и начните набирать нужную команду. Терминал автоматически подставит подходящие варианты из набранных ранее команд.

## Работа с файлами

Начну рассказ наверно с того, что в любой момент времени работы в терминале вы находитесь в некотором каталоге. При запуске терминала текущей директорией является домашний каталог пользователя, но потом вы конечно можете её поменять.

Узнать, в какой же папке вы сейчас находитесь, очень просто, достаточно посмотреть на приглашение терминала, то есть на те символы, которые печатаются автоматически в начале каждой строки:

```
user2@user2: /usr/share/applications
Файл Правка Вкладки Справка
user2@user2:/usr/share/applications$
```



Текущий каталог - это то, что между символами : и \$. Кстати, обратите внимание, перед : стоит имя пользователя и имя компьютера, разделённые символом @. Но когда вы запускаете терминал, то между : и \$ стоит символ ~:

```
user2@user2: ~
Файл Правка Вкладки Справка
user2@user2:~$
```

При запуске терминала текущим каталогом становится домашняя папка пользователя, так вот, символ ~ - это синоним адреса домашней папки текущего пользователя<sup>2)</sup>. Кстати, если вы ещё не знаете, полный адрес домашнего каталога выглядит как

`/home/логин_пользователя`

В данном случае это `/home/user2`.

Хорошо, как выяснить текущее местоположение, вроде разобрались, а как же его сменить? Для этого предназначена команда `cd`, выполните, например, команду

`cd Музыка`

Видите, текущее местоположение изменилось:

```
user2@user2: ~/Музыка
Файл Правка Вкладки Справка
user2@user2:~$ cd Музыка
user2@user2:~/Музыка$
```



На всякий случай напоминаю, что имена файлов и каталогов в Linux чувствительны к регистру символов, то есть Музыка и музыка - это два совершенно разных имени.

После команды `cd` можно указывать как полные пути относительно корня, так и относительные, отсчитывающиеся от текущего каталога. В примере выше я использовал относительный путь. А вот полный:

```
cd /home/user2/Музыка
```

Хочу сразу обратить внимание на несколько важных особенностей. Во-первых, при наборе путей так же работает автодополнение по Tab, это очень удобно. Во-вторых, использовать различные небуквенные символы и пробелы напрямую при наборе путей нельзя. Например, для того, чтобы перейти в каталог, содержащий в имени символ пробела, надо при наборе пути к такому каталогу перед пробелом поставить символ обратного слеша `\`. Вот так:

```
cd Каталог\ с\ плохими\ символами\ в\ имени\<>
```

Установка обратного слеша перед некоторыми символами называется *экранированием*. Кстати, при использовании автодополнения все слеша расставляются автоматически. Кроме того, можно просто заключить путь в двойные кавычки:

```
cd "Каталог с плохими символами в имени<>"
```

Но в этом случае автодополнение работать не будет.

Заменитель адреса домашнего каталога `~` можно использовать и при наборе путей, например:

```
cd ~/Музыка
```

9. для перемещения непосредственно в домашний каталог достаточно просто набрать `cd` без аргументов.

Для перемещения на каталог выше можно использовать команду

```
cd ..
```

Вообще, две точки обозначают всегда родительский каталог, поэтому можно делать так:

```
cd ../..
```



В принципе, всё можно как угодно комбинировать, в разумных пределах, конечно. Ну и напоследок про `cd`. Переместиться в предыдущий посещённый каталог можно командой

`cd -`

Как перемещаться по каталогам в первом приближении вроде разобрались, теперь же я расскажу про некоторые другие полезные операции. Посмотреть содержимое текущего каталога можно командой `ls`:

```
user2@user2: ~
Файл Правка Вкладки Справка
user2@user2:~$ ls
lxterminal.png screenshots Загрузки Общедоступные
Mail Видео Изображения Рабочий стол
pochta.ru Документы Музыка Шаблоны
user2@user2:~$
```

Обычно командам можно передавать различные модификаторы, например:

```
user2@user2: ~
Файл Правка Вкладки Справка
user2@user2:~$ ls -l
итого 60
-rw-rw-r-- 1 user2 user2 14382 марта 12 19:21 lxterminal.png
drwx----- 8 user2 user2 4096 марта 2 08:19 Mail
drwx----- 8 user2 user2 4096 марта 2 21:26 pochta.ru
drwxrwxr-x 2 user2 user2 4096 марта 12 19:41 screenshots
drwxr-xr-x 2 user2 user2 4096 марта 1 14:32 Видео
drwxr-xr-x 3 user2 user2 4096 марта 11 15:22 Документы
drwxr-xr-x 2 user2 user2 4096 марта 1 14:32 Загрузки
drwxr-xr-x 2 user2 user2 4096 марта 11 19:49 Изображения
drwxr-xr-x 2 user2 user2 4096 марта 1 14:32 Музыка
drwxr-xr-x 2 user2 user2 4096 марта 1 14:32 Общедоступные
drwxr-xr-x 2 user2 user2 4096 марта 11 10:59 Рабочий стол
drwxr-xr-x 2 user2 user2 4096 марта 1 14:32 Шаблоны
user2@user2:~$
```



Кстати, эта команда показывает различную дополнительную информацию о содержимом каталога.

Очень часто параметрами команд являются имена файлов или папок, именно для этого я так подробно рассказывал о способе перемещения по каталогам и использования различных имен файлов в качестве аргументов. Например, команда `cat` показывает содержимое текстового файла, и если вы хотите посмотреть содержимое файла `test.txt`, лежащего в вашем домашнем каталоге, то вы могли бы выполнить команду `cat ~/text.txt`

### Получение справки

Начинающих пользователей Linux очень часто любят пугать так называемыми манами. Дело в том, что `man` - это система справки о командах для терминала. Пользоваться ей очень легко, просто наберите в терминале

```
man команда
```

Например:

```
man ls
```

Появится собственно текст справки<sup>3)</sup>, разбитый на разделы. Перемещаться по нему можно с помощью стрелок и клавиш `PgUp` и `PgDown`, а для выхода просто нажмите `Q`.

Кроме `man`-страниц у многих утилит<sup>4)</sup> есть встроенная справка, которую обычно можно посмотреть, запустив программу с ключом `--help`:

```
утилита --help
```

Например:

```
ls --help
```

Есть и другие способы получения помощи, например похожая на `man` утилита `info`. Но чаще всего наиболее полную информацию о программе можно получить именно из `man`-страниц, а краткую справку - указав ключ `--help` при вызове.

### История введённых команд

Терминал хранит историю введённых пользователем команд, которую вы можете листать в реальном режиме стрелками вверх и вниз на клавиатуре. Это очень



удобно для повторного исполнения введённых ранее команд. А посмотреть всю историю можно командой *history*

с каждой команды в истории есть номер, выполнить снова команду с определённым номером можно набрав в терминале восклицательный знак и номер нужной команды:

```
user2@user2: ~
Файл Правка Вкладки Справка
2 lsb release -a 2> /dev/null | grep -P "(?<=Codename:)(.*)"
3 cd /usr/share/applications
4 cd ~
5 cd Музыка
6 cd ~
7 ls
8 ls -l
9 history
user2@user2:~$!8
ls -l
итого 60
-rw-rw-r-- 1 user2 user2 14382 марта 12 19:21 lxterminal.png
drwx----- 8 user2 user2 4096 марта 2 08:19 Mail
drwx----- 8 user2 user2 4096 марта 2 21:26 pochta.ru
drwxrwxr-x 2 user2 user2 4096 марта 12 19:42 screenshots
drwxr-xr-x 2 user2 user2 4096 марта 1 14:32 Видео
drwxr-xr-x 3 user2 user2 4096 марта 11 15:22 Документы
drwxr-xr-x 2 user2 user2 4096 марта 1 14:32 Загрузки
drwxr-xr-x 2 user2 user2 4096 марта 11 19:49 Изображения
drwxr-xr-x 2 user2 user2 4096 марта 1 14:32 Музыка
drwxr-xr-x 2 user2 user2 4096 марта 1 14:32 Общедоступные
drwxr-xr-x 2 user2 user2 4096 марта 11 10:59 Рабочий стол
drwxr-xr-x 2 user2 user2 4096 марта 1 14:32 Шаблоны
user2@user2:~$
```

# повторить предыдущую набранную команду можно просто написав два восклицательных знака !! Двигаться по истории набранных команд можно стрелочками вверх/вниз клавиатуры. Нажав один раз на стрелку вверх - терминал покажет последнюю набранную команду, два раза - предпоследнюю, и так далее.

Если нужно вернуться назад - нажмите стрелочку вниз.

Мне кажется, что с непривычки уже всё описанное может показаться дремучим лесом, а на самом деле это только самая верхушка айсберга, существуют ещё тысячи полезных команд и интересных приёмов работы в терминале. С помощью терминала можно редактировать файлы, слушать музыку, смотреть видео и выполнять ещё массу повседневных операций.

### Дополнительные возможности терминала

Бывает так, что вы что-то запустили в терминале и хотите прервать работу этого чего-то. Обычно это сделать очень просто, достаточно нажать на клавиатуре сочетание клавиш **Ctrl+C**<sup>6)</sup>.



Есть и другие управляющие сочетания, например Ctrl+D посылает сигнал конца файла запущенному приложению, а без запущенных утилит делает тоже, что и терминальная команда exit. Ну а если вы хотите более подробно управлять работающими программами, то посмотрите на системный монитор htop, который, правда, нужно доустанавливать отдельно.

Если у вас сложилось впечатление, что терминал - это какая-то примитивная программа, способная выполнять очень простые команды, то это ложное впечатление. На самом деле есть очень много консольных утилит с богатейшими возможностями. Например, как уже упоминалось выше, серверные версии Ubuntu поставляются без графической оболочки. С помощью только консольных утилит можно настроить и управлять сложнейшими многофункциональными серверами.

Рассмотрим простой консольный файловый менеджер, он доустанавливается отдельно. Введите в терминале команду

mc

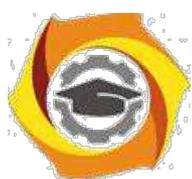
и увидите, что получится:



б. пользователей постарше, успевших поработать в DOS, это окно может вызвать острый приступ ностальгии.

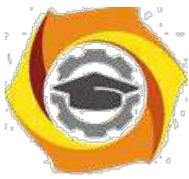
## 6.2. Задание:

3. Выполнить импорт конфигурации виртуальной машины – сервера



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

UbuntuServer01 в Oracle VM VirtualBox



- Пока идёт импорт (около 15 минут) изучить теоретический материал. Сомостоятельно выполнить поиск статей в Интернет, посвящённых Ubuntu Server.
- Запустить виртуальную машину UbuntuServer01. Войти в терминал с логином admin1 и паролем 1.
- Добавьте себя в список пользователей системы командой `sudo adduser <ваша фамилия>`, (латиницей), система потребует пароль – введите текущий – 1, затем введите свой новый пароль (при вводе символы не отображаются!) и информацию о себе.

```
admin1@ubuntuL01:~$ sudo adduser tkachuk
Добавляется пользователь «tkachuk» ...
Добавляется новая группа «tkachuk» (1001) ...
Добавляется новый пользователь «tkachuk» (1001) в группу «tkachuk» ...
Создаётся домашний каталог «/home/tkachuk» ...
Копирование файлов из «/etc/skel» ...
Введите новый пароль UNIX:
Повторите ввод нового пароля UNIX:
passwd: password updated successfully
Changing the user information for tkachuk
Enter the new value, or press ENTER for the default
 Full Name []: Евгений Ткачук
 Room Number []: 205
 Work Phone []:
 Home Phone []:
 Other []:
chfn: name with non-ASCII characters: 'Евгений Ткачук'
Данная информация корректна? [Y/n] Y
admin1@ubuntuL01:~$ _
```

- Для того, что вы могли выполнять команды администратора, введите себя в группу `sudo` командой `usermod -aG sudo <ваша фамилия>`

```
admin1@ubuntuL01:~$ sudo usermod -aG sudo tkachuk
admin1@ubuntuL01:~$
```

- Сохраните скриншот выполненных команд и включите его в отчет. Для того, чтобы сделать скриншот, надо выйти из виртуальной машины нажатием правой кнопки Shift.
- Выполните перезагрузку системы командой `sudo shutdown -r now`



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

8. Зайдите систему по своим именем, выполните все команды, описанные в теоретическом материале, скриншоты включите в отчёт.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

80

8. В отчете в свободной форме опишите свои действия, сопроводив их скриншотами.



## 8. Практическое занятие № 4 Мониторинг процессов ОС UBUNTU (Linux)

### 7.1. Теоретические сведения

Для просмотра запущенных процессов в **Ubuntu Linux** при помощи терминала, необходимо набрать в нем следующие команды:

#### **top – команда выдачи данных об активности процессов в Ubuntu**

Программа **top** динамически выдает в режиме реального времени информации 17.работающей системе, показывает запущенные процессы и потребление ими ресурсов системы. По умолчанию выдает задачи, наиболее загружающие процессор сервера, и обновляет список каждые пять секунд. При выполнении **top** в верхней части окна отображается астрономическое время, время, прошедшее с момента запуска системы, число пользователей в системе, число запущенных процессов и число процессов, находящихся в разных состояниях, данные об использовании ЦПУ, памяти и свопа. Далее идет таблица, характеризующая отдельные процессы. Число строк, отображаемых в этой таблице, определяется размером окна: сколько строк помещается, столько и выводится. Список процессов может быть отсортирован по используемому времени ЦПУ (по умолчанию), по использованию памяти, по PID, по времени исполнения. Переключать режимы отображения можно с помощью команд, которые программа **top** воспринимает. Это следующие команды (просто нажимайте соответствующие клавиши, только с учетом регистра, то есть вместе с клавишей Shift):

Shift+N — сортировка по PID;

Shift+A — сортировать процессы по возрасту;

Shift+P — сортировать процессы по использованию ЦПУ;

Shift+M — сортировать процессы по использованию памяти;

Shift+T — сортировка по времени выполнения.

Кроме команд, определяющих режим сортировки, команда **top** воспринимает еще ряд команд, которые позволяют управлять процессами в интерактивном режиме. С



помощью команды можно завершить некоторый процесс (его PID будет запрошен), а с помощью команды можно переопределить значение nice для некоторого процесса. Таким образом, эти две команды аналогичны командам **kill** и **renice**.

Команду **top** можно использовать со следующими параметрами:

**t** – Включение и выключение выдачи на экран суммарных данных.

**m** – Включение и выключение выдачи на экран информации об использовании памяти.

**A** – Сортировка строк по максимальному потреблению различных системных ресурсов. Полезна для быстрой идентификации задач, для которых в системе не хватает ресурсов.

**f** – Вход в меню интерактивного конфигурирования данных, выдаваемых на экран командой **top**. Полезна для настройки команды **top** для выполнения специфической задачи.

**o** – Позволяет вам интерактивно задавать порядок строк, выдаваемой командой **top**.

**r** – Изменение приоритета процессов с помощью команды **renice**.

**k** – Удаление процесса с помощью команды **kill**.

**z** – Переключение между цветным / монохромным вариантом выдачи изображения.

**11.** – Завершение работы, выход из программы.

### Использование Top Command - Task Manager для Ubuntu

1- Top Command

2- Просмотр списка команд – аналог help

3- Добавить убрать Field

4- Поменять порядок Field

5- Распорядок в Field

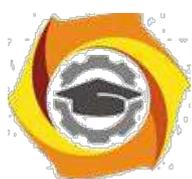
6- Сортировать по CPU

7- Отобразить процессы определенные пользователем

8- Выделить запущенные процессы

9- Смотреть абсолютный путь процессов

10- Убить процесс



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**1- Top Command**



Вы уже знакомы с **Task Manager** (управление задачей) в операционной системе **Windows**, вы зададите вопрос, есть ли что-то похожее с **Task Manager** в **Ubuntu Server** или нет

В **Windows**, **Task Manager** это полезная программа, которая позволяет узнать, какие приложения работают в системе и количество ресурсов используемое этими приложениями

**Ubuntu** обеспечивает вас подобной программой, но интерфейс этого приложения очень рудиментарный. Это приложение называется "**Top Command**"

Для запуска "**Top Command**" запустить команду **top**:

```
tran@euve251379:~$ top
```

### Top Command:

```
top - 15:13:16 up 1 day, 9:54, 2 users, load average: 0.14, 0.18, 0.49
Tasks: 70 total, 1 running, 69 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.7 us, 0.2 sy, 0.0 ni, 99.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2638908 free, 2915432 used, 2834268 buff/cache
KiB Swap: 4194304 total, 4159204 free, 35100 used. 5073064 avail Mem

 PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
 4456 tran 20 0 5315664 1.059g 44304 S 4.0 13.2 16:50.11 java
 483 tran 20 0 101604 35444 11972 S 0.3 0.4 129:03.53 Xtightvnc
 4664 tran 20 0 6145068 904964 40500 S 0.3 10.8 4:19.12 java
 4804 tran 20 0 6013124 929168 15568 S 0.3 11.1 1:43.08 java
 1 root 20 0 188968 3188 1944 S 0.0 0.0 0:01.23 systemd
 2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd/1+
 3 root 20 0 0 0 0 S 0.0 0.0 0:00.00 khelper/10+
 57 root 20 0 64104 20224 20048 S 0.0 0.2 0:07.90 systemd-jo+
 62 root 20 0 41968 992 948 S 0.0 0.0 0:00.09 systemd-ud+
 218 root 20 0 65464 1380 1252 S 0.0 0.0 0:01.66 sshd
 220 message+ 20 0 42984 1716 1356 S 0.0 0.0 0:00.18 dbus-daemon
 225 root 20 0 30480 1276 1040 S 0.0 0.0 0:00.21 cron
 232 root 20 0 28492 1468 1312 S 0.0 0.0 0:00.10 systemd-lo+
 238 syslog 20 0 184632 1688 1132 S 0.0 0.0 0:02.36 rsyslogd
```



Примечание: Для того, чтобы выйти из " **Top Command** », нажмите кнопку 'q'.

## 2- Просмотр списка команд – аналог help

Нажмите на ' h ', вы можете просмотреть список команд и инструкцию, и нажмите "ESC", чтобы вернуться к главному экрану

```
tran@euve251379: ~
Z,B,E,e Global: 'Z' colors; 'B' bold; 'E'/'e' summary/task memory scale
l,t,m Toggle Summary: 'l' load avg; 't' task/cpu stats; 'm' memory info
0,1,2,3,I Toggle: '0' zeros; '1/2/3' cpus or numa node views; 'I' Irix mode
f,F,X Fields: 'f'/'F' add/remove/order/sort; 'X' increase fixed-width

L,&,<,> . Locate: 'L'/'&' find/again; Move sort column: '<'/'>' left/right
R,H,V,J . Toggle: 'R' Sort; 'H' Threads; 'V' Forest view; 'J' Num justify
c,i,S,j . Toggle: 'c' Cmd name/line; 'i' Idle; 'S' Time; 'j' Str justify
x,y . Toggle highlights: 'x' sort field; 'y' running tasks
z,b . Toggle: 'z' color/mono; 'b' bold/reverse (only if 'x' or 'y')
u,U,o,o . Filter by: 'u'/'U' effective/any user; 'o'/'O' other criteria
n,#,^O . Set: 'n'/'#' max tasks displayed; Show: Ctrl+'O' other filter(s)
C,... . Toggle scroll coordinates msg for: up,down,left,right,home,end

k,r Manipulate tasks: 'k' kill; 'r' renice
d or s Set update interval
W,Y Write configuration file 'W'; Inspect other output 'Y'
q Quit
(commands shown with '.' require a visible task display window)
Press 'h' or '?' for help with Windows,
Type 'q' or <Esc> to continue
```

## 3- Добавить убрать Field

Есть много Field (Поле информации) которые могут быть вам интересны. По умолчанию " **Top Command** " только отобразит некоторые Field . Нажмите 'f', чтобы добавить или удалить Field

```
tran@euve251379: ~
top - 15:55:24 up 1 day, 10:36, 2 users, load average: 0.17, 0.24, 0.25
Tasks: 71 total, 1 running, 70 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.0 us, 0.6 sy, 0.9 ni, 96.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2323776 free, 2871968 used, 3192864 buff/cache
KiB Swap: 4194304 total, 4161360 free, 32944 used. 5115710 avail Mem

 PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
 4456 tran 20 0 5340000 0.990g 44772 S 12.7 12.4 18:58.80 java
 7459 tran 30 10 42988 12416 1576 S 6.0 0.1 0:13.60 distort
 483 tran 20 0 101604 35444 11972 S 1.0 0.4 130:32.36 Xtightvnc
 546 tran 20 0 558400 22808 8960 S 0.3 0.3 1:58.07 Thunar
 4455 tran 20 0 4771948 56804 13024 S 0.3 0.7 0:04.61 java
 4664 tran 20 0 6145068 905844 40500 S 0.3 10.8 4:30.01 java
 4804 tran 20 0 6013124 944948 15568 S 0.3 11.3 1:55.35 java
 7527 tran 20 0 41048 1920 1412 R 0.3 0.0 0:00.06 top
28704 tran 20 0 388212 14704 10184 S 0.3 0.2 0:39.38 xfce4-term+
 1 root 20 0 188968 3188 1944 S 0.0 0.0 0:01.24 systemd
 2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd/1+
 3 root 20 0 0 0 0 S 0.0 0.0 0:00.00 khelper/10+
 57 root 20 0 64104 20268 20092 S 0.0 0.2 0:07.92 systemd-jo+
 62 root 20 0 41968 992 948 S 0.0 0.0 0:00.09 systemd-ud+
 218 root 20 0 65464 1380 1252 S 0.0 0.0 0:01.67 sshd
```

Field(s)



Field со звездочкой (\*) это Field, которые отображены, другие Field не отображаются на "Top Command"

```
tran@euve251379: ~
Fields Management for window i:Def, whose current sort field is %CPU
Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
'd' or <Space> toggles display, 's' sets sort. Use 'q' or <Esc> to end!
* PID = Process Id GID = Group Id SUPGIDS = Supp Groups I
* USER = Effective Use GROUP = Group Name SUPGRPS = Supp Groups N
* PR = Priority PGRP = Process Group TGID = Thread Group
* NI = Nice Value TTY = Controlling T ENVIRON = Environment v
* VIRT = Virtual Image TPGID = Tty Process G vMj = Major Faults
* RES = Resident Size SID = Session Id vMn = Minor Faults
* SHR = Shared Memory nTH = Number of Thr USED = Res+Swap Size
* S = Process Statu P = Last Used Cpu nsIPC = IPC namespace
* %CPU = CPU Usage TIME = CPU Time nsMNT = MNT namespace
* %MEM = Memory Usage SWAP = Swapped Size nsNET = NET namespace
* TIME+ = CPU Time, hun CODE = Code Size (Ki nsPID = PID namespace
* COMMAND = Command Name/ DATA = Data+Stack (K nsUSER = USER namespac
PPID = Parent Proces nMaj = Major Page Fa nsUTS = UTS namespace
UID = Effective Use nMin = Minor Page Fa
RUID = Real User Id nDRT = Dirty Pages C
RUSER = Real User Nam WCHAN = Sleeping in F
SUID = Saved User Id Flags = Task Flags <s
SUSER = Saved User Na CGROUPS = Control Group
```

Используйте клавишу со стрелками вверх или вниз, чтобы перейти к нужному Field

```
tran@euve251379: ~
Fields Management for window i:Def, whose current sort field is %CPU
Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
'd' or <Space> toggles display, 's' sets sort. Use 'q' or <Esc> to end!
* PID = Process Id GID = Group Id SUPGIDS = Supp Groups I
* USER = Effective Use GROUP = Group Name SUPGRPS = Supp Groups N
* PR = Priority PGRP = Process Group TGID = Thread Group
* NI = Nice Value TTY = Controlling T ENVIRON = Environment v
* VIRT = Virtual Image TPGID = Tty Process G vMj = Major Faults
* RES = Resident Size SID = Session Id vMn = Minor Faults
* SHR = Shared Memory nTH = Number of Thr USED = Res+Swap Size
* S = Process Statu P = Last Used Cpu nsIPC = IPC namespace
* %CPU = CPU Usage TIME = CPU Time nsMNT = MNT namespace
* %MEM = Memory Usage SWAP = Swapped Size nsNET = NET namespace
* TIME+ = CPU Time, hun CODE = Code Size (Ki nsPID = PID namespace
* COMMAND = Command Name/ DATA = Data+Stack (K nsUSER = USER namespac
PPID = Parent Proces nMaj = Major Page Fa nsUTS = UTS namespace
UID = Effective Use nMin = Minor Page Fa
RUID = Real User Id nDRT = Dirty Pages C
RUSER = Real User Nam WCHAN = Sleeping in F
SUID = Saved User Id Flags = Task Flags <s
SUSER = Saved User Na CGROUPS = Control Group
```

8. Нажмите на кнопку ENTER.
9. Нажмите пробел (Whitespace), чтобы выбрать или отменить выбор Field.



```
tran@euve251379: ~
Fields Management for window 1:Def, whose current sort field is %CPU
Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
'd' or <Space> toggles display, 's' sets sort. Use 'q' or <Esc> to end!

* PID = Process Id GID = Group Id SUPGIDS = Supp Groups I
* USER = Effective Use GROUP = Group Name SUPGRPS = Supp Groups N
PR = Priority PGRP = Process Group TGID = Thread Group
NI = Nice Value TTY = Controlling T ENVIRON = Environment v
VIRT = Virtual Image TPGID = Tty Process G vMj = Major Faults
* RES = Resident Size * SID = Session Id vMn = Minor Faults
* SHR = Shared Memory nTH = Number of Thr USED = Res+Swap Size
* S = Process Statu P = Last Used Cpu nsIPC = IPC namespace
* %CPU = CPU Usage TIME = CPU Time nsMNT = MNT namespace
* %MEM = Memory Usage SWAP = Swapped Size nsNET = NET namespace
* TIME+ = CPU Time, hun CODE = Code Size (Ki nsPID = PID namespace
* COMMAND = Command Name/ DATA = Data+Stack (K nsUSER = USER namespac
PPID = Parent Proces nMaj = Major Page Fa nsUTS = UTS namespace
UID = Effective Use nMin = Minor Page Fa
RUID = Real User Id nDRT = Dirty Pages C
RUSER = Real User Nam WCHAN = Sleeping in F
SUID = Saved User Id Flags = Task Flags <s
SUSER = Saved User Na CGROUPS = Control Group
```

Нажмите на " ESC" чтобы вернуться к главному экрану

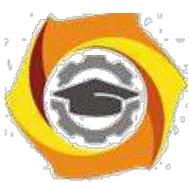
```
tran@euve251379: ~
top - 16:18:04 up 1 day, 10:59, 2 users, load average: 0.04, 0.21, 0.30
Tasks: 72 total, 1 running, 70 sleeping, 0 stopped, 1 zombie
%Cpu(s): 0.6 us, 0.2 sy, 1.5 ni, 97.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2307204 free, 2878552 used, 3202852 buff/cache
KiB Swap: 4194304 total, 4161364 free, 32940 used. 5106226 avail Mem

 PID USER RES SHR S %CPU %MEM TIME+ COMMAND SID
 8195 tran 21712 4648 S 9.3 0.3 0:31.33 ripples 465
 483 tran 30352 15044 S 2.3 0.4 131:08.91 Xtightvnc 465
 4456 tran 1.034g 44776 S 1.7 12.9 19:58.55 java 4411
 4455 tran 56804 13024 S 0.3 0.7 0:05.49 java 4411
 4664 tran 905780 40500 S 0.3 10.8 4:35.68 java 465
 1 root 3188 1944 S 0.0 0.0 0:01.25 systemd 1
 2 root 0 0 S 0.0 0.0 0:00.00 kthreadd/102329 0
 3 root 0 0 S 0.0 0.0 0:00.00 khelper/1023298 0
 57 root 20416 20240 S 0.0 0.2 0:07.96 systemd-journal 57
 62 root 992 948 S 0.0 0.0 0:00.09 systemd-udev 62
 218 root 1380 1252 S 0.0 0.0 0:01.67 sshd 218
 220 message+ 1716 1356 S 0.0 0.0 0:00.18 dbus-daemon 220
 225 root 1276 1040 S 0.0 0.0 0:00.21 cron 225
 232 root 1468 1312 S 0.0 0.0 0:00.10 systemd-logind 232
 238 syslog 1688 1132 S 0.0 0.0 0:02.38 rsyslogd 238
```

#### 4- Поменять порядок Field

Для того, чтобы улучшить вид, вы можете изменить порядок Field на экране "

Top Command"



```
Iran@euve251379: ~
top - 16:19:52 up 1 day, 11:00, 2 users, load average: 0.27, 0.22, 0.29
Tasks: 72 total, 1 running, 70 sleeping, 0 stopped, 1 zombie
%Cpu(s): 1.2 us, 0.3 sy, 1.4 ni, 97.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2302756 free, 2882536 used, 3203316 buff/cache
KiB Swap: 4194304 total, 4161364 free, 32940 used, 5102178 avail Mem

 PID USER RES SHR S %CPU %MEM TIME+ COMMAND SID
 8195 tran 24764 4648 S 9.0 0.3 0:40.89 ripples 465
 4456 tran 1.035g 44776 S 6.6 12.9 20:02.55 java 4411
 483 tran 31396 15044 S 2.0 0.4 131:11.67 Xtightvnc 465
 546 tran 22808 8960 S 0.3 0.3 2:01.94 Thunder 465
 7527 tran 1928 1416 R 0.3 0.0 0:00.67 top 5468
 1 root 0 0 S 0.0 0.0 0:01.25 systemd 1
 2 root 0 0 S 0.0 0.0 0:00.00 kthreadd/102325 0
 3 root 0 0 S 0.0 0.0 0:00.00 khelper/1021298 0
 57 root 20428 20252 S 0.0 0.2 0:07.96 systemd-journal 57
 62 root 992 948 S 0.0 0.0 0:00.09 systemd-udevd 62
 218 root 1380 1252 S 0.0 0.0 0:01.67 sshd 218
 220 message+ 1716 1356 S 0.0 0.0 0:00.18 dbus-daemon 220
 225 root 1276 1040 S 0.0 0.0 0:00.21 cron 225
 232 root 1468 1312 S 0.0 0.0 0:00.10 systemd-logind 232
 238 syslog 1688 1132 S 0.0 0.0 0:02.38 rsyslogd 238
```

Нажмите 'F', чтобы просмотреть список Field.

```
tran@euve251379: ~
Fields Management for window i:Def, whose current sort field is %CPU
Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
'd' or <Space> toggles display, 's' sets sort. Use 'q' or <Esc> to end!

* PID = Process Id GID = Group Id SUPGIDS = Supp Groups I
* USER = Effective Use GROUP = Group Name SUPGRPS = Supp Groups N
PR = Priority PGRP = Process Group TGID = Thread Group
NI = Nice Value TTY = Controlling T ENVIRON = Environment v
VIRT = Virtual Image TPGID = Tty Process G vMj = Major Faults
* RES = Resident Size * SID = Session Id vMn = Minor Faults
* SHR = Shared Memory nTH = Number of Thr USED = Res+Swap Size
* S = Process Statu P = Last Used Cpu nsIPC = IPC namespace
* %CPU = CPU Usage TIME = CPU Time nsMNT = MNT namespace
* %MEM = Memory Usage SWAP = Swapped Size nsNET = NET namespace
* TIME+ = CPU Time, hun CODE = Code Size (Ki nsPID = PID namespace
* COMMAND = Command Name/ DATA = Data+Stack (K nsUSER = USER namespac
PPID = Parent Proce nMaj = Major Page Fa nsUTS = UTS namespace
UID = Effective Use nMin = Minor Page Fa
RUID = Real User Id nDRT = Dirty Pages C
RUSER = Real User Nam WCHAN = Sleeping in F
SUID = Saved User Id Flags = Task Flags <s
SUSER = Saved User Na CGROUPS = Control Group
```

3. Нажмите клавишу со стрелкой направо, чтобы выделить Field для перемещения.
4. Нажмите стрелку вверх / вниз для перемещения Field (обозначение выше) в другое место.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

5. Нажмите кнопку ENTER для завершения.



```
tran@euve251379: ~
Fields Management for window |:Def|, whose current sort field is %CPU
Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
'd' or <Space> toggles display, 's' sets sort. Use 'q' or <Esc> to end!

* PID = Process Id GID = Group Id SUPGRIDS = Supp Groups I
* USER = Effective Use GROUP = Group Name SUPGRPS = Supp Groups N
PR = Priority PGRP = Process Group TGID = Thread Group
NI = Nice Value TTY = Controlling T ENVIRON = Environment v
VIRT = Virtual Image TPGID = Tty Process G vMj = Major Faults
* RES = Resident Size * SID = Session Id vMn = Minor Faults
* SHR = Shared Memory nTH = Number of Thr USED = Res+Swap Size
* S = Process Statu P = Last Used Cpu nsIPC = IPC namespace
* %CPU = CPU Usage TIME = CPU Time nsMNT = MNT namespace
* %MEM = Memory Usage SWAP = Swapped Size nsNET = NET namespace
* TIME+ = CPU Time, hun CODE = Code Size (Ki nsPID = PID namespace
* COMMAND = Command Name/ DATA = Data+Stack (K nsUSER = USER namespac
PPID = Parent Proces nMaj = Major Page Fa nsUTS = UTS namespace
UID = Effective Use nMin = Minor Page Fa
RUID = Real User Id nDRT = Dirty Pages C
RUSER = Real User Nam WCHAN = Sleeping in F
SUID = Saved User Id Flags = Task Flags <s
SUSER = Saved User Na CGROUPS = Control Group
```

Нажмите " ESC " чтобы вернуться к главному экрану

```
tran@euve251379: ~
top - 16:37:59 up 1 day, 11:18, 2 users, load average: 0.21, 0.27, 0.26
Tasks: 71 total, 1 running, 70 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.0 us, 0.6 sy, 0.3 ni, 95.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2366644 free, 2816668 used, 3205296 buff/cache
KiB Swap: 4194304 total, 4161452 free, 32852 used. 5171244 avail Mem

 PID RES USER SHR S %CPU %MEM TIME+ COMMAND SID
 483 28324 tran 11972 S 20.7 0.3 133:30.85 Xtightvnc 465
 4456 1.047g tran 44776 S 5.7 13.1 20:31.08 java 4411
 8717 1904 tran 1544 S 2.7 0.0 0:09.84 cwaves 465
 1 3188 root 1944 S 0.0 0.0 0:01.25 systemd 1
 2 0 root 0 S 0.0 0.0 0:00.00 kthreadd/102329 0
 3 0 root 0 S 0.0 0.0 0:00.00 khelper/1023298 0
 57 20444 root 20268 S 0.0 0.2 0:07.96 systemd-journal 57
 62 992 root 948 S 0.0 0.0 0:00.10 systemd-udev 62
 218 1380 root 1252 S 0.0 0.0 0:01.67 sshd 218
 220 1716 message+ 1356 S 0.0 0.0 0:00.18 dbus-daemon 220
 225 1276 root 1040 S 0.0 0.0 0:00.21 cron 225
 232 1468 root 1312 S 0.0 0.0 0:00.10 systemd-logind 232
 238 1688 syslog 1132 S 0.0 0.0 0:02.38 rsyslogd 238
 289 644 root 640 S 0.0 0.0 0:00.00 xinetd 289
 300 728 root 724 S 0.0 0.0 0:00.00 agetty 300
```

### 5- Сортировка в Field

Предположим, вы хотите знать, какие программы используют наибольший объем памяти, вам нужно сортировать по % MEM ( % Memory).



```
tran@euve251379: ~
top - 16:41:13 up 1 day, 11:22, 2 users, load average: 0.23, 0.27, 0.26
Tasks: 71 total, 1 running, 70 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3.9 us, 0.4 sy, 0.4 ni, 95.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2364644 free, 2818168 used, 3205796 buff/cache
KiB Swap: 4194304 total, 4161484 free, 32820 used. 5169720 avail Mem
```

| PID   | RES    | USER     | SHR    | S | %CPU | %MEM | TIME+     | COMMAND         | SID  |
|-------|--------|----------|--------|---|------|------|-----------|-----------------|------|
| 4456  | 1.048g | tran     | 44776  | S | 3.7  | 13.1 | 20:36.68  | java            | 4411 |
| 4664  | 905788 | tran     | 40500  | S | 0.3  | 10.8 | 4:41.54   | java            | 465  |
| 4804  | 847044 | tran     | 15572  | S | 0.0  | 10.1 | 2:28.97   | java            | 465  |
| 308   | 116244 | postgres | 114776 | S | 0.0  | 1.4  | 0:02.26   | postgres        | 308  |
| 309   | 114912 | postgres | 113708 | S | 0.0  | 1.4  | 0:04.32   | postgres        | 309  |
| 4455  | 56804  | tran     | 13024  | S | 0.0  | 0.7  | 0:06.45   | java            | 4411 |
| 4850  | 31244  | postgres | 23604  | S | 0.0  | 0.4  | 0:02.30   | postgres        | 4850 |
| 483   | 28800  | tran     | 11972  | S | 21.6 | 0.3  | 134:14.36 | Xtightvnc       | 465  |
| 546   | 22812  | tran     | 8960   | S | 0.3  | 0.3  | 2:04.49   | Thunar          | 465  |
| 57    | 20444  | root     | 20268  | S | 0.0  | 0.2  | 0:07.96   | systemd-journal | 57   |
| 28704 | 14704  | tran     | 10184  | S | 0.0  | 0.2  | 0:40.63   | xfce4-terminal  | 465  |
| 548   | 14208  | tran     | 10280  | S | 0.0  | 0.2  | 0:08.47   | xfdesktop       | 465  |
| 306   | 11004  | postgres | 9712   | S | 0.0  | 0.1  | 0:41.58   | postgres        | 299  |
| 544   | 9112   | tran     | 5940   | S | 0.0  | 0.1  | 0:10.86   | xfce4-panel     | 465  |
| 4663  | 8700   | tran     | 6952   | S | 0.0  | 0.1  | 0:00.02   | eclipse         | 465  |

Нажмите 'F', чтобы просмотреть список Field.

6. Выберите Field для сортировки.
7. Нажмите 'S'.
8. Нажмите "ESC", чтобы вернуться к главному экрану.

```
tran@euve251379: ~
Fields Management for window 1:Def, whose current sort field is RES
Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
'd' or <Space> toggles display, 's' sets sort. Use 'q' or <Esc> to end!
```

|           |                 |         |                 |         |                 |
|-----------|-----------------|---------|-----------------|---------|-----------------|
| * PID     | = Process Id    | GID     | = Group Id      | SUPGIDS | = Supp Groups I |
| PR        | = Priority      | GROUP   | = Group Name    | SUPGRPS | = Supp Groups N |
| NI        | = Nice Value    | PGRP    | = Process Group | TGID    | = Thread Group  |
| VIRT      | = Virtual Image | TTY     | = Controlling T | ENVIRON | = Environment v |
| * RES     | = Resident Size | TPGID   | = Tty Process G | vMj     | = Major Faults  |
| * USER    | = Effective Use | * SID   | = Session Id    | vMn     | = Minor Faults  |
| * SHR     | = Shared Memory | nTH     | = Number of Thr | USED    | = Res+Swap Size |
| * S       | = Process Statu | P       | = Last Used Cpu | nsIPC   | = IPC namespace |
| * %CPU    | = CPU Usage     | TIME    | = CPU Time      | nsMNT   | = MNT namespace |
| * %MEM    | = Memory Usage  | SWAP    | = Swapped Size  | nsNET   | = NET namespace |
| * TIME+   | = CPU Time, hun | CODE    | = Code Size (Ki | nsPID   | = PID namespace |
| * COMMAND | = Command Name/ | DATA    | = Data+Stack (K | nsUSER  | = USER namespac |
| PPID      | = Parent Proces | nMaj    | = Major Page Fa | nsUTS   | = UTS namespace |
| UID       | = Effective Use | nMin    | = Minor Page Fa |         |                 |
| RUID      | = Real User Id  | nDRT    | = Dirty Pages C |         |                 |
| RUSER     | = Real User Nam | WCHAN   | = Sleeping in F |         |                 |
| SUID      | = Saved User Id | Flags   | = Task Flags <s |         |                 |
| SUSER     | = Saved User Na | CGROUPS | = Control Group |         |                 |



```
Iran@euve251379: ~
top - 16:41:13 up 1 day, 11:22, 2 users, load average: 0.23, 0.27, 0.26
Tasks: 71 total, 1 running, 70 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3.9 us, 0.4 sy, 0.4 ni, 95.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2364644 free, 2818168 used, 3205796 buff/cache
KiB Swap: 4194304 total, 4161484 free, 32820 used, 5169720 avail Mem

 PID RES USER SHR S %CPU %MEM TIME+ COMMAND SIB
 4456 1.048g tran 44776 S 3.7 13.1 20:36.68 java 4411
 4664 905788 tran 40580 S 0.3 10.8 4:41.54 java 465
 4804 647044 tran 15572 S 0.0 10.1 2:28.97 java 465
 308 116244 postgres 114776 S 0.0 1.4 0:02.26 postgres 308
 309 114912 postgres 113708 S 0.0 1.4 0:04.32 postgres 309
 4456 56804 tran 13024 S 0.0 0.7 0:06.45 java 4411
 4850 31244 postgres 23604 S 0.0 0.4 0:02.30 postgres 4850
 483 28800 tran 11972 S 21.6 0.3 134:14.36 Xtightvnc 465
 546 22812 tran 8980 S 0.3 0.3 2:04.49 Thunar 465
 57 20444 root 20268 S 0.0 0.2 0:07.96 systemd-journal 57
28704 14704 tran 10184 S 0.0 0.2 0:40.63 xfce4-terminal 465
 548 14208 tran 10280 S 0.0 0.2 0:08.47 xfdesktop 465
 306 11004 postgres 9712 S 0.0 0.1 0:41.58 postgres 299
 544 9112 tran 5940 S 0.0 0.1 0:10.86 xfce4-panel 465
 4663 8700 tran 6952 S 0.0 0.1 0:00.02 eclipse 465
```

## 6- Сортировать по CPU

CPU (field) имеет важное значение, он говорит вам какой процесс (Process) заставляет компьютер обрабатывать больше всего на данный момент. Нажмите " **Shift** 7.Р 'для сортировки по CPU.

```
tran@euve251379: ~
top - 18:05:55 up 1 day, 12:46, 2 users, load average: 0.66, 0.54, 0.37
Tasks: 73 total, 1 running, 71 sleeping, 0 stopped, 1 zombie
%Cpu(s): 1.0 us, 0.2 sy, 0.1 ni, 98.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2261608 free, 2871348 used, 3255652 buff/cache
KiB Swap: 4194304 total, 4161928 free, 32376 used, 5108082 avail Mem

 PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
 483 tran 20 0 202476 136088 17292 S 5.3 1.6 136:22.10 Xtightvnc
 4456 tran 20 0 5358456 1.093g 44776 S 1.3 13.7 22:51.49 java
 4664 tran 20 0 6145068 908600 40580 S 0.7 10.8 5:12.30 java
10895 tran 30 10 40464 10124 1560 S 0.7 0.1 4:51.02 tessellima+
 4455 tran 20 0 4771948 56804 13024 S 0.3 0.7 0:09.83 java
 4804 tran 20 0 6017200 621304 15560 S 0.3 7.4 3:43.93 java
10514 tran 20 0 41048 1940 1436 R 0.3 0.0 0:00.26 top
 453 tran 20 0 44928 1684 1480 S 0.0 0.0 0:00.01 systemd
 454 tran 20 0 60860 488 60 S 0.0 0.0 0:00.00 (sd-pam)
 487 tran 20 0 4452 588 584 S 0.0 0.0 0:00.00 xstartup
 490 tran 20 0 4452 612 608 S 0.0 0.0 0:00.00 sh
 514 tran 20 0 11084 132 88 S 0.0 0.0 0:00.36 ssh-agent
 517 tran 20 0 43544 456 452 S 0.0 0.0 0:00.00 dbus-launch
 518 tran 20 0 43056 1428 908 S 0.0 0.0 0:00.42 dbus-daemon
```

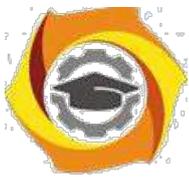
## 7- Отобразить процессы определенные пользователем

Используйте опцию '-u', который позволяет указать пользователя, и увидеть какие процессы выполняются этим пользователем.



[?](#)

**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



top -u

user\_name

Например, смотреть процессы (process) выполняемые пользователем 'Тран'

```
tran@euve251379:~$ top -u tran
```

```
top - 17:10:59 up 1 day, 11:51, 2 users, load average: 0.05, 0.09, 0.13
Tasks: 72 total, 1 running, 71 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.1 us, 0.0 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2353848 free, 2821936 used, 3212824 buff/cache
KiB Swap: 4194304 total, 4161484 free, 32820 used, 5165256 avail Mem

 PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
 4456 tran 20 0 5358456 1.094g 44776 S 0.7 13.7 21:53.29 java
 4664 tran 20 0 6145068 905808 40500 S 0.3 10.8 4:48.78 java
 4804 tran 20 0 6015176 798128 15576 S 0.3 9.5 2:48.47 java
 453 tran 20 0 44928 1684 1480 S 0.0 0.0 0:00.01 systemd
 454 tran 20 0 60860 488 60 S 0.0 0.0 0:00.00 (sd-pam)
```

## 8- Выделить запущенные процессы

Нажмите 'Z', программа " Top Command" покажет процессы, запущенные с цветами. Это позволяет легко определить запущенные процессы.

```
top - 17:17:44 up 1 day, 11:58, 2 users, load average: 0.48, 0.23, 0.17
Tasks: 72 total, 2 running, 70 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.1 us, 0.2 sy, 0.0 ni, 98.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2266396 free, 2909100 used, 3213112 buff/cache
KiB Swap: 4194304 total, 4161512 free, 32792 used, 5078314 avail Mem

 PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
 488 tran 20 0 195360 130280 11972 S 5.8 1.6 134:47.84 Xlightvnc
 4456 tran 20 0 5358456 1.093g 44776 S 1.7 13.7 22:00.62 java
 9874 tran 30 10 40468 10128 1560 S 0.7 0.1 0:56.88 tessellima+
 806 postgres 20 0 806704 11004 9712 S 0.3 0.1 0:42.22 postgres
 9887 tran 20 0 41048 1972 1416 R 0.3 0.0 0:00.07 top
 1 root 20 0 188968 3188 1944 S 0.0 0.0 0:01.06 systemd
 2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd/I+
 3 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd/I+
 57 root 20 0 64104 20476 20300 S 0.0 0.2 0:07.98 systemd-jo+
 62 root 20 0 41968 992 948 S 0.0 0.0 0:00.10 systemd-ud+
 218 root 20 0 65464 1380 1252 S 0.0 0.0 0:01.68 sshd
 220 message+ 20 0 42984 1716 1388 S 0.0 0.0 0:00.18 dbus-daemon
 225 root 20 0 30480 1276 1040 S 0.0 0.0 0:00.22 cron
 232 root 20 0 28492 1468 1612 S 0.0 0.0 0:00.11 systemd-lo+
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**9- Смотреть абсолютный путь процессов**



При работе процессов, если вы хотите, чтобы увидеть абсолютный путь файла, выполняющего этот процесс? Просто нажмите 'с'

```
tran@euve251379: ~
top - 17:24:32 up 1 day, 12:05, 2 users, load average: 0.47, 0.48, 0.32
Tasks: 71 total, 2 running, 69 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.5 sy, 15.8 ni, 83.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2316940 free, 2858484 used, 3213184 buff/cache
KiB Swap: 4194304 total, 4161512 free, 32792 used. 5129496 avail Mem

S %CPU %MEM TIME+ COMMAND
R 94.7 0.1 4:02.39 tessellimage
S 3.0 1.1 135:04.23 Xtightvnc
S 1.5 0.3 2:10.29 Thunar
S 0.0 0.0 0:01.27 systemd
S 0.0 0.0 0:00.00 kthreadd/102329
S 0.0 0.0 0:00.00 khelper/1023298
S 0.0 0.2 0:07.99 systemd-journal
S 0.0 0.0 0:00.10 systemd-udev
S 0.0 0.0 0:01.68 sshd
S 0.0 0.0 0:00.18 dbus-daemon
S 0.0 0.0 0:00.22 cron
S 0.0 0.0 0:00.11 systemd-logind
S 0.0 0.0 0:02.40 rsyslogd
S 0.0 0.0 0:00.00 xinetd
```



```
tran@euve251379: ~
top - 17:25:06 up 1 day, 12:06, 2 users, load average: 0.71, 0.53, 0.34
Tasks: 71 total, 2 running, 69 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.4 us, 0.6 sy, 16.1 ni, 83.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2285272 free, 2890000 used, 3213336 buff/cache
KiB Swap: 4194304 total, 4161512 free, 32792 used. 5097868 avail Mem

S %CPU %MEM TIME+ COMMAND
R 96.7 0.1 4:35.88 tessellimage -root
S 2.3 1.4 135:04.83 Xtightvnc :2 -desktop X -auth /home/tran/.Xauthority -g+
S 2.0 13.7 22:09.51 /usr/lib/jvm/java-8-oracle/bin/java -server -Xms256m -X+
S 0.7 9.2 2:59.63 /usr/lib/jvm/java-8-oracle/bin/java -Declipse.ignoreApp+
S 0.3 0.3 2:10.36 Thunar --daemon
S 0.3 10.8 4:52.77 /usr/bin/java -Dosgi.requiredJavaVersion=1.8 -XX:+UseG1+
S 0.0 0.0 0:01.27 init -z
S 0.0 0.0 0:00.00 [kthreadd/102329]
S 0.0 0.0 0:00.00 [khelper/1023298]
S 0.0 0.2 0:07.99 /lib/systemd/systemd-journald
S 0.0 0.0 0:00.10 /lib/systemd/systemd-udev
S 0.0 0.0 0:01.68 /usr/sbin/sshd -D
S 0.0 0.0 0:00.18 /usr/bin/dbus-daemon --system --address=systemd: --nofo+
S 0.0 0.0 0:00.22 /usr/sbin/cron -f
```

## 10- Убить процесс

В Windows вы можете закончить задание которое выполняется, используя "End Task".

Ubuntu вы можете использовать "Kill", чтобы убить процесс

8. Нажмите 'к'.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

9. Введите PID (Process ID).



## 10. Нажмите кнопку ENTER

```
tran@euve251379: ~
top - 17:38:14 up 1 day, 12:19, 2 users, load average: 0.21, 0.25, 0.29
Tasks: 72 total, 1 running, 71 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.4 us, 0.2 sy, 0.0 ni, 99.4 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8388608 total, 2253748 free, 2890620 used, 3244240 buff/cache
KiB Swap: 4194304 total, 4161920 free, 32384 used, 5088958 avail Mem
PID to signal/kill [default pid = 4456] 10421
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
4456 tran 20 0 5358456 1.094g 44776 S 1.0 13.7 22:22.66 java
10421 tran 20 0 1188340 178772 62736 S 1.0 2.1 0:07.96 firefox
4804 tran 20 0 6015176 748900 15576 S 0.3 8.9 3:06.36 java
10514 tran 20 0 41048 1932 1428 R 0.3 0.0 0:00.09 top
453 tran 20 0 44928 1684 1480 S 0.0 0.0 0:00.01 systemd
454 tran 20 0 60860 488 60 S 0.0 0.0 0:00.00 (sd-pam)
483 tran 20 0 101104 36104 17292 S 0.0 0.4 135:49.14 Xtightvnc
487 tran 20 0 4452 588 584 S 0.0 0.0 0:00.00 xstartup
490 tran 20 0 4452 612 608 S 0.0 0.0 0:00.00 sh
514 tran 20 0 11084 132 88 S 0.0 0.0 0:00.35 ssh-agent
517 tran 20 0 43544 456 452 S 0.0 0.0 0:00.00 dbus-launch
518 tran 20 0 43056 1428 908 S 0.0 0.0 0:00.42 dbus-daemon
528 tran 20 0 322192 4136 3384 S 0.0 0.0 0:00.07 xfce4-sess+
536 tran 20 0 47584 1904 1644 S 0.0 0.0 0:00.10 xfconfd
```

### Другие команды мониторинга процессов Ubuntu

#### ps – команда выдачи списка процессов Ubuntu

Команда **ps** выдаст краткий список текущих процессов. Вывод команды **ps** схож с выводом команды **top**, однако он отображает статический снимок процессов. Для того, чтобы выбрать все процессы, используете параметр - **A** или - **e**

#### Вывод большего количества данных по процессам

##### ps -A

Для того, чтобы включить выдачу всех данных (будут показаны аргументы командной строки, переданные в процесс):

##### ps -A

#### Вывод списка всех процессов Ubuntu

##### ps ax



ps ахц

**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



### Отображение потоков (LWP и NLWP)

**ps -AlFH**

### Вывод информации о параметрах безопасности Ubuntu

**ps -eo euser,ruser,suser,fuser,f,comm,label**

**ps axZ**

**ps -eM**

### Вывод дерева процессов

**ps -ejH**

**ps axjf**

**pstree**

### Отображение потоков после процессов

**ps -AlLm**

### Настраиваемая выдача данных

Позволяет выводить данные в последовательности, определяемой пользователем

**ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm**

**ps axo stat,euid,ruid,tty,tpgid,sess,pgrp,ppid,pid,pcpu,comm**

**ps -eopid,tt,user,fname,tmout,f,wchan**

### Вывод процессов, запущенных пользователем User

**ps -U User -u User u**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Вывод ID процессов, запущенных под apache**



```
ps -C apache -o pid=
```

или

```
pgrep apache
```

**Вывод имени для PID 30470**

```
ps -p 30470 -o comm=
```

**Вывод 10 процессов, потребляющих наибольшее количество памяти**

```
ps -auxf | sort -nr -k 4 | head -10
```

**Вывод 10 процессов, потребляющих наибольший ресурс процессора**

```
ps -auxf | sort -nr -k 3 | head -10
```

По умолчанию, команда `ps` выводит только информацию о процессах, запущенных в текущей сессии терминала `bash`. Для вывода информации по всем процессам необходимо ввести команду `ps` с параметром `-e`. Для отображения желаемых полей необходимо ввести команду `ps` с параметром `-o поле1,поле2,...`, где через запятую перечисляются поля, которые необходимо отобразить.

**free – использование памяти**

Команда `free` показывает общее количество свободной и используемой системой физической памяти и памяти свопинга, а также размеры буферов, используемые ядром.

```
free
```

**uptime – сообщает, как долго работает система**

Команду `uptime` можно использовать с тем, чтобы определить, как долго работает сервер. Выдаются: текущее время, сколько времени работает система,



сколько в текущий момент зарегистрировано пользователей и какова средняя нагрузка на систему в последние 1, 5 и 15 минут.

### **uptime**

#### **w – определяем, кто зарегистрирован и что они делают**

Команда w выдает информацию о том, какие пользователи сейчас находятся в системе и какие процессы запущены от их имени.

12. username

13. User

#### **mpart – использование процессами оперативной памяти**

Команда mpart выдает данные о распределении памяти между процессами. Использование этой команды позволит найти причину узких мест, связанных с использованием памяти.

mpart -d PID

Для того, чтобы получить информацию об использовании памяти процессом с pid # 26321, введите:

mpart -d 26321

#### **vmstat – активность системы, информация о системе и аппаратных ресурсах**

Команда vmstat выдает информационный отчет об активности процессов, памяти, свопинга, поблочного ввода/вывода, прерываний и процессора.

vmstat 3

#### **Выдача статистики использования памяти**

vmstat -m

#### **Получение данных об активности / неактивности страниц памяти**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**vmstat –a**



### **mpstat – использование мультипроцессора**

Команда mpstat выводит данные об активности каждого имеющегося в наличие процессора, процессор 0 будет первым. Команда mpstat -P ALL выводит данные о среднем использовании ресурсов для каждого из процессоров:

```
mpstat -P ALL
```

### **iostat – средняя загрузка процессора, активность дисков**

Команда iostat выдает статистику использования процессора, а также статистику ввода/вывода для устройств, разделов и сетевых файловых систем (NFS).

```
iostat
```

Задание:

6. Выполнить импорт конфигурации виртуальной машины – сервера UbuntuServer01 в Oracle VM VirtualBox
7. Пока идёт импорт (около 15 минут) изучить теоретический материал. Сомостоятельно выполнить поиск статей в Интернет, посвящённых Ubuntu Server.
8. Запустить виртуальную машину UbuntuServer01. Войти в терминал с логином admin1 и паролем 1.
9. Добавьте себя в список пользователей системы командой `sudo adduser <ваша фамилия>`, (латиницей) ,система потребует пароль – введите текущий – 1, затем введите свой новый пароль (при вводе символы не



отображаются!) и информацию о себе.

```
admin1@ubuntuL01:~$ sudo adduser tkachuk
Добавляется пользователь «tkachuk» ...
Добавляется новая группа «tkachuk» (1001) ...
Добавляется новый пользователь «tkachuk» (1001) в группу «tkachuk» ...
Создаётся домашний каталог «/home/tkachuk» ...
Копирование файлов из «/etc/skel» ...
Введите новый пароль UNIX:
Повторите ввод нового пароля UNIX:
passwd: password updated successfully
Changing the user information for tkachuk
Enter the new value, or press ENTER for the default
 Full Name []: Евгений Ткачук
 Room Number []: 205
 Work Phone []:
 Home Phone []:
 Other []:
chfn: name with non-ASCII characters: 'Евгений Ткачук'
Данная информация корректна? [Y/n] Y
admin1@ubuntuL01:~$ _
```

3. Для того, что вы могли выполнять команды администратора, введите себя в группу `sudo` командой `usermod -aG sudo <ваша фамилия>`

```
admin1@ubuntuL01:~$ sudo usermod -aG sudo tkachuk
admin1@ubuntuL01:~$
```

4. Сохраните скриншот выполненных команд и включите его в отчет. Для того, чтобы сделать скриншот, надо выйти из виртуальной машины нажатием правой кнопки Shift.
5. Выполните перезагрузку системы командой `sudo shutdown -r now`
6. Зайдите систему по своим именем, выполните все команды, описанные в теоретическом материале, скриншоты включите в отчёт.
7. В отчете в свободной форме опишите свои действия, сопроводив их скриншотами.



## 6. Лабораторная работа № 4 Управление пользователями в ОС UBUNTU

### Цель работы:

Изучить возможности Linux при работе с пользователями и управлении правами доступа.

### Задачи работы:

7. Рассмотреть концепцию Linux при работе с пользователями;
8. Изучить управление базами данных пользователей;
9. Рассмотреть возможности манипулирования доступом к данным.

### **8.1. Порядок выполнения работы**

13. Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу
14. Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01
15. Пока идёт импорт (около 15 минут) ознакомьтесь с теоретической частью
16. Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.
17. Выполните все действия, описанные в теоретической части (для тренировки).
18. Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.
19. Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.
20. Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

21. Экспортируйте образ сервера на свой носитель или в своё облако для последующей работы.



## 8.2. Теоретическая часть

### Работа с пользователями

Linux - это многопользовательская операционная система. Каждый пользователь в Linux принадлежит одной основной группе и одной или нескольким дополнительным группам. В Linux, как и в большинстве других операционных системах, работа с пользователями заключается в наборе следующих манипуляций: добавление пользователя/группы, удаление пользователя/группы, модификация настроек пользователя/группы. Данные манипуляции производятся с помощью команд: `useradd`, `groupadd`, `userdel`, `groupdel`, `usermod`, `groupmod`, а так же `passwd`, `gpasswd`, `id`. Существуют так же и графические средства администрирования пользователями, обычно они расположены в оболочке X в разделе Администрирование - Пользователи и группы. Однако, при администрировании Linux использование графических оболочек не приветствуется.

### UID, GID

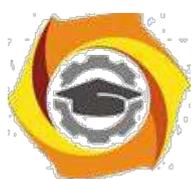
Каждый пользователь в системе имеет свой уникальный идентификационный номер (user-ID, или UID). Также пользователи могут объединяться в группы, которые в свою очередь имеют group-ID, или GID. Чтобы узнать свой UID и GID, т.е. уникальный номер пользователя и номер группы, к которой вы принадлежите, необходимо ввести команду `id` (Рис. 8.1).

```
work@work:~$ id
uid=1000(work) gid=1000(work) группы=4(adm)
,20(dialout),24(cdrom),46(plugdev),105(lpadmin)
,119(admin),122(sambashare),1000(work)
```

Рис. 8.1. Отобразить UID и GID

Пример добавления пользователя (Рис. 8.1.):

```
work@work:~$ sudo groupadd test
work@work:~$ sudo useradd -c "Test Test" -g test -m test
work@work:~$ sudo passwd test
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
work@work:~$ sudo it test
sudo: it: command not found
work@work:~$ sudo id test
uid=1001(test) gid=1002(test) groups=1002(test)
work@work:~$ sudo ls -ld /home/test/
drwxr-xr-x 2 test test 4096 2011-12-17 15:13 /home/test/
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

*Рис. 8.2. Добавление нового пользователя*



8. примере мы добавляем группу для нового пользователя (groupadd), далее создаем нового пользователя с полным именем Test Test, имеющего основную группу test и логин test, далее задаем пароль для пользователя test (passwd test)

7. проверяем параметры созданного пользователя (id и созданный каталог пользователя /home/test/). На Рис. 8.2 видно, что UID и GID - более 1000.

Данная особенность является признаком обычного пользователя. Значения ниже (меньше) 1000 (а в некоторых версиях - меньше 500) указывают на то, что пользователь является системным пользователем.

В соответствии с соглашением, системные пользователи обычно имеют id меньше, чем 100, а пользователь root имеет id, равный 0. Автоматическая нумерация обычных пользователей начинается со значения UID\_MIN, установленного в файле /etc/login.defs. Это значение обычно установлено в 500 или 1000.

Помимо учетных записей обычных пользователей и учетной записи пользователя root, в системе бывает несколько учетных записей специального назначения для демонов, таких как FTP, SSH, mail, news и т.д. Такие учетные записи часто управляют файлами, но к ним невозможно получить доступ путем обычной регистрации в системе. Поэтому обычно они имеют login shell, определенный как /sbin/nologin или /bin/false, чтобы попытки зарегистрироваться в системе терпели неудачу.

#### Управление базами данных пользователей и групп в Linux

Основные файлы, содержащие информацию о пользователях и группах, - это четыре файла в каталоге /etc.

9. /etc/passwd - файл паролей, содержащий основную информацию о пользователях;

10. /etc/shadow - файл теневых шифрованных паролей, содержащий зашифрованные пароли;

11. /etc/group - файл групп, содержащий основную информацию о группах и принадлежащих этим группам пользователях;



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

12./etc/gshadow - файл теневых групп, содержащий зашифрованные пароли групп.



Данные файлы редактировать обычным текстовым редактором крайне не рекомендуется. Они, обновляются при выполнении вышеуказанных команд, при этом при изменении - блокируются и синхронизируются.

Если все же есть острая необходимость в редактировании указанных файлов, то при помощи команды `vipw` можно безопасно редактировать файл `/etc/passwd`, а при помощи команды `viqr` безопасно редактировать файл `/etc/group`. Эти команды заблокируют необходимые файлы на то время, пока при помощи редактора `vi` будут производиться изменения. Если вы вносите изменения в файл `/etc/passwd`, команда `vipw` подскажет, что необходимо проверить, не нужно ли обновить и файл `/etc/shadow`. Подобным образом, если вы обновляете файл `/etc/group` при помощи команды `viqr`, вы получите подсказку, что необходимо обновить и файл `/etc/gshadow`. Если необходимо удалить администраторов группы, необходимо использовать команду `viqr`, поскольку команда `gpasswd` позволяет только добавлять администраторов.

12.        современных системах, файлы `passwd` и `group` не хранят пароли в открытом виде. Это сделано из соображений безопасности. Сами файлы `passwd` и `group`

должны быть доступными для чтения для всех, а зашифрованные пароли - недоступными для чтения для всех. Поэтому зашифрованные пароли хранятся в теневого файлах, и эти файлы доступны для чтения только пользователю `root`. Необходимый доступ для изменения аутентификационных данных обеспечивается при помощи `suid`-программы, которая имеет полномочия пользователя `root`, но может быть запущена любым пользователем.

### Права доступа

Операционная система Linux - это многопользовательская система, которая дает огромные возможности манипулирования доступом к данным для каждого пользователя отдельно. Это позволяет гибко регулировать отношения между пользователями, объединяя их в группы, что позволит защитить данные одного пользователя от нежелательного вмешательства других.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Бессмысленно считать, что файловая система это не самая важная часть операционной системы, поскольку все данные пользователей хранятся именно в файлах.



и UNIX-подобных системах файлы также обеспечивают доступ к периферийным устройствам, дисковым накопителям, принтерам и т.п.

### Права доступа к файлам

В свою очередь файлы имеют двух владельцев: пользователя (user owner) и группу пользователей (group owner). Для каждого файла есть индивидуальные права доступа, которые разбиты на три группы:

Доступ для пользователя-владельца файла (owner).

Доступ для группы-владельца файла (group).

Доступ для остальных пользователей (others).

Для каждой категории устанавливаются три вида доступа: (x) - право на запуск файла, (r) - право на чтение файла, (w) - право на изменение (редактирование) файла.

Для того, чтобы увидеть права доступа к файлам необходимо ввести команду ls с ключом -l (Рис. 8.3).

```
work@work:~$ ls -l Картинки/Kubuntu_leaflet.jpg
-rw-r--r-- 1 work work 658825 2010-03-26 15:21
Картинки/Kubuntu_leaflet.jpg
```

Рис. 8.3. Просмотр прав доступа к файлу

Для данного примера мы видим, что владелец имеет права на чтение, запись (первые две буквы rw), группа пользователей может лишь читать этот файл (следующая r--), остальные пользователи могут также только читать данный файл (r--).

### Изменение прав доступа

Права пользователя могут быть изменены только владельцем файла или пользователем с правами администратора системы. Для изменения прав используется команда:

```
chmod[u|g|o|a] [+|-|=] [r|w|x] name1 [name2 ...]
```

В качестве аргументов команда принимает указание классов доступа («u» - владелец-пользователь, «g» - владелец-группа, «o» - остальные пользователи, «a» - все вышеперечисленные группы вместе), права доступа («r» - чтение, «w»



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

из записи, «х» - выполнение) и операцию, которую необходимо произвести («+»  
и добавить, «-» - убрать, «=» - присвоить).



Таким образом, чтобы разрешить выполнение файла ip, который находится в директории /home/work/Загрузки всем пользователем необходимо выполнить команду (Рис. 8.4):

```
work@work:~$ chmod a+x Загрузки/ip
```

Рис. 8.4. Команда, выдающая права на исполнение файла

Далее, чтобы оставить права записи только для владельца файла необходимо выполнить (рис.5.5):

```
work@work:~$ chmod go-w Загрузки/ip
```

Рис. 8.5. Команда, позволяющая оставить права записи только для владельца файла

Рассмотрим еще несколько примеров:

- С `chmod go=w ip` - установить право на запись для всех пользователей кроме владельца;
- С `chmod a+x ip` - предоставить право на запись для всех пользователей;
- С `chmod g+x-w ip` - добавить для группы право на выполнения файла, но снять право на запись.

Права доступа можно представить в виде битовой строки, в которой каждые 3 бита определяют права доступа для соответствующей категории пользователей, как представлено в таблице 5.1:

Таблица 5.1

Представление прав доступа в виде битовой строки

|          |        |           |
|----------|--------|-----------|
| rwx      | rwx    | rwx       |
| 421      | 421    | 421       |
| user     | group  | others    |
| владелец | группа | остальные |

Таким образом, для команды `chmod 666 ip` имеем (Рис. 8.6):

```
work@work:~$ chmod 666 Загрузки/ip
work@work:~$ ls -l Загрузки/ip
-rw-rw-rw- 1 work work 226568 2010-01-18 11:11 Загрузки/ip
work@work:~$ chmod 644 Загрузки/ip
work@work:~$ ls -l Загрузки/ip
-rw-r--r-- 1 work work 226568 2010-01-18 11:11 Загрузки/ip
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

*Рис. 8.6. Пример использования команды `chmod`*

Команда:



`chmod 644 имя_файла`

устанавливает «обычные» права доступа, т.е. владелец может читать и записывать в файл, а все остальные пользователи - только читать.

#### Особенности прав доступа для каталогов

Права доступа для каталогов не столь очевидны. Это в первую очередь связано и тем, что система трактует операции чтения и записи для каталогов отлично от остальных файлов. Право чтения каталога позволяет Вам получить имена (и только имена) файлов, находящихся в данном каталоге. Чтобы получить дополнительную информацию о файлах каталога (например, подробный листинг команды `ls -l`), системы придется «заглянуть» в метаданные файлов, что требует права на выполнения для каталога. Право на выполнение также потребуется для каталога, в который Вы захотите перейти (т.е. сделать его текущим) с помощью команды `cd`.

#### T-бит, SUID и SGID

Помимо стандартных «`gwx`» значений существуют еще и буквы «`s`» и «`t`». В действительности, битовая маска прав доступа к файлам содержит 4 группы по 3 бита в каждой. Таким образом, команда `chmod 755` это всего лишь краткая запись полной формы команды: `chmod 0755`.

T-бит обычно используется с каталогами. Обычно, когда t-бит для каталога не установлен, файл в данном каталоге может удалить любой пользователь, имеющий доступ на запись к данному файлу. Устанавливая t-бит на каталог мы меняем это правило таким образом, что удалить файл из каталога может только владелец этого каталога или файла.

Установить t-бит можно при помощи команд:

`chmod a+tw имя_файла`

`chmod 1777 имя_файла`

Атрибуты SUID и SGID позволяют изменить права пользователя при запуске на выполнения файла, имеющего эти атрибуты.

Запускаемая программа получает права доступа к системным ресурсам на основе прав доступа пользователя, запустившего программу. Установка же флагов



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

SUID и SGID изменяет это правило таким образом, что назначает права доступа к системным ресурсам исходя из прав доступа владельца файла. Т.е.



запущенный исполняемый файл, которым владеет суперпользователь, получает права доступа к системным ресурсам на уровне суперпользователя (фактически неограниченные). При этом установка SUID приведет к наследованию прав владельца-пользователя файла, а установка SGID - владельца-группы.

Пользоваться такими мощными атрибутами как SUID и SGID нужно с крайней осторожностью, особенно подвергать пристальному вниманию программы и скрипты, владельцем которых является root (суперпользователь), т.к. это потенциальная угроза безопасности системы.

### Управление файлами

В ОС Linux следует различать физическую файловую систему, которая отвечает за управление дисковым пространством и размещение файлов в физических адресах диска и логическую файловую систему, которая обеспечивает логическую структуру хранения файлов - пространство имен файлов. ОС Unix и Linux могут работать с различными физическими файловыми системами (Ext2, ext3, ufs), логическое же представление файловой системы в Unix/Linux структурировано. Все файлы в логической файловой системе располагаются в виде дерева, промежуточные вершины которого соответствуют каталогам, и листья - файлам и пустым каталогам. Реально на каждом логическом диске (разделе физического дискового пакета)

располагается отдельная иерархия каталогов и файлов. Для получения общего дерева в динамике используется «монтирование» отдельных иерархий к фиксированной корневой файловой системе в качестве ветвей общего дерева. Самым верхом иерархии является корень, который имеет предопределенное имя «/» (слэш). Этот же символ используется как разделитель имен в пути. Далее в корне находятся папки с определенными для каждого дистрибутива именами (etc, home, bin, mnt, proc и т.д.).

Полное имя файла, например, /bin/sh означает, что в корневом каталоге должно содержаться имя каталога bin, а в каталоге bin должно содержаться имя файла sh. Коротким или относительным именем файла называется имя, задающее путь к файлу от текущего рабочего каталога. В каждом каталоге содержатся два



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

специальных имени, имя «.» - ссылка на текущий каталог, и имя «...» - ссылка  
«родительский» каталог данного текущего каталога, т.е. каталог,



непосредственно предшествующий данному в иерархии каталогов. Так, например, для структуры, показанной на Рис. 8.7 доступ к файлу file2 из текущего каталога (laba) возможен по полному имени: /home/myvar/file2 или по относительному имени: ../../../../myvar/file5.



*Рис. 8.7. Пример дерева каталогов*

### Типы файлов

ОС LINUX поддерживают несколько типов файлов:

- Обычные файлы (или регулярные) - представляют собой последовательность байтов. Это текстовые, исполняемые файлы и т.д. Данный тип файла отображается командой `ls -l` в виде «-» (черточка).
- у Каталоги - представляют собой особый вид файлов, которые хранятся во внешней памяти подобно обычным файлам, но их структура поддерживается самой файловой системой. Данный тип файла отображается командой `ls -l` в виде символа «d».
- у Специальные файлы устройств, бывают блочные и символьные. Данный тип файла отображается командой `ls -l` в виде символа «b» или «c» соответственно. Специальные файлы не хранят данные. Они обеспечивают механизм отображения физических внешних устройств в имена файлов файловой системы. Каждому устройству, поддерживаемому системой, соответствует, по меньшей мере, один специальный файл. При выполнении чтения или записи по отношению к специальному файлу, производится прямой вызов соответствующего драйвера устройства. При этом имена



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

специальных файлов можно использовать практически всюду, где можно использовать имена обычных файлов.



- и Ссылка (link). Данный тип файла отображается командой `ls -l` в виде символа «l». Файловая система UNIX/LINUX обеспечивает возможность связывания одного и того же файла с разными именами.
- и Именованный программный канал (pipe) - одно из средств межпроцессных взаимодействий (IPC) в ОС UNIX/LINUX. Данный тип файла отображается командой `ls -l` в виде символа «p». Именованному программному каналу обязательно соответствует элемент некоторого каталога.
- и Сокет (socket)- предоставляют весьма мощный и гибкий IPC. Данный тип файла отображается командой `ls -l` в виде символа «s». Они могут использоваться для организации взаимодействия программ на одном компьютере, по локальной сети или через Internet, что позволяет создавать распределённые приложения различной сложности. Кроме того, с их помощью можно организовать взаимодействие с программами, работающими под управлением других операционных систем.

### Ссылки

Существуют жесткие и мягкие ссылки.

Жесткая ссылка является просто еще одним именем для исходного файла и не является типом файла. Она прописывается в индексном дескрипторе исходного файла (в структуре, хранящей метаданные файла). После создания жесткой ссылки невозможно различить, где исходное имя файла, а где ссылка. Если вы удаляете один из этих файлов (точнее одно из этих имен), то файл еще сохраняется на диске (пока у него есть хоть одно имя - жесткая ссылка). Очень трудно различить первоначальное имя файла и позже созданные жесткие ссылки на него. Поэтому жесткие ссылки применяются там, где отслеживать различия и не требуется. Одно из применений жестких ссылок состоит в том, чтобы предотвратить возможность случайного удаления файла. Особенностью жестких ссылок является то, что они прямо указывают на номер индексного дескриптора, а, следовательно, такие имена могут указывать только на файлы внутри той же самой файловой системы (т. е., на том же самом носителе, на котором находится каталог, содержащий это имя).



Мягкие (символические) ссылки тоже могут рассматриваться как дополнительные имена файлов, но в то же время они представляются отдельными файлами - файлами типа мягких ссылок и являются самостоятельным типом файла. Однако блоки данных файла в системе представляются в одном экземпляре, у файла-ссылки адреса блоков данных те же, что и у исходного файла. В отличие от жестких ссылок мягкие ссылки могут указывать на файлы, расположенные в другой файловой системе, например, на монтируемом носителе, или даже на другом компьютере. Если исходный файл удален, мягкая ссылка не удаляется, но становится бесполезной. Используйте мягкие ссылки в тех случаях, когда хотите избежать путаницы, связанной с применением жестких ссылок.

Создание любой ссылки внешне подобно копированию файла, но фактически как исходное имя файла, так и ссылка указывают на один и тот же реальный файл на диске. Поэтому, например, если вы внесли изменения в файл, обратившись к нему под одним именем, вы обнаружите эти изменения и тогда, когда обратитесь к файлу по имени-ссылке.

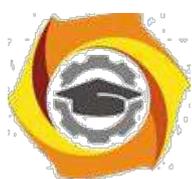
Для создания ссылки, используется команда `ln` (Рис. 8.8):

`ln [-f] файл1 [файл2 ...] целевой_файл`

Команда `ln` делает целевой\_файл ссылкой на файл1. Файл1 не должен совпадать с целевым\_файлом. Если целевой\_файл является каталогом, то в нем создаются ссылки на файл1, файл2,... с теми же именами. Только в этом случае можно указывать несколько исходных файлов. Если целевой\_файл существует

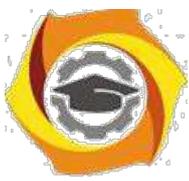
В не является каталогом, его старое содержимое теряется. Аргументы: `-f` - удаление существующего целевого файла; `-s` - создание мягкой ссылки (по умолчанию создается жесткая ссылка).

```
work@work:~$ ls
examples.desktop site1 Музыка
mysite Видео Общедоступные
Navicat Документы Рабочий стол
PHP редактор Загрузки Шаблоны
shares Картинки
work@work:~$ ln -s Картинки/Kubuntu_leaflet.jpg
work@work:~$ ls
examples.desktop shares Картинки
Kubuntu_leaflet.jpg site1 Музыка
mysite Видео Общедоступные
Navicat Документы Рабочий стол
PHP редактор Загрузки Шаблоны
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

*Рис. 8.8. Пример создания ссылки*



### Файловый менеджер

Управление файлами также можно выполнять с помощью Midnight Commander (mc) - один из файловых менеджеров с текстовым интерфейсом типа Norton Commander для UNIX-подобных операционных систем. Запуск mc из консоли выполняется с помощью команды mc.

Установить файловый менеджер mc так, как показано на Рис. 8.9, делать этого не надо, он уже установлен в виртуальной машине Ubuntu Server01.

```
work@work:~$ sudo apt-get install mc
```

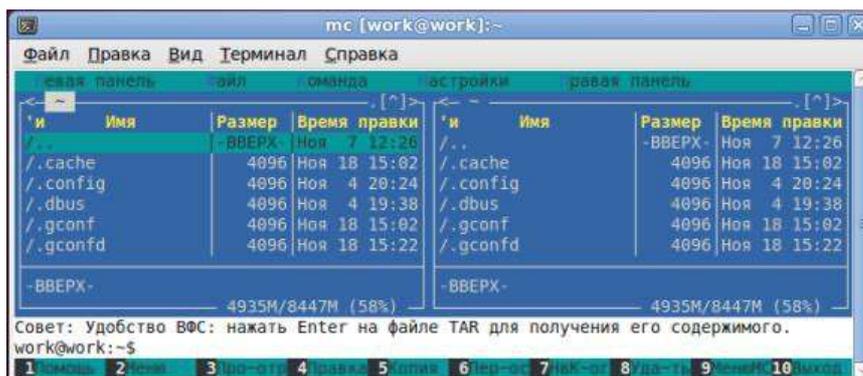
Рис. 8.9. Установка mc

Достоинство mc том, что есть встроенные средства редактирования и просмотра текстовых файлов (какими являются конфигурационные файлы). При работе с mc необходимы права [суперпользователя](#). Общая последовательность действий по редактированию конфигурационного файла:

- С запустить программу (ввод команды mc);
- С используя клавиши управления курсором и клавишу «Enter», добраться до нужного файла и выбрать его;
- С нажатием клавиши «F4» открыть файл для редактирования;
- С внести необходимые изменения;
- С сохранить их (клавиша «F2»);
- С выйти из режима редактирования (клавиша «F10»).

Просмотр файла выполняется аналогично, только клавишей «F3».

Выйти из mc можно тоже клавишей «F10». Общий вид mc приведен на Рис. 8.10.





**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

*Рис. 8.10. Midnight Commander*



### **8.3. Задания к лабораторной работе:**

- и Выведите на экран UID и GID своего пользователя;
- и Создайте группу пользователей с именем usersGroup;
- и Создайте пользователя myUser в группе usersGroup;
- и Выведите на экран UID и GID пользователя myUser;
- и В своем пользователе work создайте текстовый файл, и ограничьте к нему доступ на чтение для всех других групп пользователей. Затем, зайдите в систему от имени пользователя myUser и проверьте возможность открыть этот текстовый файл;
- и От имени пользователя work изменить права доступа к данному файлу, и проверьте изменения;
- и Для одного и того же файла создайте мягкую и жесткую ссылку в домашнем каталоге. Попробуйте создать ссылки одновременно для нескольких файлов;
- и Проверьте работу файлового менеджера mc.

### **8.4. Контрольные вопросы**

- У Расскажите про идентификационные номера пользователей и групп в Linux.
- У Расскажите о файлах Linux, содержащих информацию о пользователях и группах системы.
- У Как система Linux хранит пароли пользователей и групп?
- У Как организуется разграничение доступа к файлам в Linux?
- У Как изменить права доступа к файлу? Как сделать это с помощью битовой строки?
- У Расскажите о T-бит, SUID и SGID.
- У Расскажите про файловые системы Linux.
- У Какие типы файлов существуют в Linux?
- У Расскажите о мягких и жестких ссылках.



## И Практическое занятие 5. Работа с пакетами и репозиториями Ubuntu

### Цель работы:

Изучить возможности Linux при работе с пользователями и управлении правами доступа.

### Задачи работы:

- К Рассмотреть концепцию Linux при работе с пользователями;
- К Изучить управление базами данных пользователей;
- К Рассмотреть возможности манипулирования доступом к данным.

### **9.1. Порядок выполнения работы**

1. Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу
2. Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01
3. Пока идёт импорт (около 15 минут) ознакомьтесь с теоретической частью
4. Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.
5. Выполните все действия, описанные в теоретической части (для тренировки).
6. Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.
7. Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.
8. Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

3. Экспортируйте образ сервера на свой носитель или в своё облако для последующей работы.



## 9.2. Теоретическая часть

Первое, с чем сталкивается пользователь, впервые опробовав линуксную ОС после Windows, - необычность понятия "пакет". Однако ничего заумного в нём нет. Пакетная система работы с линуксным софтом порождена философией UNIX "делай только одну вещь". Грубо говоря, пакет - это архив, который содержит версию формата пакета, сжатую заархивированную управляющую информацию и сжатые заархивированные устанавливаемые файлы. Короче, архив в архиве. При этом пакет содержит только оригинальные файлы, а всякого рода часто встречающиеся общеупотребительные файлы включать в пакет не принято.

Управляющая информация может включать в себя ссылки на другие пакеты, необходимые для работы, т.е. оговаривать так называемые "зависимости". Пример пакета: `opera_11.01.1190_i386.deb`. Здесь `opera` - наименование пакета, `11.01.1190` - его версия, `i386` - признак, что пакет предназначен для компьютера с чипсетом архитектуры Intel не ниже 386, `deb` указывает, что пакет предназначен для ОС, совместимых с Debian GNU/Linux.

Нередко интересуются, будет ли пакет, помеченный как `i586` или `i686`, работать на конкретном интеловском компьютере. Узнайте класс своего процессора командой

```
cat /proc/cpuinfo
```

4. обратите внимание на значение параметра `cpu family`. Это число должно быть больше или равно первой цифре, следующей за символом `i` в обозначении пакета. С процессорами AMD проще: пакеты помечаются как `AMD64` или `AMD32`, в зависимости от разрядности процессора. И ещё: не устанавливайте 64-разрядные пакеты, если ваша ОС 32-разрядная: могут быть проблемы. Впрочем, при установке пакетов из репозитория этого не должно случиться. Вот, собственно, и всё, что на первых порах надо знать о понятии "пакет".



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

ОС Ubuntu, потомок ОС Debian, обеспечивает пользователю несколько инструментов для управления пакетами. Основу составляют утилита apt (акроним от "Advanced package tool") и низкоуровневая утилита dpkg (акроним



от "Debian package tool"). Кроме того, Ubuntu предлагает и другие интерфейсы а системе пакетов Debian GNU/Linux. К ним относятся в порядке возрастания уровня: псевдографический aptitude, графический Synaptic и, наконец, "Центр Приложений Ubuntu", который, в отличие от остальных, работает только с приложениями. Не буду рассматривать графические интерфейсы, их освоить легко, но возможности "не те". Также оставляю в покое aptitude как производное от apt. Порою встречающиеся утверждения, что aptitude лучше удовлетворяет зависимости, чем apt, считаю просто недоразумениями. Кроме того, начиная с версии Ubuntu 10.10, aptitude удалён из дистрибутива. По моему убеждению, чтобы быть уверенным пользователем Ubuntu, осваивать следует apt и dpkg. Репозиторий в Ubuntu — это архив программ, расположенный в интернете. Удобство установки программы из репозитория заключается в том, что пользователю не нужно беспокоиться о совместимости и безопасности устанавливаемого пакета.

### **apt**

Начиная с версии Ubuntu 16.04 доступна новая утилита apt, которая содержит наиболее часто используемые команды из apt-get и apt-cache. Утилита apt предназначена для пользователей, в то время как apt-get можно рассматривать как как инструмент более низкого уровня.

### **apt-get**

apt-get это утилита управления пакетами. apt-get требует прав суперпользователя для своей работы.

Основные команды:

4. update - Обновить информацию о пакетах, содержащихся в репозиториях.
5. install foo - Установить пакет foo. Скачивание, установка и настройка происходят автоматически. Если для настройки пакета foo нужны дополнительные сведения, будет показан запрос к пользователю.
6. upgrade - Обновление пакетов, для которых в репозитории доступны новые версии.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

7. `dist-upgrade` - Обновление пакетов, требующих разрешения зависимостей (установка дополнительных или удаление конфликтующих пакетов).



17. `remove foo` - Удаление пакета `foo` из системы.
18. `purge foo` - Удаление пакета `foo` и очистка системы от его конфигурационных файлов. Файлы настроек в домашних каталогах пользователей удалены не будут.
19. `autoremove` - Удаление пакета, который более не нужен в системе. Используется для очистки системы от ненужных пакетов. Факт, что пакет более не нужен, определяется следующим образом: если пакет был установлен не сам по себе, а как зависимость для другого пакета, который впоследствии был удалён, значит этот пакет тоже уже не нужен в системе.
20. `source foo` - Получение исходных текстов пакета `foo`.
21. `build-dep foo` - Получение зависимостей для сборки пакета `foo`.

#### ***Починка базы пакетов***

22. `apt-get` также используется для устранения сбоев в базе пакетов вызванных нарушенными зависимостями. Разрешение зависимостей производится командой:
23. `sudo apt-get install -f`

#### **apt-cache**

`apt-cache` – утилита, позволяющая выполнять запросы к кешу

АРТ. Основные команды:

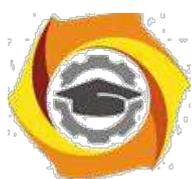
24. `search` - Поиск пакета по части названия или описания. Поддерживает регулярные выражения.
25. `show` - Информация о пакете: версия, размер, описание и т. п.
26. `depends` - Зависимости указанного пакета.
27. `rdepends` - Обратные зависимости пакета.

#### **apt-key**

`apt-key` служит для добавления ключей от репозитория в систему. Ключи защищают репозитории от возможности подделки пакета. Подробнее смотрите в статье Репозитории.

Основные команды:

- `add` - Добавление ключа в базу доверенных ключей.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

— del - Удаление ключа из доверенных.



Возможности этого семейства утилит огромны. Основная команда семейства - это `apt-get install _наименование_пакета_`. Можно устанавливать и сразу несколько пакетов, перечислив их наименования через пробелы. Тут важно понимать, что указываются только имена, ОС вытягивает из репозитория именно то, что нужно для вашего компьютера и установленной сборки Ubuntu. При установке нового пакета инсталлятор проверяет, были ли ранее установлены пакеты, от которых зависит новый пакет, если нет, то пытается взять такие пакеты из доступных репозиториях. При недоступности таковых выдаёт ошибку. Но иногда следует указывать имена расширенные, которые включают в себя мажор версии пакета. Так поступают, если в репозитории находятся несколько версий и пользователю предоставляется возможность выбирать. Например, команда

```
sudo apt-get install virtualbox-4.0
```

сначала определит, не установлена ли другая версия `virtualbox`, если да, то с санкции пользователя "снесёт" её и установит именно `virtualbox-4.0`. А вот команда

```
sudo apt-get install virtualbox
```

просто попытается установить из репозитория самую свежую версию этой замечательной виртуальной машины. При установке пакеты сначала попадают – кэш `/var/cache/apt/archives`, затем устанавливаются куда надо. Если очень интересно, куда, то это всегда можно узнать командой `whereis`, например, `whereis virtualbox`. По мере взросления вашей сборки Ubuntu в кэше накапливается много нужного и ненужного. Последнее занимает место и может быть удалено, о чём чуть позже.

Нередко требуется удалить установленный пакет - ну, погорячились, установили не то, что надо, или же новый пакет в управляющей информации не содержит сценария удаления своего предшественника. Команда



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

```
sudo apt-get remove virtualbox
```



"снесёт" установленную версию пакета `virtualbox`, однако сохранит все конфигурационные файлы с тем, чтобы ими могла воспользоваться другая версия пакета. При этом сам пакет останется в кэше, и при необходимости его можно будет установить оттуда повторно. А вот команда

```
sudo apt-get --purge remove virtualbox
```

"снесёт" и конфигурационные файлы, и сам пакет в кэше.

Многие пакеты ОС Ubuntu обновляет в порядке штатной процедуры обновления, но только те, которые поддерживает компания Canonical, а также те, для которых обновления предоставляются сообществом Ubuntu. Остальные пакеты в пределах мажора версии можно обновлять самостоятельно. Например, команда

```
sudo apt-get upgrade opera -u
```

обновит ваш ранее установленный браузер Opera, повысив минор версии, если, конечно, таковая существует в репозитории, прописанном в `/etc/apt/sources.list`. При обновлении приложения строится дерево зависимостей, проверяются обновления для них и нередко обновляется не основной пакет, а только пакеты, от которых он зависит. Необязательная опция `-u` позволяет видеть, что именно обновляется. Кстати, теперь для добавления репозитория нет необходимости редактировать файл `/etc/apt/sources.list`, достаточно воспользоваться командой `add-apt-repository _источник_`. Например, можно добавить источник deb-пакетов от Google вот такой командой:

```
sudo add-apt-repository "deb http://dl.google.com/linux/deb/ stable main"
```

После добавления нового источника пакетов никогда не забывайте проиндексировать список источников командой

```
sudo apt-get update
```

Если вы не знаете или забыли, какая версия пакета установлена и какие библиотеки он использует, то не стесняйтесь задать системе вопрос, например,



командой `apt-cache search _имя_пакета_`, при этом мажор версии пакета, естественно, не указывается, например,

```
apt-cache search sysstat
```

т ответ вы получите исчерпывающую информацию о пакетах, непосредственно касающихся вашего приложения, в данном случае утилиты сбора статистики о работе вычислительной системы *sysstat*. Теперь, зная версию пакета, вы можете определить все зависимости, которые он за собой тянет, командой

```
sudo apt-cache depends sysstat -4.0
```

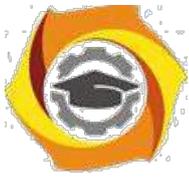
Это очень полезная команда, так как с её помощью можно получить рекомендации, какие пакеты следует доустановить для более полноценной работы приложения. Дело в том, что иногда не все зависимости указываются в основном пакете приложения; такие непрописанные зависимости называются вынесенными. Их можно, конечно, и проигнорировать, приложение будет работать, однако функциональность его останется неполной.

Чтобы узнать, какие приложения установлены, необходимо воспользоваться командой

```
sudo apt-cache search all
```

При этом на экран будет выведен только список установленных приложений с краткими комментариями о них; зависимости, библиотеки и прочее выведены не будут. Иногда бывает нужным знать, какие ветви и файлы создала или затронула установка пакета. Для этой цели служит команда `apt-file search _строка_`, где `_строка_` - это набор символов, последовательность которых встречается в полном пути. Правда, `apt-file` по умолчанию отсутствует, и эту утилиту сначала надо установить командой

```
sudo apt-get install apt-file
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Имея в распоряжении apt-file, нетрудно найти директории, затронутые установкой конкретного приложения, например, virtualbox-4.0:



```
sudo apt-file search virtualbox-4.0
```

Для очистки кэша от устаревших пакетов, которые больше не скачиваются и не нужны, имеется очень полезная команда

```
sudo apt-get autoclean
```

Чтобы очистить весь кэш от пакетов, если срочно требуется место на диске, на работоспособности системы это как будто не сказывается, но если вам потребуется переустановить какой-либо ранее установленный пакет, то придётся его заново скачивать. Делается такая чистка следующей командой, хотя я бы подобное не рекомендовал:

```
sudo apt-get clean
```

Если надо убедиться, всё ли в порядке с пакетами, установленными в системе, то это проверяют командой

```
sudo apt-get check
```

Если не получите никаких сообщений о неудовлетворённых зависимостях, то всё отлично. Правда, вынесенных зависимостей это не касается. Чтобы уладить проблемы с обнаруженными невынесенными зависимостями, просто введите

```
sudo apt-get -f install
```

Я перечислил apt-минимум, которым должен владеть пользователь Ubuntu, желающий свободно управлять пакетным хозяйством своей системы, пополняемым из репозиториев. Однако, помимо пакетов из репозиториев, имеется огромное количество и других deb-пакетов, которые разработчики предлагают на своих сайтах. Чтобы воспользоваться этим богатством, нужно спуститься уровнем ниже.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

**dpkg**



Полную информацию об этой утилите можно почерпнуть в [debianadmin.com/debianubuntu-package-management-using-dpkg.html](http://debianadmin.com/debianubuntu-package-management-using-dpkg.html). Я не рекомендую новичкам пользоваться этой утилитой для установки-удаления пакетов. `dpkg` не разрешает зависимости: если для устанавливаемого пакета требуются другие пакеты, то их нередко приходится устанавливать вручную. Кроме того, всякий раз при установке пакета требуется указывать путь к нему:

```
sudo dpkg -i /путь/пакет.deb
```

и выявлять зависимости командой:

```
dpkg --info /путь/пакет.deb
```

Неудобно! Проще скачанный пакет переместить из директория загрузок в вышеупомянутый кэш и установить его оттуда утилитой `apt-get install`, не указывая путь. Тем не менее, `dpkg` - весьма достойная утилита. Вот как без `aptitude` проще всего узнать, какие пакеты установлены в системе? Нет проблем:

```
dpkg -l
```

— помощью `dpkg` можно легко узнать, какая именно версия пакета установлена в системе и установлен ли пакет вообще; например, команда

```
dpkg -l 'virtualbox*'
```

выводит установленную версию виртуальной машины `virtualbox`, предваряя её суффиксом `ii`, но также выводит и старую версию пакета, если он не удалялся из кэша, предваряя его имя суффиксом `rc`. Более подробную информацию о версиях пакетов, по сравнению с предыдущей командой, можно получить, привлекая `grep`:

```
dpkg -l | grep 'virtualbox*'
```

Чтобы определить ширину колонок при выводе информации о пакетах, укажите её явно, например:



```
COLUMNS-150 dpkg -l 'virtualbox*'
```

Утилита dpkg также при желании позволяет посмотреть содержимое пакета:

```
dpkg -content /путь/пакет.deb
```

### О репозиториях

Владельцы репозитория скомпилировали исходный код, оформили его в виде пакета для установки и протестировали программу на совместимость с операционной системой. Для программного обеспечения из репозитория имеется определенная гарантия, что программа А, версии В, заработает в дистрибутиве версии С. Кроме этого, получение программы из репозитория гарантирует, что в программу не внесены несанкционированные изменения третьей стороной.

### Виды репозиторияев

В Ubuntu имеется штатных 4 репозитория:

- В **Main** — это основной репозиторий, содержащий официально поддерживаемые приложения.
- В **Restricted** — здесь содержится официально поддерживаемый софт, но который не лицензирован на условиях открытой лицензии GPL.
- В **Universe** — для программ, не имеющих официальной поддержки. Этот репозиторий поддерживается силами сообщества.
- В **Multiverse** — содержит ПО, которое не является свободным.

### Добавление и удаление репозиторияев

Кроме этих четырех репозиторияев существует огромное количество сторонних, которые еще иногда называют репозиториями третьей стороны. Любой желающий может создать свой частный репозиторий, который потом можно будет подключить к Ubuntu.

По умолчанию, репозитории Universe и Multiverse доступны, но если вы хотите отключить их, измените `/etc/apt/sources.list` и прокомментируйте следующие строки:



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

*deb <http://archive.ubuntu.com/ubuntu> precise universe multiverse*



```
deb-src http://archive.ubuntu.com/ubuntu precise universe multiverse
deb http://us.archive.ubuntu.com/ubuntu/ precise universe deb-src
http://us.archive.ubuntu.com/ubuntu/ precise universe
deb http://us.archive.ubuntu.com/ubuntu/ precise-updates universe
deb-src http://us.archive.ubuntu.com/ubuntu/ precise-updates
universe deb http://us.archive.ubuntu.com/ubuntu/ precise multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ precise multiverse
deb http://us.archive.ubuntu.com/ubuntu/ precise-updates multiverse deb-
src http://us.archive.ubuntu.com/ubuntu/ precise-updates multiverse deb
http://security.ubuntu.com/ubuntu precise-security universe deb-src
http://security.ubuntu.com/ubuntu precise-security universe deb
http://security.ubuntu.com/ubuntu precise-security multiverse deb-src
http://security.ubuntu.com/ubuntu precise-security multiverse
```

Добавить новый источник пакетов очень просто. Делается это всего одной командой:

```
sudo apt-add-repository ppa:user/repository
```

Здесь `ppa:user/repository` — имя зарегистрированного репозитория. Удаляется источник пакетов аналогично, с помощью программы `apt-add-repository`, но с указанием ключа `-r`:

```
sudo apt-add-repository -r ppa:user/repository
```

Следует отметить, что просто подключить новый репозиторий недостаточно, нужно попросить операционную систему загрузить из него список доступных программ актуальных версий:

```
sudo apt-get update
```

После проделанных манипуляций можно приступать к установке, которая осуществляется самым привычным образом:

```
sudo apt-get install имя-программы
```

### 9.3. Задания к лабораторной работе:



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

**В** Обновите индекс пакетов вашего виртуального сервера (...update).



- и Обновите вашу систему (...upgrade).
- и Установите утилиту измерения производительности вашей ОС sysstat.
- и Проверьте её версию.
- и Запустите утилиту командами iostat , затем mpstat.
- и Самостоятельно найдите описание и изучите команды sysstat, проверьте их.

#### **9.4. Контрольные вопросы**

- 3 Расскажите про пакеты ubuntu.
- 4 Расскажите о репозиториях ubuntu.
- 5 Как добавить репозиторий к стандартному списку Ubuntu?
- 6 Как обновит индекс пакетов Ubuntu?
- 7 Как обновить все пакеты в Ubuntu?
- 8 Как обновить только одну программу в Ubuntu?
- 9 Как установить новую утилиту в Ubuntu?
- 10 Как удалить утилиту из Ubuntu?
- 11 Как полностью удалить утилиту Ubuntu?
- 12 Как очистить кэш пакетов Ubuntu?
- 13 Как система Linux хранит пароли пользователей и групп?
- 14 Для чего служит утилита sysstat?
- 15 Перечислите команды утилиты sysstat.
- 16 Как запустить демон сбора статистик и sysstat?



## 11. Лабораторная работа №5 Управление каталогами.

### Цель работы:

Изучение структуры файловой системы и возможностей командного языка UBUNTU по управлению каталогами.

### 10.1. Порядок выполнения работы

- 21 Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу
- 22 Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01
- 23 Пока идёт импорт (около 15 минут) ознакомьтесь с теоретической частью
- 24 Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.
- 25 Выполните все действия, описанные в теоретической части (для тренировки).
- 26 Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.
- 27 Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.
- 28 Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.
- 29 Экпортируйте образ сервера на свой носитель или в своё облако для последующей работы.



## 10.2. Теоретическая часть

### *Изучаются команды:*

`mkdir, rmdir` - для создания и уничтожения каталогов;

`ls` - вывод листинга каталога. «что здесь есть?»;

`pwd` - вывод на экран полного имени текущего каталога, «где я нахожусь?»; `cd` - смена текущего каталога, «перейти в»; `find, grep` - поиск файлов в системе каталогов;

>маршрутное – имя - файла - создание пустого файла.

Эта работа посвящена изучению структуры файловой системы и возможностей командного языка Ubuntu по управлению каталогами. Для начала рассмотрим основные команды, используемые в ОС UBUNTU при работе с каталогами..

Разделителем элементов пути в системах UBUNTU служит символ `V`. В отличие от таких систем, как, например, MSDOS и Windows, в которых каждому тому соответствует отдельный корневой каталог, обозначаемый именем тома, в системе UBUNTU есть только один корневой каталог. Он обозначается `‘/’` - Все дополнительные тома, подключаются к основному дереву каталогов так, что корневой каталог каждого из этих томов становится просто одним из каталогов в файловой системе. Каталог для подключения может быть выбран произвольно. Операция подключения тома к файловой системе носит название монтирования, и может производиться в любой момент во время работы системы.

В системе обычно присутствуют следующие каталоги:

- — корневой каталог;

`/bin`— каталог с пользовательскими программами;

`/sbin` — каталог программ для администрирования системы;

`/etc` — каталог с конфигурационными файлами программ;

`/home` — каталог, в котором создаются домашние каталоги пользователей;

`/lib` — каталог с динамическими и статическими библиотеками; `/boot` — каталог, содержащий файлы системного загрузчика;

`/mnt` — каталог, в который, как правило, производится монтирование;

`/dev` — каталог, содержащий специальные файлы устройств;



/opt — каталог, в который устанавливается ПО сторонних производителей;

/usr — каталог, в котором хранятся в режиме доступа только для чтения разделяемые данные, такие как исполняемые файлы программ, документация, библиотеки и другие системные ресурсы;

/root — каталог, являющийся домашним для пользователя root;

/var — каталог, содержащий журналы, файлы баз данных, кеши разного рода; .

/tmp — каталог для хранения временных файлов.

*Для печати текущего каталога:*

```
pwd
```

*Для смены каталога:*

```
cd <путь к каталогу>
```

Пример: переход к каталогу var.

```
rogdy@ubuntu:~$ cd /var
rogdy@ubuntu:/var$ pwd
/var
rogdy@ubuntu:/var$ ls
backups cache crash games lib local lock log mail opt run spool tmp
```

Попробуйте объяснить смысл каждой строки из данного примера. Какие операции запрашивает пользователь, и какие ответы даёт система?

Если команда запущена без указания каталога, то переход производится в домашний каталог пользователя. Вообще, для указания домашнего каталога пользователя можно использовать специальный символ '~'. Так, для перехода в папку tmp, находящуюся в домашнем каталоге можно воспользоваться следующей командой:

```
rogdy@ubuntu:/var$ cd ~ /tmp
```

Домашний каталог пользователя обычно располагается в каталоге /home и называется по имени пользователя. Например, для пользователя user1 домашний каталог будет таким: /home/user1.

*Для создания каталога:*



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

`mkdir <список имен каталогов>`

```
rogdy@ubuntu:~$ mkdir KaTaLoG
```



Если требуется создать сразу несколько вложенных друг в друга каталогов,

можно воспользоваться ключом -р:

```
rogdy@ubuntu:~$ mkdir -p vsheshny/vnutrenny
```

Для удаления каталога:

```
rogdy@ubuntu:~$ rmdir KaTaLoG
```

---

Пример: создание каталогов и работа с ними.

```
rogdy@ubuntu:~$ mkdir abc
rogdy@ubuntu:~$ cd abc
rogdy@ubuntu:~/abc$ mkdir ABC
rogdy@ubuntu:~/abc$ ls
ABC
```

---

Попробуйте объяснить смысл каждой строки из данного примера. Какие операции запрашивает пользователь, и какие ответы даёт система?

Команда удаляет только пустые каталоги. Ключ -р подобен такому же ключу команды `mkdir`, и позволяет удалить сразу несколько каталогов, вложенных друг в друга, если все они пусты.

`/` Корневой каталог. Это родительский каталог для всех каталогов и файлов в файловой системе UBUNTU.

`/bin` Каталог исполняемых модулей командной строки. Данный каталог содержит все исполняемые модули «родных» команд UBUNTU.

`/dev` Каталог устройств, содержащий специальные файлы для байт-ориентированных и блок-ориентированных устройств, таких как принтеры и клавиатуры. В данном каталоге существует файл под названием `null`, который называется `bit bucket` и который может использоваться для перенаправления вывода в никуда.

`/etc` Файлы системной конфигурации и каталог исполняемых файлов. Большая часть административных файлов, а также файлов, связанных с командами, хранится здесь.

`/lib` В каталоге хранятся библиотеки компилятора C.

`/lost+found` Данный каталог содержит обрабатываемые файлы, если система отключилась ненормально. Система использует данный каталог для



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

восстановления файлов. В каждом разделе диска есть только один каталог  
lost+found.



/usr Данный каталог имеет несколько подкаталогов, таких как adm, bin, etc и include. Например, /usr/include содержит файлы заголовков для компилятора C. /home содержит домашние каталоги пользователей.

Для создания каталогов используется команда mkdir. Можно указывать как полный так и относительный путь. Поэтому можно создавать дерево каталогов: определить относительно или абсолютно корень, после чего создать относительно нового каталога новые поддирективы.

Команду ls (с ее многочисленными опциями) можно использовать для получения информации об одном или нескольких файлах или каталогах системы. Используйте ls для генерации списка файлов и каталогов в различном порядке, например по имени или по времени. Возможно распечатывать лишь отдельные детали о файлах и каталогах, например только имя файла.

### **10.3. Задание**

7. Определите уникальное имя вашего головного личного каталога. Объясните структуру полного маршрутного имени каталога.
8. Создайте два поддерева из одного и из двух каталогов.
9. С использованием команды ls проверьте факт построения дерева подкаталогов.
10. Посмотрите содержимое пустых подкаталогов, т.е. новых подкаталогов, не содержащих файлов. Объясните их содержание.
11. Сделайте текущим последний каталог меньшего дерева.
12. Определите полное маршрутное имя.
13. Смените текущий последний каталог на подкаталог большего дерева.
14. Определите его полное маршрутное имя.
15. Поместите в созданные подкаталоги по 2-3 пустых файла не выходя из текущего. Используйте при этом разные способы задания маршрутного имени подкаталогов.
16. Просмотрите содержимое каталогов. Объясните содержания каждого поля каталога.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

17. Установите в качестве текущего НОМЕ-каталога.



10. Найдите обычные файлы с определением их полных маршрутных имен. Выполните то же для различных комбинаций известных вам условий поиска файлов.
13. Прделайте предыдущие задания для файлов типа каталог.
14. Выведите на экран принадлежащую вам регистрационную запись с использованием команды `ggr`.
15. Уничтожьте все построенные вами подкаталоги. Получите подтверждение выполнения команд по содержимому домашнего каталога.
16. Проанализируйте с использованием команды `history` содержание лабораторной работы, продумайте ответы на нижеприведенные контрольные вопросы и сдайте выполненную работу преподавателю. После получения зачета по работе – уничтожьте все созданные файлы и корректно выйдите из системы.

#### **10.4. Контрольные вопросы.**

1. Какие системные имена каталогов вам известны?
2. Каким образом можно построить отдельный каталог или цепочку каталогов?
3. Для чего и каким образом переопределяются текущие каталоги?
4. Как обратиться к файлам параллельных ветвей дерева каталогов? К вышележащему каталогу?
5. Какие условия поиска файлов вы знаете? Как комбинируются условия поиска? Как осуществляется поиск по дереву каталогов?
6. Какова последовательность действий при удалении одного каталога? Цепочки каталогов?
7. Объясните назначение и содержание каждого поля каталога.
8. Как отличить по содержимому каталога типы файлов, содержащихся в ваших каталогов.
9. Какую информацию содержит «пустой» вновь созданный каталог?
10. Как осуществить поиск файлов в системе каталогов по фрагментам текста файлов?



## 11. Практическое занятие № 6. Текстовый редактор vi ОС UBUNTU

### Цель работы

Изучение основных возможностей встроенного текстового редактора vi – наиболее распространённого средства для построения текстовых файлов, исходных текстов программ и shell-процедур.

### 11.1. Порядок выполнения работы

- Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу
- Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01
- Пока идёт импорт (около 15 минут) ознакомьтесь с теоретической частью
- Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.
- Выполните все действия, описанные в теоретической части (для тренировки).
- Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.
- Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.
- Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.
- Экпортируйте образ сервера на свой носитель или в своё облако для последующей работы.



## 11.2. Теоретическая часть

### *Изучаются вопросы:*

- вход в редактор и выход, сохранение файлов;
  - ввода текста;
  - удаления фрагментов текста;
  - копирования фрагментов текста в буферную область памяти;
  - вставки содержимого буфера в текст файла;
  - редактирования (изменения) текста;
  - поиска строк файлов по их фрагментам;
- многострочных операций с файлом (префиксные команды)

vi — текстовый редактор командной строки в Ubuntu и других ОС UBUNTU.

При запуске vi не открывается новое графическое окно, просмотр и редактирование файла производится в текущем окне терминала. За счет того, что vi может работать и без Графического пользовательского интерфейса (GUI), он может использоваться для редактирования файлов.

### Запуск редактора:

а) vi &nbsp; &nbsp; myfile (одно или несколько имен файлов через пробелы для последовательного вызова их на редактирование).

Если такого файла нет, то появится начало пустого файла; курсор установится в начале первой строки.

б) vi + myfile

На экране конец файла; а курсор - в начале последней строки.

Пример:



"bloknot\_new" [New File]

в) vi +10 myfile

На экране - часть файла и строка 10 - в центре экрана;

курсор - в начале этой строки.

#### Режимы работы.

Редактор работает в нескольких режимах:

- В режиме вставки (редактирования). Нажатие на клавишу приводит к вставке соответствующего символа в редактируемый текст.
- В командном режиме
  - нажатие на любую клавишу воспринимается как команда редактору, которая немедленно исполняется;
  - в этом режиме можно ввести команду с параметрами в командной строке.

Поэтому при работе с Vi пользователю всегда нужно обращать внимание на то, в каком режиме находится редактор.

#### Командный режим.

Редактор Vi всегда начинает работу в командном режиме. В этом режиме есть два способа отдавать команды редактору.

1. нажатие практически на любую клавишу редактор воспринимает как определенную команду. Команды не отображаются, а сразу выполняются. Например, нажатие стрелок не перемещает курсор, а выводит на экран символы, соответствующие командам, расположенным по этим клавишам. Переключиться из режима вставки в командный режим - при помощи Esc.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

2. для ввода более сложных многословных команд используется командная строка, вызов которой осуществляется нажатием клавиши ":" .



### Режим вставки (редактирования).

Для перехода к этому режиму следует использовать команду insert, выполнение которой осуществляется при нажатии на клавишу "i" в командном режиме. При переключении в этот режим внизу экрана появляется надпись Insert, можно вносить изменения в текст документа.

### Выход из vi

а) с сохранением изменений: переключиться в командный режим (Esc), ввести команду

: w q

б) без сохранения изменений: переключиться в командный режим (Esc), ввести команду

: q !

### Перемещение курсора

0 - в начало строки

\$ - в конец строки

w - в начало слова

b - в начало предыдущего слова

### Удаление

dw - удаление слова

d\$ - удаление до конца строки

d0 - удаление до начала строки

d7w - удаление 7 слов

### Изменение (замена)

c\$ (или C) &nbsp;&nbsp; <текст замены (может быть из нескольких строк)> Esc - замена конца строки (от курсора);

c^ <текст замены> Esc - замена от начала строки до курсора

cc < текст > Esc - замена одной строки;

5cc < текст > Esc - замена пяти строк.

### Создание новой строки

o - пустая строка после текущей строки;



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

О - пустая строка перед текущей строкой.



### Использование буфера обмена

Занести в буфер:

uw - сохранить слово (курсор - в начале слова);

уу - сохранить одну строку ;

5уу - сохранить 5 строк; и т.п.

При выполнении команд ndd (где n - число) удаляемые n строк заносятся в буфер.

### Вставка текста из буфера:

- p - после текущей строки;

- P - перед ней.

Пример работы с текстовым редактором

Vi + myfile.txt

```
i
helo!
How are you?
```

---

```
i
helo!
hOW ARE YOU?
C$ JJJ
```

```
i
helo!
hOW ?
C$ JJJ
```

Попробуйте объяснить смысл каждой строки из данного примера. Какие операции запрашивал пользователь? Какие запросы он вводил с клавиатуры, чтобы получить тот или иной результат?

## 11.3. Задание

1. Войдите в редактор с созданием нового пустого файла с произвольным именем и расширением -.1.
2. Поместите в созданный файл текст, включающий не менее четырех строк с несколькими словами в каждой.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

3. Вставьте по одной пустой строке до и после одной из строк файла.
4. Заполните пустые строки произвольным текстом.



5. Вставьте еще по одной строке в середину текста файла без предварительного резервирования пустых строк.
6. Перейдите в режим редактирования и выполните произвольное редактирование отдельных слов и строк файла с использованием всех команд из групп «Команды изменения текста» и «Команды отмены произвольных изменений в текущей строке».
7. Выйдите из редактора с сохранением файла. Убедитесь в сохранении созданного файла.
8. Войдите опять в vi для редактирования созданного файла с использованием команд из групп "Команды копирования в буфер" и "Команды вставки буфера в текст".
9. Поменяйте местами несколько слов в строках файла.
10. Поменяйте местами несколько строк.
11. Поменяйте местами последовательно начало строки с её концом и наоборот.
12. Выполните операцию поиска строк файла по заданным их фрагментам с различными направлениями поиска (см. «Команды поиска строки файла по фрагменту её текста»).
13. Не выходя из vi, перепишите полученную в результате редактирования версию файла в файл с тем же именем, но с расширением .new.
14. Добавьте первые 3-и строки редактируемого файла к файлу .1.
16. Не выходя из vi, загрузите в буфер файл с расширением .1.
17. Последовательно удалите части строк и несколько строк с использованием команд из групп «Команды удаления текста».
18. Выйдите из редактора без сохранения файла.
19. Просмотрите и проанализируйте содержимое редактируемых файлов в текущем каталоге.
20. Уничтожьте созданные файлы

#### 11.4. Контрольные вопросы

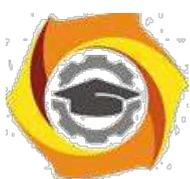


**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

1. В чем особенности и преимущества встроенного редактора vi ОС UBUNTU?



2. Какие два основных режима работы использует редактор? Как осуществляется переключение режимов?
3. Какую структуру имеет экран при редактировании файла? Назначение полей экрана?
4. Как в vi организованно редактирование открытого в нем файла?
5. Как организованна работа с клавишными командами редактора?
6. Что такое «префиксные команды» и их назначение? Как организована работа с командами этого вида?
7. Какие функции редактора вы использовали при выполнении лабораторной работы?



## 12. Практическое занятие 7. Введение в shell-программирование

### Цель работы

Ознакомление со средствами языка shell, для создания процедур обработки данных. Изучаются вопросы оформления shell-процедур.

### 12.1. Порядок выполнения работы

- Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу
- Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01
- Пока идёт импорт (около 15 минут) ознакомьтесь с теоретической частью
- Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.
- Выполните все действия, описанные в теоретической части (для тренировки).
- Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.
- Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.
- Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.
- Экпортируйте образ сервера на свой носитель или в своё облако для последующей работы.



## 12.2. Теоретическая часть

### *Изучение команд:*

Set – присваивание значения параметрам, передаваемым процедурам;

Echo, read, banner – вспомогательные команды для ввода и вывода информации;

If, then, else – команды проверки условий и ветвления вычисления в процедуре;

Test – проверка файлов, числовых величин, строк символов;

While, until, for – команды построения циклических процедур;

- правила построения и постановки значений переменных;

- вычисление арифметических выражений;

- обработка символьных строк.

Командный язык shell (в переводе - раковина, скорлупа) фактически есть язык программирования очень высокого уровня. На этом языке пользователь осуществляет управление компьютером. Обычно, после входа в систему вы начинаете взаимодействовать с командной оболочкой.

Программирование на языке Shell происходит в окне терминала, а сами операции выполняются сразу, как только нажимается клавиша Enter после введения текста функции. (В отличие, скажем, от языка C++, где для того, чтобы компьютер выполнил введённые вами функции, необходимо запускать написанную программу нажатием клавиши F9. В Shell такого нет, а значит, нужно быть внимательным при вводе функций и команд, т.к. в случае ошибки придётся переписывать всю программу заново).

Для того, чтобы, работая в терминале, переключиться в среду Shell, необходимо ввести команду sh. Признаком того, что оболочка (shell) готова к приему команд служит выдаваемый ею на экран промптер("\$").

```
rogdy@ubuntu:~$ sh
$
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Имя shell-переменной - это начинающаяся с буквы последовательность букв, цифр и подчеркиваний.

Значение shell-переменной - строка символов.



Для присваивания значений переменным может использоваться оператор присваивания "=".

```
var_1=13 - "13" - это не число, а строка из двух цифр. (аналог из C++: char
var_1[3] = "13")
```

```
var_2="ОС UBUNTU" - здесь двойные кавычки (" ") необходимы, так как в
строке есть пробел.
```

**ВАЖНО:** Обратим внимание на то, что, как переменная, так и ее значение должны быть записаны без пробелов относительно символа "=".

Возможны и иные способы присваивания значений shell-переменным. Так например запись,

```
$ DAT=`date`
```

приводит к тому, что сначала выполняется команда "date" (обратные кавычки говорят о том, что сначала должна быть выполнена заключенная в них команда), а результат ее выполнения, вместо выдачи на стандартный выход, приписывается в качестве значения переменной, в данном случае "DAT".

```
$ echo $DAT
Mon Oct 17 16:46:28 MSD 2011
```

Можно присвоить значение переменной и с помощью команды "read", которая обеспечивает прием значения переменной с (клавиатуры) дисплея в диалоговом режиме (аналог из C++: scanf или cin). Обычно команде "read" в командном файле предшествует команда "echo", которая позволяет предварительно выдать какое-то сообщение на экран. Например:

```
$ echo -n "vvedite znachenie x"
vvedite znachenie x$ read x
5
$ echo $x
5
```

При выполнении этого фрагмента командного файла, после вывода на экран сообщения

Введите трехзначное число:



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

интерпретатор остановится и будет ждать ввода значения с клавиатуры. То число, которое пользователь введёт с клавиатуры, станет значением переменной "x".



Одна команда "read" может прочитывать (присвоить) значения сразу для нескольких переменных. Если переменных в "read" больше, чем их введено (через пробелы), оставшимся присваивается пустая строка. Если передаваемых значений больше, чем переменных в команде "read", то лишние игнорируются.

**ПРЕДУПРЕЖДЕНИЕ.** На самом деле интерпретатор для продолжения работы ждет лишь нажатия клавиши. Введенное вами число воспринимается им не как число, а как последовательность символов(!). Интерпретатор не проверяет, что вы ввели. Поэтому в качестве значения переменной может оказаться любая введенная абракадабра или просто нажатие , как значение пустой строки. (Для обеспечения проверки формата ввода следует написать свою команду).

При обращении к shell-переменной необходимо перед именем ставить символ "\$" для получения значения этой переменной. Сравните две команды, по-разному набранные с клавиатуры и заметьте разницу в результатах,

распечатываемых программой на экране.

```
$ echo x
x
$ echo $x
5
```

Запись `echo x` печатает исключительно те буквы, которые введены после команды `echo`. (Аналог в C: `printf("x")` и `printf("%d", x)`).

И еще один пример. Фрагмент командного файла:

```
$ echo x = $x
```

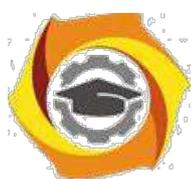
выдаст на экран

```
|x = 5
```

Что наглядно показывает, что команда `echo` может за один раз выводить на экран несколько переменных.

Команда "set" устанавливает значения параметров. Это бывает очень удобно. Например, команда "date" выдает на экран текущую дату, скажем, "Mon Oct 13 16:46:28 2011", состоящую из пяти слов, тогда

```
$ set `date`
$ echo $1 $3
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

выдаст на экран

Mon 17 \_\_\_\_\_



Команда "set" позволяет также осуществлять контроль выполнения программы, например:

- set -v** на терминал выводятся строки, читаемые shell.
- Set +v** отменяет предыдущий режим.
- Set -x** на терминал выводятся команды перед выполнением.
- Set +x** отменяет предыдущий режим.

Команда "set" без параметров выводит на терминал состояние программной среды

Как во всяком языке программирования в тексте на языке shell могут быть комментарии. Для этого используется символ "#". Все, что находится в строке (в командном файле) левее этого символа, воспринимается интерпретатором как комментарий. Например,

```
$ # ja kommentariy!
$ ## ja tozhe
$ ### i ja =P
```

---

Как во всяком процедурном языке программирования в языке shell есть операторы. Ряд операторов позволяет управлять последовательностью выполнения команд. В таких операторах часто необходима проверка условия, которая и определяет направление продолжения вычислений.

#### Команда test ("[ ")

Команда test проверяет выполнение некоторого условия. С использованием этой (встроенной) команды формируются операторы выбора и цикла языка shell.

Два возможных формата команды:

test условие

или

[ условие ]

Пробелы должны быть и между значениями и символом сравнения или операции (как, кстати, и в команде "expr") . Про последнюю следует сказать подробнее.

Именно команда expr даёт программе возможность воспринимать ту или иную переменную именно как число. (Иными словами, если сравнивать два ЯП, Shell и C, запись echo `expr \$x` будет аналогична записи printf("%d", x).



Разница в том, что в языке Си тип данных (char, int и пр.) определяется и устанавливается сразу при объявлении переменной, а в shell тип данных определяется после того, как значение было записано в переменную и может меняться в теле программы).

Пример работы с командами и операторами языка shell.

```
$ echo date: `date`
date: Mon Oct 17 17:27:46 MSD 2011
$ s=`expr $6 - $3`
$ echo $s
1994
```

Попробуйте объяснить смысл каждой строки из данного примера. Какие операции запрашивает пользователь, и какие ответы даёт система?

В shell используются условия различных "типов".

#### Условный оператор "if"

В общем случае оператор "if" имеет

структуру if [условие] ; echo \$0

then список

[elif условие

then список]

[else список]

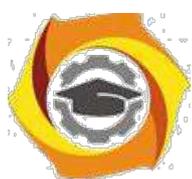
Fi

Здесь "elif" сокращенный вариант от "else if" может быть использован наряду с полным, т.е. допускается вложение произвольного числа операторов "if" (как и других операторов). Разумеется "список" в каждом случае должен быть осмысленный и допустимый в данном контексте. Конструкции

[elif условие

then список]

и



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

[else список]



не являются обязательными (в данном случае для указания на необязательность конструкций использованы квадратные скобки - не путать с квадратными скобками команды "test!").

пример использования оператора if:

```
$ z=7
$ if [expr $z > 0`] ; echo $0
> then "POLOZITELNO"
> [elif `expr &z<0` then echo "otrizatelno"]
> [else echo "null"]
> fi
sh
sh: POLOZITELNO:
```

Самая усеченная структура этого

оператора if условие

then список

fi

Обратите внимание, что структура обязательно завершается служебным словом "fi". Число "fi", естественно, всегда должно соответствовать числу "if".

Примеры.

Пусть написан расчет "if-1"

```
$ if [$1 -gt $2]
> then pwd
> else echo $0: Hello!
> fi
```

Тогда вызов расчета

if-1 12 11

даст

/home/sae/STUDY/SHELL,

а

if-1 12 13

даст

if-1 : Hello!

Оператор вызова ("case")



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Оператор выбора "case" имеет структуру:



```
case строка in
 шаблон) список команд;;
 шаблон) список команд;;
 ...
esac
```

Здесь "case" "in" и "esac" - служебные слова. "Строка" (это может быть и один символ) сравнивается с "шаблоном". Затем выполняется "список команд" выбранной строки. Непривычным будет служебное слово "esac", но оно необходимо для завершения структуры.

Пример.

```
###
case-1: Структура "case".
Уже рассматривавшийся в связи со
структурой "if" пример проще и
нагляднее можно реализовать с
помощью структуры "case".
echo -n " А какую оценку получил на экзамене?: "
read z
case $z in
 5) echo Молодец ! ;;
 4) echo Все равно молодец ! ;;
 3) echo Все равно ! ;;
 2) echo Все ! ;;
 *) echo ! ;;
esac
```

Непривычно выглядят в конце строк выбора ";;", но написать здесь ";" было бы ошибкой. Для каждой альтернативы может быть выполнено несколько команд. Если эти команды будут записаны в одну строку, то символ ";" будет использоваться как разделитель команд.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

Обычно последняя строка выбора имеет шаблон "\*", что в структуре "case" означает "любое значение". Эта строка выбирается, если не произошло совпадение значения переменной (здесь \$z) ни с одним из ранее записанных



шаблонов, ограниченных скобкой ")". Значения просматриваются в порядке записи.

```

case-2: Справочник.
Для различных фирм по имени выдается
название холдинга, в который она входит
case $1 in
 ONE|TWO|THREE) echo Холдинг: ZERO ;;
 MMM|WWW) echo Холдинг: Not-Net ;;
 Hi|Hello|Howdoing) echo Холдинг: Привет!
 ;; *) echo Нет такой фирмы ;;
esac
```

#### Оператор цикла с перечислением ("for")

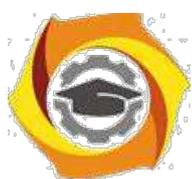
Оператор цикла "for" имеет структуру:

```
for имя in список значений
do
 список команд
done
```

где "for" - служебное слово определяющее тип цикла, "do" и "done" - служебные слова, выделяющие тело цикла. Не забывайте про "done"! Фрагмент "in список значений" может отсутствовать. Пример работы с оператором done:

```
$ a=1
$ b=2
$ c=3
$ for i in $a $b $c
> do
> echo "I love you!"
> done
I love you!
I love you!
I love you!
```

Т.о. мы видим, что переменная *i* в цикле пробегает по всем значениям, заданным в условии (т.е. по очереди принимает значения всех переменных, указанных в



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

условии), и для каждого из них выполняет ту операцию, которая заказана в теле цикла.



### Оператор цикла с истинным условием ("while")

Структура "while", также обеспечивающая выполнение расчетов, предпочтительнее тогда, когда неизвестен заранее точный список значений параметров или этот список должен быть получен в результате вычислений в цикле.

Оператор цикла "while" имеет структуру:

```
while условие
do
 список команд
done
```

где "while" - служебное слово определяющее тип цикла с истинным условием. Список команд в теле цикла (между "do" и "done") повторяется до тех пор, пока сохраняется истинность условия (т.е. код завершения последней команды в теле цикла равен "0") или цикл не будет прерван изнутри специальными командами ("break", "continue" или "exit"). При первом входе в цикл условие должно выполняться.

```
$ a=1
$ b=5
$ while [$a -lt $b]
> do
> a=`expr $a + 1`
> echo $a
> done
```

```
a=`expr $a + 1`
```

т.е. при каждой итерации значение "a" увеличивается на 1.

### Оператор цикла с ложным условием ("until")

Оператор цикла "until" имеет структуру:

```
until условие
do
 список команд
done
```



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

где "until" - служебное слово определяющее тип цикла с ложным условием.  
Список команд в теле цикла (между "do" и "done") повторяется до тех пор, пока сохраняется ложность условия или цикл не будет прерван изнутри



специальными командами ("break", "continue" или "exit"). При первом входе в цикл условие не должно выполняться.

Отличие от оператора "while" состоит в том, что условие цикла проверяется на ложность (на ненулевой код завершения последней команды тела цикла) проверяется ПОСЛЕ каждого (в том числе и первого!) выполнения команд тела цикла.

Примеры.

```
$ until false
> do
> read x
> if [$x=5]; echo $0
> then echo enough; break
> else echo some more
> fi
> done
```

Здесь программа с бесконечным циклом ждет ввода слов (повторяя на экране фразу "some more"), пока не будет введено "5". После этого выдается "enough" и команда "break" прекращает выполнение цикла.

### 12.3. Задание

1. Разработайте текст процедуры с использованием по заданию (см. ниже), вариант задания назначается преподавателем.
2. Отладьте, при необходимости отредактируйте и выполните процедуру.
3. Оформите процедуру с использованием вспомогательных команд и комментариев так, чтобы она легко читалась и чтобы результаты её работы легко анализировались.

### 12.4. Контрольные вопросы.

1. Что такое shell-процедура? Назначение?
2. Какого типа команды могут быть включены в тело процедуры?
3. Чем отличается обработка процедуры при выполнении от обработки программы на языке высокого уровня?



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

4. Что такое параметры? Для каких целей они используются? Какое число параметров может быть передано процедуре?



5. Какие вспомогательные команды вы использовали при оформлении процедуры?
6. Какого вида значения и как могут быть присвоены переменным языка shell?
7. Что такое локальные переменные и для каких целей их надо экспортировать в среду?
8. Как осуществляется ветвление вычислительного процесса процедуры?
9. Какого типа цикла в процедурах могут быть построены средствами языка shell?
10. Какие способы вызова процедур на исполнение вы знаете?

### **12.5. Варианты заданий к лабораторной работе № 12 «Введение в shell-программирование»**

Разработать shell-процедуру с комментариями, выполняющую ниже перечисленные функции.

1. Вводит последовательность из  $N$  слов и подсчитывает в каждом введенном слове число символов. Если число символов больше  $M$ , то слово выводится на экран. Значения  $N$  и  $M$  передаются в качестве параметров.
2. Вводит строку из заданного числа слов. Выделяет слова, начинающиеся на указанную параметром букву, подсчитывает число таких слов.
3. Вводит строку  $N$  слов, анализирует длину каждого слова, упорядочивает слова по их алфавиту и выводит список на экран. Значение  $N$  задается параметром.
4. Вводит заданное параметром число слов и выводит каждое слово на печать, сопровождая его порядковым номером.
5. Вводит произвольное число коротких символьных параметров, подсчитывает длину каждого из них и выводит на экран список значений длин и общее число введенных параметров.
6. Вводит несколько коротких чисел в виде параметров, подсчитывает их сумму и результат выводит на экран.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

7. Запрашивает последовательно ввод нескольких чисел со знаками и выводит на экран два списка чисел – положительных и отрицательных.



8. Запрашивает ввод строки символов, разделенных пробелами и заданной параметром длины, разбивает символы на пересекающиеся пары и выводит их на экран.
9. Ищет в личном головном каталоге пользователя созданные им файлы, выводит список их имен и распечатывается текст файла, заданного пользователем.
10. Создает новый подкаталог и помещает туда новые файлы, создаваемые пользователем по запросам процедуры. Имена новых файлов указываются параметрами.
11. Создает новый подкаталог и копирует туда из родительского каталога файлы заданного параметром типа.
12. Анализирует указанный параметром каталога и выводит на экран число файлов различного типа (обычные, директория, скрытые). Тип задается параметром.



### **13. Лабораторная работа №7 Программирование shell-процедур.**

Работа предусматривается выполнение индивидуального задания повышенной сложности по варианту согласно номеру в журнале. Работа предусматривает несколько самостоятельных выходов на машину для отладки процедуры.

Отчётом по работе является работающая процедура, продемонстрированная преподавателю с объяснениями её текста и алгоритма работы. Выполнение лабораторной работы в полном объеме является обязательным условием для получения по курсу в целом экзаменационной оценки – «отлично».

#### **13.1. Порядок выполнения работы**

- Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу
- Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01
- Пока идёт импорт (около 15 минут) ознакомьтесь с индивидуальным заданием
- Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.
- Выполните все действия, описанные в теоретической части (для тренировки).
- Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.
- Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.
- Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

- Экспортируйте образ сервера на свой носитель или в своё облако для последующей работы.



### 13.2. Индивидуальные задания к лабораторной работе

Вариант 1.

Написать shell-процедуру, которая:

- вводит передаваемое в качестве 1-го параметра количество символьных строк;
- в каждой введенной строке ищет подстроку, передаваемую в качестве второго параметра;
- заменяет каждую найденную подстроку на строку, передаваемую в качестве третьего параметра;
- выводит на экран каждую введенную строку и соответствующую ей новую строку.

Вариант 2.

Написать shell-процедуру, которая:

- вводит 2 символьные строки;
- в каждой введенной строке ищет подстроку, передаваемую в качестве параметра;
- заменяет каждую найденную подстроку на пробел;
- образует из полученных строк третью строку так, чтобы в ней чередовались слова из первой и второй строк;
- выводит на экран введенные строки и новую строку.

Вариант 3.

Написать shell-процедуру, которая:

- вводит символьную строку;
- во введенной строке ищет подстроку, передаваемую в качестве первого параметра;
- вставляет после каждой найденной подстроки символ, передаваемый в качестве второго параметра;
- удаляет из полученной строки символ, передаваемый в качестве третьего параметра;
- выводит на экран введенную и новую строку.



Вариант 4.

Написать shell-процедуру, которая:

- вводит символьную строку;
- проверяет введенную строку на совпадение со строкой, переданной в качестве 1-го параметра;
- если строки совпадают, то выдает на экран приглашение повторить ввод;
- если не совпадают, то сравнивает длину введенной строки с длиной 2-го параметра, и, в случае их равенства, выводит на экран введенную строку в обратном порядке составляющих ее символов;

Вариант 5.

Написать shell-процедуру, которая:

- вводит символьную строку;
- проверяет введенную строку на совпадение со строками, содержащимися в файле, имя которого передается в качестве 1-го параметра;
- для всех найденных совпадений заменяет соответствующие строки в файле на строку, переданную в качестве 2-го параметра;
- выводит на экран старое и новое содержимое файла, а также число найденных совпадений;

Вариант 6.

Написать shell-процедуру, которая:

- вводит символьную строку, содержащую маршрутное имя некоторого файла; проверяет введенное маршрутное имя, если оно начинается с символа /, на совпадение его первой части с маршрутным именем домашнего каталога пользователя;
- если введенное маршрутное имя содержит маршрутное имя домашнего каталога или является относительным, то проверяет существование указанного файла, в противном случае выводит на экран сообщение об ошибке;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- если файл существует, то выводит на экран его содержимое;



- если файл не существует, то создает его и записывает в него строку, передаваемую в качестве параметра;

Вариант 7.

Написать shell-процедуру, которая:

- вводит символьную строку, содержащую имя некоторого файла;
- проверяет наличие файла в домашнем каталоге или в одном из подкаталогов;
- если файл существует, то выводит на экран его содержимое;
- если файл не существует, то создает его и записывает в него с консоли некоторый текст;
- устанавливает для файла права доступа, передаваемые в качестве параметра.

Вариант 8.

Написать shell-процедуру, которая:

- Читает содержимое первого файла, передаваемого в качестве первого параметра;
- Читает содержимое второго файла, передаваемого в качестве второго параметра;
- Находит в первом файле строку, содержащую заданное третьим параметром слово;
- Вставляет содержимое второго файла после найденной строки первого файла; Выводит на экран содержимое полученного файла

Написать shell-процедуру, которая:

- Читает содержимое первого файла, передаваемого в качестве первого параметра;
- Читает содержимое второго файла, передаваемого в качестве второго параметра;
- Если число строк в первом и втором файлах одинаковое, то выводит на экран каждые 5 секунд попеременно строки из первого и второго файлов

Вариант 10.

Написать shell-процедуру, которая:

- Читает содержимое первого файла, передаваемого в качестве первого параметра;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Читает содержимое второго файла, передаваемого в качестве второго параметра;



- Выводит на экран каждые 7 секунд попеременно 2 строки из первого и 1 строку из второго файла, перемещаясь по файлам циклически

Вариант 11.

Написать shell-процедуру, которая:

- удаляет из заданного первым параметром каталога и всех подкаталогов файлы, дата последней модификации которых предшествует текущей дате минус число дней, переданное в качестве второго параметра;
- изменяет дату последней модификации всех остальных файлов указанного каталога на текущую без изменения содержимого файлов.

Вариант 12.

Написать shell-процедуру, которая:

- выводит на экран список всех пользователей системы, включенных в заданную первым параметром группу пользователей;
- для каждого из заданных третьим и следующими параметрами имен пользователей выводит на экран права доступа к заданному вторым параметром файлу.

Вариант 13.

Написать shell-процедуру, которая:

- Вводит символьную строку, содержащую некоторое целое число;
- Читает содержимое файла, передаваемого в качестве первого параметра;
- Выводит на экран каждые 6 секунд попеременно 2 строки из файла и 1 введенную строку, перемещаясь по файлу циклически

Вариант 14.

Написать shell-процедуру, которая:

- Вводит символьную строку;
- Читает содержимое файла, передаваемого в качестве первого параметра;
- Выводит на экран каждые 6 секунд попеременно 2 строки из файла и 1 введенную строку, перемещаясь по файлу циклически



Вариант 15.

Написать shell-процедуру, которая:

- Вводит символьную строку;
- Читает содержимое файла, передаваемого в качестве первого параметра;
- Выводит на экран каждые 6 секунд попеременно 2 строки из файла и 1 введенную строку, перемещаясь по файлу циклически

Вариант 16.

Написать shell-процедуру, которая:

- Вводит символьную строку, содержащую два целых числа  $m$  и  $n$ , разделенных пробелами;
- Читает содержимое файла, передаваемого в качестве первого параметра;
- Выводит на экран каждые 5 секунд попеременно  $m$  строк из файла и  $n$  строк "Будь здоров"

Вариант 17.

Написать shell-процедуру, которая:

- Вводит символьную строку, содержащую два целых числа  $m$  и  $n$ , разделенных пробелами;
- Читает содержимое файла, передаваемого в качестве первого параметра;
- Выводит на экран каждые  $m$  секунд попеременно  $n$  строк из файла и пустую строку

Вариант 18.

Написать shell-процедуру, которая:

- Вводит символьную строку, содержащую три целых числа  $k, m$  и  $n$ , разделенных пробелами;
- Читает содержимое файла, передаваемого в качестве первого параметра;
- Выводит на экран каждые  $m$  секунд попеременно  $m$  строк из файла и  $n$  пустых строк



Вариант 19.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра;
- Записывает через каждые 5 секунд в файл, имя которого передается в качестве второго параметра, попеременно строки из первого файла и текущее время и дату;
- Выводит на экран каждые 7 секунд текущее содержимое второго файла;
- При вводе с клавиатуры слова quit удаляет второй файл и завершает работу

Вариант 20.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра;
- Создает новый файл, имя которого передается в качестве второго параметра;
- Выводит на экран каждые 7 секунд очередную строку первого файла;
- Сортирует все выведенные на экран строки первого файла по длине и записывает их в новый файл;
- При вводе с клавиатуры слова quit удаляет новый файл и завершает работу

Вариант 21.

Написать shell-процедуру, которая:

- Читает содержимое файла, имя которого вводится при исполнении процедуры;
- Создает новый файл, имя которого передается в качестве параметра;
- Выводит на экран каждые 6 секунд очередные 2 строки файла;
- Сортирует выведенные на экран строки по длине и записывает их в новый файл;
- При вводе с клавиатуры слова end удаляет второй файл и завершает работу

Вариант 22.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра;
- Создает новый файл, имя которого передается в качестве второго параметра;



- Записывает в новый файл строки первого файла в обратном порядке, вставляя после каждого слова фразу "THAT IS ALL" столько раз , сколько задано третьим параметром

Вариант 23.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра;
- Создает в текущем каталоге цепочку подкаталогов с относительным маршрутным именем, повторяющим полное маршрутное имя текущего каталога;
- Создает в последнем подкаталоге новый файл, имя которого передается в качестве второго параметра;
- Записывает в новый файл строки первого файла в обратном порядке, вставляя после каждого слова фразу "THAT IS ALL" столько раз, сколько задано третьим параметром

Вариант 24.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра;
- Создает в текущем каталоге цепочку подкаталогов с относительным маршрутным именем, повторяющим полное маршрутное имя текущего каталога;
- Создает в последнем подкаталоге новый файл, имя которого передается в качестве второго параметра;
- Записывает в новый файл строки первого файла в обратном порядке, вставляя после каждого слова фразу "THAT IS ALL" столько раз, сколько задано третьим параметром

Вариант 25.

Написать shell-процедуру, которая:



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Читает содержимое файла, передаваемого в качестве первого параметра (в файле в каждой строке представлено одно целое число);



- Проверяет правильность формата содержимого файла, при ошибках выводит соответствующие сообщения и завершает работу;
- Подсчитывает сумму всех содержащихся в файле чисел;
- Выводит на экран полученную сумму

Вариант 26.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра (в файле в каждой строке представлено по два целых числа, разделенных пробелами);
- Проверяет правильность формата содержимого файла, при ошибках выводит соответствующие сообщения и завершает работу;
- Подсчитывает сумму чисел в каждой строке файла;
- Сортирует полученные суммы по убыванию и выводит их на экран

Вариант 27.

Написать shell-процедуру, которая:

- Читает содержимое двух файлов, передаваемых в качестве первых параметров (в файлах в каждой строке представлено по одному целому числу);
- Проверяет правильность формата содержимого файлов, при ошибках выводит соответствующие сообщения и завершает работу;
- Выбирает одинаковые числа в первом и втором файлах;
- Сортирует полученные числа по возрастанию и выводит их на экран

Вариант 28.

Написать shell-процедуру, которая:

- Читает содержимое двух файлов, передаваемых в качестве первых параметров (в файлах в каждой строке представлено по одному целому числу);
- Проверяет правильность формата содержимого файлов, при ошибках выводит соответствующие сообщения и завершает работу;
- Суммирует числа из первого и второго файлов, расположенные в строках с одинаковым номером;



- Сортирует полученные суммы по возрастанию и выводит их на экран

Вариант 29.

Написать shell-процедуру, которая:

- Читает содержимое двух файлов, передаваемых в качестве первых параметров (в файлах в каждой строке представлено не более, чем по три целых числа, разделенных пробелами);
- Проверяет правильность формата содержимого файлов, при ошибках выводит соответствующие сообщения и завершает работу;
- Вычисляет суммы чисел в каждой строке;
- Сортирует полученные положительные суммы по возрастанию и выводит их на экран

Вариант 30.

Написать shell-процедуру, которая:

- Читает содержимое трех файлов, передаваемых в качестве первых параметров (в файлах в каждой строке представлено не более, чем по четыре целых числа, разделенных пробелами);
- Проверяет правильность формата содержимого файлов, при ошибках выводит соответствующие сообщения и завершает работу;
- Вычисляет произведения чисел в каждой строке;
- Сортирует полученные произведения по возрастанию и выводит их на экран

Методические указания  
по дисциплине  
«Основы теории и методы оптимизации»

ВВЕДЕНИЕ

Предлагаемые методические указания составлены в соответствии с рабочей программой дисциплины «Основы теории и методы оптимизации», изучаемой студентами 3 курса очной и заочной форм обучения СКФ МТУСИ. Данные методические указания посвящены решению задачи линейного программирования симплекс-методом. Изложен теоретический материал, подкрепленный подробным решением примеров. В конце методических указаний предлагаются задачи для самостоятельного решения.

РЕШЕНИЕ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**СИМПЛЕКС-МЕТОДОМ**

Задачей линейного программирования (далее – ЗЛП) называется задача нахождения максимума или минимума линейной функции  $Z = c \cdot x + d \cdot y$  с линейными ограничениями на вектор в  $n$ -мерном арифметическом пространстве. В контрольной работе такая задача представлена заданием № 4.

Здесь  $x = (x_1, x_2, \dots, x_n)$ ,  $c = (c_1, c_2, \dots, c_n)$ , поэтому функцию  $Z$  можно представить в виде  $Z = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$ .

Функцию  $Z$  принято называть функцией цели. В формулировках задач на максимум (или минимум) функции  $Z$  обычно кратко записывают  $Z \rightarrow$  (или  $\leftarrow$ ).

Линейные ограничения на вектор являются линейными уравнениями и неравенствами с неизвестными  $x_1, x_2, \dots, x_n$ . Эти ограничения определяют некоторый многогранник, называемый многогранником допустимых решений. В задачах экономического или физического содержания естественным ограничением может быть неотрицательность переменных  $x_1, x_2, \dots, x_n$  (крайно записывается  $x_i \geq 0$ ).

Точка многогранника допустимых решений, в которой функция цели  $Z$  достигает свой максимум или минимум, называется оптимальным решением ЗЛП.

**Решение ЗЛП основано на понятии опорного решения системы линейных уравнений и оформляется с помощью специальных симплексных таблиц.**

Пусть система линейных уравнений (далее – СЛУ) представлена в виде  $Ax = b$ , где матрица  $A$  и вектор  $b$  известны. Вектор  $x$  и его координаты называются базисными, а вектор  $b$  и его координаты – свободными. Поэтому система линейных уравнений вида  $Ax = b$  называется СЛУ с выделенным базисом. Информация об этой СЛУ записывается в следующую таблицу, называемую симплексной таблицей:

|   |   |       |   |   |   |   |   |
|---|---|-------|---|---|---|---|---|
| — | — | —     | — | — | — | — | — |
|   |   | Своб. |   |   |   |   |   |
|   |   | Баз.  |   |   |   |   |   |
|   |   |       |   |   |   | 1 |   |
|   |   |       |   |   |   |   |   |

Опорным решением системы линейных уравнений  $Ax = b$  называется ее частное решение  $x = (x_1, x_2, \dots, x_n)$ ,  $x_i \geq 0$ . Если мы хотим найти другое опорное решение, то нужно перейти к равносильной СЛУ, обменяв одну координату вектора на одну координату вектора  $x$ . При этом изменятся матрица  $A$  и вектор  $b$ . Такие изменения называются модифицированными жордановыми исключениями (далее – МЖИ) и оформляются в симплексных таблицах по следующим правилам:



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

6. Из элементов матрицы выбрать генеральный элемент МЖИ  $a$  и по нему определить, какие именно координаты вектора и вектора поменяются местами в новой симплексной таблице (отмечены стрелками):

|      |       |     |     |     |     |     |     |     |
|------|-------|-----|-----|-----|-----|-----|-----|-----|
|      |       |     |     | ↑   |     |     |     |     |
|      | Своб. |     | ... |     | ... |     |     | 1   |
| Баз. |       |     |     |     |     |     |     | ... |
| 1    |       | ... | ... | ... | ... | ... | ... | ... |
| 2    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |

Генеральный элемент МЖИ не равен нулю! Строка и столбец генерального элемента также называются генеральными.

2. В новой симплексной таблице поменять местами указанные и :

|      |       |     |     |     |     |     |     |     |
|------|-------|-----|-----|-----|-----|-----|-----|-----|
|      | Своб. |     | ... |     | ... |     |     | 1   |
| Баз. |       |     |     |     |     |     |     | ... |
| 1    |       | ... | ... | ... | ... | ... | ... | ... |
| 2    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |

6. Новый базис умножить на генеральный элемент МЖИ:

|      |       |     |     |     |     |     |     |     |
|------|-------|-----|-----|-----|-----|-----|-----|-----|
|      | Своб. |     | ... |     | ... |     |     | 1   |
| Баз. |       |     |     |     |     |     |     | ... |
| 1    |       | ... | ... | ... | ... | ... | ... | ... |
| 2    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |

8. В новой симплексной таблице на месте генерального элемента записать 1:

|      |       |     |     |     |     |     |     |     |
|------|-------|-----|-----|-----|-----|-----|-----|-----|
|      | Своб. |     | ... |     | ... |     |     | 1   |
| Баз. |       |     |     |     |     |     |     | ... |
| 1    |       | ... | ... | ... | ... | ... | ... | ... |
| 2    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |
| ⋮    |       | ... | ... | ... | ... | ... | ... | ... |

9. Остальные элементы генеральной строки оставить без изменения.

10. Остальные элементы генерального столбца умножить на -1.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

5. Все остальные элементы таблицы пересчитать по правилу прямоугольников: если в исходной симплексной таблице имеется фрагмент



то в новой симплексной таблице на месте элемента следует записать число  $\frac{b_i}{a_{ij}}$ . Формула  $\frac{b_i}{a_{ij}}$  остается неизменной при любых перестановках строк или столбцов этого фрагмента.

Правила 4. – 7. можно объединить на одном рисунке:



**Пример 1.** Пусть имеется СЛУ

$$\begin{cases} 3z \\ 4 - 8x_1 + 2x_2 + 3x_3 + 7x_4 = 9 \\ 4x_1 - 6x_2 + 5x_3 + 4x_4 = 4 \end{cases}$$

Это СЛУ с выделенным базисом

матрицей  $B = \begin{pmatrix} 4 & -6 & 5 & 4 \\ -8 & 2 & 7 & 7 \end{pmatrix}$   $=(3, 2, 4)$ , свободные векторы  $=(9, 4)$ . При этих условиях можно составить

симплексную таблицу

|       |       |       |       |       |       |   |
|-------|-------|-------|-------|-------|-------|---|
|       | Своб. |       |       |       |       | 1 |
| Баз.  |       | $x_1$ | $x_2$ | $x_3$ | $x_4$ |   |
| $z$   |       | 4     | -6    | 5     | 5     | 9 |
| $x_3$ |       | -8    | 2     | 7     | 7     | 4 |

5. найти опорное решение СЛУ: базисные  $x_3 = 9, x_4 = 4 \Rightarrow x_1 = 3, x_2 = 1$ , свободные  $x_1 = x_2 = x_3 = x_4 = 0$ . Таким образом,  $x = (0, 3, 0, 0, 1)$ . Допустим, что мы хотим найти еще одно опорное решение СЛУ и в качестве генерального элемента МЖИ выбрали число 2:

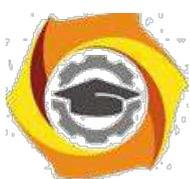
|       |       |       |       |       |       |   |
|-------|-------|-------|-------|-------|-------|---|
|       | Своб. |       |       |       |       | 1 |
| Баз.  |       | $x_1$ | $x_2$ | $x_3$ | $x_4$ |   |
| $z$   |       | 4     | -6    | 5     | 5     | 9 |
| $x_3$ |       | -8    | 2     | 7     | 7     | 4 |

Тогда по правилам 2. – 7. получается новая симплексная таблица

|       |       |       |       |       |       |    |
|-------|-------|-------|-------|-------|-------|----|
|       | Своб. |       |       |       |       | 1  |
| Баз.  |       | $x_1$ | $x_2$ | $x_3$ | $x_4$ |    |
| $z$   |       | -40   | 6     | 52    | 5     | 42 |
| $x_3$ |       | -8    | 1     | 7     | 7     | 4  |

7. новое опорное решение СЛУ: базисные  $x_3 = 42, x_4 = 4 \Rightarrow x_1 = 7, x_2 = 2$ , свободные  $x_1 = x_2 = x_3 = x_4 = 0$ . Таким образом,  $x = (7, 2, 0, 0)$ .

Если изначально базис не выделен, т.е. имеется произвольная СЛУ  $Ax = b, x \geq 0$ , то применяется метод ложного базиса: данную СЛУ записывают в виде  $0x + b = b$ , где  $b = 0$



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

будет играть роль базисного вектора, а  $a = -$  – свободного, и составляется симплексная таблица

|       |   |   |
|-------|---|---|
| Своб. | - | 1 |
| Баз.  | 0 | 1 |

Из такой симплексной таблицы невозможно получить опорное решение, поэтому нулевой базис 0 называется ложным базисом. Однако после нескольких преобразований МЖИ все базисные нули перейдут в разряд свободных элементов, а их место займут координаты вектора, и мы получим симплексную таблицу, соответствующую СЛУ с выделенным базисом, а вместе с ней и первое опорное решение. При этом столбцы нулей, перешедших

19. разряд свободных элементов, из симплексной таблицы удаляются.

### Практические замечания:

Числовой коэффициент свободного элемента может быть переведен в столбец этого элемента:

|       |     |     |       |
|-------|-----|-----|-------|
| Своб. | ... | ... | 1     |
| Баз.  | ... | ... | ...   |
| ...   | ... | ... | ...   |
| ...   | ... | ... | ... ⇒ |
| ⋮     | ⋮   | ⋮   | ⋮     |
| ...   | ... | ... | ...   |

8. Строку симплексной таблицы можно разделить на общий множитель элементов этой строки (если таковой множитель имеется и не равен нулю):

|       |     |     |     |     |     |
|-------|-----|-----|-----|-----|-----|
| Своб. | ... | ... | ... | ... | 1   |
| Баз.  | ⋮   | ⋮   | ⋮   | ⋮   | ⇒   |
| ⋮     | ⋮   | ⋮   | ⋮   | ⋮   | ⋮   |
| ...   | ... | ... | ... | ... | ... |

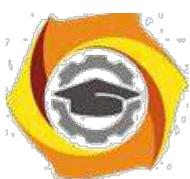
### Пример 2. Найти опорные решения СЛУ

$$3x_1 + 4x_2 + x_3 - 2x_4 = 4$$

Здесь базис не выделен, поэтому применим метод ложного базиса:

|       |   |   |   |    |    |   |
|-------|---|---|---|----|----|---|
| Своб. | 0 | 2 | 5 | -1 | 2  | 1 |
| Баз.  | 0 | 3 | 4 | 1  | -2 | 4 |
| 0     | 5 | 9 | 1 | 0  | 5  | 5 |
| 3     | 3 | 4 | 1 | -2 | 4  | 4 |

$\frac{1}{4}$   
 МЖИ



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

$$\begin{array}{c|ccc|c}
 \text{Своб.} & & & & 1 \\
 \text{Баз.} & & & & \\
 \hline
 s_1 & 1 & 9 & 0 & 5 \\
 s_2 & -3 & -7 & -10 & 5 \\
 \hline
 \end{array} \Rightarrow \begin{array}{c|ccc|c}
 \text{Своб.} & & & & 1 \\
 \text{Баз.} & & & & \\
 \hline
 s_1 & 9 & 0 & & 5 \\
 s_2 & -7 & -10 & & \\
 \hline
 \end{array} \uparrow$$

Опорное решение = (1, 0, 1, 0)

$$\begin{array}{c|ccc|c}
 \text{Своб.} & 5 & & & 1 \\
 \text{Баз.} & 9 & 1 & 0 & 5 \\
 \hline
 s_1 & & 7 & -90 & 80 \\
 \hline
 \end{array} = \begin{array}{c|ccc|c}
 \text{Своб.} & & & & 1 \\
 \text{Баз.} & & & & \\
 \hline
 s_1 & 9 & 0 & & 5 \\
 s_2 & -90 & & & 80 \\
 \hline
 \end{array} = \begin{array}{c|ccc|c}
 \text{Своб.} & & & & 1 \\
 \text{Баз.} & & & & \\
 \hline
 s_1 & 9 & 5 & 0 & 5 \\
 s_2 & 7 & -18 & & 16 \\
 \hline
 \end{array}$$

Опорное решение = (0, 5<sub>9</sub>, 16<sub>9</sub>, 0)

Остальные опорные решения = (16<sub>7</sub>, -5<sub>7</sub>, 0, 0), = (1, 0, 0, -1<sub>2</sub>) и = (0, 5<sub>9</sub>, 0, -8<sub>9</sub>) предлагаем найти самостоятельно.

Если нужно найти неотрицательные опорные решения ( $\geq 0$ ), то в каждой симплексной таблице числовые коэффициенты неизвестных должны быть только положительными, а числа в последнем столбце – только неотрицательными ( $\geq 0$ ). Для того, чтобы не утратить эти свойства при переходе по МЖИ к очередной симплексной таблице, нужно придерживаться следующих правил:

9. генеральный элемент МЖИ может быть только положительным,

10. симплексное отношение генерального элемента МЖИ должно быть наименьшим по его столбцу (принцип минимального симплексного отношения). Здесь надо пояснить: пусть – положительный элемент матрицы, а – последнее число в строке элемента :

|       |     |     |   |
|-------|-----|-----|---|
| Своб. | ... | ... | 1 |
| Баз.  |     |     | ⋮ |
|       |     |     |   |
|       |     |     |   |
|       |     |     |   |

Тогда симплексным отношением элемента называется число . Для неположительных элементов матрицы симплексные отношения не определяются.

**Пример 3.** Пусть мы ищем неотрицательные опорные решения СЛУ и уже имеется симплексная таблица:

|       |    |    |     |   |
|-------|----|----|-----|---|
| Своб. |    |    |     | 1 |
| Баз.  |    |    |     |   |
|       | -2 | 3  | -10 | 6 |
|       | 0  | -8 | 21  | 0 |
|       | -1 | 5  | 0   | 7 |
|       | -4 | 2  | -1  | 9 |

Проанализируем возможность выбора генерального элемента МЖИ по каждому столбцу матрицы :

15. в столбце 1 нет положительных чисел, поэтому в данном столбце нельзя вы-брать генеральный элемент МЖИ;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

– 24. в столбце 2 есть три положительных числа 3, 5 и 2; их симплексные отношения

$\frac{6}{3} = 2, \frac{7}{5} = 1,4$  и  $\frac{2}{2} = 1$ , соответственно; наименьшим является  $\frac{7}{5}$ , поэтому число 5 из столбца 2 можно выбрать генеральным элементом МЖИ; в столбце 3 имеется только одно положительное число 21; его можно выбрать генеральным элементом МЖИ.

Таким образом, существует только два способа выбрать генеральный элемент МЖИ в данной симплексной таблице:

|      |       |    |    |     |   |
|------|-------|----|----|-----|---|
|      | Своб. |    |    |     | 1 |
| Баз. |       |    |    |     |   |
|      |       | -2 | 3  | -10 | 6 |
|      |       | 0  | -8 | 21  | 0 |
|      |       | -1 | 5  | 0   | 7 |
|      |       | -4 | 2  | -1  | 9 |

**Решение канонической ЗЛП симплекс-методом.**

Канонической задачей линейного программирования называется задача линейного программирования с ограничениями-равенствами и условием неотрицательности неизвестных:

$$z = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (\text{или}) \quad \text{при} \quad \begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{cases} \quad x_i \geq 0$$

Напомним, что оптимальным решением ЗЛП называется точка максимума (или точка минимума соответственно) функции цели  $z$  на многограннике допустимых решений (в данном случае это многогранник  $x_i \geq 0$ ).

**Идея решения канонической ЗЛП содержится в следующей теореме:**

**Теорема.** Если каноническая ЗЛП имеет оптимальное решение, то его можно найти среди неотрицательных опорных решений СЛУ  $z = \dots$ .

Существует специальный симплекс-метод, который позволяет вести направленный перебор неотрицательных опорных решений СЛУ  $z = \dots$  с постоянным улучшением функции цели  $z$  (с увеличением  $z$  в задачах на максимум и с уменьшением  $z$  в задачах на минимум), что значительно сокращает решение задачи.

Для того, чтобы решить каноническую ЗЛП симплекс-методом, нужно переписать СЛУ  $z = \dots$  в виде СЛУ с ложным базисом  $0 + \dots = \dots$ , переписать функцию цели  $z = \dots + \dots$  в виде еще одного линейного уравнения  $z - \dots = \dots$ , расценивая как базисную переменную, и составить симплексную таблицу:

|      |       |   |  |  |   |
|------|-------|---|--|--|---|
|      | Своб. |   |  |  | 1 |
| Баз. |       |   |  |  |   |
|      |       | - |  |  | - |
|      |       | 0 |  |  |   |
|      |       |   |  |  |   |
|      |       |   |  |  |   |

Нижняя строка таблицы называется  $f$ -строкой, а вектор  $-$  называется строкой оценок. Именно знаки чисел строки оценок будут в дальнейшем указывать выбор генерального элемента МЖИ и сигнализировать об окончании решения задачи.

Следует неукоснительно соблюдать следующие требования:

12. генеральный элемент МЖИ можно выбирать только из элементов матрицы  $a_{ij}$ ,



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

\* при переходе к очередной симплексной таблице элементы  $f$ -строки преобразуются по тем же правилам МЖИ, что и другие строки таблицы,

\* поскольку задача решается с дополнительным условием неотрицательности  $\geq 0$ , то во всех симплексных таблицах должно выполняться условие  $\geq 0$ , поэтому генеральный элемент МЖИ следует выбирать из положительных элементов матрицы по принципу минимального симплексного отношения, изложенного выше (элементы  $f$ -строки в симплексных отношениях не участвуют).

Решая задачу, мы должны, в первую очередь, с помощью МЖИ вывести все нули из базиса. Тогда мы получаем начальное неотрицательное опорное решение СЛУ = и значение функции цели ( ) в этой точке. С этого момента вступают в силу правила, запускающие направленный перебор неотрицательных опорных решений СЛУ = :

в если выбрать генеральный элемент МЖИ в столбце над отрицательным числом строки оценок, то значение функции цели ( ) увеличится; когда в строке оценок не останется отрицательных чисел, задача на максимум будет решена,

в если выбрать генеральный элемент МЖИ в столбце над положительным числом строки оценок, то значение функции цели ( ) уменьшится; когда в строке оценок не останется положительных чисел, задача на минимум будет решена.

Продемонстрируем решение канонической ЗЛП симплекс-методом на примерах.

### Пример 4. Решить симплекс-методом ЗЛП

$(1) \rightarrow 1 - 2 + 3 + 2 - 4 = -1$

$(2) \rightarrow 0 + 2 + 1 + 2 - 4 = 1$

$(3) \rightarrow 0 + 1 + 4 + 2 - 4 = 3$

$(4) \rightarrow -3 + 2 + 4 - 4 = -1$

$1 - 2 + 3 - 4 = 1$

$0 + 2 + 1 + 2 = 5$

$0 + 1 + 4 + 2 = 7$

$1 + 2 - 3 + 2 + 4 = 6$

Введем ложный базис из нулей и переписыв функцию цели ( ) в виде еще одного линейного уравнения  $-3x_1 + 2x_2 - 3x_3 - 2x_4 = -1$ :

|      |       |    |    |    |    |    |
|------|-------|----|----|----|----|----|
|      | Своб. |    |    |    |    | 1  |
| Баз. |       | 1  | 2  | 3  | 4  |    |
| 0    |       | 1  | -3 | 4  | -1 | 1  |
| 0    |       | 2  | 1  | 1  | 1  | 5  |
| 0    |       | 1  | 4  | -3 | 2  | 4  |
|      |       | -3 | 1  | -1 | -2 | -1 |

В столбце переменной

и 1. Наименьшим является первое отношение. В соответствии с этим первое число указанного столбца можно выбрать генеральным элементом МЖИ:

|      |       |    |    |    |    |    |
|------|-------|----|----|----|----|----|
|      | Своб. |    |    |    |    | 1  |
| Баз. |       | 1  | 2  | 3  | 4  |    |
| 0    |       | ①  | -3 | 4  | -1 | 1  |
| 0    |       | 2  | 1  | 1  | 1  | 5  |
| 0    |       | 1  | 4  | -3 | 2  | 4  |
|      |       | -3 | 1  | -1 | -2 | -1 |

МЖИ

|      |       |    |    |    |    |   |
|------|-------|----|----|----|----|---|
|      | Своб. |    |    |    |    | 1 |
| Баз. |       | 0  | 2  | 3  | 4  |   |
| 1    |       | 1  | -3 | 4  | -1 | 1 |
| 0    |       | -2 | 7  | -7 | 3  | 3 |
| 0    |       | -1 | 7  | -7 | 3  | 3 |
|      |       | 3  | -8 | 11 | -5 | 2 |



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

|      |       |    |    |  |   |
|------|-------|----|----|--|---|
|      | Своб. |    |    |  | 1 |
| Баз. |       |    |    |  |   |
| 1    | -3    | 4  | -1 |  | 1 |
| 0    | 7     | -7 | 3  |  | 3 |
| 0    | 7     | -7 | 3  |  | 3 |
|      | -8    | 11 | -5 |  | 2 |

Повторившуюся строку удаляем:

|      |       |    |    |   |     |   |  |  |   |
|------|-------|----|----|---|-----|---|--|--|---|
|      | Своб. |    |    |   | 1   |   |  |  |   |
| Баз. |       |    |    |   |     | 0 |  |  | 1 |
| 1    | -3    | 4  | -1 | 1 | МЖИ |   |  |  |   |
| ← 0  | 7     | -7 | 3  | 3 | ⇒   |   |  |  |   |
|      | -8    | 11 | -5 | 2 |     |   |  |  |   |

|      |       |    |  |  |    |
|------|-------|----|--|--|----|
|      | Своб. |    |  |  | 1  |
| Баз. |       |    |  |  |    |
| 3,1  | -2    | 5  |  |  | 6  |
| 3,4  | 7     | -7 |  |  | 3  |
| 3    | 11    | -2 |  |  | 21 |

Поскольку нулей в базе больше нет, то мы получаем начальное неотрицательное опорное решение  $(x_1, x_2, x_3) = (2, 0, 0, 1)$  и значение функции цели  $Z = 2 \cdot 1 = 2$  в этой точке. В строке оценок есть отрицательное число  $-2$ , поэтому значение функции цели можно увеличить, если выбрать генеральный элемент МЖИ в столбце над этим числом:

|      |       |    |    |     |   |   |  |  |   |
|------|-------|----|----|-----|---|---|--|--|---|
|      | Своб. |    |    |     | 1 |   |  |  |   |
| Баз. |       |    |    |     |   | 3 |  |  | 1 |
| 3,1  | -2    | 5  | 6  | МЖИ |   |   |  |  |   |
| 3    | 7     | -7 | 3  | ⇒   |   |   |  |  |   |
| 3    | 11    | -2 | 21 |     |   |   |  |  |   |

|      |       |   |     |  |   |
|------|-------|---|-----|--|---|
|      | Своб. |   |     |  | 1 |
| Баз. |       |   |     |  |   |
| 3,3  | -2    | 1 | 6   |  |   |
| 15   | 21    | 7 | 57  |  |   |
| 15   | 51    | 2 | 117 |  |   |

В строке оценок нет отрицательных чисел, поэтому дальнейшее увеличение значения функции цели невозможно. Это означает, что задача на максимум решена. Ее оптимальное решение (точка максимума)  $(x_1, x_2, x_3) = (0, 0, 0)$  и максимальное значение функции цели  $Z = 0$ .

Вывод: оптимальное решение задачи на максимум найдено. Максимальное значение функции цели равно  $Z = 0$ .

### Пример 5. Решить симплекс-методом ЗЛП

$$Z = 4x_1 + 2x_2 + 5x_3 \rightarrow \max \text{ при } \begin{cases} x_1 + 2x_2 + 5x_3 = 9 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Введем допустимый базис из нулей и перепишем функцию цели  $Z$  в виде еще одного

$$\text{линейного уравнения } -x_1 - 4x_2 - 2x_3 - 4 = 0:$$



$$\begin{aligned} 0 + 14z_2 & & + 34z_3 + 9 & \Rightarrow \\ (0 + 14z_2 - 3z_3 + 4z_4) & & & \\ -1 \cdot 4z_2 - 2z_3 & & & \Rightarrow \end{aligned}$$

|       |   |    |    |    |    |   |
|-------|---|----|----|----|----|---|
| ↑     |   |    |    |    |    |   |
| Своб. |   |    |    |    |    | 1 |
| Баз.  | 0 | 1  | 2  | 1  | 5  | 9 |
|       | 0 | 1  | 3  | -1 | 1  | 4 |
|       |   | -1 | -4 | -2 | -1 | 0 |

|       |   |    |    |    |   |   |
|-------|---|----|----|----|---|---|
| Своб. | 0 |    |    |    |   | 1 |
| Баз.  | 0 | -1 | -1 | 2  | 4 | 5 |
|       |   | 1  | 3  | -1 | 1 | 4 |
|       |   | 1  | -1 | -3 | 0 | 4 |

|       |   |    |    |   |   |   |
|-------|---|----|----|---|---|---|
| ↑     |   |    |    |   |   |   |
| Своб. |   |    |    |   |   | 1 |
| Баз.  | 0 | -1 | 2  | 4 | 5 | 9 |
|       |   | 3  | -1 | 1 | 4 | 4 |
|       |   | -1 | -3 | 0 | 4 | 4 |

|       |                |    |     |    |    |   |   |
|-------|----------------|----|-----|----|----|---|---|
| Своб. |                |    |     |    |    | 0 | 1 |
| Баз.  | 4 <sub>4</sub> | -1 | 2   | 1  | 5  |   |   |
|       | 4 <sub>1</sub> | 13 | -6  | -1 | 11 |   |   |
|       | 4              | -4 | -12 | 0  | 16 |   |   |

|       |                |    |     |    |  |   |
|-------|----------------|----|-----|----|--|---|
| Своб. |                |    |     |    |  | 1 |
| Баз.  | 4 <sub>4</sub> | -1 | 2   | 5  |  |   |
|       | 4 <sub>1</sub> | 13 | -6  | 11 |  |   |
|       | 4              | -4 | -12 | 16 |  |   |

Поскольку нулей в базисе больше нет, то мы получаем начальное неотрицательное опорное решение  $= (11_4, 0, 0, 5_4)$  и значение функции цели  $= 16_4 = 4$  в этой точке. При этом в строке оценок нет положительных чисел, поэтому дальнейшее уменьшение значения функции цели невозможно. Это означает, что задача на минимум решена, т.е. найденное начальное неотрицательное опорное решение  $= (11_4, 0, 0, 5_4)$  уже является оптимальным решением (точкой минимума) и  $= 4$ .

### Решение произвольных ЗЛП симплекс-методом.

Все ЗЛП с ограничениями-неравенствами сводятся к каноническим ЗЛП в пространстве большей размерности путем введения дополнительных переменных, позволяющих перейти к ограничениям-равенствам. Продемонстрируем метод дополнительных переменных на примерах.

#### Пример 6. Решить симплекс-методом ЗЛП

Данная задача сформулирована в 3-мерном пространстве и не является канонической ЗЛП, поскольку первые два ограничения – неравенства. Для того, чтобы эти ограничения стали равенствами, нужно в первом из них увеличить левую часть на некоторую неотрицательную величину 4, а во втором – уменьшить левую часть на некоторую неотрицательную величину 5. Тогда система ограничений примет вид

$$\begin{aligned} 1x + 2x_2 + 3x_3 & = 4 \\ -1x + 2x_3 & = -5 \\ 3x_1 + x_2 & = 2 \end{aligned}$$

Это ограничения канонической ЗЛП в 5-мерном пространстве. Неотрицательные величины 4, 5 и есть дополнительные переменные. Далее, как в примерах 4 и 5, вводим



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

ложный базис из нулей и переписываем функцию цели ( ) в виде еще одного линейного

уравнения  $-2x_1 - 4x_2 - x_3 = 3$ :

$$\begin{aligned} 0 &= 14 - 2x_2 - 3x_4 \\ 0 &= -14x_2 - 3 \\ & \dots \\ -2x_1 - 4x_2 - x_3 &= 3 \end{aligned}$$

$$\begin{aligned} &= 4 \\ &= 3 \\ &= 2 \\ &= 3 \end{aligned}$$

|      |       |    |    |    |    |   |   |
|------|-------|----|----|----|----|---|---|
|      | Своб. |    |    |    |    |   | 1 |
| Баз. |       | 1  | 2  | 3  | 4  | 5 |   |
| 0    | 1     | 1  | 2  | 1  | 0  | 0 | 4 |
| 0    | -1    | 5  | -1 | 0  | -1 | 0 | 3 |
| 0    | 3     | 1  | -2 | 0  | 0  | 0 | 2 |
|      |       | -2 | -4 | -1 | 0  | 0 | 3 |

В столбце переменной  $x_4$  единственный положительный элемент 1. Его выберем генеральным элементом МЖИ:

|      |       |   |     |     |     |   |   |
|------|-------|---|-----|-----|-----|---|---|
|      | Своб. |   |     |     |     |   | 1 |
| Баз. |       | 1 | 2   | 3   | 4   | 5 |   |
| -0   | 1     | 1 | 2   | 1   | 0   | 0 | 4 |
| 0    | -1    | 5 | -10 | -1  |     |   | 3 |
| 0    | 3     |   | 1   | -20 |     |   | 2 |
|      |       |   | -2  | -4  | -10 | 0 | 3 |

|      |       |   |     |     |     |   |   |
|------|-------|---|-----|-----|-----|---|---|
|      | Своб. |   |     |     |     |   | 1 |
| Баз. |       | 1 | 2   | 3   | 4   | 5 |   |
| -0   | 1     | 1 | 2   | 1   | 0   | 0 | 4 |
| 0    | -1    | 5 | -10 | -1  |     |   | 3 |
| 0    | 3     |   | 1   | -20 |     |   | 2 |
|      |       |   | -2  | -4  | -10 | 0 | 3 |

|      |       |   |    |    |    |   |   |
|------|-------|---|----|----|----|---|---|
|      | Своб. |   |    |    |    |   | 1 |
| Баз. |       | 1 | 2  | 3  | 4  | 5 |   |
| -0   | 1     | 1 | 2  | 0  | 0  | 0 | 4 |
| 0    | -1    | 5 | -1 | -1 |    |   | 3 |
| 0    | 3     | 1 | -2 | 0  |    |   | 2 |
|      |       |   | -2 | -4 | -1 | 0 | 3 |

В столбце переменной  $x_1$  есть два положительных числа 1 и 3. Их симплексные отноше-

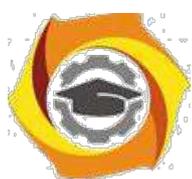
ния  $\frac{4}{1}$  и  $\frac{3}{3}$ . Наименьшим является второе отношение. В соответствии с этим число 3 ука-

занного столбца можно выбрать генеральным элементом МЖИ:

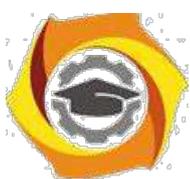
|      |       |   |    |    |     |   |   |
|------|-------|---|----|----|-----|---|---|
|      | Своб. |   |    |    |     |   | 1 |
| Баз. |       | 1 | 2  | 3  | 4   | 5 |   |
| -0   | 1     | 1 | 2  | 0  | 0   | 0 | 4 |
| 0    | -1    | 5 | -1 | -1 |     |   | 3 |
| -0   | 3     | 1 | -2 | 0  |     |   | 2 |
|      |       |   | -2 | -4 | -10 | 0 | 3 |

|      |       |   |    |    |     |   |   |
|------|-------|---|----|----|-----|---|---|
|      | Своб. |   |    |    |     |   | 1 |
| Баз. |       | 1 | 2  | 3  | 4   | 5 |   |
| -0   | 1     | 1 | 2  | 0  | 0   | 0 | 4 |
| 0    | -1    | 5 | -1 | -1 |     |   | 3 |
| -0   | 3     | 1 | -2 | 0  |     |   | 2 |
|      |       |   | -2 | -4 | -10 | 0 | 3 |

|      |       |   |    |    |     |   |   |
|------|-------|---|----|----|-----|---|---|
|      | Своб. |   |    |    |     |   | 1 |
| Баз. |       | 1 | 2  | 3  | 4   | 5 |   |
| -0   | 1     | 1 | 2  | 0  | 0   | 0 | 4 |
| 0    | -1    | 5 | -1 | -1 |     |   | 3 |
| -0   | 3     | 1 | -2 | 0  |     |   | 2 |
|      |       |   | -2 | -4 | -10 | 0 | 3 |



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

В столбце переменной  $x_2$  есть три положительных числа 2, 16 и 1. Их симплексные отношения  $2, 16$  и  $1$ . Наименьшим является второе отношение. В соответствии с этим второе число указанного столбца можно выбрать генеральным элементом МЖИ:

|            |                       |
|------------|-----------------------|
| ↑          |                       |
| Своб. Баз. | 1                     |
| $z_4$      | 10 <small>МЖИ</small> |
| 0          | 11 =                  |
| $z_1$      | 2                     |
| 3          | 13                    |
| 2          | 8                     |
| 0          | -5                    |
| 1          | -2                    |
| -10        | -7                    |
| -16        | -3                    |
| 1          | -3                    |
| 138        | 6                     |
| 16         | -5                    |
| -27        | 3                     |
| 48         | -30                   |
| 10         | -162                  |
| -1         | -27                   |
| -2         | -7                    |
| -1         | -5                    |
| 23         | 1                     |
| 16         | -3                    |
| -9         | 1                     |
| 8          | -27                   |
| 23         | 1                     |
| 11         | 11                    |
| 7          | 7                     |
| 53         | 53                    |

Поскольку нулей в базе больше нет, то мы получаем начальное неотрицательное опорное решение  $x_1 = 16^7, x_2 = 11, x_3 = 0, x_4 = 23, x_5 = 0$  и значение функции цели = 53 на нем. В строке оценок есть отрицательные числа -27 и -5, поэтому значение функции цели

можно увеличить, если выбрать генеральный элемент МЖИ в столбце над одним из этих чисел:

|            |                       |
|------------|-----------------------|
| ↑          |                       |
| Своб. Баз. | 1                     |
| $z_4$      | 23 <small>МЖИ</small> |
| $z_1$      | 11 =                  |
| -16        | 7                     |
| 8          | 53                    |
| 23         | 1                     |
| -5         | -3                    |
| -9         | -1                    |
| -27        | -5                    |
| 32         | -1                    |
| -32        | 3                     |
| -9         | 1                     |
| 8          | -72                   |
| 16         | 5                     |
| 16         | 16                    |
| 7          | 7                     |
| 88         | 88                    |
| 4          | -2                    |
| -2         | 3                     |
| -9         | 16                    |
| -9         | 10                    |
| 16         | 16                    |
| 32         | 32                    |
| 7          | 7                     |
| 88         | 88                    |

Мы получаем очередное неотрицательное опорное решение  $x_1 = 0, x_2 = 2, x_3 = 0, x_4 = 2, x_5 = 7$  и значение функции цели = 11 на нем. В строке оценок есть отрицательное число -9, поэтому значение функции цели можно еще увеличить, если выбрать генеральный элемент МЖИ в столбце над этим числом:



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

|               |       |        |
|---------------|-------|--------|
|               | ↑     |        |
| Своб.<br>Баз. | 1     | 1      |
| ← 4           | 0     | 4 - 22 |
| 2             | -2 3  | 2 =    |
| 5             | -9 16 | 7      |
|               | -9 10 | 11     |

|                    |      |    |
|--------------------|------|----|
| Своб.<br>Баз.      | 1    | 1  |
| МЖИ 4 <sub>3</sub> | 1 -2 | 2  |
| 4 <sub>2</sub>     | 2 8  | 12 |
| 4 <sub>5</sub>     | 9 46 | 46 |
| 4                  | 9 22 | 62 |

В строке оценок нет отрицательных чисел, поэтому дальнейшее увеличение значения функции цели невозможно, т.е. каноническая задача на максимум в 5-мерном пространстве решена. Ее оптимальное решение (точка максимума)

$$z = 0, \quad x_1 = 0, \quad x_2 = 0, \quad x_3 = 0, \quad x_4 = 0, \quad x_5 = 0$$

= 46 - 22, максимальное значение функции цели  
= 46 - 11, если...

ключить дополнительные переменные  $x_4$  и  $x_5$ , то мы получаем решение исходной задачи

на максимум в 3-мерном пространстве:

оптимальное решение (точка максимума)

максимальное значение функции цели

### Пример 7. Решить симплекс-методом ЗЛП

$$z(x) = 3x_1 + 2x_2 + 4x_3 \rightarrow \min \text{ при } \begin{cases} x_1 + 4x_2 + 2x_3 + x_4 = 5 \\ x_1 + 2x_2 + 2x_3 + x_5 = 4 \\ x_1, x_2, x_3, x_4, x_5 \geq 0 \end{cases}$$

Данная задача сформулирована в 4-мерном пространстве и не является канонической ЗЛП, поскольку первое ограничение – неравенство. Для того, чтобы это ограничение стало равенством, нужно уменьшить его левую часть на некоторую неотрицательную величину  $x_4$ , которая будет играть роль дополнительной переменной. Тогда система ограничений примет вид

$$\begin{cases} x_1 + 4x_2 + 2x_3 + x_4 = 5 \\ x_1 + 2x_2 + 2x_3 + x_5 = 4 \\ x_1, x_2, x_3, x_4, x_5 \geq 0 \end{cases}$$

Это ограничения канонической ЗЛП в 5-мерном пространстве. Далее, как в приме-

рах 4 и 5, вводим ложный базис из нулей и перепишем функцию цели  $z(x)$  в виде еще

одного линейного уравнения  $-3x_1 - 2x_2 - 4x_3 + x_6 = 0$ :

$$\begin{cases} -3x_1 - 2x_2 - 4x_3 + x_6 = 0 \\ x_1 + 4x_2 + 2x_3 + x_4 = 5 \\ x_1 + 2x_2 + 2x_3 + x_5 = 4 \\ x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{cases}$$

В столбце переменной  $x_2$  единственный положительный элемент 4. Его выберем

генеральным элементом МЖИ:

|               |    |    |    |    |    |   |   |
|---------------|----|----|----|----|----|---|---|
| Своб.<br>Баз. | 1  | 2  | 3  | 4  | 5  | 6 | 1 |
| 0             | 1  | 4  | 2  | 1  | -1 | 5 | 5 |
| 0             | 2  | -1 | 1  | 2  | 0  | 4 | 4 |
| 0             | -3 | -1 | -2 | -1 | 0  | 0 | 0 |

|               |    |    |    |    |    |   |   |
|---------------|----|----|----|----|----|---|---|
|               | ↑  |    |    |    |    |   |   |
| Своб.<br>Баз. | 1  | 2  | 3  | 4  | 5  | 6 | 1 |
| 0             | 1  | 4  | 2  | 1  | -1 | 5 | 5 |
| 0             | 2  | -1 | 1  | 2  | 0  | 4 | 4 |
|               | -3 | -1 | -2 | -1 | 0  | 0 | 0 |



**Негосударственное час**

**ВЫСШ  
«Техниче**

**ние**

Баз.

мюи

4.2

|   |     |   |   |    |    |    |     |
|---|-----|---|---|----|----|----|-----|
| 4 | -11 | 1 | 1 | -1 | -6 | -3 | -15 |
| 1 | 1   | 2 | 1 | -1 | 5  | 1  | 15  |
| = |     |   |   |    |    |    | =   |

Своб. 0

= 1 0 9 1 6 9 -1 21



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

$$\begin{array}{c|cccc|c}
 \text{Своб.} & & & & & 1 \\
 \text{Баз.} & & & & & \\
 \hline
 & 1 & 2 & 1 & -1 & 5 \\
 \hline
 0 & 9 & 6 & 9 & -1 & 21 \\
 \hline
 4 & & -11 & -6 & -3 & -1 & 5
 \end{array}$$

В столбце переменной  $x_1$  есть два положительных числа 1 и 9. Их симплексные отноше-

ния  $1^5$  и  $2^1_9$ . Наименьшим является второе отношение. В соответствии с этим число 9 ука-

занного столбца можно выбрать генеральным элементом МЖИ:

$$\begin{array}{c|cccc|c}
 \text{Своб.} & & & & & 1 \\
 \text{Баз.} & & & & & \\
 \hline
 & 1 & 2 & 1 & -1 & 5 \\
 \hline
 0 & 9 & 6 & 9 & -1 & 21 \\
 \hline
 4 & -11 & -6 & -3 & -1 & 5
 \end{array}
 \Rightarrow
 \begin{array}{c|cccc|c}
 \text{Своб.} & 0 & & & & 1 \\
 \text{Баз.} & & & & & \\
 \hline
 & -1 & 12 & 0 & -8 & 24 \\
 \hline
 9_1 & 1 & 6 & 9 & -12 & 21 \\
 \hline
 36 & 11 & 12 & 72 & -20 & 276
 \end{array}$$

$$\begin{array}{c|cccc|c}
 \text{Своб.} & & & & & 1 \\
 \text{Баз.} & & & & & \\
 \hline
 & 12 & 0 & -8 & 24 & \\
 \hline
 9_1 & 6 & 9 & -12 & 21 & \\
 \hline
 36 & 12 & 72 & -20 & 276 &
 \end{array}
 \Rightarrow
 \begin{array}{c|cccc|c}
 \text{Своб.} & & & & & 1 \\
 \text{Баз.} & & & & & \\
 \hline
 & 3 & 0 & -26 & & \\
 \hline
 9_1 & 6 & 9 & -1 & 21 & \\
 \hline
 9 & 3 & 18 & -5 & 69 &
 \end{array}$$

Поскольку нулей в базисе больше нет, то мы получаем начальное неотрицательное опорное решение

на нем. В строке оценок есть положительные числа 3 и 18, поэтому значение функции цели можно уменьшить, если выбрать генеральный элемент МЖИ в столбце над одним из этих чисел:

$$\begin{array}{c|cccc|c}
 \text{Своб.} & & & & & 1 \\
 \text{Баз.} & & & & & \\
 \hline
 & 3 & 0 & -2 & 6 & \\
 \hline
 9_1 & 6 & 9 & -1 & 21 & \\
 \hline
 9 & 3 & 18 & -5 & 69 &
 \end{array}
 \Rightarrow
 \begin{array}{c|cccc|c}
 \text{Своб.} & 9 & & & & 1 \\
 \text{Баз.} & & & & & \\
 \hline
 & 27 & 0 & -18 & 54 & \\
 \hline
 9_4 & 6 & 1 & -1 & 21 & \\
 \hline
 81 & -81 & -18 & -27 & 243 &
 \end{array}$$

$$\begin{array}{c|cccc|c}
 \text{Своб.} & 9 & & & & 1 \\
 \text{Баз.} & & & & & \\
 \hline
 9_2 & 3 & 0 & -2 & 6 & \\
 \hline
 9_4 & 6 & 1 & -1 & 21 & \\
 \hline
 9 & -9 & -2 & -3 & 27 &
 \end{array}$$

В строке оценок нет положительных чисел, поэтому дальнейшее уменьшение значения функции цели невозможно, т.е. каноническая задача на минимум в 5-мерном пространстве решена. Ее оптимальное решение (точка минимума)

а минимальное значение функции цели

читать дополнительную переменную  $x_5$ , то мы получаем решение исходной задачи на минимум в 4-мерном пространстве:



# Основы теории массового обслуживания

## Методические указания по практическим занятиям Практическое занятие № 1

### Вычисление вероятностных характеристик случайных процессов

#### 1. Цель занятия:

Выработать умения и навыки по вычислению вероятностных характеристик случайных процессов (СП). Выработать умения и навыки по вычислению основных взаимных характеристик двух СП.

#### 2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: (1) с. 386-399, (3) с. 4-7

#### 3. Задание:

Решить примеры: (2) №№ 756-778 (выборочно, по рекомендации преподавателя).

#### 4. Порядок выполнения:

- 4.1. Записать определение случайного процесса, его сечения и реализации.
- 4.2. Дать определение математического ожидания СП, перечислить его свойства.
- 4.3. Записать определение корреляционной функции СП, перечислить ее свойства.
- 4.4. Записать формулу, связывающую дисперсию и корреляционную функцию СП.
- 4.5. Записать формулу для вычисления нормированной корреляционной функции.
- 4.6. Записать определение взаимной корреляционной функции, перечислить ее свойства.
- 4.7. Записать формулу для вычисления нормированной взаимной корреляционной функции.
- 4.8. Записать формулу для вычисления математического ожидания суммы двух СП.
- 4.9. Записать формулу для вычисления корреляционной функции суммы двух СП.
- 4.10. Решить примеры, см. п.3.

#### 5. Отчет:

- 5.1. Определение случайного процесса, его сечения и реализации.
- 5.2. Определение математического ожидания СП, его свойства.
- 5.3. Определение корреляционной функции СП, ее свойства.
- 5.4. Формула, связывающая дисперсию и корреляционную функцию СП.
- 5.5. Формула для вычисления нормированной корреляционной функции.
- 5.6. Определение взаимной корреляционной функции, ее свойства.
- 5.7. Формула для вычисления нормированной взаимной корреляционной функции
- 5.8. Формула для вычисления математического ожидания суммы двух СП.
- 5.9. Формула для вычисления корреляционной функции суммы двух СП.
- 5.10. Решенные примеры.

#### 6. Список литературы:

1. Тихонов В.И., Шахтарин Б.И., Сизых В.В. Случайные процессы. Примеры и задачи. Том 5. Оценка сигналов, их параметров и спектров. Основы теории информации. Учебное пособие для вузов. М.: Горячая линия - Телеком, 2012.
2. Гмурман В.Е. Руководство к решению теории вероятностей и математическая статистика: учебное пособие. М.: Юрайт 2009



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**3. Ефименко В.Н., Прушинская Л.А. Теория случайных процессов. Учебное пособие. Ростов н/Д, СКФ МТУСИ, 2006.**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Практическое занятие № 2**

**Вычисление аналитических характеристик на выходе дифференциатора**

**1. Цель занятия:**

Познакомить студентов с понятием производной СП. Выработать умения и навыки по вычислению аналитических характеристик на выходе дифференциатора.

**2. Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников: (1) с. 405-409, (3) с. 9-10

**3. Задание:**

Решить примеры: (2) №№ 794-810 (выборочно, по рекомендации преподавателя).

**4. Порядок выполнения:**

- 4.1. Записать определение дифференцируемого случайного процесса и его производной.
- 4.2. Записать формулу для вычисления математического ожидания производной СП.
- 4.3. Записать формулу для вычисления корреляционной функции производной СП.
- 4.4. Записать формулу для определения взаимной корреляционной функции СП и его производной.
- 4.5. Решить примеры, см. п.3.

**5. Отчет:**

- 5.1. Определение дифференцируемого случайного процесса и его производной.
- 5.2. Формула для вычисления математического ожидания производной СП.
- 5.3. Формула для вычисления корреляционной функции производной СП.
- 5.4. Формула для определения взаимной корреляционной функции СП и его производной.
- 5.5. Решенные примеры.

**6. Список литературы:**

- г. Тихонов В.И., Шахтарин Б.И., Сизых В.В. Случайные процессы. Примеры и задачи. Том 5. Оценка сигналов, их параметров и спектров. Основы теории информации. Учебное пособие для вузов. М.: Горячая линия - Телеком, 2012.
- г. Гмурман В.Е. Руководство к решению теории вероятностей и математическая статистика: учебное пособие. М.: Юрайт 2009
- г. Ефименко В.Н., Прушинская Л.А. Теория случайных процессов. Учебное пособие. Ростов н/Д, СКФ МТУСИ, 2006.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Практическое занятие № 3**

**Вычисление аналитических характеристик на выходе интегратора.**

**1. Цель занятия:**

Познакомить студентов с понятием интеграла СП. Выработать умения и навыки по вычислению аналитических характеристик на выходе интегратора.

**2. Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников: (1) с. 409-413, (3) с. 11-17

**3. Задание:**

Решить примеры: (2) №№ 811-829 (выборочно, по рекомендации преподавателя).

**4. Порядок выполнения:**

- 4.1. Записать определение интеграла СП.
- 4.2. Записать формулу для вычисления математического ожидания интеграла СП.
- 4.3. Записать формулу для вычисления корреляционной функции интеграла СП.
- 4.4. Записать формулу для определения взаимной корреляционной функции СП и его интеграла.
- 4.5. Решить примеры, см. п.3.

**5. Отчет:**

- 5.1. Определение интеграла случайного процесса.
- 5.2. Формула для вычисления математического ожидания интеграла СП.
- 5.3. Формула для вычисления корреляционной функции интеграла СП.
- 5.4. Формула для определения взаимной корреляционной функции СП и его интеграла.
- 5.5. Решенные примеры.

**6. Список литературы:**

- в Тихонов В.И., Шахтарин Б.И., Сизых В.В. Случайные процессы. Примеры и задачи. Том 5. Оценка сигналов, их параметров и спектров. Основы теории информации. Учебное пособие для вузов. М.: Горячая линия - Телеком, 2012.
- в Гмурман В.Е. Руководство к решению теории вероятностей и математическая статистика: учебное пособие. М.: Юрайт 2009
- в Ефименко В.Н., Прушинская Л.А. Теория случайных процессов. Учебное пособие. Ростов н/Д, СКФ МТУСИ, 2006.

**Практическое занятие № 4**

**Контрольная работа №1**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Практическое занятие № 5**

**СМО с отказами**

**1. Цель занятия:**

Научить студентов строить граф состояний СМО с отказами. Выработать умения и навыки по вычислению предельных вероятностей состояний и показателей эффективности СМО с отказами.

**2. Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников: (4) с. 363-366

**3. Задание:**

Решить примеры: (4) №№ 11.1-11.3 (выборочно, по рекомендации преподавателя).

**4. Порядок выполнения:**

- 4.1. Записать определение СМО с отказами.
- 4.2. Построить граф состояний СМО с отказами.
- 4.3. Записать формулы Эрланга.
- 4.4. Записать формулы для вычисления показателей эффективности СМО с отказами.
- 4.5. Решить примеры, см. п.3.

**5. Отчет:**

- 5.1. Определение СМО с отказами.
- 5.2. Граф состояний СМО с отказами.
- 5.3. Формулы Эрланга.
- 5.4. Формулы для вычисления показателей эффективности СМО с отказами.
- 5.5. Решенные примеры.

**6. Список литературы:**

- в Тихонов В.И., Шахтарин Б.И., Сизых В.В. Случайные процессы. Примеры и задачи. Том 5. Оценка сигналов, их параметров и спектров. Основы теории информации. Учебное пособие для вузов. М.: Горячая линия - Телеком, 2012.
- в Гмурман В.Е. Руководство к решению теории вероятностей и математическая статистика: учебное пособие. М.: Юрайт 2009
- в Ефименко В.Н., Прушинская Л.А. Теория случайных процессов. Учебное пособие. Ростов н/Д, СКФ МТУСИ, 2006.



## Практическое занятие № 6

### Одноканальные и многоканальные СМО с неограниченной очередью

#### 1. Цель занятия:

Научить студентов строить граф состояний одноканальных и многоканальных СМО с неограниченной очередью. Выработать умения и навыки по вычислению предельных вероятностей состояний и показателей эффективности данных СМО.

#### 2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: (4) с. 366-367

#### 3. Задание:

Решить примеры: (4) №№ 11.4-11.6 (выборочно, по рекомендации преподавателя).

#### 4. Порядок выполнения:

- 4.1. Записать определение одноканальной СМО с неограниченной очередью.
- 4.2. Построить граф состояний одноканальной СМО с неограниченной очередью.
- 4.3. Записать формулы Литтла.
- 4.4. Записать формулы для вычисления показателей эффективности одноканальной СМО с неограниченной очередью.
- 4.5. Записать определение многоканальной СМО с неограниченной очередью.
- 4.6. Построить граф состояний многоканальной СМО с неограниченной очередью.
- 4.7. Записать формулы Литтла.
- 4.8. Записать формулы для вычисления показателей эффективности многоканальной СМО с неограниченной очередью.
- 4.9. Решить примеры, см. п.3.

#### 5. Отчет:

- 5.1. Определение одноканальной СМО с неограниченной очередью.
- 5.2. Граф состояний одноканальной СМО с неограниченной очередью.
- 5.3. Формулы Литтла.
- 5.4. Формулы для вычисления показателей эффективности одноканальной СМО с неограниченной очередью.
- 5.5. Определение многоканальной СМО с неограниченной очередью.
- 5.6. Граф состояний многоканальной СМО с неограниченной очередью.
- 5.7. Формулы Литтла.
- 5.8. Формулы для вычисления показателей эффективности многоканальной СМО с неограниченной очередью.
- 5.9. Решенные примеры.

#### 6. Список литературы:

- Тихонов В.И., Шахтарин Б.И., Сизых В.В. Случайные процессы. Примеры и задачи. Том 5. Оценка сигналов, их параметров и спектров. Основы теории информации. Учебное пособие для вузов. М.: Горячая линия - Телеком, 2012.
- Гмурман В.Е. Руководство к решению теории вероятностей и математическая статистика: учебное пособие. М.: Юрайт 2009
- Ефименко В.Н., Прушинская Л.А. Теория случайных процессов. Учебное пособие. Ростов н/Д, СКФ МТУСИ, 2006.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Практическое занятие № 7**

**СМО с ограниченной очередью.**

**1. Цель занятия:**

Научить студентов строить граф состояний СМО с ограниченной очередью. Выработать умения и навыки по вычислению предельных вероятностей состояний и показателей эффективности данной СМО.

**2. Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников: (4) с. 367-370

**3. Задание:**

Решить примеры: (4) №№ 11.14-11.23 (выборочно, по рекомендации преподавателя).

**4. Порядок выполнения:**

- 4.1. Записать определение СМО с ограниченной очередью.
- 4.2. Построить граф состояний СМО с ограниченной очередью.
- 4.3. Записать формулы Литтла.
- 4.4. Записать формулы для вычисления показателей эффективности СМО с ограниченной очередью.
- 4.5. Решить примеры, см. п.3.

**5. Отчет:**

- 5.1. Определение СМО с ограниченной очередью.
- 5.2. Граф состояний СМО с ограниченной очередью.
- 5.3. Формулы Литтла.
- 5.4. Формулы для вычисления показателей эффективности СМО с ограниченной очередью.
- 5.5. Решенные примеры.

**6. Список литературы:**

в Тихонов В.И., Шахтарин Б.И., Сизых В.В. Случайные процессы. Примеры и задачи. Том 5. Оценка сигналов, их параметров и спектров. Основы теории информации. Учебное пособие для вузов. М.: Горячая линия - Телеком, 2012.

в Гмурман В.Е. Руководство к решению теории вероятностей и математическая статистика: учебное пособие. М.: Юрайт 2009

в Ефименко В.Н., Прушинская Л.А. Теория случайных процессов. Учебное пособие. Ростов н/Д, СКФ МТУСИ, 2006.

**Практическое занятие № 8**

**Контрольная работа №2**



## Заочная форма обучения

### Практическое занятие № 1

#### Вычисление вероятностных характеристик случайных процессов

##### 1. Цель занятия:

Выработать умения и навыки по вычислению вероятностных характеристик случайных процессов (СП).

##### 2. Краткие теоретические сведения:

Справочный материал и разобранные примеры приведены в любом из следующих источников: (1) с. 386-399, (3) с. 4-7

##### 3. Задание:

Решить примеры: (2) №№ 756-778 (выборочно, по рекомендации преподавателя).

##### 4. Порядок выполнения:

- 4.1. Записать определение случайного процесса, его сечения и реализации.
- 4.2. Дать определение математического ожидания СП, перечислить его свойства.
- 4.3. Записать определение корреляционной функции СП, перечислить ее свойства.
- 4.4. Записать формулу, связывающую дисперсию и корреляционную функцию СП.
- 4.5. Записать формулу для вычисления нормированной корреляционной функции.
- 4.6. Решить примеры, см. п.3.

##### 5. Отчет:

- 5.1. Определение случайного процесса, его сечения и реализации.
- 5.2. Определение математического ожидания СП, его свойства.
- 5.3. Определение корреляционной функции СП, ее свойства.
- 5.4. Формула, связывающая дисперсию и корреляционную функцию СП.
- 5.5. Формула для вычисления нормированной корреляционной функции.
- 5.6. Решенные примеры.

##### 6. Список литературы:

- Тихонов В.И., Шахтарин Б.И., Сизых В.В. Случайные процессы. Примеры и задачи. Том 5. Оценка сигналов, их параметров и спектров. Основы теории информации. Учебное пособие для вузов. М.: Горячая линия - Телеком, 2012.
- Гмурман В.Е. Руководство к решению теории вероятностей и математическая статистика: учебное пособие. М.: Юрайт 2009
- Ефименко В.Н., Прушинская Л.А. Теория случайных процессов. Учебное пособие. Ростов н/Д, СКФ МТУСИ, 2006.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Практическое занятие № 2**

**Вычисление предельных вероятностей состояний**

**1. Цель занятия:**

Научить студентов строить граф состояний СМО. Выработать умения и навыки по вычислению предельных вероятностей состояний.

**2. Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников: (4) с. 317-328, (3) с. 33-38

**3. Задание:**

Решить примеры: (4) №№ 10.1-10.30 (выборочно, по рекомендации преподавателя).

**4. Порядок выполнения:**

- 4.1. Записать определение системы массового обслуживания (СМО).
- 4.2. Записать определение графа состояний СМО.
- 4.3. Изучить правила построения графа состояний СМО.
- 4.4. Записать правило составления системы уравнений Колмогорова.
- 4.5. Решить примеры, см. п.3.

**5. Отчет:**

- 5.1. Определение СМО.
- 5.2. Определение графа состояний СМО.
- 5.3. Правила построения графа состояний СМО.
- 5.4. Правило составления системы уравнений Колмогорова.
- 5.5. Решенные примеры.

**6. Список литературы:**

в Тихонов В.И., Шахтарин Б.И., Сизых В.В. Случайные процессы. Примеры и задачи. Том 5. Оценка сигналов, их параметров и спектров. Основы теории информации. Учебное пособие для вузов. М.: Горячая линия - Телеком, 2012.

в Гмурман В.Е. Руководство к решению теории вероятностей и математическая статистика: учебное пособие. М.: Юрайт 2009

в Ефименко В.Н., Прушинская Л.А. Теория случайных процессов. Учебное пособие. Ростов н/Д, СКФ МТУСИ, 2006.

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ для проведения  
практических занятий по дисциплине «Периферийные  
устройства и интерфейсы»**

**ПРАКТИЧЕСКАЯ РАБОТА № 1.**

**Перевод чисел. Логические элементы. Элементы и узлы  
ЭВМ**



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

## 1.1. КРАТКАЯ ТЕОРИЯ

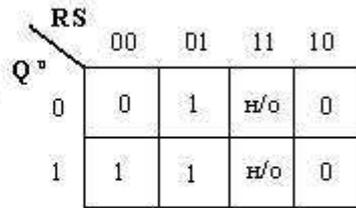
Триггер представляет собой электронное устройство с двумя состояниями устойчивого равновесия, что и делает его удобным для хранения информации в двоичной системе счисления. Бистабильный триггер имеет два выхода, причём информация на одном из выходов  $\bar{Q}$  представляет собой инверсию информации на другом выходе  $Q$  (  $\bar{\quad}$  - знак инверсии логической переменной ). Запись информации в триггер обычно осуществляют через схему управления.

Триггеры классифицируют по ряду признаков. По функциональным возможностям выделяют:

- а) триггер с раздельной установкой 0 и 1 (RS-триггер);
- б) триггер с приёмом информации по одному входу (D-триггер), другое название – триггер задержки;
- в) триггер со счётным входом (T-триггер);
- г) универсальный триггер (JK-триггер).

Таблица 1

| Такт n |   |   | Такт n+1  | Режим        |
|--------|---|---|-----------|--------------|
| $Q^n$  | R | S | $Q^{n+1}$ |              |
| $Q^n$  | 1 | 0 | 0         | Reset-уст. 0 |
| $Q^n$  | 0 | 1 | 1         | Set-уст. 1   |
| $Q^n$  | 0 | 0 | $Q^n$     | хранение     |
| $Q^n$  | 1 | 1 | н/о       | запрет       |



а)

б)

Рис.1. Таблица истинности (а) и карта Карно (б) для триггера как элемента памяти

синхронизируемые фронтом (реагируют на информационные сигналы в момент изменения сигнала на С-входе от 0 к 1 – положительный фронт или от 1 к 0 – отрицательный фронт).

Основой любого триггера является элемент памяти. Чаще всего в статических триггерах эту роль выполняет бистабильная ячейка памяти (простейший асинхронный RS-триггер), которая функционирует в соответствии с таблицей истинности, приведённой на рис. 1,а. Исходное состояние триггера характеризуется сигналом на прямом выходе  $Q^n$ , а новое состояние  $Q^{n+1}$  определяется комбинацией входных сигналов (н/о- не определённое состояние). Например, для установки триггера в состояние 0 необходимо на его входы подать такую комбинацию сигналов, при которой на прямом выходе сигнал будет иметь уровень логического 0, т. е.  $Q^{n+1}=0$ , а  $\bar{Q}^{n+1}=1$ .

Из таблицы истинности и карты Карно следует, что поведение такого элемента памяти описывается следующим характеристическим уравнением:

$$Q^{n+1} = S \vee \bar{R} * Q^n \dots (1).$$

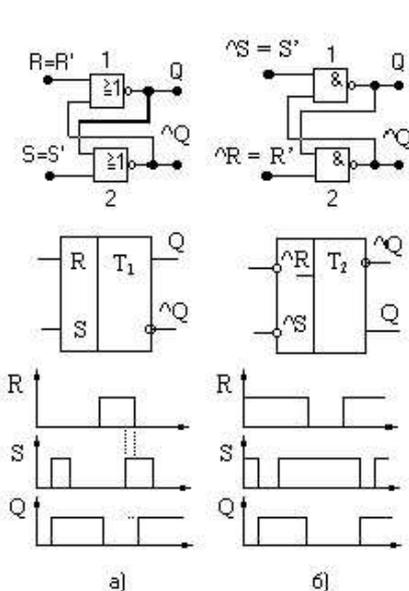


Рис.2. Функциональные схемы, условные обозначения и временные диаграммы RS-триггера на ИЛИ-НЕ (а) и И-НЕ (б)

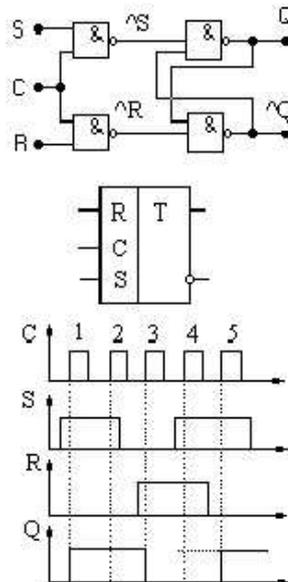


Рис.3. Функциональная схема тактируемого RS-триггера, условное обозначение и временные диаграммы

По способу приёма информации триггеры подразделяют на асинхронные (неактивируемые) и синхронные (активируемые). Первые реагируют на информационные сигналы в момент их появления на входах триггера. Вторые реагируют на информационные сигналы при наличии сигнала синхронизации на специальном входе С. Синхронные триггеры подразделяют на триггеры, синхронизируемые импульсом (реагируют при подаче на вход С уровня

1 для прямого С-входа или 0 для инверсного С-входа), и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и

и



схемы, условные обозначения и временные диаграммы **асинхронных RS-триггеров**, построенных на ИЛИ-НЕ (рис. 2,а) и на И-НЕ (рис. 2,б).

Для RS-триггера на ЛЭ ИЛИ-НЕ активным является уровень логической 1. По временной диаграмме (рис. 2,а) видно, что единичный сигнал на S-входе устанавливает триггер в состояние "1" (на его выходе Q=1). По окончании действия сигнала S=1 на входах триггера сигналы S= R=0 (режим хранения). Сигнал R=1 устанавливает триггер в состояние "0" (Q=0). Пунктиром отмечен интервал времени одновременного появления на R- и S-входах логической 1, что является запрещённым состоянием для триггера этого типа. Состояние выходного сигнала Q в этот интервал времени не определено (пунктирная линия). После сброса до 0 сигнала на R-входе Q=1 за счёт действия логической 1 на S-входе. Прекращение действия сигнала на S-входе и состояние S= R=0 устанавливает режим хранения для Q=1. Для RS-триггера на ЛЭ И-НЕ активным является уровень логического 0. По временной диаграмме (рис. 2,б) видно, что состояние S=R=1 обеспечивает режим хранения, поддерживая Q=0. Логический 0 на S-входе устанавливает Q=1, сохраняющийся и при возвращении S-входа к уровню 1. Отрицательный импульс на R-входе устанавливает Q=0. Затем в течение короткого интервала времени R=S=1 (режим хранения Q=0), и установка 1 (при S=0, Q=1), что и сохраняется в дальнейшем при R= S=1.

Большее применение находит **тактируемый RS-триггер**, переключение которого происходит при наличии импульса синхронизации C. Поэтому его характеристическое уравнение:  $Q^{n+1} = C*(S \vee R*Q^n) \vee \wedge C*Q^n \dots (4)$ . Для построения функциональной схемы устройства, реализующего данное уравнение, применим по аналогии с (2) и двойное инвертирование уравнения (4) и формулы де-Моргана. Тогда имеем  $Q^{n+1} = \wedge \wedge (C*S \vee C*\wedge R*Q^n \vee \wedge C*Q^n) = \wedge \wedge (C*S \vee Q^n*(C*\wedge R \vee \wedge C)) = \wedge \wedge (C*S \vee Q^n*(\wedge R \vee \wedge C)) = \wedge \wedge (\wedge (C*S)*\wedge(Q^n \wedge (R*C))) \dots (5)$ . Это уравнение определяет функциональную схему RS-триггера, синхронизируемого импульсом C (рис.3). Здесь же приведены его условное обозначение и временные диаграммы работы.

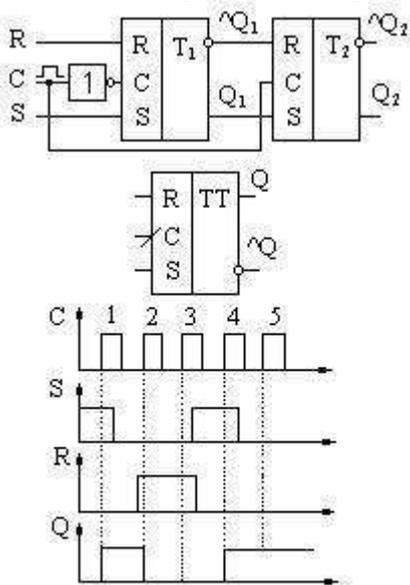


Рис.4. Функциональная схема, условное обозначение и временные диаграммы синхронизируемого передним фронтом RS-триггера

По временным диаграммам рис. 3 видно, что появление высокого уровня Q выходного сигнала триггера происходит при наличии первого импульса на C-входе (S=1). Это состояние сохраняется и после окончания действия импульса на S-входе за счёт второго тактового импульса C-входа. Установка триггера в «0» (R=1) происходит только при появлении третьего синхроимпульса, и состояние Q=0 поддерживается в течение времени действия третьего синхроимпульса и паузы между третьим и четвёртым синхроимпульсами. Одновременное действие S=R=1 при четвёртом синхроимпульсе и последующей паузе является запрещённым состоянием, поэтому выходной сигнал не определён (пунктирная линия) и установление Q=1 за счёт S=1 происходит только при пятом синхроимпульсе. Далее это состояние Q=1 сохраняется (режим хранения S=R=0).

Недостатком синхронизируемого импульсом RS-триггера является возможность изменения его состояния в течение всего времени действия синхроимпульса. Это обстоятельство устранено в RS-триггере, синхронизируемом фронтом импульса C.

На рис.4 приведены функциональная схема RS-триггера, синхронизируемого передним фронтом импульса, его условное обозначение и временные диаграммы работы. Синхроимпульс подаётся на C-вход триггера T1 через инвертор. Поэтому при C=0 T1 реагирует на изменения сигналов на входах R и S, нужным образом устанавливая Q1 и Q1-bar. В момент изменения синхроимпульса C от 0 к 1 T1 перестаёт реагировать на изменения R и S, а его последнее состояние Q1 и Q1-bar запоминается на T2, так как синхроимпульс на его C-входе равен 1. Далее при C=1 T1 не реагирует на изменения R и S, Q1 и Q1-bar постоянны. Второй триггер T2 сохраняет свое состояние.

Двухступенчатое запоминание информации в этом триггере отображается двумя буквами T в условном его обозначении. Реакция схемы на передний фронт идентифицируется отрезком под углом 45° на линии ввода сигнала в C-вход.

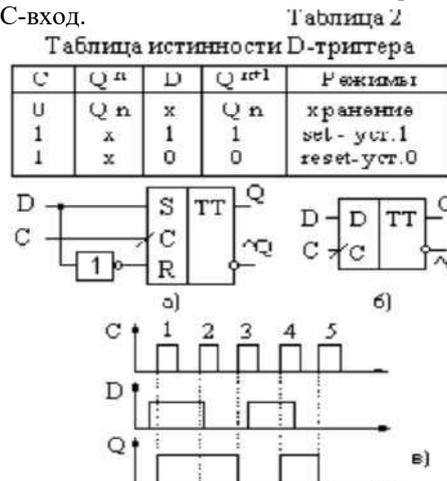


Рис.6. Функциональная схема (а), услов-

Временные диаграммы работы триггера на рис. 4 показывают переход TT в единичное состояние за счёт S=1 по переднему фронту первого синхроимпульса и возвращение Q=0 по переднему фронту второго синхроимпульса ввиду R=1. Четвёртый импульс вновь устанавливает Q=1 за счёт S=1, и далее при S= R=0 реализуется режим хранения состояния Q=1. Такое схемотехническое построение триггера существенно повышает его устойчивость к воздействию различных помех.

Аналогично рис. 4 может быть построена схема RS -триггера, синхронизируемого задним фронтом синхроимпульса. Для этого инвертор должен быть включён в цепь на C-вход второго триггера. На условном обозначении управляемого задним фронтом RS -триггера отрезок на линии C-входа размещается под углом в 135° (рис. 5).



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**D-триггер** имеет один информационный вход (D-вход) и вход для синхронизирующего импульса. Его функционирование реализуется в соответствии с таблицей истинности (табл. 2), из которой можно записать



характеристическое уравнение для D-триггера:  $Q^{n+1} = C \cdot D \vee \neg C \cdot Q^n$  ... (6). D-триггер строится на базе RS-триггера. Если у RS-триггера, работающего согласно уравнению  $Q^{n+1} = C \cdot (S \vee \neg R) \cdot Q^n \vee \neg C \cdot Q^n$  на вход S подать значение D ( $S=D$ ), а на вход R подать значение  $\neg D$  ( $R=\neg D$ ), то получим характеристическое уравнение D-триггера:  $Q^{n+1} = C \cdot (D \vee \neg D) \cdot Q^n \vee \neg C \cdot Q^n = C \cdot D \vee \neg C \cdot Q^n$  (рис. 6,а). Здесь же на рис. 6 показано условное обозначение D-триггера (б) и временные диаграммы его работы (в). Синхронизация D-триггера производится по переднему (для рис. 6) или заднему фронту тактирующего импульса.

**Т-триггер** – это триггер с одним так называемым счётным входом, изменяющий своё состояние с приходом каждого входного импульса. Функционирование триггера должно идти в соответствии с таблицей истинности табл.3, на основании которой можно записать характеристическое уравнение в виде:  $Q^{n+1} = C \cdot Q^n \vee \neg C \cdot \neg Q^n$  ... (8), а на основании последнего получить Т- триггер из D-триггера, положив  $D = \neg Q^n$ . Функциональная схема Т-триггера показана на рис. 7.

Большими логическими возможностями обладает TV-триггер, имеющий два логических входа Т и V, первый из которых является счётным, а второй – разрешающим. TV триггер при V=1 выполняет функции Т-триггера; если на входе V действует запрещающий сигнал 0, состояние триггера не меняется. Характеристическое уравнение,

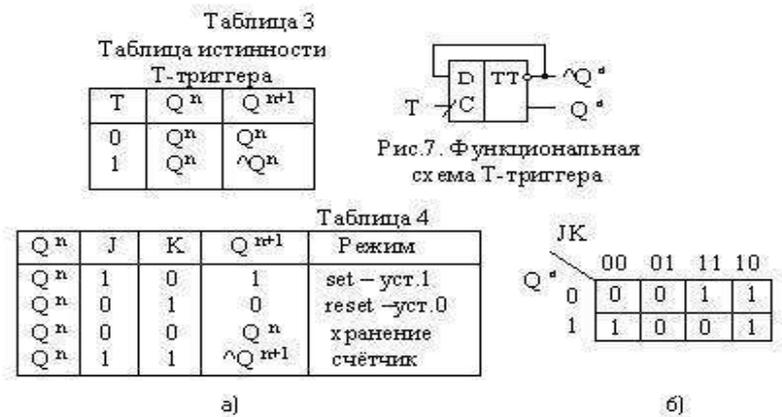


Рис.8. Таблица истинности (а) и карта Карно (б) для JK-триггера при C=1

описывающее логические возможности TV-триггера, определяется соотношением:  $Q^{n+1} = Q^n (\neg T \vee \neg V) \vee \neg Q^n T \vee V$ .

**JK-триггер** имеет два информационных входа: J и K, а так же вход для тактовых импульсов C. Правило работы JK-триггера определяется его таблицей истинности табл. 4 и картой Карно (рис. 8), на основании которых можно записать следующее характеристическое уравнение:  $Q^{n+1} = C \cdot (J \cdot \neg Q^n \vee \neg K \cdot Q^n) \vee \neg C \cdot Q^n$ , на основании чего можно составить функциональную схему JK-триггера (рис. 9,а). Этот триггер строится на базе RS-триггера введением обратных связей с прямого и инверсного выходов на два входных элемента И, объединяющие сигналы обратной связи с сигналами управления J и K. На рис. 9,б приведено условное обозначение триггера этого типа, временные диаграммы (в) иллюстрируют его работу. В момент совместного действия логических 1 на входах J и K передний фронт третьего синхроимпульса вызывает срабатывание триггера в счётном режиме, изменяя Q с величины 0 на значение 1.

При отсутствии синхроимпульсов, т.е. C=0 JK-триггер работает в режиме хранения. На базе JK-триггера можно получить схемы всех остальных триггеров, что и приведено на рис. 10.

## 1. 2. МЕТОДИКА ВЫПОЛНЕНИЯ

2.1. Включите необходимое программное обеспечение.

2.2. Соберите на монтажном столе схему асинхронного статического триггера по рис. 2,б. Убедитесь в её работоспособности, спланируйте эксперимент и результаты запишите в виде таблицы истинности  $Q^{n+1} = f(S, R, Q^n)$ . Логические переменные R и S на входах задавайте с помощью переключателя кнопочного на два положения из контейнера **Control**, подключённые одним выводом к источнику +5 вольт, а другим выводом – к земле. Индикацию состояния выходов триггера обеспечьте индикаторами из контейнера **Indicators**. Составьте характеристическое уравнение RS-триггера.

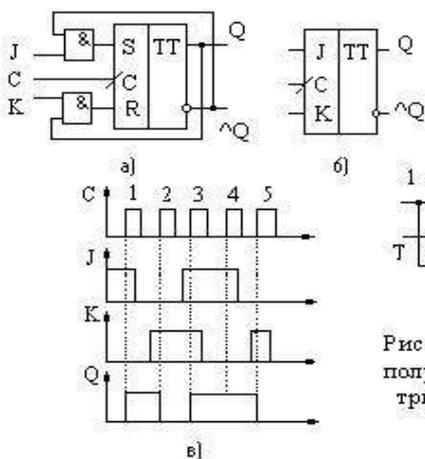


Рис.9. Функциональная схема (а), условное обозначение (б) и временные диаграммы (в)

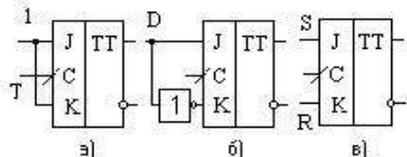


Рис.10. Использование JK-триггера для получения триггеров других типов: а) Т-триггер; б) D-триггер; в) JK-триггер



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

2.3. При исследовании схем с триггерами перед началом эксперимента необходимо установить триггера в



определённое состояние (например, в логический 0), так как моделирующая программа не может разрешить противоречие: неизвестны значения  $Q^n$  и  $\bar{Q}^n$ , а  $S=R=0$ , т.е. задан режим хранения, а состояние  $Q^n$  – неизвестно. Поэтому во всех схемах этой работы первым шагом моделирования должна быть установка триггеров в определённое состояние (логический 0). Для этого в схему синхронизируемого импульсом RS-триггера на рис. 3 в логическом элементе 4 введён переключатель с задержками во времени из контейнера **Control**, у которого Time On =100 нс, а Time Off=10000 с. (рис.11). Этот переключатель в момент начала моделирования включается и подаёт логический 0 на дополнительный вход статического элемента памяти и устанавливает его в состояние логического 0, а через 100 нс отключается и по-даёт на этот вход логическую 1. С этого момента времени элемент памяти может нормально функционировать в моделирующей программе.

Соберите схему синхронизируемого импульсом RS-триггера по рис. 11,а. Убедитесь в её работоспособности, спланируйте эксперимент и результаты запишите в виде таблицы истинности  $Q^{n+1}=f(S,R,Q^n)$  при  $C=1$ . Составьте характеристическое уравнение. Убедитесь, что в начале моделирования триггер устанавливается в состояние 0. Поместите эту схему в подсхему RST\_CI (рис.11,б).

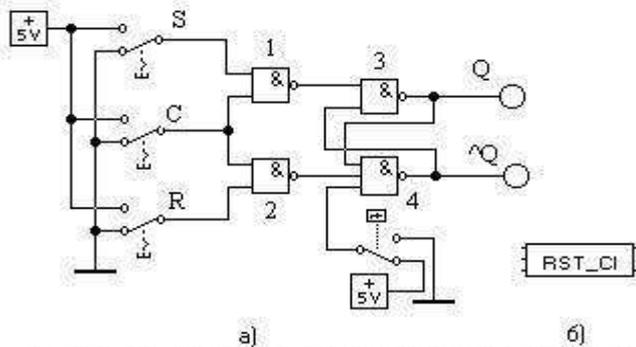


Рис.11. RS-триггер, синхронизируемый импульсом C, (а) и подсхема RST-CI на его основе (б)

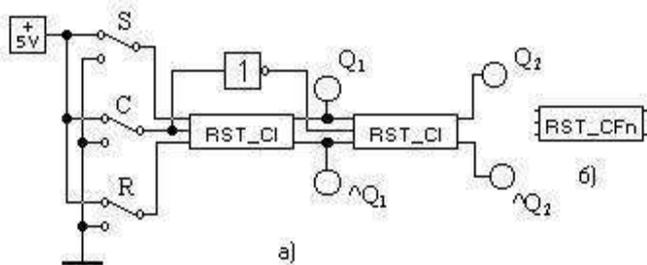


Рис.12. RS-триггер, синхронизируемый отрицательным фронтом импульса, (а) и его подсхема (б)

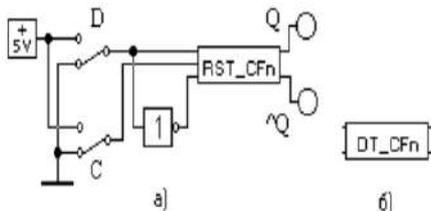


Рис.13. D-триггер, синхронизируемый отрицательным фронтом импульса, (а) и его подсхема (б)

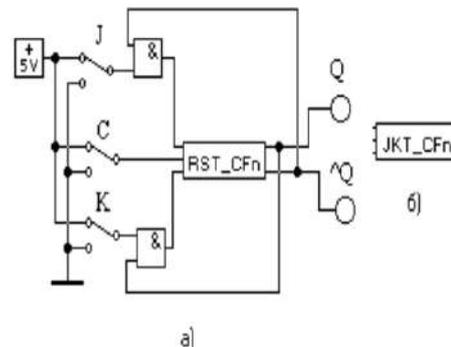


Рис.14. JK-триггер, синхронизируемый отрицательным фронтом импульса, (а) и его подсхема (б)

2.4. Используя подсхему RST\_CI, соберите RS-триггер, синхронизируемый отрицательным фронтом импульса синхронизации по рис.12,а. Спланируйте эксперимент по исследованию схемы. С помощью логического анализатора из моделирующей программы продемонстрируйте преподавателю временные диаграммы зависимостей  $Q_1$  и  $Q_2$  от R, S и C. Поместите RS-триггер с синхронизацией от отрицательного фронта импульса в подсхему RST\_CFn (рис.12,б), убрав предварительно индикацию состояний  $Q_1, \bar{Q}_1$ .

2.5. Соберите схему D-триггера по рис. 13,а. Убедитесь в её работоспособности, спланируйте эксперимент и результаты запишите в виде таблицы истинности  $Q^{n+1} = f(D,C,Q^n)$ . Составьте характеристическое уравнение для D-триггера. Поместите схему в подсхему DT\_CFn (рис.13,б).

2.6. Соберите схему JK-триггера по рис. 14,а. Убедитесь в её работоспособности, спланируйте эксперимент и результаты запишите в виде таблицы истинности  $Q^{n+1} = f(J,K,C,Q^n)$ . Составьте характеристическое уравнение для JK-триггера. Поместите эту схему в подсхему JKT\_CFn (рис.14,б).

Для использования созданных подсхем в новых схемах необходимо поместить их в контейнер пользователя файла **default.ca4** (через буфер обмена) и сохранить. Теперь при создании новой схемы следует выбирать существующий файл **default.ca4**, в котором будет появляться монтажный стол с контейнером, содержащим созданные вами подсхемы.

### 1.3. КОНТРОЛЬНЫЕ ВОПРОСЫ

- \* Как классифицируются триггеры по функциональным возможностям?
- \* На базе каких логических элементов может строиться асинхронный RS-триггер? Какие уровни логической переменной являются активными для каждого из них?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

\* Почему возникла необходимость введения тактируемых триггеров? Поясните по временной диаграмме рис. 3 работу такого триггера.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

7

- В Поясните по временной диаграмме рис. 4 работу синхронизируемого передним фронтом RS–триггера.  
В В чём функциональное отличие между RS- и JK–триггерами, поясните работу последнего по временной диаграмме рис. 9.  
В Покажите схемные решения использования JK–триггера для построения триггеров всех других видов.

Контрольные вопросы ПЗ1 (ПК-4):

- 2 Что такое система счисления?
- 2 Что такое основание системы счисления?
- 2 Как зависит разрядность чисел от величины основания системы счисления?
- 2 Почему информация в компьютере представляется в двоичной системе счисления?
- 2 Какие системы счисления используются для представления информации в компьютере?
- 2 Для чего применяются двоично-десятичные коды?
- 2 Чем отличаются формы представления чисел с фиксированной и плавающей точками?
- 2 Как кодируются знаки чисел?
- 2 Можно ли различить форматы целых и дробных чисел с фиксированной точкой?
- 2 Какие элементы формата чисел с плавающей точкой вы знаете?
- 2 Чем отличается нормализованное число с плавающей точкой от ненормализованного числа?
- 2 Для чего применяются смещенные порядки?
- 2 Какой прием используется для расширения диапазона чисел с плавающей точкой?
- 2 В чем заключается способ скрытой единицы?
- 2 Охарактеризуйте форматы двоично-десятичных чисел.
- 2 Как кодируются знаки двоично-десятичных чисел?
- 2 Для чего используются машинные коды чисел?
- 2 Чем отличаются дополнительный и обратный коды отрицательных чисел?
- 2 Как из дополнительного кода отрицательного числа получить прямой код числа?
- 2 Каким образом кодируется текстовая информация?
- 2 Перечислите составные части компьютера в порядке уменьшения функциональной сложности.
- 2 Чем различаются узлы комбинационного и накапливающего типов?
- 2 Что такое логический элемент?
- 2 Что такое функционально полная система логических элементов?
- 2 Что такое состояние триггера?
- 2 Почему триггер имеет два устойчивых состояния?
- 2 Чем различаются асинхронный и синхронный триггеры?
- 2 Что такое таблица переходов триггера?
- 2 В каких случаях используются двухтактные триггеры?
- 2 Чем различаются триггеры разных типов?
- 2 Почему асинхронный D-триггер не используют в качестве элемента памяти?
- 2 Какие сигналы нужно подать на входы синхронного RS-триггера, чтобы записать в него «1»?
- 2 Что такое дешифратор?
- 2 Сколько входов и выходов имеет одноразрядный сумматор?
- 2 Чем различаются сумматоры с последовательными, сквозными и параллельными переносами?
- 2 Почему для сложения тетрад двоично-десятичных чисел нельзя использовать обычный двоичный сумматор?
- 2 Для чего используют мультиплексоры?
- 2 Из каких элементов состоит регистр?
- 2 Почему для сдвига данных используют двухтактные триггеры?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

2 В чем состоит отличие схем суммирующего и вычитающего счетчиков?



## ПРАКТИЧЕСКАЯ РАБОТА № 2.

### Способы адресации. Микрооперации и микропрограммы

#### 2.1. КРАТКАЯ ТЕОРИЯ

Совокупность из нескольких триггеров, предназначенных для хранения двоичного кода, называется регистром. Введением дополнительной логики функциональные возможности регистров расширяются, обеспечивая возможность сдвига записанной информации в ту или иную сторону.

В соответствии с назначением различают регистры хранения, регистры сдвига и универсальные регистры.

По принципу хранения информации регистры делятся на статические и динамические. Статические регистры строят на потенциальных элементах памяти (триггерах), которые могут хранить записанную информацию сколь угодно долго (при наличии напряжения питания). Динамические регистры строят на элементах памяти такого типа, как конденсатор, которые могут хранить информацию лишь в течение некоторого промежутка времени. Поэтому в динамических регистрах необходима регенерация записанной информации.

Важнейшие характеристики регистров – разрядность и быстродействие. Разрядность определяется количеством триггеров. Быстродействие характеризуется максимальной тактовой частотой, с которой может производиться запись, чтение и сдвиг информации.

Наиболее широко распространены регистры сдвига, предназначенные для преобразования информации путём её сдвига под воздействием тактовых импульсов. Такие регистры представляют собой совокупность последовательно соединённых триггеров, как правило, двухступенчатой структуры. По направлению сдвига информации различают регистры прямого сдвига (вправо, т.е. в сторону младшего разряда), обратного сдвига (влево, т.е. в сторону старшего разряда) и реверсивные, допускающие сдвиг в обоих направлениях.

В простейшем регистре триггеры соединены последовательно: выходы  $Q$  и  $\bar{Q}$  предыдущего триггера передают бит данных на входы  $R$  и  $S$  последующего. Все тактовые входы  $C$  триггеров соединены параллельно. При таком включении единица, записанная в виде напряжений низкого и высокого уровней по входам  $R$  и  $S$  первого триггера, после подачи одного тактового импульса перейдёт во второй триггер, затем во время следующего тактового импульса она попадёт в третий триггер и так последует далее, до конца регистра.

Аналогично продвигается по регистру многобитное слово: оно поразрядно вводится на входы  $R$  и  $S$  первого триггера. Простейший регистр имеет один вход и один выход – последовательные. Вход управления также единственный – тактовый. Если ко входу каждого триггера добавить разрешающую логику, можно получить дополнительные, так называемые параллельные входы одновременной загрузки байта в регистр. Здесь, как правило, используют дополнительные защёлки, где фиксируются данные, поступившие на входы после прихода тактового импульса. В такую схему добавляется вход разрешения записи.

Можно предусмотреть также логическую схему параллельного отображения на выходе состояния каждого триггера. Тогда после заполнения регистра от последовательного или параллельного входов по команде разрешения выхода накопленное цифровое слово можно отобразить поразрядно сразу на всех параллельных выходах. Для удобства поочерёдной выдачи данных от таких регистров – буферных накопителей в шину данных обрабатывающего устройства – процессора – параллельные выходы регистров снабжаются выходными буферными усилителями, имеющими третье, разомкнутое  $Z$ -состояние. По многопроводной шине данных процессор получает цифровое слово – байт от выходов того регистра, которому дана команда разрешения выдачи.

Наиболее широко распространены регистры сдвига на  $D$ -триггерах (рис. 1). Такие регистры имеют один информационный вход, вход для импульсов синхронизации (импульсов сдвига) и установочный вход  $R$ . Выходы в регистре могут быть с каждого разряда для считывания информации в параллельном коде. Также имеется один выход с последнего (относительно входа) разряда для считывания информации последовательно во времени, т.е. последовательным кодом. Вход регистра для импульсов сдвига получается объединением  $C$ -входов всех триггеров, а установочный вход –  $R$ -входов.

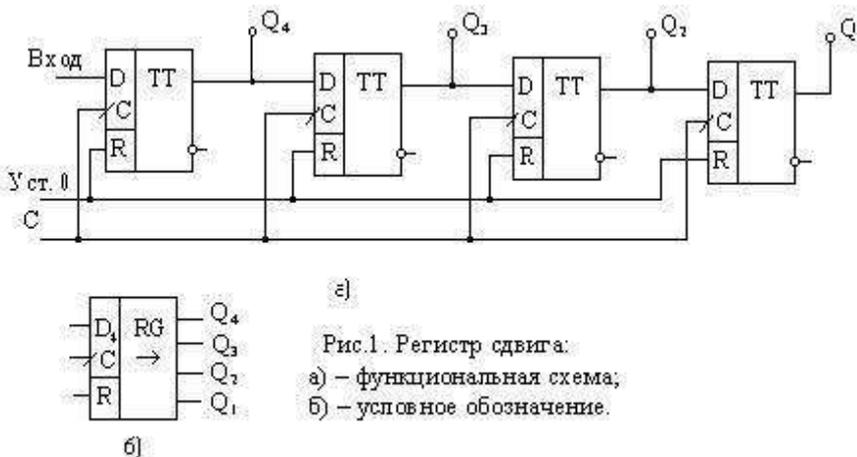


Рис.1. Регистр сдвига:  
а) – функциональная схема;  
б) – условное обозначение.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Перед записью информации регистр устанавливается в нулевое состояние подачей положительного импульса по шине «Уст. 0». Записываемая информация должна быть представлена последовательным кодом. Запись осуществляется поразрядно со стороны старшего (рис. 1) или младшего разряда (направление сдвига указывается



стрелкой на условном обозначении регистра) путём продвижения кодовой комбинации с каждым тактовым импульсом от разряда к разряду. Следовательно, для записи  $N$ -разрядного слова необходимы  $N$ -импульсов сдвига.

Считывание информации последовательным кодом осуществляется, как и запись, поразрядным сдвигом записанной кодовой комбинации к выходу с каждым тактовым импульсом. Следовательно, для считывания  $N$ -разрядного слова необходимы  $N$  импульсов сдвига. Считывание информации параллельным кодом происходит в паузе между последним импульсом сдвига одного цикла записи и первым импульсом сдвига другого цикла записи, т.е. в интервале времени, когда на  $S$ -входах триггеров нулевой уровень, и они находятся в режиме хранения.

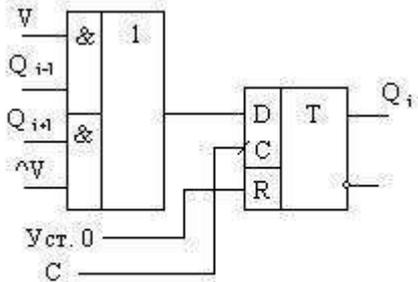


Рис.2. Разряд реверсивного регистра

Таким образом, с помощью ре-гистра сдвига можно осуществлять преобразование информации из последовательной формы представления в параллельную. Очевидно, если предусмотрена запись информации параллельным кодом, то можно преобразовывать информацию из параллельной формы представления в последовательную.

Регистры сдвига могут быть построены и на триггерах одноступенчатой структуры. В этом случае в каждом разряде регистра нужно использовать два триггера, которые управляются двумя сдвинутыми во времени тактовыми импульсами. Наличие двух триггеров в одном разряде позволяет поразрядно продвигать информацию в регистре от входа к выходу. Если бы в регистре были применены одноступенчатые триггеры по одному на разряд, то правило работы регистра было бы нарушено: при первом же импульсе сдвига

информация, записавшись в первый разряд, перешла бы во второй, затем в третий и т.д.

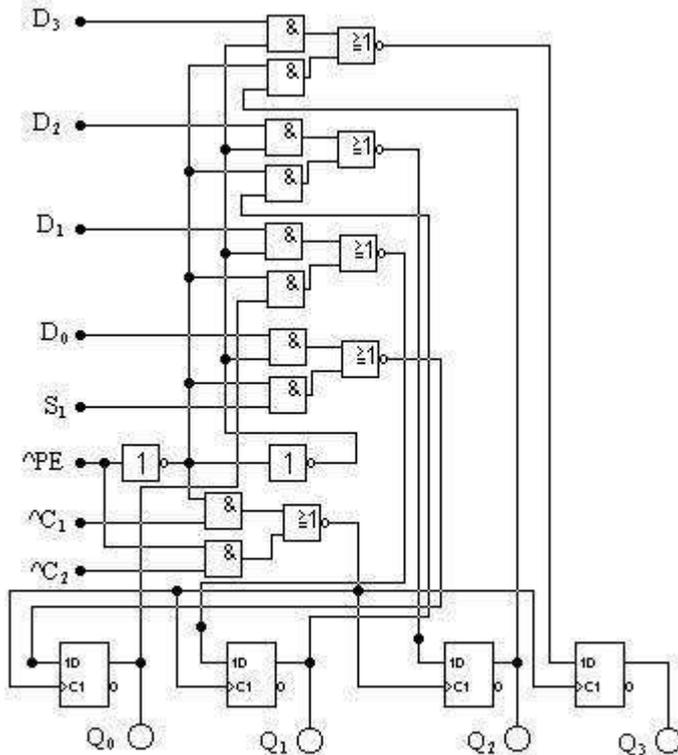


Рис.3. Схема четырёхразрядного сдвигового регистра K155IP1

Реверсивные регистры сдвига объединяют в себе свойства регистров прямого и обратного сдвига. Строятся они по тем же схемотехническим принципам, что и рассмотренные регистры, но с использованием дополнительных логических элементов в межразрядных связях. Указанная особенность реверсивного регистра показана на примере  $i$ -разряда (рис. 2), состоящего из  $D$ -триггера и логической схемы, на входы которой поданы:  $Q_{i+1}$ -сигнал с выхода младшего разряда,  $Q_{i+1}$ -сигнал с выхода старшего разряда,  $V$ -сигнал управления направлением сдвига:  $V=1$  –вправо,  $V=0$  –влево.

Существуют многорежимные регистры. Их входные и выходные линии данных объединены и образуют так называемый порт данных. Это означает, что от шины данных процессора приходит один провод (а не два), который по команде служит или входным или выходным. Число сигнальных входов и выходов микросхемы за счёт портовой организации можно уменьшить в два раза.

Подробное рассмотрение регистров проведём на примере микросхемы K155IP1 – четырёхразрядный сдвиговый регистр, принципиальная схема которого приведена на рис.3. Он имеет последовательный вход данных  $S_1$ , четыре параллельных входа  $D_0 - D_3$ , а также четыре выхода  $Q_0 - Q_3$  от каждого из

триггеров. Регистр имеет два тактового входа  $^C_1$  и  $^C_2$ . От любого из пяти входов данных код поступит на выходы синхронно с отрицательным перепадом, поданным на выбранный тактовый вход.

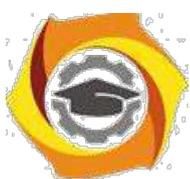
Вход разрешения параллельной загрузки  $^PE$  служит для выбора режима работы регистра. Если на вход  $^PE$  даётся напряжение высокого уровня, разрешается работа тактовому входу  $^C_2$ . В момент прихода на этот вход отрицательного перепада тактового импульса в регистр загружаются данные от параллельных входов  $D_0 - D_3$ .

Если на вход  $^PE$  подано напряжение низкого уровня, разрешается работа тактовому входу  $^C_1$ . Отрицательные фронты последовательности тактовых импульсов сдвигают данные от последовательного входа  $S_1$  на выход  $Q_0$ , затем на  $Q_1$ ,  $Q_2$  и  $Q_3$ , т.е. вправо. Сдвиг данных по регистру влево получится, если соединить выход  $Q_3$  и вход  $D_2$ , выход  $Q_2$  и вход  $D_1$ ,  $Q_1$  и  $D_0$  по схеме рис. 2. Регистр можно перевести в параллельный режим, подав на вход  $^PE$  напряжение высокого уровня. Напряжение на входе  $^PE$  можно менять только, если на обоих тактовых входах уровни низкие. Однако если на входе  $^C_1$  напряжение низкого уровня, перемена сигнала на входе  $^PE$  от низкого уровня к высокому не меняет состояния выходов.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

2.1. Включите необходимое программное обеспечение.



2.2. Соберите схему простейшего сдвигающего регистра на D-триггерах по рис. 4, воспользуясь триггерами из контейнера **Se-quential**.

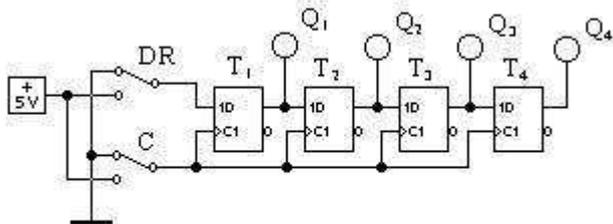


Рис.4. Сдвигающий регистр на D-триггерах

Спланируйте эксперимент и постройте временные диаграммы  $Q_i = f(D, R, C)$ .

2.3. Соберите схему для исследования универсального регистра по рис. 5. Для этого используется интегральная микросхема 74198 – восьмибитовый регистр сдвига из контейнера **Sequential** в шаблоне SHIFT REGS.

На этом рисунке:

$D_a, \dots, D_h$  - входы для параллельной записи в регистр байта

$a b c d e f g h$ ;

$Q_a, \dots, Q_h$  - выходы регистра;

$C$  - вход синхронизации;

$R$  - вход начальной установки всех разрядов в нуль;

$DR, DL$  - входы для последовательного ввода информации слева и справа;

$S_1, S_0$  - входы, задающие режим работы регистра: параллельная запись, хранение, сдвиг влево и вправо. Для

данной схемы спланировать эксперимент и составить таблицу, описывающую поведение регистра при

различных режимах ( $S_1$  и  $S_0$ ), различных значениях на входах регистра ( $D_a, \dots, D_h, DR$  и  $DL$ ). Определить в каких режимах на выполнение операции влияет  $C$ , какое

действие оказывает  $R$ . Выясните необходимые состояния управляющих сигналов для реализации различных режимов работы регистра – запись, хранение, сдвиг влево, сдвиг вправо.

Для режимов сдвига вправо и влево построить временные диаграммы.

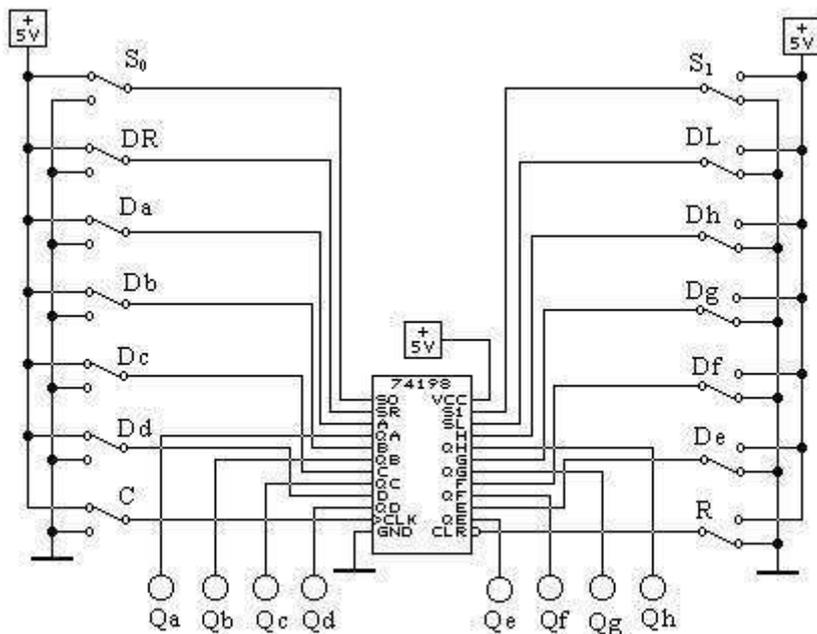


Рис.5. Схема для исследования универсального регистра

### 2.3. КОНТРОЛЬНЫЕ ВОПРОСЫ

и Что такое регистр, каково его назначение?

и Что определяет разрядность записываемого в регистр числа? Что такое быстродействие регистра?

и Как реализуется преобразование параллельного кода записанного числа в последовательный код?

и Как реализуется операция сдвига числа в регистре в ту или иную сторону?

и Поясните по принципиальной схеме рис. 3 работу регистра K155IP1.

Контрольные вопросы ПЗ2 (ПК-4):

7) Назовите основные компоненты, из которых состоит настольная или портативная компьютерная система.

8) Что такое системная шина и какова ее роль в компьютерной системе?

9) Что такое контроллер?

10) Назовите и кратко охарактеризуйте основные типы портов в компьютерной системе.

11) Какие области памяти используются в операциях ввода-вывода?

12) Каким образом контроллер устройства информирует процессор об окончании операции ввода-вывода?

13) Как организована обработка прерываний?

14) Что такое вектор прерываний?

15) Как система обрабатывает ситуацию, когда при обработке прерывания возникает другое прерывание?

16) Что такое программируемое прерывание (ловушка) и когда оно генерируется?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

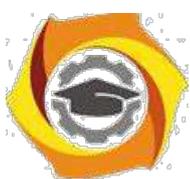
- 17) Каково назначение прерывания по таймеру?
- 18) Что такое опрос устройств и какова его роль в системе?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

11

- с Чем синхронный ввод-вывод отличается от асинхронного?
- с Что такое таблица состояния устройств и для чего она используется операционной системой?
- с В чем отличие основной памяти от внешней памяти?
- с Опишите кратко устройство жесткого диска.
- с Что такое ассоциативная память (кэш) и как она позволяет оптимизировать обращение к более медленным видам памяти?
- с Назовите основные виды внешней памяти.
- с Какова цель введения привилегированного режима и какие действия в нем разрешены, по сравнению с обычным режимом исполнения программ?
- с Каким образом система определяет текущий режим исполнения?
- с Что такое системный вызов и как они используются для организации ввода-вывода?
- с Что такое регистр базы и регистр границы и как они используются для защиты памяти?
- с Что такое таймер и как прерывания по таймеру используются операционной системой?
- с Как организована защита памяти в системах с теговой архитектурой?
- с Что такое дескриптор и для чего он предназначен?



## ПРАКТИЧЕСКАЯ РАБОТА № 3.

### Организация памяти ЭВМ

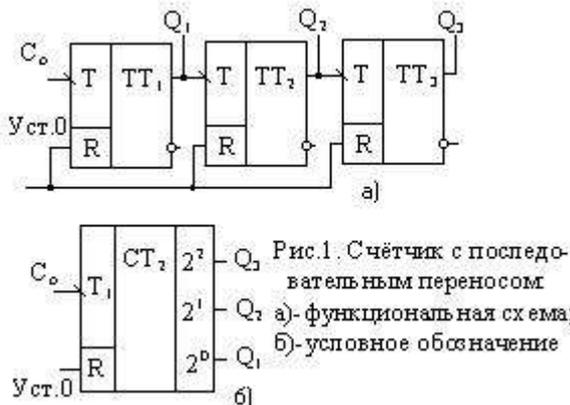
#### 3.1. КРАТКАЯ ТЕОРИЯ

Цифровым счётчиком импульсов называют последовательный цифровой узел, который осуществляет счёт поступающих на его вход импульсов. Результат счёта формируется счётчиком в заданном коде и может храниться некоторое время.

Счётчики строят на Т-триггерах и TV-триггерах с применением при необходимости логических элементов в цепях межразрядных связей. Количество триггеров N должно быть таким, чтобы множество внутренних состояний счётчика  $2^N$  было не меньше максимального числа импульсов, которое должно быть посчитано. С приходом очередного счётного импульса изменяется состояние счётчика, которое в заданном коде отображает результат счёта. Если количество счётных импульсов не ограничивать, то счётчик будет работать в режиме деления их числа на коэффициент (модуль) счёта  $K_{сч}$ , равный  $2^N$ . Через каждые  $2^N$  импульсов он будет возвращаться в начальное состояние и снова считать импульсы. Если необходимый коэффициент счёта не равен  $2^N$ , применяют различные способы сокращения числа внутренних состояний счётчика.

Таблица 1

| Номер состояния | $Q_2$ | $Q_1$ | $Q_0$ |
|-----------------|-------|-------|-------|
| 0               | 0     | 0     | 0     |
| 1               | 0     | 0     | 1     |
| 2               | 0     | 1     | 0     |
| 3               | 0     | 1     | 1     |
| 4               | 1     | 0     | 0     |
| 5               | 1     | 0     | 1     |
| 6               | 1     | 1     | 0     |
| 7               | 1     | 1     | 1     |
| 0               | 0     | 0     | 0     |



Порядок

задан табл. 1. Как следует из таблицы, с приходом очередного счётного импульса к содержимому счётчика прибавляется единица. При этом увеличивается на единицу номер состояния, являющийся десятичным эквивалентом соответствующего данному состоянию двоичного числа.

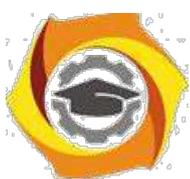
Изменение состояния каждого последующего разряда происходит при изменении состояния предыдущего разряда от 1 к 0. Это означает, что всякий раз, когда данный триггер в счётчике переходит из состояния 1 в состояние 0, на его выходе должен формироваться сигнал переноса, вызывающий срабатывание следующего триггера. Если же данный триггер переходит из 0 в 1, то сигнала переноса на его выходе не должно быть. Из табл. 1 также следует, что триггер первого, самого младшего разряда, должен менять своё состояние каждый раз с приходом очередного счётного импульса, а триггер каждого последующего разряда – вдвое реже триггера предыдущего разряда.

Описанный порядок смены состояний счётчика и характер процесса их установления могут быть реализованы, если счётчик будет построен на последовательно соединённых Т-триггерах. Каждый последующий разряд при этом будет переключаться сигналом переноса, формируемым на выходе предыдущего разряда. Счётные импульсы должны быть поданы на вход триггера самого младшего разряда. Счётчики, построенные таким образом, получили название счётчиков с последовательным переносом. Пример трёхразрядного счётчика на Т-триггерах двухступенчатой структуры приведён на рис. 1. Для установки исходного состояния служит шина «Уст.0», которой объединены R-входы всех триггеров. Нулевое состояние триггеров устанавливается подаваемым по этой шине положительным импульсом напряжения между уровнями 0 и 1. На левом поле условного графического обозначения счётчика (рис. 1,б) показано, что его входом является  $T_1$ -вход первого разряда, а на правом поле указан «вес» каждого разряда.

Вычитающий счётчик с последовательным переносом имеет обратный порядок смены состояний: с приходом очередного счётного импульса содержащееся в счётчике число уменьшается на единицу (табл. 2). Другая особенность вычитающего счётчика – триггер каждого последующего разряда переключается в противоположное состояние при изменении уровня на выходе триггера предыдущего разряда от 0 к 1, т.е. при сигнале займа, обратном сигналу переноса в суммирующем счётчике. Строится вычитающий счётчик так же, как суммирующий, но с тем отличием, что со входом каждого последующего триггера в отличие от рис. 1 соединяется инверсный выход предыдущего триггера.

Счётчики можно классифицировать по ряду признаков. По направлению счёта их делят на суммирующие (с прямым счётом), вычитающие (с обратным счётом) и реверсивные. По способу организации переноса различают счётчики с последовательным, параллельным и комбинированным (параллельно-последовательным) переносом.

Рассмотрим пример реализации трёхразрядного суммирующего счётчика в коде 4-2-1 с последовательным переносом. смены состояний счётчика



Из работы трёхразрядного счётчика следует, что в наихудшем случае новое его состояние устанавливается с задержкой, равной утроенной задержке переключения одного триггера, что вызвано последовательным во времени

Таблица 2

| № состояния | Q <sub>3</sub> | Q <sub>2</sub> | Q <sub>1</sub> |
|-------------|----------------|----------------|----------------|
| 7           | 1              | 1              | 1              |
| 6           | 1              | 1              | 0              |
| 5           | 1              | 0              | 1              |
| 4           | 1              | 0              | 0              |
| 3           | 0              | 1              | 1              |
| 2           | 0              | 1              | 0              |
| 1           | 0              | 0              | 1              |
| 0           | 0              | 0              | 0              |
| 7           | 1              | 1              | 1              |

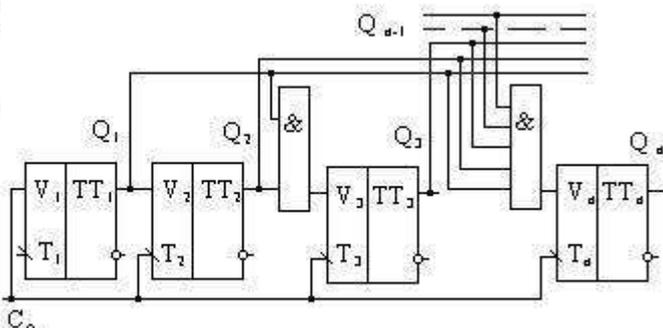


Рис.2. Счётчик с параллельным переносом

распространением сигнала переноса через все разряды счётчика. Таким образом, в счётчике с последовательным переносом неэффективно используется быстродействие триггеров, особенно при большом числе разрядов. В этом состоит существенный недостаток счётчиков с последовательным переносом, из-за которого область их применения ограничивается цифровыми устройствами с небольшим числом разрядов и невысоким быстродействием.

Один из широко применяемых способов ускорения переноса в счётчике основан на введении логических элементов, с помощью которых достигается возможность одновременного (параллельного) формирования сигнала переноса для всех разрядов. Для реализации этого способа применяют TV-триггеры. На Т-входы всех триггеров одновременно подаются счётные импульсы, а на V-вход каждого триггера поступает сигнал переноса, формируемый логической схемой в виде уровня 1. Триггеры, на V-входе которых имеется сигнал переноса, одновременно переключаются с приходом очередного счётного импульса, и, таким образом, устанавливается новое состояние счётчика. Для определения вида цепи переноса обратимся к уже рассмотренной табл.1. Из неё следует, что первый разряд, как и в счётчике с последовательным переносом, должен быть построен на Т-триггере. Если применяется TV – триггер, то его V-вход следует соединить с Т-входом.

Второй триггер перебрасывается в противоположное состояние счётным импульсом при наличии 1 на выходе первого триггера, а третий триггер перебрасывается при наличии 1 на выходах двух предыдущих триггеров. Обобщая эту закономерность на случай N-разрядного счётчика получим, что каждый последующий триггер должен переключиться под воздействием счётного импульса при наличии 1 на выходах всех предыдущих триггеров. Следовательно, для формирования сигнала переноса в каждый разряд счётчика необходимо включить элемент И и соединить его входы с прямыми выходами всех предыдущих разрядов, а выход – с V-входом триггера данного разряда. Пример суммирующего счётчика с параллельным переносом на TV – триггерах приведён на рис.2. Быстродействие этого счётчика выше, чем счётчика с последовательным переносом, поскольку оно равно быстродействию переноса одного разряда. Недостаток – необходимость включения в схему элемента И с нарастающим от разряда к разряду числом входов. Это нарушает регулярность структуры и ограничивает возможность наращивания его схемы. Частично этот недостаток можно устранить при использовании триггеров с входной логикой.

Многие серии микросхем содержат JK-триггеры с входной логикой. Для преобразования JK-триггера в TV-триггер необходимо объединить входы JиKв

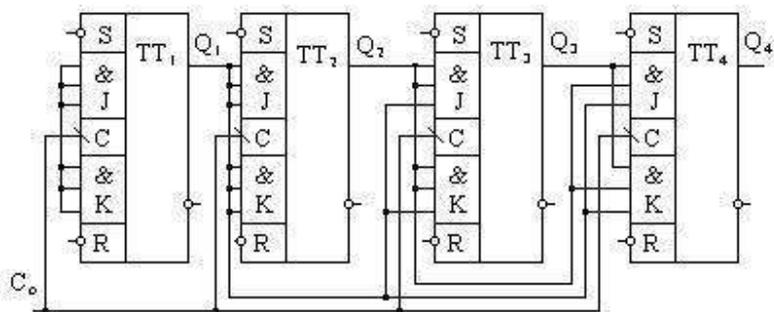
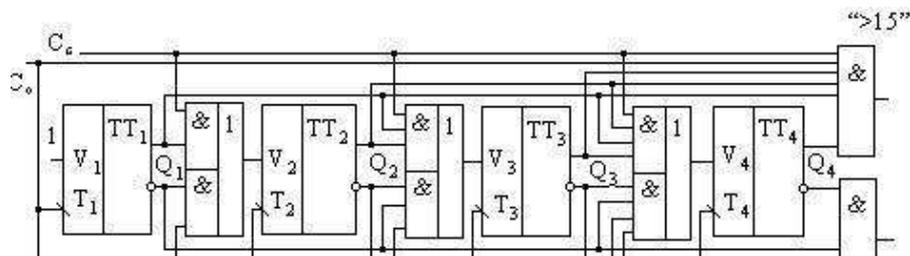


Рис.3. Счётчик на JK-триггерах с входной логикой

триггер необходимо объединить входы JиKв один, это и будет V-вход. У триггера с тремя конъюнктивно связанными J-входами и тремя конъюнктивно связанными K-входами могут быть образованы, следовательно, три конъюнктивно связанные V-входа. При реализации счётчика на таких триггерах исключаются дополнительные логические элементы в цепях переноса. Однако ограничение в числе разрядов остаётся. На таких триггерах можно построить лишь четырёхразрядный счётчик (рис. 3).

Вычитающий счётчик с параллельным переносом строится так же, как и суммирующий, но сигналы переноса снимаются с инверсных относительно используемых в суммирующем счётчике выходов триггеров.

Реверсивный счётчик, объединяющий возможности суммирующего и вычитающего, строится таким образом, чтобы обеспечивалось управление направлением счёта с помощью сигналов разрешения на реализацию операций сложения C<sub>с</sub> и вычитания C<sub>в</sub>. Поэтому его схема содержит дополнительную комбинационную часть, выполняющую указанную функцию.





Нередко счётчики с параллельным переносом, выпускаемые в виде микросхем, имеют помимо основных выходов – дополнительные, как это показано, например, на рис. 4. На одном из выходов, обозначенном «>15», сигнал 1 появляется при заполнении счётчика единицами, т.е. когда он перешёл в состояние с номером 15. Следовательно, на этом выходе формируется сигнал переноса в следующий счётчик. На другом выходе, обозначенном «<0», сигнал появляется при заполнении счётчика нулями и является сигналом займа в следующий счётчик в режиме вычитания.

Реверсивный счётчик можно построить и на Т-триггерах (рис. 5). Как и в рассмотренном ранее суммирующем счётчике, счётные импульсы поступают на Т-вход триггера через логические элементы только в том случае, если на логических элементах имеются сигналы разрешения с выходов предыдущих разрядов.

9) счётчике на рис. 5,а для счётных импульсов предусмотрены два входа. Если счётчик должен работать в режиме прямого счёта, импульсы следует подавать на вход «+1», в режиме обратного счёта – на вход «-1». При использовании такого счётчика в качестве реверсивного с одним источником импульсов необходимо предусмотреть внешнее устройство коммутации счётных импульсов на суммирующий «+1» либо на вычитающий «-1» входы.

Вариант такой коммутирующей приставки к счётчику приведён на рис. 5,б. При подаче положительных импульсов на S-вход RS-триггера на его прямом выходе установится единичный уровень, который откроет элемент 1 для счётных импульсов С<sub>0</sub>. Счётчик будет работать в режиме сложения. Если подать положительный импульс на R-вход триггера, откроется для счётных импульсов элемент 2, и счётчик будет работать в режиме вычитания.

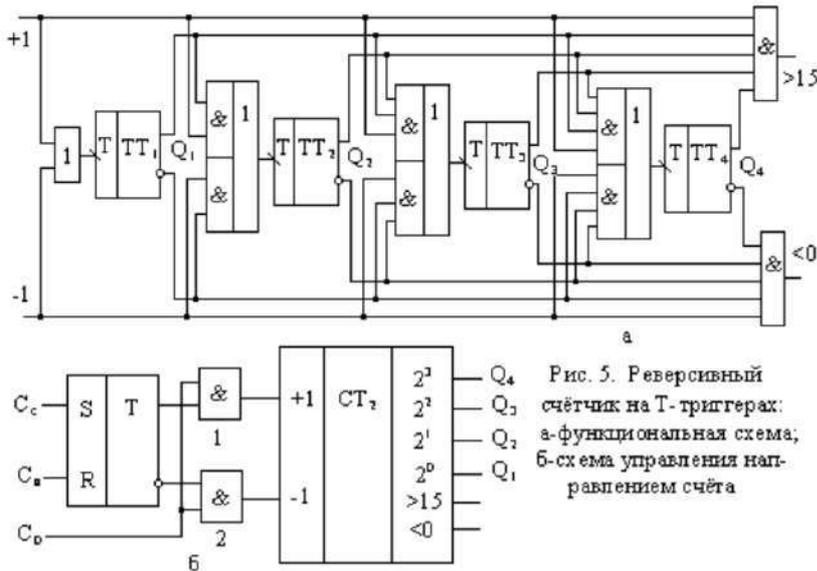


Рис. 5. Реверсивный счётчик на Т-триггерах:  
а-функциональная схема,  
б-схема управления направлением счёта

### 3.2. МЕТОДИКА ВЫПОЛНЕНИЯ

- 2.1. Включите необходимое программное обеспечение.
- 2.2. Соберите схему простейшего

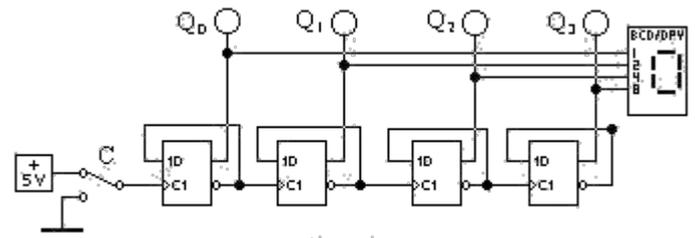


Рис.6. Суммирующий счётчик с последовательным переносом

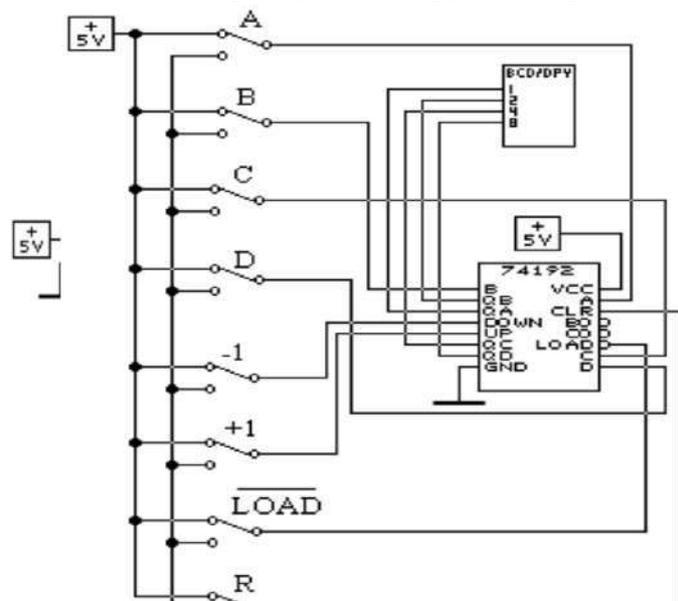
суммирующего счётчика с последовательным переносом по рис. 6. Для этого следует воспользоваться синхронизируемыми положительным фронтом D-триггерами из контейнера **Sequential**. Для индикации состояния счётчика в виде десятичного числа используется семисегментный индикатор из контейнера **Indicators**. Спланируйте эксперимент и постройте временные диаграммы  $Q_i = f(C)$  для этой схемы.

2.3. Соберите схему простейшего вычитающего счётчика (рис. 7), отличие которого от схемы на рис. 6 в том, что входы синхронизации С1 у 2-го, 3-го и 4-го триггеров соединены не с инверсными, а с прямыми выходами 1-го, 2-го и 3-го триггеров. Спланируйте эксперимент и постройте временные диаграммы  $Q_i = f(C)$  для этой схемы.

2.3. На рис. 8 приведена схема для исследования универсального двоично-десятичного счётчика. Соберите эту схему, используя счётчик **74192 (4-Bit Up/Down Counter)** из контейнера **Sequential** в шаблоне **CO-UNTERS**. Для индикации состояния счётчика в виде десятичного числа используется семисегментный индикатор из контейнера **Indicators**.

Спланируйте эксперимент и постройте временные диаграммы для различных режимов работы этого счётчика: установка в 0- (R), параллельная запись -(Load), +1 и -1. Постройте графы переходов для счётчика, установленного в начальные состояния 10-(A), 11-(B), 12-(C), 13-(D), 14-(E), 15-(F) при сигналах +1 и -1.

На рис. 8 использованы обозначения: DCBA – входы для параллельной записи в счётчик двоичного числа;





**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Load – (инверсный сигнал) – синхронизация параллельной записи  
числа в счётчик;  
R – установка счётчика в состояние 0000;



+1 – прибавление к содержимому счётчика единицы (положительный фронт); -1 – вычитание из содержимого счётчика единицы (положительный фронт).

2.4. Получите у преподавателя задание на проектирование счётчика с параллельным переносом на D-триггерах и нетрадиционным порядком счёта:

- 1 вариант: счёт – 1,3,7,4,0; 2 вариант: счёт – 0,3,6,4,0;
- 3 вариант: счёт – 1,3,7,4,8; 4 вариант: счёт – 7,5,3,1,0;
- 5 вариант: счёт – 0,3,1,4,2; 6 вариант: счёт – 5,7,3,2,0;
- 7 вариант: счёт – 3,2,1,5,7; 8 вариант: счёт – 2,3,4,1,5.

Соберите схему спроектированного счётчика и исследуйте её на соответствие вашему варианту задания. Постройте граф переходов.

### 3.3. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 8. Поясните принципы построения суммирующего и вычитающего счётчиков по табл. 1 и 2.
- 9. Как реализуется параллельное формирование сигнала переноса во всех разрядах счётчика?
- 10. Поясните построение и работу реверсивного счётчика.
- 11. Как функционируют выходы “>15” и “<0” реверсивного счётчика, каково их практическое применение?

Контрольные вопросы ПЗЗ (ПК-4):

- 15. Какие операции определяет понятие «обращение к ЗУ»?
- 16. Какие единицы измерения используются для указания емкости запоминающих устройств?
- 17. В чем отличие между временем доступа и периодом обращения к запоминающему устройству?
- 18. Чем вызвана необходимость построения системы памяти по иерархическому принципу?
- 19. Что включает в себя понятие «локальность по обращению»?
- 20. Благодаря чему среднее время доступа в иерархической системе памяти определяется быстродействующими видами ЗУ?
- 21. Что в иерархической системе памяти определяют термины «промах» и «попадание»?
- 22. На какие вопросы необходимо ответить, чтобы охарактеризовать определенный уровень иерархической памяти?
- 23. Какие виды запоминающих устройств может содержать основная память?
- 24. Охарактеризуйте возможные варианты построения блочной памяти.
- 25. Какие возможности по сокращению времени доступа к информации предоставляет блочная организация памяти?
- 26. Чем обусловлена эффективность расслоения памяти?
- 27. Какая топология запоминающих элементов лежит в основе организации полупроводниковых ЗУ?
- 28. Какое минимальное количество линий должен содержать столбец ИМС памяти?
- 29. Поясните назначение управляющих сигналов в микросхеме памяти.
- 30. Чем отличаются страничный, быстрый страничный и пакетный режимы доступа к памяти?
- 31. Чем обусловлена необходимость регенерации содержимого динамических ОЗУ?
- 32. Охарактеризуйте основные сферы применения статических и динамических ОЗУ.
- 33. Какое влияние на асинхронный режим работы памяти оказывает синхронный характер работы контроллера памяти?
- 5. Какой вид ПЗУ обладает наиболее высокой скоростью перепрограммирования?
- 6. Какими методами обеспечивается энергонезависимость ОЗУ?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

7. В чем состоит различие между режимами стандартной и запаздывающей записи в статических ОЗУ?
8. В чем проявляется специфика ОЗУ, предназначенных для видеосистем??



## ПРАКТИЧЕСКАЯ РАБОТА № 4.

### Организация внешней памяти ЭВМ. Алгоритмы умножения чисел. Алгоритмы умножения чисел

#### 4.1. КРАТКАЯ ТЕОРИЯ

Сумматор по модулю 2 – цифровой узел с  $m$  входами и одним выходом, работающий в соответствии со следующим правилом: сигнал 1 появляется на его выходе всякий раз, когда в наборе входных сигналов содержится нечётное число 1. Поэтому этот узел ещё называют схемой проверки на чётность. В частном случае при числе входов равном 2,

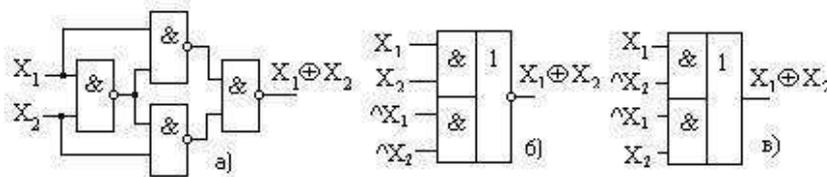


Рис.1. Сумматор по модулю 2: а) - функциональная схема на элементах И-НЕ; б) - на элементах И-ИЛИ-НЕ; в) - на элементах И-ИЛИ

сумматор по модулю 2 выполняет функцию логического элемента «Исключающее ИЛИ»: на выходе 1 будет только при 1 на одном из входов. Функциональные схемы двухвходовых сумматоров по модулю 2 приведены на рис. 1. Реализуемая здесь логическая функция  $= A \oplus B = A \oplus B$ , где символ  $\oplus$  есть признак инверсии.

При суммировании двух одноразрядных чисел имеем:  $0+0=0$ ,  $1+0=1$ ,  $0+1=1$  и  $1+1=10$ . В последнем случае число 10 (в десятичной записи это 2) оказалось двоичным двухразрядным. Появившаяся в старшем разряде суммы единица называется единицей переноса.

е схеме «Исключающее ИЛИ» несложно добавить выход переноса, т. е. генератор старшего разряда. Для этого оба суммируемых одноразрядных числа следует подать на схему И, выход которой даст необходимый старший разряд переноса  $1 \cdot 1 = 1$ . На рис. 2,а показана реализация схемы суммирования двух одноразрядных чисел, состоящая из элементов «Исключающее ИЛИ» и «И». Схема имеет два выходных сигнала: суммы и переноса  $C$ . Такая схема называется полусумматором. Таблица состояний полусумматора показана на рис. 2,б.

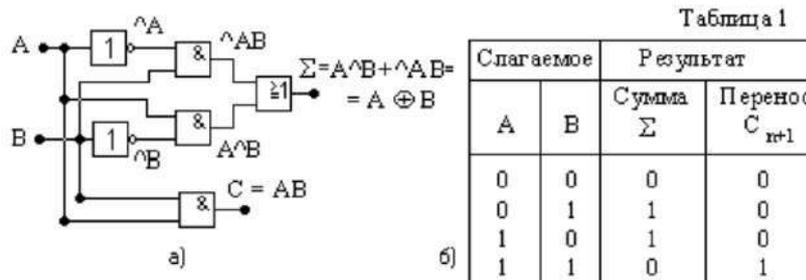


Рис.2. Полусумматор (а) и таблица его состояний (б)

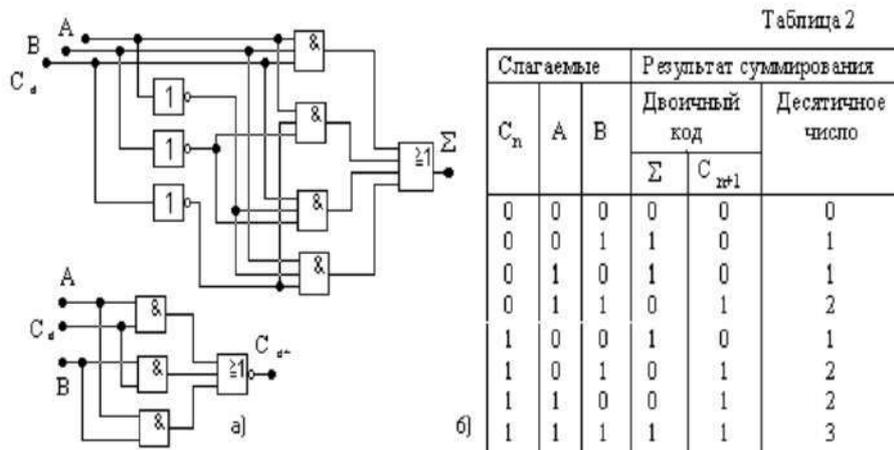


Рис.3. Полный сумматор (а) и таблица его состояний (б)



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Полный сумматор должен иметь вход для сигнала переноса  $C_n$  от со-седнего младшего разряда . Схема полного сумматора двух одноразрядных слов показана на рис. 3,а, а таблица истинности – на рис. 3,б. В последнем



столбце таблицы результаты суммирования даны в десятичной форме. В присутствии входной единицы переноса  $C_n$  сумма чисел  $A$  и  $B$  увеличивается на единицу.

Полные сумматоры многоразрядных чисел составляются из одноразрядных и могут складывать многоразрядные числа  $B$  двумя способами переноса: параллельным или последовательным. При параллельном переносе реализуется одновременное поразрядное сложение слов, при последовательном переносе - формирование сигналов переноса для старших разрядов происходит после выполнения операции сложения в младших разрядах. Поэтому время выполнения операции сложения  $N$ -разрядных чисел равно  $N \cdot T_3$ , где  $T_3$  – задержка на одном одноразрядном сумматоре.

#### 4.2. МЕТОДИКА ВЫПОЛНЕНИЯ

2.1. Включите необходимое программное обеспечение.

2.2. Соберите схему одноразрядного сумматора по рис. 4,а, экспериментально получите таблицы истинности для суммы  $S_i$  и переноса  $P_i$ . Создайте из схемы рис. 4,а подсхему  $sm$  одноразрядного сумматора по рис. 4,б.

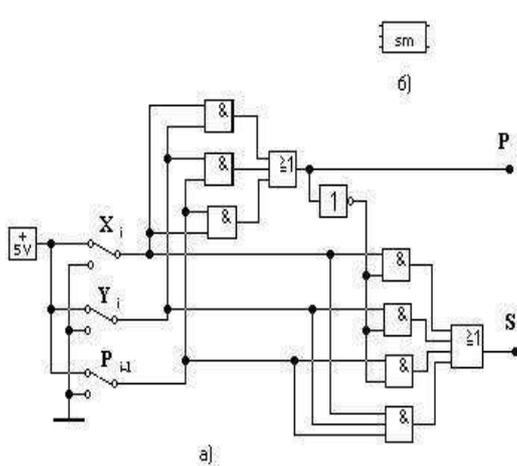


Рис.4. Одноразрядный сумматор: а) – функциональная схема; б) - подсхема

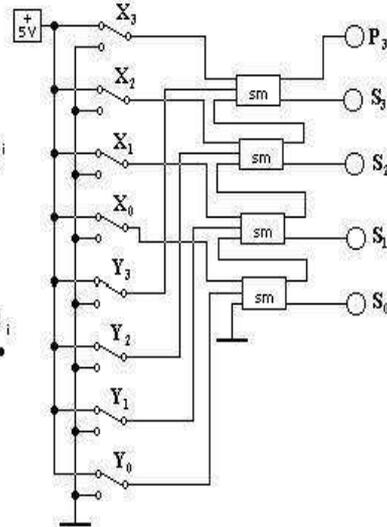


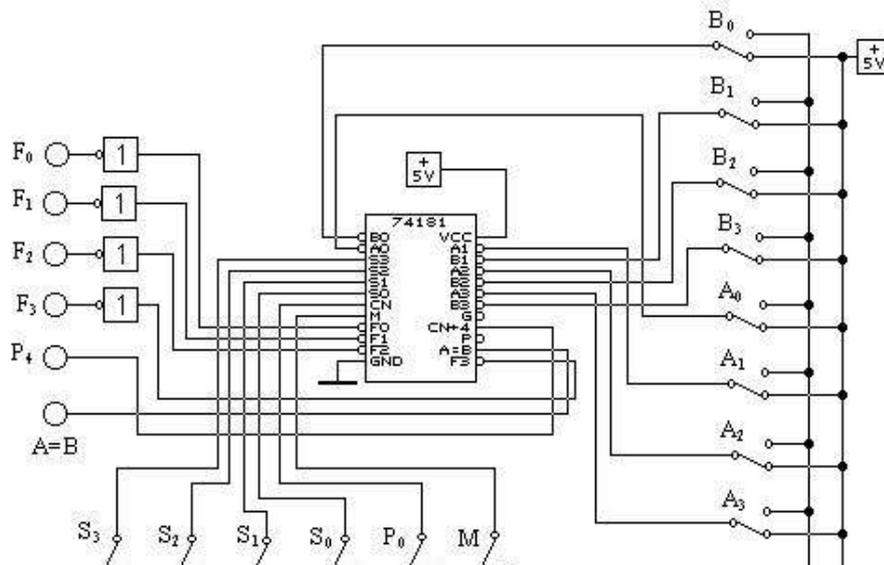
Рис.5. Четырёхразрядный сумматор

2.3. Соберите схему четырёхразрядного сумматора по рис.5 на базе подсхемы  $sm$  и исследуйте его работу при различных значениях двоичных чисел  $X$  и  $Y$ . Результаты оформить в виде табл. 3.

2.4. Соберите схему для исследования работы арифметико-логического устройства АЛУ по рис. 6. Для этого следует выбрать из контейнера **Combinational** (шаблон **ARITHMETIC**) микросхему **74181 ALU** -арифметико-логическое устройство. Это АЛУ выполняет поразрядные логические операции (при  $M=1$ ) над четырёхразрядными словами  $A$  и  $B$ , а так же арифметические операции (при  $M=0$ ). На результат логических операций не оказывает влияния значение  $P_0$  –перенос от младших разрядов. Код выполняемой операции задаётся значениями  $S_3S_2S_1S_0$ .

Обратите внимание на то, что входы АЛУ для операндов  $A$  и  $B$  – инверсные. Поэтому на переключателях  $A_i$ ,  $S_i$  значению логической единицы в АЛУ соответствуют верхние положения ключей. Выходы  $F_3F_2F_1F_0$  - так же инверсные. Для удобства исследования АЛУ выходы подключены к индикаторам через инверторы.

По клавише «F1» можно раскрыть таблицу истинности для АЛУ. В этой табл. 4 все выражения записаны для прямых значений операндов  $A$ ,  $B$  и результата  $F$ . В столбце «**LOGIC FUNCTION**» символом «**+**» обозначена поразрядная логическая операция «ИЛИ», символом «**+**» обозначена поразрядная логическая операция «ИСКЛЮЧАЮЩЕЕ ИЛИ». В столбцах «**ARITHMETIC OPERATIONS**» словами «**PLUS**» и «**MINUS**»





**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

обозначены соответствующие арифметические операции. Символами «Н», «L» в табл. 4 обозначены логические «1» - «0».

2.5. Задайте  $M=1$ . Из имеющихся 16 логических операций проверьте правильность реализации любых 4 для двух произвольных комбинаций операндов A и B в каждой из них. Из восьми наборов операндов A и B в двух случаях эти операнды должны совпадать для проверки истинности сигнала  $A=B$  на выходе АЛУ. Составьте табл. 5 результатов проверки, в которой укажите теоретические значения выходных величин для заданной логической функции и их экспериментальные значения. Сделайте вывод.

2.6. Задайте  $M=0$  при отсутствии переноса ( $P_0=0$ ). Затем повторите действия п. 2.5. для проверки реализации арифметических операций, сведя результаты в табл. 6. Сделайте вывод.

2.7. Задайте  $M=1$  при наличии переноса ( $P_0=1$ ). Затем повторите действия п. 2.5. для проверки реализации арифметических операций, сведя результаты в табл. 7. Сделайте вывод.

**Таблица 4**

| Selection<br>S3 S2 S1 S0 | M = H<br>LOGIC<br>FUNCTIONS | M = L Arithmetic Operations |                           |
|--------------------------|-----------------------------|-----------------------------|---------------------------|
|                          |                             | Cn = L ( no carry )         | Cn = H ( with carry )     |
| 0 0 0 0                  | $F=\wedge A$                | F=AMINUS1                   | F = A                     |
| 0 0 0 1                  | $F = \wedge(AB)$            | F = AB MINUS 1              | F=AB                      |
| 0 0 1 0                  | $F=\wedge A+B$              | F = A^B MINUS 1             | F=A^B                     |
| 0 0 1 1                  | F = 1                       | F = MINUS 1 ( 2's comp)     | F = Zero                  |
| 0 1 0 0                  | $F = \wedge(A+B)$           | F = A PLUS (A+^B)           | F = A PLUS (A+^B) PLUS 1  |
| 0 1 0 1                  | $F=\wedge B$                | F = AB PLUS (A+^B)          | F = AB PLUS (A+^B) PLUS 1 |
| 0 1 1 0                  | $F = \wedge(A''+''B)$       | F = A MINUS B MINUS 1       | F=AMINUSB                 |
| 0 1 1 1                  | $F = A+\wedge B$            | F=A+^B                      | F = (A+^B) PLUS 1         |
| 1 0 0 0                  | $F=\wedge AB$               | F = A PLUS (A+B)            | F = A PLUS (A+B) PLUS 1   |
| 1 0 0 1                  | $F = A''+''B$               | F=APLUSB                    | F=APLUSBPLUS1             |
| 1 0 1 0                  | F = B                       | F = A^B PLUS (A+B)          | F = AB PLUS (A+B) PLUS 1  |
| 1 0 1 1                  | F=A+B                       | F = (A+B)                   | F = (A+B) PLUS 1          |
| 1 1 0 0                  | F = 0                       | F=APLUSA                    | F=APLUSAPLUS1             |
| 1 1 0 1                  | $F=A\wedge B$               | F=ABPLUSA                   | F = AB PLUS A PLUS 1      |
| 1 1 1 0                  | F=AB                        | F = A^B PLUS A              | F = A^B PLUS A PLUS 1     |
| 1 1 1 1                  | F = A                       | F = A                       | F=APLUS1                  |

**4.3. КОНТРОЛЬНЫЕ ВОПРОСЫ**

11. Поясните работу схем сумматоров по модулю 2 на рис. 1.
12. Опишите работу схемы полусумматора и таблицы его состояний по рис. 2.
13. Поясните работу схемы полного сумматора и таблицу его истинности по рис. 3.
14. Опишите работу функциональных схем по рис. 4 и рис. 5.

Контрольные вопросы ПЗ4 (ПК-4):

1. Чем объясняется тенденция размещения стека в области старших адресов основной памяти?
2. Какая информация хранится в указателе стека?
3. Поясните назначение маски в ассоциативном запоминающем ЗУ.
26. Как реализуется запись новой информации в ассоциативное ЗУ?
5. Какие виды поиска можно осуществлять в ассоциативном ЗУ?
6. Поясните назначение и логику работы кэш-памяти.
7. Какие проблемы порождает включение в иерархию ЗУ кэш-памяти?
8. Чем обусловлено разнообразие методов отображения основной памяти на кэш-память?  
В Какому требованию должен отвечать «идеальный» алгоритм замещения содержимого кэш-памяти?
10. Какими методами обеспечивается согласованность содержимого основной и кэш-памяти?
11. Чем обусловлено введение дополнительных уровней кэш-памяти?
12. Какие факторы влияют на выбор емкости кэш-памяти и размера блока?
13. Как соотносятся характеристики обычной и дисковой кэш-памяти?
14. Какими средствами обеспечивается виртуализация памяти?
- В Существует ли ограничение на размер виртуального пространства?
16. Что определяет объем страничной таблицы?
17. Какими приемами достигают сокращения объема страничных таблиц?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

18. Какие алгоритмы замещения используются при загрузке в основную память новой виртуальной страницы?
19. Поясните назначение буфера быстрого преобразования адреса (TLB).



20. Чем мотивируется разбиение виртуальных секторов на страницы?
21. Какая часть виртуального адреса остается неизменной при его преобразовании в физический адрес?
22. Чем обусловлена необходимость защиты памяти?
23. Охарактеризуйте состав операционных устройств, входящих в АЛУ. Из каких соображений и каким образом он может изменяться?
24. Поясните понятие «операционные устройства с жесткой структурой». В чем заключается жесткость их структуры? Каковы их достоинства и недостатки?
25. Чем обусловлено название операционных устройств с магистральной структурой? Сравните магистральные структуры с жесткими структурами, выделяя достоинства, недостатки и область применения.
26. Дайте развернутую характеристику классификации операционных устройств с магистральной структурой. Поясните достоинства и недостатки «минимального» и «максимального» вариантов.
27. Поясните функциональный состав параллельного операционного блока магистрального ОПУ. Каким образом можно минимизировать количество внешних связей этого блока? Ответ сопроводите конкретным примером.
28. Чем обусловлена Специфика целочисленного сложения и вычитания? Какую роль играют в них дополнительный код? К чему бы привел отказ от дополнительного кода? Ответ поясните на примерах. Как выявляется переполнение в этих операциях?
29. Сформулируйте достоинства, недостатки и область применения четырех вариантов целочисленного «традиционного» умножения. Как учитываются знаки сомножителей?
30. Охарактеризуйте суть двух групп логических методов ускорения умножения.
31. Попарно сравните алгоритм Бута, модифицированный алгоритм Бута, алгоритм Лемана.
32. Разработайте алгоритм умножения с обработкой за шаг трех разрядов множителя.
33. Поясните суть аппаратных методов ускорения умножения, выделив три возможных подхода.
34. В чем заключается основная идея древовидных умножителей? Каковы особенности их организации?
35. С какой целью и каким образом выполняется конвейеризация матричных и древовидных умножителей? Приведите (и поясните) конкретные примеры.
36. На конкретном примере объясните, как можно нарастить разрядность аппаратного умножителя.
37. Сравните организацию целочисленного деления с восстановлением остатка и без восстановления остатка. Как учитываются при делении знаки операндов?
38. Обоснуйте возможность совмещения структур умножителя и делителя. Опишите объединенную структуру.
39. Сформулируйте четыре пути ускорения целочисленного деления. Сравните между собой их возможную реализацию.
40. Какие из операций с плавающей запятой считаются наиболее сложными? Ответ обоснуйте на конкретных примерах.
41. В чем состоит специфика умножения с плавающей запятой? Поясните эту специфику на примере.

## **ПРАКТИЧЕСКАЯ РАБОТА № 5.**

### **Реализация команд пересылки и операций со стекком. Выполнение арифметических и логических команд**

#### **5.1. КРАТКАЯ ТЕОРИЯ**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

с цифровых устройствах часто возникает задача передачи цифровой информации от различных источников к одному приёмнику. Для этого на входе канала устанавливается устройство, называемое мультиплексором



(переключателем MS), которое согласно коду адреса подключает к выходу один из источников информации. Например, из четырёх источников D0,D1,D2 и D3, которые подключены к информационным входам мультиплексора, необходимо выбрать один. Для этого должен быть указан номер информационного входа. Обычно он задаётся двоичным кодом на управляющих входах мультиплексора. Для MS с двумя информационными входами достаточно одного управляющего входа X0, для MS с четырьмя информационными входами – достаточно двух - X1 и X0 и т.д. Булева функция, описывающая MS с четырьмя информационными входами, имеет вид:  $Y=D0^{\wedge}X1^{\wedge}X0 \vee D1^{\wedge}X1X0 \vee D2X1^{\wedge}X0 \vee D3X1X0, \dots (1)$ , где Y – значение логического сигнала на выходе MS;

D0,D1,D2 и D3 – значения логических сигналов на информационных входах MS;  
X1,X0 – управляющие переменные, задающие номер информационного входа MS.

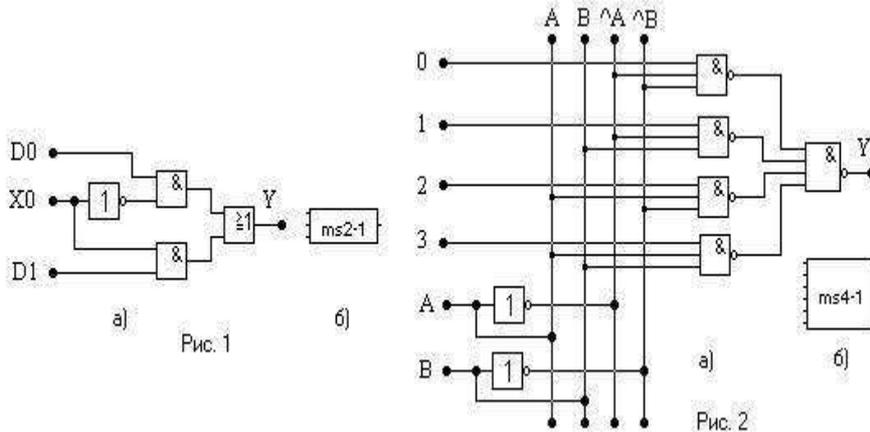
Аналогичный вид имеют выражения для MS с 2-мя, 8-ю и т.д. информационными входами. Обычно мультиплексоры обозначаются так: MS 2-1 – мультиплексор с двумя информационными входами на один выход; MS 8-1 – мультиплексор с восемью информационными входами на один выход и т.п. Мультиплексоры могут быть собраны из простейших логических элементов И, ИЛИ, НЕ или могут использоваться готовые MS в виде ИМС.

Схема MS 2-1, собранная из простейших логических элементов, приведена на рис. 1,а. Здесь же изображён вид субсхемы ms2-1 (рис. 1,б).

Схема MS4-1, собранная на базе простейших логических элементов, приведена на рис. 2,а. Здесь же показана субсхема ms4-1(рис. 2,б). Информационные входы обозначены как 0,1,2 и 3, а управляющие – А,В.

Другая сфера применения мультиплексоров – построение логических функций нескольких переменных в виде дизъюнктивной нормальной формы. Это применение основано на разложении булевой функции по 1-ой, 2-ум и т.д. переменным по методу Шеннона. В качестве примера рассмотрим применение разложения Шеннона для реализации одной и той же булевой функции  $Y=U(0,1,2,3,4, 8,12,13,15)$  от четырёх переменных А,В,С,Д с помощью мультиплексоров различного типа. Числа в скобках для функции U указывают те строки таблицы истинности, в которых Y принимает значения, равные 1. В СДНФ (совершенной дизъюнктивной нормальной форме) эта булева функция имеет вид:

$$Y=\wedge A^{\wedge}B^{\wedge}C^{\wedge}D \vee \wedge A^{\wedge}B^{\wedge}C^{\wedge}D, \dots(2)$$



где знак «\*» означает операцию логического умножения.

Для построения комбинационной схемы, реализующей функцию(2) на базе мультиплексоров MS 2-1, разложим функцию по одной переменной, начиная, например, с переменной старшего разряда - А. Сгруппируем в выражении (2) члены, содержащие переменные А и  $\wedge A$ . Получим:

$$Y=\wedge A^{\wedge}(\wedge B^{\wedge}C^{\wedge}D \vee \wedge B^{\wedge}C^{\wedge}D \vee \wedge B^{\wedge}C^{\wedge}D \vee \wedge B^{\wedge}C^{\wedge}D \vee B^{\wedge}C^{\wedge}D) \vee A^{\wedge}(\wedge B^{\wedge}C^{\wedge}D \vee B^{\wedge}C^{\wedge}D \vee B^{\wedge}C^{\wedge}D \vee B^{\wedge}C^{\wedge}D) =\wedge A^{\wedge}f_1 \vee A^{\wedge}f_2, \dots (3)$$

$$\text{где функции } f_1 = \wedge B^{\wedge}C^{\wedge}D \vee \wedge B^{\wedge}C^{\wedge}D \vee \wedge B^{\wedge}C^{\wedge}D \vee \wedge B^{\wedge}C^{\wedge}D \vee B^{\wedge}C^{\wedge}D;$$

$$f_2 = \wedge B^{\wedge}C^{\wedge}D \vee B^{\wedge}C^{\wedge}D \vee B^{\wedge}C^{\wedge}D \vee B^{\wedge}C^{\wedge}D.$$

Аналогично разложим функции  $f_1$  и  $f_2$  по переменным С,  $\wedge C$  и Д,  $\wedge D$  соответственно:

$$f_1=\wedge C^{\wedge}(\wedge B^{\wedge}\wedge D \vee \wedge B^{\wedge}D \vee B^{\wedge}\wedge D) \vee C^{\wedge}(\wedge B^{\wedge}\wedge D \vee \wedge B^{\wedge}D) =\wedge C^{\wedge}(\wedge B^{\wedge}(\wedge D \vee D) \vee B^{\wedge}\wedge D) \vee C^{\wedge}(\wedge B^{\wedge}(\wedge D \vee D)) =\wedge C^{\wedge}(\wedge B^{\wedge}1 \vee B^{\wedge}\wedge D) \vee C^{\wedge}B; \dots(4)$$

$$f_2=\wedge D^{\wedge}(\wedge B^{\wedge}\wedge C \vee B^{\wedge}\wedge C) \vee D^{\wedge}(B^{\wedge}\wedge C \vee B^{\wedge}C)=\wedge D^{\wedge}\wedge C \vee D^{\wedge}B. \dots (5)$$

Выражения (3), (4) и (5) позволяют построить принципиальную схему комбинационного устройства на MS2-1 для функции (2) - рис. 3. Схема содержит четыре мультиплексора MS2-1. Первый реализует зависимость (3), второй и четвёртый реализуют функцию  $f_1$ , а третий мультиплексор – функцию  $f_2$ .

При построении комбинационного устройства для реализации функции (2) на базе MS4-1 в исходной функции сгруппируем члены, вынося за знак скобок произведение двух переменных, (например А и В), следующим образом:



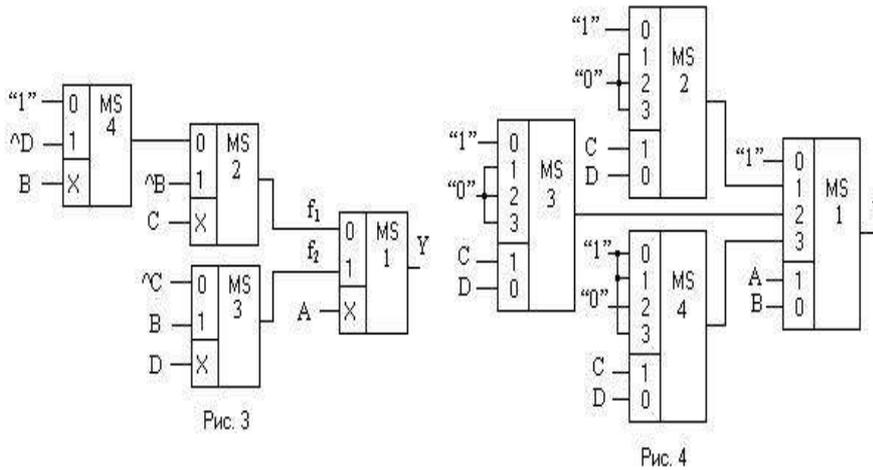
Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

$$Y = A^* B^* (C^* D \vee C^* D \vee C^* D \vee C^* D) \vee A^* B^* (C^* D) \vee A^* B^* (C^* D) \vee A^* B^* (C^* D \vee C^* D \vee C^* D) = \\ = A^* B^* I \vee A^* B^* (C^* D) \vee A^* B^* (C^* D) \vee A^* B^* (C^* D \vee C^* D \vee C^* D). \quad \dots \quad (6)$$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Полученное выражение позволяет составить принципиальную схему комбинационного устройства для этого случая - рис. 4. На информационные входы MS1 (в соответствии с их адресом, определяемым переменными A и B)



подаются соответствующие функции (в скобках). Адрес  $A^*B$  определяет нулевой информационный вход, поэтому сюда подводится логическая 1 (в соответствии с выражением для первого термина функции (6)). Адрес  $A^*B$  определяет первый информационный вход, поэтому к нему подключён выход MS2, реализующего зависимость  $(C^*D)$ . Адрес  $A^*B$  определяет второй информационный вход MS1, к которому подведён выход MS3, реализующего зависимость  $(C^*D)$ . Адрес  $A^*B$  определяет третий информационный вход MS1, на который подаётся сигнал с выхода MS4, реализующего зависимость  $(C^*D \vee C^*D \vee C^*D)$ .

Рассмотрим построение комбинационной схемы на базе мультиплексора MS8-1. Для этого функцию (2) необходимо разложить по трём переменным, например, по A, B, C, сочетания прямых или инверсных значений которых будут определять в мультиплексоре MS8-1 адреса определённых информационных входов. Сгруппируем в выражении (2) члены с одними и теми же адресами информационных входов, вынося их за скобки:

$$Y = A^*B^*C^*(^D \vee D) \vee A^*B^*C^*(^D \vee D) \vee A^*B^*C^*(^D) \vee A^*B^*C^*(D) \vee A^*B^*C^*(^D \vee D) \vee A^*B^*C^*(D) \quad (7)$$

Здесь адрес  $A^*B^*C$  определяет информационный вход MS8-1 за номером 0; адрес  $A^*B^*C$  определяет информационный вход MS8-1 за номером 1 и т.д.. Следует отметить, что в выражении (7) отсутствуют комбинации  $A^*B^*C$  и  $A^*B^*C$ , определяющие третий и пятый информационные входы MS8-1. Выражения в круглых скобках (7) определяют подаваемый на данный вход сигнал. То есть сигнал на нулевой информационный вход  $(^D \vee D)=1$  обеспечивается подачей на него логической 1. Та же самая ситуация с первым информационным входом. На второй информационный вход с адресом  $A^*B^*C$  в соответствии с (7) необходимо подать сигнал  $(^D)$ . И так далее до входа и номером 7. На те адреса, которые отсутствуют в (7) – третий и пятый, необходимо подать логические 0. С учётом всех этих замечаний выражение (7) можно записать так:

$$Y = A^*B^*C^*(1) \vee A^*B^*C^*(1) \vee A^*B^*C^*(^D) \vee A^*B^*C^*(0) \vee A^*B^*C^*(D) \vee A^*B^*C^*(0) \vee A^*B^*C^*(1) \vee A^*B^*C^*(D) \dots \quad (8)$$

Таблица 1

| Адрес инф. входа |       | Подаваемое значение |
|------------------|-------|---------------------|
| Код              | Номер |                     |
| $A^*B^*C$        | 0     | $(^D \vee D)=1$     |
| $A^*B^*C$        | 1     | $(^D \vee D)=1$     |
| $A^*B^*C$        | 2     | $(^D)$              |
| $A^*B^*C$        | 3     | (0)                 |
| $A^*B^*C$        | 4     | (D)                 |
| $A^*B^*C$        | 5     | (0)                 |
| $A^*B^*C$        | 6     | $(^D \vee D)=1$     |
| $A^*B^*C$        | 7     | (D)                 |

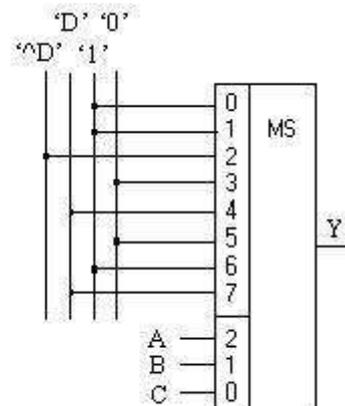


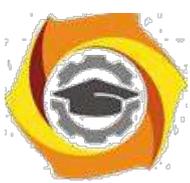
Рис. 5

Порядок программирования MS8-1 для реализации функции (2) приведён в табл.1. На рис. 5 показана принципиальная схема реализации функции (2) на мультиплексоре MS8-1. Выбор адреса того или иного канала



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

обеспечивается переменными А, В, С, подаваемыми на адресные входы 2, 1, 0. Соединение информационных входов с шинами “ $\wedge D$ ”, “D”, “1”, “0” выполнено по табл. 1.



х данной лабораторной работе для реализации функции на MS8-1 используются два MS4-1, которые выполнены в виде одной микросхемы **74153 ( 2 4 -to -1 Data Selectors/Mux )** из контейнера **Combinational (шаблон MUX)**. Обозначения на корпусе для первого MS4-1: 1C0,1C1,1C2,1C3 – информационные входы; 1Y- выход; А и В – входы сигналов управления. Для второго MS4-1: 2C0,2C1,2C2,2C3 – информационные входы; 2Y - выход; А и В – входы сигналов управления. Для объединения двух MS4-1 в один используются дополнительные входы 1G и 2G – разрешение работы 1-го и 2-го MS4-1. Для объединения 2-х MS4-1 в один MS8-1 использованы дополнительные элементы: инвертор, через который старший разряд А подаётся на вход 2G - разрешения второго MS4-1, и элемент ИЛИ, объединяющий выходы мультимплексов. Пример подсоединения микросхемы 74153 в качестве

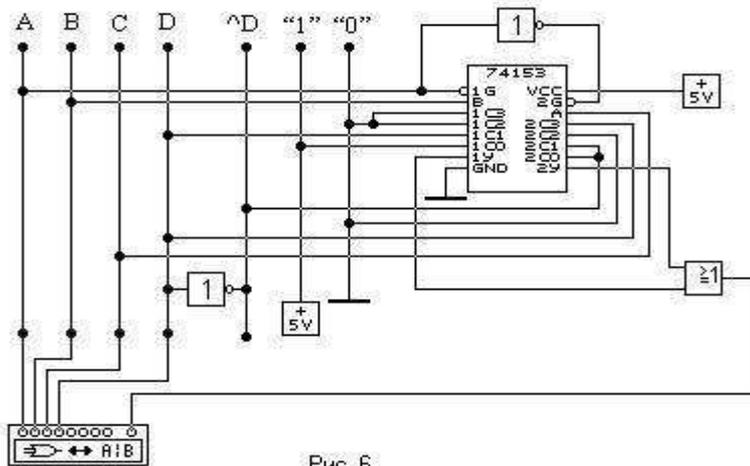


Рис. 6

мультимплексора MS8-1 для исследования с помощью логического преобразователя реализации некоторой булевой функции  $F(A,B,C,D)$  приведён на рис. 6.

Для исследования одновыходных логических схем будем использовать логический преобразователь – устройство, обеспечивающее взаимные преобразования различных способов представления логических схем. С его помощью можно:

- по таблице истинности получить логическое выражение;
- по логическому выражению получить логическую схему или таблицу истинности;
- по логической схеме построить таблицу истинности.

Прибор предусматривает возможность использования до восьми входных логических переменных от А до Н, подключаемых к входам исследуемой схемы, и вход **OUT**, к которому должен быть подключён выход исследуемой комбинационной схемы. Для активации необходимой переменной нужно щёлкнуть расположенный над ней кружок, после чего он станет тёмным. Если таблицу истинности задаёт пользователь, то после выбора входных переменных нужно активизировать переменную OUT и для каждого набора значений входных переменных задать значение выходной логической переменной. Подробная инструкция работы с этим прибором изложена в Программе Electronic Workbench для анализа электронных схем: Учебно-метод. пособие. В.М. Чухонцев. Самар. гос. тех. ун-т. Самара, 1999.- 92с.

### 5. 2. МЕТОДИКА ВЫПОЛНЕНИЯ

- в Включите необходимое программное обеспечение.
- в Используя логические элементы НЕ, И и ИЛИ из контейнера **Gates**, соберите схемы мультимплексов **MS2-1, MS4-1** согласно рис. 1,а и рис. 2,а. При этом собранные Вами схемы необходимо сохранить в виде субсхем. Для этого выделите собранную Вами схему с помощью мыши и нажмите клавиши **Ctrl** и **B**; введите имена Ваших субсхем **ms2-1** и **ms4-1** соответственно и нажмите кнопку **Copy from Circuit**. В результате схемы мультимплексов в виде отдельных субсхем будут находиться в контейнере **Custom**.

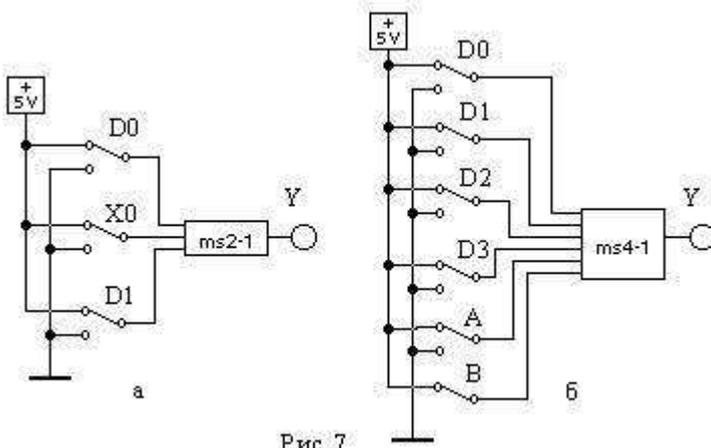


Рис. 7

Соберите схемы по рис. 7 для проверки работы субсхемы ms2-1 (рис. 7,а) и субсхемы ms4-1(рис. 7,б). Задавая управляющими сигналами номер того или иного информационного входа убедитесь в появлении на выходе субсхемы сигнала именно нужного параметра Di. Результаты эксперимента сведите в табл.

13 Используя разложение Шеннона по 1-й, 2-м и 3-м переменным, постройте схемы, реализующие заданную преподавателем функцию из приведённых ниже вариантов, на MS2-1, MS4-1 и MS8-1. Проверьте правильность выполнения задания с помощью логического преобразователя, таблицы истинности на экране которого для всех трёх вариантов схемной реализации должны совпадать с заданным вариантом.

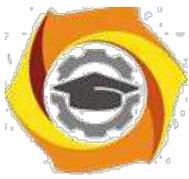
- |                         |                            |
|-------------------------|----------------------------|
| 1. $Y=U(0,5,15,10,8,9)$ | 6. $Y=U(0,15,1,10,8,11,2)$ |
| 2. $Y=U(0,1,5,4,8,14)$  | 7. $Y=U(6,5,7,14,15,10,8)$ |
| 3. $Y=U(4,5,7,15,11,9)$ | 8. $Y=U(7,0,5,15,14,13)$   |
| 4. $Y=U(2,3,4,5,13,9)$  | 9. $Y=U(0,2,3,5,7,12,14)$  |
| 5. $Y=U(1,2,5,7,12,13)$ | 10. $Y=U(10,15,3,9,6,4,0)$ |



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Числа в скобках задаваемых функций определяют те строки таблицы истинности, в которых последняя принимает значение логической единицы.

Номер варианта соответствует номеру бригады на лабораторных занятиях.



14. Определите функцию  $F(A,B,C,D)$ , реализуемую схемой на рис. 6, с помощью разложения Шеннона.

### 5.3. КОНТРОЛЬНЫЕ ВОПРОСЫ

22. Что такое мультиплексор и каковы области его применения?  
23. Поясните принцип работы мультиплексора на примере MS4-1.  
24. Объясните механизм разложения Шеннона булевой функции при разработке электрической схемы её реализации.  
25. Какие возможности обеспечивает логический преобразователь при исследовании цифровых схем?

Контрольные вопросы ПЗ5 (ПК-4):

27. Охарактеризуйте основные функции устройства управления.  
28. Какие этапы включаются в машинный цикл команды типа «Сложение»?  
29. Какие этапы входят в машинный цикл команды типа «Запись»? Обоснуйте отсутствие одного из этапов.  
30. Какие этапы входят в машинный цикл команды типа «Условный переход»? Обоснуйте отсутствие двух этапов.  
31. Дайте характеристику входной и выходной информации модели УУ.  
32. На какие две части делится структура УУ? Что входит в состав каждой части? Какое назначение имеют элементы частей УУ?  
33. Перечислите достоинства и недостатки МПА с жесткой логикой.  
34. Дайте характеристику элементов структуры МПА с программируемой логикой.  
35. Объясните принцип управления на основе МПА с программируемой логикой.  
36. Какие способы кодирования микрокоманд вы знаете? Перечислите их достоинства и недостатки.  
37. Поясните подходы к адресации микрокоманд, охарактеризуйте их сильные и слабые стороны.  
38. Какие способы организации памяти микрокоманд вы знаете, чем они обусловлены?  
39. В каких случаях в МПА следует применять ПЛМ, а не ПЗУ? Ответ аргументируйте.  
40. Сформулируйте числовые параметры МПА с программируемой логикой. От чего они зависят?  
41. Поясните методику минимизации количества слов памяти микрокоманд.  
42. Охарактеризуйте методику минимизации разрядности микрокоманд.  
43. Какие подходы к повышению быстродействия МПА с программируемой логикой  
44. Поясните достоинства и недостатки трех вариантов подключения системы ввода/вывода к процессору ВМ.  
45. Сформулируйте достоинства, недостатки и область применения двух способов организации адресного пространства ввода/вывода  
46. В чем состоит локализация данных, выполняемая модулем ввода/вывода?  
47. Опишите содержание процедуры «рукопожатия» при выполнении операции ввода.  
48. Конкретизируйте последовательность действий процессора при обмене информацией с жестким диском.  
49. Сравните известные вам методы управления вводом/выводом по трем параметрам: достоинствам, недостаткам, области применения.  
50. Поясните классификацию методов ввода/вывода по прерыванию.  
51. Охарактеризуйте известные вам режимы прямого доступа к памяти, сформулируйте их достоинства и недостатки.  
52. Сравните ввод/вывод по прерыванию с вводом/выводом при ПДП. Для какого режима ПДП эти методы наиболее близки и почему?  
53. Проведите сравнительный анализ контроллера ПДП и канала ввода/вывода. В чем их сходство? Чем они отличаются друг от друга?  
54. Опишите процесс взаимодействия ЦП и КВВ. Какая при этом используется управляющая информация?  
55. Опишите задачи посредника между КВВ и ВУ.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

56. Как формируется код символа в клавиатуре? Как он передается память компьютера?

57. Каково назначение мыши? На чем основана работа оптической мыши? Поясните принцип



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

24

ее работы.

- В чем принцип работы дисплея на базе ЭЛТ? Что такое частота регенерации изображения и чему она равна в современных дисплеях?
- Как работает ЖК-дисплей? Что такое разрешающая способность? Какой разрешающей способностью обладают современные дисплеи?
- Какие принтеры в настоящее время наиболее распространены? Какая разрешающая способность для них характерна?
- Как работает струйный принтер? Какими достоинствами и недостатками он обладает?
- Как работает лазерный принтер? Перечислите все этапы формирования изображения.



## ПРАКТИЧЕСКАЯ РАБОТА № 6.

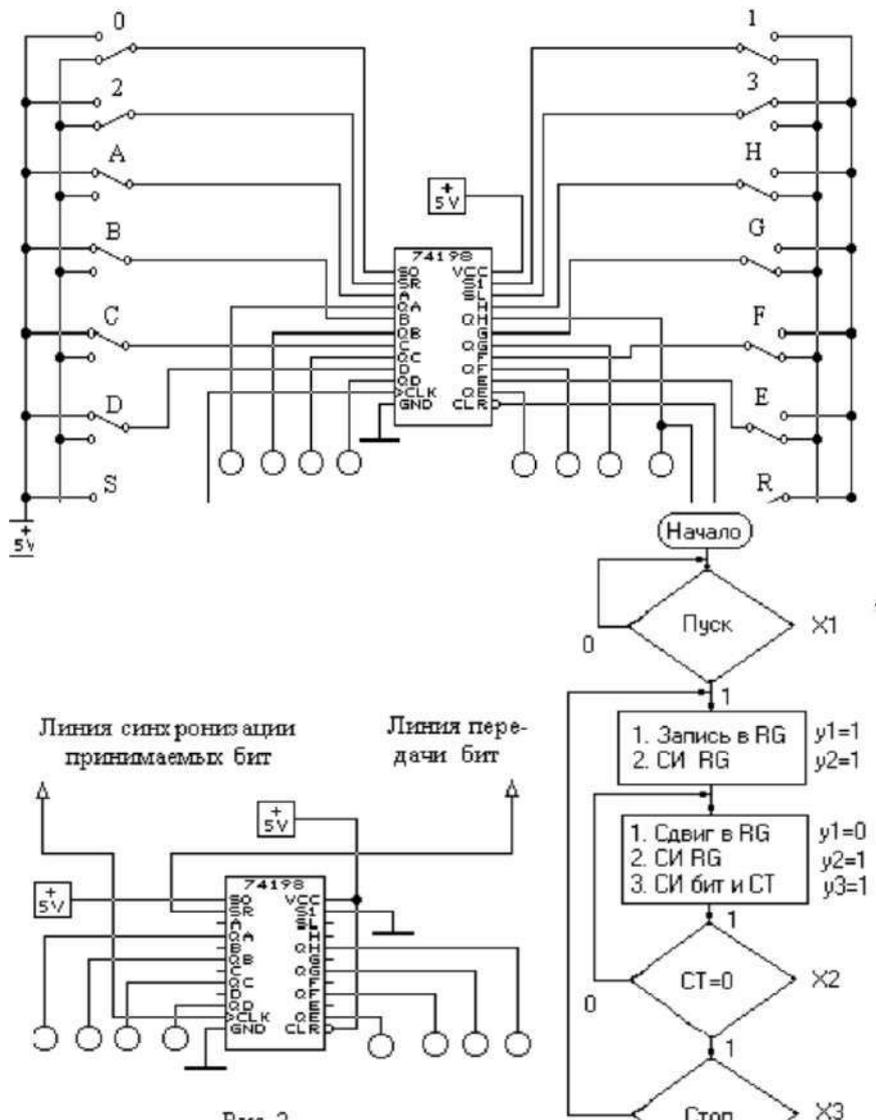
### Процессоры, контроллеры, внешние устройства

#### 6.1. КРАТКАЯ ТЕОРИЯ

При передаче данных по линиям связи или между устройствами ЭВМ часто используют преобразование параллельного кода (байта) в последовательный (биты) и обратное преобразование в параллельный код. Преобразование параллельного кода в последовательный производится с помощью сдвигающего регистра: сначала в него записывают байт (параллельная запись), а затем переводят регистр в режим сдвига и подают 8 импульсов синхронизации. Таким образом, на выходе одного из разрядов регистра (старшего или младшего в зависимости от направления сдвига) последовательно появляются значения всех разрядов записанного байта. Эта последовательность по линии связи передается в приемник, главным элементом которого является такой же сдвигающий регистр. Через последовательный вход (по импульсам синхронизации) информация последовательно вводится в регистр приемника. Через 8 импульсов синхронизации передаваемый байт в регистре приемника готов для дальнейшей обработки (обычно - записи в буферный регистр). Подробнее о регистрах можно почитать в разделе «Краткая теория» лабораторной работы № 4,

#### 6.2. МЕТОДИКА ВЫПОЛНЕНИЯ

На рис.1 приведена схема простейшего передатчика с ручным управлением. Она собрана на 8-ми битовом универсальном регистре типа 74198 из библиотеки **default** модели **ideal** (контейнер **Sequential**). Значение передаваемого байта ABCDEFGH задаётся с помощью переключателей. Режим работы регистра (запись, хранение, сдвиг влево, сдвиг вправо) задаётся с помощью переключателей (0, 1), соединенных с управляющими входами (S0,S1). Состояние разрядов регистра отображается индикаторными лампочками. С помощью переключателей (0, 1) можно задать режим параллельной записи в регистр передатчика и записать в него байт, затем задать режим сдвига и передать этот байт (последовательно по битам) по линии связи в приемник (рис. 2). Приемник соединен с передатчиком 2-мя





линиями: по одной передаются биты, по другой - СИ (импульсы синхронизации бит).

Алгоритм работы передатчика приведен на рис. 3. В нем присутствуют условия “Пуск” и “Стоп”. По сигналу “Пуск” (кнопка) начинается работа передатчика. В начале регистр переводится в режим параллельной записи (Запись в RG), на входы параллельной загрузки подается байт, а затем - СИ на вход синхронизации регистра. Для передачи данных регистр переводится в режим сдвига, вырабатывается СИ бит (который передается в приемник и используется также для подсчета количества переданных бит) и СИ RG (син-хроимпульс сдвига). Следует обратить внимание на последовательность формирования фронтов СИ бит и СИ RG: сначала необходимо передать СИ бит в приемник, а затем сдвинуть содержимое регистра передатчика. Счетчик переданных бит изменяет свое состояние (+1) на каждом СИ бит. После передачи последнего 8-го бита содержимое счетчика (по MOD 8) становится равным нулю. Если нет команды “СТОП”, то начинается цикл передачи очередного байта.

На основании приведенного алгоритма необходимо спроектировать управляющий автомат, входами которого будут команды (кнопки) “ПУСК” и “СТОП”, а также логическое условие “СТ=0”. В целом устройство будет состоять из 2-х автоматов - операционного (ОА) и управляющего (УА). ОА будет содержать 2 основных операционных элемента: универсальный регистр и двоичный счетчик по модулю 8. Эти элементы необходимо выбрать самостоятельно из контейнера **Sequential**. Управляющий автомат следует спроектировать как автомат МУРА.

Спроектированный автомат УА следует собрать, проверить правильность его функционирования согласно исходной ГСА и оформить в виде подсхемы АWT. Для удобства отладки автомата максимально используйте

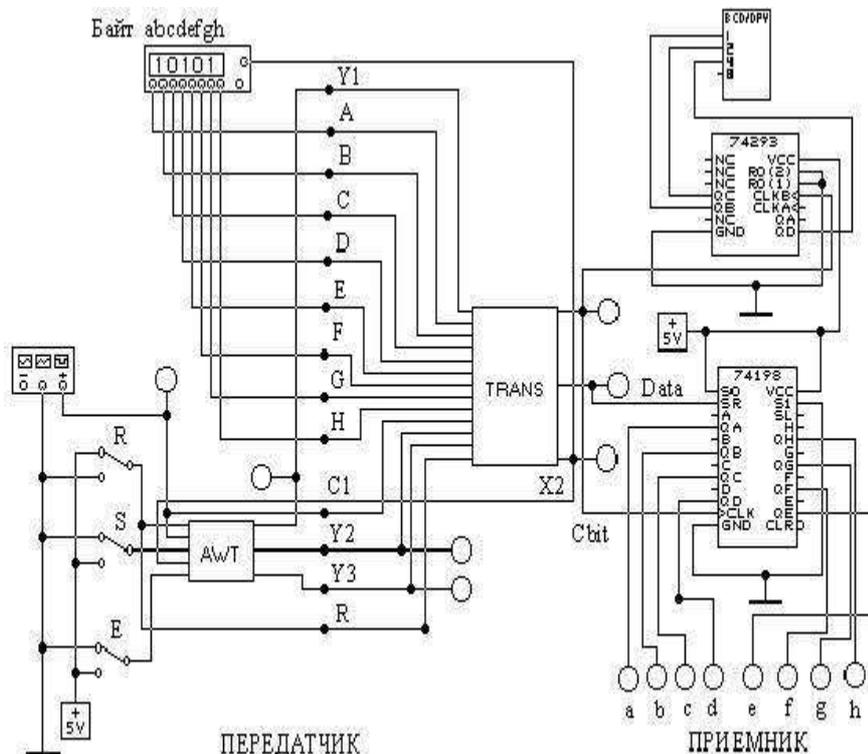


Рис. 4

элементы индикации (в том числе и цифровые индикаторы).

Перед подключением операционных элементов ОА к УА постройте временные диаграммы, из которых должна быть наглядно видна последовательность всех формируемых сигналов. В ОА для обеспечения этой последовательности можно использовать простейшие логические элементы И, ИЛИ, НЕ и т.п. При наладке ОА также используйте элементы индикации. Оформите отлаженный ОА в виде подсхемы TRANS. В качестве источника СИ для ОА и УА используйте FUNCTION GENERATOR, а источника передаваемых байт - WORD GENERATOR, который можно синхронизировать, например, логическим условием X2.

Схема приемника значительно проще схемы передатчика. На рис. 4 приведен один из вариантов общей схемы - передатчик + приемник.

приемнике кроме сдвигающего регистра показан счетчик принятых бит и индикатор количества принятых бит. Регистр приемника всегда работает в одном режиме - сдвиг вправо. В передатчике используются кнопки:

- R - начальный сброс всех элементов передатчика;
- S - Старт; кнопка должна быть нажата в течении 2 СИ;
- E - Стоп.

Частоту генератора СИ следует выбрать 0.5 ... 1 Гц, чтобы можно было успевать наблюдать за работой устройства. При отладке генератор можно заменить на кнопку.

#### ЗАДАНИЕ



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Соберите схемы (рис. 1 и рис. 2) и в ручном режиме проанализируйте работу каждой из них. Экспериментально определите сочетания управляющих команд для реализации различных режимов регистров.



- И Разработайте и соберите УА, затем ОА и устройство передачи-приёма в целом. Убедитесь в правильности работы УА и ОА.
- И Представьте принципиальные схемы Уа и ОА.
- И Нарисуйте временные диаграммы, иллюстрирующие работу устройства.

### **6.3. КОНТРОЛЬНЫЕ ВОПРОСЫ**

- ее Опишите работу схемы передатчика по рис. 1.
- ее Опишите работу схемы приёмника по рис. 2.
- ее Представьте и опишите работу субсхем управляющего и операционных автоматов передатчика.
- ее Поясните работу передающего-приёмного устройства по рис. 3.
- ее Каково назначение счётчика в цепи приёмника?

Контрольные вопросы ПЗ6 (ПК-4):

- В В чем суть идеи конвейеризации? В каких случаях в конвейер следует вводить буферные регистры? В каких случаях буферные регистры нужно заменять буферной памятью? Ответы обоснуйте.
- 23. В чем суть статического предсказания переходов? Сформулируйте достоинства и недостатки известных способов статического предсказания переходов.
- 24. В чем заключается смысл динамического предсказания переходов? Дайте развернутую характеристику достоинств и недостатков известных способов динамического предсказания переходов. Из каких соображений следует выбирать конкретную схему динамического предсказания? От чего зависит выбор?
- 25. Поясните идею суперконвейера. В чем заключаются достоинства и недостатки суперконвейеризации?
- 26. Поясните достоинства и недостатки ВМ с полным набором команд. Какие исторические причины привели к их возникновению?
- 27. Какие исторические причины способствовали появлению ВМ с сокращенным набором команд?
- 28. Перечислите основные характеристики ВМ с сокращенным набором команд.
- 29. Опишите возможности совместного использования в одной ВМ CISC-архитектуры и RISC-архитектуры.
- 30. Для чего вводится механизм регистровых окон? Поясните структуру окна. Ради какой цели окна организуются в виде циклического буфера?
- 31. Обоснуйте основные недостатки ВМ с сокращенным набором команд.
- 32. Дайте развернутую характеристику назначения и структурной организации суперскалярного процессора. Какие уровни параллелизма здесь используются?
- 33. На конкретных примерах поясните суть проблемы неупорядоченности команд в суперскалярных процессорах.
- 25. На примере конкретного программного фрагмента поясните суть метода переименования регистров. В чем состоят недостатки этого метода? Как их смягчить?
- 26. На примере конкретного программного фрагмента поясните смысл метода переупорядочивания команд. В чем заключается разница между централизованным и распределенным окном команд? Проведите сравнительный анализ организации этих окон.
- 27. Опишите назначение, организацию и порядок работы буфера восстановления последовательности. Для решения каких задач он применяется в современных процессорах?
- 28. Каким образом и при каких условиях гиперпоточковая обработка способствует повышению производительности процессора?
- 29. Сформулируйте правила объединения простых команд в командное слово сверхбольшой длины.



## Политология

Методические указания по практическим занятиям  
*Практическое занятие 1*

### ТЕОРЕТИКО-МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ ПОЛИТОЛОГИИ

**Предмет политологии.** Содержание мира политики. Структура и функции политики. Политология как наука о политике, власти, политических системах и процессах. Взаимодействие политологии с другими общественными науками: философией, историей, экономической теорией, социологией, теорией государства и права. Основные этапы развития политической науки. Политология в современной России.

**Методические основы и функции политологии.** Фундаментальный и прикладной аспекты политологии. Основные задачи и функции политологии. Методы политологических исследований.

**Основные парадигмы и школы политологии.** Теологическая, натуралистическая, социальная и рационально-критическая парадигмы. Основные доктрины и направления современной зарубежной политологии. Американская, европейская и российская политологические школы.

### Контрольные вопросы

- Дайте определение “политики”.
- Что составляет содержание мира политики?
- Дайте определение политологии.
- Назовите основные методы политических исследований. Что составляет предметную область политической науки?
- Назовите функции политологии.
- Определите место политологии в системе современных гуманитарных знаний?
- Каково значение политологических знаний в современном мире?

### Темы рефератов

Политическая наука в современной России.  
Основные парадигмы и школы политологии.  
Соотношение политики и права.  
Соотношение политики и экономики.  
Политика как сфера общественной жизни.

### *Практическое занятие 2*

### ИСТОРИЯ ПОЛИТИЧЕСКИХ УЧЕНИЙ

**Политическая мысль Древнего мира и Средневековья.** Особенности политических учений Древнего Востока. Основные этапы развития политической мысли Древней Греции. Политическое учение Платона. Аристотель о политике и государстве. Полити-



## Негосударственное частное образовательное учреждение высшего образования «Технический университет»

ческие воззрения римских мыслителей. Историческое значение античной политической мысли.

Мировоззренческие особенности политической мысли европейского средневековья. Господство теократических теорий. Освобождение политической мысли от религиозных догм как характерная черта мировоззрения позднего средневековья. Учение о гос-ударстве, власти и суверенитете Ж.Бодена.

**Политическая мысль Нового времени.** Н.Макиавелли - родоначальник политической науки. Учение о государстве, власти и суверенитете Ж.Бодена. Политические учения раннего утопического социализма (Т.Мор, Т.Кампанелла). Идеи происхождения государства в учении Г.Гроция. Политическая теория либерализма (Т.Гоббс, Д.Локк). Теория разделения властей Ш.-Л. Монтескье. Особенности политического мировоззрения просветителей. Политические взгляды Вольтера. Учение Ж.-Ж. Руссо о демократии, общественном договоре и народном суверенитете. Социалистические идеи эпохи Просвещения. Основные идейно-политические течения в США в период войны за независимость. Модели государственного устройства в американской политической мысли.

Влияние процесса развития капиталистических отношений в Европе на изменение политической идеологии. Возникновение и основные этапы развития марксистского учения. Марксистское учение о государстве и его особенности. Политическая теория анархизма.

**Основные тенденции развития политической мысли в XX веке.** Развитие марксистской политической доктрины в конце XIX начале XX веков. Разновидности немарксистских социалистических концепций, их общие черты и особенности. Западноевропейская социал-демократия и формирование идейно-политической платформы ревизионизма и реформизма. Теории элит (В.Парето, Г.Моска, Р.Михельс). Сущность и социальная направленность теории технократии.

**История политической мысли России.** Предпосылки и условия формирования древнерусской политической мысли. Социально-политические идеи периода создания и укрепления Русского централизованного государства. Развитие представлений о сильной самодержавной власти в русской политико-правовой мысли XVI века. Политические идеи русского просвещения. Зарождение революционно-демократической традиции в русской политической мысли XIX века. Истоки либеральной традиции. Особенности русского либерализма. Концепции политического развития России: славянофилы - западники. Русский анархизм о политике и государстве. Основные течения народничества. Русский консерватизм. Религиозно-нравственная традиция в русской политической мысли. Ленинское учение о государстве, революции и диктатуре пролетариата. Российская политическая мысль в поисках общегосударственной идеи и выхода из кризисной ситуации.

### Контрольные вопросы

6. Какой характер носили первые политические воззрения?
7. В чем проявилось несовершенство политических учений Древнего Востока?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

7. Какие типы государств Платон относил к “правильным” и “неправильным” (“иска-женным”)?
8. Каким представлял себе наилучшее политическое устройство Аристотель?
9. Чем отличаются политические воззрения римских мыслителей от греческих?
10. Какая форма правления была политическим идеалом Н.Макиавелли?
11. Каково содержание теории государственного суверенитета Ж.Бодена?
12. Как характеризовал наилучший общественный строй Т.Кампанелла?
13. Что оказывает влияние на форму организации государственной власти, по мнению Ш.-Л. Монтескье?
14. Какую форму государственного устройства Ж.-Ж. Руссо считал идеальной?
15. В чем заключается сущность “теории фракций”?
16. Назовите принципы правового государства, выделенные И.Кантом?
17. Как рассматривает Г.Гегель гражданское общество и государство?
18. Какие этапы можно выделить в истории утопического социализма?
19. Назовите основные положения теории К.Маркса?
20. Как определял К.Маркс политическую власть?
21. Каковы взгляды К.Маркса на государство?
22. Что нового внес М.Вебер в исследование проблем власти?
23. Каковы основные положения теории “Москва - Третий Рим”?
24. Сторонником какого государственного строя был В.Н.Татищев?

### **Темы рефератов**

Основные идеи триады Платона: “Политик”, “Законы”, “Государство”.

Проблема государственного устройства в трудах Аристотеля.

Проекты идеальной организации общества в “Государстве” Платона и “Политике” Аристотеля.

Политические взгляды Цицерона в его работах “О государстве” и “О законах”.

Основные идеи трактата Н.Макиавелли “Государь”.

Д.Локк о государственном правлении в работе “Два трактата о правлении”.

### *Практическое занятие 3*

#### **ТЕОРИЯ ПОЛИТИЧЕСКОЙ ВЛАСТИ**

**Сущность и природа власти.** Происхождение и сущность власти. Концепции власти. Структура власти. Основания власти.

**Политическая власть.** Понятие “политическая власть”. Механизм осуществления политической власти. Горизонтальная и вертикальная структура политической власти в обществе. Принцип разделения властей. Легальность и легитимность политической власти. Типы легитимного господства по М.Веберу. Уровни легитимации власти. Суверенитет власти.

**Субъекты политической власти.** Соотношение понятий “элита” и “политическая элита”. Политическая элита как необходимое структурное звено в механизме политиче-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

ской власти. Теории элит. Основные черты и структура политической элиты в России. Политическое лидерство: природа, функции, типы и стили. Политическое лидерство в современной России.

### **Контрольные вопросы**

5. В чем сущность власти?
6. Дайте определение “политической власти”.
7. Назовите источники и ресурсы власти.
8. Какие функции власти Вам известны?
9. Что такое “легитимность власти”?
10. Каковы основные положения теории элит?
11. В чем заключается сущность “железного закона олигархии” Р.Михельса?
12. Какова роль масс в политической концепции Р.Михельса?
13. Что вкладывает в понятие “правлящий класс” Г.Моска?

### **Темы рефератов**

Происхождение власти и ее источники.  
Лоббизм в современной России.  
Теории элит В.Парето, Г.Моска: сравнительный анализ.  
“Железный закон” олигархических тенденций Р.Михельса.  
Современные концепции элиты.

### *Практическое занятие 4*

## **ПОЛИТИЧЕСКИЕ ПРОЦЕССЫ**

**Политический процесс.** Определение понятия, виды и способы осуществления.

**Политическое развитие.** Понятие “политической развитие”. Критерии политического развития.

**Политическая модернизация.** Концепции модернизации в современной политической науке. Типы модернизации. Сущность политической модернизации. Тенденции и перспективы политической модернизации в России.

**Политические кризисы и конфликты.** Политические кризисы: Сущность и понятие. Разновидности политического кризиса. Причины и природа политических конфликтов. Типы политических конфликтов: конфликты интересов, конфликты ценностей, конфликты идентификаций. Методы разрешения конфликтов. Сущность этнополитических конфликтов, их виды. Способы разрешения и урегулирования.

**Политическая активность и политическое участие.** Типы политического участия. Ортодоксальное и неортодоксальное политическое участие. Конвенциональное и неконвенциональное политическое участие. Теории политического участия. Электоральное поведение. Политический протест. Политический терроризм. Природа абсен-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

теизма.

### **Контрольные вопросы**

7. Дайте определение политическому развитию.
8. Назовите основные группы противоречий политической модернизации.
9. Что такое “политический конфликт”?
10. Назовите стадии политического конфликта.
11. Каковы функции и методы разрешения социально-политических конфликтов?
12. Назовите субъектов этнополитического конфликта.
13. Определите понятие “политического риска”.
14. Назовите виды политического участия.

### **Темы рефератов**

Модели политической модернизации.  
Перспективы политической модернизации в России.  
Типы политических конфликтов.  
Проблемы урегулирования этнополитических конфликтов.  
Особенности электорального поведения россиян.

### *Практическое занятие 5*

## **ПОЛИТИЧЕСКОЕ СОЗНАНИЕ И КУЛЬТУРА**

**Политическое сознание.** Политическое сознание как отражение бытия политической жизни. Сущность, структура и уровни политического сознания. Массовое и индивидуальное политическое сознание. Политические ценности, потребности и интересы.

**Политическая социализация.** Содержание политической социализации: прямая и косвенная политическая социализация. Модели политической социализации: системная, гегемонистская, плюралистическая, конфликтная. Проблемы политической социализации в России.

**Политическая культура.** Понятие политической культуры. Ее роль в обществе. Структура и функции. Типы политической культуры: патриархальная, подданническая, активистская. Политические субкультуры. Специфика российской политической культуры: история и современность.

**Политические идеологии.** Подходы к пониманию термина “идеология”. Возникновение политических идеологий. Типы идеологий. Основные функции политических идеологий.

### **Контрольные вопросы**

7. Дайте определение “политической социализации”.
8. Что означает понятие “абсентеизм”?
9. Каковы структура и функции политической культуры?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

9. Назовите типы политических культур.

10. Какие типы политических идеологий Вам известны?

### **Темы рефератов**

Политическая социализация личности: сущность факторы, механизмы.

Специфика российской политической культуры.

Эволюция либеральной политической идеологии.

Феномен политического абсентеизма.

### *Практическое занятие 6*

## **ГЕОПОЛИТИЧЕСКИЙ АНАЛИЗ СОВРЕМЕННЫХ МЕЖДУНАРОДНЫХ ОТНОШЕНИЙ.**

**Геополитика: теоретические основы изучения и проблемы практики.** Подходы к понятию “геополитика”. Сущность и современное понимание геополитической теории. Структура и функции геополитики. Возрастание роли геополитики в современном ми-ре. Многополярность как новая геополитическая модель мира. Место России в совре-менной геополитической картине мира. Америка и Европа: формирование новой гео-политической обстановки.

**Международные отношения и международная политика.** Особенности современно-го этапа международных отношений. Основные факторы внешней политики. Взаимо-связь внутренней и внешней политики. Защита национально-государственных интере-сов и обеспечение национальной безопасности.

### **Контрольные вопросы**

11. Что такое “геополитика”?
12. Каковы основные функции геополитики?
13. Дайте определение “географическому детерминизму”.
14. Что такое турбулентность?
15. Какие существуют концепции национальной безопасности?
16. Дайте определение понятия “национальный интерес”.
17. Назовите основные приоритеты национальной безопасности России.
18. Какие геополитические регионы современного мира Вам известны?

### **Темы рефератов**

Национальные интересы России.

Российские геополитические традиции.

Геополитические регионы современного мира.



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

**Заочная форма обучения**

*Практическое занятие 1*

**ТЕОРЕТИКО-МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ ПОЛИТОЛОГИИ**

**Предмет политологии.** Содержание мира политики. Структура и функции политики. Политология как наука о политике, власти, политических системах и процессах. Взаимодействие политологии с другими общественными науками: философией, историей, экономической теорией, социологией, теорией государства и права. Основные этапы развития политической науки. Политология в современной России.

**Методические основы и функции политологии.** Фундаментальный и прикладной аспекты политологии. Основные задачи и функции политологии. Методы политологических исследований.

**Основные парадигмы и школы политологии.** Теологическая, натуралистическая, социальная и рационально-критическая парадигмы. Основные доктрины и направления современной зарубежной политологии. Американская, европейская и российская политологические школы.

**Контрольные вопросы**

6. Дайте определение “политики”.
7. Что составляет содержание мира политики?
8. Дайте определение политологии.
9. Назовите основные методы политических исследований. Что составляет предметную область политической науки?
10. Назовите функции политологии.
11. Определите место политологии в системе современных гуманитарных знаний?
12. Каково значение политологических знаний в современном мире?

**Темы рефератов**

Политическая наука в современной России.  
Основные парадигмы и школы политологии.  
Соотношение политики и права.  
Соотношение политики и экономики.  
Политика как сфера общественной жизни.

*Практическое занятие 2*

**ПОЛИТИЧЕСКИЕ ПРОЦЕССЫ**

**Политический процесс.** Определение понятия, виды и способы осуществления.

**Политическое развитие.** Понятие “политическое развитие”. Критерии политического



развития.

**Политическая модернизация.** Концепции модернизации в современной политической науке. Типы модернизации. Сущность политической модернизации. Тенденции и перспективы политической модернизации в России.

**Политические кризисы и конфликты.** Политические кризисы: Сущность и понятие. Разновидности политического кризиса. Причины и природа политических конфликтов. Типы политических конфликтов: конфликты интересов, конфликты ценностей, конфликты идентификаций. Методы разрешения конфликтов. Сущность этнополитических конфликтов, их виды. Способы разрешения и урегулирования.

**Политическая активность и политическое участие.** Типы политического участия. Ортодоксальное и неортодоксальное политическое участие. Конвенциональное и неконвенциональное политическое участие. Теории политического участия. Электоральное поведение. Политический протест. Политический терроризм. Природа абсентеизма.

### **Контрольные вопросы**

6. Дайте определение политическому развитию.
7. Назовите основные группы противоречий политической модернизации.
8. Что такое “политический конфликт”?
9. Назовите стадии политического конфликта.
10. Каковы функции и методы разрешения социально-политических конфликтов?
11. Назовите субъектов этнополитического конфликта.
12. Определите понятие “политического риска”.
13. Назовите виды политического участия.

### **Темы рефератов**

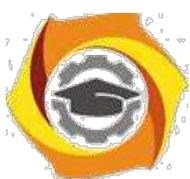
Модели политической модернизации.  
Перспективы политической модернизации в России.  
Типы политических конфликтов.  
Проблемы урегулирования этнополитических конфликтов.  
Особенности электорального поведения россиян.

Методические рекомендации для магистров по организации и выполнению самостоятельной работы по учебной практике " Практика по получению первичных навыков научно-исследовательской работы".

## **14. ОБЩИЕ ПОЛОЖЕНИЯ**

1.1. Отчет о НИР - научно-технический документ, который содержит систематизированные данные о научно-исследовательской работе, описывающий процесс или результаты научно-технического исследования, или состояние научно-технической проблемы.

1.2. При выполнении НИР, кроме заключительного отчета о работе в целом, могут быть составлены промежуточные отчеты по отдельным этапам НИР.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

1.3. Отчет о НИР подлежит обязательному нормоконтролю в организации-исполнителе. При проведении нормоконтроля рекомендуется руководствоваться ГОСТ 2.111.

1.4. Ответственность за достоверность данных, содержащихся в отчете, и за соответствие его требованиям настоящего стандарта несет организация-исполнитель.

## 8. СТРУКТУРА ОТЧЕТА

Структурными элементами отчета о НИР являются:

20. **титульный лист;**
21. список исполнителей;
22. **реферат;**
23. **содержание;**
24. перечень сокращений, условных обозначений, символов, единиц и тер-минов;
25. **введение;**
26. **основная часть;**
27. **заключение;**
28. список использованных источников;
29. приложения.

Обязательные структурные элементы выделены полужирным шрифтом. Остальные структурные элементы включают в отчет по смотрению исполнителя НИР с учетом требований разд.3 и 4.

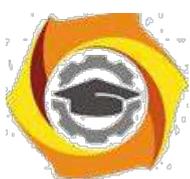
## 9. ТРЕБОВАНИЯ К СТРУКТУРНЫМ ЭЛЕМЕНТАМ ОТЧЕТА

### 3.1. Титульный лист

3.1.1. Титульный лист является первой страницей отчета о НИР, служит источником информации, необходимой для обработки и поиска документа.

3.1.2. На титульном листе приводят следующие сведения:

11. наименование организации - исполнителя НИР;
12. индекс УДК;
13. коды Высших классификационных группировок Общесоюзного классификатора промышленной и сельскохозяйственной продукции для НИР, предшествующих постановке продукции на производство;
14. номера, идентифицирующие отчет;
15. грифы согласования и утверждения;
16. наименование работы;



наименование отчета;

16. вид отчета (заключительный, промежуточный);
17. номер (шифр) темы;
18. должности, ученые степени, ученые звания, фамилии и инициалы руководи-телей организации - исполнителя НИР, руководителей НИР;
19. место и дата составления отчета.

3.1.3. Если отчет о НИР состоит из двух и более книг, каждая книга должна иметь свой титульный лист, соответствующий титульному листу первой книги и содержащий сведения, относящиеся к данной книге.

### 3.2. Список исполнителей

3.2.1. В список исполнителей должны быть включены фамилии и инициалы, должности, ученые степени, ученые звания руководителей НИР, ответственных исполнителей, исполнителей и соисполнителей, принимавших творческое участие в выполнении работы. В конце списка помещают подпись нормоконтролера.

3.2.2. Если отчет выполнен одним исполнителем, его должность, ученую степень, ученое звание, фамилию и инициалы, а также подпись нормоконтролера следует указывать на титульном листе отчета.

### 3.3. Реферат

3.3.1. Общие требования к реферату на отчет о НИР - по ГОСТ 7.9.

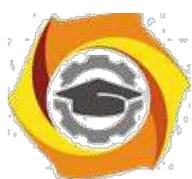
3.3.2. Реферат должен содержать:

25. сведения об объеме отчета, количестве иллюстраций, таблиц, приложений, количестве книг отчета, количестве использованных источников;
26. перечень ключевых слов;
27. текст реферата.

3.3.2.1. Перечень ключевых слов должен включать от 5 до 15 слов или словосочетаний из текста отчета, которые в наибольшей мере характеризуют его содержание и обеспечивают возможность информационного поиска. Ключевые слова приводятся в именительном падеже и печатаются прописными буквами в строку через запятые.

3.3.2.2. Текст реферата должен отражать;

22. объект исследования или разработки;
23. цель работы;
24. метод исследования и аппаратуру;
25. полученные результаты и их новизну;
26. основные конструктивные, технологические и технико-эксплуатационные характеристики;
27. степень внедрения;
28. рекомендации по внедрению или итоги внедрения результатов НИР;
29. область применения;
30. экономическую эффективность или значимость работы;
31. прогнозные предположения о развитии объекта исследования.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Если отчет не содержит сведений по какой-либо из перечисленных структурных частей реферата, то в тексте реферата она опускается, при этом последовательность изложения сохраняется.

#### 3.4. Содержание

3.4.1 Содержание включает введение, наименование всех разделов, подразделов, пунктов (если они имеют наименование) и заключение с указанием номеров страниц, с которых начинаются эти элементы отчета о НИР.

3.4.2. При составлении отчета, состоящего из двух и более книг, в каждой из них должно быть свое содержание. При этом в первой книге помещают содержание всего отчета с указанием номеров книг, в последующих - только содержание соответствующей книги. Допускается в первой книге вместо содержания последующих книг указывать только их наименование.

3.4.3. В отчете о НИР объемом не более 10 страниц содержание допускается не составлять.

3.5. Перечень сокращений, условных обозначений, символов, единиц и терминов

3.5.1. Принятые в отчете малораспространенные сокращения, условные обозначения, символы, единицы и специфические термины должны быть представлены в виде отдельного списка.

3.5.2. Если сокращения, условные обозначения, символы, единицы и термины повторяются в отчете менее трех раз, отдельный список не составляют, а расшифровку дают непосредственно в тексте отчета при первом упоминании.

#### 3.6. Введение

3.6.1. Введение должно содержать оценку современного состояния решаемой научно-технической проблемы, основание и исходные данные для разработки темы, обоснование необходимости проведения НИР, сведения о планируемом научно-техническом уровне разработки, о патентных исследованиях и выводы из них, сведения о метрологическом обеспечении НИР. Во введении должны быть показаны актуальность и новизна темы, связь данной работы с другими научно-исследовательскими работами.

3.6.2. Во введении промежуточного отчета по этапу НИР должны быть приведены цели и задачи исследований на этапе, их место в выполнении НИР в целом и обоснование выделения этапа.

3.6.3. Во введении заключительного отчета о НИР помещают перечень наименований всех подготовленных промежуточных отчетов по этапам и их инвентарные номера.

#### 3.7. Основная часть

3.7.1. Основная часть отчета должна содержать данные, отражающие существо, методику и основные результаты выполненной НИР.

3.7.2. Основная часть должна содержать:

13. выбор направления исследований, включающий обоснование выбора принятого направления исследования, методы решения задач и их сравнительную оценку, разработку общей методики проведения НИР;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- \* теоретические и (или) экспериментальные исследования, включающие определение характера и содержания теоретических исследований, методы исследований, методы расчета, обоснование необходимости проведения экспериментальных работ, принципы действия разработанных объектов, их характеристики, обоснование выбранного метрологического обеспечения работ, данные об объектах измерений, измеряемых величинах и средствах измерений, их метрологические характеристики, оценку правильности и экономичности выбора средств измерений (в том числе и нестандартизируемых) и методик выполнения измерений, сведения об их аттестации, оценку погрешности измерений, полученные экспериментальные данные;
- \* обобщение и оценку результатов исследований, включающие оценку полноты решения поставленной задачи и предложения по дальнейшим направлениям работ, оценку достоверности полученных результатов и их сравнение с аналогичными результатами отечественных и зарубежных работ, обоснование необходимости проведения дополнительных исследований, отрицательные результаты, приводящие к необходимости прекращения дальнейших исследований.

3.7.3. В зависимости от особенностей выполненной НИР основную часть излагают в виде текста, таблицы, сочетания иллюстраций и таблиц или сочетания текста, иллюстраций и таблиц.

Основную часть отчета следует делить на разделы и пункты. Разделы основной части могут делиться на пункты или на подразделы и пункты. Пункты, при необходимости, могут делиться на подпункты. Каждый пункт должен содержать законченную информацию.

3.7.4. Представление в отчете данных о свойствах веществ и материалов - по ГОСТ 7.54.

3.7.5. Единицы физических величин в отчете - по ГОСТ 8.417.

3.8. Заключение

Заключение должно содержать:

в краткие выводы по результатам выполненной НИР или отдельных ее этапов, оценку полноты решений поставленных задач, разработку рекомендаций и исходных данных по конкретному использованию результатов НИР, оценку технико-экономической эффективности внедрения. Если определение технико-экономической эффективности невозможно, необходимо указать народнохозяйственную, научную, социальную значимость работы;

воценку научно-технического уровня выполненной НИР в сравнении с лучшими достижениями в данной области.

3.9. Список использованных источников

Список должен содержать сведения об источниках, использованных при составлении отчета. Сведения об источниках приводятся в соответствии с требованиями ГОСТ 7.1.

3.10. Приложения

3.10.1. В приложения рекомендуется включать материалы, связанные с выполненной НИР, которые по каким-либо причинам не могут быть включены в основную часть.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

и приложения могут быть включены:

- 1) материалы, дополняющие отчет;
- 2) промежуточные математические доказательства, формулы и расчеты;
- 3) таблицы вспомогательных цифровых данных;
- 4) протоколы испытаний;
- 5) описание аппаратуры и приборов, применяемых при проведении экспериментов, измерений и испытаний;
- 6) заключение метрологической экспертизы;
- 7) инструкции, методики, описания алгоритмов и программ задач, решаемых ЭВМ, разработанных в процессе выполнения НИР; 8) иллюстрации вспомогательного характера;
- 9) копию технического задания на НИР, программы работ, договора или другого исходного документа для выполнения НИР;
- 10) протокол рассмотрения выполненной НИР на научно-техническом совете;
- 11) акты внедрения результатов НИР и др.

3.10.2. В приложения следует включать иллюстрации, таблицы, распечатки с ЭВМ, выполненные на листах формата А3.

3.10.3. В приложения к отчету о НИР, предшествующей постановке продукции на производство, должен быть включен проект технического задания на разработку (модернизацию) продукции или документ (заявку, протокол, контракт и др.), содержащий обоснованные технико-экономические требования к продукции.

3.10.4. В приложения к отчету о НИР, в составе которой предусмотрено проведение патентных исследований, должен быть включен отчет о патентных исследованиях, оформленный по ГОСТ 15.011, библиографический список публикаций и патентных документов, полученных в результате выполнения НИР, - по ГОСТ 7.1.

#### \* ПРАВИЛА ОФОРМЛЕНИЯ ОТЧЕТА

##### 4.1. Общие требования

4.1.1. Страницы текста отчета о НИР и включенные в отчет иллюстрации, таблицы и распечатки с ЭВМ должны соответствовать формату А4 по ГОСТ 9327. Допускается представлять иллюстрации, таблицы и распечатки с ЭВМ на листах формата А3.

4.1.2. Отчет о НИР должен быть выполнен машинописным способом или с применением печатающих и графических устройств вывода ЭВМ на одной стороне листа белой бумаги через полтора интервала. Для отчетов, выполненных на печатающих и графических устройствах вывода ЭВМ, высота букв и цифр должна быть не менее 1.8 мм.

Текст отчета следует печатать, соблюдая следующие размеры полей: левое - не менее 30 мм, правое - не менее 10 мм, верхнее - не менее 15 мм, нижнее - не менее 20 мм.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

4.1.3. Вне зависимости от способа выполнения отчета качество напечатанного текста и оформление иллюстраций, таблиц, распечаток с ЭВМ должно удовлетворять требованию их четкого воспроизведения (электрографическое копирование, микрофильмирование).

4.1.4. При выполнении отчета необходимо соблюдать равномерную плотность, контрастность и четкость изображения по всему отчету. В отчете должны быть четкие, не расплывшиеся линии, буквы, цифры и знаки. Все линии, буквы, цифры и знаки должны быть одинаково черными по всему отчету.

4.1.5. Вписывать в отпечатанный текст отчета отдельные слова, формулы, знаки допускается только черными чернилами или черной тушью, при этом плотность вписанного текста должна быть максимально приближена к плотности основного изображения.

4.1.6. Опечатки, описки и графические неточности допускается исправлять подчисткой или закрашиванием белой краской и нанесением на том же месте исправленного изображения машинописным способом или от руки черными чернилами или черной тушью.

4.1.7. Фамилии, названия учреждений, организаций, фирм, название изделий и другие имена собственные в отчете приводят на языке оригинала. Допускается транслитерировать имена собственные и приводить названия организаций В переводе на языке отчета с добавлением (при первом упоминании) оригинального названия.

4.1.8. Сокращение русских слов и словосочетаний в отчете - по ГОСТ 7.12.

4.1.9. Наименования структурных элементов отчета «СПИСОК ИСПОЛНИТЕЛЕЙ», «РЕФЕРАТ», «СОДЕРЖАНИЕ», «ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ И ТЕРМИНОВ», «ВВЕДЕНИЕ», «ЗАКЛЮЧЕНИЕ», «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ» служат заголовками структурных элементов отчета.

4.1.10. Заголовки структурных элементов отчета и разделов основной части следует располагать в середине строки без точки в конце и печатать прописными буквами, не подчеркивая.

4.1.11. Заголовки подразделов и пунктов следует начинать с абзацного отступа и печатать с прописной буквы вразрядку, не подчеркивая, без точки в конце.

4.1.12. Если заголовок включает несколько предложений, их разделяют точками. Переносы слов в заголовках не допускаются.

4.1.13. Расстояние между заголовками структурных элементов отчета и разделов основной части и текстом должно быть не менее 3 - 4 интервалов.

4.1.14. Пункты и подпункты основной части следует начинать печатать с абзацного отступа.

#### 4.2. Нумерация страниц отчета

4.2.1. Страницы отчета следует нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всему тексту отчета. Номер страницы проставляют в правом верхнем углу без точки в конце.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

4.2.2. Титульный лист включают в общую нумерацию страниц отчета. Номер страницы на титульном листе не проставляют.

4.2.3. Иллюстрации и таблицы, расположенные на отдельных листах, и распечатки с ЭВМ включают в общую нумерацию страниц отчета. Иллюстрации, таблицы и распечатки с ЭВМ на листе формата А3 учитывают как одну страницу.

4.3. Нумерация разделов, подразделов, пунктов, подпунктов и книг отчета

4.3.1. Разделы, подразделы, пункты, подпункты и книги отчета следует нумеровать арабскими цифрами.

4.3.2. Разделы отчета должны иметь порядковую нумерацию в пределах основной части отчета и обозначаться арабскими цифрами с точкой, например, 1., 2., 3. и т.д.

4.3.3. Пункты должны иметь порядковую нумерацию в пределах каждого раздела или подраздела. Номер пункта включает номер раздела и порядковый номер подраздела или пункта, разделенные точкой, например 1.1., 1.2., 1.3. или 1.1.1., 1.1.2., 1.1.3. и т.д.

4.3.4. Номер подпункта включает номер раздела, подраздела, пункта и порядковый номер подпункта, разделенные точкой, например 1.1.1.1., 1.1.1.2. и т.д.

4.3.5. Если раздел или подраздел имеет только один пункт или пункт имеет один подпункт, то нумеровать пункт (подпункт) не следует.

4.3.6. Книги отчета должны иметь порядковую нумерацию в пределах всего отчета. Номер каждой книги следует проставлять арабскими цифрами на титульном листе под указанием вида отчета, например, «Книга 3».

4.4. Иллюстрации

4.4.1. Иллюстрации (чертежи, графики, схемы, диаграммы, фотоснимки) следует располагать в отчете непосредственно после текста, в котором они упоминаются впервые, или на следующей странице.

На все иллюстрации должны быть даны ссылки в отчете.

4.4.2. Чертежи, графики, диаграммы, схемы, помещаемые в отчете, должны соответствовать требованиям государственных стандартов ЕСКД.

4.4.3. Фотоснимки размером меньше формата А4 должны быть наклеены на стандартные листы белой бумаги.

4.4.4. Иллюстрации должны иметь название, которое помещают над иллюстрацией. При необходимости под иллюстрацией помещают поясняющие данные (подрисовочный текст).

Иллюстрация обозначается словом «Рис.», которое помещают после поясняющих данных.

4.4.5. Иллюстрации следует нумеровать арабскими цифрами порядковой нумерацией в пределах всего отчета.

4.4.6. Если в отчете только одна иллюстрация, ее нумеровать не следует и слово «Рис.» под ней не пишут.

4.4.7. Иллюстрацию следует выполнять на одной странице. Если иллюстрация не уместится на одной странице, можно переносить ее на другие страницы, при этом название иллюстрации помещают на первой странице, поясняющие данные - к каждой странице и под ними указывают «Рис. , лист ».



#### 4.5. Таблицы

4.5.1. Цифровой материал должен оформляться в виде таблиц.

4.5.2. Таблицу следует располагать в отчете непосредственно после текста, в котором она упоминается впервые, или на следующей странице. На все таблицы должны быть ссылки в отчете.

4.5.3. Таблицы следует нумеровать арабскими цифрами порядковой нумерацией в пределах всего отчета. Номер следует размещать в правом верхнем углу над заголовком таблицы после слова «Таблица».

4.5.4. Если в отчете одна таблица, ее не нумеруют и слово «Таблица» не пишут.

4.5.5. Оформление таблиц в отчете - по ГОСТ 2.105.

#### 4.6. Перечисления и примечания

4.6.1. Перечисления, при необходимости, могут быть приведены внутри пунктов или подпунктов. Перечисления следует нумеровать порядковой нумерацией арабскими цифрами со скобкой, например 1), 2), 3) и т.д., и печатать строчными буквами с абзацного отступа.

В пределах одного пункта или подпункта не допускается более одной группы перечислений.

4.6.2. Примечания следует помещать в отчете при необходимости пояснения содержания текста, таблицы или иллюстрации. Примечания размещают непосредственно после пункта, подпункта, таблицы, иллюстрации, к которым они относятся, и печатают с прописной буквы с абзацного отступа.

Слово «Примечание» следует печатать с прописной буквы с абзацного отступа вразрядку и не подчеркивать.

4.6.3. Одно примечание не нумеруют. Несколько примечаний следует нумеровать порядковой нумерацией арабскими цифрами точкой, например:  
Примечание.

.....

.....

Примечания:

1.....

2.....

#### 4.7. Формулы и уравнения

4.7.1. Пояснение значений символов и числовых коэффициентов следует приводить непосредственно под формулой в той же последовательности, в которой они даны в формуле. Значение каждого символа и числового коэффициента следует давать с новой строки. Первую строку пояснения начинают со слова «где» без двоеточия.

4.7.2. Уравнения и формулы следует выделять из текста в отдельную строку. Выше и ниже каждой формулы или уравнения должно быть оставлено не менее одной свободной строки. Если уравнение не умещается в одну строку, оно должно быть перенесено после знака равенства (=) или после знаков плюс (+), минус (-), умножения (x), деления (:), или других математических знаков.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

4.7.3. Формулы в отчете следует нумеровать порядковой нумерацией в пределах всего отчета арабскими цифрами в круглых скобках в крайнем правом положении на строке.

Если в отчете только одна формула или уравнение, их не нумеруют. 4.8. Ссылки

4.8.1. Ссылки на источники следует указывать порядковым номером по списку источников, выделенным двумя косыми чертами.

Наряду с общим списком допускается приводить ссылки на источники в подстрочном примечании. Оформление ссылок - по ГОСТ 7.1.

4.8.2. Ссылки на разделы, подразделы, пункты, подпункты, иллюстрации, таблицы, формулы, уравнения, перечисления, приложения следует указывать их порядковым номером, например: «... в разд. 4», «... по п.3.3.4», «... в подпункте 2.3.4.1, перечисление 3», «... по формуле (3)», «... в уравнении (2)», «... на рис.8», «... в приложении б».

Если в отчете одна иллюстрация, одна таблица, одна формула, одно уравнение, одно приложение, следует при ссылках писать «на рисунке», «в таблице», «по формуле», «в уравнении», «в приложении».

#### 4.9. Титульный лист

4.9.1. Титульный лист содержит реквизиты:

– наименование министерства (ведомства) или другого структурного образования, в систему которого входит организация-исполнитель, наименование организации (в том числе и сокращенное);

– индекс УДК, код ВКГ ОКП (для отчетов о НИР, предшествующих разработке и модернизации продукции) и номер государственной регистрации, предоставляемые организацией-исполнителем, а также надпись «Инв. №» - эти данные размещаются одно под другим;

– специальные отметки (при наличии в отчете численных данных о свойствах веществ и материалов в этой части проставляется аббревиатура ГСССД - государственная служба стандартных справочных данных);

– гриф согласования, гриф утверждения.

Гриф согласования состоит из слова «СОГЛАСОВАНО», должности с указанием наименования организации, ученой степени, ученого звания лица, согласовавшего отчет, его личной подписи, ее расшифровки, даты согласования, печати согласующей организации.

Если согласование проводилось письмом, следует указывать сокращенное наименование согласующей организации, исходящий номер и дату письма.

Гриф утверждения состоит из слова «УТВЕРЖДАЮ», должности с указанием наименования организации, ученой степени, ученого звания лица, утвердившего отчет, личной подписи, ее расшифровки и даты утверждения отчета. Здесь же проставляется печать организации, утвердившей отчет.

Подписи и даты подписания должны быть выполнены только черными чернилами или тушью.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Элементы даты приводят арабскими цифрами в одной строке в следующей последовательности: день месяца, месяц, год, например: дату 1 октября 2016 г.. следует оформлять 01.10.16;

и вид документа, строчными буквами с первой прописной - наименование зарегистрированной НИР, прописными буквами - наименование отчета, строчными буквами в круглых скобках - вид отчета (промежуточный или заключительный), строчными буквами с первой прописной номер книги отчета (если наименование зарегистрированной НИР совпадает с наименованием отчета, его печатают прописными буквами);

и шифр государственной научно-технической программы, шифр работы, присвоенный организацией-исполнителем;

и должности, ученые степени, ученые звания руководителей организации-исполнителя НИР, руководителей НИР, затем оставляют свободное поле для личных подписей и помещают инициалы и фамилии лиц, подписавших отчет, ниже личных подписей проставляют даты подписания (если на титульном листе не размещаются все необходимые подписи, то допускается переносить их на следующую страницу;

и город и год выпуска отчета.

#### 4.10. Список исполнителей

4.10.1. Фамилии и инициалы, должности, ученые степени, ученые звания в списке следует располагать столбцом. Слева указывают должности, ученые степени, ученые звания исполнителей и соисполнителей, затем оставляют свободное поле для подлинных подписей, справа указывают инициалы и фамилии исполнителей и соисполнителей. Возле каждой фамилии в скобках следует указывать номер раздела (подраздела) и фактическую часть работы, подготовленную конкретным исполнителем. Для соисполнителей следует также указывать наименование организации-соисполнителя.

4.11. Перечень сокращений, условных обозначений, символов, единиц и терминов

Перечень должен располагаться столбцом. Слева в алфавитном порядке приводят сокращения, условные обозначения, символы, единицы и термины, справа - их детальную расшифровку.

#### 4.12. Список использованных источников

Сведения об источниках следует располагать в порядке появления ссылок на источники в тексте отчета и нумеровать арабскими цифрами с

точкой. 4.13. Приложения

4.13.1. Приложения следует оформлять как продолжение отчета на его последующих страницах или в виде отдельной книги, располагая приложения в порядке появления на них ссылок в тексте отчета.

4.13.2. Каждое приложение должно начинаться с новой страницы и иметь содержательный заголовок, напечатанный прописными буквами. В правом верхнем углу над заголовком прописными буквами должно быть напечатано слово «ПРИЛОЖЕНИЕ».



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Если приложений в отчете более одного, их следует нумеровать арабскими цифрами порядковой нумерацией.

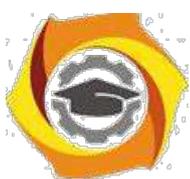
4.13.3. При оформлении приложений отдельной книгой отчета на титуль-ном листе под номером книги следует писать прописными буквами слово «ПРИЛОЖЕНИЯ».

4.13.4. При необходимости текст приложений может быть разделен на разделы, подразделы, пункты, которые следует нумеровать в пределах каждого приложения в соответствии с требованиями подраздела 4.3.

4.13.5. Имеющиеся в тексте приложения иллюстрации, таблицы, формулы 19) уравнения следует нумеровать в пределах каждого приложения в соответствии с требованиями подразделов 4.4, 4.5 и 4.7.

4.13.6. Перечисления и примечания в тексте приложения оформляют и нумеруют в соответствии с требованиями подраздела 4.6.

4.13.7. Если в качестве приложения в отчете используется документ, имеющий самостоятельное значение и оформляемый согласно требованиям к документу данного вида, его вкладывают в отчет без изменений в оригинале. На титульном листе документа в правом верхнем углу печатают слово «ПРИЛОЖЕНИЕ» и проставляют его номер, а страницы, на которых размещен документ, включают в общую нумерацию страниц отчета.



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

ПРИЛОЖЕНИЕ  
*Пакет документов по практике*



Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет УГМК»  
Кафедра информационных технологий

ЗАДАНИЕ  
на практику

Студент: \_\_\_\_\_

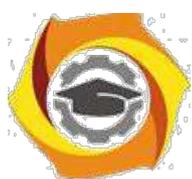
Группа: \_\_\_\_\_

Вид практики: Практика по получению первичных навыков научно-исследовательской работы

Срок практики: \_\_\_\_\_

Место практики: \_\_\_\_\_

г. Верхняя Пышма  
2019 г.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**с СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИКИ**

Практика проводится на предприятии в соответствии с заданием на практику. По результатам практики студент предоставляет отчет по практике.

**Содержание практики**

| <b>№ п.п.</b> | <b>Разделы (этапы) практики</b>                                                                                                      | <b>Всего дней</b> |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <b>1</b>      | <b>Выездная практика на предприятии</b>                                                                                              | <b>12</b>         |
| <b>1.1</b>    | Инструктаж по соблюдению правил внутреннего трудового распорядка, требований охраны труда, пожарной безопасности и ТБ                | 1                 |
| <b>1.2</b>    | Анализ научно-исследовательских работ, проведенных на предприятии, направленных на управление и устойчивое развитие электрохозяйства | 4                 |
| <b>1.3</b>    | Осуществить поиск и отбора патентной документации в соответствии с производственной задачей.                                         | 6                 |
| <b>1.4</b>    | Подготовка отчета по практике                                                                                                        | 1                 |

Задание на практику получил(а) \_\_\_\_\_

Подпись

Дата

ФИО

**«Согласовано»**

**Заведующий кафедрой ИТ**

Подпись

**Горбатов С.В.**

ФИО



**ИНСТРУКЦИЯ ПО ОФОРМЛЕНИЮ ДОКУМЕНТОВ ПО ПРАКТИКЕ**

отчете по практике (приложение 2) должны содержаться следующие документы, подписанные руководителем практики и заверенные печатью организации:

- 10) формуляр отчета по практике (приложение 3);
- 11) отзыв руководителя практики от предприятия (приложение 4);
- 12) справка о проведении инструктажей обучающихся по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, правилами внутреннего трудового распорядка (приложение 1);

Приложение 1

**СПРАВКА**

**12. проведении инструктажа обучающихся по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, правилами внутреннего трудового распорядка**

| <b>Дата и время прохождения инструктажа</b> | <b>Наименование инструктажа (краткое содержание, характер требований)</b> | <b>Подпись, /расшифровка подписи, должность инструктора//</b> | <b>Подпись, /расшифровка подписи инструктируемого/</b> |
|---------------------------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------|--------------------------------------------------------|
| 03.07.2017 г.<br>с 9-00 до 9-30             | Требования охраны труда:<br>1.<br>2.                                      | Иванов И.А., мастер<br>АО «__»                                | Сидоров А.А.                                           |
| 03.07.2017 г.<br>с 9-30 до 10-00            | Требования техники безопасности:<br>1.<br>2.                              | Иванов И.А., мастер<br>АО «__»                                | Сидоров А.А.                                           |
| 03.07.2017 г.<br>с 10-00 до 10-30           | Требования пожарной безопасности:<br>1.<br>2.                             | Иванов И.А., мастер<br>АО «__»                                | Сидоров А.А.                                           |
| 03.07.2017 г.<br>с 10-30 до 11-00           | Правилами внутреннего трудового распорядка:<br>1.<br>2..                  | Петров А.С. специалист кадрового управления АО<br>«__»        | Сидоров А.А.                                           |

Руководитель практики от предприятия: \_\_\_\_\_ И.О.Фамилия  
Руководитель по персоналу  
организации УГМК: \_\_\_\_\_ И.О.Фамилия

М.П.



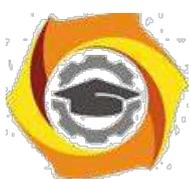
### Структура отчета по практике

34. *Титульный лист*
35. *Содержание* (включает названия разделов с указанием страниц, с которых они начинаются)
36. *Введение* (включает цели, задачи, сроки практики, краткую характеристику предприятия, технологических процессов, оборудования в соответствии с направлением подготовки)
37. *Основная часть* (включает описание выполненной на практике работы, с подробным раскрытием видов работ, оборудования, приборов, инструментов, технологий и т.д.)
38. *Заключение* (включает выводы, рациональные предложения, инновации сделанные во время прохождения практики)
39. *Список использованной литературы и нормативных документов*
40. *Приложения* (содержат вспомогательный материал, не включенный в основную часть отчета: таблицы, схемы, заполненные формы отчетности и т.д.)

*Все разделы Отчета по практике должны быть логически связаны между собой.*

*Оформление отчета должно быть в соответствии с:*

ГОСТ 7.32-2017 - «Система стандартов по информации, библиотечному и издательскому делу. Отчёт о научно-исследовательской работе. Структура и правила оформления».



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

ТИТУЛЬНЫЙ ЛИСТ



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет УГМК»  
Кафедра информационных технологий

**Отчет  
по практике**

---

Наименование практики

**Выполнил:**

|                |        |       |         |
|----------------|--------|-------|---------|
| _____          | Ф.И.О. | _____ | Подпись |
| <b>Студент</b> | _____  | _____ | _____   |
| <b>Группа</b>  | _____  | _____ | _____   |

**Руководитель практики:**

|       |        |       |         |
|-------|--------|-------|---------|
| _____ | Ф.И.О. | _____ | Подпись |
|-------|--------|-------|---------|

**Руководитель практики от кафедры:**

|               |        |       |         |
|---------------|--------|-------|---------|
| Горбатов С.В. | _____  | _____ | Подпись |
|               | Ф.И.О. |       |         |

**Оценка:** \_\_\_\_\_



**ФОРМУЛЯР ОТЧЕТА ПО ПРАКТИКЕ**

\_\_\_\_\_  
*Название практики*

1. Студент \_\_\_\_\_

*ФИО*

проходил(а) практику с «\_\_» \_\_\_\_\_ 20\_\_ по «\_\_» \_\_\_\_\_ 2017  
в \_\_\_\_\_

\_\_\_\_\_  
*Название организации, адрес*

ФИО руководителя организации (объекта практики) \_\_\_\_\_

2. За время \_\_\_\_\_

\_\_\_\_\_ были выполнены следующие вид работ: \_\_\_\_\_ (в соответствии с заданием на  
практику)

3. Заключение руководителя практики от организации:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Руководитель практики от предприятия: \_\_\_\_\_

*ФИО*

*Подпись*

Заключение руководителя практики от ТУ УГМК:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Общая оценка по практике: \_\_\_\_\_

Руководитель практики от ТУ УГМК: \_\_\_\_\_

Горбатов С.В.

*ФИО*

*Подпись*



## ОТЗЫВ РУКОВОДИТЕЛЯ ПРАКТИКИ ОТ ПРЕДПРИЯТИЯ

\_\_\_\_\_  
*Название практики*

9. Студент \_\_\_\_\_

*ФИО*

*В данном отзыве необходимо отметить следующие моменты: качества, умения и навыки, которые проявил студент в процессе прохождения практики; замечания; отношение к выполнению заданий, умение применять знания в производствен-ных*

*условиях, соблюдение трудовой дисциплины и др.*

ОБЩЕЕ ЗАКЛЮЧЕНИЕ:

ЗАМЕЧАНИЯ

Руководитель практики от предприятия:

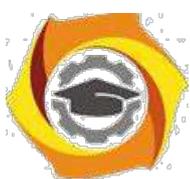
Фамилия, имя, отчество:

Должность:

Подпись

Дата: \_\_\_\_\_20\_\_

*(печать организации)*



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*Тематика самостоятельной работы*

| № | Наименование работы                                                                                                                                                                                             |
|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Подготовительный этап:<br>- общие методические указания по подготовке материала для ВКР;<br>- инструктаж по соблюдению правил внутреннего трудового распорядка, требований охраны труда и пожарной безопасности |
| 2 | Основной этап:<br>- сбор, обработка материала для разделов ВКР;<br>- работа над схемными и расчетными решениями, направленными на работу ИТ сервисов предприятия;                                               |
| 3 | Заключительный этап.<br>Составление отчета по практике.                                                                                                                                                         |

***Самостоятельная работа № 1, 2***

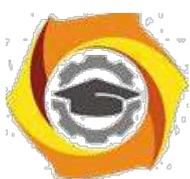
Подготовительный этап:  
в общие методические указания по подготовке материала для ВКР;  
в инструктаж по соблюдению правил внутреннего трудового распорядка, требований охраны труда и пожарной безопасности

Основной этап:  
в сбор, обработка материала для разделов ВКР;  
в работа над схемными и расчетными решениями, направленными на устойчивое энергохозяйство предприятия;

При выполнении задания по Практике магистрант должен использовать современную учебную и научную литературу, обратиться к аналитическим исследованиям, обзорам и проблемным работам по теме магистерской диссертации; использовать нормативную документацию, инструкции, в том числе на английском языке, программы развития, реализующиеся на предприятии.

**Примерные темы для индивидуальных заданий практики:**  
в подготовка и обоснование проектных решений (схемных и расчетных) для ВКР;  
в подготовка технико-экономического обоснования предложений для магистерского проекта;  
- анализ и представление результатов исследования для магистерского проекта;

в формулировки уточнений технического задания на проект;  
в разработка и использование средства автоматизации при проектировании и технологической подготовке производства;  
в обоснование методов анализа вариантов, разработки и поиска компромиссных решений для магистерского проекта;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- создание и анализ моделей, позволяющих прогнозировать электропотребление на предприятии;
- приобретение умения выбирать электрооборудование для систем электрохозяйства предприятий;
- приобретение навыков управления электрохозяйством предприятия, подразделения;
- оценка рисков и определение мер по обеспечению безопасности технологических процессов на предприятии, в подразделении;
- анализ материальных балансов предприятия во взаимосвязи с энергетическими балансами;
- анализ потенциала и потребности предприятия в объекте собственной генерации;
- приобретение опыта в применении аналитических методов для получения лучших инженерных решений с целью достижения целевых показателей энергоэффективности предприятия, подразделения.

***Самостоятельная работа № 3***

Составление отчета по практике.

**ИНСТРУКЦИЯ ПО ОФОРМЛЕНИЮ ДОКУМЕНТОВ ПО  
ПРАКТИКЕ**

отчете по практике (приложение 2) должны содержаться следующие документы, подписанные руководителем практики и заверенные печатью организации:

- в формуляр отчета по практике (приложение 3);
- в отзыв руководителя практики от предприятия (приложение 4);
- всправка о проведении инструктажей обучающихся по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, правилами внутреннего трудового распорядка (приложение 1);



**СПРАВКА**

**- проведении инструктажа обучающихся по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, правилами внутреннего трудового распорядка**

| Дата и время прохождения инструктажа | Наименование инструктажа (краткое содержание, характер требований) | Подпись, /расшифровка подписи, должность инструктора// | Подпись, /расшифровка подписи инструктируемого/ |
|--------------------------------------|--------------------------------------------------------------------|--------------------------------------------------------|-------------------------------------------------|
| 03.07.2017 г.<br>с 9-00 до 9-30      | Требования охраны труда:<br>1.<br>2.                               | Иванов И.А., мастер<br>АО «__»                         | Сидоров А.А.                                    |
| 03.07.2017 г.<br>с 9-30 до 10-00     | Требования техники безопасности:<br>1.<br>2.                       | Иванов И.А., мастер<br>АО «__»                         | Сидоров А.А.                                    |
| 03.07.2017 г.<br>с 10-00 до 10-30    | Требования пожарной безопасности:<br>1.<br>2.                      | Иванов И.А., мастер<br>АО «__»                         | Сидоров А.А.                                    |
| 03.07.2017 г.<br>с 10-30 до 11-00    | Правилами внутреннего трудового распорядка:<br>1.<br>2..           | Петров А.С. специалист кадрового управления АО<br>«__» | Сидоров А.А.                                    |

Руководитель практики от предприятия: \_\_\_\_\_ И.О.Фамилия

Руководитель по персоналу  
организации УГМК:

\_\_\_\_\_ И.О.Фамилия

М.П.



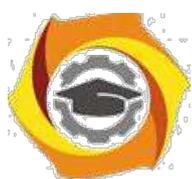
### Структура отчета по практике

- в *Титульный лист*
- в *Содержание* (включает названия разделов с указанием страниц, с которых они начинаются)
- в *Введение* (включает цели, задачи, сроки практики, краткую характеристику предприятия, технологических процессов, оборудования в соответствии с направлением подготовки)
- в *Основная часть* (включает описание выполненной на практике работы, с подробным раскрытием видов работ, оборудования, приборов, инструментов, технологий и т.д.)
- в *Заключение* (включает выводы, рациональные предложения, инновации сделанные во время прохождения практики)
- в *Список использованной литературы и нормативных документов*
- в *Приложения* (содержат вспомогательный материал, не включенный в основную часть отчета: таблицы, схемы, заполненные формы отчетности и т.д.)

*Все разделы Отчета по практике должны быть логически связаны между собой.*

*Оформление отчета должно быть в соответствии с:*

ГОСТ 7.32-2017 - «Система стандартов по информации, библиотечному и издательскому делу. Отчёт о научно-исследовательской работе. Структура и правила оформления».



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

ТИТУЛЬНЫЙ ЛИСТ



Негосударственное частное образовательное  
учреждение выс-  
шего образования  
«Технический университет УГМК»  
Кафедра информационных технологий

**Отчет  
по практике**

---

Наименование практики

**Выполнил:**

|         |        |       |         |
|---------|--------|-------|---------|
| _____   | Ф.И.О. | _____ | Подпись |
| Студент | _____  | _____ | _____   |
| Группа  | _____  | _____ | _____   |
| _____   | _____  | _____ | _____   |

**Руководитель практики:**

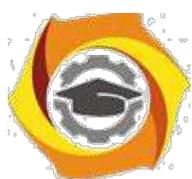
|       |        |       |         |
|-------|--------|-------|---------|
| _____ | Ф.И.О. | _____ | Подпись |
|-------|--------|-------|---------|

**Руководитель практики от кафедры:**

Горбатов С.В.

|       |        |       |         |
|-------|--------|-------|---------|
| _____ | Ф.И.О. | _____ | Подпись |
|-------|--------|-------|---------|

**Оценка:** \_\_\_\_\_



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



**ФОРМУЛЯР ОТЧЕТА ПО ПРАКТИКЕ**

\_\_\_\_\_  
*Название практики*

1. Студент \_\_\_\_\_

*ФИО*

проходил(а) практику с « \_\_\_\_ » \_\_\_\_\_ 20\_\_ по « \_\_\_\_ » \_\_\_\_\_ 2017  
в \_\_\_\_\_

\_\_\_\_\_  
*Название организации, адрес*

ФИО руководителя организации (объекта практики) \_\_\_\_\_

2. За время \_\_\_\_\_

\_\_\_\_\_ были выполнены следующие вид работ: \_\_\_\_\_ (в соответствии с заданием на  
практику)

3. Заключение руководителя практики от организации:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Руководитель практики от предприятия: \_\_\_\_\_

*ФИО*

*Подпись*

Заключение руководителя практики от ТУ УГМК:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Общая оценка по практике: \_\_\_\_\_

Руководитель практики от ТУ УГМК: \_\_\_\_\_

Горбатов С.В.

*ФИО*

*Подпись*



## ОТЗЫВ РУКОВОДИТЕЛЯ ПРАКТИКИ ОТ ПРЕДПРИЯТИЯ

\_\_\_\_\_  
*Название практики*

- Студент \_\_\_\_\_  
*ФИО*

- *данном отзыве необходимо отметить следующие моменты: качества, умения и навыки, которые проявил студент в процессе прохождения практики; замечания; отношение к выполнению заданий, умение применять знания в производствен-ных*

*условиях, соблюдение трудовой дисциплины и др.*

ОБЩЕЕ ЗАКЛЮЧЕНИЕ:

ЗАМЕЧАНИЯ

Руководитель практики от предприятия:

Фамилия, имя, отчество:

Должность:

Подпись

Дата: \_\_\_\_\_20\_\_

*(печать организации)*



## Прикладные программные системы

Технологии алгоритмизации для типовых прикладных программных систем

### 1. Цель занятия:

Выработать умения и навыки по составлению программ, содержащих методы пользователя различного типа.

### 2. Рекомендации:

Изучить материалы лекций №№1,2.

#### Краткая теория

#### Объявление методов на примере Java-программы

Методы - это подпрограммы, присоединенные к конкретным определениям классов. Они описываются внутри определения класса на том же уровне, что и переменные объектов. При объявлении метода задаются тип возвращаемого им результата и список параметров. Общая форма объявления метода такова:

*тип имя\_метода (список формальных параметров)*

```
{
тело метода:
}
```

Тип результата, который должен возвращать метод может быть любым, в том числе и типом void - в тех случаях, когда возвращать результат не требуется. Список формальных параметров - это последовательность пар типидентификатор, разделенных запятыми. Если у метода параметры отсутствуют, то после имени метода должны стоять пустые круглые скобки. *class Pt*

```
{ int x, y;
void init(int a, int b)
{ x = a;
y = b;
}
}
```

#### Вызов метода

В Java параметры примитивных типов передаются *по значению*, а это означает, что у метода нет доступа к исходной переменной, использованной в качестве параметра. Заметим, что все объекты передаются по ссылке, можно изменять содержимое того объекта, на который ссылается данная переменная.

#### Скрытие переменных представителей

В языке Java не допускается использование в одной или во вложенных областях видимости двух локальных переменных с одинаковыми именами, но при этом не запрещается объявлять формальные параметры методов, чьи имена совпадают с именами переменных представителей.

Давайте рассмотрим в качестве примера иную версию метода init, в которой формальным параметрам даны имена x и y, а для доступа к одноименным переменным текущего объекта используется ссылка *this*. *class Pt*

```
{ int x, y;
void init(int x, int y)
{ this.x = x;
this.y = y
}
}

class PtDvat
{
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
public static void main(String args[])
{
 Pt ob1 = new Pt(); Pt ob2 =
 new Pt(); ob1.init(10,20);
 ob2.init(42,99);
 System.out.println("x = " + ob1.x + " y = " + ob1.y);
 System.out.println("x = " + ob2.x + " y = " + ob2.y);
}
}
```

### **3. Порядок выполнения задания**

- 3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первая задача; №по журналу +3 – вторая задача; и №по журналу +5 – третья задача (если достигнуто окончание списка вариантов заданий, то перейти в его начало).
- 3.2. Составить программу по заданию.
- 3.3. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.
- 3.4. Оформить отчет для каждой из 3 задач, включив в него задание, блоксхему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.

### **4. Варианты заданий:**

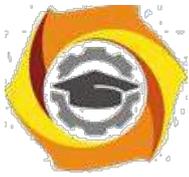
#### **Составить методы пользователя для решения следующих задач**

1. Упорядочить элементы массивов А и В в порядке возрастания. Массив передавать в качестве параметра-переменной процедуре Sort.
2. Используя подпрограмму с параметром типа integer вычисления факториала, составить программу для вычисления .
3. Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив процедуру, выполняющую это действие и передавая массив в качестве параметра.
4. Составить программу вычисления значения функции где  $X_i, Y_i, V_i$  - элементы массивов, состоящих из 15 элементов. Вычисления произведения осуществить в подпрограмме. В качестве параметров процедуре Mult передается массив и количество элементов, для которых нужно найти произведение.
5. Составить программу, позволяющую преобразовать массивы А и В следующим образом: из каждого элемента массива вычитается среднее значение всех элементов массива. Массивы передать процедуре в качестве параметров. 6. Используя метод (SQ(int x)), найти среди чисел a, b, c такие, которые можно представить в виде суммы квадратов двух натуральных чисел. 7. Составьте программу, выдающую письма стандартной формы, адресованные разным людям. От письма к письму должны меняться только фамилия и адрес, поэтому для печати писем примените метод, получающий эти данные из основной программы в качестве параметров.
8. Составьте программу, которая считывает числа М и N и находит их наибольший общий делитель (M,N - целые). Используйте метод NOD(int Z).
9. Используя метод, найти точку пересечения прямых  $A1*X + B1*Y = C1$  и  $A2*X + B2*Y = C2$ . Если они параллельные, выдавать об этом сообщение.
10. Составьте программу, которая считывает числа М и N и проверяет, простые они или нет ( M ,N - целые, передаются в качестве параметра). Используйте метод Prost(int Y).



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

11. Составить программу, позволяющую преобразовать массивы А и В следующим образом: из каждого элемента массива вычитается максимальный элемент соответствующего массива.
12. Используя метод, транспонировать матрицы А, В размера  $M*N$ . Матрица передается методу в качестве параметра.
13. Напишите метод-заставку к программе вычисления математических функций в виде  
\*\*\*\*\*  
\* Программа \*  
\* вычисления математических функций \*  
\*\*\*\*\* Фамилия автора должна передаваться методу в качестве параметра.
14. Используя метод, найти среднее арифметическое массивов X и Y с N элементами каждый. Массив передается методу в качестве параметра.
15. Используя метод  $\text{SumKv}(\text{int } d)$ , найти среди чисел a, b, c такие, которые можно представить в виде суммы квадратов двух натуральных чисел.
16. Составить программу, позволяющую преобразовать массивы А и В следующим образом: каждый элемент массива возводится в квадрат. Массив передается методу как параметр-переменная.
18. Используя метод  $\text{delit}(\text{int } X)$ , напечатать делители числа M, вводимого с клавиатуры. Реализовать для произвольного количества чисел. Признак конца ввода – число 0.
19. Составить программу, позволяющую преобразовать массивы А и В следующим образом: из каждого элемента массива вычитается минимальный элемент соответствующего массива. Массив передается методу в качестве параметра-переменной.
20. Дан массив целых чисел. Отсортировать числа в порядке возрастания. Описать метод сортировки поиска максимального числа. Массив передать в качестве параметра-переменной.
21. Используя метод  $\text{Skub}(\text{int } X)$ , найти среди чисел 1,2,...,N такие, которые можно представить в виде суммы кубов двух натуральных чисел.
22. Используя метод пользователя, найти  $C = AA + AB$ , где А и В - квадратные матрицы размера m. В методе реализовать умножение матриц, передав их в качестве параметров.
23. Используя метод, найти общие делители для пар целых чисел, вводимых с клавиатуры. Пара чисел передается методу в качестве параметров.  
Реализовать для произвольного количества таких пар чисел.
24. Используя метод, найти  $C = AB + AD$ , где А - квадратная матрица размера  $m \times m$ , В, D - векторы с m элементами. В методе реализовать умножение матрицы на вектор, передав их в качестве параметров.
25. Составить программу подсчета числа всех натуральных чисел, меньших M, квадрат суммы цифр которых равен X. Числа M и X передаются методу в качестве параметров. В основном методе задается 5 пар таких чисел.
26. Написать метод-заставку, печатающую титульный лист курсовой работы студента. В качестве параметров передаются фамилии и инициалы студента и преподавателя, название дисциплины.
27. Составить программу вычисления значения функции осуществить в методе (передать массив в качестве параметра).
28. Используя методы, составить программу для вычисления функции  $Z = (X1+Y1)/(X1*Y1)+X2*Y2$  где X1, X2 - корни уравнения  $X^2 - 4*X - 1 = 0$ ; Y1, Y2 - корни уравнения  $2*Y^2 + A*Y - A^2 = 0$ .
29. Используя метод  $\text{SumKub}(\text{int } d)$ , найти среди чисел a, в, с такие, которые можно представить в виде суммы кубов трех натуральных чисел.
30. Используя метод, составить программу для нахождения всех натуральных делителей чисел N и K. Натуральное число передается методу в качестве параметра.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**МУ к ПЗ 2**

Управляющие операторы C#

**1. Цель занятия:**

Выработать умения и навыки по составлению программ, содержащий классы, наследующие другие классы.

**2. Рекомендации:**

Изучить материалы лекций №№4-6.

**Краткая теория**

**Наследование. Виртуальные функции. Указатели на объекты**

Наследование - это механизм посредством которого один класс (производный или потомок) может наследовать свойства другого класса (базового или предка).

Базовый класс определяет все качества, которые являются общими для всех производных классов.

Пример:

```
//Базовый класс class B {
 int i; public:
 void set_i(int n); int
get_i();
};

//Производный класс D class D : public
B
{
 int j; public:
 void set_j(int n); int
mul();
};
```

После имени класса D стоит двоеточие, за которым стоит ключевое слово public и имя класса B. Это означает, что класс D будет наследовать все компоненты класса B. Само ключевое слово public информирует компилятор о том, что т.к. класс B будет наследоваться, то все открытые элементы базового класса будут открытыми элементами производного класса. Однако все закрытые элементы базового класса остаются закрытыми.



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

Пример:

```
// Простой пример наследования.

#include <iostream.h>

// Задание базового класса class base {
 int i; public:
 void set_i(int n); int
get_i();
};

// Задание производного класса class derived :
public base
{
 int j; public:
 void set_j(int n); int
mul();
};

// Установка значения i в базовом классе void base::set_i(int
n)
{
 i = n;
}

// Возврат значения i в базовом классе int base::get_i()
{
 return i;
}
```



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

```
// Установка значения j в производном классе void derived::set_j(int
n)
{
j = n;
}

// Возврат значения i из base и, одновременно, j из derived int derived::mul()
{
// производный класс может вызывать функции-члены базового класса return j * get_i();
}

main()
{
derived ob;

ob.set_i(10); // загрузка i в base ob.set_j(4); //
загрузка j в derived

cout << ob.mul(); // вывод числа 40

return 0;
}
```

При определении mul() вызывается функция get\_i()- базового класса B, а не производного D, что указывает на то, что открытые члены базового класса становятся открытыми членами производного. Но в функции mul() вместо прямого доступа к i, необходимо вызывать get\_i(), потому что закрытые члены базового класса(i) остаются закрытыми для производных классов.

Виртуальные функции.

Проблема: как будет вызываться функция производного класса, имеющая такое же название, что функция базового класса.



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

Пример.

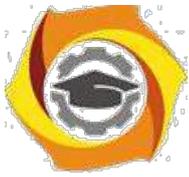
```
#include <stdio.h> class base
{ public:
 int i;
 base(int x); //конструктор void func()
 {
 printf("Базовая функция %d",i); return;
 };
};

//текст конструктора base::base(int x)
{ i=x;
return; };

class der1: public base
{ public:
 der1(int x):base(x) {}; //конструктор
 void func()
 {
 printf("Функция из производного класса %d", i*i); return;
 }
};

main() { base * pc; //указатель на базовый класс
 base ob(2); //создать экземпляр объекта базового класса der1 ob1(2); //создать
 экземпляр объекта производного класса pc=&ob; //указатель на объект
 базового класса pc->func(); //вызов функции базового класса pc=&ob1;
 //указатель на объект производного класса pc->func(); //попытка вызова
 функции производного класса return 0;
}
```

На первый взгляд кажется, что в первом случае будет вызываться функция базового класса, а во втором функция производного. Однако при проверке Вы легко убедитесь, что и в том и в другом



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

случае будет вызвана функция базового класса. При этом, компилятору трудно «понять», какую реально функцию мы имеем в виду и он на стадии компилирования подставляет во всех тех случаях, где встречается имя func() адрес функции базового класса.

Такой процесс установки адресов называется "ранним связыванием" или "статическим связыванием". Если же мы хотим, чтобы во втором случае, т.е.

когда указатель pc указывал на производный класс вызывалась функция этого класса, ее еще в базовом классе следует указать как виртуальную. В нашем случае вместо строки void func() следует написать virtual void func(). Это напоминает «перегрузку» функций, но «перегружаемые» функции отличаются друг от друга типом или аргументами, здесь же функции должны быть идентичны.

В случае использования виртуальных функций адрес вызываемой функции будет определяться в процессе выполнения кода программы. Такой процесс называется "поздним связыванием" или "динамическим связыванием".

Пример: #include

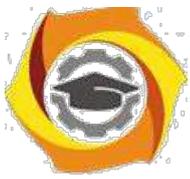
```
<stdio.h> class base {
public:
 int i;
 base(int x); //конструктор virtual void func()
 {
 printf("Базовая функция %d\n",i); return;
 };
};

//текст конструктора base::base(int x)
{ i=x;
return; };

class der1: public base {
public:
 der1(int x):base(x) {}; //конструктор void func()
 {
 printf("Функция из производного класса %d\n", i*i); return;
 } };

class der2: public base { public:
 der2(int x):base(x) {}; //конструктор
};

main() { base * pc; //указатель на базовый класс
```



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

```
base ob(2); //создать экземпляр объекта базового класса der1 ob1(2); //создать
экземпляр объекта производного класса 1 der2 ob2(2); //создать экземпляр объекта
производного класса 2 pc=&ob; //указатель на объект базового класса pc-
>func(); //вызов функции базового класса pc=&ob1; //указатель на объект
производного класса 1 pc->func(); //попытка вызова функции производного класса
pc=&ob2; //указатель на объект производного класса 2 pc->func(); //попытка
вызова функции производного класса return 0;

}
```

Введен еще один производный класс. В нем функция func() не определена. В этом случае будет вызываться функция класса родителя. Т.е. появится строка: «Базовая функция 2». При этом, принцип прост: если Вы хотите, чтобы вызывалась функция родительского класса, не определяйте ее в производном. Еще один вопрос может возникнуть в связи с данным примером: как быть, если мы хотим, чтобы для класса объектов der2 вызывалась функция класса der1. Решение очень просто - сделайте класс der2 наследником не класса base, а класса der1.

В производных классах функция, определенная в базовом классе как виртуальная может определяться, а может и нет. Если Вы хотите, чтобы во всех производных классах обязательно была определена виртуальная функция, то в базовом классе ее надо определить следующим образом:

```
virtual void func() = 0;
```

В этом случае базовый класс называется агрегатным и от него нельзя будет создавать экземпляры объектов, зато во всех производных классах компилятор «потребует» определить данную виртуальную функцию и, тем самым, уменьшить вероятность ошибок.

### 3. Порядок выполнения задания

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первая задачи, решение которой производится в методе класса№1; №по журналу +3 – вторая задача, решение которой производится в методе класса№2; и №по журналу +5 – третья задача, решение которой производится в методе класса №3 (если достигнуто окончание списка вариантов зданий, то перейти в его начало).

3.2. Исследовать как меняется код программы в зависимости от последовательности наследования:

- 1,2,3;
- 2,1,3;
- 2,3,1;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- 3,2,1;
- 3,1,2; - 1,3,2.

Отразить выводы в отчете.

- 3.3. Составить программу по заданию.
- 3.4. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.
- 3.5. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и скрин-шот результата выполнения задания и представить его на проверку.

#### **4. Варианты заданий:**

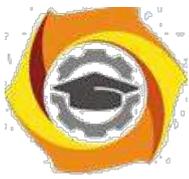
##### **Составить функции пользователя для следующих задач**

1. Составить программу для перевода длины в метрах в длину в сантиметрах, определив функцию, выполняющую это преобразование и передавая длину в метрах в качестве параметра.
2. Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив функцию, выполняющую это действие, и передавая массивы в качестве параметра.
3. Даны числа S, T. Получить с использованием функции пользователя  $F(T, -2S, 1.17) + F(2.2, T, S-T)$  где  $F(A, B, C) = (2A - B - \sin(C)) / (5 + C)$
4. Составить программу перевода двоичной записи натурального числа в десятичную, описав соответствующую функцию с параметром. Перевод осуществлять для чисел, вводимых с клавиатуры. Признак конца ввода - число 0.
5. Даны числа S, T. Получить с использованием функции пользователя с параметрами  $G(1, \sin(S)) + 2G(T * S, 24) - G(5, -S)$ , где  $G(A, B) = (2A + B * B) / (A * B * 2 + B * 5)$ .
6. Составить программу для расчета значений гипотенузы треугольника, определив функцию, выполняющую этот расчет. Катеты передаются в качестве параметров.
7. Найти периметр десятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, которые передаются функции в качестве параметров из основной программы.
8. Найти периметр шестиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами. Координаты передаются функции в качестве параметров из основной программы.



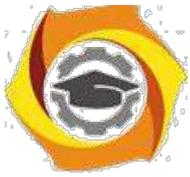
**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

9. Найти площадь пятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, и процедуру вычисления площади треугольника по трем сторонам. Описать функции с соответствующими формальными параметрами.
10. Составить программу вывода на экран всех натуральных чисел, не превосходящих  $N$  и делящихся на каждую из своих цифр. Описать соответствующую функцию, получающую из основной программы в качестве параметра натуральное число и возвращающую TRUE, если оно удовлетворяет указанному условию.
11. Используя подпрограмму - функцию, составить программу для нахождения максимального из трех чисел. Числа передаются функции в качестве параметров.
12. Используя подпрограмму - функцию, составить программу для печати знаков трех чисел, введенных с клавиатуры и передаваемых функции в качестве параметра.
13. Используя подпрограмму - функцию, составить программу для возведения чисел в целую положительную степень. Число передается функции в качестве параметра из основной программы. Расчет вести для чисел, пока не будет введено число, равное 0.
14. Используя подпрограмму - функцию, составить программу для вычисления функции  $Z=(X1+Y1)/(X1*Y1)$ , где  $X1$  - первый корень уравнения  $X^2-4*X-1=0$ ;  $Y1$  - первый корень уравнения  $2*Y^2 + A*Y - A^2 = 0$  ( $A$  - произвольное).
15. Задав функцию, вывести на печать средние арифметические двух массивов, введенных с клавиатуры. Массив передается функции в качестве параметра.
16. Задав функцию, рассчитать и вывести на печать максимальные значения в трех парах чисел, вводимых с клавиатуры. Пара чисел передается функции в качестве параметра.
17. Найти периметр восьмиугольника, координаты вершин которого заданы. Определить функцию вычисления расстояния между двумя точками, заданными своими координатами. Координаты передать функции в качестве параметров.
18. Даны четыре пары чисел. Получить с использованием функции пользователя наибольший общий делитель для каждой пары.
19. Даны числа  $A, B, C$ . Получить с использованием функции пользователя наименьшее значение. Числа передаются функции из основной программы в качестве параметров.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

20. Даны числа  $x = 1, 2, \dots, N$ . Получить с использованием функции пользователя значения  $3 * P(x+3) * P(x)$  для заданных  $x$ , где  $P(x) = 10 * x^3 - 14 * x^2 + 12 * x - 2$ .
21. Составить программу для расчета значений катета треугольника, определив функцию, выполняющую этот расчет. Гипотенуза и второй катет передаются в качестве параметров.
22. Даны целые числа  $a, b, c, d$ . Проверить с использованием функции пользователя их четность. Число для проверки передается в функцию в качестве параметра из основной программы.
23. Для каждого из 10 введенных с клавиатуры чисел напечатать сообщение: является ли оно простым или нет, описав функцию логического типа, возвращающую значение “ИСТИНА”, если число, переданное ей в качестве параметра, является простым.
24. Даны числа  $S, T$ . Получить с использованием функции пользователя  $Y(T, S) = G(12, S) + G(T, S) - G(2S - 1, S * T)$ , где  $G(A, B) = (2 * A + B * B) / (A * B * 2 + B * 5)$ .
25. Определите функцию, определяющую, какой целой степенью числа 2 является ее аргумент (если число не является степенью двойки - выдать соответствующее сообщение).
26. Определите функцию, подсчитывающую сумму  $N$  первых элементов целочисленного массива  $A$ .  $N$  и массив  $A$  передать в качестве параметров.
27. Вычислить количество простых чисел, не превосходящих заданного  $N$ . Описать функцию логического типа, возвращающую значение true, если число простое и false в противном случае.
28. Используя подпрограмму - функцию с параметрами, составить программу для вычисления функции  $F(x, y) = (2x^3 - 4x^2 + x + 1) / (9y^3 + y + 4) + 3y^2 + 5y$ .
29. Составить программу для перевода веса в граммах в вес в килограммах, определив функцию, выполняющую это преобразование. Вес в граммах передается функции в качестве параметра.
30. Даны числа  $S, T$ . Получить с использованием функции пользователя  $G(12, S) + G(T, S) - G(2S - 1, S * T)$  где  $G(A, B) = (2 * A + B * B) / (A * B * 2 + B * 5)$ .



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

## Исследование управляющих операторов Java

### 1. Цель занятия:

Выработать умения и навыки по составлению программ с инкапсулированными данными (инкапсуляция) и перегруженными методами (полиморфизм).

### 2. Рекомендации:

Изучить материалы лекций №№4,5.

#### Краткая теория

#### Классы. «Перегруженные» функции.

ООП позволяет разбить задачу на ряд самостоятельных связанных между собой подзадач. Каждая подзадача содержит коды и данные, относящиеся к объекту, что упрощает решение задачи в целом и позволяет решать задачи большего объема.

Понятие объекта тесно связано с понятием класса.

Класс – это дальнейшее развитие понятия структуры. Он позволяет создавать новые типы и определять функции, манипулирующие с этими типами. По сути, класс это тип.

Объект - это представитель определенного класса. В общем, виде объект можно рассматривать как переменную, определенную программистом. ООП использует механизмы инкапсуляции, полиморфизма и наследования. Инкапсуляция позволяет создавать объекты - данные, процедуры и функции, манипулирующие с этими данными. Данные, доступные для использования внутри объекта - private, данные доступные извне - public.

Полиморфизм позволяет одно имя функции использовать для решения разных задач (общих для класса действий). В зависимости от данных выполняются те или иные действия.

Наследование позволяет одному объекту наследовать свойства другого объекта, т.е. порожденный класс наследует свойства родительского класса и добавляет собственные свойства.

Класс используется для создания объектов. Основная форма имеет вид:

```
class имя класса
```

```
{
```

```
 закрытые функции и переменные public:
```

```
 открытые функции, функции-члены и переменные
```

```
}
```

```
 список объектов; //не является обязательным
```

Закрытые функции и переменные это члены (members) класса, которые доступны только для других членов этого класса. Открытые функции и переменные доступны для любой части программы, в которой находится класс. Функции, объявленные внутри описания класса, называются функциями членами (member functions).

Для определения функций-членов используется форма:

```
тип имя класса :: имя функции-члена (параметры)
```

```
{
```

```
 тело функции
```

```
}
```

Два двоеточия :: после имени класса называются операцией расширения области видимости (scope resolution operator).

Класс только определяет тип объектов, а сами объекты не задает (память для их хранения не выделяется). Для создания объектов имя класса используется как спецификатор типа данных. После создания объекта к открытым членам класса можно обращаться, используя операцию «точка».

Пример.

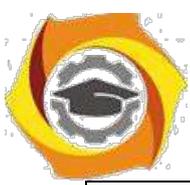
```
#include <iostream.h>
```

```
class class1
```

```
{ int a;
```

```
public:
```

```
//объявлен класс с именем class1
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

|                                      |  |
|--------------------------------------|--|
| //доступна для функций членов class1 |  |
|--------------------------------------|--|

```
int kwadrat(int b); //функция член класса class1
};
int class1::kwadrat(int b) //определение функции «kwadrat()»-члена класса class1 {
a=b*b; return a; }
void main()
{
```

|                    |                                    |
|--------------------|------------------------------------|
| class1 c; int z,x; | //создается объект «С» типа class1 |
|--------------------|------------------------------------|

```
x=3; z=c.kwadrat(x);
cout<<"\n"<<z<<"\n"; //вычисление и вывод
квадрата числа 3 }
```

«Перегруженные» функции

Две или более функции называются перегруженными, если они имеют одно и то же имя. Обычно функции отличаются количеством и типом аргументов. Транслятор автоматически на основании количества или типов аргументов выберет правильный вариант.

Пример.

```
#include <iostream.h>
```

```
void k(int a); //прототип первой функции void k(int a, float b);
```

```
//прототип второй функции
```

|                    |                           |
|--------------------|---------------------------|
| void k(int a)<br>{ | //описание первой функции |
|--------------------|---------------------------|

```
cout << a << "\n";
```

```
}
```

|                             |                              |
|-----------------------------|------------------------------|
| void k(int a,<br>float b) { | //описание второй<br>функции |
|-----------------------------|------------------------------|

```
cout << a << "\n" << b << "\n";
```

```
} main()
```

```
{
```

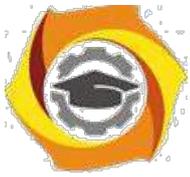
|                                   |                                                     |
|-----------------------------------|-----------------------------------------------------|
| k(4); k(5,<br>10.2);<br>return 0; | //вызов первой<br>функции //вызов<br>второй функции |
|-----------------------------------|-----------------------------------------------------|

```
}
```

### 3. Порядок выполнения задания

- 3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первая задача; №по журналу +3 – вторая задача; и №по журналу +5 – третья задача (если достигнуто окончание списка вариантов заданий, то перейти в его начало).
- 3.2. Составить программу по заданию.
- 3.3. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.
- 3.4. Оформить отчет для каждой из 3 задач, включив в него задание, блоксхему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.

### 4. Варианты заданий:



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Составить методы с одинаковыми названиями для решения следующих задач**

1. Найти среднее арифметическое положительных чисел, введенных с клавиатуры. Всего ввести  $N$  различных чисел.
2. Ввести с клавиатуры  $N$  чисел. Найти сумму тех из них, которые принадлежат интервалу  $(2;9)$ .
3. Для  $N$  введенных с клавиатуры чисел найти сумму положительных кратных 3.
4. Для арифметической прогрессии 4, 9, 14, 19... найти первые  $n$  членов этой прогрессии.
5. Найти сумму отрицательных значений функции  $Z=\sin(5-x)/\cos(x-2)$  для  $x$ , изменяющегося на отрезке  $[-5,12]$  с шагом 1.
6. Найти среднее арифметическое отрицательных чисел, введенных с клавиатуры. Всего ввести  $N$  различных чисел.
7. Найти среднее арифметическое чисел, принадлежащих отрезку  $[2,184]$ , кратных 2 и введенных с клавиатуры. Всего ввести  $N$  различных чисел.
8. Найти сумму значений функции, больших 2  $Z=\sin(1/x)+5\cos(1/(x-3))+x$  для  $x$ , изменяющегося на отрезке  $[-3,8]$  с шагом 1.
9. Найти  $n$  членов последовательности  $x_1 = x_2 = x_3 = 1$ ;  $x_k = x_{k-1} + x_{k-3}$ .
10. Вычислить последовательность  $N$  чисел  $A_0 = x$ ,  $A_1 = 2$ ,  $A_k = A_{k-1} - A_{k-2}$ .
11. Для  $x_1 = 0,3$  и  $x_2 = -0,3$  найти  $x_k = k + \sin(x_{k-2})$  для  $k$ , изменяющегося следующим образом:  $k = 3,4,\dots,14$ .
12. Составить таблицу перевода дюймов в сантиметры для расстояний от 1 до 13 дюймов с шагом 1.
13. Вывести на печать значения функции, меньшие 2,  $Z=\sin(1/x)+5\cos(x-3)+x$  для  $x$ , изменяющегося на отрезке  $[-7,4]$  с шагом 1.
14. Напечатать таблицу значений функции  $Y = \operatorname{tg}(x/b) + x/(b-2)$  для  $x$ , изменяющегося от 0 до 10 с шагом 1 ( $b$  - произвольное число).
15. Вычислить  $N$ -ый член последовательности  $x_k = x_{k-2} - x_{k-1}$ ,  $x_0 = 2,4$   $x_1 = 3,8$ .
16. Составить таблицу перевода суток (от 1 до 7) в часы, минуты, секунды.
17. Вычислить  $N$ -ый член последовательности  $x_k = x_{k-1} + (2/3)x_{k-2} + 1$ ,  $x_1 = -1$ ,  $x_2 = 1,38$ .
18. Напечатать значения функции  $z = 1/(x-2) + 1/(x-5) + \ln(12,8-X)$  для  $x$ , изменяющегося на отрезке  $[-4,14]$  с шагом 1.
19. Вывести на печать отрицательные значения функции  $z=\sin(5-x)/\cos(x-2)$  для  $x$ , изменяющегося на отрезке  $[-6,13]$  с шагом 1 (учесть область допустимых значений функции).
20. Из  $N$  введенных с клавиатуры чисел напечатать кратные 3 и меньшие 58. 21. Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(1,11)$  и являются четными.
22. Из  $N$  введенных с клавиатуры чисел напечатать положительные, кратные 3.
23. Вывести на печать значения функции  $z = \sin(x/(x-2))$ , находящиеся в интервале  $(-0,4;0,8)$  для  $x$ , изменяющегося от 8 до -6 с шагом 1.
24. Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(2;9)$ .
25. Для геометрической прогрессии 2, 6, 18, 54, 162 ... определить первые  $n$  членов этой прогрессии.
26. Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые не принадлежат интервалу  $(1;5)$ .
27. Найти  $n$  членов последовательности  $x_1 = x_2 = x_3 = 1$ ;  $x_k = x_{k-1} - 2x_{k-3}$ .
28. Вычислить последовательность  $N$  чисел  $A_0 = x$ ,



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

$$A_1 = 2, A_k = A_{k-1} + A_{k-2}$$

29. Составить таблицу перевода килограммов (от 1 до 13) в граммы с шагом

30. Найти сумму значений функции  $Y = \cos(x/A) + x/(A-2)$  для  $x$ , изменяющегося от 2 до 13 с шагом 1 ( $A$  - произвольное число).

## Практическое занятие №4 Решение задач в Java

### 1. Цель занятия:

Выработать умения и навыки по составлению программ, содержащий классы, наследующие другие классы.

### 2. Рекомендации:

Изучить материалы лекций №№4-6.

#### Краткая теория

#### Наследование. Виртуальные функции. Указатели на объекты

Наследование - это механизм посредством которого один класс

(производный

или потомок) может наследовать свойства другого класса (базового или предка).

Базовый класс определяет все качества, которые являются общими для всех производных классов.

Пример: //Базовый класс

```
class B {
 int i; public: void
 set_i(int n); int get_i();
};
```

//Производный класс D class D : public

```
B
{
 int j; public: void
 set_j(int n); int mul();
};
```

После имени класса D стоит двоеточие, за которым стоит ключевое слово

public и имя класса B. Это означает, что класс D будет наследовать все компоненты класса B.

Само ключевое слово public информирует компилятор о том,

что т.к. класс B будет наследоваться, то все открытые элементы базового класса

будут открытыми элементами производного класса. Однако все закрытые элементы базового класса остаются закрытыми.

Пример:

```
// Простой пример наследования. #include <iostream.h>
```

```
// Задание базового класса
```

```
class base {
 int i; public: void
 set_i(int n); int get_i();
};
```

```
// Задание производного класса class derived :
```

```
public base
{
 int j; public: void
 set_j(int n); int mul();
};
```

```
// Установка значения i в базовом классе
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

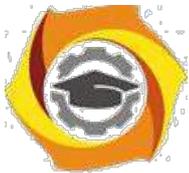
```
void base::set_i(int n)
{ i = n; }
// Возврат значения i в базовом классе
int base::get_i()
{ return i; }
// Установка значения j в производном классе
void derived::set_j(int n)
{ j = n; }
// Возврат значения i из base и, одновременно, j из derived
int derived::mul()
{
// производный класс может вызывать функции-члены базового класса
return j * get_i();
}
int main() { derived ob; ob.set_i(10); // загрузка i
в base ob.set_j(4); // загрузка j в derived
cout << ob.mul(); // вывод числа 40
return 0; }
```

При определении `mul()` вызывается функция `get_i()`- базового класса `B`, а не производного `D`, что указывает на то, что открытые члены базового класса становятся открытыми членами производного. Но в функции `mul()` вместо прямого доступа к `i`, необходимо вызывать `get_i()`, потому что закрытые члены базового класса (`i`) остаются закрытыми для производных классов. Виртуальные функции.

Проблема: как будет вызываться функция производного класса, имеющая такое же название, что функция базового класса. Пример.

```
#include <stdio.h>
class
base { public: int i;
base(int x); //конструктор void func()
{ printf("Базовая функция %d",i);
return;
};
};
//текст конструктора base::base(int x)
{ i=x;
return; };
class der1: public base
{ public:
der1(int x) :base(x) {} ; //конструктор void func()
{
printf("Функция из производного класса %d", i*i); return;
} };
int main() { base * pc; //указатель на базовый класс base ob(2); //создать
экземпляр объекта базового класса der1 ob1(2); //создать экземпляр объекта
производного класса pc=&ob; //указатель на объект базового класса pc->func();
//вызов функции базового класса pc=&ob1; //указатель на объект производного
класса
pc->func(); //попытка вызова функции производного класса return 0; }
```

На первый взгляд кажется, что в первом случае будет вызываться функция базового класса, а во втором функция производного. Однако при проверке Вы легко убедитесь, что и в том и в другом случае будет вызвана функция базового класса. При этом, компилятору трудно «понять», какую реально функцию мы имеем в виду и он на стадии компилирования подставляет во всех тех случаях, где встречается имя `func()` адрес функции базового класса. Такой процесс установки адресов называется "ранним связыванием" или "статическим связыванием". Если же мы хотим, чтобы во втором случае, т.е. когда указатель `pc` указывал на производный класс вызывалась функция этого класса, ее еще в базовом классе следует указать как виртуальную. В нашем случае вместо строки `void func()` следует написать `virtual void func()`.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Это напоминает «перегрузку» функций, но «перегружаемые» функции отличаются друг от друга типом или аргументами, здесь же функции должны быть идентичны. В случае использования виртуальных функций адрес вызываемой функции будет определяться в процессе выполнения кода программы. Такой процесс называется "поздним связыванием" или "динамическим связыванием". Пример:

```
#include <stdio.h> class
base { public: int i;
base(int x); //конструктор virtual void func()
{ printf("Базовая функция %d\n",i);
return;
}; };
//текст конструктора base::base(int x)
{ i=x; return; }; class der1: public
base { public:
der1(int x) :base(x) {}; //конструктор void func()
{
printf("Функция из производного класса %d\n", i*i); return;
} }; class der2: public base {
public:
der2(int x) :base(x) {}; //конструктор
}; main() { base * pc; //указатель на базовый класс base ob(2); //создать экземпляр
объекта базового класса der1 ob1(2); //создать экземпляр объекта производного
класса 1 der2 ob2(2); //создать экземпляр объекта производного класса 2 pc=&ob;
//указатель на объект базового класса pc->func(); //вызов функции базового класса
pc=&ob1; //указатель на объект производного класса 1 pc->func(); //попытка
вызова функции производного класса pc=&ob2; //указатель на объект
производного класса 2
pc->func(); //попытка вызова функции производного класса return 0; }
```

Введен еще один производный класс. В нем функция func() не определена. В этом случае будет вызываться функция класса родителя. Т.е. появится строка: «Базовая функция 2». При этом, принцип прост: если Вы хотите, чтобы вызывалась функция родительского класса, не определяйте ее в производном. Еще один вопрос может возникнуть в связи с данным примером: как быть, если мы хотим, чтобы для класса объектов der2 вызывалась функция класса der1. Решение очень просто - сделайте класс der2 наследником не класса base, а класса der1.

В производных классах функция, определенная в базовом классе как виртуальная может определяться, а может и нет. Если Вы хотите, чтобы во всех производных классах обязательно была определена виртуальная функция, то в базовом классе ее надо определить следующим образом: virtual void func() = 0;

В этом случае базовый класс называется агрегатным и от него нельзя будет создавать экземпляры объектов, зато во всех производных классах компилятор «потребует» определить данную виртуальную функцию и, тем самым, уменьшить вероятность ошибок. **3. Порядок выполнения задания**

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первая задачи, решение которой производится в методе класса №1; №по журналу +3 – вторая задача, решение которой производится в методе класса №2; и №по журналу +5 – третья задача, решение которой производится в методе класса №3 (если достигнуто окончание списка вариантов заданий, то перейти в его начало).

3.2. Исследовать как меняется код программы в зависимости от последовательности наследования:

- 1,2,3;
- 2,1,3;
- 2,3,1;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- 3,2,1;
- 3,1,2;
- 1,3,2.

Отразить выводы в отчете.

3.3. Составить программу по заданию.

3.4. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.

3.5. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.

#### **4. Варианты заданий:**

**Составить функции пользователя для следующих задач** 1. Составить программу для перевода длины в метрах в длину в сантиметрах, определив функцию, выполняющую это преобразование и передав длину в метрах в качестве параметра.

2. Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив функцию, выполняющую это действие, и передавая массивы в качестве параметра.

3. Даны числа S, T. Получить с использованием функции пользователя  $F(T, 2S, 1.17)+F(2.2, T, S-T)$  где  $F(A, B, C) = (2A-B-\sin(C))/(5+C)$

4. Составить программу перевода двоичной записи натурального числа в десятичную, описав

соответствующую функцию с параметром. Перевод осуществлять для чисел, вводимых с клавиатуры. Признак конца ввода - число 0. 5. Даны числа S, T. Получить с использованием функции пользователя с параметрами

$G(1, \sin(S))+2G(T*S, 24)-G(5, -S)$ , где  $G(A, B) = (2A+B*B)/(A*B*2+B*5)$ .

6. Составить программу для расчета значений гипотенузы треугольника, определив функцию, выполняющую этот расчет. Катеты передаются в качестве параметров. 7. Найти периметр десятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, которые передаются функции в качестве параметров из основной программы.

8. Найти периметр шестиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами. Координаты передаются функции в качестве параметров из основной программы.

9. Найти площадь пятиугольника, координаты вершин которого заданы. Определить процедуру

вычисления расстояния между двумя точками, заданными своими координатами, и процедуру вычисления площади треугольника по трем сторонам. Описать функции с соответствующими формальными параметрами.

10. Составить программу вывода на экран всех натуральных чисел, не превосходящих N и делящихся на каждую из своих цифр. Описать соответствующую функцию, получающую из основной программы в качестве параметра натуральное число и возвращающую TRUE, если оно удовлетворяет указанному условию.

11. Используя подпрограмму - функцию, составить программу для нахождения максимального из трех чисел. Числа передаются функции в качестве параметров.

12. Используя подпрограмму - функцию, составить программу для печати знаков трех чисел,

введенных с клавиатуры и передаваемых функции в качестве параметра.

13. Используя подпрограмму - функцию, составить программу для возведения чисел в целую



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

положительную степень. Число передается функции в качестве параметра из основной программы. Расчет вести для чисел, пока не будет введено число, равное 0.

14. Используя подпрограмму - функцию, составить программу для вычисления функции  $Z=(X1+Y1)/(X1*Y1)$ , где  $X1$  - первый корень уравнения  $X^2-4*X-1=0$ ;  $Y1$  - первый корень уравнения  $2*Y^2 + A*Y - A^2 = 0$  ( $A$  - произвольное).

15. Задав функцию, вывести на печать средние арифметические двух массивов, введенных с клавиатуры. Массив передается функции в качестве параметра.

16. Задав функцию, рассчитать и вывести на печать максимальные значения в трех парах чисел,

вводимых с клавиатуры. Пара чисел передается функции в качестве параметра.

17. Найти периметр восьмиугольника, координаты вершин которого заданы. Определить функцию вычисления расстояния между двумя точками, заданными своими координатами. Координаты передать функции в качестве параметров.

18. Даны четыре пары чисел. Получить с использованием функции пользователя наибольший общий делитель для каждой пары.

19. Даны числа  $A, B, C$ . Получить с использованием функции пользователя наименьшее значение. Числа передаются функции из основной программы в качестве параметров.

20. Даны числа  $x = 1, 2, \dots, N$ . Получить с использованием функции пользователя значения  $3*P(X+3)*P(X)$  для заданных  $x$ , где  $P(X) = 10*X^3 - 14*X^2 + 12*X - 2$ .

21. Составить программу для расчета значений катета треугольника, определив функцию, выполняющую этот расчет. Гипотенуза и второй катет передаются в качестве параметров.

22. Даны целые числа  $a, b, c, d$ . Проверить с использованием функции пользователя их четность.

Число для проверки передается в функцию в качестве параметра из основной программы.

23. Для каждого из 10 введенных с клавиатуры чисел напечатать сообщение: является ли оно

простым или нет, описав функцию логического типа, возвращающую значение "ИСТИНА", если число, переданное ей в качестве параметра, является простым. 24. Даны числа  $S, T$ . Получить с использованием функции пользователя  $Y(T,S)=G(12,S)+G(T,S)-G(2S-1,S*T)$ , где  $G(A,B)=(2*A+B*B)/(A*B*2+B*5)$ .

25. Определите функцию, определяющую, какой целой степенью числа 2 является ее аргумент

(если число не является степенью двойки - выдать соответствующее сообщение).

26. Определите функцию, подсчитывающую сумму  $N$  первых элементов целочисленного массива  $A$ .  $N$  и массив  $A$  передать в качестве параметров.

27. Вычислить количество простых чисел, не превосходящих заданного  $N$ . Описать функцию логического типа, возвращающую значение true, если число простое и false в противном случае.

28. Используя подпрограмму - функцию с параметрами, составить программу для вычисления функции  $F(X,Y) = (2X^3-4*X^2+X+1)/(9*Y^3+Y+4) + 3*Y^2+5*Y$ .

29. Составить программу для перевода веса в граммах в вес в килограммах, определив функцию, выполняющую это преобразование. Вес в граммах передается функции в качестве параметра.

30. Даны числа  $S, T$ . Получить с использованием функции пользователя  $G(12, S)+G(T, S)-G(2S-1, S*T)$  где  $G(A, B) = (2*A+B*B)/(A*B*2+B*5)$ .



**Методические рекомендации  
по дисциплине**

**Проектирование клиент-серверных приложений**

**МУкЛР1**

**Исследование web - приложений как клиент-серверных структур**

2.1.1 Цели занятия

Выработать навыки составления программ в виде WEB приложений, исследуя процесс их проектирования.

2.1.2 Теоретические основы и пример выполнения

Состав клиент-серверных приложений, как составного программного обеспечения (ПО), очевидно, входят серверная и клиентская части. Как правило, обе эти составляющие приложения разрабатываются как интерфейсные продукты, т.е. содержащие специальные модули (блоки) для обеспечения связи человека и вычислительной среды. От этого зависит, в конечном итоге, насколько простым или сложным будет использование приложения, а также его оптимальное функционирование.

Поэтому, важно научиться составлять удобные в использовании и функционально оптимальные клиент-серверные продукты.

Составление web-приложения выполняется в среде разработки Visual Studio на языке программирования C# [7, 8].

В настоящее время, C# включен в семейство продуктов этой среды разработки ПО на различных языках высокого уровня – мощной интегрированной среде разработки приложений различного уровня сложности и назначения. В этом пакете используется единая интегрированная среда разработки (IDE), состоящая из нескольких элементов: строки меню, панели инструментов, различных закрепленных или автоматически скрываемых окон инструментов в левой, нижней или правой областях, а также области редакторов. Набор доступных



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

окон инструментов, меню и панелей инструментов зависит от типа проекта или файла, в котором выполняется разработка (рисунок 2.3).

5

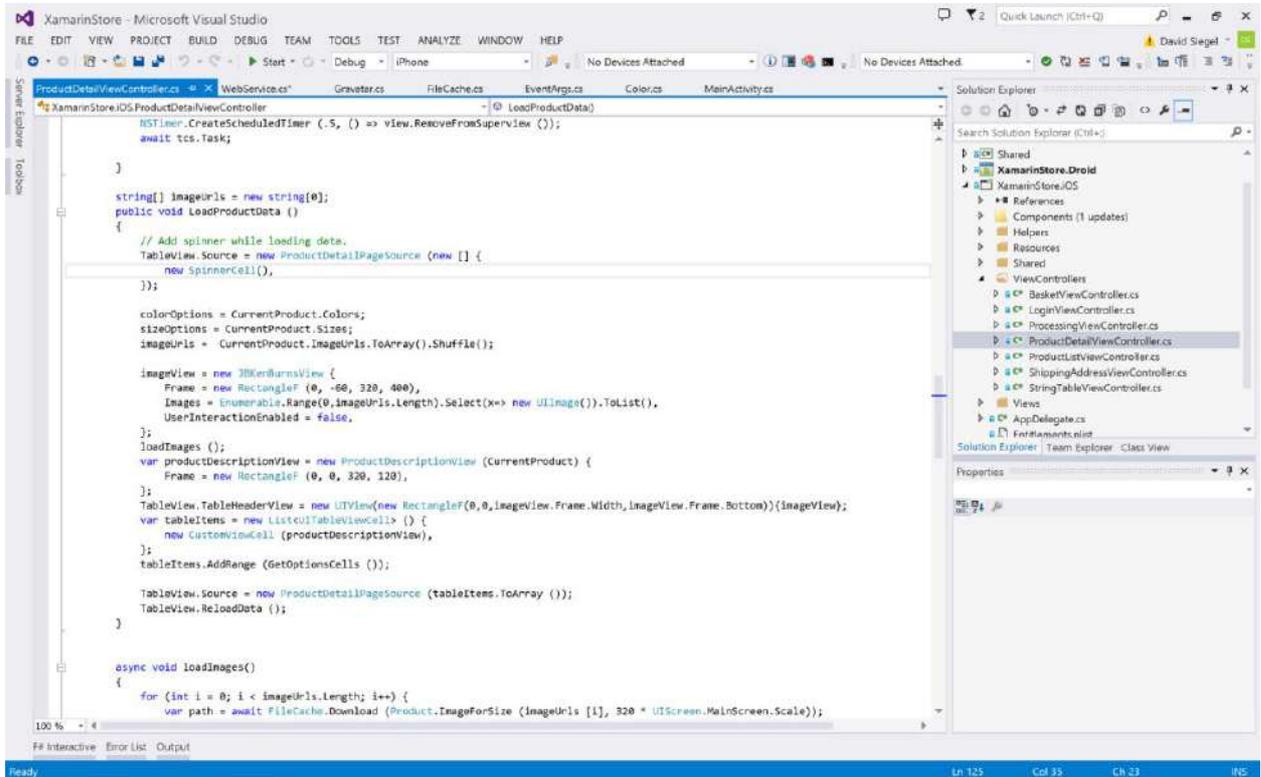
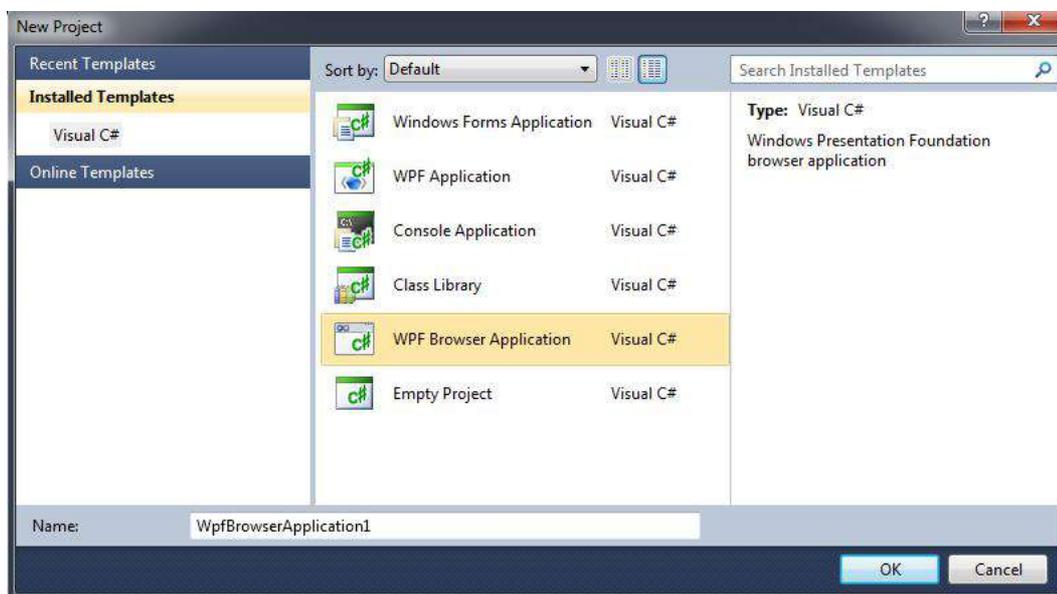


Рисунок 2.3 – Окно среды разработки Visual Studio

39. этой интерактивной среде разработки программ можно создавать как приложения Windows Forms, так и web приложения в виде browser application (рис.2.4).





## Рисунок 2.4- Создание проекта для web - приложения

6

Среда разработки автоматически формирует весь необходимый код во время «перетаскивания» на предоставляемую форму нужных элементов управления.

Далее, написание кода приложения очень похоже на работу в стандартном проекте Windows Form, когда интерфейс приложения составляется «перетаскиванием» элементов управления на исходную форму.

### 2.1.2.1 Основные элементы графического интерфейса пользователя

Для создания графических интерфейсов web - приложений используется обширная библиотека Form языка C#. Принцип составления оконного интерфейса, как указывалось ранее, прост - создается пустая форма, которая наследует (использует) библиотеку готовых решений Form. Далее, перетаскивая на эту форму необходимые элементы управления и отображения составляется клиентская часть для решения задачи, указанной на практическом занятии.

Элементов управления в указанной библиотеке множество, рассмотрим только часто употребляемые и необходимые для решения поставленных на занятии задач.

Основными элементами форм являются:

- В label – метка для отображения справочной информации;
- В textBox – текстовое поле для вводимых и выводимых данных;
- В button – кнопка для исполнения требуемых действий.

Приведем пример проектирования web - приложения и покажем как программируются указанные элементы интерфейса.

### 2.1.2.2 Пример выполнения задания

Задание: составить web – приложение для отображения обучающего контента.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Так выглядит стартовая форма приложения с размещенными на ней необходимыми кнопками, текстовыми полями и тому подобное (рисунок 2.5).

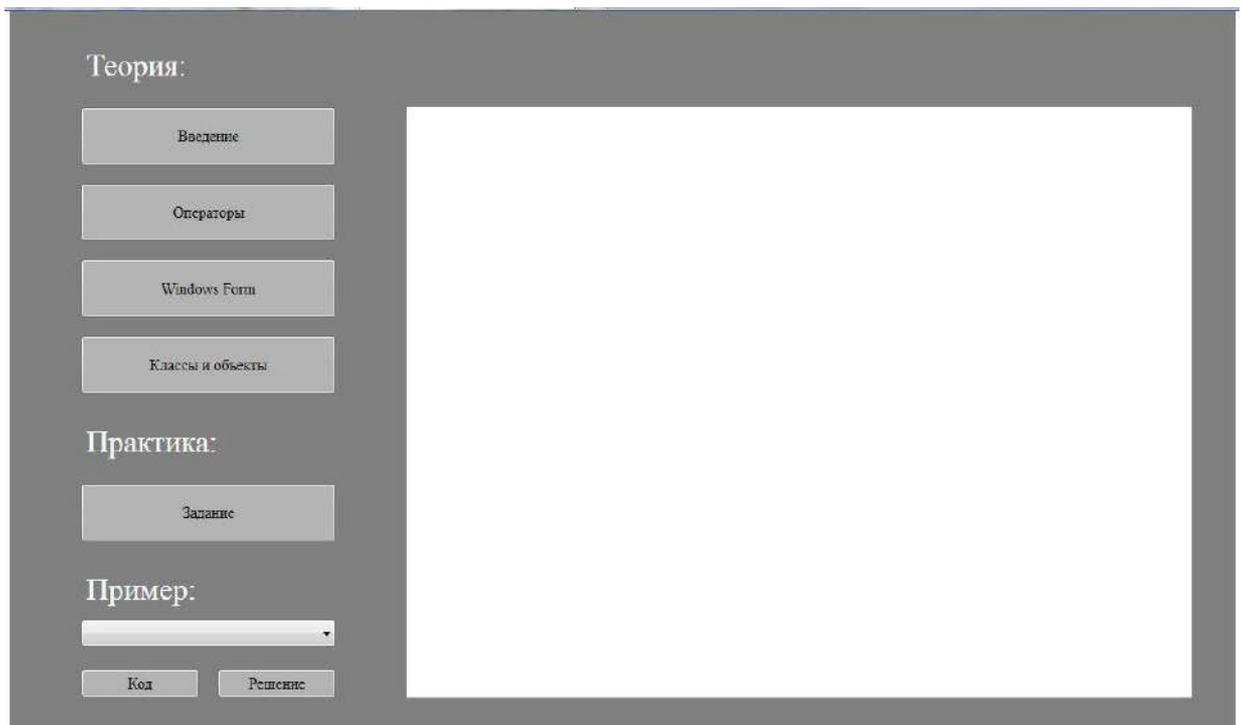


Рисунок 2.5- Стартовое окно приложения

Ее код в хaml представлении задается в виде таблицы заданных размеров (листинг 2.1).

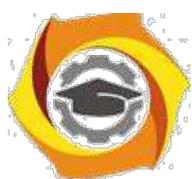
#### Листинг 2.1- Описание формы

```
<Grid Height="684" Width="1130" Background="Gray">
```

Соответствующий этой странице код в C# показан в листинге 2.2.

#### Листинг 2.2- Программный код формы 1

```
8) Необходимые автоматически подключаемые библиотеки
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*using System.Windows.Media;*



```
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
```

в Пространство имен проекта

```
namespace WpfBrowserApplication1
{
 9) Наследование класса Page со всеми элементами управления
 public partial class Page1 : Page
 {
 // Конструктор стартовой формы, инициализирующий ее
 КОМПОНЕНТЫ
 public Page1()
 {
 InitializeComponent();
 }

 private void button1_Click(object sender, RoutedEventArgs e)
 {
 string puti = "I:\\Pril\\Lex";
 webBrowser1.Navigate(puti + "\\\" + "1.mht");
 }

 private void button2_Click(object sender, RoutedEventArgs e)
 {
 string puti = "I:\\Pril\\Lex";
 webBrowser1.Navigate(puti + "\\\" + "2.mht");
 }

 private void button3_Click(object sender, RoutedEventArgs e)
 {
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
string puti = "I:\\Pril\\Lex";
```



```
 webBrowser1.Navigate(puti + "\\\" + "3.mht");
 }

 private void button4_Click(object sender, RoutedEventArgs e)
 {
 string puti = "I:\\Pril\\Lex";
 webBrowser1.Navigate(puti + "\\\" + "4.mht");
 }

 private void button5_Click(object sender, RoutedEventArgs e)
 {
 string puti = "I:\\Pril\\Lex";
 webBrowser1.Navigate(puti + "\\\" + "p1.mht");
 }

 private void button6_Click(object sender, RoutedEventArgs e)
 {
 if (comboBox1.SelectedIndex == 0)
 {
 string puti = "I:\\Pril\\Lex";
 webBrowser1.Navigate(puti + "\\\" +
"Lab1.mht");
 }

 if (comboBox1.SelectedIndex == 1)
 {
 string puti = "I:\\Pril\\Lex";
 webBrowser1.Navigate(puti + "\\\" +
"Lab2.mht");
 }
 }
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
if (comboBox1.SelectedIndex == 2)
```



```
{
 string puti = "I:\\Pril\\Lex";
 webBrowser1.Navigate(puti + "\\\" +
"Lab3.mht");
}
}

private void button7_Click(object sender, RoutedEventArgs e)
{
 if (comboBox1.SelectedIndex == 0)
 {
 NavigationService.Navigate(new
Uri("/Page2.xaml", UriKind.Relative));
 }

 if (comboBox1.SelectedIndex == 1)
 {
 NavigationService.Navigate(new
Uri("/Page3.xaml", UriKind.Relative));
 }

 if (comboBox1.SelectedIndex == 2)
 {
 NavigationService.Navigate(new
Uri("/Page4.xaml", UriKind.Relative));
 }
}
}
}
```

Из листинга видно, что в самом начале автоматически создается стартовая страница Page1 со всеми необходимыми библиотеками, которая наследует класс Page, в котором есть все необходимые элементы управления. Конструктор этой



страницы при обращении к ней инициализирует все размещенные на ней компоненты (см. комментарии в листинге 2.2).

На этой странице размещены кнопки управления (компонент `button`), «выпадающие» меню для выбора одного вида занятий из списка (элемент `comboBox`), метка для помещения заглавий разделов страницы (элемент `label`), а также окно для просмотра выбранного контента (`webBrowser`).

### 2.1.2.3 Разработка страниц раздела «Теория»

Срабатывают управляющие элементы, описанные выше, по клику на них левой клавишей мышки. При этом включаются их обработчики событий. Например, для отображения материала раздела «Введение» после нажатия на эту кнопку на стартовой странице выполняется следующий код обработчика событий этой кнопки (листинг 2.3).

#### Листинг 2.3- Код обработчика события «Нажатие» кнопки «Введение»

```
private void button1_Click(object sender, RoutedEventArgs e)
{
 string puti = "I:\\Pril\\Lex";
 webBrowser1.Navigate(puti + "\\\" + "1.mht");
}
```

Видно, что строковой переменной `puti` присваивается путь к соответствующему файлу. Далее компоненту этой страницы `webBrowser1`, который нужен для отображения выбираемого контента на ней, присваивается содержимое файла, который находит и возвращает метод `Navigate()` согласно пути, который храниться в переменной `puti`.

Аналогично обрабатываются события использования других компонентов страницы.

Так, чтобы отобразить нужную теорию необходимо будет нажать соответствующую кнопку на стартовой форме.



Т.е. при нажатии кнопки идет обращение к файлу, берется из него информация и выводится на экран (рисунки 2.6, 2.7).

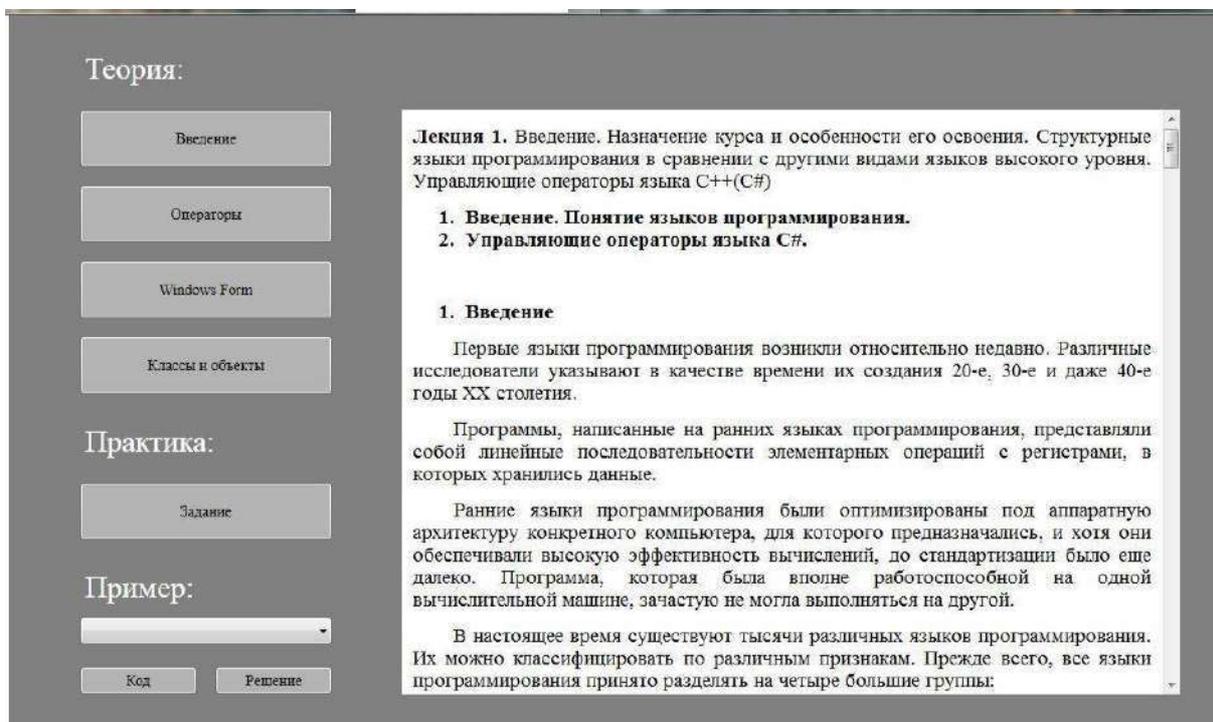


Рисунок 2.6 – Отображение контента в области «Теория»

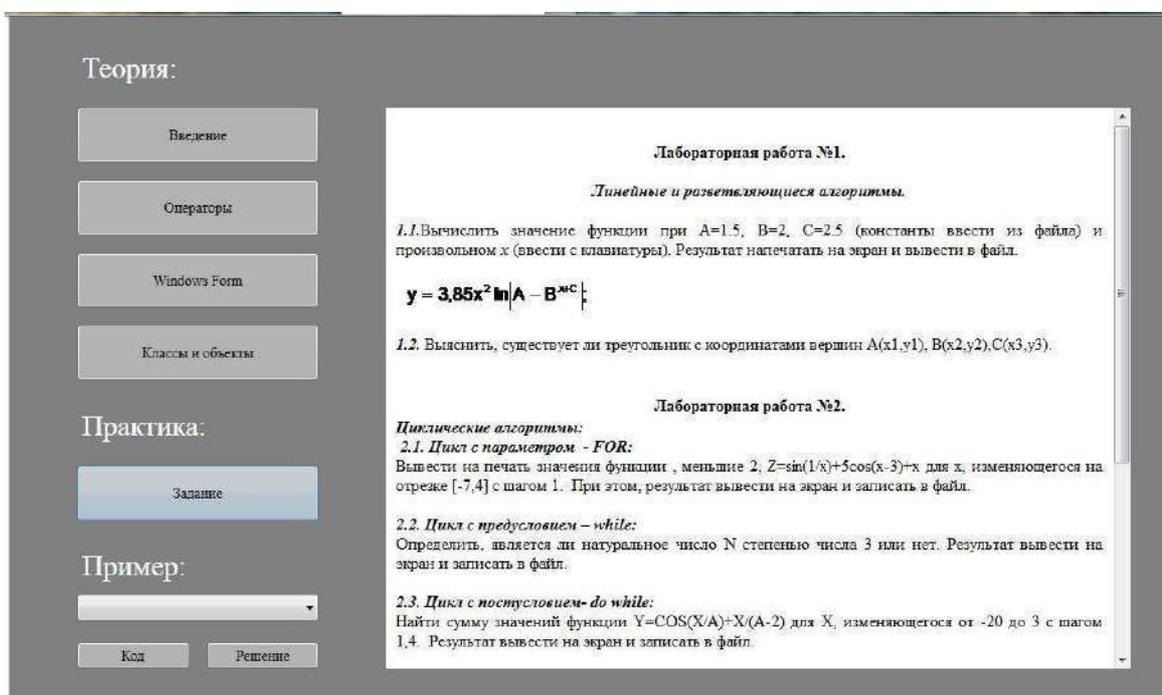
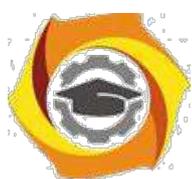


Рисунок 2.7 – Отображение контента в области «Практика»

Как было сказано выше, файлы контента должны быть представлены в формате html и размещены в папке Lех по указанному пути.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Теперь нужно разработать следующие страницы приложения.



#### 2.1.2.4 Разработка страниц раздела «Практика»

Этот раздел стартовой страницы нужен для реализации практической части курса. Он содержит кнопку «Задание», по нажатию на которую выводится для прочтения текст задания для практических занятий и лабораторных работ (листинг 2.4).

Листинг 2.4- Код обработчика события кнопки «Задание»

```
private void button5_Click(object sender,
RoutedEventArgs e)
{
 string pup = "I:\\ Pril\\Lex";
 webBrowser1.Navigate(pup + "\\\" + "p1.mht");
}
```

7. разделе «Пример» находится comboBox для того, чтобы можно было выбрать нужную лабораторную работу, которые задаются списком в его свойстве Select item (рисунок 2.8).

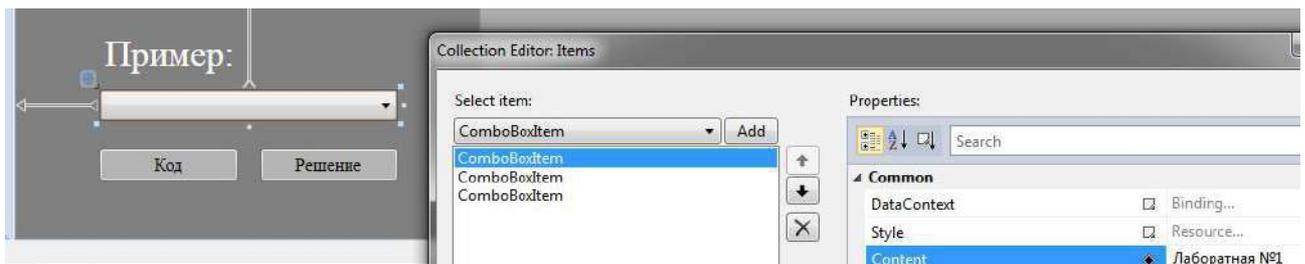
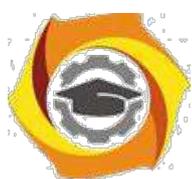


Рисунок 2.8 – Определение списка лабораторных работ

После выбора номера лабораторной работы по нажатию кнопки «Код» должен быть переход на страницу этой лабораторной работы (листинг 2.5).

Листинг 2.5- Код обработчика события кнопки «Код»

```
private void button6_Click(object sender,
RoutedEventArgs e)
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

{



```
if (comboBox1.SelectedIndex == 0)
{
 string pup = "I:\\ Pril\\Lex";
 webBrowser1.Navigate(pup + "\\\" + "Lab1.mht");
}

if (comboBox1.SelectedIndex == 1)
{
 string pup = "I:\\ Pril\\Lex";
 webBrowser1.Navigate(pup + "\\\" + "Lab2.mht");
}

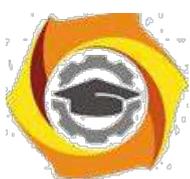
if (comboBox1.SelectedIndex == 2)
{
 string pup = "I:\\ Pril\\Lex";
 webBrowser1.Navigate(pup + "\\\" + "Lab3.mht");
}
}
```

Видно, что выбор нужной лабораторной работы выполняется через удовлетворение соответствующему условию в приведенном обработчике событий. При этом, в webBrowser1 загружается текст программы соответствующей зада-чи.

Кнопка «Решение» должна запускать еще одну страницу для выбранной лабораторной работы, где можно ввести исходные данные и получить решение в интерактивном режиме (листинг 2.6).

Листинг 2.6- Код обработчика события кнопки «Решение»

```
private void button7_Click(object sender,
RoutedEventArgs e)
{
 if (comboBox1.SelectedIndex == 0)
 {
 NavigationService.Navigate(new
Uri("/Page2.xaml", UriKind.Relative));
 }
}
```



```
}

if (comboBox1.SelectedIndex == 1)
{
 NavigationService.Navigate(new
Uri("/Page3.xaml", UriKind.Relative));
}
if (comboBox1.SelectedIndex == 2)
{
 NavigationService.Navigate(new
Uri("/Page4.xaml", UriKind.Relative));
}
}
```

Из этого обработчика событий видно, что по выполнению соответствующего условия метод `Navigate(new Uri(, ))` запускает соответствующую страницу с новым адресом из списка возможных страниц (Page2.xaml-Page4.xaml).

После запуска одной из лабораторных работ, управление должно передаваться этой странице (рисунок 2.9).

Лабораторная работа №1

Линейные и разветвляющиеся алгоритмы

Задание 1

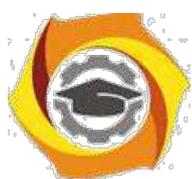
Данные:  
A = 1.5 B = 2 C = 2.5 X =

Ответ:  
 $Z = (X^2 + |B*x - 3*C|)/(Log(x^3)+B*C+A) =$

Задание 1

Данные:  
A1 =  B1 =  C1 =   
A2 =  B2 =  C2 =

Ответ:



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Рисунок 2.9 – Страница выбранной лабораторной работы



На этих страницах есть нужные элементы управления, чтобы можно было подставив исходные данные, получить решение задачи из лабораторной работы. Вернуться на стартовую страницу приложения можно по соответствующей кнопке (листинг 2.7).

Листинг 2.7- Код обработчика события кнопки «Вернуться к выбору»

```
private void button1_Click(object sender,
RoutedEventArgs e)
{
 NavigationService.Navigate(new Uri("/Page1.xaml",
UriKind.Relative));
}
```

#### 2.1.2.6 Подготовка контента приложения

Для того, чтобы можно было отображать обучающие материалы в оболочке web - приложения они должны быть представлены в html формате, с которым наиболее простым образом могут работать web приложения. Подготовка контента идет по следующему алгоритму:

12. проставляются необходимые гиперссылки в файлах – исходниках, которые открываются в редакторе MS Word;
13. файл сохраняется в указанной папке (Lex) в формате html.

Текст всей программы представлен в приложении А.

#### 2.1.3 Порядок проведения исследований

Алгоритм выполнения заданий ЛР следующий:

25. выбрать 5 задач по следующему правилу: номер по журналу- первая за-дача; номер каждой последующей задачи определяется прибавлением цифры 3 к номеру первой задачи, который только что вычислили (если достигнуто окончание списка вариантов задач, то перейти в его начало);



27. составить методы (функции) решения всех задач, поместить их в класс Form1 и снабдить пользовательским интерфейсом в виде web – приложения;
28. исследовать запуск приложения через Internet Explorer, разместив приложение на Local host. Сделать выводы о достоинствах и недостатках web – приложений как определенного вида клиент – серверных структур;
29. оформить отчет для всего приложения в целом, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения каждой задачи и представить его на проверку.

#### 2.1.4 Варианты заданий

Найти среднее арифметическое положительных чисел, введенных с клавиатуры. Всего ввести N различных чисел.

Ввести с клавиатуры N чисел. Найти сумму тех из них, которые принадлежат интервалу (2;9).

Для N введенных с клавиатуры чисел найти сумму положительных кратных трем.

Для арифметической прогрессии 4, 9, 14, 19... найти первые n членов этой прогрессии.

Найти сумму отрицательных значений функции  $Z = \sin(5-x)/\cos(x-2)$  для x, изменяющегося на отрезке [-5,12] с шагом 1.

Найти среднее арифметическое отрицательных чисел, введенных с клавиатуры. Всего ввести N различных чисел.

Найти среднее арифметическое чисел, принадлежащих отрезку [2,184], кратных 2 и введенных с клавиатуры. Всего ввести N различных чисел.

Найти сумму значений функции, больших 2  $Z = \sin(1/x) + 5\cos(1/(x-3)) + x$  для x, изменяющегося на отрезке [-3,8] с шагом 1.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

9. Найти  $n$  членов последовательности  $x_1 = x_2 = x_3 = 1; x_k = x_{k-1} + x_{k-3}$ .



25. Вычислить последовательность  $N$  чисел  $A_0 = x$ ,  $A_1 = 2$ ,  $A_k = A_{k-1} - A_{k-2}$ .
26. Для  $x_1 = 0,3$  и  $x_2 = -0,3$  найти  $x_k = k + \sin(x_{k-2})$  для  $k$ , изменяющегося следующим образом:  $k = 3, 4, \dots, 14$ .
27. Составить таблицу перевода дюймов в сантиметры для расстояний от 1 до 13 дюймов с шагом 1.
28. Вывести на печать значения функции, меньшие 2,  $Z = \sin(1/x) + 5\cos(x-3) + x$  для  $x$ , изменяющегося на отрезке  $[-7, 4]$  с шагом 1.
29. Напечатать таблицу значений функции  $Y = \operatorname{tg}(x/b) + x/(b-2)$  для  $x$ , изменяющегося от 0 до 10 с шагом 1 ( $b$  - произвольное число).
30. Вычислить  $N$ -ый член последовательности  $x_k = x_{k-2} - x_{k-1}$ ,  $x_0 = 2,4$   $x_1 = 3,8$ .
31. Составить таблицу перевода суток (от 1 до 7) в часы, минуты, секунды.
32. Вычислить  $N$ -ый член последовательности  $x_k = x_{k-1} + (2/3)x_{k-2} + 1$ ,  $x_1 = -1$ ,  $x_2 = 1,38$ .
33. Напечатать значения функции  $z = 1/(x-2) + 1/(x-5) + \ln(12,8-X)$  для  $x$ , изменяющегося на отрезке  $[-4, 14]$  с шагом 1,23.
34. Вывести на печать отрицательные значения функции  $z = \sin(5-x)/\cos(x-2)$  для  $x$ , изменяющегося на отрезке  $[-6, 13]$  с шагом 1 (учесть область допустимых значений функции).
- 7) Из  $N$  введенных с клавиатуры чисел напечатать кратные 3 и меньшие 58.
- 8) Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(1, 11)$  и являются четными.
- 9) Из  $N$  введенных с клавиатуры чисел напечатать положительные, кратные трем.
- 10) Вывести на печать значения функции  $z = \sin(x/(x-2))$ , находящиеся в интервале  $(-0,4; 0,8)$  для  $x$ , изменяющегося от 8 до -6 с шагом 1.
- 11) Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(2; 9)$ .
- 12) Для геометрической прогрессии 2, 6, 18, 54, 162 ... определить первые  $n$  членов этой прогрессии.



- С Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые не принадлежат интервалу  $(1;5)$ .
27. Найти  $n$  членов последовательности  $x_1 = x_2 = x_3 = 1; x_k = x_{k-1} - 2x_{k-3}$ .  
Вычислить последовательность  $N$  чисел  $A_0 = x, A_1 = 2, A_k = A_{k-1} + A_{k-2}$   
Составить таблицу перевода килограммов (от 1 до 13) в граммы с шагом
7. Найти сумму значений функции  $Y = \cos(x/A) + x/(A-2)$  для  $x$ , изменяющегося от 2 до 13 с шагом 1 ( $A$  - произвольное число).



## МУкЛР2

### Исследование простых двухуровневых клиент-серверных приложений в Java

#### 2.2.1 Цели занятия

Выработать умения и навыки по составлению программ простых одноуровневых клиент-серверных приложений на основе технологии сокетов.

#### 2.2.2 Теоретические основы и пример выполнения

##### 2.2.2.1 Создание сетевых приложений с использованием TCP

Java поддерживает классы и методы, которые позволяют устанавливать соединение с удаленным компьютером, используя протокол TCP. В отличие от UDP, TCP является протоколом, ориентированным на установление соединения, которое гарантирует надежную связь между приложениями клиента и сервера. Взаимодействие с использованием протокола TCP, начинается после установления соединения между сокетами клиента и сервера. Сокет сервера "слушает" запросы на установление соединения, отправленные сокетами клиентов, и устанавливает соединение. После установления соединения между приложениями клиента и сервера, они могут взаимодействовать друг с другом.

Java упрощает сетевое программирование, путем инкапсуляции функциональности соединения сокета TCP в классы сокета, в которых класс Socket предназначен для создания сокета клиента, а класс ServerSocket для создания сокета сервера.

##### 2.2.2.2 Идентификация методов классов Socket и ServerSocket

Socket является базовым классом, поддерживающим протокол TCP. Класс Socket обеспечивает методы для потокового ввода/вывода, облегчает выполнение операций чтения и записи в сокет и является обязательным для программ, выполняющих сетевое взаимодействие.



Для создания объектов класса Socket используются следующие конструкторы, определенные в классе Socket:

13. `public Socket (InetAddress IP_address, int port)` - создает объект Socket,  
который соединяется хостом, заданным в параметрах `IP_address` и `port`;
14. `public Socket (String hostname, int port)` – создает объект Socket, который соединяется с хостом, заданным параметрами имя хоста или IP адрес и `port`, который сервер "слушает".

`ServerSocket` представляет собой класс, используемый программами сервера для прослушивания запросов клиентов. `ServerSocket` реально не выполняет сервис, но создает объект `Socket` от имени клиента, через который выполняется взаимодействие с сокетом клиента.

Для создания и инициализации объектов `ServerSocket` используются следующие конструкторы, определенные в классе `ServerSocket`:

7. `public ServerSocket(int port_number)`- создает сокет сервера на заданный порт на локальной машине. Клиентам следует использовать этот порт, чтобы общаться с сервером. Если номер порта 0, то сокет сервера создается на любой свободный порт локальной машины;
8. `public ServerSocket(int port, int backlog)` - создает сокет сервера на заданный порт на локальной машине. Второй параметр задает максимальное количество соединений клиентов, которые сокет сервера поддерживает на заданном порту;
9. `public ServerSocket(int port, int backlog, InetAddress bindAddr)` - создает сокет сервера на заданный порт. Третий параметр используется для создания сокета сервера хоста, подключенного к нескольким физическим



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

линиям (multi-homed host). Сокет сервера принимает запросы клиента только с заданных IP адресов.

**2.2.2.3 Создание северной части по протоколу TCP/IP в Java**



Для создания серверного приложения сокета TCP, необходимо выполнить следующие шаги:

9. создать объект сокета сервера `ServerSocket` (класс `Server`);
10. прослушивать запросы клиента на соединение;
11. запустить сервер;
12. создать поток соединения для запросов клиентов.

#### 2.2.2.4 Разработка класса `Server`

Класс `Server` наследует класс `Thread` и, таким образом, поддерживает многопоточное выполнение. Объект `ServerSocket` "слушает" запросы клиентов и конструктор класса `Server` создает объект `ServerSocket`. Сообщение об ошибке отображается, если возникает исключительная ситуация при запуске сервера.

Фрагмент кода для конструктора сервера показан в листинге 2.8.

#### Листинг 2.8- Фрагмент конструктора класса `Server`

```
public Server()
{
 try
 {
 serverSocket = new ServerSocket(1001);
 }
 catch(IOException e)
 {
 fail(e, "Невозможно запустить сервер.");
 }
 System.out.println("Сервер запущен. . .");
 this.start(); // Запускается поток
}
```

В приведенном фрагменте кода используется общий метод обработки ошибок `fail()`, который обеспечивает обработку всех исключительных ситуаций. Метод принимает два аргумента (объект `Exception` и объект `String`) и выводит сообщение об ошибке.



Фрагмент кода для метода `fail()` выглядит следующим образом (листинг 2.9):

Листинг 2.9- Формирование сообщение об ошибке

```
public static void fail(Exception e, String str)
{
 System.out.println(str + "." + e);
}
```

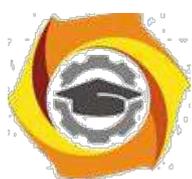
#### 2.2.2.5 «Прослушивание» запросов клиентов

Метод `run()` класса `Server`, как любой поток, который реализует интерфейс `Runnable`, содержит инструкции для потока. В этом случае сервер переходит в бесконечный цикл и прослушивает запросы клиентов. Когда сервер обнаруживает подключение клиента, метод `accept()` класса `ServerSocket` выполняет соединение. При этом сервер создает объект класса `Connection` для клиента. Объект класса `Socket` передается конструктору класса `Connection` и взаимодействие между клиентом и сервером выполняется через этот сокет.

Фрагмент кода для метода `run()` выглядит показан на листинге 2.10.

Листинг 2.10- Метод для прослушивания запросов клиентов

```
public void run()
{
 try
 {
 while(true)
 {
 Socket client = serverSocket.accept();
 Connection con = new Connection(client);
 }
 }
 catch(IOException e)
 {
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*fail(e, "Не прослушивается");*



```
}
}
```

#### 2.2.2.6 Запуск сервера

Фрагмент кода для метода `main()` приведен в листинге 2.11.

Листинг 2.11- Запуск сервера

```
public static void main(String args[])
{
 new Server();
}
```

16.     этом фрагменте кода создается объект класса `Server`, который запускает

поток.

#### 2.2.2.7 Создание потокового соединения

Класс `Connection` нужен для потокового соединения с клиентом (листинг 2.12).

Листинг 2.12- Класс для создания потокового соединения

```
class Connection extends Thread
{
 protected Socket netClient;
 protected BufferedReader fromClient;
 protected PrintStream toClient;
 public Connection(Socket client) {

 netClient = client;
 try
 {
 fromClient = new BufferedReader(new
InputStreamReader(netClient.getInputStream())); toClient =
new PrintStream(netClient.getOutputStream());
 }
 }
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
catch(IOException e)
{
```



```
try
{
netClient.close();
}
catch(IOException e1)
{
System.err.println("Unable to set up streams"
15. e1);
return;
}
}
this.start();
}
public void run()
{
String clientMessage;
try
{
for(;;)
{
clientMessage = fromClient.readLine();
if(clientMessage == null)
break;
5. Посылает подтверждение клиенту
toClient.println("Received");
}
}
catch(IOException e)
{}
finally
{
try
{
netClient.close();
```



}

**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



```
 catch(IOException e)
 {}
 }
}
```

В представленном фрагменте кода класс `Connection` создает объект `fromClient` класса `BufferedReader`, который получает ввод от клиента, используя метод `getInputStream()`. Объект класса `PrintStream` (`toClient`) дает возможность серверу отправлять клиенту данные, используя метод `getOutputStream()`. Таким образом, возникают необходимые (прием и передача) возможности взаимодействия.

Когда клиент соединяется с сервером, сервер использует метод `readLine()` объекта `fromClient`, чтобы запомнить сообщение, посланное клиентом в переменной `clientMessage` типа `String`. Метод `println()` используется для вывода сообщения “Received” сокету.

Для выхода из системы сервер завершает цикл. При этом выполняется блок `finally` для закрытия сокета клиента. Закрытие сокета является важным действием, поскольку сохранение активного соединения неизбежно приводит к потере памяти сервера. Блок `finally` обеспечивает закрытие ранее установленного соединения. Следует отметить, что сервер является многопоточным, и каждый клиент получает свой собственный поток на сервере.

### 2.2.2.8 Пример создания класса `Server`

В листинге 2.13 показан создания класса `Server`, который принимает запросы соединения клиента и посылает строку `Login`, как ответное сообщение клиенту.

#### Листинг 2.13- Пример кода класса `Server`

```
import java.io.*;
import java.net.*;
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
public class Server extends Thread {
```



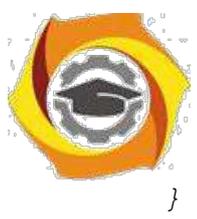
```
ServerSocket serverSocket; // Определяется переменная
server-Socket

public Server() {
 try {
 /*
 6.Создание объекта ServerSocket, который принимает запросы
 7.соединения от клиентов через порт 1001
 */
 serverSocket = new ServerSocket(1001);
 System.out.println(serverSocket.toString());
 } catch (IOException e) {
 fail(e, "Could not start
 server."); }
 System.out.println("Сервер запущен . . .");
 /* Стартует поток */
 this.start();
 }

 public static void fail(Exception e, String str)
 { System.out.println(str + "." + e); }

 public void run() {
 try {
 while (true) {
 /* Принимаются запросы от клиентов */
 Socket client = serverSocket.accept();
 /*
 6. Создается объект соединение, чтобы
 управлять взаимодействием
 и клиента с сервером
 */

 }
 } catch (IOException e) {
 fail(e, "Not listening");
 }
 }
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



```
}

public static void main(String args[])
{ /* Запускается сервер */
new Server();
}

class Connection extends Thread {
/* Declare the variables */
protected Socket netClient;
protected BufferedReader
fromClient; protected PrintStream
toClient; public Connection(Socket
client) { netClient = client; try {

/* Создается входной поток, чтобы принимать данные от
клиента */
fromClient = new BufferedReader(new
InputStreamReader(netClient.getInputStream()));

/* Создается выходной поток, чтобы посылать
данные клиенту */
toClient = new
PrintStream(netClient.getOutputStream());
} catch (IOException e)
{ try {
/* Закрывается сокет клиента */
netClient.close();
} catch (IOException e1) {
System.err.println("Unable to set up streams" +
e1);
return;
}
}

/* Start the thread */
this.start();
```



```
}
public void run() {
 String login, password;
 try {
 while (true) {
 toClient.println("Login: ");
 /* Принимается login как ввод от клиента */
 login = fromClient.readLine();
 /* Завершить соединение, когда Bye вводится как login */
 if (login == null || login.equals("Bye")) {
 System.out.println("Exit");
 return;
 } else
 System.out.println(login + "
logged in");
 В Посылается подтверждение клиенту
 toClient.println("Password: ");
 /* Принимается пароль то клиента
 */ password =
 fromClient.readLine(); }
 } catch (IOException e)
 { } finally {
 try {
 netClient.close();
 } catch (IOException e) {
 }
 }
 }
}
```

#### 2.2.2.9 Создание клиента ТСР/IP



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Первый шаг выполнения клиента состоит в установлении соединения между клиентом и сервером. Для установления соединения между клиентом и



сервером необходимо создать объект Socket. Для создания приложения клиента сокета TCP необходимо выполнить следующие задачи:

- В создать сокет клиента, используя объект Socket;
- В читать и писать в сокет;
- В закрыть соединение.

#### 2.2.2.10 Создание сокета клиента

Объект сокета клиента создается с помощью конструктора класса Socket, принимающего два параметра, IP адрес и номер порта, которые прослушиваются сервером. Следующий фрагмент кода используется для создания сокета клиента (листинг 2.14).

#### Листинг 2.14- Создание сокета клиента

```
try
{
 Socket clientSocket = new Socket("127.0.0.1", 1001);
}
catch (UnknownHostException e)
{
 System.err.println("Неопределенное имя хоста ");
 System.exit(1);
}
```

В предыдущем фрагменте кода IP адрес равный '127.0.0.1' и порт равный '1001' определяют сокет, на котором сервер прослушивает запросы клиента.

#### 2.2.2.11 Чтение и запись в сокет

После установления соединения между клиентом и сервером, клиент посылает запрос на сервер через сокет. Чтение и запись в сокет аналогичны чтению из файла и записи в файл. Чтобы обеспечить возможность клиенту общаться с сервером, необходимо выполнить следующие действия:



- объявляются два объекта по одному для классов `PrintStream` и `BufferedReader`. Эти объекты будут использоваться для чтения и записи в сокет `socket`. `PrintStream out = null; // Объект для записи в сокет`  
`BufferedReader in = null; // Объект для чтения из сокета;`
- объекты `PrintStream` и `BufferedReader` связываются с сокетом. `out = new PrintStream(clientSocket.getOutputStream()); in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));`
- методы `getInputStream()` и `getOutputStream()` класса `Socket` позволяют клиенту взаимодействовать с сервером. Метод `getInputStream()` позволяет объекту `BufferedReader` читать из сокета, а метод `getOutputStream()` позволяет объекту `PrintStream` писать в сокет.

Объявляется еще один объект класса `BufferedReader` для связи со стандартным входом, чтобы данные, введенные в приложении клиенте, могли передаваться на сервер. Следующий фрагмент кода используется для чтения данных из окна консоли (листинг 2.15).

#### Листинг 2.15- Чтение данных из окна консоли

```
BufferedReader stdin = new BufferedReader(new
InputStreamReader(System.in));
String str;
while((str = stdin.readLine()).length() != 0)
{
out.println(str);
}
```

Представленный фрагмент кода позволяет пользователю вводить данные с клавиатуры. Цикл `while` продолжается до тех пор, пока пользователь не введет символ завершения ввода (`Ctrl-Z`).

#### 2.2.2.12 Закрытие соединения

Операторы, представленные ниже, закрывают потоки и соединения с сервером: `out.close(); in.close(); stdin.close()`.



Сокет клиента принимает имя пользователя и пароль, и обеспечивает связь. Для завершения соединения, пользователь должен ввести 'Bye'. Можно использовать следующий код для создания класса Client (листинг 2.16).

#### Листинг 2.16- Пример создания класса Client

```
import java.net.*;
import java.io.*;
public class Client {
public static void main(String[] args) throws IOException {
Socket clientSocket;
PrintStream out = null;
BufferedReader in = null;
try {
/* Создается объект сокет, чтобы соединиться с
сервером */
clientSocket = new Socket("127.0.0.1", 1001);
/* Создается выходной поток, чтобы посылать данные на
сервер */
out = new
PrintStream(clientSocket.getOutputStream());
/* Создается входной поток, чтобы принимать данные с
сервера */
in = new BufferedReader(new InputStreamReader(
clientSocket.getInputStream()));
} catch (UnknownHostException e) {
System.err.println("Unidentified hostname
"); System.exit(1);
} catch (IOException e) {
System.err.println("Couldn't get I/O
"); System.exit(1);
}
/* Создается входной поток, чтобы читать данные из окна консо-
```



ЛИ

**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



```
*/
BufferedReader stdin = new BufferedReader(new
InputStreamReader(
 (System.in)));
/* Чтение из сокета */
String login = in.readLine();
System.out.println(login);
/* Прием login */
String logName = stdin.readLine();
out.println(logName);
/* Чтение из сокета */
String password = in.readLine();
System.out.println(password);
/* Прием password */
String pass = stdin.readLine();
out.println(pass);
String str = in.readLine();
System.out.println(str);
while ((str = stdin.readLine()) != null) {
 out.println(str);
 if (str.equals("Bye"))
 break;
}
out.close();
in.close();
stdin.close();
}
}
```

Представленный выше код сохраняется как Client.java. При выполнении класса Client, отображается сообщение для ввода Login. После приема Login на консоли класса Client появляется сообщение для ввода password. Рис. 7 показы-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

вает вывод класса Client, который посылает login XXXX и password для входа на Server. Введенный login в окне консоли класса Client передается классу



Server. Когда на сервер передается строка « Bye » в качестве login от клиента, соединение между клиентом и сервером прекращается.

### 2.2.3 Порядок проведения исследований

Алгоритм выполнения задач практического занятия следующий:

13. выбрать 5 задач по следующему правилу: номер по журналу- первая за-дача; номер каждой последующей задачи определяется прибавлением цифры 3 к номеру первой задачи, который только что вычислили (если достигнуто окончание списка вариантов задач, то перейти в его начало);
14. составить классы Swrver и Client и установить соединение между ними;
15. составить методы (функции) решения всех задач, поместить их в класс Server. Все исходные данные вводить в консоли клиента, передавать их на сервер, а полученные результаты решений задач выводить на экран в консоли клиента;
16. оформить отчет для всего приложения в целом, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения каждой задачи и представить его на провер-ку.

### 2.2.4 Варианты заданий

В Составить программу для перевода длины в метрах в длину в санти-метрах, определив функцию, выполняющую это преобразование и передав длину в метрах в качестве параметра.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

/ Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив функцию, выполняющую это действие, и передавая массивы в качестве параметра.



в Даны числа  $S, T$ . Получить с использованием функции пользователя  $F(T, -2S, 1.17) + F(2.2, T, S - T)$  где  $F(A, B, C) = (2A - B - \sin(C)) / (5 + C)$

13) Составить программу перевода двоичной записи натурального числа в десятичную, описав соответствующую функцию с параметром. Перевод осуществлять для чисел, вводимых с клавиатуры. Признак конца ввода - число 0. 5.

Даны числа  $S, T$ . Получить с использованием функции пользователя с параметрами:

$$G(1, \sin(S)) + 2G(T * S, 24) - G(5, -S), \text{ где } G(A, B) = (2A + B * B) / (A * B * 2 + B * 5).$$

10) Составить программу для расчета значений гипотенузы треугольника, определив функцию, выполняющую этот расчет. Катеты передаются в качестве параметров.

15. Найти периметр десятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, которые передаются функции в качестве параметров из основной программы.

18. Найти периметр шестиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами. Координаты передаются функции в качестве параметров из основной программы.

14) Найти площадь пятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, и процедуру вычисления площади треугольника по трем сторонам. Описать функции с соответствующими формальными параметрами.

- Составить программу вывода на экран всех натуральных чисел, не превосходящих  $N$  и делящихся на каждую из своих цифр. Описать соответствующую функцию, получающую из основной программы в качестве параметра натуральное число и возвращающую TRUE, если оно удовлетворяет указанному условию.



13. Используя подпрограмму - функцию, составить программу для нахождения максимального из трех чисел. Числа передаются функции в качестве параметров.

7 Используя подпрограмму - функцию, составить программу для печати знаков трех чисел, введенных с клавиатуры и передаваемых функции в качестве параметра.

13. Используя подпрограмму - функцию, составить программу для возведения чисел в целую положительную степень. Число передается функции в качестве параметра из основной программы. Расчет вести для чисел, пока не будет введено число, равное 0.

25. Используя подпрограмму - функцию, составить программу для вычисления функции  $Z=(X1+Y1)/(X1*Y1)$ , где  $X1$  - первый корень уравнения  $X^2-4*X-1=0$ ;  $Y1$  - первый корень уравнения  $2*Y^2 + A*Y - A^2 = 0$  ( $A$  - произвольное).

В Задав функцию, вывести на печать средние арифметические двух массивов, введенных с клавиатуры. Массив передается функции в качестве параметра.

13. Задав функцию, рассчитать и вывести на печать максимальные значения в трех парах чисел, вводимых с клавиатуры. Пара чисел передается функции в качестве параметра.

- Найти периметр восьмиугольника, координаты вершин которого заданы. Определить функцию вычисления расстояния между двумя точками, заданными своими координатами. Координаты передать функции в качестве параметров.

12. Даны четыре пары чисел. Получить с использованием функции пользователя наибольший общий делитель для каждой пары.

11) Даны числа  $A, B, C$ . Получить с использованием функции пользователя наименьшее значение. Числа передаются функции из основной программы в качестве параметров.



14) Даны числа  $x = 1, 2, \dots, N$ . Получить с использованием функции пользователя значения  $3 * P(X+3) * P(X)$  для заданных  $x$ , где  $P(X) = 10 * X^3 - 14 * X^2 + 12 * X - 2$ .

В Составить программу для расчета значений катета треугольника, определив функцию, выполняющую этот расчет. Гипотенуза и второй катет передаются в качестве параметров.

16. Даны целые числа  $a, b, c, d$ . Проверить с использованием функции пользователя их четность. Число для проверки передается в функцию в качестве параметра из основной программы.

и Для каждого из 10 введенных с клавиатуры чисел напечатать сообщение: является ли оно простым или нет, описав функцию логического типа, возвращающую значение “ИСТИНА”, если число, переданное ей в качестве параметра, является простым.

7. Даны числа  $S, T$ . Получить с использованием функции пользователя  $Y(T, S) = G(12, S) + G(T, S) - G(2S - 1, S * T)$ , где  $G(A, B) = (2 * A + B * B) / (A * B * 2 + B * 5)$ .

10. Определите функцию, определяющую, какой целой степенью числа 2 является ее аргумент (если число не является степенью двойки - выдать соответствующее сообщение).

14. Определите функцию, подсчитывающую сумму  $N$  первых элементов целочисленного массива  $A$ .  $N$  и массив  $A$  передать в качестве параметров.

15. Вычислить количество простых чисел, не превосходящих заданного  $N$ . Описать функцию логического типа, возвращающую значение true, если число простое и false в противном случае.

11. Используя подпрограмму - функцию с параметрами, составить программу для вычисления функции  $F(X, Y) = (2X^3 - 4 * X^2 + X + 1) / (9 * Y^3 + Y + 4) + 3 * Y^2 + 5 * Y$ .

12. Составить программу для перевода веса в граммах в вес в килограммах, определив функцию, выполняющую это преобразование. Вес в граммах передается функции в качестве параметра.



6. Даны числа  $S, T$ . Получить с использованием функции пользователя  $G(12, S)+G(T, S)-G(2S-1, S*T)$  где  $G(A, B) = (2*A+B*B)/(A*B*2+B*5)$ .



## МУкЛРЗ

### Исследование многоуровневых клиент-серверных приложений в Java

#### 2.3.1 Цели занятия

Развить навыки составления клиент-серверных приложений с основной обработкой информации на стороне клиента путем исследования достоинств и недостатков такого подхода.

#### 2.3.2 Теоретические основы и пример выполнения

Основным отличием разработки такой разновидности приложения является то, что вся обработка информации, еще говорят: бизнес – логика приложения выполняется на стороне клиента.

Есть преимущества и недостатки такого подхода к созданию клиент-серверных приложений.

Достоинства в том, что сервер является очень простым и служит для организации и хранения данных, а на стороне клиента происходит вся основная обработка информации.

Недостатки – это сложная структура клиентской части, реализация которой предъявляет жесткие требования к ресурсам устройств, на которых она располагается.

9. точки зрения технологии разработки ничего не меняется по сравнению с созданием классических клиент – серверных разработок, которые были рассмотрены в предыдущей лабораторной работе. Поэтому, необходимо взять за основу алгоритм решения задачи, показанный выше и разработать клиент – серверное приложение по выбранному варианту с той только разницей, что решение логических заданий теперь необходимо представить на стороне клиентской части приложения. Другими словами, все методы, которые нужно будет составить для решения указанных заданий, должны быть размещены в одном из классов клиентской части приложения, а исходные данные – в классе сервера.



### 2.3.3 Порядок проведения исследований

Алгоритм выполнения заданий лабораторной работы следующий:

- > выбрать 5 задач по следующему правилу: номер по журналу- первая задача; номер каждой последующей задачи определяется прибавлением цифры 3 к номеру первой задачи, который только что вычислили (если достигнуто окончание списка вариантов задач, то перейти в его начало);
- > составить классы `Server` и `Client` и установить соединение между ними;
- > составить методы (функции) решения всех задач, поместить их в класс `Server`. Все исходные данные вводить в консоли клиента, передавать их на сервер, а полученные результаты решений задач выводить на экран в консоли клиента;
- > оформить отчет для всего приложения в целом, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и `skrin-shert` результата выполнения каждой задачи и представить его на проверку.

### 2.3.4 Варианты заданий

8. **НАИБОЛЬШИЙ ПРЯМОУГОЛЬНИК.** Дана матрица, состоящая из нулей и единиц. Найти наибольший по площади прямоугольник, состоящий из одних единиц.
9. **ПЕРИОД ДРОБИ.** Найти период десятичной дроби, равной  $M/N$ , где  $M$  и  $N$  - натуральные числа.
10. **НЕСОСТАВЛЯЕМОЕ ЧИСЛО.** Задан массив натуральных чисел. Найти минимальное натуральное число, не представимое суммой чисел из массива. Каждый элемент массива может входить в сумму не более одного раза.



- > **СООТВЕТСТВИЕ ШАБЛОНУ.** Установить соответствие имени файла заданному шаблону. Шаблон называется строка, в которой “,” означает любой символ, а “\*” означает любую последовательность символов, в том числе пустую.
- > **КОД ГРЕЯ.** Построить N -разрядный код Грея. Кодом Грея называется такая последовательность N -разрядных двоичных чисел, в которой два соседних, а также первое и последнее число отличаются одним разрядом. Например, для N=2 один из кодов Грея таков: 00, 01, 11, 10.
- > **ОБМЕН ЧАСТЕЙ МАССИВА.** Задан одномерный массив и граница между двумя его частями. Поменять порядок расположения частей массива, не делая лишних присваивания. Например: 1 2 3 4 5 6 7 → 5 6 7 1 2 3 4.
- > **ЛЕВЫЕ ПОВОРОТЫ.** Ломаная задана координатами своих узлов. Сосчитать количество левых поворотов при движении вдоль ломаной.
- > **ПЯТНА НА ШКУРЕ.** Подсчитать количество черных пятен на белой шкуре. Шкуру представить в виде 0,1 - матрицы, где 0 - белый, а 1 - черный цвет.
- > **ТОЖДЕСТВО КОМПОСТЕРОВ.** Имеется два автобусных талона. Установить, пробиты ли они одним компостером или разными. Талон можно вкладывать в компостер любой стороной так, чтобы края талона были параллельны краям компостера и все отверстия компостера помещались на талоне. Отверстия компостера размещаются на матрице размером M N.
- > **ПОДСЧЕТ СЛОВ.** Написать программу подсчета количества слов в файле с русским текстом, которые начинаются с заданного буквосочетания. Учесть возможность переноса слов.
- 7. **ВЫРАВНИВАНИЕ ТЕКСТА.** В ASCII файле находится русский текст. Выровнять его по левой и правой границам, добиваясь наиболее плотного и равномерного распределения слов в строках.
- 10. **ТОЛКОВЫЙ СЛОВАРЬ.** В толковом словаре одни слова определяются через другие так, что нигде нет порочного круга. Описать процедуру, ко-



торая пронумеровывает слова таким образом, чтобы слова с большим номером определялись через слова с меньшими номерами.

**С ЛЕВЫЕ ПОВОРОТЫ.** Ломаная задана координатами своих узлов. Описать функцию подсчета количества левых поворотов при движении вдоль ломаной, а также процедуру вывода изображения ломаной на экран.

**к) МАТРИЦА ОТНОШЕНИЙ.** Задана матрица вида

$$\begin{array}{cccccc} X_1 & X_2 & X_3 & \dots & X_N \\ Y_1 & R_{11} & R_{12} & \dots & R_{1N} \\ Y_2 & R_{21} & R_{22} & \dots & R_{2N} \\ \dots & & & & \\ Y_M & R_{M1} & R_{M2} & \dots & R_{MN}, \end{array}$$

где  $X_i$  и  $Y_j$  целые числа, а  $R_{ij}$  - арифметические отношения:  $<$ ,  $=$ ,  $>$ . Найти значения  $X_i$  и  $Y_j$ , которые удовлетворяли бы отношениям  $R_{ij}$ .

**С НЕУБЫВАЮЩАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ.** В программе описать процедуру, печатающую самую длинную неубывающую последовательность в последовательности целых чисел, избежав при этом полного перебора.

**16) БОЙ ШАШЕК.** На доске стоят белые шашки и одна черная. Описать процедуру, определяющую, какие шашки может побить черная за один ход.

**17) КРАТЧАЙШИЙ ПУТЬ КОНЯ.** Имеется бесконечная шахматная доска. В программе описать функцию, подсчитывающую, за какое наименьшее число ходов конь сможет перейти с поля  $(x_1, y_1)$  на поле  $(x_2, y_2)$ ?

**18) ВЗАИМНЫЙ ЗАЧЕТ ДОЛГОВ.** Есть множество предприятий, которые должны друг другу. Произвести взаимный зачет долгов на максимальную сумму.



- 11) САМОРОДКИ. Есть множество золотых самородков известного веса. Описать процедуру, которая позволяет разделить самородки на две кучи, наиболее близкие по весу.



## МУк ЛР4

### Исследование серверной части WEB приложений

#### 2.4.1 Цели занятия

Исследовать серверную часть web – приложения, размещенного в сети Интернет. Изучить особенности работы подобных приложений.

#### 2.4.2 Теоретические основы и пример выполнения Опишем алгоритм составления типового web – приложения.

- 19.Создание проекта типа web – приложение: WpfBrowserApplication1 в Visual Studio C#.
- 20.Составление необходимого интерфейса, используя библиотеку управляющих элементов.
- 21.Отладка приложения на Local host.
- 22.Размещение приложения на компьютере, подключенного к Internet и доступ к нему по указанному URL адресу.

##### 2.4.2.1 Пример разработки web – приложения

Составим приложение для сортировки двумерного массива случайных простых чисел методом вставки.

Приложение должно задавать двумерный массив размерностью  $m \times n$  элементов, заполнять его случайными простыми числами, затем переписывать его построчно в одномерный массив, сортировать его указанным методом по возрастанию элементов и снова переписывать построчно в двумерный массив.

Реализуем алгоритм, указанный в пункте 2.2.2.

Пункты 1 и 2 дают следующий проект в среде разработки (рисунок 2.10).

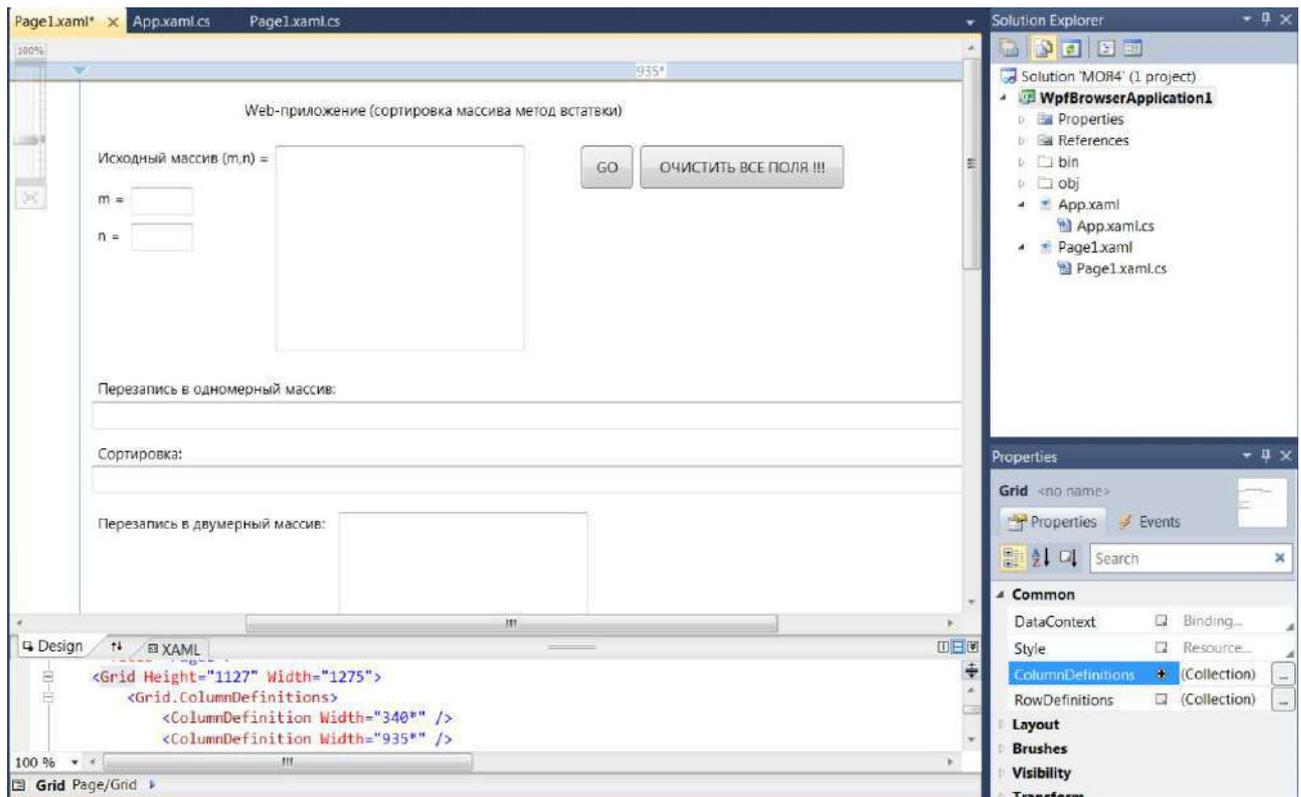


Рисунок 2.10- Вид проекта и интерфейс приложения

Отладка приложения (пункт 3 алгоритма, приведенного в 2.2.2) заключается в его корректном запуске и устранении ошибок.

Так, перед тем, как запускать проект нужно во вкладке Свойства проект Visual Studio и конкретно в свойстве Безопасность дать полное «доверие» проекту (рисунок 2.11).

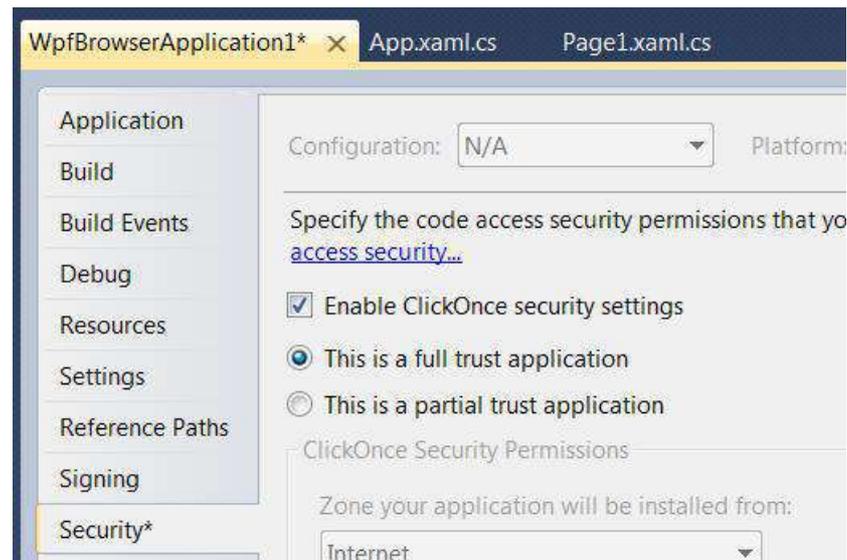
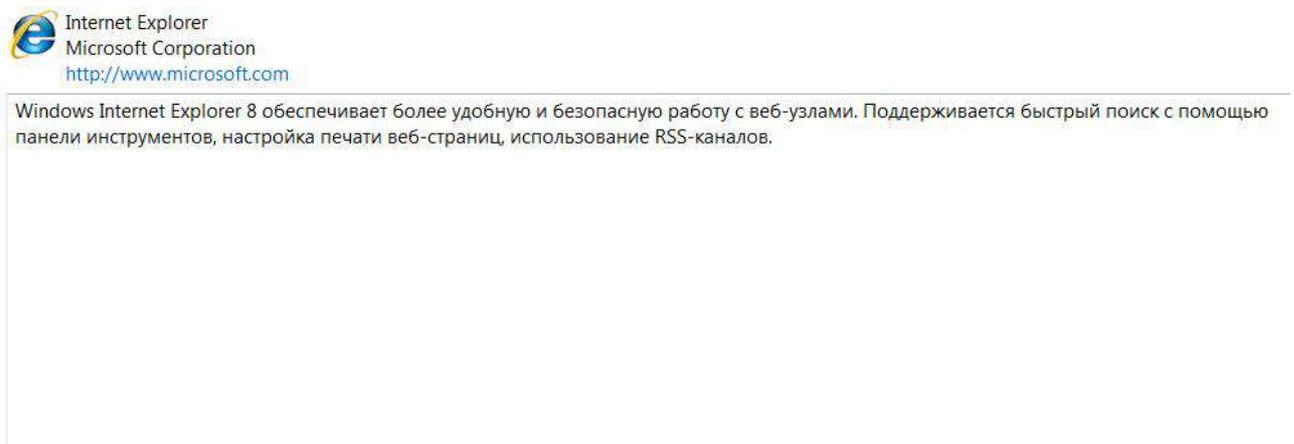


Рисунок 2.11- Открытие полных прав для проекта

Далее, необходимо установить Internet Explorer браузером по умолчанию, для этого в Пуске и в Программах по умолчанию нужно выбрать эту программу (рисунок 2.12).



Для этой программы используются все умолчания

- **Использовать эту программу по умолчанию**  
Использовать выбранную программу для открытия всех типов файлов и протоколов, которые она может открывать по умолчанию.

Рисунок 2.12- Настройка браузера по умолчанию

Далее, запускаем приложение и используем его (рисунок 2.13).

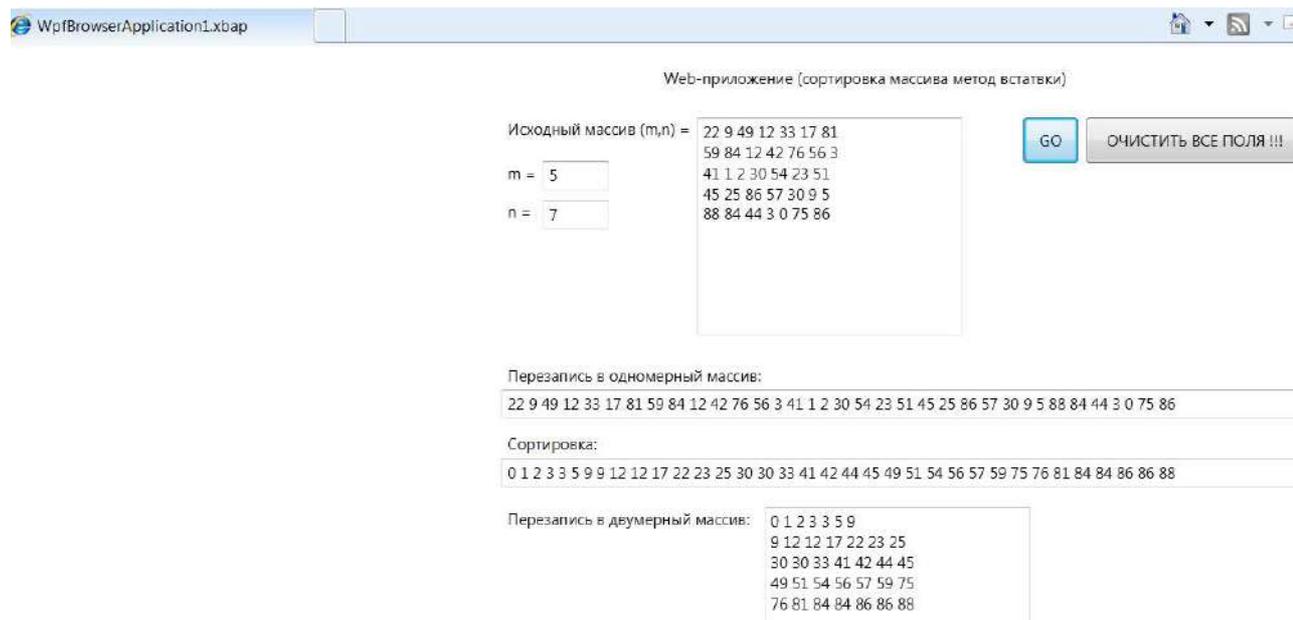
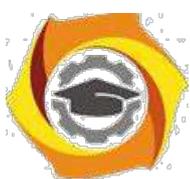


Рисунок 2.13- Проверка работы web – приложения

Для реализации 4-го пункта алгоритма из 2.2.2 запустим удаленно через сеть Интернет разработанное приложение. Для этого необходимо его разместить на зарегистрированном ресурсе и указать его URL адрес.

Протокол TCP запрашивает два элемента данных: IP адрес и номер порта. URL - это стандартизированный способ записи адреса ресурса в сети Интернет. Интернет протокол (Internet Protocol - IP) обеспечивает логический адрес, называемый IP-адресом сетевого устройства. Каждому доменному имени компьютера в системе доменных имен (DNS - Domain Name System) соответствует IP-адрес. IP-адреса, используемые в Интернет, имеют определенный формат. Каждый адрес представляется 32-разрядным числом (т.е. 4-мя байтами), состоящим из четырех 8-разрядных чисел (байт), каждое в диапазоне значений от 0 до 255. Поэтому, для зарегистрированного в сети Интернет ресурса, например, <http://www.skf-mtusi.ru/> имеется свой IP адрес. Следовательно, чтобы удаленно запустить разработанное приложение нужно указать, например, <http://www.dm.ru/WpfBrowserApplication1.exe>. Очевидно, что указанный домен dm.ru должен быть зарегистрирован в сети Интернет.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Текст программы web – приложения показан в листинге 2.17.



Листинг 2.17- Текст программы web – приложения

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfBrowserApplication1
{
 B <summary>
 B Interaction logic for Page1.xaml
 B </summary>
 public partial class Page1 : Page
 {
 public Page1()
 {
 InitializeComponent();
 }
 private void button1_Click(object sender, RoutedEventArgs e)
 {
 met1();
 }
 Random rd1 = new Random();
 void met1()
 {
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
int i = 0, j = 0;
```



```
int n = int.Parse(textBox4.Text);
int m = int.Parse(textBox5.Text);

int[,] X = new int[n, m];
for (i = 0; i < n; i++)
{
 for (j = 0; j < m; j++)
 {
 X[i, j] = rd1.Next(90);
 }
}
for (i = 0; i < n; i++)
{
 for (j = 0; j < m; j++)
 {
 textBox1.Text += X[i, j] + " ";
 }
 textBox1.Text += "\n";
}
int[] Y = new int[n * m];
int index = 0;
for (i = 0; i < n; i++)
{
 for (j = 0; j < m; j++)
 {
 Y[index] = X[i, j];
 index++;
 }
}
for (index = 0; index < n * m; index++)
{
 textBox6.Text += Y[index] + " ";
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
int outt, inn, temp;
```



```
for (outt = 1; outt < n * m; outt++) // out - раз-
делительный маркер
{
 temp = Y[outt]; //Скопировать помеченный эле-
мент
 inn = outt; //Начать перемещения с out
 while(inn>0 && Y[inn-1] >= temp) // Пока не
найден меньший элемент
 {
 Y[inn] = Y[inn-1]; // Сдвинуть элемент
вправо
 --inn; // Перейти на одну позицию влево
 }
 Y[inn] = temp; // Вставить помеченный элемент
}
int r = 0;
for (r = 0; r < n * m; r++)
{
 textBox2.Text += Y[r] + " ";
}
index = 0;
r = 0;
for (i = 0; i < n; i++)
{
 for (j = 0; j < m; j++)
 {
 X[i, j] = Y[r];
 r++;
 }
}
for (i = 0; i < n; i++)
{
 for (j = 0; j < m; j++)
 {
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
textBox3.Text += X[i, j] + " ";
```



```
 }
 textBox3.Text += "\n";
 }
}
private void button1_Click_1(object sender,
RoutedEventArgs e)
{
 met1();
}
private void button2_Click_1(object sender,
RoutedEventArgs e)
{
 textBox1.Text = "";
 textBox2.Text = "";
 textBox3.Text = "";
 textBox4.Text = "";
 textBox5.Text = "";
 textBox6.Text = "";}}}
```

### 2.4.3 Порядок проведения исследований

Алгоритм выполнения заданий ЛР следующий:

- В** выбрать 5 задач по следующему правилу: номер по журналу- первая задача; номер каждой последующей задачи определяется прибавлением цифры 3 к номеру первой задачи, который только что вычислили (если достигнуто окончание списка вариантов задач, то перейти в его начало);
- В** составить методы (функции) решения всех задач, поместить их в класс Form1 и снабдить пользовательским интерфейсом в виде web – приложения;
- В** исследовать запуск приложения через Internet Explorer, разместив приложение на Local host. Сделать выводы о достоинствах и недостатках web – приложений как определенного вида клиент – серверных структур;



и оформить отчет для всего приложения в целом, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения каждой задачи и представить его на проверку.

#### 2.4.4 Варианты заданий

В Найти среднее арифметическое положительных чисел, введенных с клавиатуры. Всего ввести  $N$  различных чисел.

В Ввести с клавиатуры  $N$  чисел. Найти сумму тех из них, которые принадлежат интервалу  $(2;9)$ .

- Для  $N$  введенных с клавиатуры чисел найти сумму положительных кратных 3.

- Для арифметической прогрессии 4, 9, 14, 19... найти первые  $n$  членов этой прогрессии.

- Найти сумму отрицательных значений функции  $Z=\sin(5-x)/\cos(x-2)$  для  $x$ , изменяющегося на отрезке  $[-5,12]$  с шагом 1,23.

В Найти среднее арифметическое отрицательных чисел, введенных с клавиатуры. Всего ввести  $N$  различных чисел.

12. Найти среднее арифметическое чисел, принадлежащих отрезку  $[2,184]$ ,

кратных 2 и введенных с клавиатуры. Всего ввести  $N$  различных чисел.

9. Найти сумму значений функции, больших 2  $Z=\sin(1/x)+5\cos(1/(x-3))+x$  для  $x$ , изменяющегося на отрезке  $[-3,8]$  с шагом 1,123.

11. Найти  $n$  членов последовательности  $x_1 = x_2 = x_3 = 1; x_k = x_{k-1} + x_{k-3}$ .

Вычислить последовательность  $N$  чисел  $A_0 = x, A_1 = 2, A_k = A_{k-1} - A_{k-2}$ .

Для  $x_1 = 0,3$  и  $x_2 = -0,3$  найти  $x_k = k + \sin(x_{k-2})$  для  $k$ , изменяющегося следующим образом:  $k = 3,4,\dots,14$ .

и Составить таблицу перевода дюймов в сантиметры для расстояний от 1 до 13 дюймов с шагом 1.



- Вывести на печать значения функции, меньшие 2,  $Z = \sin(1/x) + 5\cos(x - 3) + x$  для  $x$ , изменяющегося на отрезке  $[-7, 4]$  с шагом 1.
- Напечатать таблицу значений функции  $Y = \operatorname{tg}(x/b) + x/(b-2)$  для  $x$ , изменяющегося от 0 до 10 с шагом 1 ( $b$  - произвольное число).
- Вычислить  $N$ -й член последовательности  $x_k = x_{k-2} - x_{k-1}$ ,  $x_0 = 2,4$ ,  $x_1 = 3,8$ .  
Составить таблицу перевода суток (от 1 до 7) в часы, минуты, секунды.  
Вычислить  $N$ -й член последовательности  $x_k = x_{k-1} + (2/3)x_{k-2} + 1$ ,  $x_1 = -1$ ,  $x_2 = 1,38$ .

Напечатать значения функции  $z = 1/(x-2) + 1/(x-5) + \ln(12,8-X)$  для  $x$ , изменяющегося на отрезке  $[-4, 14]$  с шагом 1,152.

Вывести на печать отрицательные значения функции  $z = \sin(5-x)/\cos(x-)$  для  $x$ , изменяющегося на отрезке  $[-6, 13]$  с шагом 1,541 (учесть область допустимых значений функции).

- Из  $N$  введенных с клавиатуры чисел напечатать кратные 3 и меньшие 58.

С Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(1, 11)$  и являются четными.

С Из  $N$  введенных с клавиатуры чисел напечатать положительные, кратные 3.

- Вывести на печать значения функции  $z = \sin(x/(x-2))$ , находящиеся в интервале  $(-0,4; 0,8)$  для  $x$ , изменяющегося от 8 до -6 с шагом 1,235.

- Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые принадлежат интервалу  $(2; 9)$ .

- Для геометрической прогрессии 2, 6, 18, 54, 162 ... определить первые членов этой прогрессии.

Ввести с клавиатуры  $N$  чисел. Напечатать те из них, которые не принадлежат интервалу  $(1; 5)$ .

- Найти  $n$  членов последовательности  $x_1 = x_2 = x_3 = 1$ ;  $x_k = x_{k-1} - 2x_{k-3}$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Вычислить последовательность  $N$  чисел  $A_0 = x, A_1 = 2, A_k = A_{k-1} + A_{k-2}$



- Составить таблицу перевода килограммов (от 1 до 13) в граммы с шагом  
ГОМ
- Найти сумму значений функции  $Y = \cos(x/A) + x/(A-2)$  для  $x$ , изменяющегося от 2 до 13 с шагом 1 ( $A$  - произвольное число).



## МУкЛР5

### Исследование простых двухуровневых клиент-серверных приложений в C#

#### 2.5.1 Цели занятия

Выработать умения и навыки составлять типовые программы решения задач на выбранном языке программирования, снабженные элементами графического интерфейса пользователя в виде клиент-серверных приложений.

#### 2.5.2 Теоретические основы и пример выполнения

Для создания клиент-серверного приложения в C#, аналогично предыдущей ЛР, необходимо создать два класса Server и Client, используя технологию сокетов.

##### 2.5.2.1 Программирование класса Server

Вначале в этом классе создадим серверное соединение и используем для этого систему сокетов.

Сокеты- это концепция сетевого программирования, когда существуют два вида «программных разъемов»- сокетов: клиентские и серверные.

Серверный сокет как розетка, которая «висит на стене» готовая к работе, в ожидании, когда к ней подключат штекер. Точно так же серверный сокет переходит в режим ожидания подключения на определенном адресе и определенном порту.

Клиентский сокет как вилка, которую втыкают в розетку. Как только клиент подключается к серверному сокету, информация начинает передаваться между ними.

Для того чтобы создать виртуальное подключение между клиентом и сервером, надо знать место, где находится нужный нам серверный сокет.

В нашем случае, создается слушатель событий подключений в сети по протоколу TCP IP – TcpListener, в качестве параметра которому передается так



называемая точка входа, которую определяет метод `EndPoint` из переданных ему IP адреса компьютера, на котором будет запущен сервер (пока это вариант `localhost`, т.е. IP 127.0.0.1) и номер порта `Port`, на котором создаваемой сервер-ное подключение будет прослушивать подключения клиентов. Т.о., в кон-структоре класса `Server` появится следующий слушатель (листинг 2.18).

#### Листинг 2.18- Серверный слушатель подключения клиентов

```
TcpListener listener;

public Program(int Port)
{

 listener = new TcpListener(new End-
Point(IPAddress.Parse("127.0.0.1"), Port));
 listener.Start();
 Console.WriteLine("Сервер запущен, ожидает подклю-
чений...");
 ...}
```

Видно, что после запуска прослушивания сетевых подключений выдается сообщение о том, что сервер готов и ожидает подключений.

Далее, нужно в это подключение с помощью метода `AcceptTcpClient()` и создать новый поток для него с помощью метода `Thread` из одноименной библиотеки, которую нужно подключить заранее: `using System.Threading`.

В качестве параметра этому методу передается метод `clientThread`, в котором для каждого нового клиента открывается TCP соединение в новом потоке (листинг 2.19).

#### Листинг 2.19- Метод создания нового TCP соединения для нового клиента

```
static void clientThread(Object StateInfo)
{
 new Client((TcpClient)StateInfo);
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

В итоге получим следующий текст класса Server (листинг 2.20).



### Листинг 2.20- Класс Server

```
class Server
{
 TcpListener listner;
 public Server (int Port)
 {
 listner = new TcpListener(new IPEndPoint(
IPAddress.Parse("127.0.0.1"), Port));
 listner.Start();
 Console.WriteLine("Сервер запущен, ожидает подклю-
чений...");
 while (true)
 {
 TcpClient client = listner.AcceptTcpClient();
 10. Создаем поток

 Thread Thread = new Thread(new ParameterizedThread-
Start(clientThread));
 // И запускаем этот поток, передавая ему при-
нятого клиента
 Thread.Start(client);
 }
 }

 static void clientThread(Object StateInfo)
 {
 new Client((TcpClient)StateInfo);
 }

 static void Main(string[] args)
 {
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*new Program (12000) ;*



```
}
}
```

Видно, что сюда же включен и главный метод модуля – метод Main, из которого происходит запуск всего приложения и сервер начинает «прослушивание» на порту №12000, который в качестве параметра передается конструктору класса.

### 2.5.2.2 Класс Client

Этот класс необходим для считывания запроса от клиентских приложений, поиска нужного контента согласно запросу и для его передачи в сеть клиентам.

Начнем с задания объектов для методов чтения из потока и записи в поток. Для этого создаются новые экземпляры соответствующих классов (листинг 2.21).

#### Листинг 2.21- Организация потокового чтения и записи

```
StreamReader sr = new StreamReader(client.GetStream()); //
чтение из потока клиента

StreamWriter sw = new StreamWriter
client.GetStream()); // запись в поток клиента

sw.AutoFlush = true;
```

Этот пример клиент-серверного приложения обеспечивает запрос от клиента на сервер необходимого файла информации, передачу его клиенту, сохранение в темповском файле на клиенте и отображение в клиентской части приложения. Поэтому, определим какой файл необходим: прочитаем из потока в запросе клиента значение строковой переменной str, конвертируем его в целочисленную переменную kluch, и по ее значению в структуре switch определим необходимый файл (листинг 2.22).

#### Листинг 2.22- Определение требуемого контента



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*// Принимаем передачу от клиента*



```
string cl = sr.ReadLine(); // Отображение типа
обучающего модуля

Console.WriteLine(cl);

string str = sr.ReadLine();

int kluch = int.Parse(str); // по kluch определяем
нужный файл

string FilePath = "C:/www/"; // составляем путь к
файлу

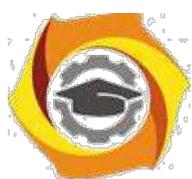
switch (kluch)
{
 case 1:
 {
 FilePath += "L1.htm";

 Console.WriteLine("Путь к файлу: " +
FilePath);
 }

 break;

 case 2:
 {
 FilePath += "L2.htm";

 Console.WriteLine("Путь к файлу: " +
FilePath);
 }
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*break;*



}

Видно, что весь контент преобразован в html формат и находится в соответствующих папках на сервере. Аналогично программируется определение оставшихся файлов обучающих материалов.

Далее нужно открыть найденный файл. Сделаем это, проверяя правильность его открытия с помощью блока try/catch и если возникнет ошибка открытия, выдадим соответствующее сообщение (листинг 2.23).

#### Листинг 2.23- Безопасное открытие файла контента

*16.Открываем файл, страхуясь на случай ошибки*

```
FileStream
FS; try
{
 FS = new FileStream(FilePath, FileMode.Open,
FileAccess.Read, FileShare.Read);
 sw.WriteLine("Запрашиваемый файл открыт и го-
тов к передаче");
}
catch (Exception)
{
 • Если случилась ошибка, посылаем клиенту
ошибку открытия файла
 sw.WriteLine("Ошибка открытия файла");
 return;
}
```

Учитывая, что данные передаются в сети в формате байт, нужно зарезервировать буфер для чтения материалов из открытого файла. Затем, перенести



эти данные в буфер и через него передать их клиенту в ответе на запрос. Весь этот процесс будет выполняться до конца данных в открытом файле (листинг 2.24).

#### Листинг 2.24- Передача контента в ответном сообщении клиенту

и *Задаем буфер для размещения данных из выбранного файла*

```
byte[] Buffer = new byte[1024];
```

о *Переменная для хранения количества байт, передаваемых клиенту*

```
int Count;
```

В *Пока не достигнут конец файла*

```
while (FS.Position < FS.Length)
```

```
{
```

В *Читаем данные из файла и пишем их в буфер*

с *И передаем их клиенту*

```
client.GetStream().Write(Buffer, 0, Count);
```

```
}
```

```
sw.WriteLine("Файл передан");
```

В *Закроем файл и соединение*

```
FS.Close();
```

```
sw.WriteLine("Соединение
```

```
прервано"); client.Close();
```



Из листинга следует, что по завершению передачи данных клиенту мы закрываем файл и соединение.

### 2.5.3 Порядок проведения исследований

Алгоритм выполнения заданий лабораторной работы следующий:

11. выбрать 5 задач по следующему правилу: номер по журналу- первая задача; номер каждой последующей задачи определяется прибавлением цифры 3 к номеру первой задачи, который только что вычислили (если достигнуто окончание списка вариантов задач, то перейти в его начало);
12. составить классы `Server` и `Client` и установить соединение между ними;
13. составить методы (функции) решения всех задач, поместить их в класс `Server`. Все исходные данные вводить в консоли клиента, передавать их на сервер, а полученные результаты решений задач выводить на экран в консоли клиента;
14. оформить отчет для всего приложения в целом, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и `skrin-shert` результата выполнения каждой задачи и представить его на проверку.

### 2.5.4 Варианты заданий

С Вывести на печать положительные значения функции  $y = \sin(x) + 5\cos(x-2)$  для  $x$  изменяющегося на отрезке  $[-5, 12]$  с шагом 1,2.

В Вывести на печать значения функции  $z = \operatorname{tg}(2x) - \sin(x)$  для  $x$  изменяющегося на отрезке  $[-3, 3]$  с шагом 0,3.

В Ввести с клавиатуры и напечатать модули  $N$  чисел; если введено отрицательное число, ввод и печать прекратить.

В Вывести на печать значения функции  $z = \ln(x) + \operatorname{tg}(2x)$ , большие 1, для  $x$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

изменяющегося на отрезке  $[3, 8]$  с шагом 0,9.



и Определить, является ли натуральное число  $N$  степенью числа 5 или нет.

и Напечатать значения функции  $y = \ln(x + 12/x)$ , где значения  $x$  вводятся с клавиатуры. При вводе числа, не входящего в область определения функции, вычисления прекратить.

10 Напечатать значения функции  $y = \ln(x - 1/x)$ , где значения  $x$  вводятся с клавиатуры. При вводе числа, не входящего в область определения функции, вычисления прекратить.

Для  $x$  из интервала  $(-2; 8)$  с шагом 0,75 вычислить  $y = (4x - 3x + \text{tg}(x))/A$ , где  $A$  вводится с клавиатуры.

Вывести на печать значения функции  $z = \sin(x) + \cos(x)$ , находящиеся в интервале  $(-0,2; 0,8)$  для  $x$  изменяющегося на отрезке  $[4, -6]$  с шагом 0,91.

12. Дано натуральное число  $N$ . Получить наименьшее число вида  $4^k$ , большее  $N$ .

В Для  $x$  из интервала  $(2; 8)$  с шагом 0,75 вычислить  $y = (4x - 3x + \cos(x))/A$ , где  $A$  вводится с клавиатуры.

Найти первый член последовательности  $\ln(9n)/(n \cdot n)$ , меньший 1, для  $n$

изменяющегося следующим образом:  $n = 1, 2, 3, \dots$

в Определить, является ли натуральное число  $N$  степенью числа 3 или нет.

16. Вывести на печать отрицательные значения функции  $z = \cos(x) - 5\sin(x - 2)$  для  $x$  изменяющегося на отрезке  $[-3, 11]$  с шагом 0,9.

- Ввести с клавиатуры и напечатать квадраты  $N$  чисел, если введено кратное 3 положительное число, ввод и печать прекратить.

- Вывести на печать отрицательные значения функции  $z = \text{tg}(x) + 5\cos(x - 2)$  для  $x$  изменяющегося на отрезке  $[12, 1]$  с шагом 1,2.

В Ввести с клавиатуры и напечатать  $N$  чисел, если введено равное нулю или кратное 2 число, ввод и печать прекратить.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**В** Вывести на печать значения функции  $z=\ln(|x|)+\text{tg}(2x)$ , большие 2 для  $x$  изменяющегося на отрезке  $[-3, -8]$  с шагом 0,9.



• Найти первый отрицательный член последовательности  $\sin(\operatorname{tg}(n/2))$  для  $n$  изменяющегося на следующем образом:  $n=1,2,3\dots$  .

11. Напечатать значения функции  $y=\ln(x+12/x)$ , где значения  $x$  вводятся с клавиатуры. При вводе числа, не входящего в область определения функции, вычисления прекратить.

11. Найти первую цифру в целом положительном числе.

12. Дано натуральное число  $N$ . Получить наибольшее число вида  $3^k$ , меньшее  $N$ .

• Вывести на печать значения функции  $z=\sin(x)+\cos(x)$ , находящиеся в интервале  $(-0,3;0,7)$  для  $x$  изменяющегося на отрезке  $[-4,6]$  с шагом  $0,91$ .

• Дано натуральное число  $N$ . Получить наименьшее число вида  $5^k$ , большее  $N$ .

• Для  $x$  из интервала  $(-2;8)$  с шагом  $0,75$  вычислить  $y=(4x-3x+\operatorname{tg}(x))/A$ , где  $A$  вводится с клавиатуры.

• Найти первый член последовательности  $\ln(9n/(n*n+1))$ , меньший  $0$ , для  $n$  изменяющегося на следующем образом:  $n=1,2,3\dots$  .

• Определить, является ли натуральное  $N$  степенью числа  $4$  или нет.

• Вывести на печать положительные значения функции  $z=\sin(x)-5\cos(x-2)$  для  $x$  изменяющегося на отрезке  $[5,-12]$  с шагом  $1,2$ .

• Напечатать значения функции  $Y = \sqrt{2x^2 - x^3}$  для произвольных  $x$ , вводимых с клавиатуры. При вводе числа, не входящего в область определения функции, ввод и печать прекратить.

и Найти первый отрицательный член последовательности  $\cos(\operatorname{ctg}(n))$  для  $n$  изменяющегося на следующим образом:  $n=1,2,3\dots$  .



## МУКЛР6

### Показатели качества в клиент-серверных приложениях в С#. Проектирование архитектуры и дизайна клиент-серверных приложений

#### 2.6.1 Цели занятия

Практически освоить основные правила проектирования архитектуры и дизайна приложений. Практически потренироваться в составлении интерфейсов клиентской части.

#### 2.6.2 Теоретические основы и пример выполнения

Состав клиент-серверных приложений, как составного программного обеспечения (ПО), очевидно, входят серверная и клиентская части. Как правило, обе эти составляющие приложения разрабатываются как интерфейсные продукты, т.е. содержащие специальные модули (блоки) для обеспечения связи человека и вычислительной среды. От этого зависит, в конечном итоге, насколько простым или сложным будет использование приложения, а также его оптимальное функционирование.

Поэтому, важно научиться составлять удобные в использовании и функционально оптимальные клиент-серверные продукты.

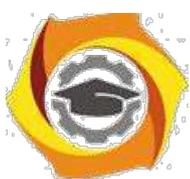
Тема данного практического занятия больше относится к реализации клиентских частей приложения. Однако, правила разработки приложений относятся в равной степени и к созданию серверной части приложения.

Сейчас при разработке ПО большое значение придается его архитектуре. И это не случайно. У любого программного продукта (ПП) есть свое конкретное назначение, которое определяется тем, как с помощью ПП пользователь может удовлетворить потребности в трех областях: пользовательской (интерфейс),



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

предметной (бизнес, исследования, потребительская и т.д.) и в системной с точки зрения структурности продукта. Поэтому, удачная архитектура приложения



делает его гибким, простым в использовании и сопровождении и высоко функциональным продуктом и наоборот.

Под **архитектурой** клиент-серверной системы будем понимать ее структуру, включающую программные элементы, видимые извне свойства этих элементов и взаимоотношения между ними [5].

Типовая архитектура для каждой из частей клиент-серверной системы представлена на рисунке 2.14.

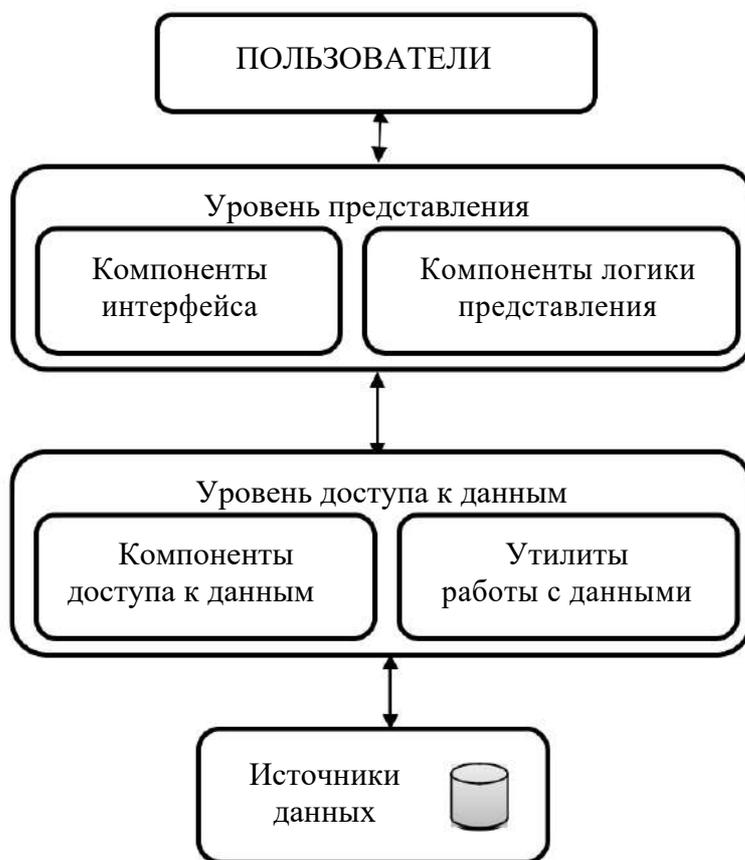


Рисунок 2.14-Типовая структура клиент-серверной системы

Для серверной части характерно еще и то, что ее пользователями могут быть не только люди, а и клиентские части системы.

Дадим краткие пояснения приведенной структуры.

Источники данных – это исходный уровень структуры, т.е. непосредственно данные, которые необходимо передавать или преобразовывать, если это



структура серверной части системы, или можно просто использовать, как говорят: «на стороне клиента». Традиционно, все данные в клиент-серверных системах располагаются на серверной части в виде базы данных, клиентские части системы имеют к ним доступ при обращении на сервер. Однако, структура системы позволяет размещать и хранить данные и на клиентских частях, если это необходимо [6].

Уровень доступа к данным обеспечивает работу с данными, поэтому он включает компоненты доступа к данным и утилиты работы с данными. Под «доступом к данным» подразумеваются алгоритмы их записи, извлечения и обработки. Соответственно этому и выбираются утилиты, их реализующие. Так, если используется база данных, то этот уровень реализуется в виде приложения, управляющего базой данных, если же данные просто размещаются по какому-то принципу на дисках сервера, то и приложение реализует поисковые алгоритмы доступа к ним.

Уровень представления служит для необходимой обработки данных и представления их в требуемом виде согласно клиентским запросам, а также для обеспечения возможности пользователю вообще работать с приложением. Поэтому этот уровень реализует алгоритмы обработки данных и компоненты пользовательского интерфейса. Обычно, это больше касается пользовательских частей системы, т.к. именно через них пользователи работают с клиент-серверными системами, однако и на серверной части возможны минимальные реализации этого уровня для настройки и эксплуатации серверной ее части. Сюда, как показано на рисунке 2.1, входят компоненты интерфейса пользователя и логики представления. Компоненты логики представления необходимы для реализации сценария подачи необходимой информации пользователю. Интерфейсные компоненты, в свою очередь, обеспечивают представление нужной информации пользователю.

Настоящее практическое занятие включает отработку всех описанных уровней реализации клиентского приложения. При этом, особое внимание уделяется уровню представления, т.к. уровень доступа к данным и сами данные



могут на клиентской стороне клиент-серверного приложения отсутствовать вообще.

Клиентскую часть приложения необходимо будет разработать, используя Visual Studio- интегрированную среду разработки программных продуктов, снабженную всеми необходимыми средствами написания, отладки и компиляции программ [7]. Воспользуемся языком программирования C# потому, что именно он представляет массу инструментов для создания интерфейсных приложений [8].

#### 2.6.2.1 Основные элементы графического интерфейса пользователя

Для создания клиентских частей клиент-серверных приложений используется обширная библиотека Form языка C# для приложений типа Windows Forms. Принцип составления оконного интерфейса прост- создается пустая форма, которая наследует (использует) библиотеку готовых решений Form. Далее, перетаскивая на эту форму необходимые элементы управления и отображения составляется клиентская часть для решения задачи, указанной на практическом занятии.

Элементов управления в указанной библиотеке множество, рассмотрим только часто употребляемые и необходимые для решения поставленных на занятии задач.

Основными элементами форм являются:

- К label – метка для отображения справочной информации;
- К textBox – текстовое поле для вводимых и выводимых данных;
- К button – кнопка для исполнения требуемых действий.

Приведем пример решения одной из задач и покажем как программируются указанные элементы интерфейса.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

2.6.2.2 Пример выполнения задания



Задание: даны действительные числа  $x, y$ . Вычислить значение функции  $z = \ln(x) - x/y$ .

Из задания следует, что необходимо будет на форму ввести две переменные и вычислить значение функции для их определенных величин. Это задается видом искомой функции. В данном случае – это логарифм, а областью определения этой функции являются только положительные числа, большие нуля. Т.е., функция имеет смысл только для  $x, y > 0$ . На форме также, очевидно, должна быть кнопка для выполнения расчетов и метка, либо текстовое поле для отображения результата.

Изначально создается проект Windows Forms с названием Form1 (это имя будет сформировано по умолчанию и может быть изменено). Далее, используя панель инструментов Visual Studio в режиме конструктора размещаем элементы интерфейса на форму (рисунок 2.15).

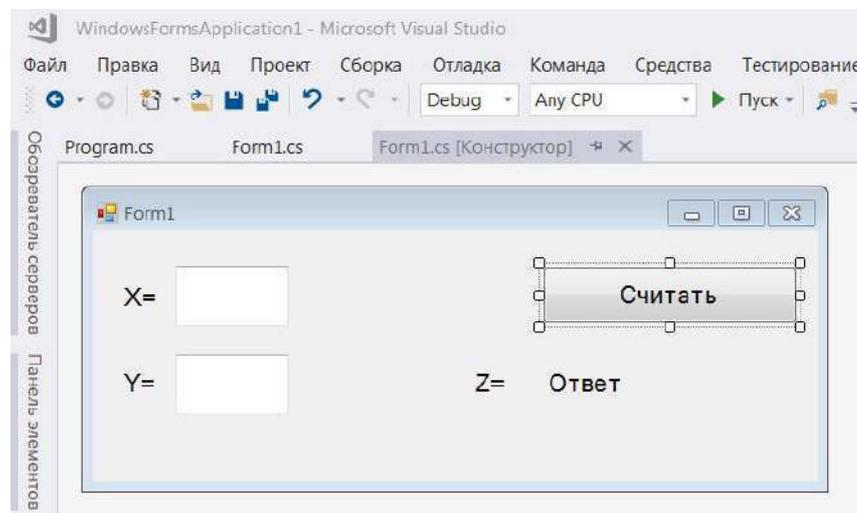


Рисунок 2.15- Конструирование формы интерфейса

При этом необходимый код программы создается компоновщиком форм автоматически.

Далее необходимо создать метод для вычисления функции  $Z$ . Дело в том, что весь проект содержит два класса: Program и Form1 (рисунок 2.16).

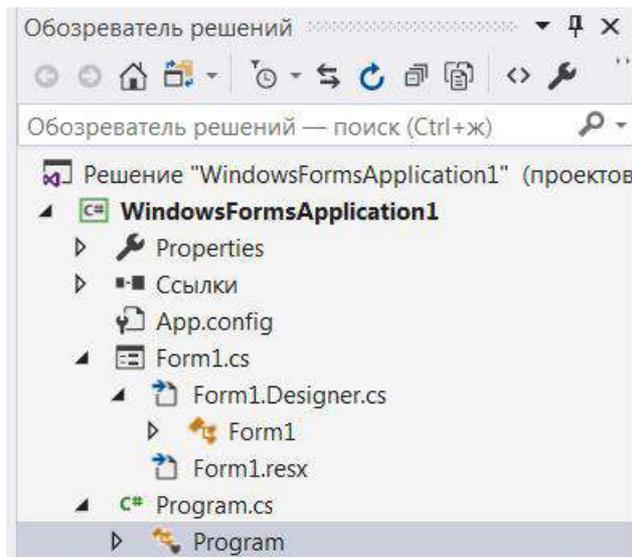


Рисунок 2.16- Состав клиентской части

Program нужен для запуска всего проекта и формирования разработанной формы, он содержит основной метод всего проекта – метод Main(). Класс Form1 содержит все необходимое для задания и отображения элементов формы. Коды этих классов, которые содержатся в файлах Program.cs и Form1.cs, представлены в листинге 2.25.

#### Листинг 2.25- Коды классов проекта

```
namespace WindowsFormsApplication1
{
 static class Program // код класса Program
 {
 В <summary>
 В Главная точка входа для приложения.
 В </summary>
 [STAThread]
 static void Main()
 {
 Application.EnableVisualStyles();
 Application.SetCompatibleTextRenderingDefault(false);
 Application.Run(new Form1());
 }
 public static double Metod(int x, int y) // Метод для
 вычисления Z
 {
 double z=0;
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*if* ( $y \neq 0$ )



```
 {
 z = Math.Log(x) - x / y;
 }
 else {
 z = 0;
 }
 return z;
 }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
 public partial class Form1 : Form // Код класса Form1
 {
 public Form1 ()
 {
 InitializeComponent();
 }

 private void button1_Click(object sender, EventArgs e)
 // Обработчик кнопки
 {
 label4.Text = Con-
 vert.ToString(Program.Metod(Convert.ToInt16(textBox1.Text),
 Convert.ToInt16(textBox2.Text)));
 }
 }
}
```

В классе Program составлен метод Metod(int x, int y), в котором вычисляется искомая функция. При этом, вычисления осуществляются по нажатию на кнопку «Считать», обработчик которой содержится в классе Form1. Обработчик события – это специальный метод (или еще говорят: «функция»), который создается автоматически, если дважды кликнуть по кнопке (или по другому эле-



менту интерфейса) на форме в режиме конструктора. Этот метод как бы «слушает» изменение состояния выбранного элемента интерфейса и если, например, кнопка будет нажата, то выполнится программный код, содержащийся в ее обработчике событий (листинг 2.1).

В обработчике кнопки находится строка присваивания значения метке `label4`, которая служит для отображения результатов вычислений. Оператор `label4.Text` указывает на то, что объект `label4` обращается к методу `Text`, чтобы поместить текстовую переменную в метку. Вообще, в `C#` ввод/вывод возможен только для текста. Поэтому, правая часть оператора присваивания содержит вызов метода `Method(int x, int y)`, который расположен в классе `Program` и содержит два параметра- переменные `X` и `Y`. Эти значения конвертируются сначала из текста текстовых полей `textBox1` и `textBox2`, куда они вводятся на форму, а потом результат выполнения всего метода конвертируется в текст и выводится в метку на форме как результат всей задачи.

### 2.6.3 Порядок выполнения задач ПЗ

Алгоритм выполнения задач практического занятия следующий:

\endash выбрать 5 задач по следующему правилу: номер по журналу- первая задача; номер каждой последующей задачи определяется прибавлением цифры 3 к номеру первой задачи, который только что вычислили (если достигнуто окончание списка вариантов задач, то перейти в его начало);

\endash составить методы (функции) решения всех задач, поместить их в класс `Form1` и снабдить пользовательским интерфейсом в `Windows Forms`. Это клиентская часть будущего клиент-серверного приложения, которое разрабатывается на следующих занятиях;

\endash оформить отчет для всего приложения в целом, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и `skrin-shert` результата выполнения каждой задачи и представить его на проверку.



#### 2.6.4 Варианты заданий

Решить задачи, используя оператор условного перехода if:

6. Даны действительные числа  $A, B, C, D$ . Выяснить, можно ли уместить прямоугольник со сторонами  $A, B$  внутри прямоугольника со сторонами  $C, D$ .
7. Даны действительные числа  $x, y, z$ . Найти минимальное из них.
8. Даны действительные положительные числа  $A, B, C$ . Выяснить, пройдет ли кирпич с ребрами  $A, B, C$  в прямоугольное отверстие со сторонами  $x, y$ .
4. Определить, лежит ли точка  $D(c, b)$ , где  $c = \sqrt{a_1 + a_2}$ ,  $b = a_1 + 0,7a_3$ , внутри прямоугольника со сторонами 5 и 10 ( $a_1, a_2, a_3$  - произвольные числа)?
8. Выяснить, существует ли треугольник с координатами вершин  $A(x_1, y_1)$ ,  $B(x_2, y_2)$ ,  $C(x_3, y_3)$ , если да, то найти его площадь.
6. Даны действительные числа  $A, B, C$ . Проверить выполняются ли неравенства  $A < B < C$ , если да, то присвоить  $A = B + C$  иначе  $A = C - B$ .
8. Даны действительные числа  $x, y$ . Вычислить значение функции  $z = \log(x - y) - x/y$
9. На плоскости расположена окружность радиуса  $R$  с центром в начале координат. Определить положение точки  $x$  с координатами  $(A, B)$  относительно окружности.
10. Даны круг радиуса  $R$  и квадрат со стороной  $A$ . Определить их взаимное положение.
11. Вывести на печать переменные  $A, B, C$  в порядке их возрастания
12. Проверить, какие из чисел  $A, B, C, D$  принадлежат интервалу  $(1, 25)$ .
13. Даны действительные числа  $A, B$ . Если они оба отрицательные, то заменить каждое из них его квадратом, иначе положительные из них увеличить в два раза.
6. Выяснить, существует ли треугольник с координатами вершин  $A(x_1, y_1)$ ,  $B(x_2, y_2)$ ,  $C(x_3, y_3)$ .



16. Даны действительные числа  $x, y$ . Вычислить значение функции  $z = \log(x/y) - 1/x$

Даны действительные числа  $A, B$ . Если они оба неотрицательные, то заменить каждое из них его кубом, иначе отрицательные из них заменить их модулями.

Даны площадь квадрата  $S_1$  и круга  $S_2$ . Определить поместится ли круг  
9. квадрат и наоборот.

На плоскости расположена окружность радиуса  $R$  с центром в начале координат. Определить, лежат ли точки  $A(x_1, y_1)$  и  $B(x_2, y_2)$  на окружности.

Составить программу вычисления корней системы уравнений с двумя неизвестными методом Крамера.

$$a_1x + b_1y = c_1$$

Проверить, что главный определитель системы не должен

$$a_2x + b_2y = c_2$$

быть равен 0.

17. Даны действительные числа  $A, B, C, D$ . Выяснить, можно ли уместить прямоугольник со сторонами  $A, B$  внутри прямоугольника со сторонами  $C, D$ .

18. Вывести на печать переменные  $A, B, C$  в порядке их убывания

19. Даны действительные числа  $x, y, z$ . Найти максимальное из них.

20. Проверить, какие из чисел  $A, B, C, D$  не принадлежат интервалу  $(3, 15)$ .

21. Даны действительные числа  $x, y$ . Вычислить значение функции  $z = \ln(x) - x/y$ .

22. Даны действительные числа  $A, B$ . Если они имеют разные знаки, то напечатать их произведение, иначе напечатать их квадраты.

23. Выяснить, существует ли треугольник с длинами сторон  $A, B, C$ . Если да, то найти его площадь.

24. Даны действительные числа  $x, y$ . Вычислить значение функции  $z = \arcsin(x) - y$ .

у Даны действительные числа  $x, y, z$ . Получить максимальное из них по



модулю.

**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



9. Даны действительные числа  $x, y$ . Вычислить значение функции  $z = \arcsin(x+y)$ .

В На каком из интервалов  $(- ; k_1), (k_1; k_2), (k_2; + )$  лежит точка  $x$ ? Где  $k_1, k_2, x$  – вводимые произвольные числа, причем  $k_1 < k_2$ .

6. Лежат ли обе точки  $D(a_1; b_1)$  и  $C(a_2; b_2)$  внутри круга радиуса  $R$  с центром в начале координат? Если такой точки нет, выдать соответствующее сообщение.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Дубаков А.А. Сетевое программирование: учебное пособие / А.А. Дубков – СПб: НИУ ИТМО, 2013. – 248 с.
- 2 Грамотная клиент-серверная архитектура: как правильно проектировать и разрабатывать web API. [Электронный ресурс] // [сайт] [2018], URL: <https://tproger.ru/articles/web-api/>. (дата обращения: 03.05.2018).
6. Технологии разработки клиент-серверных приложений. [Электронный ресурс] // [сайт] [2018], URL: <https://bourabai.ru/dbt/client2.htm>. (дата обращения: 05.05.2018).
- 4 Сенина А.А., Тузовский А.Ф Обзор основных современных технологий разработки web-приложений. [Электронный ресурс] // [сайт] [2018], URL: <https://core.ac.uk/download/pdf/53095773.pdf>. (дата обращения: 03.05.2018).
- 5 Басс Лен, Клементс Пол, Кацман Рик. Архитектура программного обеспечения на практике. 2-е издание. — СПб.: Питер, 2006. — 575 с.
- 6 П. Дейтел, Х. Дейтел, Э. Дейтел, М. Моргано. Android для программистов: создаём приложения. — СПб.: Питер, 2013. — 560 с.
- 7 Краткое описание Visual Studio [Электронный ресурс] // [сайт] [2018], URL: <http://habrahabr.ru>. (дата обращения: 10.05.2018).
7. Фленов М. Е. Библия C#. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2011. — 560 с.:



## Приложение А

### Код программы web – приложения ЛР № 1

*Код стартовой страницы Page1*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfBrowserApplication1
{
 - <summary>
 - Interaction logic for Page1.xaml
 - </summary>
 public partial class Page1 : Page
 {
 public Page1()
 {
 InitializeComponent();
 }

 private void button1_Click(object sender, RoutedEventArgs
e) {
 //string pup = System.IO.Directory.GetCurrentDirectory();
 - string task = zad.Text string pup =
 "I:\\Pril\\Lex"; webBrowser1.Navigate(pup
 + "\\\" + "1.mht");
 }
 }
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- `webBrowser1.Navigate("E:\\Lobz\\Laba3\\Lex\\1.html");`



```
}

private void button2_Click(object sender, RoutedEventArgs
e) {
 string pup = "I:\\Pril\\Lex";
 webBrowser1.Navigate(pup + "\\\" + "2.mht");
}

private void button3_Click(object sender, RoutedEventArgs
e) {
 string pup = "I:\\Pril\\Lex";
 webBrowser1.Navigate(pup + "\\\" + "3.mht");
}

private void button4_Click(object sender, RoutedEventArgs
e) {
 string pup = "I:\\Pril\\Lex";
 webBrowser1.Navigate(pup + "\\\" + "4.mht");
}

private void button5_Click(object sender, RoutedEventArgs
e) {
 string pup = "I:\\Pril\\Lex";
 webBrowser1.Navigate(pup + "\\\" + "p1.mht");
}

private void button6_Click(object sender, RoutedEventArgs
e) {
 if (comboBox1.SelectedIndex == 0)
 {
 string pup = "I:\\Pril\\Lex";
 webBrowser1.Navigate(pup + "\\\" + "Lab1.mht");
 }

 if (comboBox1.SelectedIndex == 1)
 {
 string pup = "I:\\Pril\\Lex";
 webBrowser1.Navigate(pup + "\\\" + "Lab2.mht");
 }
}
```



```
 }

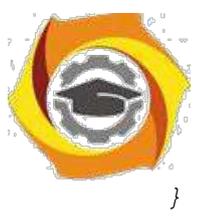
 if (comboBox1.SelectedIndex == 2)
 {
 string pup = "I:\\Pril\\Lex";
 webBrowser1.Navigate(pup + "\\\" + "Lab3.mht");
 }
}

private void button7_Click(object sender, RoutedEventArgs
e) {
 if (comboBox1.SelectedIndex == 0)
 {
 NavigationService.Navigate(new Uri("/Page2.xaml",
UriKind.Relative));
 }

 if (comboBox1.SelectedIndex == 1)
 {
 NavigationService.Navigate(new Uri("/Page3.xaml",
UriKind.Relative));
 }

 if (comboBox1.SelectedIndex == 2)
 {
 NavigationService.Navigate(new Uri("/Page4.xaml",
UriKind.Relative));
 }
}

private void comboBox1_SelectionChanged(object sender, Se-
lectionChangedEventArgs e)
{
}
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



```
Код страницы Page2
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfBrowserApplication1
{
 15. <summary>
 16. Логика взаимодействия для Page2.xaml
 17. </summary>
 public partial class Page2 : Page
 {
 public Page2()
 {
 InitializeComponent();
 }

 private void button1_Click(object sender, RoutedEventArgs
e) {
 NavigationService.Navigate(new Uri("/Page1.xaml",
UriKind.Relative));
 }

 private void button5_Click(object sender, RoutedEventArgs
e) {
 double A = 1.5, B = 2, C = 2.5;

```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
double x = double.Parse(textBox1.Text);
```



```
double z = (Math.Pow(x, 2) + Math.Abs(B * x - 3 * C)) /
(Math.Log(Math.Pow(x, 3) + B * C + A));
label9.Content = "Z = (X^2 + |B*x -
3*C|)/(Log(x^3)+B*C+A) = " + Math.Round(z, 3);
}
```

```
private void button2_Click(object sender, RoutedEventArgs
e) {
```

```
double opr, a1, a2, b1, b2, c1, c2;
```

```
a1 = Convert.ToInt16(textBox2.Text);
```

```
a2 = Convert.ToInt16(textBox3.Text);
```

```
b1 = Convert.ToInt16(textBox5.Text);
```

```
b2 = Convert.ToInt16(textBox4.Text);
```

```
c1 = Convert.ToInt16(textBox7.Text);
```

```
c2 = Convert.ToInt16(textBox6.Text);
```

```
opr = (a1 * b2) - (b1 * a2);
```

```
if (opr == 0)
```

```
{
```

```
label16.Content = "Определитель равен '0'";
```

```
}
```

```
else
```

```
{
```

```
label16.Content = "Определитель не равен '0'";
```

```
}
```

```
}
```

```
}
```

```
}
```

Код страницы Page3

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Windows;
```

```
using System.Windows.Controls;
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*using System.Windows.Data;*



```
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfBrowserApplication1
{
 В <summary>
 В Логика взаимодействия для Page3.xaml
 В </summary>
 public partial class Page3 : Page
 {
 public Page3()
 {
 InitializeComponent();
 }

 private void button5_Click(object sender, RoutedEventArgs
e) {
 double x1, x2, x;
 double z;

 x1 = Convert.ToDouble(textBox1.Text);
 x2 = Convert.ToDouble(textBox2.Text);
 for (x = x1; x < x2; x++)
 {
 z = (1 / (x + 2)) + (1 / (x + 5)) + Math.Log(12, 8 -
x);
 textBox3.Text += "Z [" + x + "] = " + Math.Round(z,
2) + "\r\n";
 }
 }

 private void button1_Click(object sender, RoutedEventArgs
e) {
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*double y1, y2, y;*



```
double Z = 3;

//y1 = Convert.ToDouble(textBox6.Text);
y1 = double.Parse(textBox10.Text); y2 =
Convert.ToDouble(textBox5.Text);

y = y1;
while ((Z > 2) && (y != y2))
{
 Z = Math.Log(Math.Abs(y)) + Math.Tan(2 * y);
 y = y - 0.9;
 textBox6.Text += "Z [" + y + "] = " + Math.Round(Z,
2) + "\r\n";
}
}

private void button2_Click(object sender, RoutedEventArgs
e) {
 double k1, k2, k;
 double Z = 3;

 k1 = Convert.ToDouble(textBox7.Text);
 k2 = Convert.ToDouble(textBox8.Text);

 k = k1;

 do
 {
 Z = Math.Cos(k) - 5 * Math.Sin(k - 2);
 k = k - 0.9;
 textBox9.Text += "Z [" + k + "] = " + Math.Round(Z,
2) + "\r\n";
 }
 while ((Z < 0) && (k != k2));
}

private void button3_Click(object sender, RoutedEventArgs
e) {
```



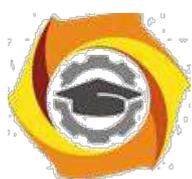
```
NavigationService.Navigate(new Uri("/Page1.xaml",
UriKind.Relative));
 }
 }
}
```

Код страницы Page4

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfBrowserApplication1
{
 10. <summary>
 11. Логика взаимодействия для Page4.xaml
 12. </summary>
 public partial class Page4 : Page
 {
 public Page4()
 {
 InitializeComponent();
 }

 private void Page_Loaded(object sender, RoutedEventArgs e)
 {
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

}



```
private void button5_Click(object sender, RoutedEventArgs e) {
 double i, k, sum = 0;

 textBox1.Text += "B = ";
 for (i = 1; i < 8; i++)
 {
 if ((i % 2) == 0)
 {
 k = Math.Sin(i) + 3;
 textBox1.Text += Math.Round(k, 0) + "
"; sum = sum + k;
 }
 else
 {
 k = i + Math.Cos(i - 1);
 textBox1.Text += Math.Round(k, 0) + "
"; sum = sum + k;
 }
 };
 textBox3.Text = Convert.ToString(Math.Round(sum, 0));
}

private void button3_Click(object sender, RoutedEventArgs e) {
 NavigationService.Navigate(new Uri("/Page1.xaml",
UriKind.Relative));
}

private void button1_Click(object sender, RoutedEventArgs e) {
 textBox2.Text = "";
 if (Convert.ToBoolean(radioButton1.IsChecked == true))
 {
 double[,] brr = new double[2, 2];
 double k = 2;
 }
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
for (int i = 0; i < 2; i++)
```



```
{
 for (int j = 0; j < 2; j++)
 {
 k = 1 / (k * k);
 k++;
 textBox2.Text += Math.Round(k, 0) + " ";
 }
 textBox2.Text += "\r" + "\n";
}

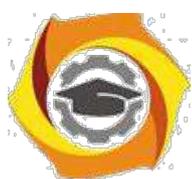
}

if (radioButton2.IsChecked == true)
{
 double[,] brr = new double[3, 3];
 double k = 2;

 for (int i = 0; i < 3; i++)
 {
 for (int j = 0; j < 3; j++)
 {
 k = 1 / (k * k);
 k++;
 textBox2.Text += Math.Round(k, 0) + " ";
 }
 textBox2.Text += "\r" + "\n";
 }
}

if (radioButton3.IsChecked == true)
{
 double[,] brr = new double[4, 4];
 double k = 2;

 for (int i = 0; i < 4; i++)
 {
 for (int j = 0; j < 4; j++)
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

{



```
k = 1 / (k * k);
k++;
textBox2.Text += Math.Round(k, 0) + " ";
}
textBox2.Text += "\r" + "\n";
}
}
}
}
```

## МУ к ПЗ-1

### Простые двухуровневые клиент-серверные приложения в Java

#### 2.1.1 Цели занятия

Выработать умения и навыки по составлению программ простых одноуровневых клиент-серверных приложений на основе технологии сокетов.

#### 2.1.2 Теоретические основы и пример выполнения

##### 2.1.2.1 Создание сетевых приложений с использованием TCP

Java поддерживает классы и методы, которые позволяют устанавливать соединение с удаленным компьютером, используя протокол TCP. В отличие от UDP, TCP является протоколом, ориентированным на установление соединения, которое гарантирует надежную связь между приложениями клиента и сервера. Взаимодействие с использованием протокола TCP, начинается после установления соединения между сокетами клиента и сервера. Сокет сервера "слушает" запросы на установление соединения, отправленные сокетами клиентов, и устанавливает соединение. После установления соединения между приложениями клиента и сервера, они могут взаимодействовать друг с другом.

Java упрощает сетевое программирование, путем инкапсуляции функциональности соединения сокета TCP в классы сокета, в которых класс Socket предназначен для создания сокета клиента, а класс ServerSocket для создания сокета сервера.



#### 2.1.2.2 Идентификация методов классов Socket и ServerSocket

Socket является базовым классом, поддерживающим протокол TCP. Класс Socket обеспечивает методы для потокового ввода/вывода, облегчает выполнение операций чтения и записи в сокет и является обязательным для программ, выполняющих сетевое взаимодействие.

Для создания объектов класса Socket используются следующие конструкторы, определенные в классе Socket:



26. `public Socket (InetAddress IP_address, int port)` - создает объект `Socket`, который соединяется хостом, заданным в параметрами `IP_address` и `port`.

27. `public Socket (String hostname, int port)` – создает объект `Socket`, который соединяется с хостом, заданным параметрами имя хоста или IP адрес и `port`, который сервер "слушает".

Некоторые полезные методы класса `Socket` представлены в таблице 2.1.

Таблица 2.1- Основные методы класса `Socket`

Метод	Описание
<code>public InetAddress()</code> <code>public getInetAddress()</code>	Возвращает объект <code>InetAddress</code> , который содержит IP-адрес, с которым соединяется объект <code>Socket</code>
<code>public InputStream()</code> <code>public getInputStream()</code>	Возвращает входной поток для объекта <code>Socket</code>
<code>public InetAddress()</code> <code>public getLocalAddress()</code>	Возвращает объект <code>InetAddress</code> , содержащий локальный адрес, с которым соединяется объект <code>Socket</code>
<code>public int getPort()</code>	Возвращает удаленный порт, с которым соединяется объект <code>Socket</code>
<code>public int getLocalPort()</code>	Возвращает локальный порт, с которым соединяется объект <code>Socket</code>
<code>public OutputStream()</code> <code>public getOutputStream()</code>	Возвращает выходной поток объекта <code>Socket</code>
<code>void close()</code>	Закрывает объект <code>Socket</code>
<code>public string toString()</code>	Возвращает IP-адрес и номер порта сокета клиента как <code>String</code>

`ServerSocket` представляет собой класс, используемый программами сервера для прослушивания запросов клиентов. `ServerSocket` реально не выполняет сервис, но создает объект `Socket` от имени клиента, через который выполняется взаимодействие с сокетом клиента.

Для создания и инициализации объектов `ServerSocket` используются следующие конструкторы, определенные в классе `ServerSocket`:



34. `public ServerSocket(int port_number)`- создает сокет сервера на заданный порт на локальной машине. Клиентам следует использовать этот порт, чтобы общаться с сервером. Если номер порта 0, то сокет сервера создается на любой свободный порт локальной машины;
35. `public ServerSocket(int port, int backlog)` - создает сокет сервера на заданный порт на локальной машине. Второй параметр задает максимальное количество соединений клиентов, которые сокет сервера поддерживает на заданном порту;
36. `public ServerSocket(int port, int backlog, InetAddress bindAddr)` - создает сокет сервера на заданный порт. Третий параметр используется для создания сокета сервера хоста, подключенного к нескольким физическим линиям (multi-homed host). Сокет сервера принимает запросы клиента только с заданных IP адресов.

Некоторые полезные методы класса `ServerSocket` представлены в таблице 2.2.

Таблица 2.2- Основные методы класса `ServerSocket`

Метод	Описание
<code>public InetAddress()</code> <code>public InetAddress()</code>	<code>getInetAddress()</code> Возвращает объект <code>InetAddress</code> , который содержит адрес объекта <code>ServerSocket</code>
<code>public int getLocalPort()</code>	Возвращает номер порта, с которого объект <code>ServerSocket</code> слушает запросы клиента
<code>public Socket accept() throws IOException</code>	Заставляет сокет сервера слушать соединение клиента и принимать его. После установления соединения клиента с сервером метод возвращает сокет клиента
<code>public void bind(SocketAddress address)</code>	Связывает объект <code>ServerSocket</code> с заданным



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

throws IOException

адресом (IP адрес и порт). Этот метод вы-



	зывает исключительную ситуацию IOException, когда происходит ошибка
<code>public void close() throws IOException</code>	Закрывает объект <code>ServerSocket</code> . Этот метод вызывает исключительную ситуацию <code>IOException</code> , когда происходит ошибка
<code>public String toString()</code>	Возвращает IP адрес и номер порта сокета сервера как <code>String</code>

### 2.1.2.3 Создание северной части по протоколу TCP/IP в Java

Для создания серверного приложения сокета TCP, необходимо выполнить следующие шаги:

24. создать объект сокета сервера `ServerSocket` (класс `Server`);
25. прослушивать запросы клиента на соединение;
26. запустить сервер;
27. создать поток соединения для запросов клиентов.

### 2.1.2.4 Разработка класса `Server`

Класс `Server` наследует класс `Thread` и, таким образом, поддерживает многопоточное выполнение. Объект `ServerSocket` "слушает" запросы клиентов и конструктор класса `Server` создает объект `ServerSocket`. Сообщение об ошибке отображается, если возникает исключительная ситуация при запуске сервера.

Фрагмент кода для конструктора сервера показан в листинге 2.1

#### Листинг 2.1- Фрагмент конструктора класса `Server`

```
public Server()
{
 try
 {
 serverSocket = new ServerSocket(1001);
 }
 catch(IOException e)
```



{

**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



```
fail(e, "Невозможно запустить
сервер."); }
System.out.println("Сервер запущен. . .");
this.start(); // Запускается поток
}
```

28. приведенном фрагменте кода используется общий метод обработки ошибок `fail()`, который обеспечивает обработку всех исключительных ситуаций. Метод принимает два аргумента (объект `Exception` и объект `String`) и выводит сообщение об ошибке.

Фрагмент кода для метода `fail()` выглядит следующим образом (листинг 2.2):

Листинг 2.2- Формирование сообщение об ошибке

```
public static void fail(Exception e, String str)
{
 System.out.println(str + "." + e);
}
```

#### 2.1.2.5 «Прослушивание» запросов клиентов

Метод `run()` класса `Server`, как любой поток, который реализует интерфейс `Runnable`, содержит инструкции для потока. В этом случае сервер переходит в бесконечный цикл и прослушивает запросы клиентов. Когда сервер обнаруживает подключение клиента, метод `accept()` класса `ServerSocket` выполняет соединение. При этом сервер создает объект класса `Connection` для клиента. Объект класса `Socket` передается конструктору класса `Connection` и взаимодействие между клиентом и сервером выполняется через этот сокет.

Фрагмент кода для метода `run()` выглядит показан на листинге 2.3.

Листинг 2.3- Метод для прослушивания запросов клиентов

```
public void run()
{
 try
```



{

**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



```
while(true)
{
 Socket client = serverSocket.accept();
 Connection con = new Connection(client);
}
}
catch(IOException e)
{
 fail(e, "Не прослушивается");
}
}
```

#### 2.1.2.6 Запуск сервера

Фрагмент кода для метода `main()` приведен в листинге 2.4.

#### Листинг 2.4- Запуск сервера

```
public static void main(String args[])
{
 new Server();
}
```

№       этом фрагменте кода создается объект класса `Server`, который запускает

поток.

#### 2.1.2.7 Создание потокового соединения

Класс `Connection` нужен для потокового соединения с клиентом (листинг 2.5).

#### Листинг 2.5- Класс для создания потокового соединения

```
class Connection extends Thread
{
 protected Socket netClient;
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
protected BufferedReader fromClient;
protected PrintStream toClient;
public Connection(Socket client) {
```



```
netClient = client;
try
{
fromClient = new BufferedReader(new
InputStreamReader(netClient.getInputStream())); toClient =
new PrintStream(netClient.getOutputStream());
}
catch(IOException e)
{
try
{
netClient.close();
}
catch(IOException e1)
{
System.err.println("Unable to set up streams"
50. e1);
return;
}
}
this.start();
}
public void run()
{
String clientMessage;
try
{
for(;;)
{
clientMessage = fromClient.readLine();
if(clientMessage == null)
break;
42. Посылает подтверждение клиенту
toClient.println("Received");
```



}

**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



```
 }
 catch (IOException e)
 {}
 finally
 {
 try
 {
 netClient.close();
 }
 catch (IOException e)
 {}
 }
}
```

В представленном фрагменте кода класс `Connection` создает объект `fromClient` класса `BufferedReader`, который получает ввод от клиента, используя метод `getInputStream()`. Объект класса `PrintStream` (`toClient`) дает возможность серверу отправлять клиенту данные, используя метод `getOutputStream()`. Таким образом, возникают необходимые (прием и передача) возможности взаимодействия.

Когда клиент соединяется с сервером, сервер использует метод `readLine()` объекта `fromClient`, чтобы запомнить сообщение, посланное клиентом в переменной `clientMessage` типа `String`. Метод `println()` используется для вывода сообщения “Received” сокету.

Для выхода из системы сервер завершает цикл. При этом выполняется блок `finally` для закрытия сокета клиента. Закрытие сокета является важным действием, поскольку сохранение активного соединения неизбежно приводит к потере памяти сервера. Блок `finally` обеспечивает закрытие ранее установленно-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

го соединения. Следует отметить, что сервер является многопоточным, и каждый клиент получает свой собственный поток на сервере.



### 2.1.2.8 Пример создания класса Server

В листинге 2.6 показан создания класса Server, который принимает запросы соединения клиента и посылает строку Login, как ответное сообщение клиенту.

#### Листинг 2.6- Пример кода класса Server

```
import java.io.*;
import java.net.*;
public class Server extends Thread {
 ServerSocket serverSocket; // Определяется переменная
server-Socket
 public Server() {
 try {
 /*
39. Создание объекта ServerSocket, который принимает запросы
40. соединения от клиентов через порт 1001
*/
 serverSocket = new ServerSocket(1001);
 System.out.println(serverSocket.toString());
 } catch (IOException e) {
 fail(e, "Could not start
server."); }
 System.out.println("Сервер запущен . . .");
 /* Стартует поток */
 this.start();
 }
 public static void fail(Exception e, String str)
 { System.out.println(str + "." + e); }

 public void run() {
 try {
 while (true) {
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*/\* Принимаются запросы от клиентов \*/*



```
Socket client = serverSocket.accept();
/*
```

46. Создается объект соединение, чтобы управлять взаимодействием

В клиента с сервером  
\*/

```
}
} catch (IOException e) {
fail(e, "Not listening");
}
}

public static void main(String args[]) {
/* Запускается сервер */
new Server();
}
}
```

```
class Connection extends Thread {
/* Declare the variables */
protected Socket netClient;
protected BufferedReader
fromClient; protected PrintStream
toClient; public Connection(Socket
client) { netClient = client; try {
```

/\* Создается входной поток, чтобы принимать данные от клиента \*/

```
fromClient = new BufferedReader(new
InputStreamReader(netClient.getInputStream()));
```

/\* Создается выходной поток, чтобы посылать данные клиенту \*/

```
toClient = new
PrintStream(netClient.getOutputStream());
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
} catch (IOException e) {
```



```
try {
 /* Закрывается сокет клиента */
 netClient.close();
} catch (IOException e1) {
 System.err.println("Unable to set up streams" +
 e1);
 return;
}
}

/* Start the thread */
this.start();
}

public void run() {
 String login, password;
 try {
 while (true) {
 toClient.println("Login: ");
 /* Принимается login как ввод от клиента */
 login = fromClient.readLine();
 /* Завершить соединение, когда Bye вводится как login */
 if (login == null || login.equals("Bye")) {
 System.out.println("Exit");
 return;
 } else
 System.out.println(login + "
 logged in");
 35.Посылается подтверждение клиенту
 toClient.println("Password: ");
 /* Принимается пароль то клиента
 */ password =
 fromClient.readLine(); }
 } catch (IOException e)
 { } finally {
 try {
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*netClient.close();*



```
} catch (IOException e) {
}
}
}
}
```

#### 2.1.2.9 Создание клиента TCP/IP

Первый шаг выполнения клиента состоит в установлении соединения между клиентом и сервером. Для установления соединения между клиентом и сервером необходимо создать объект `Socket`. Для создания приложения клиента сокета TCP необходимо выполнить следующие задачи:

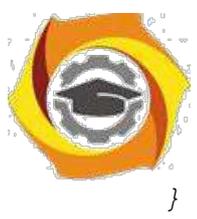
37. создать сокет клиента, используя объект `Socket`;
38. читать и писать в сокет;
39. закрыть соединение.

#### 2.1.2.10 Создание сокета клиента

Объект сокета клиента создается с помощью конструктора класса `Socket`, принимающего два параметра, IP адрес и номер порта, которые прослушиваются сервером. Следующий фрагмент кода используется для создания сокета клиента (листинг 2.7).

#### Листинг 2.7- Создание сокета клиента

```
try
{
 Socket clientSocket = new Socket("127.0.0.1", 1001);
}
catch (UnknownHostException e)
{
 System.err.println("Неопределенное имя хоста ");
 System.exit(1);
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



В предыдущем фрагменте кода IP адрес равный '127.0.0.1' и порт равный '1001' определяют сокет, на котором сервер прослушивает запросы клиента.

#### 2.1.2.11 Чтение и запись в сокет

После установления соединения между клиентом и сервером, клиент посылает запрос на сервер через сокет. Чтение и запись в сокет аналогичны чтению из файла и записи в файл. Чтобы обеспечить возможность клиенту общаться с сервером, необходимо выполнить следующие действия:

11. объявляются два объекта по одному для классов `PrintStream` и `BufferedReader`. Эти объекты будут использоваться для чтения и записи в сокет `socket`.  
`PrintStream out = null; // Объект для записи в сокет`  
`BufferedReader in = null; // Объект для чтения из сокета;`
12. объекты `PrintStream` и `BufferedReader` связываются с сокетом. `out = new PrintStream(clientSocket.getOutputStream()); in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));`

Методы `getInputStream()` и `getOutputStream()` класса `Socket` позволяют клиенту взаимодействовать с сервером. Метод `getInputStream()` позволяет объекту `BufferedReader` читать из сокета, а метод `getOutputStream()` позволяет объекту `PrintStream` писать в сокет.

Объявляется еще один объект класса `BufferedReader` для связи со стандартным входом, чтобы данные, введенные в приложении клиенте, могли передаваться на сервер. Следующий фрагмент кода используется для чтения данных из окна консоли (листинг 2.8).

#### Листинг 2.8- Чтение данных из окна консоли

```
BufferedReader stdin = new BufferedReader(new
InputStreamReader(System.in));
String str;
while((str = stdin.readLine()).length() != 0)
{
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
out.println(str);
```



}

Представленный фрагмент кода позволяет пользователю вводить данные с клавиатуры. Цикл `while` продолжается до тех пор, пока пользователь не введет символ завершения ввода (`Ctrl-Z`).

#### 2.1.2.12 Закрытие соединения

Операторы, представленные ниже, закрывают потоки и соединения с сервером: `out.close()`; `in.close()`; `stdin.close()`.

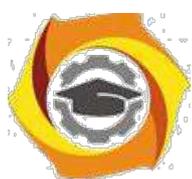
Сокет клиента принимает имя пользователя и пароль, и обеспечивает связь. Для завершения соединения, пользователь должен ввести 'Bye'. Можно использовать следующий код для создания класса `Client` (листинг 2.9).

#### Листинг 2.9- Пример создания класса `Client`

```
import java.net.*;
import java.io.*;
public class Client {
public static void main(String[] args) throws IOException {
Socket clientSocket;
PrintStream out = null;
BufferedReader in = null;
try {
/* Создается объект сокет, чтобы соединиться с
сервером */
clientSocket = new Socket("127.0.0.1", 1001);
/* Создается выходной поток, чтобы посылать данные на
сервер */
out = new
PrintStream(clientSocket.getOutputStream());
/* Создается входной поток, чтобы принимать данные с
сервера */
in = new BufferedReader(new InputStreamReader(
clientSocket.getInputStream()));
```



```
 } catch (UnknownHostException e) {
 System.err.println("Unidentified hostname
 "); System.exit(1);
 } catch (IOException e) {
 System.err.println("Couldn't get I/O
 "); System.exit(1);
 }
 /* Создается входной поток, чтобы читать данные из окна консо-
ЛИ
 */
 BufferedReader stdin = new BufferedReader(new
 InputStreamReader(
 (System.in)));
 /* Чтение из сокета */
 String login = in.readLine();
 System.out.println(login);
 /* Прием login */
 String logName = stdin.readLine();
 out.println(logName); /* Чтение из
сокета */
 String password = in.readLine();
 System.out.println(password);
 /* Прием password */
 String pass = stdin.readLine();
 out.println(pass);
 String str = in.readLine();
 System.out.println(str);
 while ((str = stdin.readLine()) != null) {
 out.println(str);
 if (str.equals("Bye"))
 break;
 }
 out.close();
 in.close();
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

```
stdin.close();
```



}  
}

Представленный выше код сохраняется как Client.java. При выполнении класса Client, отображается сообщение для ввода Login. После приема Login на консоли класса Client появляется сообщение для ввода password. Рис. 7 показывает вывод класса Client, который посылает login XXXX и password для входа на Server. Введенный login в окне консоли класса Client передается классу Server. Когда на сервер передается строка « Bye » в качестве login от клиента, со-единение между клиентом и сервером прекращается.

### 2.1.3 Порядок выполнения задач ПЗ

Алгоритм выполнения задач практического занятия следующий:

19. выбрать 5 задач по следующему правилу: номер по журналу- первая за-дача; номер каждой последующей задачи определяется прибавлением цифры 3 к номеру первой задачи, который только что вычислили (если достигнуто окончание списка вариантов задач, то перейти в его начало);
20. составить классы Server и Client и установить соединение между ними;
21. составить методы (функции) решения всех задач, поместить их в класс Server. Все исходные данные вводить в консоли клиента, передавать их на сервер, а полученные результаты решений задач выводить на экран в консоли клиента;
22. оформить отчет для всего приложения в целом, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и скрин-шерт результата выполнения каждой задачи и представить его на провер-  
ку.

### 2.1.4 Варианты заданий:

19. Составить программу для перевода длины в метрах в длину в санти-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

метрах, определив функцию, выполняющую это преобразование и передав длину в метрах в качестве параметра.



21. Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив функцию, выполняющую это действие, и передавая массивы в качестве параметра.

22. Даны числа  $S, T$ . Получить с использованием функции пользователя  $F(T, -2S, 1.17) + F(2.2, T, S - T)$  где  $F(A, B, C) = (2A - B - \sin(C)) / (5 + C)$

19. Составить программу перевода двоичной записи натурального числа в десятичную, описав соответствующую функцию с параметром. Перевод осуществлять для чисел, вводимых с клавиатуры. Признак конца ввода - число 0.

5. Даны числа  $S, T$ . Получить с использованием функции пользователя

с параметрами  $G(1, \sin(S)) + 2G(T * S, 24) - G(5, -S)$ , где  $G(A, B) = (2A + B * B) / (A * B * 2 + B * 5)$ .

19. Составить программу для расчета значений гипотенузы треугольника, определив функцию, выполняющую этот расчет. Катеты передаются в качестве параметров.

17. Найти периметр десятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, которые передаются функции в качестве параметров из основной программы.

14. Найти периметр шестиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами. Координаты передаются функции в качестве параметров из основной программы.

11. Найти площадь пятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, и процедуру вычисления площади треугольника по трем сторонам. Описать функции с соответствующими формальными параметрами.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

И Составить программу вывода на экран всех натуральных чисел, не превосходящих  $N$  и делящихся на каждую из своих цифр. Описать соответствующую функцию, получающую из основной программы в качестве пара-



метра натуральное число и возвращающую TRUE, если оно удовлетворяет указанному условию.

- Используя подпрограмму - функцию, составить программу для нахождения максимального из трех чисел. Числа передаются функции в качестве параметров.

- Используя подпрограмму - функцию, составить программу для печати знаков трех чисел, введенных с клавиатуры и передаваемых функции в качестве параметра.

- Используя подпрограмму - функцию, составить программу для возведения чисел в целую положительную степень. Число передается функции в качестве параметра из основной программы. Расчет вести для чисел, пока не будет введено число, равное 0.

13) Используя подпрограмму - функцию, составить программу для вычисления функции  $Z=(X1+Y1)/(X1*Y1)$ , где  $X1$  - первый корень уравнения  $X^2-4*X-1=0$ ;  $Y1$  - первый корень уравнения  $2*Y^2 + A*Y - A^2 = 0$  ( $A$  - произвольное).

и Задав функцию, вывести на печать средние арифметические двух массивов, введенных с клавиатуры. Массив передается функции в качестве параметра.

- Задав функцию, рассчитать и вывести на печать максимальные значения в трех парах чисел, вводимых с клавиатуры. Пара чисел передается функции в качестве параметра.

- Найти периметр восьмиугольника, координаты вершин которого заданы. Определить функцию вычисления расстояния между двумя точками, заданными своими координатами. Координаты передать функции в качестве параметров.

В Даны четыре пары чисел. Получить с использованием функции пользователя наибольший общий делитель для каждой пары.



- Даны числа  $A, B, C$ . Получить с использованием функции пользователя наименьшее значение. Числа передаются функции из основной программы в качестве параметров.

В Даны числа  $x = 1, 2, \dots, N$ . Получить с использованием функции пользователя значения  $3 * P(x+3) * P(x)$  для заданных  $x$ , где  $P(x) = 10 * x^3 - 14 * x^2 + 12 * x - 2$ .

в Составить программу для расчета значений катета треугольника, определив функцию, выполняющую этот расчет. Гипотенуза и второй катет передаются в качестве параметров.

к Даны целые числа  $a, b, c, d$ . Проверить с использованием функции пользователя их четность. Число для проверки передается в функцию в качестве параметра из основной программы.

О Для каждого из 10 введенных с клавиатуры чисел напечатать сообщение: является ли оно простым или нет, описав функцию логического типа, возвращающую значение “ИСТИНА”, если число, переданное ей в качестве параметра, является простым.

в Даны числа  $S, T$ . Получить с использованием функции пользователя  $Y(T, S) = G(12, S) + G(T, S) - G(2S - 1, S * T)$ , где  $G(A, B) = (2 * A + B * B) / (A * B * 2 + B * 5)$ .

В Определите функцию, определяющую, какой целой степенью числа 2 является ее аргумент (если число не является степенью двойки - выдать соответствующее сообщение).

8) Определите функцию, подсчитывающую сумму  $N$  первых элементов целочисленного массива  $A$ .  $N$  и массив  $A$  передать в качестве параметров.

9) Вычислить количество простых чисел, не превосходящих заданного  $N$ . Описать функцию логического типа, возвращающую значение true, если число простое и false в противном случае.

к Используя подпрограмму - функцию с параметрами, составить



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

программу для вычисления функции  $F(X,Y) = (2X^3 - 4X^2 + X + 1) / (9Y^3 + Y + 4) + 3Y^2 + 5Y$ .



16. Составить программу для перевода веса в граммах в вес в килограммах, определив функцию, выполняющую это преобразование. Вес в граммах передается функции в качестве параметра.

в Даны числа  $S, T$ . Получить с использованием функции пользователя  $G(12, S)+G(T, S)-G(2S-1, S*T)$  где  $G(A, B) = (2*A+B*B)/(A*B*2+B*5)$ .



## МУ к ПЗ-2

### Простые двухуровневые клиент-серверные приложения в C#

#### 2.2.1 Цели занятия

Выработать умения и навыки составлять типовые программы решения задач на выбранном языке программирования, снабженные элементами графического интерфейса пользователя в виде клиент-серверных приложений.

#### 2.2.2 Теоретические основы и пример выполнения

Для создания клиент-серверного приложения в C#, аналогично предыдущему ПЗ, необходимо создать два класса Server и Client, используя технологию сокетов.

##### 2.2.2.1 Программирование класса Server

Вначале в этом классе создадим серверное соединение и используем для этого систему сокетов.

Сокеты- это концепция сетевого программирования, когда существуют два вида «программных разъемов»- сокетов: клиентские и серверные.

Серверный сокет как розетка, которая «висит на стене» готовая к работе, в ожидании, когда к ней подключат штекер. Точно так же серверный сокет переходит в режим ожидания подключения на определенном адресе и определенном порту.

Клиентский сокет как вилка, которую втыкают в розетку. Как только клиент подключается к серверному сокету, информация начинает передаваться между ними.

Для того чтобы создать виртуальное подключение между клиентом и сервером, надо знать место, где находится нужный нам серверный сокет.

32.     нашем случае, создается слушатель событий подключений в сети по протоколу TCP IP – TcpListener, в качестве параметра которому передается так называемая точка входа, которую определяет метод IPEndPoint из переданных



ему IP адреса компьютера, на котором будет запущен сервер (пока это вариант localhost, т.е. IP 127.0.0.1) и номер порта Port, на котором создаваемой серверное подключение будет прослушивать подключения клиентов. Т.о., в конструкторе класса Server появится следующий слушатель (листинг 2.10).

Листинг 2.10- Серверный слушатель подключения клиентов

```
TcpListener listener;

public Program(int Port)
{

 listener = new TcpListener(new IPEndPoint(IPAddress.Parse("127.0.0.1"), Port));
 listener.Start();
 Console.WriteLine("Сервер запущен, ожидает подключений...");
 ...}
```

Видно, что после запуска прослушивания сетевых подключений выдается сообщение о том, что сервер готов и ожидает подключений.

Далее, нужно в это подключение с помощью метода `AcceptTcpClient()` и создать новый поток для него с помощью метода `Thread` из одноименной библиотеки, которую нужно подключить заранее: `using System.Threading`.

В качестве параметра этому методу передается метод `clientThread`, в котором для каждого нового клиента открывается TCP соединение в новом потоке (листинг 2.11).

Листинг 2.11- Метод создания нового TCP соединения для нового клиента

```
static void clientThread(Object StateInfo)
{
 new Client((TcpClient)StateInfo);
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

В итоге получим следующий текст класса Server (листинг 2.12).



### Листинг 2.12- Класс Server

```
class Server
{
 TcpListener listner;
 public Server (int Port)
 {
 listner = new TcpListener(new IPEnd-
Point(IPAddress.Parse("127.0.0.1"), Port));
 listner.Start();
 Console.WriteLine("Сервер запущен, ожидает подклю-
чений...");

 while (true)
 {

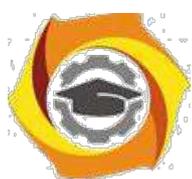
 TcpClient client = listner.AcceptTcpClient();
 20. Создаем поток

 Thread Thread = new Thread(new ParameterizedThread-
Start(clientThread));
 // И запускаем этот поток, передавая ему при-
нятого клиента

 Thread.Start(client);
 }
 }

 static void clientThread(Object StateInfo)
 {
 new Client((TcpClient)StateInfo);
 }

 static void Main(string[] args)
 {
 new Program(12000);
 }
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

}



}

Видно, что сюда же включен и главный метод модуля – метод `Main`, из которого происходит запуск всего приложения и сервер начинает «прослушивание» на порту №12000, который в качестве параметра передается конструктору класса.

#### 2.2.2.2 Класс `Client`

Этот класс необходим для считывания запроса от клиентских приложений, поиска нужного контента согласно запросу и для его передачи в сеть клиентам.

Начнем с задания объектов для методов чтения из потока и записи в поток. Для этого создаются новые экземпляры соответствующих классов (листинг 2.13).

#### Листинг 2.13- Организация потокового чтения и записи

```
StreamReader sr = new StreamReader(client.GetStream()); //
чтение из потока клиента

StreamWriter sw = new StreamWriter(client.GetStream()); // запись в поток клиента

sw.AutoFlush = true;
```

Этот пример клиент-серверного приложения обеспечивает запрос от клиента на сервер необходимого файла информации, передачу его клиенту, сохранение в темповском файле на клиенте и отображение в клиентской части приложения. Поэтому, определим какой файл необходим: прочитаем из потока в запросе клиента значение строковой переменной `str`, конвертируем его в целочисленную переменную `kluch`, и по ее значению в структуре `switch` определим необходимый файл (листинг 2.14).



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Листинг 2.14- Определение требуемого контента



13. Принимаем передачу от клиента

```
string cl = sr.ReadLine(); // Отображение типа
обучающего модуля

Console.WriteLine(cl);

string str = sr.ReadLine();

int kluch = int.Parse(str); // по kluch определяем
нужный файл

string FilePath = "C:/www/"; // составляем путь к
файлу

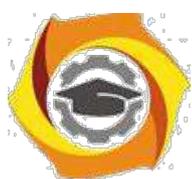
switch (kluch)
{
 case 1:
 {
 FilePath += "L1.htm";

 Console.WriteLine("Путь к файлу: " +
FilePath);
 }

 break;

 case 2:
 {
 FilePath += "L2.htm";

 Console.WriteLine("Путь к файлу: " +
FilePath);
 }
}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

}



```
break;

}
```

Видно, что весь контент преобразован в html формат и находится в соответствующих папках на сервере. Аналогично программируется определение оставшихся файлов обучающих материалов.

Далее нужно открыть найденный файл. Сделаем это, проверяя правильность его открытия с помощью блока try/catch и если возникнет ошибка открытия, выдадим соответствующее сообщение (листинг 2.15).

#### Листинг 2.15- Безопасное открытие файла контента

*14.Открываем файл, страхуясь на случай ошибки*

```
FileStream FS;

try
{

 FS = new FileStream(FilePath, FileMode.Open,
FileStream.Read, FileShare.Read);

 sw.WriteLine("Запрашиваемый файл открыт и го-
тов к передаче");

}

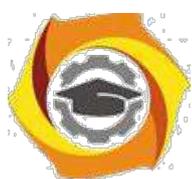
catch (Exception)
{

 13. Если случилась ошибка, посылаем клиенту
ошибку открытия файла

 sw.WriteLine("Ошибка открытия файла");

 return;

}
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

}



Учитывая, что данные передаются в сети в формате байт, нужно зарезервировать буфер для чтения материалов из открытого файла. Затем, перенести эти данные в буфер и через него передать их клиенту в ответе на запрос. Весь этот процесс будет выполняться до конца данных в открытом файле (листинг 2.16).

#### Листинг 2.16- Передача контента в ответном сообщении клиенту

```
У Задаем буфер для размещения данных из выбранного
 файла byte[] Buffer = new byte[1024];
 10. Переменная для хранения количества байт, пере-
 даваемых клиенту
 int Count;
 ' Пока не достигнут конец файла
 while (FS.Position < FS.Length)
 {
 Читаем данные из файла и пишем их в буфер
 И передаем их клиенту
 client.GetStream().Write(Buffer, 0, Count);
 }
 sw.WriteLine("Файл передан");
 В Закроем файл и соединение
 FS.Close();
 sw.WriteLine("Соединение
 прервано"); client.Close();
```



Из листинга следует, что по завершению передачи данных клиенту мы закрываем файл и соединение.

### 2.2.3 Порядок выполнения задач ПЗ

Алгоритм выполнения задач практического занятия следующий:

- и выбрать 5 задач по следующему правилу: номер по журналу- первая задача; номер каждой последующей задачи определяется прибавлением цифры 3 к номеру первой задачи, который только что вычислили (если достигнуто окончание списка вариантов задач, то перейти в его начало);
- и составить классы `Server` и `Client` и установить соединение между ними;
- и составить методы (функции) решения всех задач, поместить их в класс `Server`. Все исходные данные вводить в консоли клиента, передавать их на сервер, а полученные результаты решений задач выводить на экран в консоли клиента;
- и оформить отчет для всего приложения в целом, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и `screen-shert` результата выполнения каждой задачи и представить его на проверку.

### 2.2.4 Варианты заданий

- Вывести на печать положительные значения функции  $y=\sin(x)+5\cos(x-2)$  для  $x$  изменяющегося на отрезке  $[-5, 12]$  с шагом 1,2.

- Вывести на печать значения функции  $z=\text{tg}(2x)-\sin(x)$  для  $x$  изменяющегося на отрезке  $[-3, 3]$  с шагом 0,3.

11. Ввести с клавиатуры и напечатать модули  $N$  чисел; если введено отрицательное число, ввод и печать прекратить.

12. Вывести на печать значения функции  $z=\ln(x)+\text{tg}(2x)$ , большие 1, для  $x$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

изменяющегося на отрезке  $[3, 8]$  с шагом 0,9.



- Определить, является ли натуральное число  $N$  степенью числа 5 или нет.  
- Напечатать значения функции  $y=\ln(x+12/x)$ , где значения  $x$  вводятся с клавиатуры. При вводе числа, не входящего в область определения функции, вычисления прекратить.

- Напечатать значения функции  $y=\ln(x-1/x)$ , где значения  $x$  вводятся с клавиатуры. При вводе числа, не входящего в область определения функции, вычисления прекратить.

Для  $x$  из интервала  $(-2;8)$  с шагом 0,75 вычислить  $y=(4x-3x+\text{tg}(x))/A$ , где  
- вводится с клавиатуры.

Вывести на печать значения функции  $z=\sin(x)+\cos(x)$ , находящиеся в интервале  $(-0,2; 0,8)$  для  $x$  изменяющегося на отрезке  $[4,-6]$  с шагом 0,91.

в Дано натуральное число  $N$ . Получить наименьшее число вида  $4^k$ , большее  $N$ .

С Для  $x$  из интервала  $(2;8)$  с шагом 0,75 вычислить  $y=(4x-3x+\cos(x))/A$ , где  $A$  вводится с клавиатуры.

15. Найти первый член последовательности  $\ln(9n)/(n*n)$ , меньший 1, для  $n$  изменяющегося следующим образом:  $n=1,2,3... .$

Определить, является ли натуральное число  $N$  степенью числа 3 или нет.

28. Вывести на печать отрицательные значения функции  $z=\cos(x)-5\sin(x-2)$  для  $x$  изменяющегося на отрезке  $[-3, 11]$  с шагом 0,9.

Ввести с клавиатуры и напечатать квадраты  $N$  чисел, если введено кратное 3 положительное число, ввод и печать прекратить.

16. Вывести на печать отрицательные значения функции  $z=\text{tg}(x)+5\cos(x-2)$  для  $x$  изменяющегося на отрезке  $[12, 1]$  с шагом 1,2.

В Ввести с клавиатуры и напечатать  $N$  чисел, если введено равное нулю или кратное 2 число, ввод и печать прекратить.

В Вывести на печать значения функции  $z=\ln(|x|)+\text{tg}(2x)$ , большие 2 для  $x$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

изменяющегося на отрезке  $[3, -8]$  с шагом 0,9.



с Найти первый отрицательный член последовательности  $\sin(\operatorname{tg}(n/2))$  для  $n$  изменяющегося на следующем образом:  $n=1,2,3\dots$  .

Напечатать значения функции  $y=\ln(x+12/x)$ , где значения  $x$  вводятся с клавиатуры. При вводе числа, не входящего в область определения функции,

вычисления прекратить.

40. Найти первую цифру в целом положительном числе.

41. Дано натуральное число  $N$ . Получить наибольшее число вида  $3^k$ , меньшее  $N$ .

В Вывести на печать значения функции  $z=\sin(x)+\cos(x)$ , находящиеся в интервале  $(-0,3;0,7)$  для  $x$  изменяющегося на отрезке  $[-4,6]$  с шагом  $0,91$ .

В Дано натуральное число  $N$ . Получить наименьшее число вида  $5^k$ , большее  $N$ .

9) Для  $x$  из интервала  $(-2;8)$  с шагом  $0,75$  вычислить  $y=(4x-3x+\operatorname{tg}(x))/A$ , где  $A$  вводится с клавиатуры.

в Найти первый член последовательности  $\ln(9n/(n*n+1))$ , меньший  $0$ , для  $n$  изменяющегося на следующем образом:  $n=1,2,3\dots$  .

10) Определить, является ли натуральное  $N$  степенью числа  $4$  или нет.

11) Вывести на печать положительные значения функции  $z=\sin(x)-5\cos(x-2)$  для  $x$  изменяющегося на отрезке  $[5,-12]$  с шагом  $1,2$ .

8. Напечатать значения функции  $Y = \sqrt{2x^2 - x^3}$  для произвольных  $x$ , вводимых с клавиатуры. При вводе числа, не входящего в область определения функции, ввод и печать прекратить.

14. Найти первый отрицательный член последовательности  $\cos(\operatorname{ctg}(n))$  для  $n$  изменяющегося на следующем образом:  $n=1,2,3\dots$  .



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Дубаков А.А. Сетевое программирование: учебное пособие / А.А. Дубков – СПб: НИУ ИТМО, 2013. – 248 с.

2 Грамотная клиент-серверная архитектура: как правильно проектировать и разрабатывать web API. [Электронный ресурс] // [сайт] [2018], URL: <https://tproger.ru/articles/web-api/>. (дата обращения: 03.05.2018).

26. Технологии разработки клиент-серверных приложений. [Электронный ресурс] // [сайт] [2018], URL: <https://bourabai.ru/dbt/client2.htm>. (дата обращения: 05.05.2018).

4 Сенина А.А., Тузовский А.Ф Обзор основных современных технологий разработки web-приложений. [Электронный ресурс] // [сайт] [2018], URL: <https://core.ac.uk/download/pdf/53095773.pdf>. (дата обращения: 03.05.2018).

5 Басс Лен, Клементс Пол, Кацман Рик. Архитектура программного обеспечения на практике. 2-е издание. — СПб.: Питер, 2006. — 575 с.

6 П. Дейтел, Х. Дейтел, Э. Дейтел, М. Моргано. Android для программистов: создаём приложения. — СПб.: Питер, 2013. — 560 с.

7 Краткое описание Visual Studio [Электронный ресурс] // [сайт] [2018], URL: <http://habrahabr.ru>. (дата обращения: 10.05.2018).

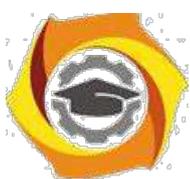
30. Фленов М. Е. Библия C#. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2011. — 560 с.:

### Психология и педагогика

#### Методические указания по практическим занятиям

#### I. Введение

##### 1. Цели изучения дисциплины



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Овладение студентами основными понятиями общей психологии; освоение понятиями психологии управления персоналом, малыми группами и коллективом в целом; освоение основных понятий педагогической науки.

## 2. Планируемые результаты обучения

Изучение дисциплины направлено на формирование у выпускника способности решать профессиональные задачи в соответствии с **технологическим** видом деятельности.

Результатом освоения дисциплины являются сформированные у выпускника следующие компетенции:

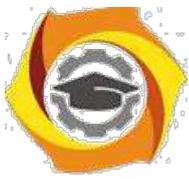
<b>Указания к практическим занятиям</b>
<b>15. Методические указания к практическим занятиям очной формы обучения Модуль I.</b>
<b>Практическое занятие 1. Специфика психологического познания</b>
Целью проводимого занятия является формирование знаний: основных направлений психологических исследований и методологии психологических исследований; особенности исторического развития понятий личности; специфические человеческие особенности и свойства (речь, сознание, трудовая деятельность); значение социально-культурного влияния на формирование личности.
<b>Форма проведения:</b> контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.
<b>Вопросы к контрольной работе:</b>
27. История развития психологии в древнем мире и новое время.
28. Психология в XIX веке.
29. Основные психологические направления и теории в XX веке.
30. Психоанализ
31. Бихевиоризм
32. Гуманистическая психология
33. Концепции личности в духовно ориентированной психологии
34. Личность как система отношений
35. Личность в системе межличностных связей

### Дополнительные вопросы для обсуждения и самопроверки:

- в Специфика развития учения о душе в греческой философии
- в Понимание психологии христианскими мыслителями.
- в Психофизические вопросы философии Нового времени.
- в Основные понятия психоанализа З.Фрейда?
- в В чем специфика бихевиоризма?
- в Кто является апологетом гуманистической психологии?
- в Что можно отметить в концепции личности в духовно ориентированной психологии?
- в Категория любви в жизни личности

### Литература:

В Психология и педагогика : учеб. пособие / О.В. Пастюк. — Москва : ИНФРА-М, 2018. — с с. <https://znanium.com/catalog/product/924013>



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

и Психология массовых коммуникаций : учебник / А.М. Руденко, А.В. Литвинова; под ред. А.М. Руденко. — Москва. : РИОР : ИНФРА-М, 2019. — 303 с.

<https://znanium.com/catalog/product/1048186>

и Общая психология : учеб. пособие / Г.С. Абрамова. — 2-е изд., стереотип. — М. : ИНФРА-М, 2018. — 496 с. <https://znanium.com/catalog/product/944631>

и Психология личности : учебник / П.С. Гуревич. — 2-е изд. — М. : ИНФРА-М, 2018. — 479 с. <https://znanium.com/catalog/product/968740>

и Христианская психология : учеб. пособие / С.В. Кондратьев, О.В. Кондратьева. — М. : ИНФРА-М, 2019. — 191 с. <https://znanium.com/catalog/product/982137>

и Возрастная психология : учеб. пособие / Б.Р. Мандель. — М. : Вузовский учебник : ИНФРА-М, 2019. — 352 с. <https://znanium.com/catalog/product/1002742>



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Практическое занятие 2. Основные модели и исследования личности в психологической науке**

Целью проводимого занятия является формирование знаний: микро и метафазы развития личности; понятие экзистенциального генезиса личности; проблемы темперамента; основные структуры бытия личности (воля, способности, мотивации).

**Форма проведения:** контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

**Вопросы к контрольной работе:**

- х Проблема личности в психоанализе, бихевиоризме и гуманистической психологии
- х Концепции личности в духовно ориентированной психологии.
- х Теория темперамента
- х Потребности и мотивы
- х Воля
- х Типология характеров
- х Теория темперамента
- х Потребности и мотивы
- х Психологические представления о воле человека
- х Учение о способностях
- в Познавательная сфера личности
- в Неординарность личности
- в Проблема самосознания

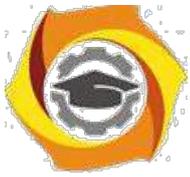
**Дополнительные вопросы для обсуждения и самопроверки:**

- 14 В чем особенность психологического учения о личности?
- 15 Каковы элементы теории темперамента?
- 16 Какие вы знаете основные потребности и мотивы в психике человека?
- 17 Что такое воля и в чем ее проявления?
- 18 Какая особенность в учении о способностях?
- 19 Какова роль я-концепции в профессиональной деятельности

**Литература:**

- 15. Психология и педагогика : учеб. пособие / О.В. Пастюк. — Москва : ИНФРА-М, 2018.
- 26. с. <https://znanium.com/catalog/product/924013>
- 58. Психология массовых коммуникаций : учебник / А.М. Руденко, А.В. Литвинова; под ред. А.М. Руденко. — Москва. : РИОР : ИНФРА-М, 2019. — 303 с. <https://znanium.com/catalog/product/1048186>
- 59. Общая психология : учеб. пособие / Г.С. Абрамова. — 2-е изд., стереотип. — М. : ИНФРА-М, 2018. — 496 с. <https://znanium.com/catalog/product/944631>
- 60. Психология личности : учебник / П.С. Гуревич. — 2-е изд. — М. : ИНФРА-М, 2018. — 479 с. <https://znanium.com/catalog/product/968740>
- 61. Христианская психология : учеб. пособие / С.В. Кондратьев, О.В. Кондратьева. — М. : ИНФРА-М, 2019. — 191 с. <https://znanium.com/catalog/product/982137>
- 62. Возрастная психология : учеб. пособие / Б.Р. Мандель. — М. : Вузовский учебник : ИНФРА-М, 2019. — 352 с. <https://znanium.com/catalog/product/1002742>

**Практическое занятие 3. Язык и сознание**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Целью проводимого занятия является формирование знаний: проблема сознания; связь мышления и языка; акты сознания, восприятия, памяти, представления; специфика способности внимания; темпоральность сознания; структура самосознания; понятие характера; врожденность и приобретенность характера; устойчивые формы поведения.

**Форма проведения:** контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

**Вопросы к контрольной работе:**

- Категории сознания в психологии
- Категория значения и сознание
- Гипотеза лингвистической относительности
- Проблема языка в психологии.
- Развитие индивидуального сознания
- Конституционные модели характеров
- Психологические типы К. Юнга
- Типологические модели акцентуаций характера
- Физиогномика: основные модели и интерпретации
- Система психологических способностей человека

**Дополнительные вопросы для обсуждения и самопроверки:**

- Изложите известные вам категории сознания в психологии?
- Какое место в психологии сознания занимает категория значения?
- О чем говорит гипотеза лингвистической относительности?
- Как осуществляется развитие индивидуального сознания?
- Какие вам известны конституционные модели характеров?
- Какие психологические типы отмечает в своем учении К. Юнг?
- В чем особенность типологических моделей акцентуаций характера?
- Как связаны между собой строение тела и характер?
- Какие виды характеров вам известны?

**Литература:**

и Психология и педагогика : учеб. пособие / О.В. Пастюк. — Москва : ИНФРА-М, 2018. — 496 с. <https://znanium.com/catalog/product/924013>

В Психология массовых коммуникаций : учебник / А.М. Руденко, А.В. Литвинова; под ред. А.М. Руденко. — Москва. : РИОР : ИНФРА-М, 2019. — 303 с. <https://znanium.com/catalog/product/1048186>

В Общая психология : учеб. пособие / Г.С. Абрамова. — 2-е изд., стереотип. — М. : ИНФРА-М, 2018. — 496 с. <https://znanium.com/catalog/product/944631>

В Психология личности : учебник / П.С. Гуревич. — 2-е изд. — М. : ИНФРА-М, 2018. — 479 с. <https://znanium.com/catalog/product/968740>

В Христианская психология : учеб. пособие / С.В. Кондратьев, О.В. Кондратьева. — М. : ИНФРА-М, 2019. — 191 с. <https://znanium.com/catalog/product/982137>

В Возрастная психология : учеб. пособие / Б.Р. Мандель. — М. : Вузовский учебник : ИНФРА-М, 2019. — 352 с. <https://znanium.com/catalog/product/1002742>

**Модуль II.**

**Практическое занятие 4. Психология коммуникации.**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Целью проводимого занятия является формирование знаний: типы общения (когнитивное, деятельное, мотивационное); средства и структура общения; самовосприятие и обратная связь; общение как взаимодействие.

**Форма проведения:** контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

**Вопросы к контрольной работе:**

34. Проблема другого в психологии.
35. Психологические законы общения, уровни общения.
36. Проблемы группового взаимодействия: дружба, семья, коллектив

**Дополнительные вопросы для обсуждения и самопроверки:**

30. Опишите психологические принципы межличностных отношений.
31. Как происходит социально-психологические влияния?
32. Обоснуйте законы психологического взаимодействия в группе?
33. Каким образом осуществляется общение личности и группы?
34. Что вам известно о проблеме лидерства в психологии?

**Литература:**

26) Психология и педагогика : учеб. пособие / О.В. Пастюк. — Москва : ИНФРА-М, 2018.

В с. <https://znanium.com/catalog/product/924013>

26. Психология массовых коммуникаций : учебник / А.М. Руденко, А.В. Литвинова; под ред. А.М. Руденко. — Москва. : РИОР : ИНФРА-М, 2019. — 303 с.  
<https://znanium.com/catalog/product/1048186>

27. Общая психология : учеб. пособие / Г.С. Абрамова. — 2-е изд., стереотип. — М. : ИНФРА-М, 2018. — 496 с. <https://znanium.com/catalog/product/944631>

28. Психология личности : учебник / П.С. Гуревич. — 2-е изд. — М. : ИНФРА-М, 2018. — 479 с. <https://znanium.com/catalog/product/968740>

29. Христианская психология : учеб. пособие / С.В. Кондратьев, О.В. Кондратьева. — М. : ИНФРА-М, 2019. — 191 с. <https://znanium.com/catalog/product/982137>

30. Возрастная психология : учеб. пособие / Б.Р. Мандель. — М. : Вузковский учебник : ИН-ФРА-М, 2019. — 352 с. <https://znanium.com/catalog/product/1002742>

**Практическое занятие 5. Социально-психологические феномены**

Целью проводимого занятия является формирование знаний: проблема направленности личности; проблема темперамента; уяснение понятий генотип и фенотип; понимание природы воли и свободы; представление о качествах решительности и самостоятельности.

**Форма проведения:** контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

**Вопросы к контрольной работе:**

- 89 Физическое развитие человека
- 90 Половое и нравственное развитие личности
- 91 Аномальное поведение и методы психологической помощи
- 92 Модель идеального специалиста-профессионала
- 93 Самопознание как важнейшая предпосылка самосовершенствования
- 94 Методы психофизического саморегулирования
- 95 Проблемы возрастной психологии.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

14. Проблема лидерства в психологии.
15. Психология конфликта.
16. Проблемы социально-психологических влияний на человека.
17. Аспекты социализация личности
18. Социальная зрелость личности

**Дополнительные вопросы для обсуждения и самопроверки:**

- / Что означает понятие «личность» в психологии?
- / Какое значение имеет социализация личности?
- / Опишите психологическую Я-концепцию личности?
- / Социальная зрелость личности
- / Как происходит психофизическое развитие личности?
- / Какие законы возрастания личности вам известны?
- / Проанализируйте этапы нравственного развития человека?
- / Поясните аномальное поведение и методы психологической помощи?

**Литература:**

- / Психология и педагогика : учеб. пособие / О.В. Пастюк. — Москва : ИНФРА-М, 2018. — с. <https://znanium.com/catalog/product/924013>
- / Психология массовых коммуникаций : учебник / А.М. Руденко, А.В. Литвинова; под ред. А.М. Руденко. — Москва. : РИОР : ИНФРА-М, 2019. — 303 с. <https://znanium.com/catalog/product/1048186>
- / Общая психология : учеб. пособие / Г.С. Абрамова. — 2-е изд., стереотип. — М. : ИНФРА-М, 2018. — 496 с. <https://znanium.com/catalog/product/944631>
- / Психология личности : учебник / П.С. Гуревич. — 2-е изд. — М. : ИНФРА-М, 2018. — 479 с. <https://znanium.com/catalog/product/968740>
- / Христианская психология : учеб. пособие / С.В. Кондратьев, О.В. Кондратьева. — М. : ИНФРА-М, 2019. — 191 с. <https://znanium.com/catalog/product/982137>
- / Возрастная психология : учеб. пособие / Б.Р. Мандель. — М. : Вузковский учебник : ИНФРА-М, 2019. — 352 с. <https://znanium.com/catalog/product/1002742>

**Практическое занятие 6. Теория образования и воспитания.**

Целью проводимого занятия является формирование знаний: уяснение природы образовательной системы; структура образования; менеджмент в образовании; история теории воспитания; принципы воспитания; проблемы семейного воспитания; типы личностного созревания и возрастания; психология педагогики; инновационные методы в образовании и его психологические условия и последствия.

**Форма проведения:** контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

**Вопросы к контрольной работе:**

- В Образование как способ вхождения человека в мир науки и культуры
- В Современное мировое образовательное пространство
- В Образовательный менеджмент
- В Система образования в современной России
- В Проблема качества образования
- В Сущность воспитательного процесса
- В Методы организации воспитательного процесса



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- \endash Воспитание в коллективе
- \endash Социальная природа семьи.
- \endash Особенность семейного воспитания.
- \endash Роль семьи в формировании навыков семейной жизни.
- \endash Проблемы инновационного обучения
- \endash Стили педагогического общения
- \endash Проблемное обучение и деловые игры

**Дополнительные вопросы для обсуждения и самопроверки:**

28. Представьте позитивную статистическую оценку системе образования в современной Рос-сии.
29. В чем смысл осуществления менеджментом качества образования?
30. Объясните сущность воспитательного процесса?
31. Какие вам известны методы организации воспитательного процесса?
32. Какие особенности имеет воспитание в коллективе?
33. Конфликт и методы его урегулирования?
34. В чем проявляет себя социальная природа семьи?
35. Какие вам известны стили семейного воспитания?
36. Какова роль семьи в формировании навыков семейной жизни?
37. Какие существуют стили педагогического общения?
38. Какова специфика проблемного обучения и деловых игр?
39. Охарактеризуйте систему образования как способ вхождения человека в мир науки и культуры.

**Литература:**

37. Психология и педагогика : учеб. пособие / О.В. Пастюк. — Москва : ИНФРА-М, 2018. —  
—  
28. с. <https://znanium.com/catalog/product/924013>
29. Психология массовых коммуникаций : учебник / А.М. Руденко, А.В. Литвинова; под ред. А.М. Руденко. — Москва. : РИОР : ИНФРА-М, 2019. — 303 с. <https://znanium.com/catalog/product/1048186>
30. Общая психология : учеб. пособие / Г.С. Абрамова. — 2-е изд., стереотип. — М. : ИНФРА-М, 2018. — 496 с. <https://znanium.com/catalog/product/944631>
31. Психология личности : учебник / П.С. Гуревич. — 2-е изд. — М. : ИНФРА-М, 2018. — 479 с. <https://znanium.com/catalog/product/968740>
32. Христианская психология : учеб. пособие / С.В. Кондратьев, О.В. Кондратьева. — М. : ИНФРА-М, 2019. — 191 с. <https://znanium.com/catalog/product/982137>
33. Возрастная психология : учеб. пособие / Б.Р. Мандель. — М. : Вузовский учебник : ИН-ФРА-М, 2019. — 352 с. <https://znanium.com/catalog/product/1002742>

**Методические указания к практическим занятиям заочной формы обучения**

**Практическое занятие 1. Специфика психологического познания**

Целью проводимого занятия является формирование знаний: основных направлений психологических исследований и методологии психологических исследований; особенности исторического развития понятий личности; специфические человеческие особенности и свойства (речь, сознание, трудовая деятельность); значение социально-культурного влияния на формирование личности.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Форма проведения:** контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

**Вопросы к контрольной работе:**

- № История развития психологии в древнем мире и новое время.
- № Психология в XIX веке.
- № Основные психологические направления и теории в XX веке.
- № Психоанализ
- № Бихевиоризм
- № Гуманистическая психология
- № Концепции личности в духовно ориентированной психологии
- № Личность как система отношений
- № Личность в системе межличностных связей

**Дополнительные вопросы для обсуждения и самопроверки:**

- 51. Специфика развития учения о душе в греческой философии
- 52. Понимание психологии христианскими мыслителями.
- 53. Психофизические вопросы философии Нового времени.
- 54. Основные понятия психоанализа З.Фрейда?
- 55. В чем специфика бихевиоризма?
- 56. Кто является апологетом гуманистической психологии?
- 57. Что можно отметить в концепции личности в духовно ориентированной психологии?
- 58. Категория любви в жизни личности

**Литература:**

- 43. Психология и педагогика : учеб. пособие / О.В. Пастюк. — Москва : ИНФРА-М, 2018.
- 41. с. <https://znanium.com/catalog/product/924013>
- 47. Психология массовых коммуникаций : учебник / А.М. Руденко, А.В. Литвинова; под ред. А.М. Руденко. — Москва. : РИОР : ИНФРА-М, 2019. — 303 с. <https://znanium.com/catalog/product/1048186>
- 48. Общая психология : учеб. пособие / Г.С. Абрамова. — 2-е изд., стереотип. — М. : ИНФРА-М, 2018. — 496 с. <https://znanium.com/catalog/product/944631>
- 49. Психология личности : учебник / П.С. Гуревич. — 2-е изд. — М. : ИНФРА-М, 2018. — 479 с. <https://znanium.com/catalog/product/968740>
- 50. Христианская психология : учеб. пособие / С.В. Кондратьев, О.В. Кондратьева. — М. : ИНФРА-М, 2019. — 191 с. <https://znanium.com/catalog/product/982137>
- 51. Возрастная психология : учеб. пособие / Б.Р. Мандель. — М. : Вузовский учебник : ИН-ФРА-М, 2019. — 352 с. <https://znanium.com/catalog/product/1002742>

**Практическое занятие 2. Основные модели и исследования личности в психологической науке**

Целью проводимого занятия является формирование знаний: микро и метафазы развития личности; понятие экзистенциального генезиса личности; проблемы темперамента; основные структуры бытия личности (воля, способности, мотивации).

**Форма проведения:** контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

**Вопросы к контрольной работе:**

В Проблема личности в психоанализе, бихевиоризме и гуманистической



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- ООО Концепции личности в духовно ориентированной психологии.
- ООО Теория темперамента
- ООО Потребности и мотивы
- ООО Воля
- ООО Типология характеров
- ООО Теория темперамента
- ООО Потребности и мотивы
- ООО Психологические представления о воле человека
- ООО Учение о способностях
- 17. Познавательная сфера личности
- 18. Неординарность личности
- 19. Проблема самосознания

**Дополнительные вопросы для обсуждения и самопроверки:**

- 36. В чем особенность психологического учения о личности?
- 37. Каковы элементы теории темперамента?
- 38. Какие вы знаете основные потребности и мотивы в психике человека?
- 39. Что такое воля и в чем ее проявления?
- 40. Какая особенность в учении о способностях?
- 41. Какова роль я-концепции в профессиональной деятельности

**Литература:**

- 32. Психология и педагогика : учеб. пособие / О.В. Пастюк. — Москва : ИНФРА-М, 2018.  
—  
к с. <https://znanium.com/catalog/product/924013>
- 40. Психология массовых коммуникаций : учебник / А.М. Руденко, А.В. Литвинова; под ред. А.М. Руденко. — Москва. : РИОР : ИНФРА-М, 2019. — 303 с.  
<https://znanium.com/catalog/product/1048186>
- 41. Общая психология : учеб. пособие / Г.С. Абрамова. — 2-е изд., стереотип. — М. : ИНФРА-М, 2018. — 496 с. <https://znanium.com/catalog/product/944631>
- 42. Психология личности : учебник / П.С. Гуревич. — 2-е изд. — М. : ИНФРА-М, 2018. — 479 с. <https://znanium.com/catalog/product/968740>
- 43. Христианская психология : учеб. пособие / С.В. Кондратьев, О.В. Кондратьева. — М. : ИНФРА-М, 2019. — 191 с. <https://znanium.com/catalog/product/982137>
- 44. Возрастная психология : учеб. пособие / Б.Р. Мандель. — М. : Вузовский учебник : ИН-ФРА-М, 2019. — 352 с. <https://znanium.com/catalog/product/1002742>

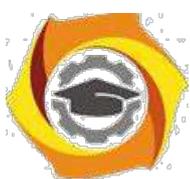
**Практическое занятие 3. Возрастная психология**

Целью проводимого занятия является формирование знаний: проблема воспитания; типы личностного созревания и возрастания; проблема воспитания и образования человека; психология педагогики.

**Форма проведения:** контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

**Вопросы к контрольной работе:**

- 13. Физическое развитие
- 14. Половое развитие
- 15. Нравственное развитие
- 16. Аномальное поведение и методы психологической помощи



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Дополнительные вопросы для обсуждения и самопроверки:**

14. Как происходит психофизическое развитие личности?
15. Какие законы возрастания личности вам известны?
16. Проанализируйте этапы нравственного развития человека?
17. Поясните аномальное поведение и методы психологической помощи?

**Литература:**

23. Психология и педагогика : учеб. пособие / О.В. Пастюк. — Москва : ИНФРА-М, 2018.
20. с. <https://znanium.com/catalog/product/924013>
23. Психология массовых коммуникаций : учебник / А.М. Руденко, А.В. Литвинова; под ред. А.М. Руденко. — Москва. : РИОР : ИНФРА-М, 2019. — 303 с. <https://znanium.com/catalog/product/1048186>
24. Общая психология : учеб. пособие / Г.С. Абрамова. — 2-е изд., стереотип. — М. : ИНФРА-М, 2018. — 496 с. <https://znanium.com/catalog/product/944631>
25. Психология личности : учебник / П.С. Гуревич. — 2-е изд. — М. : ИНФРА-М, 2018. — 479 с. <https://znanium.com/catalog/product/968740>
26. Христианская психология : учеб. пособие / С.В. Кондратьев, О.В. Кондратьева. — М. : ИНФРА-М, 2019. — 191 с. <https://znanium.com/catalog/product/982137>
27. Возрастная психология : учеб. пособие / Б.Р. Мандель. — М. : Вузовский учебник : ИН-ФРА-М, 2019. — 352 с. <https://znanium.com/catalog/product/1002742>

### **III. Методические рекомендации по курсу**

Курс по дисциплине «Психология и педагогика» рассчитан на изучение в течение одного семестра, включает лекционные, практические занятия и самостоятельную работу студентов. Для контроля уровня сформированности компетенций, качества знаний, умений и навыков особое внимание уделяется стимулированию самостоятельной работы студентов.

Содержание дисциплины «Психология и педагогика» разделено на два модуля, по окончании изучения которых осуществляется текущий контроль усвоения учебного материала. В течение семестра проводятся модульно-рейтинговые мероприятия, в том числе, и в форме компьютерного тестирования для проверки самостоятельной работы студентов.

Изучение данного предмета является трудоемкой задачей, в связи с чем, предусматривается обязательное посещение лекционных занятий для получения системы целостных знаний. Рекомендованные учебники и учебные пособия не всегда являются достаточными источниками знаний, так как, во-первых, учебный курс носит междисциплинарный характер, во-вторых, лекционный материал непосредственно дает квинтэссенцию ответов на экзаменационные вопросы.

После лекций проводятся практические занятия, они, прежде всего, предназначены для закрепления теоретических знаний и выработки таких практических навыков, как, аргументированное отстаивание собственного видения рассматриваемых проблем, владение приемами ведения дискуссии, полемики и диалога. Специфика курса заключается в том, что многие темы требуют дискуссионного обсуждения. На практические занятия выносятся материал, дополняющий и углубляющий лекционный, а также, материал, не вошедший в лекционный курс, поэтому при подготовке к практическим занятиям необходимо самостоятельное прочтение и осмысление рекомендуемых работ.

Темы курса рассчитаны на закрепление и систематизацию лекционного материала, самостоятельное освоение дополнительной литературы, а также, нацелены на формирование навыков



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

по практическому применению полученных знаний в сфере образования и общественной коммуникации.

## Руководство по подготовке курсовых работ (проектов) и выпускных квалификационных работ

### ВВЕДЕНИЕ

Заключительным этапом обучения студента в высшем учебном заведении является государственная итоговая аттестация, состоящая из государственного экзамена и защиты выпускной квалификационной работы.

Целью итоговой государственной итоговой аттестации является определение теоретической и практической подготовленности студента к решению профессиональных задач, установленных государственным образовательным стандартом высшего образования.

Выпускная квалификационная работа (ВКР) должна представлять собой законченную проектную, технологическую или научно-исследовательскую разработку по одному из актуальных теоретических или технических направлений и иметь расчетно-графическую или исследовательскую часть. Разделы по конструированию, безопасности жизнедеятельности, экологии и экономики не являются обязательными.

Допускается выполнение комплексных ВКР по одной теме группой студентов, каждый из которых разрабатывает самостоятельно определенную часть общей системы или устройства. При этом возможно выполнение комплексной темы студентами разных направлений подготовки, каждый из которых отвечает за разработку различных по характеру задач.

Реальные ВКР выполняются по заданию предприятий или выпускающей кафедры высшего учебного заведения. Результаты, выводы и рекомендации таких ВКР используются на конкретных предприятиях, в научных или методических разработках кафедры, в практической организации учебного процесса (например, в лабораторном практикуме) и др.

При выполнении ВКР необходимо соблюдать соответствующие стандарты, нормативы и требования.

#### 1 Общие положения

ТУ УГМК утверждает перечень тем выпускных квалификационных работ, предлагаемых обучающимся (далее – перечень тем), и доводит его до сведения обучающихся не позднее, чем за 6 месяцев до даты начала государственной итоговой аттестации.

Темы выпускных квалификационных работ и состав руководителей утверждается приказом по ТУ УГМК.

С учетом направлений профессиональной деятельности, изложенных в ГОСТе, руководитель ВКР вместе со студентом определяет тему ВКР. Студенту предоставляется право выбора темы выпускной квалификационной работы вплоть до предложения своей тематики с обоснованием целесообразности ее разработки. Возможные направления по выбору тематики выпускных квалификационных работ приведены в приложении А.

Если заведующий кафедрой сочтёт её соответствующей требованиям руководящих документов, то студент оформляет заявление на ВКР (приложение Б).

На основании заявлений студентов, завизированных руководителем, заведующим выпускающей кафедрой и деканом, составляется приказ о закреплении за студентами тем и руководителей ВКР в срок не позже даты начала подготовки к государственной итоговой аттестации, определенной планом-графиком учебного процесса.

Выпускная квалификационная работа должна продемонстрировать умение автора применять для решения задач и анализа полученных результатов знания, полученные в области общетехнических и специальных дисциплин,

Процесс и результаты выполнения выпускной квалификационной работы должны показать умение автора свободно обращаться с технической документацией — текстовыми материалами, схемами, программным обеспечением для этих объектов, а также получать новые знания посредством информационного поиска в научно-технической литературе и мировых информационных ресурсах.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

## 2 Требования к оформлению выпускной квалификационной работы

### 2.1 Структура пояснительной записки

Выпускная квалификационная работа состоит из пояснительной записки, приложений к ней, графического материала или его презентации, а также доклада об основных этапах работы и результатах, полученных при выполнении ВКР.

Пояснительная записка должна в краткой и четкой форме содержать постановку задачи (при создании программного продукта – описание предметной области). Кроме того, она должна раскрывать творческий замысел, содержать методы исследования, используемые технологии моделирования и программирования, принятые методы анализа и расчета, описание проведенных экспериментов и тестов, их анализы и выводы, технико-экономическое сравнение вариантов, а при необходимости сопровождаться иллюстрациями, графиками, эскизами, диаграммами, схемами и т.п. Минимальный объем пояснительной записки должен составлять 50 листов, максимальный 100 листов.

Пояснительная записка ВКР должна содержать следующие структурные элементы (приложение В), расположенные в следующем порядке:

- титульный лист;
- задание на выполнение ВКР;
- отзыв руководителя;
- аннотация;
- содержание;
- введение;
- основные разделы в соответствии с утвержденным заданием на ВКР;
- заключение;
- перечень сокращений (при необходимости);
- список использованных источников;
- приложения (при необходимости);
- акты внедрения в учебном процессе или во внешней организации (при наличии);
- свидетельства о регистрации программного продукта, патенты (при наличии);
- перечень графического материала, выносимого на защиту; – документ о результате проверки ВКР на наличие заимствований.

К ВКР прилагается рецензия, которая, в отличие от остальных документов, в пояснительную записку не вшивается.

Если в ходе выполнения ВКР разработано ПО, то к ВКР прилагается акт сдачи/приемки разработанного ПО (приложение Г). При внедрении программного продукта или других результатов ВКР оформляется акт внедрения, образцы которого для внедрения в ТУ УГМК и в сторонней организации приведены в приложении Д. При государственной регистрации программы – копия свидетельства о регистрации. Данные документы включаются в состав ВКР (при наличии).

### 2.2 Задание на ВКР

Задание составляется и подписывается студентом и руководителем ВКР. Затем задание утверждается заведующим кафедрой.

Непосредственное руководство ВКР студента осуществляет руководитель, который:

- оказывает студенту помощь в разработке календарного графика работы на весь период работы над ВКР;
- рекомендует студенту необходимую литературу, справочные и другие материалы по теме;
- регулярно по расписанию проводит консультации (не реже одного раза в неделю);
- проверяет ход выполнения ВКР;



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- дает письменный отзыв о работе студента.

Студент обязан регулярно посещать назначенные консультации. При пропуске консультаций без уважительной причины или при значительном отставании его работы от графика кафедра принимает по отношению к студенту соответствующие административные меры.

Бланк задания оформляется в 3-х экземплярах, один хранится на кафедре, второй в деканате, третий находится у студента. Графа с номером приказа, утверждающего тему, заполняется после его подписания.

Важно отметить, что после утверждения тем ВКР директором СКФ изменение темы **не допускается**.

Образец бланка «Задание на выпускную квалификационную работу» приведен в приложении В.

## 2.3 Аннотация

Аннотация для выпускной квалификационной работы не должна превышать 1 листа.

Аннотация должна содержать краткие сведения о содержании работы и итогах ее проведения.

В аннотации должны быть приведены сведения о количестве листов (страниц) пояснительной записки, количестве иллюстраций, таблиц, использованных источников, приложений, листов графического материала. Аннотация размещается перед «содержанием» и в рамку не помещается.

Слово «АННОТАЦИЯ» пишется заглавными буквами и размещается по центру строки. Образец аннотации приведен в приложении В.

## 2.4 Содержание

В пояснительной записке помещают содержание с указанием номеров страниц, с которых начинаются: введение, номера и наименование всех разделов, подразделов, заключение, перечень сокращений (при наличии), список использованных источников, наименования всех приложений (при наличии), перечень графического материала, выносимого на защиту. Сам графический материал размещается на следующих листах после Перечня графического материала, страницы которого не нумеруются и в содержание не включаются. Допускается не помещать в содержание заголовки 3-го уровня (пункты).

Первый лист содержания должен быть оформлен в рамке с чертежным штампом по форме 2, а последующие листы пояснительной записки в рамке с чертежным штампом по форме 2а.

Размеры и примеры заполнения граф в основной надписи для пояснительной записки и чертежей по ГОСТ 2.104 приведены в приложении Е.

Слово «СОДЕРЖАНИЕ» пишется заглавными буквами и размещается по центру строки (приложение В).

## 2.5 Введение

Задача введения — создать общее представление о решаемой задаче, предметной области, к которой она относится, а также дать общую оценку актуальности и значимости предполагаемой разработки.

В соответствии с этим, во введении пояснительной записки следует кратко:

а) оценить современное состояние и актуальную проблематику рассматриваемой предметной области;

б) в общих словах описать цель выполняемой работы, изложить в кратком и понятном виде неформальную постановку задачи; изложить историю вопроса;

в) провести краткий аналитический обзор доступных информационных источников по данной тематике и дать оценку значимости предполагаемой разработки;



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

г) на основе анализа общих тенденций развития предметной области подтвердить актуальность решаемой научно-технической задачи и предполагаемый научно-технический эффект разработки.

Слово «ВВЕДЕНИЕ» пишется заглавными буквами и размещается по центру строки.

## 2.6 Основные технические разделы пояснительной записки. Общие требования

ВКР должна быть написана хорошим научно-техническим языком. Это означает как соблюдение общих норм литературного языка и правил грамматики, так и учет особенностей научной речи: ее точности, однозначности терминологии, некоторых правил применения форм речи.

Основная часть пояснительной записки должна охватывать широкий круг вопросов, среди которых обязательными являются:

- а) строгая формальная постановка решаемой задачи;
- б) обоснование и подробное описание метода решения рассматриваемой проблемы;
- в) детальное описание проектных решений, реализующих метод решения;
- г) обоснование выбора конкретного инструментария, позволяющего эффективно решить сформулированную задачу;
- д) описание методики тестирования (моделирования) результирующего продукта и методов оценки его качества.

План изложения основной части пояснительной записки определяется особенностями конкретной темы. Поэтому одной из важных задач автора квалификационной работы является построение последовательности изложения, позволяющей наиболее логично описать существо проблемы и метода решения.

Наименование, содержание и очерёдность следования разделов проистекают из задания на ВКР.

В некоторых отдельных случаях, по согласованию с руководителем ВКР, наименование и порядок разделов основной части пояснительной записки может не полностью совпадать с заданием на ВКР.

Можно указать ряд требований, общих для всех выпускных квалификационных работ. В общем случае разделы основной части должны содержать:

- 1) обоснование научно-технической значимости разработки (конкретно по теме работы);
- 2) анализ научно-технического состояния разработок по теме работы: обзор и анализ существующих систем и обоснование необходимости разработки (аналитической, функциональной и реализационной). Определение необходимых эксплуатационных свойств разработки, определение требований к системе, выбор перспективных направлений;
- 3) аналитический обзор, на основе которого осуществляется выбор (построение) разрабатываемой инфокоммуникационной системы; 4) конструирование, расчет, моделирование или тестирование.

Основная часть пояснительной записки разделяется на разделы и подразделы (ГОСТ 2.105-95).

Разделы должны иметь порядковые номера в пределах всего документа, обозначенные арабскими цифрами **без точки**. Подразделы должны иметь нумерацию в пределах каждого раздела. Номер подраздела состоит из номеров раздела и подраздела, разделенных точкой. В конце номера подраздела точка не ставится.

Подразделы могут состоять из нескольких пунктов. Нумерация пунктов должна быть в пределах подраздела, и номер пункта должен состоять из номеров раздела, подраздела и пункта, разделенных точками.

Пункты при необходимости могут быть разбиты на подпункты, которые должны иметь порядковую нумерацию в пределах каждого пункта. Внутри пунктов или подпунктов могут быть приведены перечисления.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Подпункты в лист содержания не вносятся.

Перед каждой позицией перечисления следует ставить дефис или строчную букву, после которой ставится круглая скобка. После скобки текст перечисления начинается со строчной буквы. Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится круглая скобка, а запись производится со второго абзацного отступа, как показано в примере. **Использование маркеров (символов) в списках не допускается.** Пример

- a) \_\_\_\_\_;
- б) \_\_\_\_\_;
- 1) \_\_\_\_\_;
- 2) \_\_\_\_\_;
- в) \_\_\_\_\_.

Каждый раздел, подраздел, пункт, подпункт записывают с абзацного отступа.

В списках первая строка элемента списка выполняется с абзацного отступа, а остальные строки, без него. В сложных списках отступы выполняются с учетом удобства прочтения списка.

Все разделы и подразделы должны иметь заголовки. Заголовки пунктов допускается не отделять пустой строкой от последующего текста. Пункты могут не иметь заголовков.

Заголовки должны четко и кратко отражать содержание разделов, подразделов.

Заголовок раздела (подраздела) печатают, отделяя от номера пробелом, начиная с прописной буквы, не приводя точку в конце и не подчеркивая. При этом номер раздела (подраздела) печатают после абзацного отступа (ГОСТ 2.105-95). Заголовки разделов выделяют увеличенным размером шрифта (16 пт).

Переносы частей слов в заголовках не допускаются. Если заголовок состоит из двух и более предложений, их разделяют точкой.

В заголовках следует избегать сокращений (за исключением общепризнанных аббревиатур, единиц величин и сокращений, входящих в условные обозначения).

При переносе текста на следующую страницу после наименования подраздела (пункта) рекомендуется записать не менее двух строк.

В выпускной квалификационной работе каждый раздел должен начинаться с нового листа.

## 2.7 Заключение

В разделе «Заключение» должен содержаться анализ результатов выполненной работы, оценка полноты решения поставленных задач, рекомендации по конкретному использованию результатов работы, ее экономическая, научная, социальная значимость.

В разделе «Заключение» описываются уже свершившиеся факты: «проведено исследование», «спроектировано», «разработано».

Существенно, чтобы словесные утверждения, касающиеся чисто качественных результатов, иллюстрировались количественными показателями, их сравнением с соответствующими показателями известных объектов и систем.

Заголовок «ЗАКЛЮЧЕНИЕ» записывают прописными буквами по центру строки.

## 2.8 Перечень сокращений

Перечень сокращений содержит перечень непредусмотренных стандартами сокращений слов и терминов, если их в тексте более двадцати и каждое из них повторяется не менее трех раз. В остальных случаях достаточно их расшифровки непосредственно в тексте в скобках при первом упоминании.

Перечень располагается столбцом, в котором слева в алфавитном порядке приводятся сокращения, справа его детальная расшифровка. Например:

- ВОК – волоконно-оптический кабель;
- ОВ – оптическое волокно;



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

СЦИ – синхронная цифровая иерархия; EDF

– оптическое волокно; MUX –

мультиплексор.

Заголовок «ПЕРЕЧЕНЬ СОКРАЩЕНИЙ» записывают прописными буквами симметрично тексту.

## 2.9 Список использованных источников

Сведения об источниках приводят в соответствии с ГОСТ Р 7.0.5–2008 СИБИД, ГОСТ 7.1–2003 СИБИД.

Список должен содержать все источники, использованные при выполнении работы (проекта) - пособия, справочники, каталоги, прейскуранты, стандарты, инструкции, альбомы, статьи, законы, электронные ресурсы, включая документы *WWW*. Источники следует располагать в порядке появления ссылок на них в тексте пояснительной записки. Источники в списке нумеруют в порядке упоминания в тексте пояснительной записки арабскими цифрами без точки. На весь заимствованный материал (формулы, текст, таблицы и др.) ссылки обязательны. Ссылки на источник информации указывают в тексте порядковым номером по списку источников, выделенным двумя квадратными скобками, например, [1].

Заголовок «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ» записывают прописными буквами по центру строки.

Примеры оформления списка приведены в приложении Ж.

## 2.10 Приложения

Если в тексте ПЗ есть приложения (плакаты, таблицы, распечатки программ, копии научных публикаций студента и т.д.), то на них дают ссылки в тексте, а в «Содержании» перечисляют все приложения с указанием их обозначений и заголовков.

В тексте пояснительной записки на все приложения должны быть даны ссылки. Приложения располагают в порядке ссылок на них в тексте пояснительной записки. Каждое приложение следует начинать с нового листа с указанием наверху посередине строки слова «Приложение», записанное обычным шрифтом, строчными буквами с первой прописной. Приложение должно иметь содержательный заголовок, который записывают симметрично относительно текста с прописной буквы отдельной строкой обычным шрифтом. Под ними в скобках может быть указан статус приложения, используя слова: «обязательное», «рекомендуемое» или «справочное».

Допускается размещение на одной странице двух (и более) последовательно расположенных приложений, если их можно полностью изложить на одной странице.

Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением Ё, З, Й, О, Ч, Я, Ъ, Ы.

Если в пояснительной записке ВКР одно приложение, оно обозначается «Приложение А».

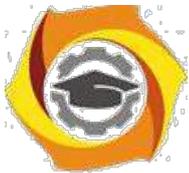
Текст каждого приложения, при необходимости, может быть разделен на разделы, подразделы, пункты и подпункты, которые нумеруют в пределах каждого приложения. Перед номером ставится обозначение этого приложения (например, А 2.1 – первый подраздел второго раздела приложения А).

Рисунки, таблицы, формулы, помещаемые в приложении, обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения, например,

... Рисунок А.3, формула (В.1), таблица В.1.

Приложения должны иметь общую с остальной частью пояснительной записки сквозную нумерацию страниц.

В случае разработки программного продукта для направления подготовки «Информатика и вычислительная техника» (09.03.01) по профилям «Вычислительные машины, комплексы,



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

системы и сети», «Программное обеспечение и интеллектуальные системы» при необходимости, по указанию руководителя ВКР может быть составлено «Техническое задание на создание программного продукта».

Цель технического задания — определить структуру, содержание и технические характеристики разрабатываемого программного или информационного продукта таким образом, чтобы заказчик и исполнитель имели одинаковое понимание того, каким должен быть результат разработки и какие функции он должен выполнять.

В пояснительной записке к выпускной квалификационной работе техническое задание оформляется как приложение.

Требования к содержанию и оформлению технического задания определяются ГОСТ 19.201-78 ЕСПД.

В общем случае техническое задание должно содержать следующие разделы.

*Введение.*

Указывается наименование, краткая характеристика области применения разрабатываемого программы или программного изделия и объекта.

*Основания для разработки.* Здесь

должны быть указаны:

- а) документ (документы), на основании которых ведется разработка;
- б) организация, утвердившая этот документ, и дата его утверждения. *Назначение разработки.*

Должно быть указано функциональное и эксплуатационное назначение программы или программного изделия.

Требования к программе или программному изделию.

Раздел должен содержать следующие подразделы:

- требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т.п.;
- требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования и программным средствам, используемым программой
- условия эксплуатации;
- требования к составу и параметрам технических средств;
- требования к информационной и программной совместимости;
- требования к обеспечению надежного функционирования (обеспечения устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа и т.п.); – специальные требования.

*Требования к программной документации.*

Приводится предварительный состав программной документации и, при необходимости, специальные требования к ней.

*Технико-экономические показатели.*

В разделе «Технико-экономические показатели» должны быть указаны: ориентировочная экономическая эффективность, предполагаемая годовая потребность, экономические преимущества разработки по сравнению с лучшими отечественными и зарубежными образцами или аналогами. *Стадии и этапы разработки.*

Описываются необходимые стадии разработки, этапы и содержание работ, а также, сроки разработки и исполнители.

*Порядок контроля и приемки.*

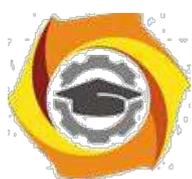
Указываются виды испытаний и общие требования к приемке работы.

В техническое задание *допускается* включать приложения.

В приложения к техническому заданию могут входить:

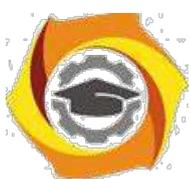
- перечни научно-исследовательских и других работ, обосновывающих разработку;
- схемы алгоритмов, таблицы, описания, обоснования, расчеты и другие документы, которые могут быть использованы при разработке; – другие источники разработки.

В зависимости от особенностей разрабатываемой системы, допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Образец технического задания приведен в приложении И.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

### 3 Нормативные документы, используемые при выполнении выпускных квалификационных работ

При подготовке ВКР необходимыми основными стандартами являются следующие:

- ГОСТ Р 1.5–2004 Стандарты национальные Российской Федерации. Правила построения, изложения, оформления и обозначения;
- ГОСТ 1.5–2001 Стандарты межгосударственные, правила и рекомендации по межгосударственной стандартизации. Общие требования к построению, изложению, оформлению, содержанию и обозначению;
- ГОСТ 2.004-88 ЕСКД. Общие требования к выполнению конструкторских и технологических документов на печатающих и графических устройствах ЭВМ.
- ГОСТ 2.105–95 ЕСКД. Общие требования к текстовым документам;
- ГОСТ 2.106–96 ЕСКД. Текстовые документы;
- ГОСТ 7.9–95 СИБИД. Реферат и аннотация. Общие требования.
- ГОСТ 19.404–79 ЕСПД. Пояснительная записка. Требования к содержанию и оформлению;
- ГОСТ 21.101–97 СПДС. Основные требования к проектной и рабочей документации;
- ГОСТ 24.301–80 Система технической документации на АСУ. Общие требования к текстовым документам;
- ГОСТ 7.32–2001 (ИСО 5966–82) Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления;
- ГОСТ 2.102–68 ЕСКД. Виды и комплектность конструкторских документов;
- ГОСТ 19.202-78 ЕСПД. Спецификация. Требования к содержанию и оформлению.
- ГОСТ 2.104–2006 ЕСКД. Основные надписи;
- ГОСТ 2.109–73 ЕСКД. Основные требования к чертежам;
- ГОСТ 2.301–68 ЕСКД. Форматы;
- ГОСТ 2.304–81 ЕСКД. Шрифты чертежные;
- ГОСТ 2.316–68 ЕСКД. Правила нанесения на чертежах надписей, технических требований и таблиц;
- ГОСТ 2.321–84 ЕСКД. Обозначения буквенные;
- ГОСТ 2.605–68 ЕСКД. Плакаты учебно-технические. Общие технические требования;
- ГОСТ 2.701–84 ЕСКД. Схемы Виды и типы. Общие требования к выполнению;
- ГОСТ 2.710–81 ЕСКД. Обозначения буквенно-цифровые в электрических схемах;
- ГОСТ 2.721–74 ЕСКД. Обозначения условные графические в схемах. Обозначения общего применения;
- ГОСТ 3.1119–83 ЕСТД. Общие требования к комплектности и оформлению комплектов документов на единичные технологические процессы;
- ГОСТ 3.1120–83 ЕСТД. Общие правила отражения и оформления требований безопасности труда в технологической документации;
- ГОСТ Р 7.0.5–2008 СИБИД. Библиографическая ссылка. Общие требования и правила составления
- ГОСТ 7.1–2003 СИБИД. Библиографическая запись. Библиографическое описание. Общие требования и правила составления;
- ГОСТ 7.12.93 СИБИД. Библиографическая запись. Сокращение слов на русском языке;
- ГОСТ 8.417–2002 Государственная система обеспечения единства измерений. Единицы величин;
- Р 50–77–88 ЕСКД. Правила выполнения диаграмм.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Программные документы, и в их числе – программа, оформляются в соответствии со следующими нормативными документами Единой системы программной документации (ЕСПД):

ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.

ГОСТ 19.301-79 ЕСПД. Порядок и методика испытаний.

ГОСТ 19.401-78 ЕСПД. Текст программы. Требования к содержанию и оформлению.

ГОСТ 19.503-79 ЕСПД. Руководство системного программиста.

Требования к содержанию и оформлению.

ГОСТ 19.505-79 ЕСПД. Руководство оператора.

ГОСТ 19.508-79 ЕСПД. Руководство по техническому обслуживанию.

Требования к содержанию и оформлению.

ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем.

Условные обозначения и правила выполнения.

## 4 Требования к оформлению текста пояснительной записки

### 4.1 Общие требования

Текст пояснительной записки (ПЗ) должен быть набран с использованием компьютера (ПК), распечатан на бумаге формата А4 на одной стороне листа. Текст, включая все заголовки, печатается через полтора интервала. Используется шрифт «*Times New Roman*», размер шрифта – 14 пт.

Расстояние от верхней или нижней строки текста до верхней или нижней рамки формы должно быть не менее 5 мм. Абзацы в тексте начинают отступом, равным 1,25 см.

Расстояние между заголовком и предыдущим текстом должно быть равно утроенному межстрочному расстоянию, заголовком и следующим за ним текстом должно быть равно удвоенному межстрочному расстоянию, между заголовками раздела и подраздела - одному межстрочному расстоянию.

**Буквы русского и греческого алфавитов, а также цифры набираются прямым шрифтом. Буквы латинского алфавита могут быть набраны или курсивом или прямым шрифтом, но обязательно однообразно по всему тексту пояснительной записки ВКР.** При наборе текста в таблицах возможно уменьшение размера букв шрифта до 10-12-го.

Математические символы  $lg$ ,  $const$ ,  $min$ ,  $max$  и т.д. набираются прямым шрифтом. Между цифровыми значениями величины и ее размерностью следует ставить знак неразрывного пробела. Для расстановки переносов использовать «автоматическую расстановку переносов».

При оформлении выпускных квалификационных работ необходимо соблюдать следующие размеры полей на листе:

- правое – 10 мм;
- левое – 30 мм;
- верхнее – 20 мм; –
- нижнее – 20 мм.

Нумерация страниц пояснительной записки и приложений, входящих в состав записки, должна быть *сквозная*.

Опечатки, описки и графические неточности, обнаруженные в процессе выполнения документа, допускается исправлять подчисткой или закрашиванием белой краской и нанесением на том же месте исправленного текста (графики) машинописным способом или от руки черными чернилами (пастой).

Счёт листов ведётся с «Титульного листа» пояснительной записки, но номера проставляются, начиная с листа «Содержание». Например, если титульный лист, задание, отзыв и аннотация занимают, соответственно, по одной странице, то страница содержания будет иметь номер 5.

«Основная надпись» (приложение Е, рисунок Е.1) наносится только на листе «Содержание» (см. приложение В). Все наименования, включенные в содержание, пишутся строчными буквами, кроме первой прописной.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Содержание включает наименование всех разделов, подразделов и пунктов с указанием номеров страниц, на которых размещается начало материала разделов (подразделов, пунктов). С целью уменьшения объема «СОДЕРЖАНИЯ», по желанию автора, пункты могут в него не включаться. Все разделы начинаются с нового листа.

## 4.2 Стиль изложения

Текст пояснительной записки должен быть лаконичным, четким, формальным и не допускающим различных толкований.

Изложение ведётся в безличной форме. Не допускаются выражения от первого лица (типа «Я решил...», «Я применил...» и т.п.). Следует писать: «Исходя из приведенных обоснований, принято решение...», «...Для достижения этой цели применено...» и т.д.

При изложении обязательных требований в тексте должны применяться «категорические» слова "должен", "следует", "необходимо", "требуется, чтобы", "разрешается только", "не допускается", "запрещается", "не следует" и т.д.

При изложении других положений следует применять слова – "могут быть", "как правило", "при необходимости", "может быть", "в случае, если" и т.д.

Допускается использовать (например, при анализе состояния вопроса) повествовательную форму изложения текста, например, "применяют", "указывают", "считают более предпочтительным", "нет единой точки зрения", "считается доказанным" и т. п.

В пояснительной записке должны правильно применяться научнотехнические термины, обозначения и определения, установленные соответствующими стандартами, а при их отсутствии – общепринятые в научно-технической литературе.

В тексте пояснительной записки не допускается:

- применять для одного и того же понятия различные научно-технические термины, близкие по смыслу (синонимы), а также иностранные слова и термины при наличии равнозначных слов и терминов в русском языке;
- сокращать обозначения единиц физических величин, если они употребляются без цифр, за исключением единиц физических величин в таблицах, и в расшифровках буквенных обозначений, входящих в формулы и рисунки;
- применять сокращения слов, кроме установленных правилами русской орфографии, а также соответствующими стандартами.

В тексте пояснительной записки, за исключением формул, таблиц и рисунков, не допускается:

- применять математический знак минус (-) перед отрицательными значениями (следует писать слово "минус");
- применять без числовых значений математические знаки, например: > (больше); < (меньше); = (равно); ≥ (больше или равно); ≠ (не равно); < (меньше или равно), а также знаки: % (процент), № (номер).

В пояснительной записке необходимо применять стандартизованные единицы физических величин, их наименования и обозначения в соответствии с ГОСТ 8.417-81 ГСИ.

Применение в тексте пояснительной записки разных систем обозначения единиц физических величин не допускается. Наряду с единицами системы СИ, при необходимости в скобках указывают единицы ранее применявшихся систем, разрешенных к применению.

Единица физической величины одного и того же параметра в тексте должна быть постоянной.

Числовые значения величин в тексте должны указываться с требуемой точностью.

Если приводится ряд числовых значений, выраженных в одной и той же единице физической величины, то её указывают только после последнего числового значения, при этом в ряду величин осуществляется выравнивание числа знаков после запятой, например: 1,50; 1,75; 2,00 м.

При указании диапазона числовых значений физической величины, обозначение единицы физической величины указывается после последнего числового значения диапазона.

Примеры:



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- от 1 до 5 мм;
- от 10 до 100 кг;
- от плюс 0 до минус 40<sup>0</sup> С.

Недопустимо отделять единицу физической величины от числового значения (переносить их на разные строки или страницы).

## 4.3 Оформление формул

В формулах в качестве символов следует применять обозначения, установленные соответствующими государственными стандартами. Желательно для набора формул использовать встроенный в текстовый редактор «Word» редактор формул «Equations» или отдельную программу, встраиваемую в текстовый редактор, например, «MathType». В формулах буквы русского и греческого алфавита и цифры пишутся прямым шрифтом, а буквы латинского алфавита – курсивом. Размер основного шрифта в формулах должен быть, как и во всем тексте пояснительной записки, равен 14. Пояснения символов и числовых коэффициентов в формулах, если они не пояснены ранее, должны быть приведены непосредственно под формулой. Пояснение каждого символа даётся с новой строки в той последовательности, в которой символы приведены в формуле. Первая строка пояснения начинается со слова «где» без двоеточия после него.

Пример 1. Плотность образца,  $\rho$  кг/м<sup>3</sup>, вычисляются по формуле:

$$\rho = m / V, \quad (1)$$

где  $m$  – масса образца, кг,  $V$  – объем образца, м<sup>3</sup>.

Формулы, следующие одна за другой и не разделенные текстом, разделяют запятой.

Выравнивание нумерации формул должно производиться по правому краю страницы.

Пример 2. Реакция фильтра нижних частот соответственно на положительный и отрицательный скачки напряжения на его входе описывается соотношениями:

$$U_{\text{вых}}(t) = U_{\text{вх}}(1 - e^{-t/RC}), \quad (2)$$

$$U_{\text{вых}}(t) = U_{\text{вх}}e^{-t/RC}, \quad (3)$$

где  $t$  – время, с,

$U_{\text{вх}}$  – изменение (скачок) напряжения на входе фильтра,  $B$ ,  $R$  – сопротивление, Ом,  $C$  – емкость, Ф.

Переносить формулы на следующую строку допускается только на знаках выполняемых операций, причем знак в начале следующей строки повторяют. При переносе формулы на знаке умножения применяют знак умножения в виде «х».

Формулы нумеруются сквозной нумерацией арабскими цифрами, которые записываются на уровне формулы справа в конце строки в круглых скобках. Одну формулу (единственную) тоже обозначают – (1).

Ссылки в тексте на формулы дают по их порядковым номерам, указываемым в скобках, например, в формуле (1).

Допускается нумерация формул в пределах раздела. В этом случае номер формулы состоит из номера раздела и порядкового номера формулы, разделенных точкой, например, (3.1).

Для формул, помещаемых в приложениях, должна применяться отдельная нумерация арабскими цифрами в пределах каждого приложения.

Математические уравнения в тексте размещаются в таком же порядке, как и формулы.

## 4.4 Оформление иллюстраций

Количество иллюстраций должно быть достаточным для пояснения излагаемого текста. Иллюстрации выполняются в соответствии ЕСКД и располагаются, возможно, ближе к соответствующим частям текста.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Все иллюстрации (чертежи, графики, диаграммы, схемы, компьютерные распечатки, фотографии), включаемые в текстовый материал записки, именуются рисунками.

Иллюстрации могут нумероваться арабскими цифрами сквозной нумерацией. В этом случае первый по тексту рисунок обозначается, как Рисунок 1, а последний, например одиннадцатый рисунок, как Рисунок 11.

Также допускается нумеровать иллюстрации в пределах раздела. В этом случае номер иллюстрации состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой, например, Рисунок 1.1.

При ссылках на иллюстрации следует писать: «...в соответствии с рисунком 2» при их сквозной нумерации и «... в соответствии с рисунком 1.2» при нумерации в пределах раздела. Сокращения «рис.» не допускаются.

Иллюстрации, при необходимости, могут иметь наименование и пояснительные данные (подрисуночный текст). Точка после подрисуночного текста не ставится. Слово «Рисунок» и наименование помещают после рисунка и пояснительных данных (если они есть), выравниваются по центру и располагают следующим образом:

## Рисунок 1.1– Алгоритм работы программы

Если в пояснительной записке используются подрисуночные подписи, то они должны быть под каждым рисунком в ПЗ или везде отсутствовать. **Ссылки по тексту ПЗ на каждый рисунок обязательны.**

Рисунки, как правило, размещаются на той же или на следующей странице, где размещена ссылка на данный рисунок.

Допускается размещение рисунка в альбомном расположении страницы. В этом случае подрисуночная надпись должна быть ориентирована таким образом, чтобы читалась при повороте раскрытой книги ПЗ по часовой стрелке на 90 градусов. При этом ориентация рамки страницы не изменяется.

## 4.5 Оформление таблиц

Таблицу, в зависимости от ее размера, помещают под текстом, в котором впервые дана ссылка на нее, или на следующей странице, а, при необходимости, в приложении к документу. Начало и конец таблицы отделяются от текста одним пробелом.

Название таблицы должно отражать её содержание, быть точным, кратким.

Таблица \_\_\_\_\_ - \_\_\_\_\_  
(номер) (название таблицы)

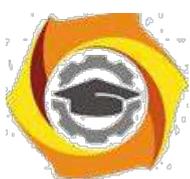
Объект.....	Изделие.....	Измерение...	Отклонение..	Примечание..
1	2	3	4	5

Название помещают над таблицей (выравнивание по левому краю с абзацным отступом, как и по тексту ПЗ), начинают с заглавной буквы, не подчеркивают.

Таблицы можно нумеровать арабскими цифрами сквозной нумерацией. Если таблица одна, то она должна быть обозначена "Таблица 1".

Также допускается нумеровать таблицы в пределах раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы, разделенных точкой. В тексте пояснительной записки на все таблицы должны быть приведены ссылки, при ссылке следует писать слово "таблица" с указанием её номера. Таблицы оформляют в соответствии с рисунком 1. **Ссылки на каждую таблицу обязательны.**

Заголовки граф и строк таблицы следует писать с прописной буквы, а подзаголовки граф – со строчной буквы, если они составляют одно предложение с заголовком, или с прописной буквы, если они имеют самостоятельное значение. В конце заголовков и подзаголовков таблиц



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

точки не ставят. Заголовки и подзаголовки граф указывают в единственном числе. Таблицы следует ограничивать линиями слева, справа и снизу.

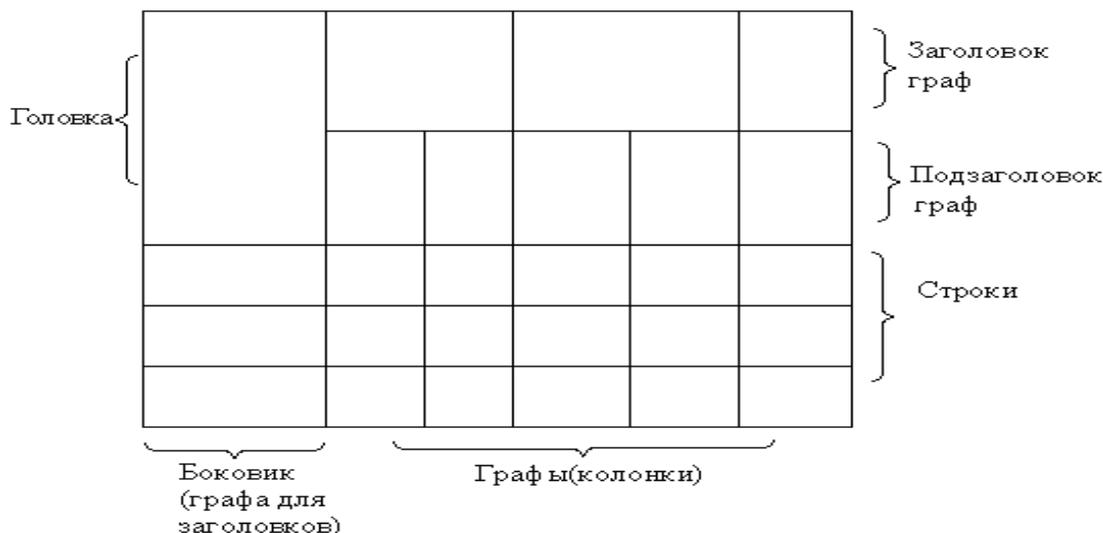


Рисунок 1 – Пример оформления таблицы

Разделять заголовки и подзаголовки боковика и граф диагональными линиями не допускается.

Горизонтальные и вертикальные линии, разграничивающие строки таблицы, допускается не проводить. Заголовки граф записывают параллельно строкам таблицы. Допускается перпендикулярное расположение заголовка граф.

Заголовок таблицы должен быть отделен линией от остальной части таблицы. Высота строк таблицы должна быть не менее 8 мм.

*Графу "номер по порядку" в таблицу включать не допускается.*

**(ГОСТ 2.105-95).**

Таблицу, в зависимости от её размера, помещают непосредственно после первой ссылки на неё или (при отсутствии места) сразу на следующей странице.

Допускается помещать таблицу вдоль длинной стороны листа пояснительной записки (т.е. вертикально).

Таблицу с большим количеством строк допускается переносить на другой лист, при этом в первой части таблицы нижнюю горизонтальную линию, ограничивающую таблицу, не проводят. При переносе продолжения таблицы на другую сторону листа заголовок помещают только над её первой частью.

Таблица 1.1-Контроль.....

Объект ....	Изделие.....	Измерение...	Отклонение..	Примечание..
1	2	3	4	5

Продолжение таблицы 1.1

1	2	3	4	5

Слово «Таблица» указывают один раз слева над первой частью таблицы; над другими её частями, если они переносятся на следующую страницу, пишут слова «Продолжение таблицы» с указанием номера (обозначения) таблицы, как показано ниже.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Если повторяющийся в графе таблицы текст состоит только из одного слова, его допускается заменять кавычками; если он состоит из двух и более слов, при первом повторении его заменяют словами «То же», а далее кавычками. Если предыдущая фраза является частью последующей, то допускается заменить её словами «То же» и добавить дополнительные сведения. При наличии в таблице горизонтальных линий текст необходимо повторять.

Не допускается заменять кавычками повторяющиеся в таблице цифры, математические знаки, знаки процента и номера, типоразмеров изделий, обозначения нормативных документов.

Если цифровые или иные данные в какой-либо строке таблицы не приводятся, то в ней ставят прочерк (тире).

При указании в таблице последовательных интервалов чисел, охватывающих все числа ряда, их следует записывать «От...до...включ.».

В интервале, охватывающем числа ряда, между крайними числами ряда в таблице допускается ставить тире.

Цифры в графах таблиц должны проставляться так, чтобы разряды чисел во всей графе были расположены один под другим, если они относятся к одному показателю. При этом количество десятичных знаков для всех значений должно быть одинаково.

При необходимости пояснения отдельных данных, приведенных в таблице, эти данные следует обозначать надстрочными знаками сноски.

Знак сноски ставят непосредственно после того слова, числа или символа, к которому даётся пояснение, и такой же знак ставят перед текстом пояснения к этой сноске.

Знак сноски выполняют арабскими цифрами со скобкой и помещают на уровне верхнего обреза шрифта. Нумерация сносок – отдельная для каждой таблицы.

***В случае необходимости допускается в таблицах уменьшать размер шрифта до 12 или 10 пунктов, а также сокращать межстрочный интервал с полуторного до одинарного.***

Допускается вместо цифр выполнять сноски звёздочками. При этом не рекомендуется применять более четырех звёздочек. Сноски располагают в конце таблицы с абзацного отступа.

При наличии в пояснительной записке цифрового материала небольшого по объёму, его следует давать не таблицей, а текстом, располагая цифровые данные в виде колонок.

## Пример.

Конфигурация рекомендуемого компьютера:

*CPU*.....2-2,5 ГГц; *RAM*..... 1-2 Гб;

*HDD* ..... >250 Гб.



## 5 Оформление графического материала, выносимого на защиту

### 5.1 Общие требования

Графический материал предназначен для демонстрации полученных результатов в процессе ВКР. Кроме того его умелое составление и использование позволит структурировать доклад и окажет значительную помощь студенту.

Графический материал должен отвечать требованиям действующих стандартов по соответствующему направлению науки, техники или технологии и может выполняться:

- неавтоматизированным методом – карандашом, пастой, чернилами или тушью;
- автоматизированным методом – с применением проекционных, графических и печатающих устройств вывода ЭВМ.

Графический материал, предназначенный для демонстрации при публичной защите работы оформляется в электронном виде с помощью программ подготовки презентаций (*MS Power Point* или *Open Office Impress*).

Каждый графический конструкторский документ (чертеж), выполненный в виде самостоятельного документа, должен иметь рамку и в правом нижнем углу листа основную надпись по ГОСТ 2.104.

Количество необходимых графических материалов, выносимых на защиту, согласовывается с руководителем ВКР. Количество должно рассчитываться так, чтобы все основные положения, разработанные в проекте, были отражены (обычно от 5 до 12 чертежей и(или) плакатов).

Плакаты оформляются без рамок с заголовками, расположенными по центру строки, а чертежи в рамках с основной надписью по форме 2 (см. приложение Е).

Весь графический материал, выносимый на защиту оформляется на листах формата А4 и подшивается в конце пояснительной записки в порядке, приведенном в подразделе 2.1.

### 5.2 Требования к оформлению презентаций

В электронном виде графический материал представляется на защиту только в формате презентаций (выполненный с помощью программ *MS Power Point* или *Open Office Impress*). Иллюстративный материал, выполненный в форматах *doc*, *docx*, *pdf*, *cdr* и др. к защите не принимается.

Как правило, в зависимости от специфики кафедры, применяются различные подходы:

1. Для направления подготовки 09.03.01 «Информатика и вычислительная техника» минимальное количество: первый слайд - тема, цель и постановка задачи ВКР, исходные данные, 2-3 слайда с алгоритмами работы определенных программных модулей, 1-2 слайда - с интерфейсом разработанного программного продукта, 1-2 слайда - с основным фрагментом листинга программы.

2. Для направления подготовки 11.03.02 «Инфокоммуникационные технологии и системы связи»: первый слайд – тема, цель и постановка задачи ВКР, исходные данные, 2-3 слайда - анализ используемых и перспективных технологий, 2-3 слайда, включающие структурные, функциональные, конструкционные или принципиальные схемы и рисунки, формулы, применяемые для решения, 1-2 слайда показывающие результаты, полученные в данной предметной области.

При создании презентации необходимо соблюдать следующие правила:

- наличие титульного листа с указанием темы ВКР, исполнителя ВКР и руководителя с его ученой степенью и званием;
- количество слайдов: не менее 5 и не более 12;
- слайд должен содержать минимально возможное количество слов.
- текст должен быть легко читаем. Рекомендуется для заголовков – размер шрифта не менее 24, для информации – не менее 18. Тип шрифта «Arial» или «Times New Roman»;



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- фон слайда и текст должны иметь высокую контрастность;
- текст, символы, обозначения и т.д. на чертежах, рисунках, фотографиях и другие иллюстрационных материалах должны, разборчиво читаться на расстоянии просмотра;
- не размещать на слайде рисунков и фотографий не несущих информации о решаемых в ВКР задачах, например, внешний вид стойки мультиплексора или вид антенны станции РРЛ;
- не выносить на слайды рамки и основную надпись чертежей (при их наличии).

## 5.3 Схемное описание алгоритмов

При схемном описании алгоритм изображается геометрическими фигурами (блоками), связанными по управлению линиями (направлениями потока) со стрелками. В блоках записывается последовательность действий.

Данный способ по сравнению с другими способами записи алгоритма имеет ряд преимуществ. Он наиболее нагляден: каждая операция вычислительного процесса изображается отдельной геометрической фигурой. Кроме того, графическое изображение алгоритма наглядно показывает разветвления путей решения задачи в зависимости от различных условий, повторение отдельных этапов вычислительного процесса и другие детали.

Оформление программ должно соответствовать определенным требованиям. В настоящее время действует единая система программной документации (ЕСПД), которая устанавливает правила разработки, оформления программ и программной документации. В ЕСПД определены и правила оформления блок-схем алгоритмов (ГОСТ 10.002-80, ГОСТ 10.003-80 ЕСПД).

Схему алгоритма следует выполнять как единое целое, однако в случае необходимости допускается обрывать линии, соединяющие блоки.

Если при обрыве линии продолжение схемы находится на этом же листе, то на одном и другом конце линии изображается специальный символ соединитель — окружность диаметром 0,5 мм. Внутри парных окружностей указывается один и тот же идентификатор. В качестве идентификатора, как правило, используется порядковый номер блока, к которому направлена соединительная линия. Если схема занимает более одного листа, то в случае разрыва линии вместо окружности используется межстраничный соединитель. Внутри каждого соединителя указывается адрес — откуда и куда направлена соединительная линия. Адрес записывается в две строки: в первой указывается номер листа, во второй — порядковый номер блока.

Схема должна содержать все разветвления, циклы и обращения к подпрограммам, содержащиеся в программе.

Все существующие алгоритмы делятся на три вида: линейные, ветвящиеся и циклические.

Линейный алгоритм при каждом исполнении предписывает однократное выполнение всех действий алгоритма в определенной последовательности.

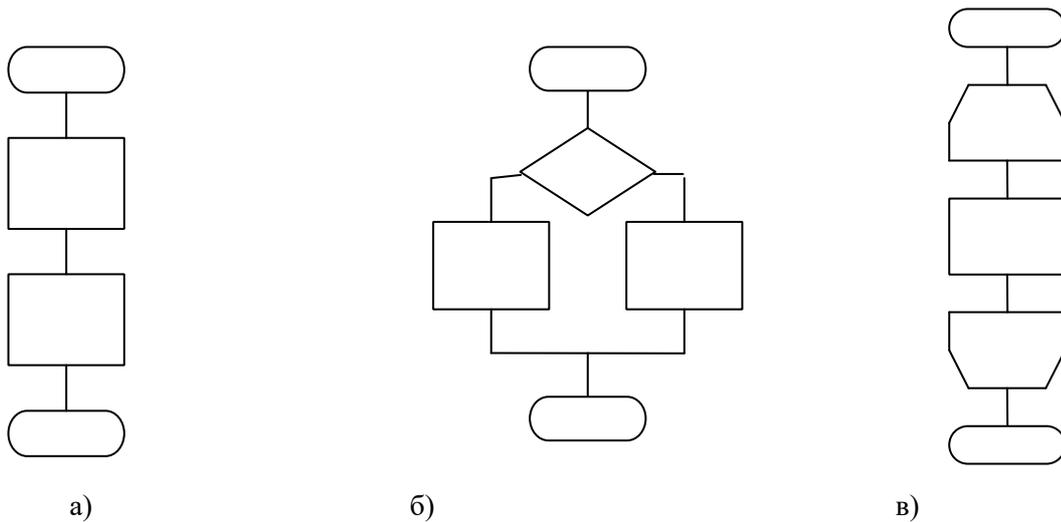
Ветвящийся алгоритм описывает несколько возможных последовательностей действий и при каждом исполнении предписывает выполнение одной из последовательностей действий в зависимости от определенных условий.

Циклический алгоритм при каждом исполнении предписывает многократное выполнение одной и той же последовательности действий.

Графические изображения различных видов алгоритмов приведены на рисунке 2.1.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



а) б) в)  
Рисунок 2 – Графические изображения алгоритмов: а) линейный алгоритм;  
б) ветвящийся алгоритм; в) циклический алгоритм.

#### 5.4 Схемы электросвязи

Принцип действия радиоэлектронной аппаратуры (РЭА) описывают схемы электрические. Согласно ГОСТ 2.701-75 схемой называется конструкторский графический документ, на котором все составные части изделия и связи между ними изображаются в виде условных графических обозначений (УГО).

Правила выполнения схем и УГО приведены в стандартах седьмой классификационной группы ЕСКД. Для электрических схем, например:

ГОСТ 2.701-84. Схемы. Виды и типы. Общие требования к выполнению.

ГОСТ 2.702-76. Правила выполнения электрических схем.

ГОСТ 2.702-81. Правила выполнения электрических схем цифровой электрической техники.

ГОСТ 2.710-81. Обозначения буквенно-цифровые в электрических схемах.

ГОСТ 2.759-82. Обозначения условные графические в электрических схемах. Элементы аналоговой техники.

Электрические схемы по назначению подразделяются на следующие основные типы:

- схемы электрические структурные (СЭС).
- схемы электрические функциональные (СЭФ).
- схемы электрические принципиальные (СЭП).

Они имеют свои кодовые обозначения, которые проставляются в основной надписи чертежа после буквенно-цифрового кода:

- для СЭС АБВГ XXX. XXX. Э1.
- для СЭФ АБВГ XXX. XXX. Э2.
- для СЭП АБВГ XXX. XXX. Э3.

АБВГ – код организации разработчика;

XXX. XXX. – код классификационной характеристики.

Э1, Э2, Э3 – коды соответственно структурной, функциональной и принципиальной электрических схем.

Схема электрическая структурная (код Э1) – это графический конструкторский документ, на котором показаны основные функциональные части изделия, их назначения и взаимосвязи. Эта схема общего ознакомления. В ней функциональные части изображаются в виде прямоугольников или квадратов и иногда в виде УГО.

Внутри прямоугольников наименования каждой функциональной части указывают в виде полной надписи или аббревиатуры. Допускается вместо надписи указывать нумерацию



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

функциональных частей, которая расшифровывается в специальной таблице, располагаемой в поле чертежа над основной надписью.

Схема электрическая функциональная (код Э2) – это графический конструкторский документ, который разъясняет определённые процессы протекающие в отдельных функциональных частях или в изделии в целом.

На СЭФ части приборов, назначения которых не требуют особых пояснений, обычно изображают упрощённо или с помощью УГО. Функциональные части, требующие детального рассмотрения, показывают отдельно. Если функциональный узел представляет собой стандартную технологическую единицу, например микросхему, то на схеме указывают только те его выводы, которые необходимы для обозначения направления электрического сигнала.

Схема электрическая принципиальная (код Э3) является графическим конструкторским документом, который определяет полный состав элементов и связи между ними и даёт детальное представление о принципах работы изделия. На СЭП также изображают все электрические элементы или устройства необходимые для осуществления и контроля в изделии заданных электрических процессов, а также элементы (соединения, зажимы и т.п.), которыми заканчиваются входные и выходные цепи.

Для создания схем электросвязи допускается использование приложения *MS Visio* или других программных продуктов, в которых предусмотрено построение схем – *Planar, Cisco Packet Tracer* и др.

## 5.5 Оформление листингов программ

В пояснительной записке и приложениях при разработке программы для ЭВМ должны приводиться фрагменты кода программного продукта, которые оформляются в виде листингов.

Слово «Листинг» и наименование помещают перед фрагментом программного кода (выравнивание по левому краю), начинают с заглавной буквы, не подчеркивают. Листинги нумеруются арабскими цифрами сквозной нумерацией. Если листинг один, то он обозначается так:

### Листинг 1 - Процедура сортировки одномерного массива

Допускается нумеровать листинги в пределах раздела. В этом случае номер листинга состоит из номера раздела и порядкового номера листинга, разделенных точкой, например, Листинг 1.1

При ссылке на коды программ следует писать: «...код данной процедуры представлен листингом 2», при их сквозной нумерации и «... код данной процедуры представлен листингом 2.1» при нумерации в пределах раздела. **Ссылки по тексту на листинги обязательны.**

Для написания программного кода используется шрифт «*Courier new*» прямым начертанием (не курсив), обычный (нежирный). Размер шрифта 10-12 пт.

## 6 Нормоконтроль

Нормоконтролёр назначается из числа высококвалифицированных специалистов в области стандартизации.

Нормоконтроль предполагает выполнение следующих работ:

1) нормоконтроль ВКР в целом, а именно проверку документов, которая начинается только при наличии всех подписей лиц, ответственных за содержание и выполнение документов, кроме утверждающей подписи заведующего кафедрой;

2) нормоконтроль пояснительной записки, включающий проверку правил оформления пояснительной записки по ГОСТ 2.105-95 ЕСКД (Общие требования к текстовым документам):

– соответствие структуры пояснительной записки заданию на ВКР и действующим методическим материалам на кафедре;

– внешний вид записки;



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- соблюдение действующей научно-технической терминологии; – наличие ссылок на источники информации;
- правильность оформления таблиц, иллюстраций, приложений;
- соблюдение обозначений единиц физических величин;
- наличие и правильность ссылок на стандарты и другую нормативные документы;
- правильность нумерации и оформления наименований разделов и подразделов, иллюстраций и таблиц, списка использованных источников и соответствие их содержанию;

3) нормоконтроль спецификации предусматривает проверку её соответствия ГОСТ 2.106-96 ЕСКД (Текстовые документы);

4) составление перечня замечаний и предложений нормоконтролера, предполагающего анализ и оформление результатов нормоконтроля ВКР, доведение замечаний до должностных лиц с целью устранения ошибок в данных работах и выявления причин их появления.

## 7 Приемка программного продукта

При разработке программного или информационного продукта для профилей «Программное обеспечение и интеллектуальные системы (ПОиИС)» и «Вычислительные машины, комплексы, системы и сети (ВМКСС)» (09.03.01) приемка его проводится с целью выяснить его работоспособность и соответствие его функций заданию на ВКР.

Комиссия по приемке назначается заведующим кафедрой из числа сотрудников кафедры, специализирующихся в области программирования. В случае положительного решения результаты тестирования продукта оформляются комиссией в виде акта приемки, в котором указываются как достоинства разработки, так и обнаруженные недостатки. На основании этого делается вывод о возможной оценке результатов разработки.

Акт приемки, наряду с пояснительной запиской, является обязательным при защите ВКР по профилю «Программное обеспечение и интеллектуальные системы» предоставлении квалификационной работы на защиту. При неработающем программном продукте или отсутствии акта приемки ВКР к защите не допускается. Если ВКР выполнена по профилю ВМКСС и в задании отсутствует пункт разработки ПО с использованием языков высокого уровня, то для таких ВКР «Акт приемки программного продукта не требуется».

Принятый комиссией программный продукт в виде исходных кодов, а также скомпилированная программа в виде *exe*-файла записываются на дисковый носитель, который вкладывается в конверт, приклеенный к обложке пояснительной записки, или в файл, подшитый к записке.

В отдельных случаях, когда, например, по объективным причинам (особая платформа, размеры и т.п.) студент не может представить продукт в полном объеме, ему может быть предложен особый порядок приемки результатов разработки.

## 8 Подготовка к защите и защита ВКР

### 8.1 Предварительная защита ВКР

Предварительная защита выпускной квалификационной работы проводится в сроки, назначаемые заведующим кафедрой, не позднее, чем за 5 дней до назначенного срока защиты. Предварительная защита проходит в присутствии группы ведущих преподавателей кафедры (3-4 человека), также назначаемой заведующим кафедрой. Целью предварительной защиты является общая оценка степени готовности работы, а автора — к ее защите, определение правильности структуры и содержания доклада, соответствие содержанию графических материалов и др.

К моменту предварительной защиты у автора ВКР должны быть уже готовы презентация к докладу и пояснительная записка.

При подготовке ВКР к защите студент должен пройти следующие этапы:



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

1. Выпускная квалификационная работа представляется руководителю не позже чем за 10 дней до защиты. Завершенной считается работа в черновом варианте, содержащая все разделы, предусмотренные заданием, в том числе и отзыв руководителя ВКР.

Отзыв руководителя должен быть напечатан или написан разборчивым почерком и должен содержать следующие вопросы:

- актуальность работы;
- соответствие содержания работы заданию на ВКР;
- полноту, глубину и обоснованность решения поставленных вопросов;
- оценку личного вклада автора ВКР, уровень его теоретической подготовки, инициативность, умение самостоятельно решать инженернотехнические задачи, работать со специальной литературой;
- перспективы внедрения или внедрение разработки (в производстве, учебный процесс и др.);
- правильность расчетных и графических материалов;
- стиль расчетно-пояснительной записки;
- недостатки ВКР по содержанию и оформлению;
- при высоком научном уровне подготовки студента – рекомендацию для поступления в магистратуру;
- соответствие автора ВКР квалификации «бакалавр» по соответствующему направлению подготовки.

2. После проверки руководителем ВКР пояснительной записки на соответствие заданию и требованиям по использованию заимствований, подписанная студентом и руководителем работа со всеми ее структурными элементами, указанными в подразделе 2.1, представляется на нормоконтроль не позднее чем за 8 дней до защиты.

При получении замечаний по нарушению требований ГОСТов студент в указанные нормоконтроллером сроки устраняет их и повторно представляет исправленную работу. При этом студент должен представить нормоконтроллеру и листы ПЗ с его пометками по указанным замечаниям.

3. После получения разрешающей подписи нормоконтроллера законченная работа представляется заведующему выпускающей кафедрой для проверки на комплектность и соответствие работы заданию.

Законченной считается работа в чистовом варианте, прошедшая нормоконтроль, с отзывом руководителя, а также с распечатанными результатами проверки на объем заимствования.

Заведующий кафедрой после проверки всего комплекта ВКР дает разрешение на ее переплет в жесткую обложку.

Переплетённая работа представляется заведующему выпускающей кафедрой для допуска к защите не позднее 7 дней до защиты.

В это же время заведующему кафедрой автором ВКР передается пояснительная записка в электронном виде в полной комплектации в виде файла в формате PDF. Титульный лист, задание, отзыв и содержание включаются в файл без подписей. Также заведующему кафедрой передается и файл с графическими материалами в формате PPT или PPTX.

4. ВКР, допущенная к защите, направляется заведующим кафедрой на рецензию. Срок получения рецензии не должен превышать 5 дней до защиты. Состав рецензентов утверждается директором ТУ УГМК по представлению заведующего кафедрой из числа специалистов производства, научных учреждений, а также преподавателей и сотрудников других вузов аналогичного профиля.

В рецензии в обязательном порядке должны быть отражены:

- актуальность темы ВКР;



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- соответствие содержания работы заданию;
- полнота, глубина и обоснованность решения поставленных в работе задач;
- уровень теоретической подготовки автора ВКР, умение самостоятельно решать технические задачи с использованием ЭВМ и программных средств;
- использование специальной литературы и других материалов;
- правильность расчетных и графических материалов;
- стиль пояснительной записки;
- замечание по поводу наличия в проекте самостоятельных оригинальных или интересных разработок;
- замечание по поводу соблюдения требований существующих стандартов;
- возможность использования материалов ВКР в производстве, научных и других организациях, учебном процессе; - недостатки ВКР;
- общая оценка работы.

5. Заведующий кафедрой не позднее, чем за 5 календарных дней до дня защиты знакомит с рецензией руководителя ВКР, автора ВКР и направляет пояснительную записку с рецензией в ГЭК для защиты.

6. Выпускная квалификационная работа в комплекте и рецензия, приложенная к ней, передаются в государственную экзаменационную комиссию не позднее, чем за 2 календарных дня до дня защиты выпускной квалификационной работы.

## 8.2 Защита квалификационной работы

Защита (доклад результатов ВКР, ответы на вопросы) осуществляется перед Государственной экзаменационной комиссией (ГЭК). Состав ГЭК утверждается ректором головного университета на основе кандидатур предложенных директором ТУ УГМК, заведующими выпускающими кафедрами.

Председатель комиссии и некоторые ее члены, составляющие не менее 50% от общего числа членов комиссии, не являются сотрудниками ТУ УГМК. Как правило, это руководители крупных предприятий связи (работодатели, лица, имеющие практически опыт работы и управления). В комиссию также могут входить преподаватели различных кафедр (в том числе и выпускающей). Общий состав комиссии должен включать не менее 5 человек. Порядок защиты ВКР следующий:

1. Для защиты квалификационной работы подготавливается доклад, в котором излагаются основные положения и результаты выполнения квалификационной работы. Продолжительность доклада не более 4–6 минут (23 страницы машинописного текста через 1,5 интервала).

2. Доклад должен отражать суть выполненной работы и не должен включать обсуждение мелких незначительных деталей, если только эти детали не носят принципиальный или оригинальный характер.

3. Доклад должен сопровождаться компьютерной презентацией и графическими материалами, представленными в ней, иллюстрирующими основные результаты выполненной работы.

4. Весь материал, выносимый на слайды, обязательно должен быть идентичен иллюстрациям, представленным в ВКР. При этом перед началом защиты автор ВКР должен выдать каждому члену комиссии раздаточный материал в папке-скоросшивателе, где на



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

листах формата А4 распечатаны все графические единицы (содержание презентации) в порядке их представления в докладе.

5. По окончании доклада студенту задаются вопросы председателем и членами комиссии. Вопросы, как правило, связаны с темой ВКР или с получаемой квалификацией и могут касаться специальных учебных дисциплин, которые имеют отношение к представленному на защиту материалу.

6. После доклада и ответов на вопросы комиссии зачитывается отзыв руководителя ВКР и оглашается рецензия. Если в рецензии поставлены вопросы по содержанию ВКР, автор ВКР отвечает на них. В случае если рецензент выразил замечания: или соглашается с ними или аргументировано доказывает комиссии, что не согласен с ними.

7. По докладу и ответам студента на вопросы комиссия делает выводы о степени владения им материалом ВКР и умении аргументировано отстаивать свою точку зрения.

8. Оценка ВКР производится на закрытом заседании ГЭК. При ее оценке принимаются во внимание актуальность и научно-практическая ценность темы, степень раскрытия темы, качество выполнения и оформления работы, оценка ВКР рецензентом и оценка работы студента в ходе выполнения ВКР в отзыве руководителя, а также содержание доклада и ответы на вопросы. ВКР оценивается по четырех-балльной системе (отлично, хорошо, удовлетворительно, неудовлетворительно). Оценка объявляется после окончания защиты всех работ.

9. Обучающиеся, не прошедшие государственной итоговой аттестации в связи с неявкой на государственное аттестационное испытание по уважительной причине (временная нетрудоспособность, исполнение общественных или государственных обязанностей, вызов в суд, транспортные проблемы (отмена рейса, отсутствие билетов), погодные условия), вправе пройти ее в течение 6 месяцев после завершения государственной итоговой аттестации;

10. Обучающийся имеет право подать в апелляционную комиссию письменную апелляцию о нарушении, по его мнению, установленной процедуры проведения государственного аттестационного испытания и (или) несогласии с результатами государственного экзамена;

11. Лицо, не прошедшее государственную итоговую аттестацию, может повторно пройти государственную итоговую аттестацию не ранее чем через 10 месяцев и не позднее чем через пять лет после срока проведения государственной итоговой аттестации, которая не пройдена обучающимся. Указанное лицо может повторно пройти государственную итоговую аттестацию не более двух раз.

## Приложение А

Примерное направление тематики выпускных квалификационных работ  
(справочное)

### **«Информатика и вычислительная техника» (09.03.01).**

- 1. Программное обеспечение информационных систем в связи (по областям применения).**
- 2. Аспекты и методы использования аппаратных и программных средств вычислительной техники в прикладных задачах**
- 3. Исследование методов и аспектов использования программного обеспечения сетей и телекоммуникаций в информационных системах.** – Компьютерные сети.
  - Компоненты сетей.
  - Разработка спецификаций и конструирование компьютерных сетей.
- 4. Программы для бизнес-приложений инфокоммуникационных систем.**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Офисные автоматизированные системы.
- Информационные системы для руководства.
- Системы управления персоналом.
- Информационные системы в маркетинге.
- Информационные системы в эккаунтинге (сфера бизнеса, связанная со сбором, обработкой, анализом финансовой информации).
- Приложения для ERP (систем планирования ресурсов предприятия связи).

**5. *Исследование проблем защиты информационных систем от разрушения систем безопасности.***

- Методы контроля информации в информационных системах.
- Формальная политика безопасности при использовании ПО. – Пароли.
- Шифрование.
- Процедуры идентификации и аутентификации.
- Процедуры создания резервных копий.
- Управление рисками разрушения системы безопасности в условиях доступа к Интернету и межорганизационного обмена данными.
- Компьютерные вирусы, их типы и передвижение.
- Определение и борьба с вирусной инфекцией информационных систем.

**6. *Исследование способов самостоятельных расчетов и самостоятельной разработки бизнес приложений конечными пользователями.***

- Услуги, предоставляемые конечным пользователям информационными системами.
- Управление сетевыми услугами.
- Самостоятельные расчеты конечных пользователей.
- Использование концепции информационного центра для помощи пользователям в самостоятельной работе.
- Самостоятельная разработка бизнес приложений конечными пользователями.

**7. *Исследование методов управления информационными системами, базирующимися на Интернет.***

- Использование информационно-поисковых систем в Интернет.
- Бизнес приложения в Интернет.
- Методы разработки Интернет-приложений.
- Методы внедрения Интернет-приложений.

**8. *ПО развития и совершенствования учебного процесса.***

- Разработка современных систем тестирования на основе элементов искусственного интеллекта.
- Разработка электронных учебных пособий и учебно-методических материалов.
- Разработка обучающих программ с использованием мультимедиа технологий и интерактивного взаимодействия с пользователем.
- Разработка программ дистанционного обучения и тестирования.
- Разработка системы электронного документооборота кафедры, факультета, университета.
- Разработка ПО информационной сети для организации взаимодействия и выхода в сеть Интернет компьютеров кафедры.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Направление подготовки  
«Инфокоммуникационные технологии и системы связи» (11.03.02)**

**1. ЦСП**

- Реализация корпоративного доступа в Интернет на основе xDSL.
- Атмосферная оптическая система передачи информации между офисами.
- Разработка участка цифровой сети связи для радиопередачи сигналов структуры SDH.

**2. ВОЛС**

- Волоконно-оптическая линия связи между населенными пунктами ... с разработкой приемника.
- Волоконно-оптическая линия связи на участке ...

**3. РРЛ**

- Межрайонная радиорелейная линия для системы транковой связи.
- Радиорелейный доступ к магистральной сети для решения проблемы «последней мили».
- Радиорелейная линия между абонентскими пунктами ...

**4. Спутниковые системы**

- Система спутниковой связи VSAT.
- Ведомственная или персональная спутниковая система передачи информации на основе геостационарных спутников.
- Система спутниковой связи с использованием высокоэллиптических орбит.

**5. Системы радиосвязи с подвижными объектами**

- Абонентский терминал системы сотовой связи для организации связи в сотах большого радиуса.
- Проект участка беспроводной цифровой системы связи для передачи информации между базовыми станциями системы сотовой связи. – Проект сети связи по технологии LTE.

**6. Радиовещание и телевидение.**

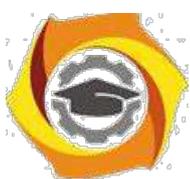
- Сеть IP-телевидения города (района)
- Проект участка беспроводной цифровой системы связи для передачи информации между базовыми станциями системы сотового телевидения.
- Малогабаритный трансивер прямой видимости с разработкой передающего устройства.

**7. Транковая связь**

- Передающее устройство транковой станции с использованием CDMA.
- Использование системы транковой связи стандарта «TETRA» службой МЧС.

**8. Разработка и модернизация сетей на основе протоколов и технологий:  
WIFI, LTE, NGN, KTB, MPLS и др.**

- Сеть доступа на базе технологии Wi-Fi г. \*\*\*.
- Сотовая сеть мобильной связи для г. \*\*\* на базе технологии LTE.
- Сотовая сеть мобильной связи LTE с анализом технологии MIMO.
- Модернизация сети связи города с применением технологий NGN (3G).



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

- Транспортная сеть NGN районного масштаба на базе технологии MPLS.
- Фемтосотовые сети предприятия.
- Волоконно-оптическая сеть доступа к мультисервисным услугам.
- Мультисервисная сеть связи г. \*\*\*.
- Первичная сеть связи на базе новой генерации (NGSDH).
- Широкополосный доступ в Internet по каналам КТВ.
- Транспортная сеть районного масштаба на базе технологии MPLS. – Модернизация сети связи на основе MPLS – технологий.

## **9. Разработка и модернизация сетей на основе волоконно-оптических технологий**

- Модернизация сельской телефонной сети за счет внедрения ВОСП. – Реконструкция сети связи на базе ВОЦСП синхронной иерархии.

## **10. Коммутационные системы**

- Модернизация узла связи города (района) на основе \*\*\*.
- Цифровая система коммутации \*\*\* на телефонной сети.
- Модернизация сети связи на основе ЦСК выбранного типа.
- Программируемый коммутатор (Softswitch) в сетях связи NGN городского масштаба.

## **11. Расчет сети предприятия**

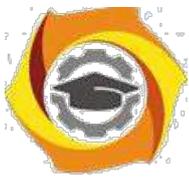
- Сеть IP-телефонии предприятия.
- Корпоративная сеть связи.
- Мультисервисная ведомственная выделенная сеть связи.
- Комплексная автоматизированная система охраны.
- Система комплексного видеонаблюдения предприятия.

## **12. Сети документальной электросвязи**

- Модернизация Главного узла СДЭС на основе системы \*\*\*.

## **13. Защита информации**

- Совершенствование комплексной системы защиты информации для телекоммуникационной системы.
- Разработка программного блока защиты от несанкционированного доступа на основе идентификации голоса пользователя.
- Разработка программного средства защиты информации на основе классификации биометрических характеристик пользователя.
- Разработка системы информационной безопасности корпоративных компьютерных систем с применением усовершенствованной электронной подписи.
- Разработка программы защиты информации от несанкционированного доступа на основе вейвлет-анализа голоса пользователя.
- Разработка программы аутентификации пользователя на основе анализа спектра акустического сигнала его голоса.
- Организация IP видеонаблюдения на предприятии связи для предотвращения несанкционированного доступа.
- Разработка рекомендаций по обеспечению компьютерной безопасности предприятия связи.
- Совершенствование системы безопасности информации ограниченного доступа на предприятии связи.
- Разработка комплекса мероприятий по модернизации существующей системы информационной безопасности предприятия связи.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Разработка метода защиты информации в телекоммуникационной системе на основе поточного шифрования.
- Совершенствование политики безопасности информации в системе сотовой связи.
- Разработка методики обнаружения современных средств негласного съема информации с технических каналов связи.
- Разработка системы защиты информации в корпоративной локальной сети ПАО «Ростелеком».
- Разработка системы физической защиты на предприятии связи.
- Выбор технологии проектирования систем защиты информации.
- Имитация многолучевого канала на основе IEEE 802.11b.
- Исследование ошибок в операционных системах.
- Модель системы управления информационной безопасностью в условиях неопределенности воздействия.
- Организация защиты персональных данных в организации
- Разработка алгоритма и программного обеспечения маскирования данных, исследование вопросов стойкости к частотному анализу.
- Разработка комплекса режимных мероприятий по сохранности конфиденциальной информации на примере ООО.
- Разработка корпоративной сети предприятия с подключением удаленных филиалов по каналам VPN.

## **Сжатие и хранение информации**

2

### **27) ИЗУЧЕНИЕ АЛГОРИТМА ХАФФМАНА**

#### **Цель**

- В Изучить принципы реализации алгоритма Хаффмана, его характеристики и ограничения;
- В Совершенствовать навыки использования языков и сред программирования для реализации алгоритмов сжатия;
- В Совершенствовать навыки анализа, обобщения и систематизации полученных результатов, навыки составления и оформления отчетных материалов, навыки точного и лаконичного представления докладов на вопросы технического характера.

#### **Учебные вопросы**

31. Разработка словесного описания алгоритма, схемы алгоритма;
32. Подготовка исходного текста программы;
33. Отладка и оценка работоспособности программы сжатия.

#### **Литература для подготовки к занятию**

- 96 Ватолин Д. и др. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2003. - 384 с.
- 97 Тропченко А.Ю., Тропченко А.А. Методы сжатия изображений, аудиосигналов и видео: Учебное пособие. - СПб.: СПбГУ ИТМО,



2009. - 108 с.

**Содержание отчета**

19. Название работы;
20. Для каждого задания: название задания и материал в объеме, указан-ном в задании.

Одним из наиболее широко распространенных видов сервисных программ являются программы, предназначенные для архивации, упаковки файлов путем сжатия хранимой в них информации.

**Сжатие информации** — это процесс преобразования информации, хранящейся в файле, к виду, при котором уменьшается избыточность в ее представлении и соответственно требуется меньший объем памяти для хранения.

Сжатие информации в файлах производится за счет устранения избыточности различными способами, например за счет упрощения кодов, исключения из них постоянных битов или представления повторяющихся символов или повторяющейся последовательности символов в виде коэффициента повторения и соответствующих символов.



**Цель процесса сжатия** – получение более компактного выходного потока информационных единиц из некоторого изначально некомпактного входного потока при помощи некоторого их преобразования.

Основными техническими характеристиками процессов сжатия и результатов их работы являются:

4. **степень сжатия** (compress rating) или отношение (ratio) объемов исходного и результирующего потоков;

5. **скорость сжатия** - время, затрачиваемое на сжатие некоторого объема информации входного потока, до получения из него эквивалентного выходного потока;

6. **качество сжатия** - величина, показывающая на сколько сильно упакован выходной поток, при помощи применения к нему повторного сжатия по этому же или иному алгоритму.

Все способы сжатия можно разделить на две категории: обратимое и необратимое сжатие.

Под **необратимым** сжатием подразумевают такое преобразование входного потока данных, при котором выходной поток, основанный на определенном формате информации, представляет, с некоторой точки зрения, объект, достаточно похожий по внешним характеристикам на входной поток, однако отличается от него объемом. Данный подход реализован в популярных форматах представления видео и фото информации, известных как JPEG / JFIF алгоритмы и JPG и GIF форматы файлов. Необратимое сжатие невозможно применять в областях, в которых необходимо иметь точное соответствие информационной структуры входного и выходного потоков.

**Обратимое** сжатие всегда приводит к снижению объема выходного потока информации без изменения его информативности, т.е. без потери информационной структуры. Более того, из выходного потока, при помощи восстанавливающего или декомпрессирующего алгоритма, можно получить входной, а процесс восстановления называется декомпрессией или распаковкой и только после процесса распаковки данные пригодны для обработки в соответствии с их внутренним форматом

Первые теоретические разработки в области сжатия информации относятся к концу 40-х годов. В конце семидесятых появились работы Шеннона, Фано и Хафмана. К этому времени относится и создание алгоритма FGK (Faller, Gallager, Knuth), где используется идея "сродства", а получатель и отправитель динамически меняют дерево кодов. Пропускная способность каналов связи более дорогостоящий ресурс, чем дисковое пространство, по этой причине сжатие данных до или во время их передачи еще более актуально.

Здесь целью сжатия информации является экономия пропускной способности и в конечном итоге ее увеличение. Все известные алгоритмы сжатия сводятся к шифрованию входной информации, а принимающая сторона выполняет дешифровку принятых данных.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Сжатие информации без потерь осуществляется статистическим кодированием или на основе предварительно созданного словаря. Статистические алгоритмы (напр., схема кодирования Хаффмана) присваивают каждому



входному символу определенным кодом. При этом наиболее часто используемому символу присваивается наиболее короткий код, а наиболее редкому – более длинный. Таблицы кодирования создаются заранее и имеют ограниченный размер. Этот алгоритм обеспечивает наибольшее быстродействие и наименьшие задержки. Для получения высоких коэффициентов сжатия статистический метод требует больших объемов памяти.

### ***Задание 1.1 Разработка словесного описания алгоритма, схемы алгоритма***

Сжатие файла по алгоритму Хаффмана основывается на частоте повторения символов. Для сжатия необходимо подсчитать сколько раз встречается каждый символ из расширенного набора ASCII. После подсчета частоты вхождения каждого символа, необходимо отсортировать получившуюся таблицу (символ – частота вхождения) по частоте. Каждую ссылку из таблицы назовем "узлом". В дальнейшем (в дереве) мы будем позже размещать указатели, которые будут указывать на этот "узел".

На вход программы подается файл, содержащий текст. Затем осуществляется анализ текста, выполняется подсчет частот находящихся в нём различных символов, заполнение массивов данных и их сортировка.

Следующий шаг - построение бинарного дерева. Элементы, имеющие большие частоты, располагаются справа, а элементы с меньшими частотами - слева.

При построении кода для каждого символа программа проходит по левым ссылкам узлов дерева до первого встреченного листа и обнаруживает первый элемент для объединения. После этого лист удаляется из дерева, и начинается аналогичный проход для поиска второго элемента, который после обнаружения также удаляется из дерева.

/ коду всех символов первого из найденных листов добавляется единица, к коду всех символов второго листа добавляется ноль. Так, например, если символ «а» имел код 01, то, если он окажется в первом из найденных листов, код примет вид 101, если же во втором, то 001.

После нахождения двух элементов происходит их объединение в один новый элемент, причем частоты обоих суммируются, и получившееся значение присваивается новому элементу. Для нахождения места расположения образованного элемента в дереве сравниваются частоты слева и справа от текущей позиции, начиная с корневого узла. На основании этих сравнений выполняется передвижение по дереву либо влево, либо вправо. Если находится узел, частота которого равна частоте образованного элемента, происходит его



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

вставка в текущую позицию, а найденный узел прикрепляется к встав-ленному слева.



Возможен случай, когда частота образованного элемента больше, чем частота узла, стоящего справа относительно текущей позиции. Если при этом выполнить действия, аналогичные действиям выше, и вставить новый элемент в дерево на место текущего, то возможна ошибка обхода дерева, в результате чего образованные коды символов будут некорректны. Для того чтобы это исправить запускается алгоритм, перестраивающий бинарное дерево. После подобной перестройки алгоритм образования новых узлов повторяется.

Программа продолжает свою работу до тех пор, пока в дереве не останется один элемент. Это означает, что все элементы были обработаны и для всех символов, которые располагались в исходном файле, были построены коды.

После этого выводится исходная строка и количество бит, занимаемых этой строкой, закодированная строка и количество бит, которые требуются для ее кодирования.

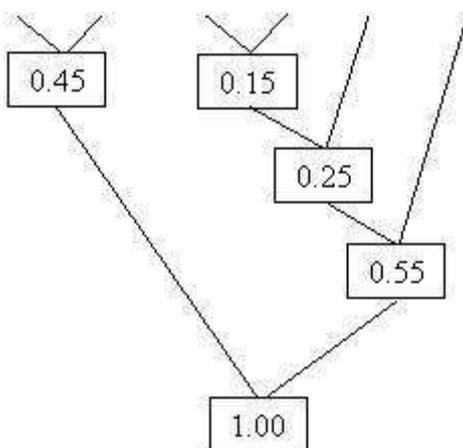
/ качестве входных можно использовать тексты, содержащие символы русского и английского алфавитов, фрагменты кода программ и другие символы.

Пример:

Допустим, исходный файл длиной в 100 байт и состоит из 6 различных символов. Частота вхождения каждого из символов в файл представлена в таблице:

A	B	C	D	E	F
0.25	0.2	0.05	0.1	0.1	0.3

В дальнейшем элементы таблицы считаются узлами, а их вероятность – весом узлов. Дерево формируется следующим образом. Дальнейший порядок следующий.



1. Элементы таблицы сортируются в порядке убывания их частот.

2. Сначала выбираются два узла с наименьшим весом (два последних элемента), вместо них создается новый узел (Вершина #n), с весом, равным их сумме. Новая вершина становится элементом таблицы вместо использованных двух узлов и занимает свое место в порядке убывания весов.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

/ Повторяется данная операция шага 2 до тех пор, пока не останется один узел с весом в единицу.



В Далее выполняется кодирование. Для этого выполняется обход дерева от корня (узла с весом в 1) и спуск до самих символов (узлов, листьев дерева). При этом прослеживаются вверх по дереву все повороты ветвей и, если выполняется левый поворот, то запоминается 0-й бит, если правый - 1-й бит.

Таким образом, для символа А получается код 00, аналогично для других:

A	B	C	D	E	F
00	01	1000	1001	101	11

Подсчет сжатия показывает:

было 100 байт = 800 бит

стало  $25*2 + 20*2 + 5*4 + 10*4 + 10*3 + 30*2 = 240$  бит = 30

байт Получается, что сжатие на 70%.

Для разархивирования необходимо сохранить дерево либо непосредственно таблица частот, и тогда при разархивировании потребуется дерево восстановить заново.

Для сохранения таблицы частот потребуется:

6 символов = 6 байт;

6 частот =  $6*4 = 24$  байт, итого  $30+6+24 = 60$  байт.

Также обычно сохраняют служебную информацию (время создания, время редактирования, атрибуты, имя файла):  $4+4+1+11$  (длину файла считают 11 символов, включая расширение). Всего 20 байт.

Получается архив в  $20+60 = 80$  байт.

Окончательно сжатие на 20%. В среднем алгоритм Хаффмана даёт 20-25% сжатие (для текстовых документов).

При этом нет необходимости в байтах-флагах, т. е. не надо разделять битовые серии.

Недостатком следует признать необходимость двух проходов во время архивирования: первый – подсчёт количества вхождений каждого байта, второй – непосредственно архивирование.

\endash *отчете представить:*

1.Схему алгоритма;

2.Описание переменных.

**Задание 1.2. Подготовка исходного текста программы**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

40. соответствии с разработанным алгоритмом составить программу его реализации на одном из языков программирования. Использовать освоенные среды программирования.



***В отчете представить:***

1. Исходный текст программы.
2. Комментарии к созданным блокам и процедурам.

***Задание 1.3. Отладка и оценка работоспособности программы сжатия***

Провести отладку программы на примере подготовленного тестового файла для сжатия.

Выполнить тестовые прогоны для оценки параметров сжатия.

***В отчете представить:***

1. Отлаженную программу сжатия;
38. Данные тестовых прогонов на различных файлах (время работы, степень сжатия, влияние уникальных символов на степень сжатия).

***Вопросы для самопроверки***

29. В чем смысл понятия: избыточность сообщения?
30. Что определяет коэффициент избыточности?
31. Алгоритм построения дерева Хаффмана.
32. Какие задачи решает кодирование методом Хаффмана?
33. При кодировании каких данных можно использовать сжатие данных с потерями? Ответ обоснуйте.
34. В чем преимущества и недостатки статических методов и словарного сжатия?
35. Каким образом кодирование по алгоритму Хаффмана через префиксный код гарантирует минимальную длину кода?
36. За счет чего в методе Хаффмана поддерживается однозначность соответствия кода кодируемому символу?
37. Почему алгоритм Хаффмана малоэффективен для файлов маленьких размеров?
38. Выполните кодирование по методу Хаффмана через префиксный код символов, которые встречаются с вероятностями 0,3; 0,2; 0,1; 0,1; 0,1; 0,05; 0,05; 0,04; 0,03; 0,03. Сравните полученный результат с данными программной реализации.
39. Докажите, что метод Хаффмана кодирует информацию без потерь.
40. Порядок работы алгоритма Хаффмана
41. Построение оптимального кодового дерева.
42. Средняя длина кода и ее расчет.



### 34. ИЗУЧЕНИЕ АЛГОРИТМА LZ

#### Цель

№ Изучить принципы реализации алгоритма Лемпеля-Зива, его характеристики и ограничения;

№ Совершенствовать навыки использования языков и сред программирования для реализации алгоритмов сжатия;

№ Совершенствовать навыки анализа, обобщения и систематизации полученных результатов, навыки составления и оформления отчетных материалов, навыки точного и лаконичного представления докладов на вопросы технического характера.

#### Учебные вопросы

59. Разработка словесного описания алгоритма, схемы алгоритма;

60. Подготовка исходного текста программы;

61. Отладка и оценка работоспособности программы сжатия.

#### Литература для подготовки к занятию

44. Ватолин Д. и др. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2003. - 384 с.

45. Тропченко А.Ю., Тропченко А.А. Методы сжатия изображений, аудиосигналов и видео: Учебное пособие. - СПб.: СПбГУ ИТМО, 2009. - 108 с.

#### Содержание отчета

42. Название работы;

43. Для каждого задания: название задания и материал в объеме, указанном в задании.

Алгоритм RLE (Run Length Encoding - кодирование длин повторов) предполагает представление изображения в виде последовательности байтов по строкам раstra. Сжатие происходит повторяющихся байтов путем замены их парами "Счетчик повторений, значение байта".

Признаком счетчика служат единицы двух старших бит в байте. Оставшиеся 6 бит указывают количество повторений значения второго байта этой пары:

--



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

11 |

1-й байт  
значение счетчика

2-й байт  
Код



Алгоритм рассчитан на деловую графику - изображения с большими областями повторяющегося цвета. В общем случае возможны файлы, для которых сжатый файл может быть больше исходного.

1977 году Абрахам Лемпель и Якоб Зив предложили алгоритм сжатия данных, названный позднее LZ77. Этот алгоритм используется в программах архивирования текстов compress, lha, pkzip и arj. Модификация алгоритма LZ78 применяется для сжатия двоичных данных. В основе лежит метод LZW (по символам имен разработчиков Lempel, Ziv и Welch), предполагающий сжатие за счет кодирования одинаковых цепочек байтов.

Эти модификации алгоритма защищены патентами США. Алгоритм предполагает кодирование последовательности бит путем разбиения ее на фразы с последующим кодированием этих фраз. Суть алгоритма заключается в следующем.

Если в тексте встретится повторение строк символов, то повторные строки заменяются ссылками (указателями) на исходную строку. Ссылка имеет формат <префикс, расстояние, длина>. Префикс в этом случае равен 1. Поле расстояние идентифицирует слово в словаре строк. Если строки в словаре нет, генерируется код символ вида <префикс, символ>, где поле префикс = 0, а поле символ соответствует текущему символу исходного текста. Отсюда видно, что префикс служит для разделения кодов указателя от кодов символ. Введение кодов <символ> позволяет оптимизировать словарь и поднять эффективность сжатия. Главная алгоритмическая проблема здесь заключается в оптимальном выборе строк, так как это предполагает значительный объем переборов.

Альтернативой статистическому алгоритму стала схема сжатия, основанная на динамически изменяемом словаре (напр., алгоритмы Лемпеля-Зива). Данный метод предполагает замену потока символов кодами, записанными в памяти в виде словаря (таблица перекодировки). Соотношение между символами и кодами меняется вместе с изменением данных. Таблицы кодирования периодически меняются, что делает метод более гибким. Размер небольших словарей лежит в пределах 2-32 килобайт, но более высоких коэффициентов сжатия можно достичь при заметно большими словарями до 400 килобайт.

Реализация алгоритма возможна в двух режимах: непрерывном и пакетном. Первый использует для создания и поддержки словаря непрерывный поток символов. При этом возможен многопротокольный режим (например, TCP/IP и DECnet). Словари сжатия и декомпрессии должны изменяться синхронно, а канал должен быть достаточно надежен (напр., X.25 или PPP), что гарантирует отсутствие искажения словаря при повреждении или потере пакета. При искажении одного из словарей оба ликвидируются и должны быть созданы вновь.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Пакетный режим сжатия также использует поток символов для создания и поддержания словаря, но поток здесь ограничен одним пакетом и по этой причине синхронизация словарей ограничена границами кадра. Для па-



кетного режима достаточно иметь словарь объемом, порядка 4 Кбайт. Непрерывный режим обеспечивает лучшие коэффициенты сжатия, но задержка получения информации (сумма времен сжатия и декомпрессии) при этом больше, чем в пакетном режиме.

***Задание 2.1. Разработка словесного описания  
алгоритма, схемы алгоритма***

Процесс сжатия выглядит достаточно просто. Считываются последовательно символы входного потока и проверяются, есть ли в созданной таблице словаря такая строка. Если строка есть, то считывается следующий символ, а если строки нет, то заносится в поток код для предыдущей найденной строки, заносится строка в таблицу и начинается поиск снова.

Инициализированная таблица словаря должна содержать все возможные строки, состоящие из одного символа. При сжатии байтовых данных в таблице будет 256 строк, а номер строки будет ее кодом. Для кода очистки и кода конца информации зарезервированы значения 256 и 257. Для 12-битового кода кодировки для строк еще остается довольно много значений.

Например, пусть сжимается последовательность 45, 55, 55, 151, 55, 55, В Тогда, согласно алгоритму, помещается в выходной поток сначала код очистки <256>, потом добавляется к изначально пустой строке "45" и проверяется, есть ли строка "45" в таблице. Поскольку при инициализации занесены в таблицу все строки длиной в один символ, то строка "45" есть в таблице. Далее читается следующий символ 55 из входного потока и проверяется, есть ли строка "45, 55" в таблице. Такой строки в таблице пока нет. В таблицу заносится строка "45, 55" (с первым свободным кодом 258) и записывается в поток код <45>. Можно коротко представить архивацию так:

"45" - есть в таблице;

"45, 55" - нет. Добавляется в таблицу <258>"45, 55". В поток: <45>;

"55, 55" - нет. В таблицу: <259>"55,55". В поток: <55>;

"55,151" - нет. В таблицу: <260>"55, 151". В поток: <55>;

"151,55" - нет. В таблицу: <261>"151,55". В поток: <151>;

"55, 55" - есть в таблице;

"55,55,55" - нет. В таблицу: "55,55, 55" <262>. В поток: <259>.

Последовательность кодов для данного примера, попадающих в выходной поток: <256>, <45>, <55>, <55>, <151>, <259>.

Особенность LZW заключается в том, что для декомпрессии не надо сохранять таблицу строк в файле для распаковки. Алгоритм построен таким



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

образом, что есть возможность восстановить таблицу строк, пользуясь только потоком кодов.



Для каждого кода надо добавлять в таблицу строку, состоящую из уже присутствующей там строки и символа, с которого начинается следующая строка в потоке.

***В отчете представить:***

- ООО Схему алгоритма процедуры сжатия;
- ООО Имена и типы использованных переменных.

***Задание 2.2. Подготовка исходного текста программы***

20. соответствии с разработанным алгоритмом составить программу его реализации на одном из языков программирования. Использовать освоенные среды программирования.

***21. отчете представить:***

- 1. Исходный текст программы.
- 2. Комментарии к созданным блокам и процедурам.

***Задание 2.3. Отладка и оценка работоспособности программы сжатия***

Провести отладку программы на примере подготовленного тестового (или иного) файла для сжатия.

Выполнить тестовые прогоны для оценки параметров сжатия.

***В отчете представить:***

- 1. Отлаженную программу сжатия;
- 42. Данные тестовых прогонов на различных файлах (время работы, степень сжатия).

***Вопросы для самопроверки***

- 33. В чем состоит основной принцип работы алгоритма LZ77?
- 34. В чем смысл принципа скользящего окна?
- 35. В чем состоит основной принцип работы алгоритма LZ78?
- 36. Основное отличие алгоритма LZ78 от алгоритма LZ77?
- 37. Приведите определение процесса сжатия данных.
- 38. Приведите определение неискажающего сжатия цифровых данных (сжатие без потерь).
- 39. Приведите определение сжатия цифровых данных с регулируемыми потерями.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

40. Приведите определение архиватора. Приведите примеры пространственных архиваторов.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

14

- к Приведите формулы подсчета коэффициента сжатия, степени сжатия.
- к Приведите формулу подсчета симметричности по времени алгоритма сжатия. В каких случаях оправдано применение несимметричных по времени алгоритмов сжатия данных?
- к Поясните смысл термина «масштабирование изображений» при использовании архиваторов.
- к В рекламе на архиватор А указано, что он имеет коэффициент сжатия 20, в рекламе на архиватор В указано, что он обеспечивает степень сжатия 80. Какой из архиваторов формирует более компактный файл сжатых данных?
- к Какие алгоритмы сжатия без потерь Вам известны?
- к Сформулируйте идею сжатия данных статистическими алгоритмами.
- к Какой принцип положен в основу алгоритмов RLE?
- к Какие файлы являются наилучшими объектами для сжатия алгоритмом RLE?
- к Сформулируйте идею алгоритма кодирования по ключевым словам.
- к Для каких данных этот алгоритм наиболее эффективен?
- к Сформулируйте сущность алгоритма LZ.
- к Почему используется вариант алгоритма Лемпеля-Зива со скользящим окном?
- к Опишите процесс сжатия данных алгоритмом Лемпеля-Зива.
- к Чем отличается алгоритм арифметического кодирования от алгоритма Хаффмена?



## 45. ИЗУЧЕНИЕ АЛГОРИТМА СЖАТИЯ СТАНДАРТА JPEG

### Цель

17. Изучить принципы реализации алгоритма сжатия стандарта JPEG, его характеристики и ограничения;
18. Совершенствовать навыки использования языков и сред программирования для реализации алгоритмов сжатия;
19. Совершенствовать навыки анализа, обобщения и систематизации полученных результатов, навыки составления и оформления отчетных материалов, навыки точного и лаконичного представления докладов на вопросы технического характера.

### Учебные вопросы

18. Разработка словесного описания алгоритма, схемы алгоритма;
19. Подготовка исходного текста программы;
20. Отладка и оценка работоспособности программы сжатия.

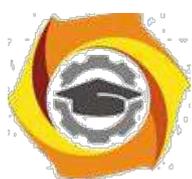
### Литература для подготовки к занятию

24. Ватолин Д. и др. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2003. - 384 с.
25. Тропченко А.Ю., Тропченко А.А. Методы сжатия изображений, аудиосигналов и видео: Учебное пособие. - СПб.: СПбГУ ИТМО, 2009. - 108 с.
26. Программная реализация дискретного косинусного преобразования алгоритма JPEG. Методические указания для проведения лабораторной работы. - Томск: Изд. ТПУ, 2000. - 16с.

### Содержание отчета

21. Название работы;
22. Для каждого задания: название задания и материал в объеме, указанном в задании.

Классические алгоритмы сжатия без потерь перестали удовлетворять требованиям, предъявляемым к сжатию. Многие изображения практически не сжимались, хотя "на взгляд" обладали явной избыточностью. Это привело к созданию нового типа алгоритмов, сжимающих с потерей информации. Как правило, степень сжатия и, следовательно, степень потерь качества в них можно задавать. При этом достигается компромисс между размером и качеством изображений.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Одна из серьезных проблем машинной графики заключается в том, что до сих пор не найден адекватный критерий оценки потерь качества изобра-



жения. А теряется оно постоянно - при оцифровке, при переводе в ограниченную палитру цветов, при переводе в другую систему цветопредставления для печати и, что особенно важно, при сжатии с потерями.

JPEG (Joint Photographic Expert Group) - один из новых и достаточно мощных алгоритмов, разработанных для сжатия 24-битовых изображений. Практически он является стандартом де-факто для полноцветных изображений. Оперирует алгоритм областями  $8 \times 8$ , на которых яркость и цвет меняются сравнительно плавно. Вследствие этого при разложении матрицы такой области в двойной ряд по косинусам значимыми оказываются только первые коэффициенты. Таким образом, сжатие в JPEG осуществляется за счет плавности изменения цветов в изображении.

21. целом алгоритм основан на дискретном косинусном преобразовании (в дальнейшем - ДКП), применяемом к матрице изображения для получения некоторой новой матрицы коэффициентов. Для получения исходного изображения применяется обратное преобразование.

ДКП раскладывает изображение по амплитудам некоторых частот. Таким образом, при преобразовании мы получаем матрицу, в которой многие коэффициенты либо близки, либо равны нулю. Кроме того, благодаря несовершенству человеческого зрения можно аппроксимировать коэффициенты более грубо без заметной потери качества изображения.

### ***Задание 3.1. Разработка словесного описания алгоритма, схемы алгоритма***

Разработать алгоритмы, выполняющие следующие функции:

- а) формирование ДКП матрицы. Вычисление осуществлять с точностью 6 знаков после запятой (точки);
- б) формирование транспонированной ДКП матрицы;
- в) формирование из исходной матрицы изображения (дает преподаватель)  $P_{DCT}$ -матрицу. Перед выполнением ДКП нужно из всех элементов матрицы изображения вычесть число 128; в этой части программы, предусмотреть возможность использования матрицы ДКП, получаемой по одной из возможных формул;
- г) составление матрицы квантования для качества, задаваемого с клавиатуры;
- д) реализация квантования, т. е. получение  $F_{DCT}$ ;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

е) преобразование  $I_{DCT}^{PB}$  в матрицу распакованного изображения.



***В отчете представить:***

20. Схема этапов работы алгоритма JPEG и краткое описание функций, выполняемых на этапах;
21. Последовательность выполнения ДКП. Формулы для подсчета элементов матриц ДКП;
22. Схемы алгоритмов преобразования на этапах JPEG;
23. Имена и типы использованных переменных.

***Задание 3.2. Подготовка исходного текста программы***

18. соответствии с разработанным алгоритмом составить программу его реализации на одном из языков программирования. Использовать освоенные среды программирования.

Все данные (предварительно сформированные) можно поместить в один файл последовательно друг за другом. Матрицы располагаются по строкам, а векторы – в виде вектора строки. В этом случае формируется одна последовательность операторов для открытия файла для чтения. Возможно создать для каждой матрицы и вектора отдельные файлы. После чтения данных из файлов, все файлы должны быть закрыты.

Ввод матрицы выполняется по строкам с помощью двойного цикла.

Транспонированной матрицей  $A^T$  называется матрица, у которой столбцы соответствуют строкам, а строки столбцам матрицы  $A$ , например:

$$A = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} \qquad A^T = \begin{vmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{vmatrix} .$$

Из примера видно, что, если матрица квадратная, то диагональные элементы у матрицы  $A$  и  $A^T$  одни и те же, а переставляются элементы симметричные относительно главной диагонали. Эта операция реализуется операторами

$$c := a_{ij}; a_{ij} := a_{ji}; a_{ji} := c;$$

выполняемыми в двойном цикле: во внешнем по  $i$  во внутреннем по  $j$ .

Умножение матрицы  $A[m \times n]$  на матрицу  $B[n \times l]$  выполняется по формуле



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

$$C_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$



где  $j = 1, 2, \dots, l$ ,  $i = 1, 2, \dots, m$ ,  $C_{ij}$  – элементы матрицы  $C$ , получающейся в результате умножения. Полученная матрица  $C$  имеет размер  $m \times l$ .

Для вычисления элементов матрицы  $C[m \times l]$  необходимо использовать тройной цикл: по  $i$ , по  $j$ , по  $k$ .

Например при перемножении следующих матриц ( $m = 4$ ,  $n = 5$ ,  $l = 3$ ) получаем:

$$\begin{vmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 1 \\ 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 1 & 2 \end{vmatrix} * \begin{vmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{vmatrix} = \begin{vmatrix} 15 & 30 & 45 \\ 31 & 62 & 93 \\ 20 & 40 & 81 \\ 27 & 54 & 81 \end{vmatrix}.$$

***В отчете представить:***

1. Исходный текст программы.
2. Комментарии к созданным блокам и процедурам.

***Задание 3.3. Отладка и оценка работоспособности программы сжатия***

Провести отладку программы на примере подготовленного тестового (или иного) файла для сжатия.

Выполнить тестовые прогоны для оценки параметров сжатия.

***В отчете представить:***

1. Отлаженную программу сжатия;
2. Распечатки матриц ДКП, полученные по формулам, соответствующих им транспонированных матриц;
15. Распечатку матрицы блока сжимаемого изображения, матрицу  $R_{DST}$ , полученные при использовании ДКП;
16. Данные тестовых прогонов на различных файлах (время работы, степень сжатия).

**Вопросы для самопроверки**

12. За счет чего осуществляется сжатие в алгоритме JPEG?
13. Для чего необходимо преобразование цветового пространства изображения?
14. Почему в алгоритме не применяется субдискретизация компонентов яркости?
15. Назовите виды сегментации изображений и их отличия. Какая сегментация используется в алгоритме JPEG?

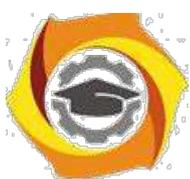


**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

16. Для чего необходимо дискретное косинусное преобразование?



- И На каком этапе сжатия происходит потеря качества изображения? Как они связаны с фактором качества?
- И Каким образом проявляются потери качества в алгоритме JPEG?
- И Какие алгоритмы кодирования использует JPEG?
- И Приведите примеры мер потери информации и опишите их недостатки.
- И В чем отличие сжатия цветных изображений от процедуры сжатия полутоновых изображений по стандарту JPEG?
- И Обнуление какого коэффициента ДКП вызовет самую сильную деградацию изображения?
- И Перечислите основные этапы работы алгоритма JPEG? Кратко сформулируйте, что делается на каждом этапе.
- И Приведите формулы перехода из цветовой системы RGB в цветовую систему YUV?
- И Приведите формулы перехода из цветовой системы YUV в систему RGB?
- И Объясните цель субдискретизации. Какие особенности зрения человека используются?
- И Для каких изображений субдискретизация не проводится?
- И Цель и порядок выполнения дискретного косинусного преобразования (ДКП)?
- И В каких стандартных алгоритмах сжатия кроме JPEG используется ДКП?
- И На каких этапах работы JPEG происходят невосстанавливаемые потери информации?
- И Нужно ли при сжатии запоминать значение «качество»? Если нужно, то – для чего?
- И Обязательно ли JPEG предполагает сжатие с потерями?
- И JPEG – открытый или закрытый стандарт? Может ли разработчик программного продукта внести какие-то свои изменения?
- И Каким образом подсчитывается коэффициент сжатия?
- И Каким образом подсчитывается степень сжатия?



## · ИЗУЧЕНИЕ АРХИВАТОРОВ

### Цель

- Изучить принципы реализации архиваторов, их особенности и параметры эффективности сжатия;
- Совершенствовать навыки анализа, обобщения и систематизации полученных результатов, навыки составления и оформления отчетных материалов, навыки точного и лаконичного представления докладов на вопросы технического характера.

### Учебные вопросы

1. Работа с архиватором;
2. Исследовать сжимаемость файлов различных типов;
3. Исследовать влияние метода сжатия на коэффициент сжатия и время сжатия;
4. Исследовать влияние размера словаря на коэффициент сжатия.

### Литература для подготовки к занятию

- Ватолин Д. и др. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2003. - 384с.
- Тропченко А.Ю., Тропченко А.А. Методы сжатия изображений, аудиосигналов и видео: Учебное пособие. - СПб.: СПб ГУИТМО, 2009. - 108с.

### Актуальность занятия

1. Многие виды информации в настоящее время просто не могут сохраняться и обрабатываться в несжатом виде (велико время, например, для видеосигналов);
2. Архиваторы – общедоступный универсальный инструмент сохранения всех необходимых данных для обработки информации. Используется массово. Информация в настоящее время сохраняется практически всегда в заархивированном виде (инсталляции).
3. Теоретические положения сжатия применяются для скрытия признаков обмена данными.

### Содержание отчета

- 14) Название работы;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

15) Для каждого задания: название задания и материал в объеме, указанном в задании.



### ***Задание 4.1. Работа с архиватором***

Изучить и выполнить основные операции с архиватором.

Индивидуальные задания:

В папке **"Мои документы/ФАМ"** создать папку **"АрхивФАМ"** (файлы, необходимые для архивирования подобрать по указанию преподавателя).

***В отчете представить:***

- и Перечень типовых операций с архиватором;
- и Созданный архив.

WinRAR—это мощное средство создания архивов и управления ими.

***Возможности WinRAR:***

- полная поддержка архивов RAR и ZIP;
- оригинальный высокоэффективный алгоритм сжатия данных;
- специальный алгоритм мультимедиа сжатия;
- поддержка технологии перетащить-и-оставить (drag&drop);
- интерфейс командной строки;
- управление архивами других форматов;
- метод непрерывного (solid) архивирования, при использовании которого может быть достигнута на 10–50% более высокая степень сжатия, чем дают обычные методы;
- поддержка многотомных архивов;
- создание самораспаковывающихся (SFX) обычных и многотомных архивов с помощью стандартного или дополнительных модулей SFX;
- восстановление физически поврежденных архивов;
- поддержка кодировки Unicode в именах файлов;
- другие дополнительные функции, например, шифрование, добавление архивных комментариев и пр.

***Интерфейс и основные инструменты***

Доступ к WinRar можно получить двумя путями:



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- используя меню "Пуск" (рис.4.1,а);



- выполнив щелчок ПКМ по файлу или папке, которую необходимо архивировать и из выпавшего списка выбрать пункт "добавить в архив" (рис.4.1,б).

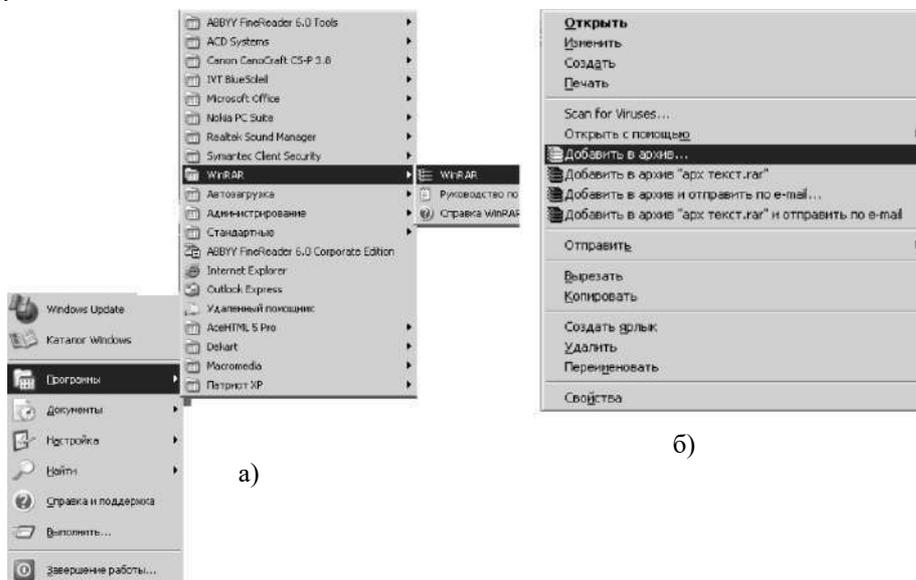
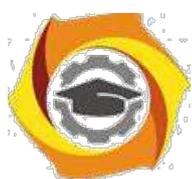


Рис.4.1. Способы доступа к архиватору: а) через меню ПУСК; б) из контекстного меню

Если используется первый способ (рис.4.1,а), то открывается стандартное окно архиватора (рис.4.2).



Рис.4.2. Стандартное окно архиватора



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*Архивация файлов (группы файлов)*



Архивировать (упаковывать в архив) можно отдельные файлы, группы файлов, папки с файлами.

Перед началом архивации необходимо определить параметры будущего архива:

- В состав архива (какие файлы будут входить в архив);
- В вид архива: обычный или самораспаковывающийся;
- В размер словаря, необходимого для архива;
- В конкретное место (диск, папка), в котором будет создан архив.

### ***Общий порядок архивирования файла***

Для создания архива необходимо: открыть архиватор, с помощью встроенного проводника найти файл, который следует упаковать в архив, выбрать параметры архивации, указать место, в котором будет создан архив и завершить операцию архивации.

#### **Пример 1.**

Создать архив на рабочем столе. Метод сжатия – максимальный. В состав архива включить файл `kalendar_2007.doc`, находящийся по адресу `D:\doc\KAG`.

#### **Решение.**

- Открыть WinRar и выполнить последующие шаги как показано на рис.4.3.

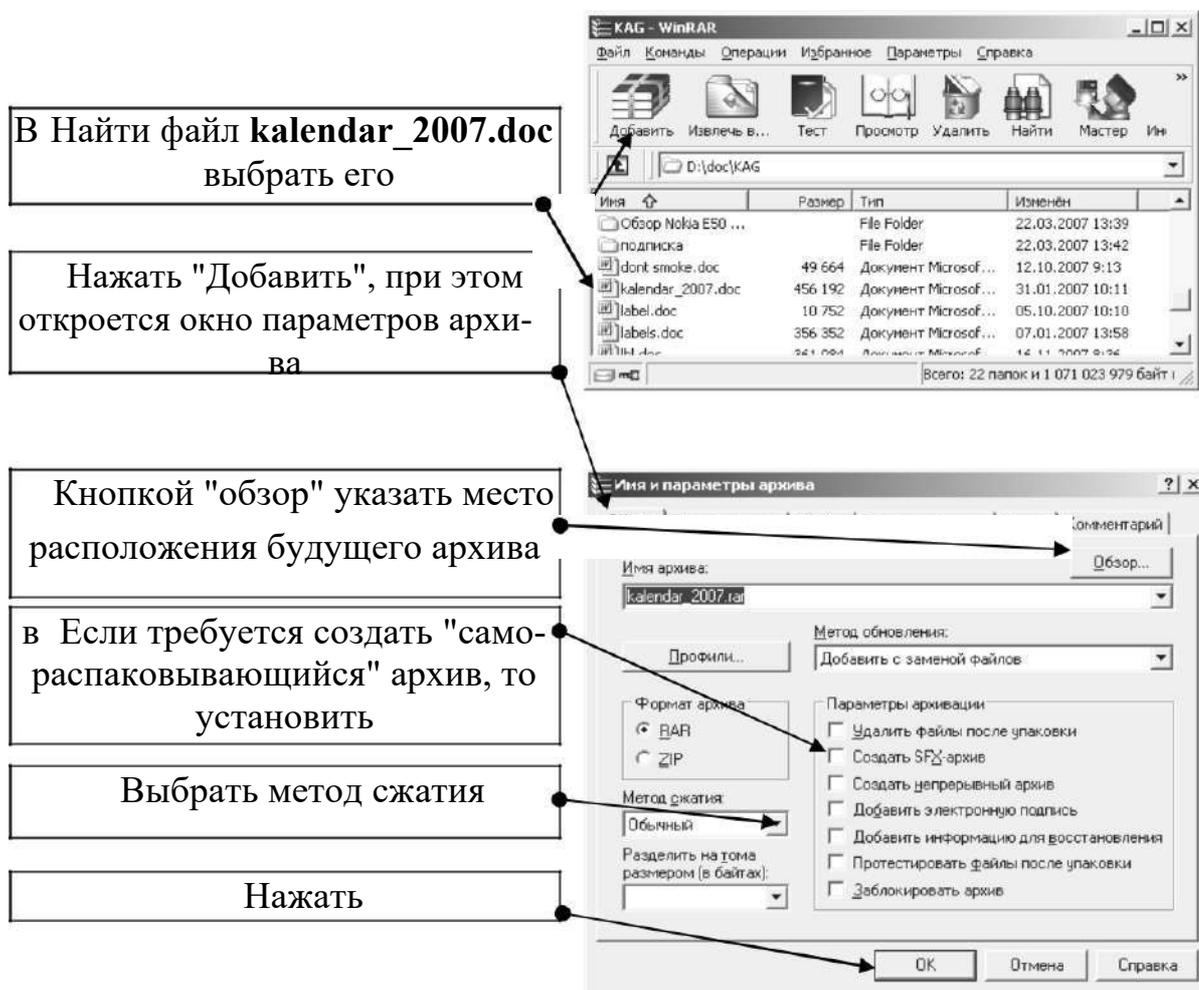


Рис.4.3. Этапы создания архива

к результате этих действий на рабочем столе появится архив **kalendar\_2007.rar** (рис.4.4). В случае создания самораспаковывающегося архива создаётся файл **kalendar\_2007.exe** (рис.4.5).



Рис.4.4. Архив RAR



Рис.4.5. Самораспаковывающийся архив

Если требуется создать архив с **определённым размером словаря**, то перед тем, как нажать кнопку **Ок** (см.рис.4.3) необходимо в окне "Имя и параметры архива" выбрать вкладку "**Дополнительно**" и в открывшемся окне нажать кнопку "**Параметры сжатия**" и установить размер словаря (рис.4.6).

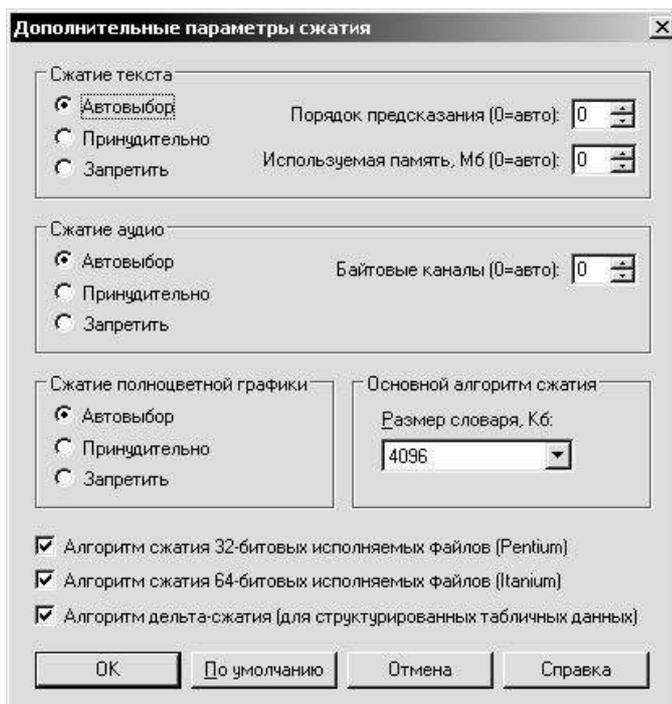
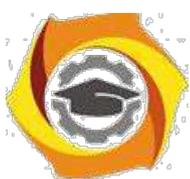


Рис.4.6. Настройка параметров сжатия

### *Общий порядок разархивирования (распаковки) архива*

Перед извлечением файлов из архива необходимо выяснить, в каком месте на диске должен оказаться распакованный файл (группа файлов).

Мнение о том, что, якобы, сначала можно распаковать, а затем переместить в другое место – ошибочно, так как, во-первых – может тривиально не хватить места на диске для хранения распакованного архива, а во-вторых – к папке, в которой находится архив, может быть организован доступ "только чтение", что не позволит сохранить распакованные файлы.

Пример 2.

На диске **D** имеется архив **calendar\_2007.rar**. Необходимо распаковать архив в папку **buffer**, которая находится на диске **D**.

Решение.

- в Открыть архиватор WinRar.
- в С помощью проводника архиватора найти архив **calendar\_2007.rar**.
- в В окне архиватора выполнить двойной щелчок ЛКМ по найденному архиву, при этом архив раскроется и пользователь увидит **какие файлы входят в состав архива**. Далее действуйте по этапам рис.4.7.

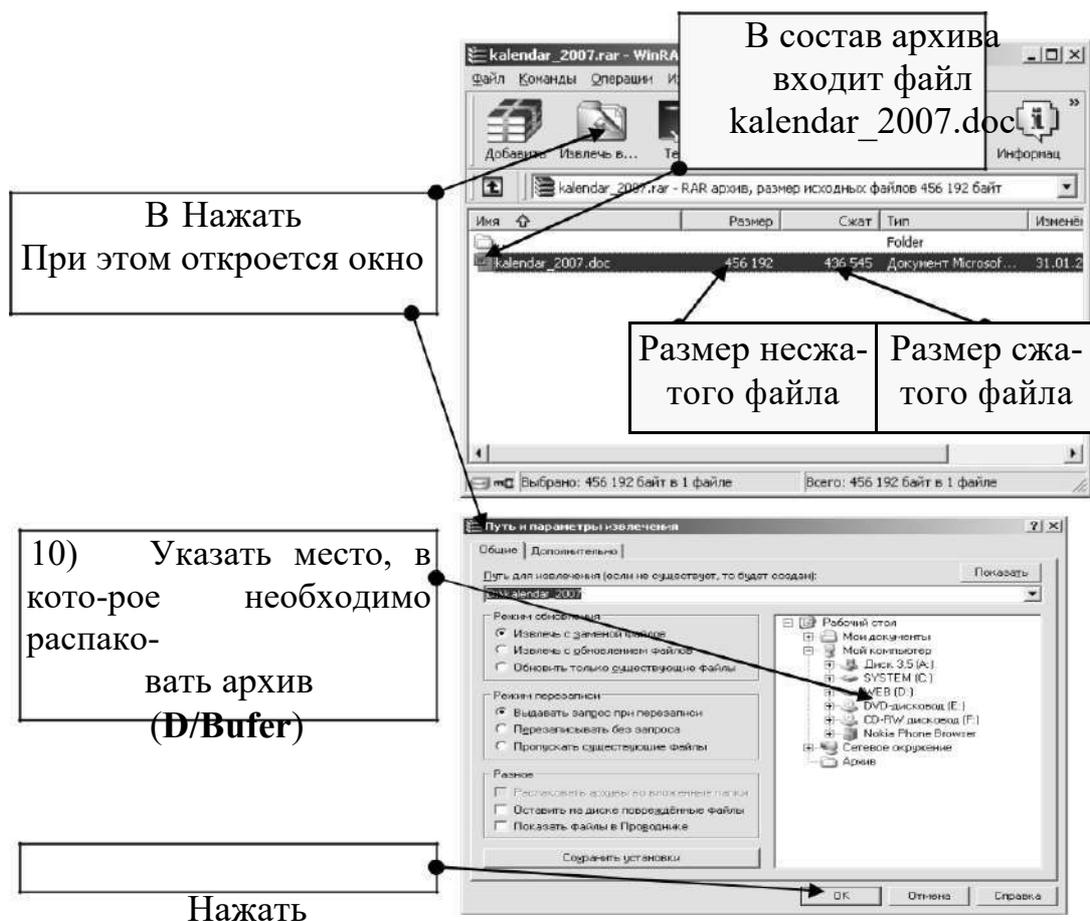
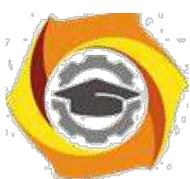


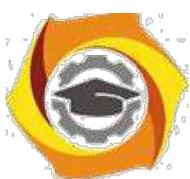
Рис.4.7. Завершающие этапы распаковки

к результату выполнения п.п.1–6 в папке **buffer** появится файл **calendar\_2007.doc**.

### *Особенности распаковки самораспаковывающегося архива*

Самораспаковывающиеся архивы предназначены для распаковки в компьютерах, на которых не установлена программа- архиватор, либо версия архиватора более ранняя и не способна распаковать архив, созданный в более поздней версии архиватора. Общий недостаток этого вида архивов - несколько больший размер архивного файла за счёт того, что в нём помимо заархивированных файлов содержится фрагмент программы- архиватора, который и осуществляет распаковку архива.

Для распаковки такого архива необходимо выполнить двойной щелчок по самораспаковывающемуся архиву (он имеет расширение **.exe**, что воспринимается операционной системой как приложение), в открывшемся окне указать место, в которое необходимо распаковать архив, и нажать кнопку "**Извлечь**".



### Изменение состава архива

Состав уже созданного архива может быть изменён. Под изменением будем понимать добавление новых файлов, или удаление существующих файлов из архива.

### Общий порядок изменения состава архива

17. Открыть архиватор WinRAR.
18. Найти архив, состав которого требуется изменить.
19. Выполнить двойной щелчок ЛКМ по значку найденного архива

Нажать

состав архива  
входит 1 файл

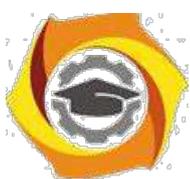
в В открывшемся окне  
выбрать требуемые  
файлы

Нажать **Ок**

7. При необходимости изменить параметры архива

33. Нажать **Ок**

Рис.4.8. Изменение состава архива



Теперь в архиве **calendar\_2007.rar** уже два файла

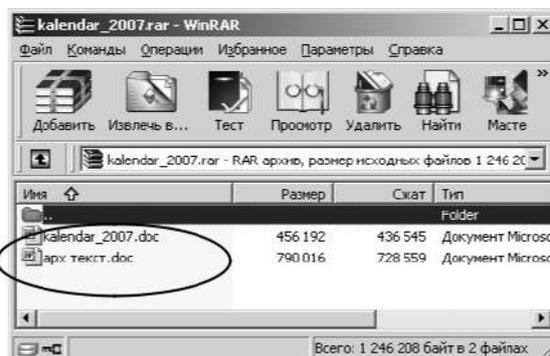


Рис.4.8. Продолжение

Если необходимо **удалить** некоторые файлы из архива, то необходимо:  
21. окне архиватора открыть архив и просмотреть, какие файлы входят в его состав.

Выделить ЛКМ удаляемый файл и нажать кнопку "Удалить" на панели инструментов архиватора, при этом из состава архива удаляется выбранный файл.

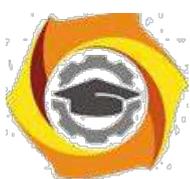
### *Задание*

#### **Создать самораспаковывающийся архив**

14. Создать самораспаковывающийся архив из файла **Word.doc**. (Архив должен быть создан в папке **АрхивФИО**).
15. Извлечь файлы из созданного архива (файлы должны сразу оказаться на Рабочем столе)

#### **Изменить состав архива**

15. В созданный при выполнении предыдущих задач архив **music.rar** добавьте любые 1-2 файла.
16. В созданный самораспаковывающийся архив добавьте любые 1-2 файла.
17. Просмотрите содержимое получившегося архива и извлеките из него только один из входящих в его состав файлов.
18. Удалите из состава этого архива любой файл.



**Задание 4.2. Исследовать сжимаемость файлов различных типов**

14. Создать непрерывные архивы файлов: **music.mp3**, **video.wmv**, **text.txt**, **Word.doc**, **XP.gif** (архивы должны быть созданы сразу в папке **АрхивФАМ**). Метод сжатия – **максимальный**, размер словаря – везде одинаковый.

15. Записать в конспект размеры: исходного файла и сжатого файла во всех созданных архивах.

16. Результаты оформить в видета блицы Excel с использованием формулы расчёта коэффициента сжатия:

Имя файла	Исходный размер (байт)	Размер архива (байт)	Коэфф. сжатия
-----------	------------------------	----------------------	---------------

$$\text{Коэфф.сжатия} = \text{Исходный размер}/\text{Размер архива}$$

**В отчете представить:**

- У Таблицу эксперимента в формате Excel;
- У Графическое представление результатов в Excel;
- У Выводы о причине таких результатов.

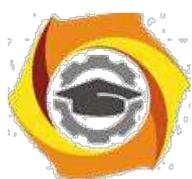
**Задание 4.3. Исследовать влияние метода сжатия на коэффициент сжатия и время сжатия**

1. Создать непрерывные архивы файла music.mp3, изменяя методы сжатия: **скоростной**, **быстрый**, **обычный**, **хороший**, **максимальный**. Время сжатия показывается в процессе создания архива.
2. Записать: исходный размер файла, метод сжатия, время сжатия и размер архива.
3. Результаты оформить в виде таблицы Excel с использованием формулы расчёта коэффициента сжатия:

Метод сжатия	Исходный размер(байт)	Размер архива(байт)	Коэфф. сжатия	Время сжатия(с)
--------------	-----------------------	---------------------	---------------	-----------------

**В отчете представить:**

11. Таблицу экспериментна в формате Excel;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

12. Графическое представление результатов в Excel;



' Выводы о причине таких результатов.

***Задание 4.4. Исследовать влияние размера словаря на коэффициент сжатия***

Создать непрерывные архивы файла music.mp3, изменяя размер словаря: **64, 128, 256, 512, 1024, 2048, 4096**.

Записать: исходный размер файла, размер словаря, размер архива.

Результаты оформить в виде таблицы Excel с использованием формулы расчёта коэффициента сжатия:

Размер словаря(кБ)	Исходный размер(байт)	Размер архива(байт)	Коэфф.сжатия
--------------------	-----------------------	---------------------	--------------



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

***В отчете представить:***

- Таблицу эксперимента в формате Excel;
- Графическое представление результатов в Excel;
- Выводы о причине таких результатов.

***Вопросы для самопроверки***

- В Для чего создаются архивы?
- В Что такое самораспаковывающийся архив?
- В Перечислите различные способы запуска архиватора 7Zip.
- В Какое расширение имеет архив, созданный стандартным архиватором Windows?
- В Перечислите, какие архиваторы Вам известны.
- В Какие файлы поддаются более эффективному сжатию? Почему?
- В Какие параметры архиватора настраиваются?
- В Поясните отличие операции создания архива и добавления в архив.
- В Какие расширения имеют архив и самораспаковывающийся архив?
- В Возможна ли пересылка архива по электронной почте?
- В Как защитить данные архива?
- В Что такое словарь архива и на что может повлиять его размер?
- В Практически все архиваторы обеспечивают возможность просмотра файлов, содержащихся в архивах. Почему эти файлы нельзя редактировать?

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ для проведения  
лабораторных работ по дисциплине «Системы  
искусственного интеллекта»**

***Лабораторная работа №1.***

***Задача линейного разделения двух классов. Линейное разделение классов.  
Алгоритм обучения перцептрона по отдельным примерам. Геометрическая  
интерпретация линейного разделения классов. Настройка весового вектора.***

**Аннотация:** Рассматриваются многослойные рекуррентные сети (перцептронная сеть обратной связью, рекуррентная сеть Эльмана, сеть RTRN) и их использование для идентификации динамических объектов.

***Введение***

Многослойные *рекуррентные сети* представляют собой развитие однонаправленных сетей перцептронного типа за счет добавления в них соответствующих обратных связей. Обратная связь может исходить либо из выходного, либо из скрытого слоя нейронов. В каждом контуре такой связи



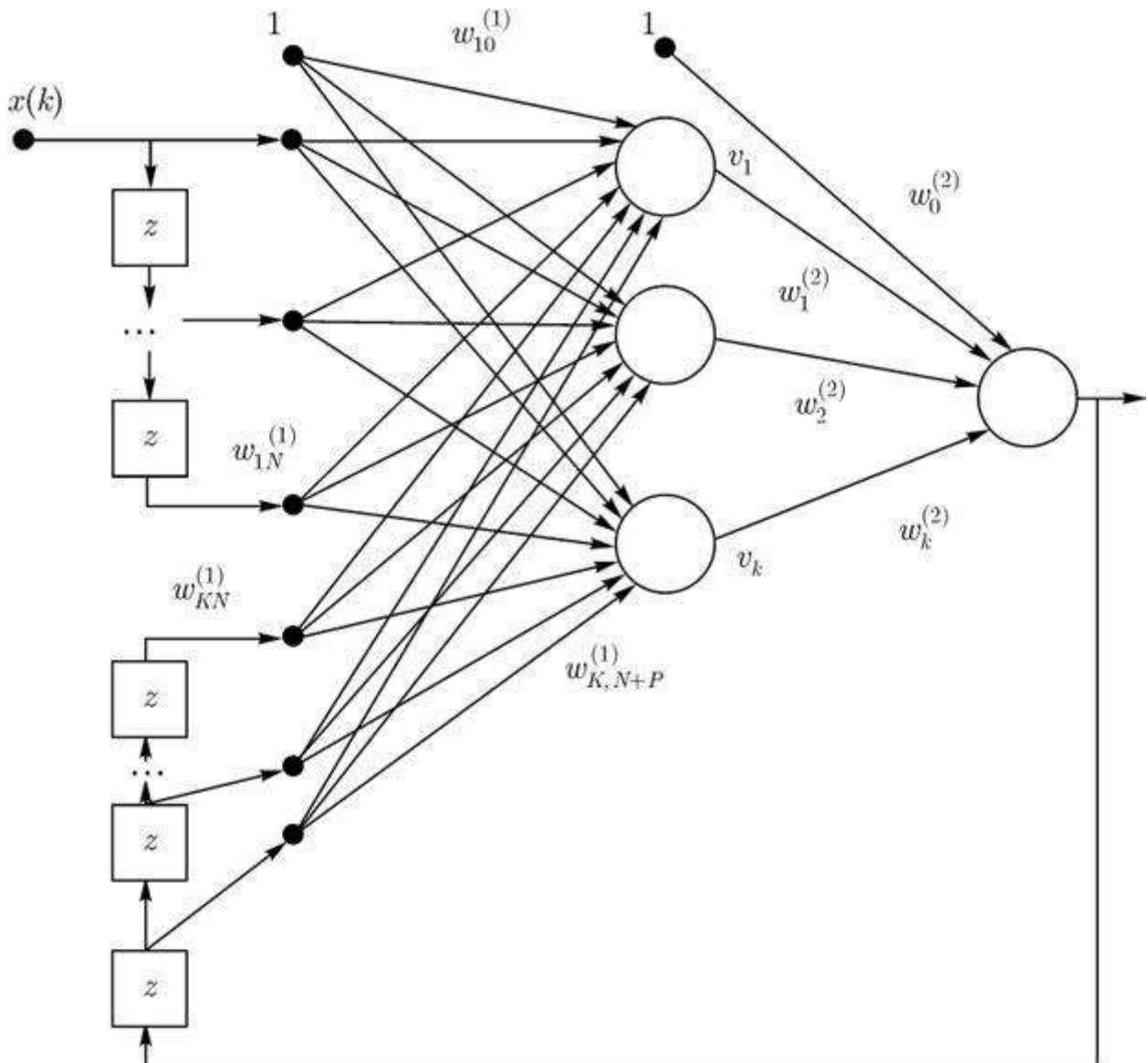
## Негосударственное частное образовательное учреждение высшего образования «Технический университет»

присутствует элемент единичной задержки, благодаря которому *поток* сигналов может считаться однонаправленным (выходной сигнал предыдущего временного *цикла* рассматривается как априори заданный, который просто увеличивает *размерность* входного вектора сети). Представленная подобным образом *рекуррентная сеть*, с учетом способа формирования выходного сигнала, функционирует как однонаправленная персептронная *сеть*. Тем не менее, *алгоритм* обучения такой сети, адаптирующий значения синаптических весов, является более сложным из-за зависимости сигналов в момент времени  $t$  от их значений в предыдущие моменты и соответственно из-за более громоздкой формулы для расчета вектора градиента.

При обсуждении *рекуррентных сетей*, в которых в качестве выходного элемента используется многослойный *персептрон*, рассмотрим наиболее известные структуры сетей *RMLP*, *RTRN*, Эльмана.

### ***Персептронная сеть с обратной связью***

Один из простейших способов построения рекуррентной сети на базе однонаправленной НС состоит во введении в персептронную *сеть* обратной связи. В дальнейшем мы будем сокращенно называть такую *сеть RMLP* (англ.: *Recurrent MultiLayer Perceptron* - рекуррентный многослойный *персептрон*). Ее обобщенная структура представлена на рис. 1 (  $\mathcal{Z}$  - единичные элементы запаздывания).



[увеличить](#)

[изображение](#)

Рис. 1. Структура сети RMLP

Это динамическая сеть, которая характеризуется запаздыванием входных и выходных сигналов, объединяемых во входной вектор сети. Рассуждения будут касаться только одного входного узла  $x(k)$  и одного выходного нейрона, а также одного скрытого слоя. Такая система реализует отображение:

$$y(k+1) = f(x(k), x(k-1), \dots, x(k-(N-1)), y(k-1), \dots, y(k-1)) \quad (1)$$

где  $N - 1$  - количество задержек входного сигнала, а  $P$  - количество задержек выходного сигнала. Обозначим  $K$  количество нейронов в скрытом слое. В этом



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

случае сеть *RMLP* можно характеризовать тройкой чисел  $(N, P, K)$ . Подаваемый на вход сети вектор  $x$  имеет вид:



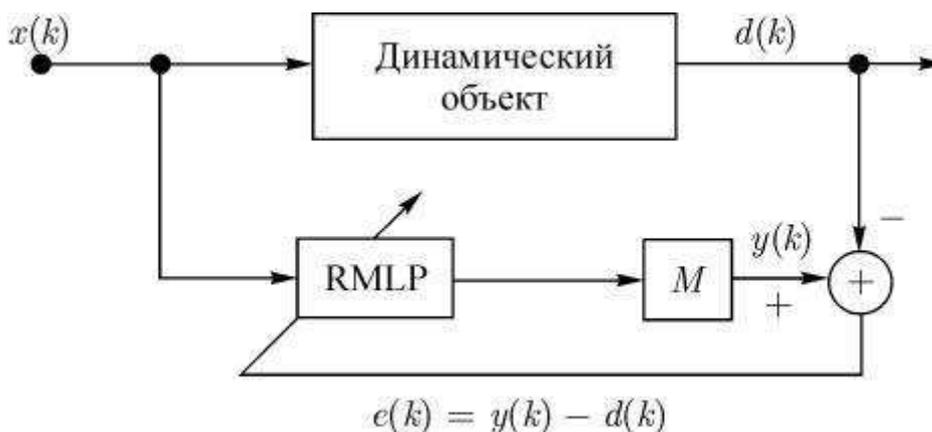
$$x(k) = [1, x(k), x(k-1), \dots, x(k-(N-1)), \\ y(k-P), y(k-P+1), \dots, y(k-1)]^T.$$

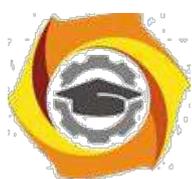
Допустим, что все нейроны имеют сигмоидальную функцию активации. Обозначим  $u_i$  взвешенную сумму сигналов  $i$ -го нейрона скрытого слоя, а  $g$  - взвешенную сумму сигналов выходного нейрона. При введенных обозначениях выходные сигналы конкретных нейронов описываются зависимостями

$$u_i = \sum_{j=0}^{N+P} w_{ij}^{(1)} x_j \\ v_i = f(u_i) \\ g = \sum_{i=0}^K w_i^{(2)} v_i \\ y = f(g)$$

Сеть *RMLP* повсеместно применяется для моделирования динамических процессов в режиме "онлайн". Типичным примером ее приложения может служить имитация нелинейных динамических объектов, для которых *сеть RMLP* выступает в роли модели, а *алгоритм* уточнения весов - в роли процедуры идентификации параметров этой модели (рис. 2). Идентифицированная модель может в последующем использоваться для управления данным объектом. Именно по этой причине сети *RMLP* наиболее популярны для имитации систем управления машинами, устройствами и динамическими процессами.

В результате сравнения выходного сигнала модели  $y(k)$  с выходным сигналом динамического объекта  $d(k)$  рассчитывается значение погрешности  $e(k) = y(k) - d(k)$ , управляющей процессом уточнения параметров нейронной сети. Символом  $M$  на рис. 2 обозначен коэффициент усиления модуля, масштабирующего выходной сигнал сети  $y(k)$  таким образом, чтобы его динамический уровень лежал в том же диапазоне, что и уровень выходного сигнала динамического объекта  $d(k)$ .





**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Рис. 2.** Схема включения сети RMLP при решении задачи идентификации



### *Рекуррентная сеть Эльмана*

Рекуррентная сеть Эльмана характеризуется частичной рекуррентностью в форме обратной связи между скрытым и входным слоем, реализуемой с помощью единичных элементов запаздывания  $Z$ . Обобщенная структура этой сети представлена на рис. 3.

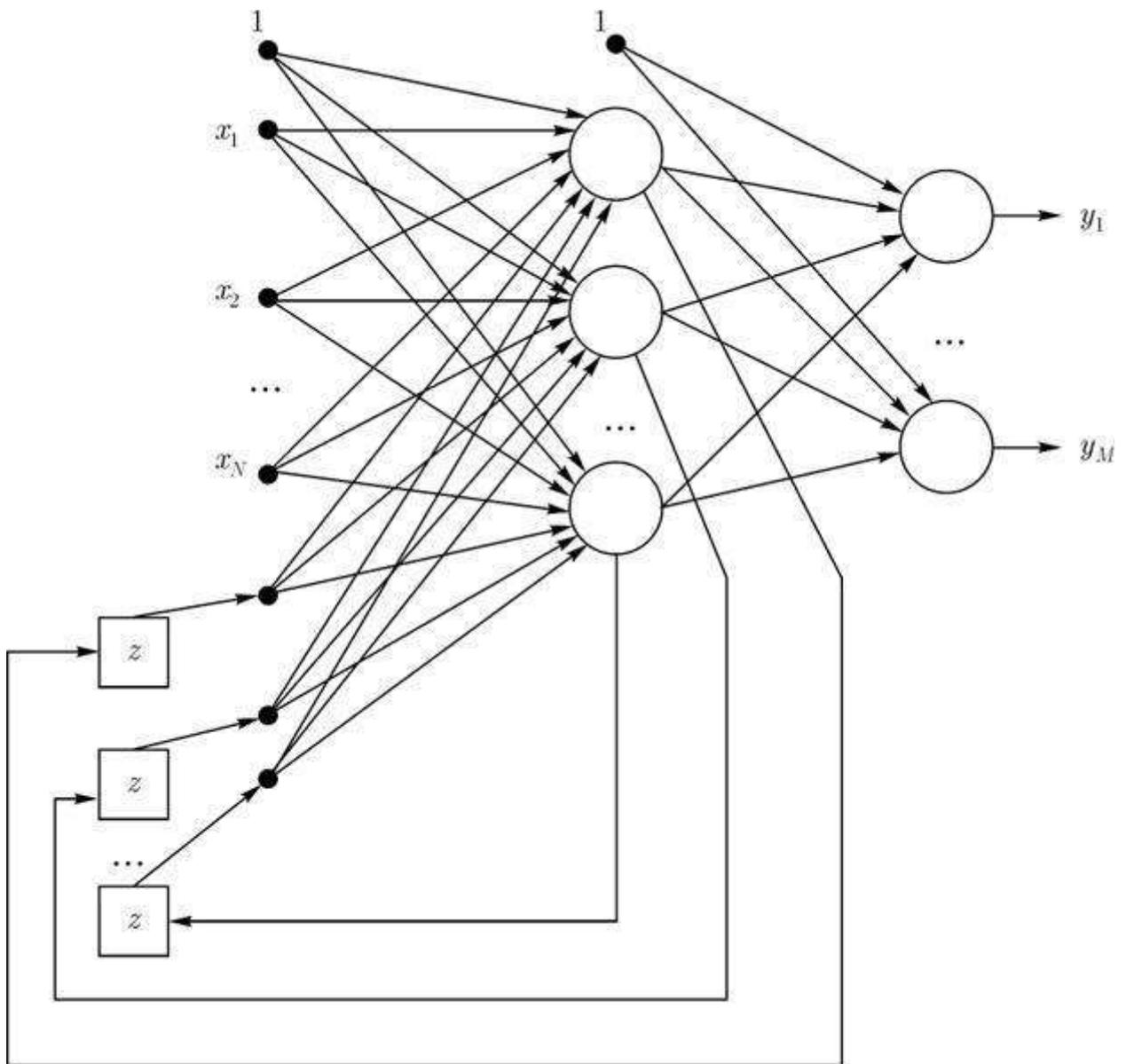
Каждый скрытый *нейрон* имеет свой аналог в контекстном слое, образующем совместно с внешними входами сети *входной* слой. *Выходной слой* состоит из нейронов, однонаправленно связанных только с нейронами скрытого слоя, подобно сети *RMLP*. Обозначим внутренний *вектор* возбуждения сети  $X$  (в его состав входит также единичный сигнал

поляризации), состояния скрытых нейронов -  $v \in R^K$ , а выходные сигналы сети -  $y \in R^M$ . При таких обозначениях *входной вектор* сети в момент  $t$  имеет форму

$$X(k) = [x_0(k), x_1(k), \dots, x_N(k), v_1(k-1), v_2(k-1), \dots, v_K(k-1)].$$

Веса синаптических связей первого (скрытого) слоя сети обозначим  $w_{ij}^{(1)}$ , а второго (выходного) слоя -  $w_{ij}^{(2)}$ . Если взвешенную сумму  $i$ -го нейрона скрытого слоя обозначить  $u_i$ , а его выходной сигнал -  $v_i$ , то

$$u_i(k) = \sum_{j=0}^{N+K} w_{ij}^{(1)} x_j(k),$$
$$v_i(k) = f_1(u_i(k)).$$



[увеличить](#)

[изображение](#)

Рис. 3. Структура сети Эльмана

Веса  $w_{ij}^{(1)}$  образуют матрицу  $W^{(1)}$  синаптических связей скрытого слоя, а  $f_1(u_i)$  - функция активации  $i$ -го нейрона этого слоя. Аналогично можно обозначить взвешенную сумму  $i$ -го нейрона выходного слоя  $g_i$ , а соответствующий ему выходной сигнал сети -  $y_i$ . Эти сигналы описываются формулами

$$g_i(k) = \sum_{j=0}^K w_{ij}^{(2)} v_j(k),$$
$$y_i(k) = f_2(g_i(k)).$$

В свою очередь, веса  $w_{ij}^{(2)}$  образуют матрицу  $W^{(2)}$ , описывающую синаптические связи



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

нейронов выходного слоя;  $f_2(g_i)$  - функция активации  $i$ -го нейрона выходного слоя.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

8

*Сеть* Эльмана естественным образом предназначена для моделирования временных рядов.

34. частности, она решает задачу прогнозирования амплитуды сигнала на основе текущего значения входного сигнала и запомненных значений из предыдущего временного цикла. Задача прогноза временных рядов возникает в финансовой области: прогноз котировок товаров и ценных бумаг на бирже, курсов валют, показателей банковской деятельности. В экономике прогноз может быть связан, например, с анализом уровня производства в различных отраслях промышленности и сельского хозяйства, процента трудовой занятости населения, роста средней заработной платы. Проблема прогноза временных рядов возникает в многочисленных экологических задачах и самых разнообразных технических приложениях.

Для прогноза временных рядов могут применяться *статистические методы*. В этом случае должна быть построена *динамическая модель* данных (например, регрессионная модель) изучаемого явления. Для простейших задач такая модель может быть построена известными методами. Однако для практических задач, примеры которых приведены выше, построение подобной динамической модели представляет собой сложную аналитическую задачу. Эти приложения связаны обычно не со скалярными, а с векторными временными рядами. Например, в финансовой сфере прогноз котировок товара зависит от вектора динамических данных, которые включают цены открытия и закрытия торговой сессии, среднюю и максимальную цены торговой сессии, суммарный уровень заявок, *валютные курсы* и пр.

В том случае, когда адекватной математической модели изучаемых временных рядов не существует, удобным инструментом для решения задачи прогноза является нейросетевой экстраполятор динамических данных.

Задача прогноза векторного временного ряда ставится следующим образом:

• задана реализация временного ряда  $x_j = (x_{j1}, x_{j2}, \dots, x_{jM})$ , на интервале времени  $[\Delta, T\Delta]$  с постоянным интервалом дискретности  $\Delta$ ;

21. требуется построить оценку значения временного ряда (обычно одной его координаты) в момент времени  $(T + t_{pr})\Delta$ , где  $t_{pr}\Delta$  - заданное время прогноза.

Из логических соображений или путем статистического анализа имеющейся реализации можно установить, сколько предшествующих значений относительно произвольного текущего момента времени  $j\Delta$  определяюще связаны с прогнозируемым значением. Это означает, что если представить прогнозируемое значение  $y_j = x_{j+t,mm}$  -ой координаты вектора  $x$  как функцию его предшествующих измерений:

$$y_j = \Phi(x_j, x_{j-1}, \dots, x_{j-q+1}),$$

то выбор значения  $q$  устанавливает "память" экстраполятора. Значение  $q$  определяет также входной вектор для нейронной сети, которая строится для решения задачи прогноза. Размерность этого вектора равна  $M * q$ .

Таким образом, задача прогноза данных на нейронной сети сведена к задаче



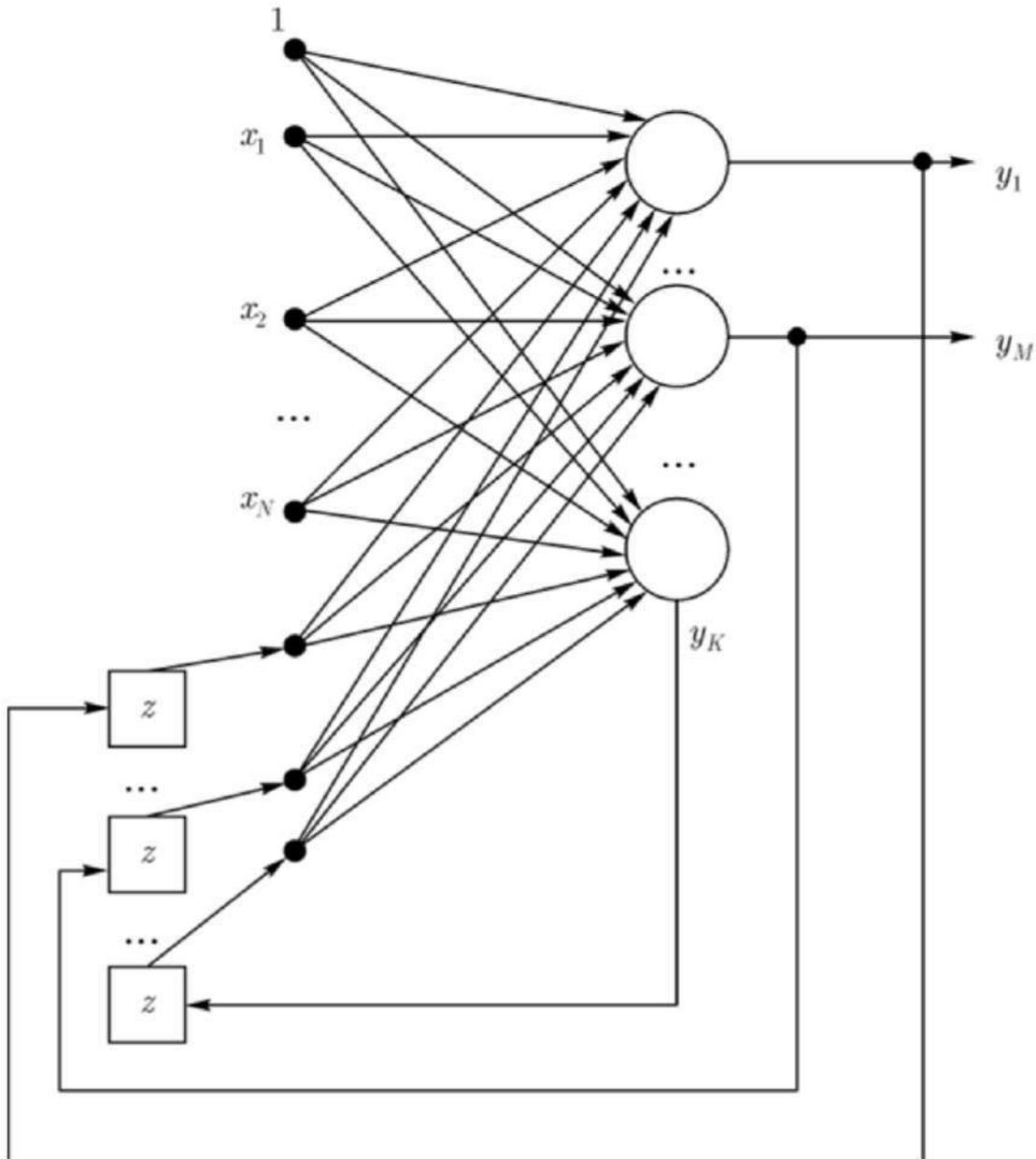
**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

воспроизведения функции многих переменных  $\Phi(x_j, x_{j-1}, \dots, x_{j-q+1})$  по обучающей выборке. данным



### Сеть RTRN

Среди рекуррентных сетей особую внимания заслуживает сеть типа RTRN (англ.: *Real Time Recurrent Network*), предложенная Р.Вильямсом и Д.Зипсером и предназначенная для обработки сигналов в реальном времени. Сеть RTRN - частный случай сети Эльмана.



[увеличить](#)

[изображение](#)

Рис. 4. Структура сети RTRN

Обобщенная структура сети представлена на рис. 4. Сеть содержит  $N$  входных узлов,  $K$  скрытых нейронов и  $K$  соответствующих им узлов контекстного слоя. Из  $K$  скрытых нейронов только  $M$  составляют выход сети. Обозначим взвешенную сумму  $i$ -го нейрона скрытого слоя  $u_i$ , а выход этого нейрона  $y_i$ . Вектор  $x(k)$  и смещенный



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

(задержанный) на один цикл *вектор*  $u(k - 1)$  образуют расширенный *вектор* активации  $x(k)$ , возбуждающий нейроны сети:



$$x(k) = [1, x_1(k), x_2(k), \dots, x_N(k), y_1(k-1), \dots, y_K(k-1)]^T.$$

После описания входного вектора сети в момент  $t$  можно определить состояние всех нейронов согласно зависимостям:

$$u_i(k) = \sum_{j=0}^{N+K} w_{ij} x_j(k)$$

$$y_i(k) = f(u_i(k))$$

причем  $f()$  обозначает непрерывную функцию активации нейрона (как правило, сигмоидальную). На рис. 4 видно, что сеть *RTRN* представляет собой частный случай сети Эльмана, в которой веса выходного слоя постоянны и равны дельте Кронекера,

т.е.  $w_{ij} = \delta_{ij} = 1$  для  $i = j$  или  $0$  для  $i \neq j$ . В этом случае можно применять алгоритм обучения Вильяма-Зипсера.

/ Выбрать случайные начальные значения весов сети, составляющих матрицу  $W$  и равномерно распределенных в заданном интервале (обычно в диапазоне от  $-1$  до  $1$ ).

/ Рассчитать состояние всех  $K$  нейронов для очередного момента  $t = 0, 1, 2, \dots$  с использованием формул (1) и (2). На этой основе можно определить входной вектор  $x(k)$ , возбуждающий нейроны в момент  $t$ .

3. Рассчитать значения

$$dy_i(k)/dw_{ab} = (df_1(u_i)/du_i)[\delta_{ja}x_b + \sum_{k=1}^K (dy_i(k-1)/dw_{ab})w_{i,k+N}]$$

4. Уточнить значения весов по алгоритму наискорейшего спуска согласно формуле

$$w_{ab}(k+1) = w_{ab}(k) - \alpha \sum_{i=1}^K [y_i(k) - d_i(k)](dy_i(k)/dw_{ab})$$

для  $a = 1, 2, \dots, K$  и  $b = 0, 1, 2, \dots, N + K$ .

Шаги (2-4) повторять вплоть до стабилизации значений всех весов сети.

#### Контрольные вопросы ЛР (ПК-4):

- / Функциональная структура использования СИИ.
- / Модели и методы решения задач.
- / Классификация представления задач.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

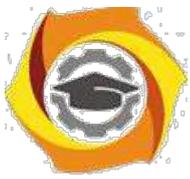
- / Логические модели.
- / Нейрофизиологическая аналогия.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

11

- / Реализация булевых функций нейронными сетями.
  - / Каково назначение основных элементов управления Lazarus?
  - / Какие существуют типы элементов управления Lazarus?
  - / Каково назначение свойств элементов управления в Lazarus?
  - / Как можно изменить свойства элементов управления?
  - / Как влияют свойства на интерфейс программы?
  - / Понятие информационной системы. Классы ИС.
  - / Структура однопользовательской и многопользовательской, малой и корпоративной ИС.
  - / Основные особенности современных проектов ИС.
  - / Как называются информационные системы, ориентированные на коллективное использование информации членами рабочей группы и чаще всего строящиеся на ЛВС.
16. Как называются информационные системы, основанные гипертекстовых документах и мультимедиа.



**Лабораторная работа №2.**

**Виды нейронных сетей и способы организации их функционирования. Виды сетей. функционирование сетей. настройка нейронных сетей для решения задач. Предобработка данных. Интерпретация ответов сети. оценка способности сети решить задачу. Сдача теста № 2.**

**Адаптивная резонансная теория (ART)**

Серьезная проблема для нейронных сетей - правильное соотношение стабильности / пластичности при запоминании образов. Существуют наборы эталонов (даже состоящие всего из 4-х векторов), которые при циклическом предъявлении в обучении дают никогда не сходящиеся наборы параметров сети. Предъявление всего одного нового образа в обучающем множестве часто приводит к долгому *переобучению*. Если *сеть* работает в реальном времени, например, обрабатывает сенсорную информацию, то обучающее множество может все время меняться. Для большинства моделей нейронных сетей это приводит к отсутствию обучения вообще.

Человеческая *память*, напротив, эффективно хранит и корректирует запоминаемые образы. Ни предъявление нового образа, ни изменение старых не приводит к уничтожению памяти или невозможности запоминания. Даже удаление части нервной ткани чаще всего не прерывает работу сети и не стирает запомненные образы, а лишь делает их менее четкими.

*Сеть ART* - попытка приблизить механизм запоминания образов в искусственных НС к биологическому. Результатом работы *ART* является устойчивый набор запомненных образов и возможность выборки "похожего" вектора по произвольному предъявленному на входе вектору. Важное качество *ART* - динамическое запоминание новых образов без полного *переобучения* и отсутствие потерь уже запомненных образов при предъявлении новых.

**Сеть ART-1**

*Сеть ART-1* предложена Карпенгером и Гроссбергом в 1986 г. Она представляет собой векторный *классификатор* и обучается без учителя, лишь на основании предъявляемых входных векторов. *ART-1* работает только с двоичными векторами, состоящими из нулей и единиц. Позже было предложено много разновидностей этой модели. *ART-2* запоминает и классифицирует непрерывные входные векторы. *Группа* моделей с суффиксом "*MAP*" (*ARTMAP* и др.) классифицирует и входные, и выходные вектора, а также строит связи между ними.

**Архитектура и работа**

Структура сети *ART-1* (далее *ART*) представлена на рис. 1. Входной вектор сети распознавания запоминается каждый нейрон  $m = 1, \dots, M$  имеет  $N$  компонент. В слое классов образов, по одному классу на

Основную работу по классификации производят слой сравнения и слой распознавания. Схемы приемников (Прм1, Прм2) и схема сброса управляют режимом работы сети и могут быть реализованы в виде обычных логических схем или в виде нейронов.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Работа блоков *APT* определяется следующими формулами:

$$\text{Прм1} : G1 = (\forall_n x_n) \wedge \neg(\forall_m R_m).$$



Выход Прм1 обеспечивает единичный сигнал для слоя сравнения, если на вход сети подан вектор (нулевой вектор на входе недопустим) и если выход слоя распознавания равен нулю.

$$\text{Прм2} : G2 = \bigvee_n x_n$$

Если на вход подан вектор  $x$ , то блок Прм2 формирует на выходе единичный сигнал и тем самым разрешает работу слоя распознавания.

Схема сброса:  $G3 = (\sum_n C_n / \sum_n x_n) < \rho$ .

Проверяет критерий сходства для векторов  $x$  и  $C$ . Критерий состоит в сравнении количества единиц в векторах  $x$ ,  $C$ . Количества единиц сравниваются в виде отношения с некоторым пороговым уровнем сходства  $\rho$ . Если порог не превышен, то сходство считается плохим и схема сброса вырабатывает сигнал торможения для нейрона в слое

распознавания. Выход схемы сброса - двоичный вектор с компонентами. Схема сброса является динамической и "помнит" свое состояние в течение одной классификации. Порог  $\rho$  является внешним параметром по отношению к сети и задается пользователем в интервале от 0 до 1. Чем меньше  $\rho$ , тем менее похожие векторы будут отнесены сетью к одному классу.

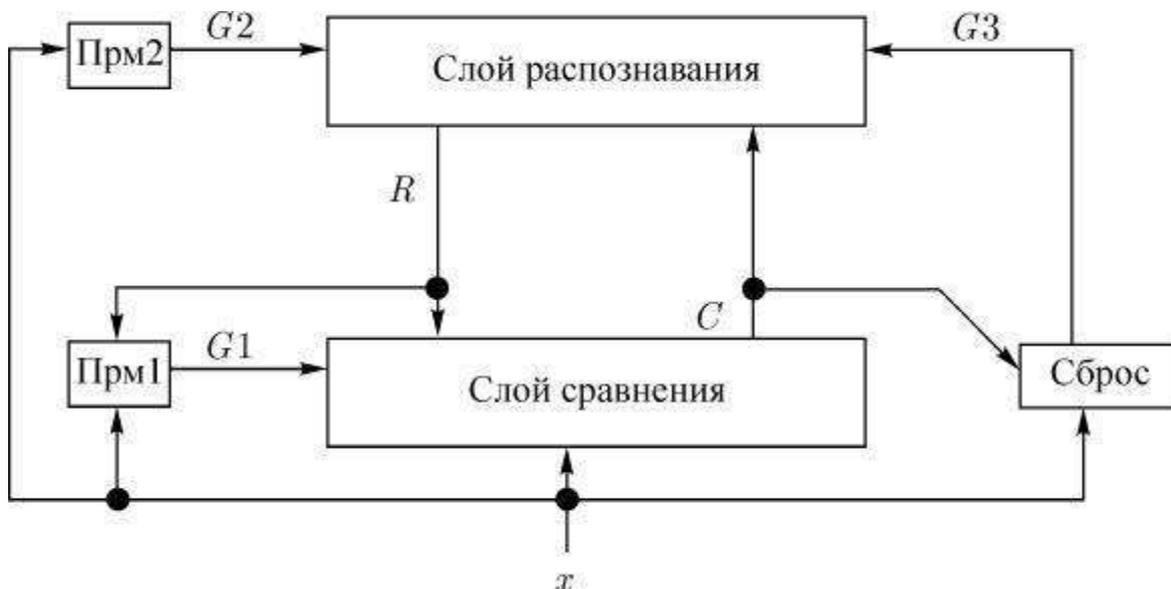
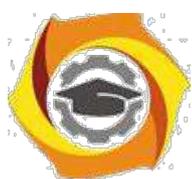


Рис. 1. Структурная схема АРТ

### Слой сравнения

Каждый нейрон в слое сравнения имеет порог, равный двум. На вход одного нейрона в слое сравнения подаются: сигнал  $G1$  с единичным весом, одна компонента  $x^n$  с единичным весом и все выходы слоя распознавания,  $M$  компонент с вектором весов  $T^n$ , где  $n$  - номер нейрона в слое сравнения. Весовые коэффициенты  $T$  - двоичные. В нейроне используется нелинейность в виде жесткой ступеньки: если активация нейрона  $NET_n$  превышает порог  $\Theta = 2$ , то на выходе нейрона будет единица, иначе - ноль. Это "правило 2/3": для активации нейрона достаточно два сигнала из трех.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Работа слоя определяется формулами:



$$P_n = T^n R = \sum_m T_m^n R_m$$

$$NET_n = P_n + x_n + G1$$

$$C_n = \{0, \text{ если } NET_n < 2; 1, \text{ если } NET_n \geq 2\}$$

Работой слоя управляет сигнал  $G1$ . Если  $G1 = 1$ , то  $x$  проходит без изменений на выход слоя сравнения, благодаря наличию единичного сигнала  $G1$  на входе нейрона. Если  $G1 = 0$ , то на выходе имеем  $C = x \wedge P$ , т.е. вектор  $C$  будет логическим произведением двоичных векторов  $x$  и  $P$ .

### Слой распознавания

Каждый нейрон в слое распознавания имеет следующие входывходящий сигнал  $G2$  с единичным весом, один компонентанта  $G3_m$  с большим отрицательным весом ( $m$  - номер нейрона) и  $N$  сигналов со слоя сравнения с вектором весов  $B^m$  (у вектора  $B^m$  всего  $N$  компонент,  $B_1^m, \dots, B_N^m$ ).

Нейроны слоя распознавания не содержат нелинейных элементов, но обладают следующей особенностью: Каждый нейрон в слое связан со всеми остальными нейронами этого же слоя обратными тормозящими связями и положительной обратной связью с самим собой (как во втором слое сети Хемминга, см. Лекцию 10).

Такой способ связности называется латеральным торможением. Это приводит к тому, что только один нейрон в слое распознавания может быть активирован. Между нейронами существует конкуренция, и нейрон с максимальным выходом "подавляет" все остальные нейроны в слое, выигрывая "состязание". Его выход становится равным единице, остальных нейронов - нулю, т.е. вектор  $R$  имеет только одну единичную компоненту, остальные - нули.

Веса  $B^m$  имеют действительные значения. Работа слоя определяется формулой:

$$R_m = f(B^m, C) \wedge G2 \wedge \neg(G3_m),$$

где  $f(B^m, C)$  - выход  $m$ -го нейрона, равный нулю или единице.

Отсюда видно, что сигнал  $G2$  "разрешает" работу слоя распознавания, а сигнал  $G3$  позволяет выборочно затормозить любые нейроны в слое.

### Работа сети АРТ

Решение задачи классификации с помощью АРТ содержит следующие этапы: инициализация, распознавание, сравнение, поиск, обучение.

В Инициализация.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

а) выбираем *параметр*  $\rho$ , исходя из требуемой детальности классификации;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

15

б) создаем *сеть* в памяти. Количество нейронов должно быть достаточным, чтобы запомнить все ядра классов (д). Изначально все нейроны слоя сравнения считаются "невыделенными", их веса приравняются к одинаковым небольшим значениям:

$$B_n^m = B_{нач} < L / (L + N - 1),$$

где  $L > 1$  - некоторая константа (обычно  $L = 2$ ). Все в слое сравнения также выбираются одинаковыми и равными единице:  $T_m^n = 1$ .

Такой выбор весов обеспечивает возможность поиска на невыделенном нейроне, если нет подходящих выделенных нейронов, и правильное обучение.

**Распознавание.**

а) предъявляем *вектор*  $x$  на входе. До этого момента  $G2 = 0$  и *выход* слоя распознавания равен нулю:  $R = 0$ .

б) *вектор*  $x$  есть ненулевой компонент, поэтому  $G1$  становится равным единице т.к.  $R = 0$ . Сигнал  $G1$  "подпитывает" нейроны слоя сравнения и  $x$  без изменений проходит через слой сравнения:  $C = x$ .

в) *весовые коэффициенты*  $B_n^m$  имеют нормированные ядра классов в слое распознавания активируются несколько нейронов, благодаря *двухзначности* порога остается один *нейрон* с *выходом*  $R_{m0} = 1$ , а остальные *гаснут*.  $m_0$  - номер *выш* правшего нейрона.

**3. Сравнение.**

а) *выход*  $R \neq 0$  приводит к  $G1 = 0$ , что снимает "подкачку" нейронов в слое сравнения. *Весовые коэффициенты*  $T_m^n$  имеют смысл *нормированных* двоичных ядер классов. На вход слоя сравнения передается один ненулевой *вход* слоя распознавания.  $R_{m0} = 1$ . *единица* умножается на *весовые коэффициенты*, давая в сумме сигнал

$$NET_n = x_n + T_{m_0}^n.$$

Порог всех нейронов равен 2, поэтому *выход* слоя сравнения равен

$$C_n = x_n \wedge T_{m_0}^n.$$

Следовательно, *выход* слоя сравнения на этом этапе - логическое *произведение* входного сигнала и двоичного ядра класса из слоя сравнения.

б) *модуль сброса* вычисляет второй критерий сходства (первый - максимум произведения



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

(  $B^m, x$  ) в слое распознавания). Если количества единиц в векторе  $C$  и векторе  $x$  близки, то сходство считается хорошим и выносится решение о принадлежности вектора  $x$  к классу  $m0$ .

41. *Поиск.*



а) если критерий сходства не выполняется, схема сброса вырабатывает сигнал  $G3_{m0} = 1$ , который тормозит нейроны  $m_0$  в слое распознавания. Сигнал  $G3_{m0} = 1$  составляет равным 1 до окончания данной классификации. Выход нейрона  $m_0$  становится равным 0, а, следовательно, и весь вектор  $R = 0$ . Сигнал  $G1$  становится равным нулю и вектор  $x$  снова проходит через слой сравнения без изменений, вызывая новый цикл поиска (шаги 2в-3б), пока критерий сходства не будет удовлетворен.

При соответствующем выборе начальных значений весов  $B$  поиск всегда закончится на нераспределенном нейроне слоя распознавания. Для него будет выполнен критерий сходства, т.е. все веса равны 1. Если все нейроны в слое сравнения критерий сходства не выполняется, следует аварийная остановка либо расширение сети введением нового нейрона в слое распознавания и новых входов в слое сравнения.

### 5. Обучение.

Независимо от того, найдены ли на этапе поиска распределенный нейрон или нераспределенный, обучение происходит следующим образом: корректируются веса выходящих из этого нейрона  $m_0$  в слое распознавания и веса  $T_{m0}^n$  для всех  $n$  в слое сравнения.

Различают быстрое и медленное обучение. При быстром обучении коррекции весов имеют вид:

$$B_n^{m0} = LC_n / (L + \sum_n C_n - 1) \equiv \Delta B_n^{m0},$$

где  $L > 1$  - константа.

Веса в слое сравнения - двоичные:  $T_{m0}^n$ .

В результате такого алгоритма обучения ядра  $T$  изменяются, несущественные компоненты обнуляются в процессе обучения. Если какой-то компонент вектора  $T_{m0}^n$  стал нулевым на какой-то итерации обучения, она никогда не вернется к единице. В этом проявляется асимметрия АРТ по отношению к значениям 0 и 1. Эта асимметрия имеет серьезные отрицательные последствия для модели, приводя к деградации ядер классов в случае зашумленных входных векторов в случае мленных входных векторов.

Медленное обучение меняет ядра малыми коррекциями:

$$B_n^{m0} \rightarrow \beta \Delta B_n^{m0} + (1 - \beta) B_n^{m0},$$

$$T_{m0}^n \rightarrow \beta C_n + (1 - \beta) T_{m0}^n,$$

где  $\beta$  мало и характеризует скорость обучения.

В результате каждой итерации обучения ядра меняются незначительно.

Видно, что веса  $B$  в любой момент времени могут быть однозначно рассчитаны через  $T$ . Таким образом, кодирование информации о ядрах в АРТ в рассмотренной модели является



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

веса , таким образом, *кодирование* информации о ядрах в *APT* в рассмотренной модели является избыточным в смысле расхода памяти.



### **Необходимость поиска**

В сети *APT* используются два критерия "похожести" векторов. Первый - максимум скалярного произведения  $\max_m(B^m, x)$  при выборе "победителя" в слое распознавания. Второй - критерий сходства в блоке сброса:

$$\left( \sum_n C_n / \sum_n x_n \right) \Big|_{x,C} \geq \rho.$$

Таким образом, задача классификации в сети *APT* состоит в том, чтобы найти ядро с максимальным скалярным произведением  $(B^m, x)$ , соблюдая при этом условие выполнения критерия сходства. Эти два критерия не являются эквивалентными, поэтому и фаза поиска, и фаза распознавания являются необходимыми и не могут быть опущены.

### **Положительные качества и недостатки *APT***

Сеть *APT* решает дилемму стабильности-пластичности и позволяет быстро запоминать новые образы без утраты старых. Как и в случае других моделей НС, на обычных машинах фон-неймановского типа сети работают медленно и неэффективно. Для решения задачи нужно найти максимум скалярного произведения, что требует около  $3NM$  операций с плавающей запятой, и вычислить в худшем случае  $M$  критериев сходства. Для этого необходимы существенные вычислительные затраты. На параллельном компьютере операции расчета скалярных произведений могут быть распараллелены, но расчет критериев сходства все равно выполняется последовательно. Таким образом, даже на параллельной машине сеть *APT* является требовательной к ресурсам.

Тем не менее, одна итерация для запоминания каждого входного вектора - редкая экономичность для нейронных сетей. Вспомним, что многослойный перцептрон для запоминания нового вектора требует полного переобучения.

39. сети *APT* есть несколько существенных недостатков.

Чувствительность к порядку предъявления векторов. Большинство разновидностей *APT* весьма чувствительны к порядку предъявления входных векторов  $x$ . Картины ядер классов, сформированные сетью, принципиально меняются при различных видах упорядочения.

Невозможность классификации зашумленных векторов. Пусть входные векторы содержат шум.

Если компонента незашумленного входного вектора равна  $x_n$ , то предъявленные сети значения будут определяться вероятностным законом:

$$p(x_n) = 1 - \varepsilon,$$

$$p(-x_n) = \varepsilon,$$

где  $\varepsilon$  - малое положительное число, характеризующее уровень шума.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Если такие данные будут предъявлены *APT*, то будет наблюдаться деградация и размножение классов. Если сетью сформировано правильное *ядро* для класса, к которому



относится вектор  $\mathcal{X}$ , то как только компонента  $\mathcal{X}_n$  примет нулевое значение за счет шума (если векторы предъявляются не однократно), соответствующая компонента ядра также будет обнулена. Т.к. случайное нулевое значение может принять любая компонента  $\mathcal{X}$ , то с течением времени все компоненты ядра будут обнулены, запомненная информация об этом классе - утрачена. Если после этого предъявить незашумленный вариант вектора  $\mathcal{X}$ , то для него будет выделен новый нейрон, т.е. сформирован новый класс. Это явление называется размножением классов. Через некоторое время в сети будет множество нейронов с нулевыми весами, и все нейроны будут распределены. Работа сети прекратится. Это явление определяется исходной асимметрией алгоритмов *APT* относительно значений 0 и 1. Существуют методы для устранения асимметрии и предотвращения размножения классов.

**Контрольные вопросы ЛР (ПК-4):**

43. Виды нейронных сетей и способы организации их функционирования.
44. Виды сетей. Функционирование сетей.
45. Настройка нейронных сетей для решения задач.
46. Интерпретация ответов сети.
47. Оценка способности сети решить задачу.
48. Цели и задачи предпроектной стадии создания ИС.
49. Модели деятельности организации ("как есть" и "как должно быть").
50. Типовое проектирование ИС. Понятие типового проекта, предпосылки типизации.
51. Объекты типизации. Методы типового проектирования
52. Оценка эффективности использования типовых решений.
53. Типовое проектное решение (ТПР).
54. Классы и структура ТПР.
55. Основные понятия организационного бизнес-моделирования.
56. Миссия компании, дерево целей и стратегии их достижения.
57. Статическое описание компании: бизнес-потенциал компании.
58. Функционал компании.



**Лабораторная работа №3**

**Задача нелинейного разделения двух классов. Метод максимума правдоподобия. нейрофизиологическая аналогия. Реализация булевых функций нейронными сетями. Выделение выпуклых областей сдача теста № 3..**

**Значимость параметров и сигналов. Сокращение описания (контрастирование) сетей.**

Сокращение множества параметров и входных сигналов обученной нейронной сети преследует цели:

35. упрощение специализированных устройств;
36. сокращение объема используемой памяти и увеличение быстродействия;
37. удешевление сбора данных;
38. обеспечение (или облегчение) интерпретации результатов обработки данных.

Существует два способа сокращения (редукции) описания:

№ редукция "снизу вверх" - постепенное удаление параметров от наименее значимых к более значимым;

№ редукция "сверху вниз" - выделение наиболее значимых параметров и постепенное дополнение их менее значимыми.

Способ редукции "снизу вверх":

62. определяются наименее значимые параметры и устраниаются вместе с соответствующими элементами системы;

63. оставшиеся параметры модифицируются так, чтобы наилучшим способом решить задачу;

64. циклически повторять пп. 1-2 до тех пор, пока задача не будет решаться с удовлетворительной точностью.

**Определение значимости параметров на основании функции оценки**

Есть набор  $x^i, i = 1, \dots, n$  размерности  $N, M$  - мерный вектор параметров  $w$  и функция оценки  $H(x, w)$ , оценивающая работу системы с параметрами  $w$  на векторе  $x$  (например, расстояние от вектора выходных сигналов системы до нужного ответа или до множества правильно интерпретируемых ответов). Требуется выделить наименее значимые параметры  $w_k, k \in \{1, \dots, M\}$  и компоненты данных  $x_j$  и модифицировать систему, отбрасывая наименее значимые параметры. Процедура отбрасывания неоднозначна. Простейший вариант - обращение в ноль - не всегда лучший: он не учитывает корреляции между данными. Учитывая корреляцию, следует отбрасываемые компоненты заменять на функции остающихся компонент.

Пусть для каждого  $w_k$  определено фиксированное значение  $w_k^0$ . Отбрасывание  $j$  -ой



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

компоненты для  $i$ -го примера означает приравнивание  $x_j := x_j^0$ . В качестве простейшего варианта примем  $w_k^0 = 0$  и для любого  $i$  полагаем



$$x_j^0 = (1/n) \sum_{p=1}^n x_j^p$$

(параметры обращаются в ноль, данные заменяются средним *по* выборке). Более тонкие методы предполагают замену отбрасываемых параметров и сигналов на некоторые функции оставшихся.

Показатели значимости вычисляются в два этапа: сначала они оцениваются для одного вектора (примера), потом для всей выборки.

1. Для данного  $x^p$  значимости  $w_k$  и  $x_j$  оцениваются как

$$\begin{aligned}\chi(w_k|x^p) &= |\partial H(x^p, w)/\partial w_k| \times |w_k - w_k^0|, \\ \chi(x_j|x^p) &= |\partial H(x^p, w)/\partial x_j| \times |x_j - x_j^0|.\end{aligned}$$

Здесь  $\chi$  - вычисленные в линейном приближении абсолютные величины изменения  $H$  при сокращении описания. Оценка на всей выборке  $x^p, p = 1, \dots, n$  может проводиться *по-разному*. Например, может использоваться одна из следующих норм:

1. Сумма модулей:

$$\begin{aligned}\chi(w_k) &= \sum_p \chi(w_k|x^p), \\ \chi(x_j) &= \sum_p \chi(x_j|x^p).\end{aligned}$$

46. *Максимум модуля*

$$\begin{aligned}\chi(w_k) &= \max_p \chi(w_k|x^p), \\ \chi(x_j) &= \max_p \chi(x_j|x^p).\end{aligned}$$

Часто приходится иметь дело с системой, которая меняет свои параметры (например, в ходе обучения). Тогда к моменту принятия решения о значимости может быть накоплена *информация* о частных производных  $H$  в разных точках  $w \in \{w_1, \dots, w_q\}$ . Ее можно использовать следующим образом.

Обозначим угловыми скобками процедуру усреднения *по* множеству параметров  $\{w^1, \dots, w^q\}$ :

$$\langle f(w, \dots) \rangle = (1/q) \sum_{i=1}^q f(w^i, \dots)$$



положим

$$\chi(w_k|x^p) = \langle |\partial H(x, w)/\partial w_k|_{\chi=\chi^p} \rangle \cdot w_k - |w_k^0|,$$
$$\chi(x_j|x^p) = \langle |\partial H(x, w)/\partial x_j|_{\chi=\chi^p} \rangle \cdot x_j^p - |x_j^{p0}|.$$

Усредняются абсолютные значения производных, а приращения берутся в тех точках, в которых будет проводиться процедура сокращения описания. Усреднение параметров  $w$  по нескольким значениям важно для нелинейных систем, в которых производные  $H$  могут сильно меняться от точки к точке.

Главная задача при сокращении описания - сохранить качество работы системы, оцениваемое с помощью  $H$ . Для этого требуется знать назначение системы и иметь способ оценки ее соответствия своему назначению.

Возможен другой подход, не предполагающий никакого знания о способах оценки. Ставится задача сохранить описание, минимально изменяя функционирование системы. В этом случае роль оценки играет изменение выходного сигнала системы после сокращения.

#### **Определение значимости параметров по изменению выходных сигналов системы**

Значимость параметров определяется практически так же, как и с помощью функции оценки.

Пусть  $x$  - вектор данных, а  $w$  - вектор параметров. Пусть задан набор векторов  $\{x^p\}$  на которых будет оцениваться функционирование системы, и определены значения

44.  $x_j^{p0}, w_k^0$  (в

простейшем случае  $w_k^0 = 0, x_j^{p0} = (1/n) \sum_{p=1}^n x_j^p$ ).

Вычислим в линейном приближении изменение вектора  $F$  при обращении  $w_k$  в  $w_k^0$  и  $x_j^p$  в  $x_j^{p0}$ :

$$\Delta F = \partial F(x, w)/\partial w_k|_{\chi=\chi^p} \times (w_k^0 - w_k),$$

$$\Delta F = \partial F(x, w)/\partial x_j|_{\chi=\chi^p} \times (x_j^p - x_j^{p0}).$$

Пусть в пространстве выходных сигналов системы задана некоторая норма (например, евклидова). Тогда положим:

$$\chi(w_k|x^p) = \|\partial F(x, w)/\partial w_k\|_{\chi=\chi^p} \times |w_k^0 - w_k|,$$

$$\chi(x_j|x^p) = \|\partial F(x, w)/\partial x_j\|_{\chi=\chi^p} \times |x_j^p - x_j^{p0}|.$$

Таким образом, для каждого  $w_k$  и любого  $x_j$  определен вектор показателей



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

значимости. *Координаты* вектора соответствуют точкам  $x^p$ . Теперь нужно вычислить норму этого вектора и объявить ее показателем значимости (можно в качестве нормы взять *максимум* модуля или сумму модулей). При использовании евклидовой нормы в пространстве выходных сигналов бывает удобно и далее выбирать такую же норму, полагая:



$$\chi(w_k) = \left( \sum_p \chi(w_k | x^p)^2 \right)^{1/2},$$
$$\chi(x_j) = \left( \sum_p \chi(x_j | x^p)^2 \right)^{1/2}.$$

Подход к определению значимости через изменение выходного сигнала не имеет альтернатив в том случае, когда рассматриваемая система является лишь подсистемой в некоторой системе (например, *сумматор* или *нейрон* в нейронной сети). Тогда при изменении параметров этой подсистемы приходится ограничиться требованием: выходной сигнал подсистемы должен изменяться как можно меньше, чтобы не нарушать функционирование всей системы.

### **Сокращение числа выходов в адаптивном линейном сумматоре (путь "снизу вверх")**

Рассмотрим адаптивный линейный сумматор, вычисляющий линейную функцию  $F(x, w) = w_0 + (x, w)$ .

Решим задачу о сокращении числа выходных сигналов. Рассмотрим *определение* значимости по изменению выходного сигнала. Заметим, что:

$$\partial F / \partial w_0 = 1, \quad \partial F / \partial w_i |_{\chi = \chi^p} = x_i^p, \quad \partial F / \partial w_i |_{\chi = \chi^p} = w_i, \quad i = 1, \dots, N.$$

Уничтожить  $i$ -й выходной сигнал можно двумя способами:

заменой параметра  $w_i$  на 0;

заменой  $x_i$  на постоянную величину не зависящую от  $P$ .

53. В последнем случае получаем новую функцию

$$F_{1i} = w_0 + x_i^0 w_i + \sum_{j=1, j \neq i}^N x_j w_j$$

Такое преобразование означает, что одновременно с уничтожением  $i$ -й выходной связи  $w_0$  приобретает новое значение:

$$w_0 := w_0 + x_i^0 w_i.$$

При этом можно добиться меньшего изменения, чем просто при  $i$ -го приравнении  $w_i$  к нулю. Поэтому остановимся на замене  $i$ -го выходного сигнала на постоянную величину  $x_i^0$ . Значение этой постоянной определим исходя из минимизации



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

изменения  $F(x^p, w)$ . Минимизация этого изменения, вычисленного в евклидовой норме, дает:

$$x_i^0 = (1/n) \sum_{p=1}^n x_i^p$$



Таким образом, оптимальной является замена  $x_i$  на его среднее значение по исходной выборке. В обозначениях теории вероятностей:

$$x_i^0 = M(x_i), \chi(x_i) = n^{1/2} w_i \sigma(x_i).$$

где  $\sigma(x_i)$  - среднеквадратичное отклонение от  $x_i^0$  на выборке  $\{x^p\}$ .

Значимость замены оценивается как

$$\chi(x_i) = |w_i| \sigma(x_i).$$

При исключении сигналов по одному, они сортируются в соответствии со значениями  $\chi(x_i)$  и отбрасываются (заменяются средним) сначала те, что соответствуют меньшим  $\chi(x_i)$ . Заметим, что поэтому путь "снизу вверх" универсален, но не оптимален. В частности, для сумматоров и других элементов, линейных по параметрам (например, квадратичных сумматоров), существует учитывающий все корреляции путь исключения "сверху вниз" с ортогонализацией. Далее ограничимся оценкой значимости по изменению выходного сигнала.

#### **Показатели значимости для нейрона с дифференцируемым нелинейным элементом**

Эти показатели ищутся почти так же как для сумматора. Пусть

$$F(x, w) = \varphi(w_0 + (x, w))$$

тогда

$$\partial F / \partial x_i |_{x=x^p} = \varphi'(w_0 + (x^p, w)) \cdot w_i.$$

В евклидовой норме (что соответствует методу наименьших квадратов) получаем:

$$\chi(x_i) = |w_i| \left[ \sum_p (\varphi'(w_0 + (x^p, w)))^2 (x_i^p - M(x_i))^2 \right]^{1/2},$$

т.е. произведение модуля параметра  $w_i$  на среднеквадратичное отклонение с весами. Роль веса играет квадрат производной функции в точке  $w_0 + (x^p, w)$ .

#### **Показатели значимости для нейрона с пороговым нелинейным элементом (персептрона)**

Эти показатели требуют для своего вычисления еще одного эвристического хода, т.к. прямо воспользоваться предыдущими формулами невозможно. Пусть функция, вычисляемая нейроном, имеет вид



$$F(x, w) = h(w_0 + (x, w)),$$
$$h(x) = \begin{cases} 1, & x > 0, \\ 0, & x \leq 0 \end{cases}$$

Для каждого вектора данных  $x^p$  необходимо оценить *значимость* изменения аргумента функции  $h$  при замене  $x_i^p$  на  $x_i^0$ . Значение  $h(a)$  меняется только тогда, когда  $a$  меняет знак, поэтому естественно оценивать *значимость* изменения  $\Delta a$  переменной  $a$  в масштабе, определенном текущим значением переменной, т.е. *значимость*  $\Delta a$  оценивается, как  $|\Delta a/a|$ . Из этого эвристического рассуждения получаем:

$$(x_i|x^p) = |w_i| \cdot |x_i^p - x_i^0|/|w_0 + (x^p, w)|.$$

Если положить  $x_i^0 = M(x_i)$  и воспользоваться евклидовой мерой, то вновь получим *произведение* модуля параметров  $w_i$  на среднеквадратичное отклонение с весом. В качестве весов выступают квадраты величин, обратных выходным сигналам сумматоров:

$$\chi(x_i) = |w_i| \cdot \left[ \sum_p (x_i^p - M(x_i))^2 / (w_0 + (x^p, w))^2 \right]^{1/2}.$$

Полученные *выражение* для показателей значимости позволяют упорядочить основные элементы НС "снизу вверх", начиная с исключения самых малозначимых параметров.

**Сокращение описания "сверху вниз" - набор достаточного семейства наиболее значимых параметров**

Метод исключения параметров "сверху вниз" с ортогонализацией применим не ко всяким функциям  $F(x, w)$ , а только к таким, которые имеют вид:

$$F(x, w) = \varphi\left(\sum_i w_i f_i(x)\right).$$

Достоинство метода - автоматический учет корреляции между  $f_i(x)$ . Рассмотрим устройства, вычисляющие функции

$$F(x, w) = \sum_i w_i f_i(x).$$

В них относятся линейные сумматоры, квадратичные сумматоры и

др. Пусть заданы векторы данных:



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

$$x^p = (x_1^p, \dots, x_i^p, \dots, x_N^p), p = 1, \dots, n, i = 1, \dots, N.$$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Поставим задачу свращения описания следующего образа так определить некоторое наименьшее возможное множество индексов и набор чисел, чтобы норма отклонения, где, не превышала отклонения, где, не превышала некоторой  $\epsilon$  перед заданной величины. Все функции рассматриваются на конечном множестве. Для любой функции евклидова норма:

$$\|\varphi\| = \left[ \sum_p \varphi^2(x^p) \right]^{1/2}.$$

С каждой функцией  $f(x)$  связан  $n$ -мерный вектор  $f$  с компонентами  $f^p = f(x^p)$ . Вектор с координатами является линейной комбинацией векторов с координатами. Линейную оболочку семейства векторов обозначим. Построим в пространстве ортонормированный базис с помощью последовательной ортогонализации векторов. Каждый следующий шаг ортогонализации выполним так, чтобы величина проекции на новый вектор базиса была максимальной из возможных. Процесс ортогонализации продолжим, пока, где - проекция на построенную ортогональную систему. По окончании процесса полагаем.

Опишем вычисления более детально.

1. Проводим нормировку: для любого полагаем.

2. Вычисляем (модуль проекции вектора на вектор),. Находим среди этих чисел максимальное (пусть его номер), полагаем, исключаем из множества, получаем, исключаем из множества нулевые векторы, если таковые существуют, проводим нормировку.

3. Пусть определены векторы и не более чем нормированных векторов. Среди векторов ищем такой, для которого принимает максимальное значение, полагаем, исключаем из множества векторов; полагаем, исключаем из этого множества нулевые векторы, если таковые существуют, нормируем:

 $g_{i,max}^l$  $\{g_i^l\}$ 

$$g_i^{l+1} = g_i^l - (g_i^l, e_{l+1})e_{l+1}$$

$$g_i^{l+1} = g_i^{l+1} / \|g_i^{l+1}\|$$

Вычисления проводим, пока  $\|F\|^2 - \sum_{i=1,l} (F, e_i)^2 > \varepsilon^2$ .



После завершения вычислений имеем набор ортонормированных векторов  $e_1, \dots, e_l, l \leq n$ . Они являются линейными комбинациями векторов  $f_{i,1max}, \dots, f_{i,lmax}$ .

. Коэффициенты разложения  $e_j$  по набору  $f_{i,1max}, \dots, f_{i,lmax}$  могут быть вычислены и сохранены в ходе ортогонализации. Полагаем  $J = \{i_{1max}, \dots, i_{lmax}\}$ .

Тогда  $F' = \sum_{j=1,l} (F, e_j) e_j = \sum_{j \in J} \beta_j f_j$ . Это и есть решение задачи.

Числа  $\beta_j$  выражаются через коэффициенты разложения

векторов  $e_i, i = 1, \dots, l$  по  $f_j, j \in J$  и скалярные произведения  $(F, e_j)$ :  
если  $e_i = \sum_{j \in J} q_{ij} f_j$ , то  $\beta_j = \sum_{i=1,l} (F, e_i) q_{ij}$ .

Разложение  $e_i$  по  $f_j, j \in J$  имеет рекурсивную форму:

$$e_1 = f_{i,1max} / \|f_{i,1max}\|,$$

$$e_2 = [f_{i,2max} - (f_{i,2max}, e_1)e_1] / \|f_{i,2max} - (f_{i,2max}, e_1)e_1\|,$$

...

$$e_j = [f_{i,jmax} - \sum_{r=1,j-1} (f_{i,jmax}, e_r)e_r] / \|f_{i,jmax} - \sum_{r=1,j-1} (f_{i,jmax}, e_r)e_r\|,$$

...

Для функций вида  $c$  дифференцируемой функцией процедура аналогична с точностью до замены скалярного произведения: используется скалярное произведение с весами  $(f, g) = \sum_p V_p f^p g^p$ , где  $V_p = ((\sum_i w_i f_i(x^p)))^2$ . В этом скалярном произведении вычисляются все нормы и проводится ортогонализация.

Для функций с пороговой нелинейностью на выходе используем скалярное произведение с весами  $V_p = |\sum_i w_i f_i(x^p)|^2$ .

Описанная процедура сокращения "сверху вниз" с ортогонализацией особенно важна для упрощения элементов сложных сетей, в структуре которых и вектор входных сигналов элемента может быть далек от исходных данных, и его выходной сигнал далек от оцениваемого выхода всей сложной системы.

Процедуры анализа значимости и сокращения описания выделяют наиболее важные параметры и связи в НС. По аналогии с обработкой изображения их называют процедурами контрастирования или редукции.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Роль контрастирования (редукции) не сводится только к сокращению описания: более общая задача - привести параметры системы к выделенному набору значений, в частности, уменьшить *разрядность*, что важно для удешевления специализированных устройств, экономии памяти и т.д.



### Рекурсивное контрастирование и бинаризация

Рекурсивное контрастирование состоит в модификации параметров системы - одного за другим. Для этого параметры должны быть как-то линейно упорядочены  $w_1, \dots, w_N$ . При модификации  $w_i$  используются модифицированные значения  $w_1, \dots, w_{i-1}$  и немодифицированные  $w_{i+1}, \dots, w_N$ .

Пусть для сумматора задана обучающая выборка входных векторов  $x_1, \dots, x_n$  и соответствующих выходных сигналов  $f_1, \dots, f_n$ , а также известны значения параметров, которые реализует сумматор:  $f_i = w_0 + (x^i, w)$ . Требуется произвести бинаризацию сумматора, т.е. найти числа  $a, b$  и вектор  $\beta$  с координатами 0 или 1, чтобы значения функции  $\varphi(x) = a + b(x, \beta)$  на выборке  $\{x^i\}$  как можно меньше отличались от  $f_i$ . Критерием такого отличия будем считать  $H = \sum_{i=1, n} (f_i - \varphi(x^i))^2$ . Построим координаты вектора  $\beta$  по порядку  $\beta_1, \beta_2, \dots$ .

Пусть построены  $\beta_1, \dots, \beta_{i-1}$ . Обозначим  $\beta^{0i} = (\beta_1, \dots, \beta_{i-1}, 0, \dots, 0)$  (последние  $N - i + 1$  координат нули),  $\beta^{1i} = (\beta_1, \dots, \beta_{i-1}, 1, 0, \dots, 0)$  (последние  $N - i$  координат нули),  $\alpha^i = (0, \dots, 0, \alpha_{i+1}, \dots)$  (первые  $i$  координат - нули).

Введем функции:

$$\varphi_0^i(x) = a_{0i} + b_{0i}(x, \beta^{0i}) + (x, \alpha^i),$$

$$\varphi_1^i(x) = a_{1i} + b_{1i}(x, \beta^{1i}) + (x, \alpha^i),$$

$$H_{0i} = \sum_{j=1, n} (f_j - \varphi_0^i(x^j))^2,$$

$$H_{1i} = \sum_{j=1, n} (f_j - \varphi_1^i(x^j))^2.$$

Определим параметры  $a_{0i}, b_{0i}, a_{1i}, b_{1i}$  из условий  $H_{0i} \rightarrow \min, H_{1i} \rightarrow \min$ , минимизируя функции  $H_{0i}$  и  $H_{1i}$ . Пусть  $h_{0i} = \min H_{0i}$  и  $h_{1i} = \min H_{1i}$ . Если  $h_{1i} \geq h_{0i}$ , то полагаем  $\beta_i = 0$ , в противном случае  $\beta_i = 1$ .

После того как построены все  $\beta_i, i = 1, \dots, N$  ( $N$  - размерность вектора данных), автоматически определяются  $a$  и  $b$ : если  $N = 0$ , то полагаем  $a = a_{0N}, b = b_{0N}$ , иначе  $a = a_{1N}, b = b_{1N}$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Если бинаризация проведена, а необходимая *точность* не достигнута, то можно построить второй бинаризованный *сумматор*, корректирующий ошибку первого --- так, чтобы в сумме они хорошо аппроксимировали работу исходного сумматора на элементах обучающей выборки. В



описанной процедуре делаем замену  $f_i := f_i - \varphi(x^i)$  и для этих исходных данных вновь строим бинаризованный сумматор по алгоритму рекурсивной бинаризации. Повторяем такое построение, пока не будет достигнута удовлетворительная точность. В результате получим набор бинаризованных сумматоров, которые в совокупности (т.е. в результате суммирования выходных сигналов) достаточно точно аппроксимируют исходный. При появлении весов, определяющих значимость отдельных примеров из обучающей выборки, рекурсивная бинаризация проводится точно так же, только в функциях  $H$  появляются веса.

Если требуется тем же путем упростить любой другой элемент, линейный по параметрам,  $F(x, w) = \sum_i w_i f_i(x)$ , то вместо обучающей выборки  $\{x^j\}$  берем семейство векторов  $\{y^j\}$  с координатами  $y_i^j = f_i(x^j)$ . После такого  $y_i^j = f_i(x^j)$  преобразования рассматриваемый элемент превращается в обычный сумматор, для которого последовательность действий уже описана.

#### Контрольные вопросы ЛР (ПК-4):

- 000 Метод максимума правдоподобия.
- 000 Нейрофизиологическая аналогия.
- 000 Реализация булевых функций нейронными сетями.
- 000 Выделение выпуклых областей.
- 000 Особенности задачи оптимизации, возникающей при обучении нейронных сетей.
- 000 Учет ограничений при обучении.
- 000 Выбор направления минимизации.
- 000 Паран-методы.
- 000 Одношаговый квазиньютоновский метод и сопряженные градиенты.
- 000 Комплексная схема нечеткого планирования.
- 000 Процессный подход к организации деятельности организации.
- 000 Связь концепции процессного подхода с концепцией матричной организации.
- 000 Основные элементы процессного подхода.
- 000 Границы процесса, ключевые роли, дерево целей, дерево функций, дерево показателей.
- 000 Выделение и классификация процессов.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

29

## Лабораторная работа №4.

**Градиентные алгоритмы обучения сети. Универсальный путь обучения. Особенности задачи оптимизации, возникающей при обучении нейронных сетей. Учет ограничений при обучении. Выбор направления минимизации. партан-методы. Одношаговый квазиньютоновский метод и сопряженные градиенты. Сдача теста № 4.**

## **Электронная реализация нейронных сетей**

В качестве единицы производительности нейросетей принято "число соединений в секунду" -

*CPS* (*connections per second*). Под соединением здесь понимается *умножение* входного сигнала на весовой коэффициент и *сложение* с накопленной суммой.

*Анализ нейросетевых алгоритмов* позволяет сделать следующие выводы:

22. При решении плохо формализованных задач моделирования, прогнозирования и распознавания, которые обычно сводятся к конструированию областей многомерного пространства, достаточно малоразрядных представлений входов и весов и операций с фиксированной точкой. Это обусловлено тем, что входные сигналы нормируются и количество их значений невелико.
23. При решении хорошо формализованных задач (например, задач комбинаторной оптимизации) существенна точность вычислений, что требует полноразрядных представлений чисел и операций с плавающей точкой.

Электронные нейронные сети обычно используются в качестве *акселераторов* для персональных ЭВМ при решении соответствующих классов задач (обработки сигналов и изображений, распознавания образов и т.п.).

## **Нейрочипы**

Нейрочипы подразделяются на цифровые, аналоговые и гибридные. Они могут включать в себя схемы настройки весов при обучении или предусматривать внешнюю загрузку весов. Наибольшую проблему при создании нейрочипов представляют схемы умножения, так как именно они лимитируют скорость вычислений.

Аналоговые реализации используют простые физические эффекты для выполнения нейросетевых преобразований. Обеспечение заданной точности требует тщательного проектирования и изготовления.

Гибридные нейрочипы используют комбинацию аналогового и цифрового подходов. Например, входы могут быть аналоговыми, веса могут загружаться как цифровые и выходы могут быть цифровыми. Существуют нейрочипы, в которых используется *представление данных* частотой или шириной импульсов.

## **Нейропроцессор NM6403**

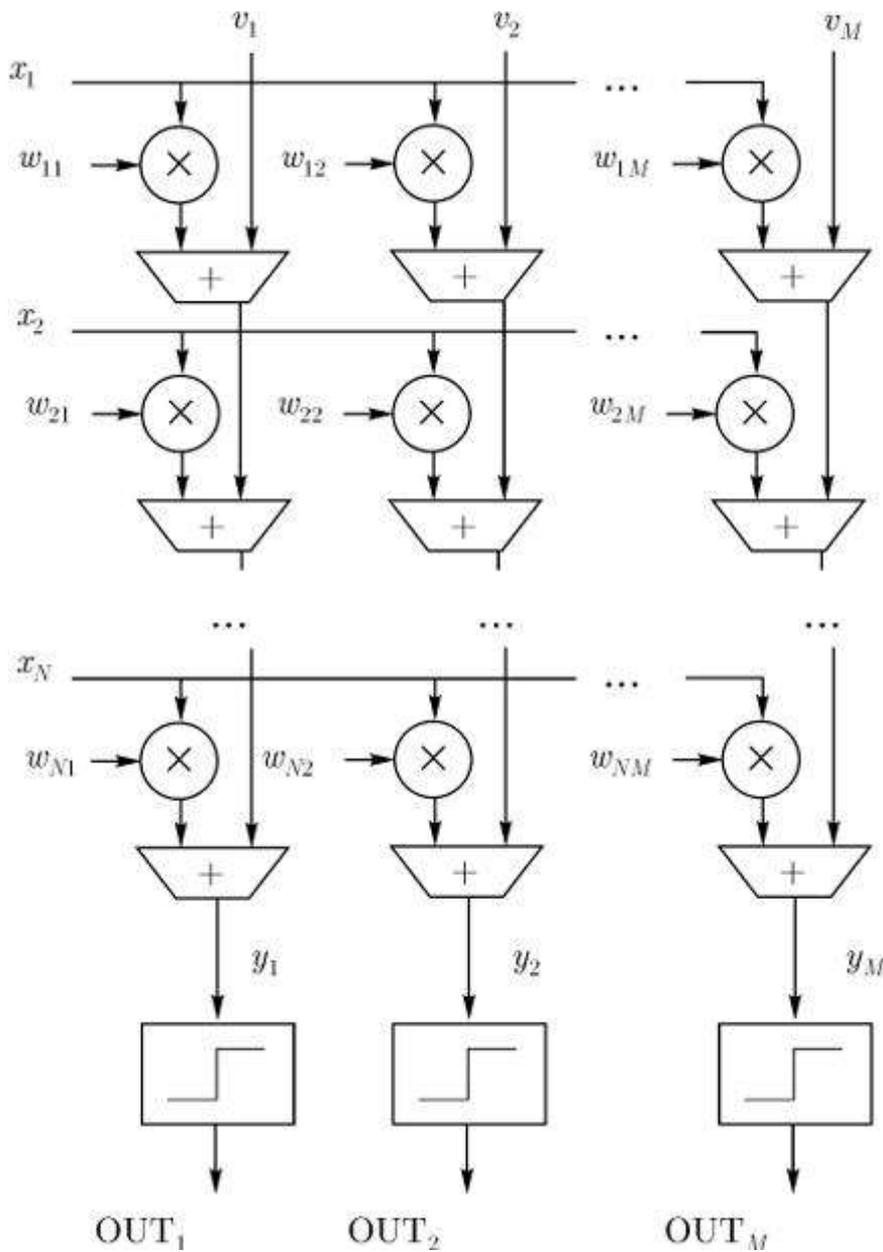
Отечественный нейропроцессор NM6403, разработанный в НПО "Модуль" ([www.module.ru](http://www.module.ru)), имеет скалярный *процессор* (скалярное RISC-ядро) и *векторный процессор* для обработки двоичных векторов произвольной разрядности в пределах 1-64 битов. Скалярный *процессор* выполняет всю подготовку данных для работы векторного процессора.

Модель слоя нейронов, эмулируемого процессором NM6403, показана на рис. 1. В общем случае слой имеет  $N$  входов  
43. состоит из  $M$  нейронов.  $M$ -й нейрон выполняет операцию умножения - накопления над  $N$  данными  $x_1, \dots, x_N$ , поданными на соответствующие входы нейрона.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

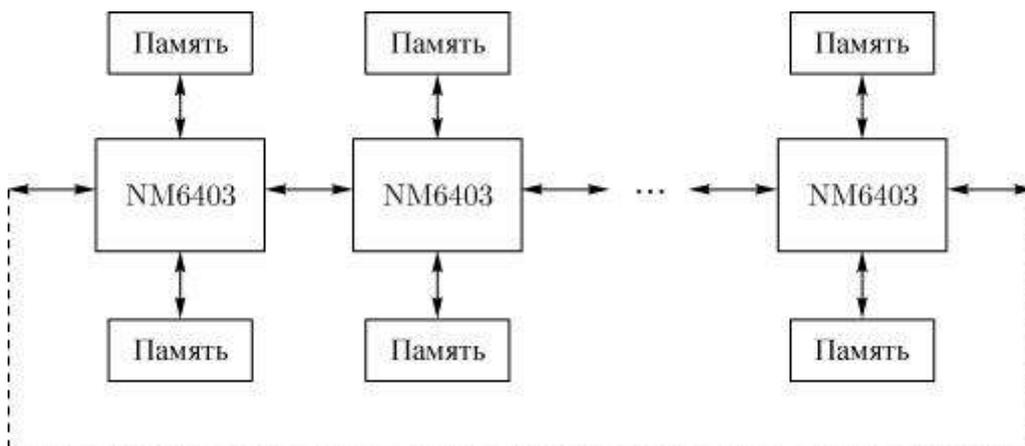
Нейропроцессор NM6403 имеет два встроенных линка, совместимых с линками сигнального микропроцессора TMS320C40. Кроме того, интерфейсы локальной и глобальной памяти позволяют без дополнительного оборудования подсоединять два нейропроцессора к общему блоку памяти.



[увеличить изображение](#)

Рис. 1. Модель слоя нейронов

Мультипроцессорная система (рис. 2) из  $K$  нейропроцессоров NM6403 эмулирует нейронную сеть в  $K$  раз быстрее, чем один нейропроцессор.





**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Рис. 2.** Линейная (кольцевая) система из нейропроцессоров NM6403



### **Оптическая реализация нейронных сетей**

*Мощность* нейронной сети определяется большим количеством связей: отдельные элементы имеют относительно малые вычислительные мощности. Обеспечение требуемой связности в электронных цепях остается серьезной проблемой, особенно при реализации нейронных сетей с полным графом соединений. Электронные интегральные цепи являются существенно планарными с рельефностью, обусловленной множеством слоев.

Проблему связей можно решить при использовании оптических систем для реализации НС. Взаимное соединение нейронов с помощью световых лучей не требует изоляции между сигнальными путями: световые потоки могут пересекаться, не влияя друг на друга, и сигнальные пути могут располагаться в трех измерениях. Плотность путей передачи ограничена только размерами источников и детекторов. Все сигнальные пути могут работать одновременно, тем самым обеспечивая огромную *скорость передачи* данных.

41. оптических НС величины оптических весов могут запоминаться в голограммах с высокой степенью плотности (до  $10^{12}$  бит на куб. см.). Веса могут модифицироваться в процессе работы сети.

к сожалению, возникает множество практических проблем при попытках оптической реализации нейронных сетей. Оптические устройства имеют собственные физические характеристики, часто не соответствующие требованиям искусственных нейронных сетей. Хотя они в действительности пригодны для обработки изображений, все же изображения от оптических нейронных сетей, полученные до настоящего времени, были разочаровывающе плохими. Однако достаточно взглянуть на первые пробы телевизионных передач, чтобы понять, какой огромный прогресс возможен в повышении качества изображения. Несмотря на эти трудности, а также на такие проблемы, как *стоимость*, размеры и критичность к ориентации, потенциальные возможности оптических систем побуждают попытки проведения интенсивных и широких исследований. В этой области происходят стремительные изменения, и в ближайшее время ожидаются важные улучшения.

Конфигурации оптических НС в основном подразделяются на две категории: векторно-матричные умножители и голографические корреляторы.

### **Векторно-матричные умножители**

46. качестве *матрицы весов* (рис. 3) используется фотопленка, у которой прозрачность каждого квадрата пропорциональна весу. *Выход* каждого фотодетектора является сверткой между входным вектором и соответствующим столбцом *матрицы весов*. Умножение выполняется параллельно. При использовании соответствующих высокоскоростных светодиодов и фотодетекторов *умножение* вектора на матрицу может быть выполнено менее, чем за наносекунду. Более того, скорость умножения практически не зависит от размерности массива. Это позволяет наращивать сети без существенного увеличения времени вычислений. Возможность менять веса основана на использовании жидкокристаллического клапана вместо фотографического негатива.

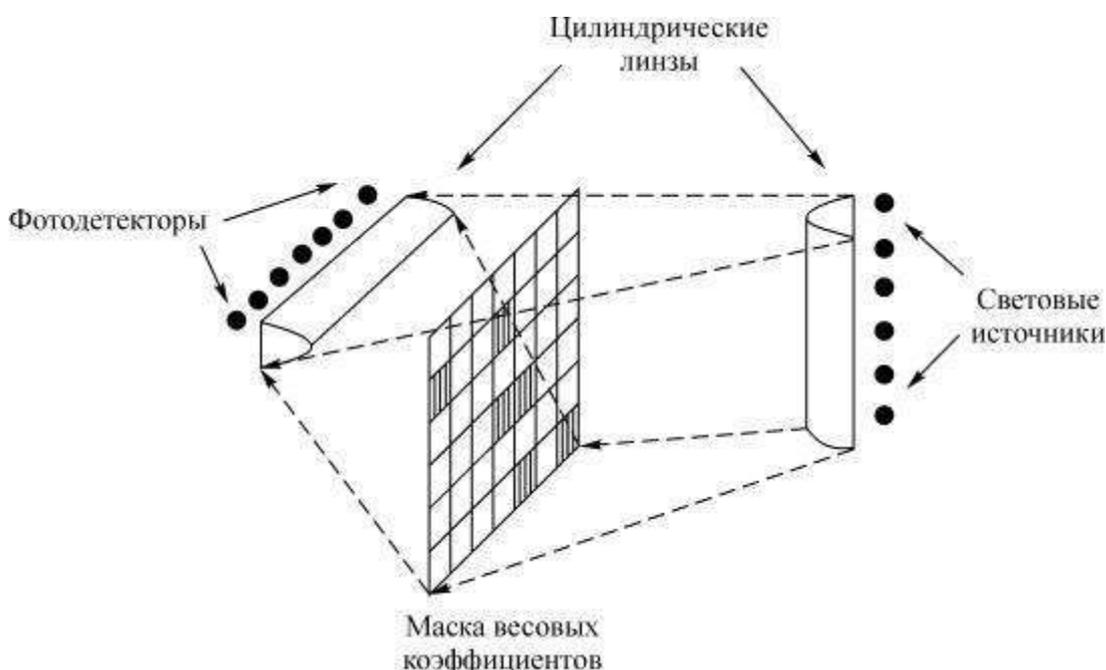




Рис. 3. Электронно-оптический векторно-матричный умножитель

### ***Голографические корреляторы***

20. голографических корреляторах образцы изображений запоминаются в виде голограммы (плоской или объемной) и восстанавливаются при когерентном освещении в петле обратной связи.

Входное изображение (возможно, зашумленное или неполное) коррелируется оптически одновременно со всеми запомненными изображениями. Корреляции обрабатываются пороговой функцией и подаются на вход системы, где наиболее сильные корреляции усиливают (и, возможно, корректируют или завершают) входное изображение. Этот процесс повторяется многократно, и усиленное изображение при каждом проходе изменяется, пока система не стабилизируется на требуемом изображении.

Оптические нейронные сети предлагают огромные выгоды с точки зрения скорости и плотности внутренних связей. Они могут быть использованы (в той или иной форме) для реализации сетей фактически с любой архитектурой.

21. настоящее время ограничения электронно-оптических устройств создают множество серьезных проблем, которые должны быть решены прежде, чем оптические нейронные сети получат широкое применение. Однако, учитывая, что большое количество превосходных исследователей работает над этой проблемой, а также большую поддержку со стороны военных, можно надеяться на быстрый прогресс в этой области.

### **Контрольные вопросы ЛР (ПК-4):**

27. Особенности задачи оптимизации, возникающей при обучении нейронных сетей.
28. Учет ограничений при обучении.
29. Выбор направления минимизации.
30. Парган-методы.
31. Одношаговый квазиньютоновский метод и сопряженные градиенты.
32. Максимизация консенсуса.
33. Синхронное и асинхронное функционирование машины Больцмана.
34. Решение задачи коммивояжера машиной Больцмана.
35. Свойства, определяемые пользователем (UDP).
36. Диаграммы потоков данных (Data Flow Diagramming): работы.
37. Диаграммы потоков данных (Data Flow Diagramming): внешние сущности (ссылки).
38. Диаграммы потоков данных (Data Flow Diagramming): потоки работ.
39. Диаграммы потоков данных (Data Flow Diagramming): хранилища данных.
40. Внемашинное информационное обеспечение.
41. Основные понятия классификации информации.



**Лабораторная работа №5. Рекуррентные сети как ассоциативные запоминающие устройства. Автоассоциативная сеть Хопфилда. Обучение сети Хопфилда по правилу Хебба. Сеть Хемминга. Двухнаправленная ассоциативная память.**

**Аннотация:** Рассматриваются: нейронная сеть Хопфилда как автоассоциативная память, обучаемая с использованием метода Хебба и проекционного метода; гетероассоциативная память на основе сети Хемминга и двухнаправленная ассоциативная память.

**Введение**

Отдельную группу нейронных сетей составляют *сети с обратной связью* между различными слоями нейронов. Это так называемые рекуррентные сети. Их общая черта состоит в передаче сигналов с выходного либо скрытого слоя на *входной* слой.

Благодаря обратной связи при подаче сигнала на входы сети, в ней возникает переходный процесс, который завершается формированием нового устойчивого состояния, отличающегося в общем случае от предыдущего. Если функцию активации нейрона обозначить  $f(u)$ , где  $u$  - взвешенная сумма его возбуждений, то состояние нейрона можно определить выходным сигналом  $y = f(u) = f(w_1x_1 + \dots + w_Nx_N)$ . Изменение состояния  $i$ -го нейрона можно описать системой дифференциальных уравнений

$$\tau_1(\partial u_i / \partial t) = w_{i1}f(u_1) + \dots + w_{iN}f(u_N) - u_i - b_i$$

для  $i = 1, 2, \dots, N$ , где  $b_i$  - пороговое значение.

Рекуррентной сети можно поставить в соответствие энергетическую функцию Ляпунова

$$E = -(1/2) \sum_j \sum_{i \neq j} w_{ij}y_iy_j + \sum_{i=1,N} (1/R_i) \int_0^x f_i^{-1}(y_i)dy_i + \sum_{i=1,N} b_iy_i.$$

Изменение состояния какого-либо нейрона инициализирует изменение энергетического состояния сети в направлении минимума ее энергии вплоть до его достижения. В пространстве состояний локальные энергетические минимумы  $E$  представлены точками стабильности, называемыми аттракторами из-за тяготения к ним ближайшего окружения. Благодаря наличию аттракторов, рекуррентные сети могут быть использованы как устройства ассоциативной памяти.

*Ассоциативная память* играет роль системы, определяющей взаимную зависимость векторов. В случае, когда на взаимозависимость исследуются компоненты одного и того же вектора, говорят об автоассоциативной памяти. Если же взаимозависимыми оказываются два различных вектора, можно говорить о памяти гетероассоциативного типа. К первому классу относится *сеть Хопфилда*, а ко второму - *сеть Хемминга* и *сеть* типа ВАР (*Bidirectional Associative Memory* - *двухнаправленная ассоциативная память*).

Задача ассоциативной памяти сводится к запоминанию обучающих векторов, чтобы при представлении нового вектора система могла сгенерировать ответ - какой из запомненных ранее векторов наиболее близок к вновь поступившему образу. Часто в качестве меры близости отдельных множеств применяется *расстояние Хемминга*.

При использовании двоичных значений (0,1) *расстояние Хемминга* между двумя



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

векторами  $y = (y_1, y_2, \dots, y_n)$  и  $d = (d_1, d_2, \dots, d_n)$  определяется в виде



$$d_H(y, d) = \sum_{i=1, n} (d_i(1 - y_i) + (1 - d_i)y_i)$$

При биполярных значениях элементов обоих векторов *расстояние* Хемминга рассчитывается по формуле

$$d_H(y, d) = (1/2)(n - \sum_{i=1, n} d_i y_i)$$

*Мера* Хемминга равна числу несовпадающих *компонент* двух векторов. Она равна нулю, когда  $y = d$ .

### **Автоассоциативная сеть Хопфилда**

Структура *сети Хопфилда* представляется в виде системы с непосредственной обратной связью выхода со входом (рис. 1). Выходные сигналы нейронов являются одновременно входными сигналами сети:  $x_i(k) = y_i(k - 1)$ . В классической *сети Хопфилда* отсутствует автосвязь (*связь* выхода нейрона с его собственным входом), что соответствует  $w_{ii} = 0$ , а *матрица* весов является симметричной:  $W = W^T$ . Отсутствие автосвязи 23. *симметричность матрицы весов* являются достаточными (но не необходимыми!) условиями сходимости итерационных (переходных) процессов в *сети Хопфилда*.

Далее в данной лекции предполагаем, что каждый *нейрон* имеет биполярную ступенчатую функцию активации со значениями  $\pm 1$ . Это означает, что выходной сигнал  $i$ -го нейрона определяется функцией

$$y_i = \text{sgn}\left(\sum_{j=0, N} w_{ij}x_j + b_i\right)$$

где  $N$  обозначает количество нейронов,  $N = n$ .

Далее допустим, что порог срабатывания является компонентой вектора  $x$ . Тогда основную зависимость, определяющую *сеть Хопфилда*, можно представить в виде

$$y_i(k) = \text{sgn}\left(\sum_{j=0, N} w_{ij}y_j(k - 1)\right) \quad 1)$$

с начальным условием  $y_j(0) = x_j$ .

29. процессе функционирования *сети Хопфилда* можно выделить два режима: обучения и классификации. В режиме обучения на основе известных векторов подбираются *весовые коэффициенты* сети. В режиме классификации при фиксированных значениях весов и вводе конкретного начального состояния нейронов возникает переходный процесс вида (1),



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

завершающийся в одном из локальных минимумов, для которого  $y(k) = y(k - 1)$ .

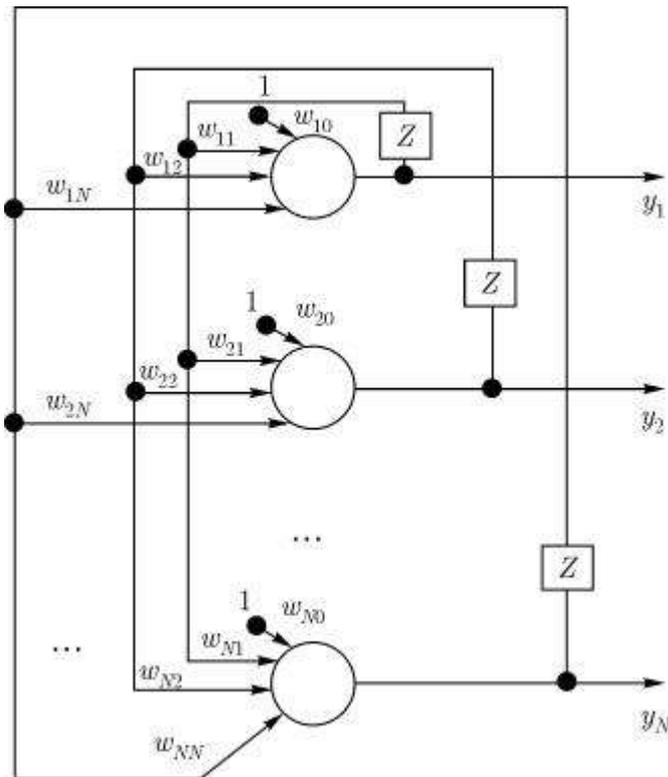


Рис. 1. Структура сети Хопфилда

### Обучение сети Хопфилда по правилу Хебба

Для одного обучающего вектора  $\mathbf{x}$  значения весов могут быть вычислены по правилу Хебба

$$w_{ij} = (1/N)x_i x_j,$$

поскольку тогда

$$(1/N) \left( \sum_{j=1}^N x_i x_j x_j \right) = x_i$$

(вследствие биполярных значений элементов вектора  $\mathbf{x}$  всегда  $x_j^2 = (\pm 1)^2 = 1$ ).

При вводе большего количества обучающих векторов  $\mathbf{x}^{(k)}, k = 1, 2, \dots, p$  веса  $w_{ij}$  подбираются согласно обобщенному правилу Хебба

$$w_{ij} = (1/N) \sum_{k=0}^p x_i^{(k)} x_j^{(k)}.$$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Важным параметром ассоциативной памяти является ее емкость. Под емкостью понимается максимальное число запомненных образов, которые классифицируются с допустимой



погрешностью  $\varepsilon_{max}$ . Показано, что при использовании для обучения правила Хебба и при  $\varepsilon_{max} = 0.01$  (1% компонентов образа отличается от нормального состояния) максимальная емкость памяти составит всего лишь около 13,8% от количества нейронов, образующих ассоциативную память. Столь малая емкость обусловлена тем, что сеть Хебба хорошо запоминает только взаимно ортогональные векторы или близкие к ним.

### Обучение сети Хопфилда методом проекций

Лучшие результаты, чем при использовании правила Хебба, можно получить, если для обучения использовать псевдоинверсию. В основе этого подхода лежит предположение, что при правильно подобранных весах каждый поданный на вход сети вектор вызывает генерацию самого себя на выходе сети. В матричной форме это можно представить в виде

$$WX = X,$$

где  $W$  - матрица весов сети размерностью  $N \times N$ , а  $X$  - прямоугольная матрица размерностью  $N \times p$ , составленная из  $p$  обучающих векторов  $x^{(i)}, i = 1, 2, \dots, p$ . Решение такой линейной системы уравнений имеет вид

$$W = XX^+,$$

где знак  $+$  обозначает псевдоинверсию.

Если обучающие векторы линейно независимы, последнее выражение можно упростить и представить в виде

$$W = X(X^T X)^{-1} X^T. \quad 2)$$

Здесь псевдоинверсия заменена обычной инверсией квадратной матрицы  $X^T X$  размерностью  $p \times p$ .

Выражение (2) можно записать в итерационной форме, не требующей расчета обратной матрицы. В этом случае (2) принимает вид итерационной зависимости от последовательности обучающих векторов  $x^{(i)}, i = 1, 2, \dots, p$ :

$$y^{(i)} = (W^{(i-1)} - E)x^{(i)}, \\ W^{(i)} = W^{(i-1)} - (y^{(i)} y^{(i)T}) / (y^{(i)T} y^{(i)})$$

при начальных условиях  $W^{(0)} = 0$ . В результате предъявления  $p$  векторов матрица весов сети принимает значение  $W = W^{(p)}$ . Описанный здесь метод называется методом проекций. Применение его увеличивает максимальную емкость



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

сети Хопфилда до  $N - 1$ . Увеличение емкости обусловлено тем, что в методе проекций требование ортогональности векторов заменено гораздо менее жестким требованием их линейной независимости.



Модифицированный вариант метода проекций - метод  $\Delta$ -проекций — градиентная форма алгоритма минимизации. В соответствии с этим методом веса подбираются с помощью процедуры, многократно повторяемой на всем множестве обучающих векторов:

$$W \leftarrow W + (h/N)(x^{(i)} - Wx^{(i)})x^{(i)T}, h \in (0.7, 0.9).$$

Обучающие векторы предъявляются многократно вплоть до стабилизации значений весов.

### ***Сеть Хемминга***

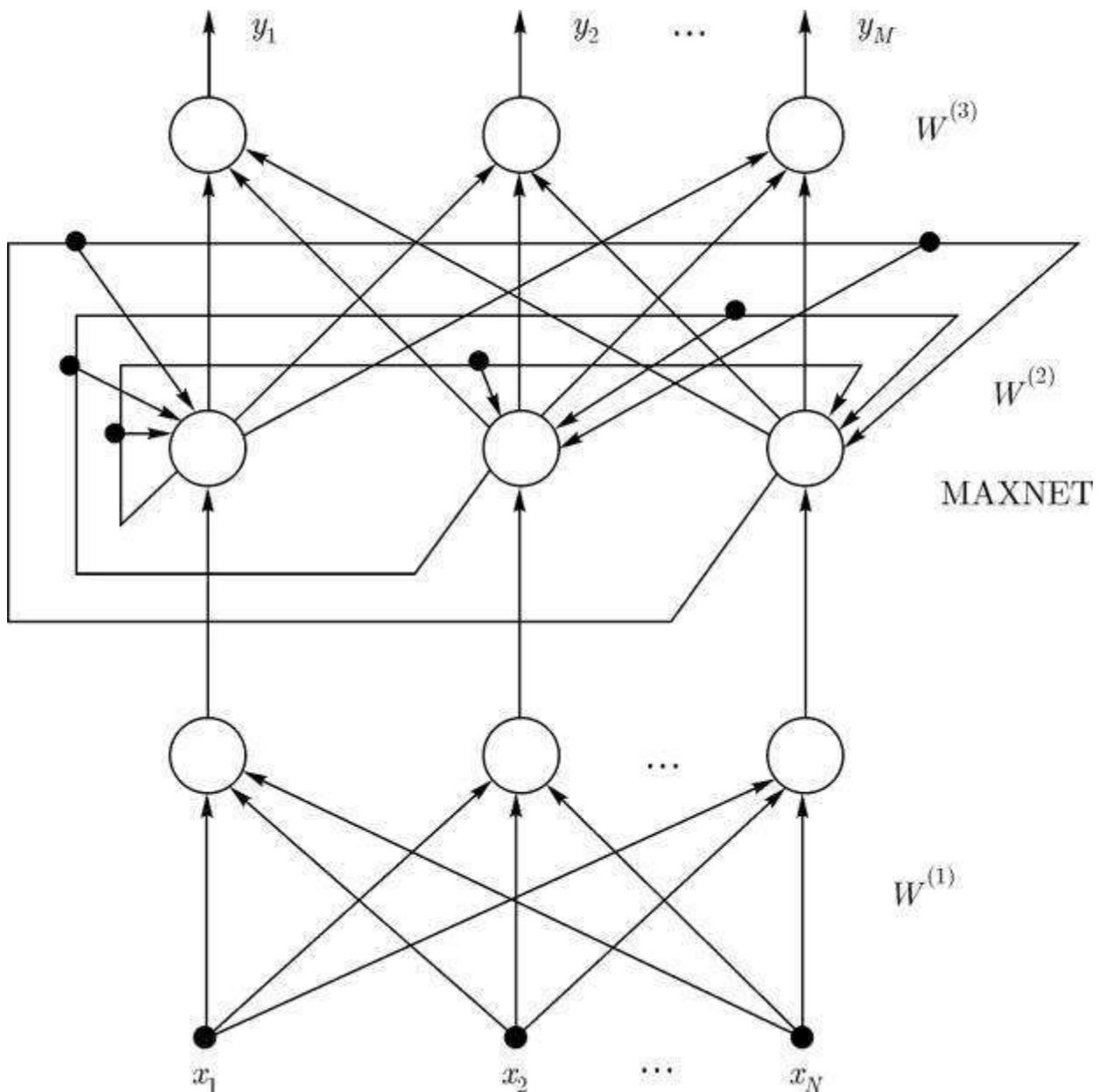
*Сеть* Хемминга включает в себя три слоя (рис.2).

Первый слой имеет однонаправленное распространение сигналов от входа к выходу и фиксированные значения весов.

Второй слой состоит из нейронов, связанных обратными связями по принципу "каждый с каждым", при этом в каждом нейроне слоя существует автосвязь (*связь* входа нейрона со своим собственным выходом). Разные нейроны в слое связаны отрицательной (тормозящей) обратной связью с весом  $-\varepsilon$ , при этом величина  $\varepsilon$  обычно обратно пропорциональна количеству образов.

22. собственным входом *нейрон* связан положительной (возбуждающей) обратной связью с весом, равным +1. Пороговые веса нейронов приняты равными нулю. Нейроны этого слоя функционируют в режиме **WTA**, при котором в каждой фиксированной ситуации активизируется только один *нейрон*, а остальные пребывают в состоянии покоя.

Выходной однонаправленный слой формирует выходной *вектор*, соответствующий входному вектору.



[увеличить](#)

[изображение](#)

Рис. 2. Структура сети Хемминга

Сеть Хемминга считается гетероассоциативным запоминающим устройством с парой связанных между собой векторов  $(x, y)$ , где  $x$  и  $y$  - входной и выходной биполярные векторы сети.

Веса первого слоя соответствуют векторам  $x_i, i = 1, \dots, p$ , т.е.

$$w_{ij}^{(1)} = x_{ij}.$$

Аналогично, веса выходного слоя соответствуют векторам образов  $y_i$ , связанных с  $x_i$ :

$$w_{ij}^{(3)} = y_{ij}.$$



Во втором слое (MAXNET), функционирующем в режиме *WTA* (Winner Takes *ALL* - "Победитель забирает все"), каждый *нейрон* должен усиливать собственный сигнал и ослаблять сигналы остальных нейронов. Для этого принимается

$$w_{ij}^{(2)} = 1,$$

а также

$$-1/(p - 1) < w_{ij}^{(2)} < 0, \quad i \neq j.$$

Для обеспечения сходимости итерационного процесса во втором слое веса

$$w_{ij}^{(2)} = -1/(p - 1) + \xi,$$

где  $\xi$  - достаточно малая *случайная величина*,  $|\xi| \ll 1/(p - 1)$ .

Нейроны первого слоя рассчитывают расстояния Хемминга  $d_H(x, y)$  между поданным на вход сети вектором  $x$  и векторами весов  $w_i = x_i, i = 1, 2, \dots, p$  нейронов этого слоя. Значения выходных сигналов нейронов первого слоя определяются по формуле

$$y_i^{(1)} = 1 - d_H(x, y)/N,$$

где  $N$  - число *компонент* вектора  $x$ .

Сигналы  $y_i^{(1)}$  становятся начальными состояниями нейронов второго слоя. Этот слой определяет "победителя", т.е. *нейрон*, выходной сигнал которого близок к 1. Такой *нейрон* указывает на *вектор* образа с минимальным расстоянием Хемминга до входного вектора  $x$ . *Функция* активации для нейронов второго слоя задается выражением

$$f(y) = \begin{cases} y, & \text{если } y > 0, \\ 0, & \text{если } y < 0. \end{cases}$$

Итерационный процесс во втором слое завершается, когда активным остается только один *нейрон* (победитель), тогда как остальные нейроны пребывают в нулевом состоянии.

Победитель через веса  $w_{ij}^{(3)}$  линейных нейронов выходного слоя представляет *вектор*  $Y_i$ , который соответствует вектору  $x_i$ , признанному вторым слоем ближайшим к входному вектору  $x$ .

Достоинством сети Хемминга считается небольшое количество взвешенных связей между нейронами. Многочисленные эксперименты доказали, что *сеть* Хемминга дает лучшие результаты, чем *сеть Хопфилда*. Единственная проблема, связанная с сетью Хемминга, проявляется в случае,



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

когда зашумленные образы находятся на одинаковом (в смысле Хемминга) расстоянии от двух или более эталонов. В этом случае выбор сетью Хемминга одного из эталонов становится случайным.



### *Двунаправленная ассоциативная память*

Обобщением *сети Хопфилда* на случай двухслойной рекуррентной структуры, позволяющей кодировать множества двух взаимосвязанных векторов, считается двунаправленное ассоциативное запоминающее устройство, называемое ВАРМ

(*Bidirectional Associative Memory*) (рис. 3). Сигналы распространяются в двух направлениях. Если в первом цикле сигналы вначале проходят в одну сторону для задания состояний нейронов-получателей, то в следующем цикле эти нейроны сами становятся источниками, посылающими сигналы в обратную сторону. Процесс повторяется до достижения состояния равновесия.

Функция активации нейронов имеет пороговый характер. Для обеспечения лучших характеристик сети на этапе обучения используются только биполярные сигналы. Матрица весов  $W$ , связывающая обе части сети, является действительной и в общем случае несимметричной. При прямом распространении сигналов веса описываются матрицей  $W$ , а при обратном — матрицей  $W^T$ .

Пусть входные обучающие данные представляют собой множество пар  $\{(x_i, y_i), i = 1, 2, \dots, m\}$  биполярных векторов. На основе этого множества формируется матрица

$$W = \sum_{i=1}^n x_i^T y_i.$$

24. результате процесса двунаправленной обработки сигналов формируются два стабильных вектора  $x_f$  и  $y_f$ , удовлетворяющих уравнениям

$$\begin{aligned} y_f &= f(x_f W), \\ x_f &= f(y_f W^T) = f(W y_f^T) \end{aligned}$$

Каждой промежуточной точке  $(x_k, y_k)$  можно сопоставить энергетическую функцию

$$E_k = -x_k W y_k^T,$$

которая убывает при каждом изменении состояния вплоть до достижения локального минимума

$$E_{min} = -x_f W y_f^T, \quad f \in 1, 2, \dots, m.$$

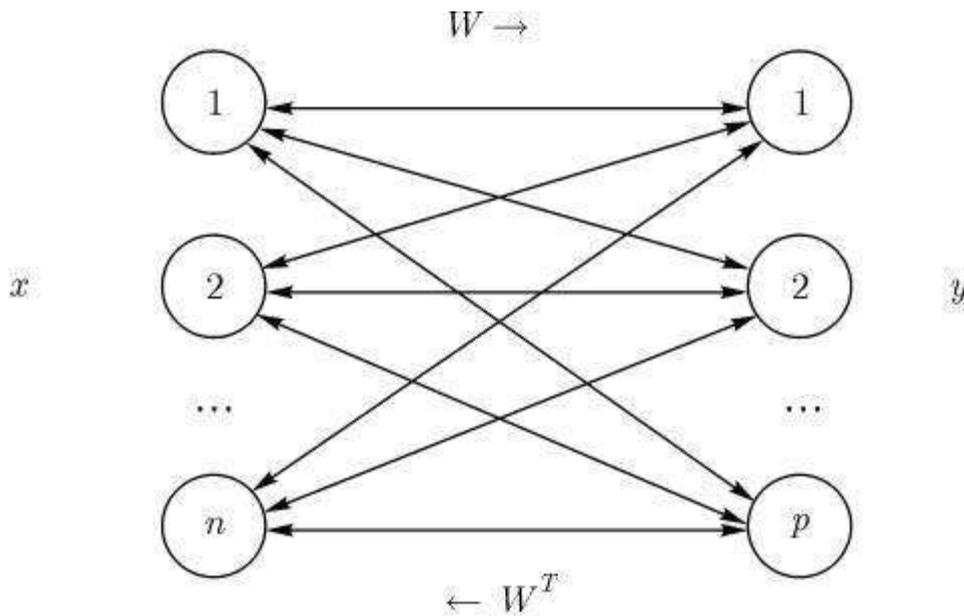


Рис. 3. Структура сети ВАРМ

19. режиме распознавания при начальных значениях векторов, совпадающих с использованными при обучении, сеть распознает их безошибочно. При искажении векторов  $x$  и  $y$  сеть ВАРМ не всегда способна откорректировать эти векторы и распознает их с определенными погрешностями. Если размерности векторов  $x$  и  $y$  обозначить соответственно  $n$  и  $p$ , то удовлетворительное качество распознавания можно получить при выполнении зависимости

$$m < \sqrt{\min(n, p)},$$

где  $m$  - число запоминаемых в сети ВАРМ пар векторов.

#### Контрольные вопросы ЛР5(ПК-4):

17. Автоассоциативная сеть Хопфилда.
18. Обучение сети Хопфилда по правилу Хебба.
19. Сеть Хемминга. Двухнаправленная ассоциативная память.
20. Инструментальный Комплекс Для Создания Статических Экспертных Систем.
21. Средства представления знаний и стратегии управления.
22. Инструментальный комплекс для создания экспертных систем реального времени.
23. Понятия и основные требования к системе кодирования информации.
24. Внутримашинное информационное обеспечение.
25. Проектирование экранных форм электронных документов.
26. Информационная база и способы ее организации.
27. Моделирование информационного обеспечения.
28. Моделирование данных. Метод IDEFI.
29. Уровни отображения модели. Создание физической модели.
30. Уровни физической модели; таблицы; правила валидации и значение по умолчанию.
31. Уровни физической модели; индексы; триггеры и хранимые процедуры.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

32. Уровни физической модели; проектирование хранилищ данных; вычисление размера БД.
33. Уровни физической модели; таблицы; прямое и обратное проектирование



**Лабораторная работа №6. Решение задач комбинаторной оптимизации рекуррентными сетями. Решение задачи коммивояжера сетью Хопфилда. Машина Больцмана. Функция консенсуса. Максимизация консенсуса. Синхронное и асинхронное функционирование машины Больцмана. Решение задачи коммивояжера машиной Больцмана.**

**Аннотация:** Рассматривается решение задачи коммивояжера сетью Хопфилда и машиной Больцмана. Оцениваются параметры функции энергии нейронных сетей, обеспечивающие решение задачи коммивояжера.

**Решение задачи коммивояжера сетью Хопфилда**

Рассмотрим задачу коммивояжера для  $n$  городов. Известны расстояния  $d_{XY}$  между каждой парой городов  $X, Y$ ; коммивояжер, выходя из одного города, должен посетить  $n - 1$  других городов, заходя по одному разу в каждый, и вернуться в исходный. Требуется определить порядок обхода городов, при котором общее пройденное расстояние минимально.

Пусть сеть Хопфилда состоит из  $N = n^2$  нейронов, а состояние нейронов описывается двойными индексами  $v_{Xi}$ , где индекс  $X$  связан с именем города,  $i$  - с позицией города в маршруте коммивояжера. Запишем функцию вычислительной энергии для сети, предназначенной решать задачу коммивояжера. В ней состояние с наименьшей энергией должно соответствовать самому короткому маршруту. Функция энергии должна удовлетворять следующим требованиям:

- 1) должна поддерживать устойчивое состояние в форме матрицы

$$V = \{v_{Xi}\}, \quad 1)$$

17. которой строки соответствуют городам, столбцы - их номерам в маршруте; в каждой строке и каждом столбце только одна единица, остальные нули;

И из всех решений вида (1) функция энергии должна поддерживать те, которые соответствуют коротким маршрутам.

Таким требованиям удовлетворяет функция энергии в виде:

$$E = (A/2) \sum_X \sum_i \sum_{j \neq i} v_{Xi} v_{Xj} + (B/2) \sum_i \sum_X \sum_{Y \neq X} v_{Xi} v_{Yi} + (C/2) (\sum_X \sum_i v_{Xi} - n)^2 + (D/2) \sum_X \sum_{X \neq Y} \sum_i d_{XY} v_{Xi} (v_{Y, i-1} - 1), \quad (1)$$

где первые три члена поддерживают первое требование, четвертый член — второе. Первый член равен нулю, если каждая строка  $X$  содержит не более одной единицы. Второй равен нулю, если каждый столбец  $i$  содержит не более одной единицы. Третий равен нулю, если в матрице всего  $n$  единиц. Короткие маршруты поддерживает четвертый член. В нем индексы  $i$  берутся по модулю  $n$  для того, чтобы показать, что  $n$ -й город соседствует в



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

маршруте с  $(n - 1) - M$ , т.е.  $v_{Y,n+j} = v_{Y,j}$ . Четвертый член численно равен *длине маршрута*. Каноническое выражение для функции вычислительной энергии имеет вид



$$E = -(1/2) \sum_X \sum_i \sum_Y \sum_j W_{Xi,Yj} v_{Xi} v_{Xj} - \sum_{xi} I_X \quad 3)$$

Из (2) и (3) получаем веса сети Хопфилда:

$$W_{Xi,Yj} = -A\delta_{XY}(1 - \delta_{ij}) - B\delta_{ij}(1 - \delta_{XY}) - C - Dd_{XY}(\delta_{j,i+1} + \delta_{j,i-1}),$$

$$I_{Xi} = Cn.$$

Здесь  $\delta$  - символ Кронекера.

*Моделирование работы сети Хопфилда* показало, что лучшее по качеству решение дает сеть, нейроны которой имеют сигмовидную характеристику, а сеть, в которой нейроны имеют ступенчатые переходы, приходила к финальным состояниям, соответствующим маршрутам немного лучшим, чем случайные. Многочисленные исследования показывают, что качество решения задачи минимизации функции энергии (2) существенно зависит от выбора производной сигмовидной униполярной функции активации нейрона в окрестности нуля. При малой величине производной минимумы энергии оказываются в центре гиперкуба решений (некорректное решение), при большой величине производной сеть Хопфилда попадает в вершину гиперкуба, соответствующую локальному минимуму функции энергии. Кроме того, на качество решения существенное влияние оказывает выбор коэффициентов  $A, B, C, D$ . Поиск методов оптимального выбора этих коэффициентов является в настоящее время предметом интенсивных исследований.

### ***Машина Больцмана***

Математической основой для решения комбинаторных оптимизационных задач на машине Больцмана является алгоритм, моделирующий затвердевание жидкостей или расплавов (алгоритм имитации отжига). Он базируется на идеях из двух различных областей: статистической физики и комбинаторной оптимизации. Машина Больцмана (МБ) способна реализовать этот алгоритм параллельно и асинхронно. МБ задается

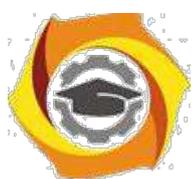
четверкой  $B = (N, E, W, V_0)$ ,  $N$  - число нейронов,  $E = \{(i, j)\}$  - множество связей между нейронами, при этом все автосвязи принадлежат этому множеству, т.е.  $(i, i) \in E$ .

Каждый нейрон может иметь состояние 0 или 1. Состояние  $V_k$  МБ определяется состояниями нейронов  $V_k = (v_1^k, \dots, v_N^k), V_0$  - начальное состояние.

Каждая связь  $(i, j)$  имеет вес  $w_{ij}$  - вещественное число, множество связей  $W$ . Связь  $(i, j)$  называется активной в состоянии  $V_k$ , если  $v_i^k v_j^k = 1$ .

Вес связи  $(i, j)$  интерпретируется как количественная мера желательности, чтобы эта связь была активной. При  $w_{ij} \gg 0$  - активность очень желательна, при  $w_{ij} \ll 0$  - активность очень нежелательна. Как и в модели Хопфилда, связи в МБ симметричны, т.е.  $w_{ij} = w_{ji}$ .

### ***Функция консенсуса***



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Для состояния  $V_k$  МБ вводится понятие консенсуса



$$C_k = \sum_{i,j} w_{ij} v_i^k v_j^k.$$

Каждая *связь* в этой сумме учитывается со своим весом. Консенсус  $C_k$  интерпретируется как количественная *мера* желательности, чтобы все связи  $(i, j)$  в состоянии  $V_k$  были активны. Для состояния  $V_k$  определяется множество соседей  $V^{(k)}$ . Соседнее состояние  $V_{k(i)} \in V^{(k)}$  получается из  $V_k$  при изменении состояния нейрона  $i$ ,

$$V_j^{k(i)} = \begin{cases} v_j^k & \text{если } j \neq i \\ 1 - v_j^k & \text{если } j = i \end{cases}$$

Разница консенсусов соседних состояний  $V_k$  и  $V_{k(i)}$  равна

$$\Delta C_{kk(i)} = C_{k(i)} - C_k = (1 - 2v_i^k) \left( \sum_{(i,j) \in E(i)} w_{ij} v_i^k + w_{ii} \right),$$

где  $E(i)$  - множество связей нейрона  $i$ . Видно, что  $\Delta C_{kk(i)}$  для всех  $V_{k(i)} \in V^{(k)}$  могут вычисляться параллельно.

### **Максимизация консенсуса**

Переход МБ из одного состояния в другое с максимизацией консенсуса происходит путем выполнения пошаговой процедуры. На каждом ее шаге выполняется *слепое испытание*, состоящее из двух частей:

1. для данного состояния  $V_k$  генерируется соседнее  $V_{k(i)}$ ,
2. оценивается, может ли быть принято состояние  $V_{k(i)}$  или же, по результату испытания -, иначе,  $V_k$ .

Состояние  $V_{k(i)}$  принимается с вероятностью

$$P_{kk(i)}(t) = 1/[1 + \exp(\Delta C_{kk(i)}/t)], \quad 4)$$

где  $t \geq 0$  - управляющий параметр ("температура").

Процесс максимизации консенсуса начинается с высокого значения  $t_0$  параметра  $t$  и случайно выбранного начального состояния  $V_0$ . В течение процесса параметр  $t$  уменьшается от  $t_0$  до 0. По мере того как  $t$  приближается к нулю, нейроны все реже изменяют свои состояния,



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

· наконец, МБ стабилизируется в финальном состоянии. Практически, МБ стабилизируется в состоянии, соответствующем *локальному максимуму* консенсуса, который близок (или равен) глобальному. Сходимостью МБ управляют следующие параметры:



1. Начальное значение параметра  $t$  для каждого нейрона  $i$

$$t_0^{(i)} = \sum_{(i,j) \in E(i)} |w_{ij}| + |w_{ii}|.$$

2. Правило понижения  $t$

$$t_{j+1}^{(i)} = \alpha t_j^{(i)},$$

где  $\alpha$  - положительное число, меньшее единицы, но близкое к ней.

3. Число  $L$  испытаний, которые проводятся без изменения  $t$  ( $L$  — функция от  $N$ ).

4. Число  $M$  последовательных испытаний, не приводящих к изменению состояния машин ( $M$  - функция от  $N$ ), как критерий завершения процесса.

### Синхронное и асинхронное функционирование машины Больцмана

Для выполнения синхронного процесса все множество нейронов разбивается на непересекающиеся подмножества  $\{M_1, \dots, M_m\}$ , такие, что нейроны, попавшие в одно подмножество, не связаны друг с другом. В каждом подмножестве синхронизация элементов подмножества происходит одновременно. Вибранного подмножества могут одновременно изменять свои состояния в соответствии с заданной вероятностью.

В асинхронном параллельном процессе все нейроны могут изменять свои состояния только в зависимости от величины вероятности. Практически асинхронный параллелизм может быть выполнен следующим образом. Случайно выбирается подмножество  $M$ , содержащее  $q = 2N/3$  нейронов. Для каждого нейрона из этого подмножества устанавливается состояние в соответствии с  $P_{kk(i)}(t)$ . Получившееся в результате состояние есть результат одного асинхронного шага.

### Решение задачи коммивояжера машиной Больцмана

Общий подход к программированию комбинаторных оптимизационных задач состоит в следующем:

каждое решение представляется набором  $\{x_1, \dots, x_N\}$ ,  $x_i \in \{0, 1\}$  — число нейронов в структуре связей и веса выбираются так, что

R1. Все локальные максимумы функции консенсуса соответствуют приемлемым решениям задачи;

R2. Чем лучше приемлемое решение, тем больше консенсус соответствующего состояния машины Больцмана.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Перефразируем для МБ *задачу коммивояжера*.



$R1$  и Состояние МБ соответствует **локальному максимуму** функции консенсуса, если и только если это состояние соответствует приемлемому маршруту.

$R2$ . Чем короче маршрут, тем выше консенсус соответствующего состояния МБ.

Каждый нейрон соответствует элементу матрицы  $n \times n$ , состояния нейронов обозначаются  $v_{Xi}$  ( $n$  - число городов). Функция консенсуса

$$C_k = \sum_{(Xi, Yj)} w_{Xi, Yj} v_{Xi}^k v_{Yj}^k.$$

Множество связей в сети определяется как **объединение** трех непересекающихся подмножеств:

$E_d$  - множество связей, несущих информацию о расстояниях между городами.

$$E_d = \{(Xi, Yj) | (X \neq Y) \wedge (i = (j + 1) \bmod n)\};$$

$E_i$  : множество ингибиторных (запретительных) связей,

$$E_i = \{(Xi, Yj) | (i \neq j) \wedge (X = Y) \vee (i = j) \wedge (X \neq Y)\};$$

$E_b$  : множество связей смещений,

$$E_b = \{(Xi, Yj) | (X = Y) \wedge (i = j)\}.$$

Здесь  $X, Y, i, j = 1, \dots, n$ . Общее число связей равно  $2n^3 - n^2$ .

Ингибиторные связи гарантируют, что, в конце концов, ни в одной строке и ни в одном столбце не будет более одной единицы. Связи смещений гарантируют, что хотя бы в одной единице есть в каждом столбце и в каждой строке. Таким образом, связи гарантируют выполнение ограничений  $E_i$  и  $E_b$  веса задают одинаковый вклад в консенсус для всех приемлемых маршрутов.

Связь  $(Xi, Yj) \in E_d$  активна только в том случае, когда в маршруте есть прямой путь из города  $X$  в город  $Y$ . Вес связи  $(Xi, Yj) \in E_d$  равен расстоянию между городами  $X$  и  $Y$  с отрицательным знаком. Следовательно, для данного маршрута отрицательный вклад связи из  $E_d$  в консенсус пропорционален длине пути, поэтому максимизация функции консенсуса соответствует минимизации длины маршрута.



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

Доказано, что для консенсуса  $C_k$  выполняются требования  $R1$  и  $R2$ , если и только если веса связей выбраны следующим образом:

$$\begin{aligned}\forall (X_i, Y_j) \in E_d : w_{X_i, Y_j} &= -d_{XY}, \\ \forall (X_i, Y_j) \in E_i : w_{X_i, Y_j} &< -\min(\mu_X, \mu_Y),\end{aligned}$$



$$\forall (X_i, Y_j) \in E_b : w_{X_i, Y_j} > \mu_X,$$

где

$$\mu_X = \max\{d_{XP} + d_{XQ} | P, Q = 1, \dots, n \wedge (P \neq Q)\}.$$

При  $d = 0,95, L = 10, M = 100$  было проведено 100 испытаний для  $n = 10$  и 25 испытаний для  $n = 30$  при различных начальных состояний МБ. Для  $n = 10$  получено оптимальное решение, для  $n = 30$  получено решение на 14% хуже оптимума. Вероятностный механизм функционирования МБ дает возможность получать на ней несколько лучшие результаты, чем на модели Хопфилда.

#### Контрольные вопросы ЛР6(ПК-4):

- Решение задачи коммивояжера сетью Хопфилда.
- Машина Больцмана.
- Функция консенсуса.
- Максимизация консенсуса.
- Синхронное и асинхронное функционирование машины Больцмана.
- Решение задачи коммивояжера машиной Больцмана.
- Мягкая экспертная система.
- Определение мягкой экспертной системы.
- Сравнение нечеткой и мягкой экспертных систем.
- Представление знаний в мягкой экспертной системе.
- Содержание баз знаний и данных мягкой экспертной системы.
- Основные элементы диаграмм взаимодействия — объекты, сообщения.
- Диаграммы состояний.
- Диаграммы начального состояния, конечного состояния, переходы.
- Вложенность состояний.
- Диаграммы внедрения: подсистемы, компоненты, связи.



**Лабораторная работа №7. Самоорганизация (самообучение) нейронных сетей. Классификация без учителя. Метод динамических ядер в классификации без учителя. Алгоритмы обучения сетей с самоорганизацией. Алгоритм Кохонена. Применение сетей в самоорганизацией. Компрессия данных. Прогнозирование нагрузок энергетической системы.**

**Аннотация:** Рассматриваются: метод динамических ядер в классификации без учителя, алгоритмы обучения сетей с самоорганизацией и их применение к компрессии данных и прогнозированию.

**Ключевые слова:** [вектор](#), [мера](#), [евклидово расстояние](#), [расстояние](#), [определение](#), [значение](#), [класс](#), [число классов](#), [выборка](#), [пространство](#), [ядро](#), [разбиение](#), [минимум](#), [решающее правило](#), [сумматор](#), [поиск](#), [максимум](#), [норма](#), [сеть](#), [погрешность](#), [вес](#), [vector](#), [quantization](#), [алгоритм](#), [WTA](#), [ALL](#), [активность](#), [нейрон](#), [скалярное произведение](#), [MOST](#), [функция](#), [feature](#), [MAP](#), [самоорганизующаяся карта](#), [Размещение](#), [кластер](#), [кадр](#), [компонент](#), [знание](#)

### ***Классификация без учителя***

Задан набор объектов, каждому объекту поставлен в соответствие *вектор* значений признаков (строка таблицы). Требуется разбить эти объекты на классы эквивалентности. Для каждого нового объекта нужно:

- Найти класс, к которому он принадлежит.
- Использовать новую информацию, полученную об этом объекте, для исправления (коррекции) правил классификации.

Отнесение объекта к классу проводится путем его сравнения с типичными элементами разных классов и выбора из них ближайшего.

Простейшая *мера* близости объектов - квадрат *евклидова расстояния* между векторами значений их признаков (чем меньше *расстояние*, тем ближе объекты).

Соответствующее *определение* признаков типичного объекта - среднее арифметическое *значение* признаков по выборке, представляющей *класс*. Другая *мера* близости, возникающая при обработке сигналов, изображений и т.п. - квадрат коэффициента корреляции (чем он больше, тем ближе объекты). Возможны и иные варианты.

Если *число классов  $m$*  заранее определено, то задачу классификации без учителя можно поставить следующим образом.

### ***Метод динамических ядер в классификации без учителя***

Пусть задана *выборка* предобработанных векторов данных - *пространство* векторов данных. Каждому классу будет соответствовать некоторое *ядро* - *пространство* ядер.

Для любых  $i$  и  $j$  определим меру близости  $d_{ij}$ , а для каждого набора  $X$  из ядер  $Y$  и  $Z$  любого



$$w \subseteq W, W$$

# Негосударственное частное образовательное учреждение

$$x \in E \quad w \in W \quad d(x, w)$$

разбиения  $\{x\}$  на  $k$  классов  $\{x\} = P_1 \cup P_2 \cup \dots \cup P_k$  определим критерий качества



$$D = D(w_1, \dots, w_k, P_1, \dots, P_k) = \sum_{i=1}^k \sum_{x \in P_i} d(x, w_i) \quad (1)$$

Требуется найти набор ядер  $w_1, \dots, w_k$  и разбиение  $P_1, \dots, P_k$ , минимизирующие  $D$ . Шаг алгоритма разбиваем на 2 этапа:

1) Для фиксированного набора ядер  $w_1, \dots, w_k$  ищем минимизирующее  $D$  разбиение  $P_1, \dots, P_k$ ; оно дается следующим решающим правилом:  $x \in P_i$ , если  $d(x, w_i) < d(x, w_j)$  при  $i \neq j$  (когда для минимума  $d(x, w_i)$  достигается при нескольких значениях  $i$ , выбор между ними может быть сделан произвольно).

2) Для каждого  $P_i, i \in 1, \dots, k$ , полученного на первом этапе, отыскивается  $w_i \in W$ , минимизирующее критерий качества

$$D_i = \sum_{x \in P_i} d(x, w_i).$$

Начальные значения  $w_1, \dots, w_k, P_1, \dots, P_k$  выбираются произвольно либо по какому-нибудь эвристическому правилу. Если ядру  $w_i$  ставится в соответствие элемент сети, вычисляющей по входному сигналу  $x$  функцию  $d(x, w_i)$ , то решающее правило для классификации дается инверсным оператором программирования: каждый элемент  $x$  принадлежит классу  $P_i$ , если выходной сигнал  $i$ -го элемента  $d(x, w_i)$  меньше всех остальных. Мера близости  $d$  выбирается такой, чтобы легко можно было найти ядро  $w_i$ , минимизирующее  $D_i$  для данного  $P_i$ .

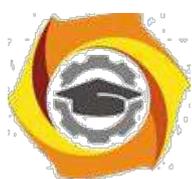
В простейшем случае пространство ядер  $W$  совпадает с  $E$ , а  $d(x, w_i)$  - положительно определенная квадратичная форма от  $x - w_i$ , например, квадрат евклидова расстояния. Тогда ядро  $w_i$ , минимизирующее  $D_i$ , есть центр масс класса  $P_i$ :

$$w_i = (1/|P_i|) \sum_{x \in P_i} x,$$

где  $|P_i|$  - число элементов в  $P_i$ .

Пусть векторы пространства  $E$  нормированы. Тогда

$$(x, x) = (w_i, w_i) = 1.$$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

2)



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Так как  $d(x, w_i) = (x - w_i, x - w_i) = (x, x) - 2(x, w_i) + (w_i, w_i)$ , то с учетом (2) упрощается решающее правило, разделяющее классы:

$$x \in P_i, \text{ если } (x, w_i) > (x, w_j) \text{ при } i \neq j,$$

поскольку минимум  $d(x, w_i)$  достигается при максимуме  $(x, w_i)$ . Такое решающее правило реализуется с помощью  $k$  сумматоров, вычисляющих  $(x, w_i)$ , и интерпретатора, выбирающего сумматор с максимальным выходным сигналом. Номер этого сумматора есть номер класса, к которому относится  $x$ .

Задача поиска ядра для класса превращается в поиск вектора  $w_i$ , максимизирующего изирующего

$$D_i = \sum_{x \in P_i} (x, w).$$

Этот максимум достигается в точке

$$w = \sum_{x \in P_i} x / \left\| \sum_{x \in P_i} x \right\|$$

где  $\| \dots \|$  - евклидова норма.

В тех простейших случаях, когда ядро класса точно определяется как среднее арифметическое (или нормированное среднее арифметическое) элементов класса, а решающее правило основано на сравнении выходов линейных и нелинейных сумматоров, нейронную сеть, реализующую метод динамических ядер, называют сетью Кохона и называют сетью Кохона

В определении ядра  $w_i$  для сетей Кохона вводят сумму  $\sum_{x \in P_i} x$ . Это позволяет вводить новые динамические ядра, обрабатывающие один или несколько чисел в выходном сигнале в  $P_i$  нового примера.

Если число классов заранее не определено, то полезным критерием слияния классов  $Y_i$  и  $Y_j$  является расстояние между их ядрами меньше, чем среднее расстояние от элементов класса до ядра в одном из них:

$$(y^i, y^j) < \max \left[ (1/|Y_i|) \sum_{x \in Y_i} \rho(x, y^i), (1/|Y_j|) \sum_{x \in Y_j} \rho(x, y^j) \right],$$

где  $|Y|$  - число элементов в  $Y$ . Использовать критерий слияния классов можно



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

где  $n$  - число элементов в  $S$ . Использовать критерий слияния классов можно так: сначала принимаем гипотезу о достаточном числе классов, строим их, минимизируя  $D$ , затем некоторые  $Y_i$  объединяем, повторяем минимизацию  $D$  с новым числом классов и т.д.



### Алгоритмы обучения сетей с самоорганизацией

Целью обучения сети с самоорганизацией на основе конкуренции нейронов считается такое упорядочение нейронов (подбор значений их весов), которое минимизирует значение ожидаемого искажения, оцениваемого погрешностью аппроксимации входного вектора  $x$  значениями весов нейрона-победителя. При  $\mathcal{D}$  входных векторах  $x$  и применении евклидовой метрики эта погрешность, называемая также погрешностью квантования, может быть выражена в виде

$$E = (1/p) \sum_{i=1}^p \|x^i - w_{win}\|^2, \quad 3)$$

где  $w_{win}$  - вес нейрона-победителя при предъявлении вектора  $x^i$ .

Этот подход также называется векторным квантованием (англ. *Vector Quantization* - VQ) или кластеризацией. Номера нейронов-победителей при последовательном предъявлении векторов  $x^i$  образуют так называемую кодовую таблицу. При классическом решении задачи кодирования применяется алгоритм  $K$ -усреднений (англ. K-means), носящий имя обобщенного алгоритма Ллойда.

Для нейронных сетей аналогом алгоритма Ллойда считается алгоритм WTA (англ.: Winner Takes All - "победитель получает все"). В соответствии с ним после предъявления вектора  $x$  рассчитывается активность каждого нейрона. Победителем признается нейрон с самым сильным выходным сигналом, т.е. тот, для которого скалярное

произведение  $(x, w)$  оказывается наибольшим. В предыдущем разделе было показано, что при использовании нормализованных векторов это равнозначно наименьшему евклидову расстоянию между входным вектором и вектором весов нейронов. Победитель получает право уточнить свои веса в направлении вектора  $x$  согласно правилу

$$w_{win} \leftarrow w_{win} + \alpha(x - w_{win}),$$

где  $\alpha$  - коэффициент обучения. Веса остальных нейронов уточнению не подлежат. Алгоритм позволяет учитывать усталость нейронов путем подсчета количества побед каждого из них и поощрять элементы с наименьшей активностью для выравнивания их шансов. Такая модификация применяется чаще всего на начальной стадии обучения с последующим отключением после активизации всех нейронов. Подобный способ обучения реализован в виде режима CWTA (Conscience Winner Takes All) и считается одним из лучших и наиболее быстрых алгоритмов самоорганизации.

Помимо алгоритмов WTA, в которых в каждой итерации может обучаться только один нейрон, для обучения сетей с самоорганизацией широко применяются алгоритмы типа WTM (англ.: Winner Takes Most - "победитель получает больше"), в которых, кроме победителя, уточняют значения своих весов и нейроны из его ближайшего окружения. При этом, чем дальше какой-либо нейрон находится от победителя, тем меньше изменяются его веса. Процесс уточнения вектора весов может быть определен обобщенной зависимостью, которая здесь представляется в виде

$$w_i \leftarrow w_i + \alpha G(i, x)[x - w_i]$$



для  $G(i, x)$  <sup>всех</sup> нейронов, расположенных в окрестности победителя.

Если функция определяется в форме

$$G(i, x) = \{1 \text{ для } i = I, 0 \text{ для } i \neq I\},$$

где  $I$  обозначает номер победителя, то мы получаем классический алгоритм WTA. Существует множество вариантов алгоритма WTM, отличающихся прежде всего формой функции  $G(i, x)$ . Для дальнейшего изучения выберем классический алгоритм Кохонена.

### **Алгоритм Кохонена**

Алгоритм Кохонена относится к наиболее старым алгоритмам обучения сетей с самоорганизацией на основе конкуренции, и в настоящее время существуют различные его версии. В классическом алгоритме Кохонена сеть инициализируется путем присписывания нейронам определенных позиций в пространстве и связывания их с соседями на постоянной основе. Такая сеть называется самоорганизующейся картой признаков (сеть SOFM - Self-Organizing Feature Map). В момент выбора победителя уточняются не только его веса, но также и веса его соседей, находящихся в ближайшей окрестности. Таким образом, нейрон-победитель подвергается адаптации вместе со своими соседями. В классическом алгоритме

Кохонена функция соседства  $G(i, x)$  определяется в виде

$$G(i, x) = \{1 \text{ для } d(i, I) \leq L, 0 \text{ для } d(i, I) > L\}.$$

В этом выражении  $d(i, I)$  обозначает евклидово расстояние между векторами весов нейрона-победителя  $I$  и  $i$ -го нейрона. Коэффициент  $L$  выступает в роли уровня соседства, его значение уменьшается в процессе обучения до нуля. Соседство такого рода называется прямоугольным.

Другой тип соседства, часто применяемый в картах Кохонена, - это соседство гауссовского типа, при котором функция  $G(i, x)$  задается формулой

$$G(i, x) = \exp(-d^2(i, x)/2\lambda^2).$$

Степень адаптации нейронов-соседей определяется не только евклидовым расстоянием между  $i$ -м нейроном и победителем ( $I$ -м нейроном), но также и уровнем соседства  $\lambda$ . В отличие от соседства прямоугольного типа, где каждый нейрон, находящийся в окрестности победителя, адаптировался в равной степени, при соседстве гауссовского типа уровень адаптации различен и зависит от значения функции Гаусса. Как правило, гауссовское соседство дает лучшие результаты обучения и обеспечивает лучшую организацию сети, чем прямоугольное соседство.

Самоорганизующаяся карта признаков проходит два этапа обучения. На первом этапе элементы упорядочиваются так, чтобы отражать пространство входных элементов, а на втором происходит уточнение их позиций. Как правило, процесс представляется визуально путем использования двумерных данных и построения соответствующей поверхности. Например, входные векторы выбираются случайным образом на основе однородного распределения в некотором квадрате, и начинается обучение карты. В определенные моменты в ходе обучения строятся изображения карты путем использования соответствия, показанного на рис. 1. Элементы

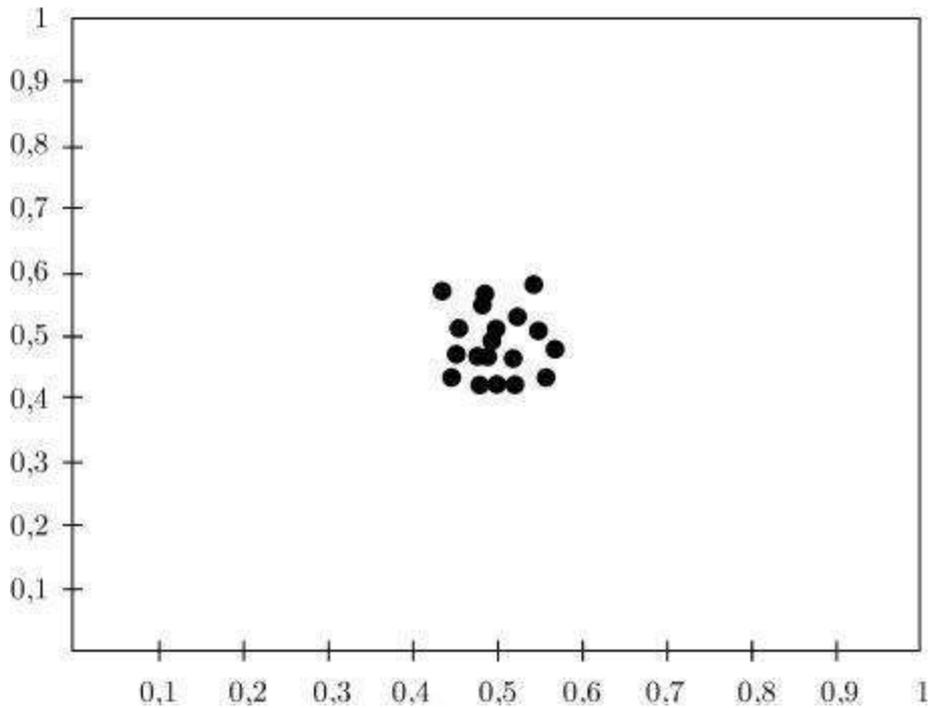


**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

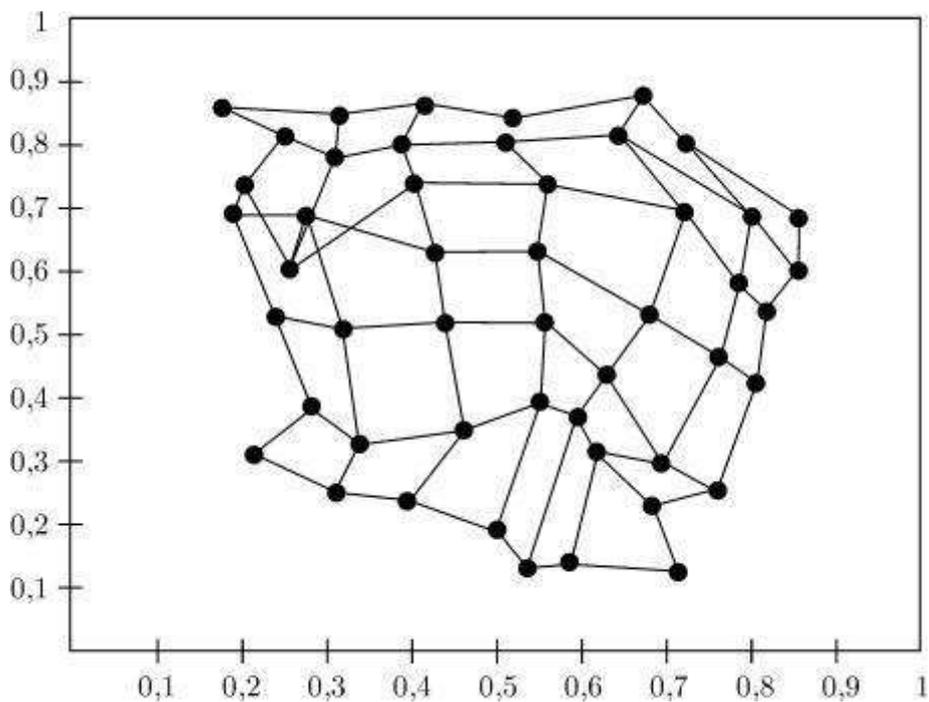
соединяются линиями, чтобы показать их относительное *размещение*. Сначала карта выглядит сильно "измятой", но постепенно в ходе обучения она разворачивается и расправляется. Конечным



результатом обучения является карта, покрывающая все входное *пространство* и являющаяся достаточно регулярной (т.е. элементы оказываются распределенными почти равномерно). Для примера была рассмотрена карта с топологией квадрата из 49 элементов, и для 250 точек данных, взятых из единичного квадрата, было проведено ее обучение, которое начиналось со случайного набора весовых значений, задающих *размещение* кластерных элементов в центре входного пространства, как показано на рис. 1. На рис. 2 и 3 иллюстрируется процесс разворачивания карты в течение времени. Как и для других типов сетей, в данном случае результат обучения зависит от учебных данных и выбора параметров обучения.



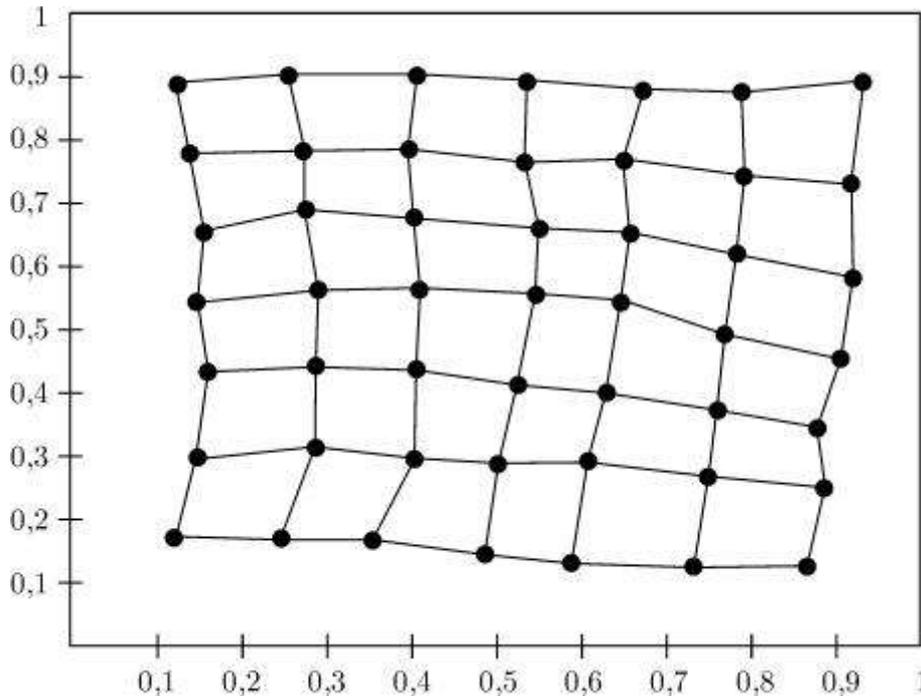
**Рис. 1.** Весовые векторы инициализируются случайными значениями из диапазона 0.4-0.6





**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Рис. 2.** Карта по прошествии 20 итераций



**Рис. 3.** Карта незадолго до окончания обучения. Элементы теперь упорядочены, и карта станет еще более регулярной по окончании финальной фазы сходимости

### ***Применение сетей с самоорганизацией***

Главным свойством сети Кохонена считается компрессия данных, состоящая в том, что образующие *кластер* группы данных представляются единственным вектором весов нейрона-победителя. При разделении данных на кластеры и представлении каждого кластера одним из нейронов достигается значительное сокращение объема используемой под данные памяти, которое и называется компрессией. Это компрессия с потерей информации, которая сопровождается определенной погрешностью квантования.

### ***Компрессия данных***

Примером использования компрессионных свойств сети Кохонена может считаться сжатие изображений, предназначенное для уменьшения количества информации, представляющей конкретный образ, при сохранении погрешности восстановления на заданном уровне.

Пусть изображение разделяется на одинаковые кадры размером  $n_x \times n_y$  пикселей. Образующие *кадр* пиксели представляют собой компоненты входного вектора  $\mathcal{X}$ .

*Сеть* с самоорганизацией содержит  $n$  нейронов, каждый из которых имеет входом *вектор*  $\mathcal{X}$ . Обучение сети при помощи одного из алгоритмов самоорганизации состоит в подборе таких весов конкретных нейронов, при которых минимизируется *погрешность* квантования (3). В результате обучения формируется структура сети, при которой вектору  $\mathcal{X}$  каждого кадра соответствует *вектор* весов нейрона победителя. В процессе предъявления очередного кадра выбирается номер нейрона-победителя. Номера нейронов-победителей образуют кодовую таблицу, а веса этих нейронов представляют средние значения, соответствующим конкретным компонентам вектора  $\mathcal{X}$  (т.е. уровням интенсивности пикселей, составляющих *кадр*).



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Поскольку количество нейронов обычно намного меньше количества кадров, то можно получить существенное сокращение объема данных, описывающих исходное изображение. В итоге коэффициент компрессии изображения равен



$$K = N \cdot n_x n_y T / (N \cdot \lg_2 n + n \cdot n_x n_y t),$$

где  $n_x$  и  $n_y$  - размеры кадра в осях  $x$  и  $y$ ,  $N$  - количество кадров,  $n$  - количество нейронов, а  $T$  и  $t$  - количество битов для представления соответственно градаций интенсивности пиксела и значений весов. Этот подход позволяет получить степень компрессии изображений порядка 16 при значении коэффициента сигнал/шум (PSNR) около 26-28 дБ.

### **Прогнозирование нагрузок энергетической системы**

Рассмотрим решение задачи прогнозирования часовых нагрузок в электроэнергетической системе на 24-часовом интервале. Пусть имеется база данных, содержащая векторы профильных нагрузок дня

$$p_j = [p(j, 1), p(j, 2), \dots, p(j, 24)],$$

где компонент  $p(j, k)$  соответствует действительной нагрузке в  $k$ -й час суток. Множество профильных векторов подается на вход сети Кохонена, состоящей из  $n$  нейронов. Процесс самоорганизации сети приводит к автоматической кластеризации данных и к сопоставлению каждому кластеру одного из нейронов сети. Этот *нейрон* считается победителем, а его веса наилучшим образом адаптируются к усредненным весам профильных векторов, составляющих *кластер*. Характерная особенность состоит в том, что соседние векторы имеют сходные профильные характеристики.

Близость весов нейронов, расположенных недалеко друг от друга, объясняется тем, что один и тот же день в разные годы при небольших отличиях в часовых нагрузках может возбуждать различные нейроны, которые образуют кластеры, группирующие данные сходных классов.

*Знание* таблицы распределения побед конкретных нейронов сети позволяет относительно легко предвидеть профили часовых нагрузок для произвольного дня года. С этой целью создаются таблицы принадлежности каждого дня года к области доминирования определенного нейрона с обозначением количества его побед для всех дней в прошлом. Для выбора прогнозируемого профиля нагрузок актуального дня в требуемом месяце рассчитываются усредненные значения весов нейронов победителей, которые указывали в прошлом на требуемый день. Если количество

побед  $i$ -го нейрона, соответствующего  $j$ -му дню, обозначить  $k_{ji}$ , а соответствующие векторы

весов класса -  $w_i$ , то прогнозируемый профильный вектор  $j$ -го дня рассчитывается по формуле

$$p_j = \frac{\sum_{i=1} k_{ji} w_i}{\sum_{i=1} k_{ji}}$$



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

**Контрольные вопросы ЛР7(ПК-4):**

- к Классификация без учителя.
  - к Метод динамических ядер в классификации без учителя.
  - к Алгоритмы обучения сетей с самоорганизацией.
  - к Алгоритм Кохонена.
  - к Применение сетей с самоорганизацией.
  - к Компрессия данных.
  - к Прогнозирование нагрузок энергетической системы.
- 
- Модуль лингвистической обработки.
  - Лингвистический анализ.
  - Формирование просодических характеристик.
  - Синтезатор русской речи.
  - Язык формальной записи правил синтеза.
  - Интонационное обеспечение.
  - Аллофонная база данных.
  - Лингвистический анализ.
  - Инструментарий синтеза русской речи.
  - Система распознавания речи.
  - Акустическая и лингвистическая модели.
  - Классификация систем распознавания.



*Лабораторная работа №8. Нечеткие и гибридные нейронные сети. Интеллектуальные информационные системы в условиях неопределенности и риска. Нечеткие множества. Лингвистические переменные. Нечеткие правила вывода. Системы нечеткого вывода Мамдани-Заде. Фазификатор. Дефазификатор. Модель Мамдани-Заде как универсальный аппроксиматор. Нечеткие сети TSK (Такаги-Сугено-Канга). Гибридный алгоритм обучения нечетких сетей. Мягкая экспертная система. Определение мягкой экспертной системы. Сравнение нечеткой и мягкой экспертных систем. Представление знаний в мягкой экспертной системе. Содержание баз знаний и данных мягкой экспертной системы.*

**Аннотация:** Рассматриваются: математические основы нечетких систем, преимущества и алгоритмы обучения нечетких нейронных сетей, нечеткие сети с генетической настройкой, экспертные системы на основе гибридных НС.

***Интеллектуальные информационные системы в условиях неопределенности и риска***

в помощью *символьной обработки информации* не удастся решить прикладные задачи многих предметных областей, если для них невозможно получить полную информацию и если их *определение* недостаточно полно. Такая ситуация характерна для:

- сложных технических систем;
- систем экономического планирования;
- социальных систем большой размерности;
- систем принятия решений и т.п.

Выходом является использование систем, основанных на мягких вычислениях, которые включают в себя:

- нечеткую логику и вероятностные вычисления;
  - нейрокомпьютинг - обучение, адаптация, классификация, системное моделирование
- В идентификация;
- генетические вычисления - синтез, настройка и оптимизация с помощью систематизированного случайного поиска и эволюции.

Эти составные части не конкурируют друг с другом, а создают эффект взаимного усиления (гибридные системы). Наряду с термином "мягкие вычисления" используется термин "вычислительный интеллект" - научное направление, где решаются задачи искусственного интеллекта на основе теории нечетких систем, нейронных сетей и эволюционных (генетических) вычислений.

Нечеткие нейронные сети с генетической настройкой параметров (гибридные системы) демонстрируют взаимное усиление достоинств и нивелирование недостатков отдельных методов:

11) Представление знаний в нейронных сетях в виде *матриц весов* не позволяет объяснить результаты проведенного распознавания или прогнозирования, тогда как в системах вывода на базе нечетких правил результаты воспринимаются как ответы на вопросы "почему?".

12) Нейронные сети обучаются с помощью универсального алгоритма, т.е. трудоемкое *извлечение знаний* заменяется сбором достаточной по объему обучающей выборки. Для



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

нечетких систем вывода *извлечение знаний* включает в себя сложные процессы формализации понятий, определение функций принадлежности, формирование правил вывода.



к Нечеткие нейронные сети обучаются как нейронные сети, но их результаты объясняются как в системах нечеткого вывода.

***Нечеткие множества***

Понятие нечетких множеств (fuzzy sets) как *обобщение* обычных (четких) множеств было введено Л.Заде в 1965 г.. Традиционный способ представления элемента множества  $A$  состоит в применении характеристической функции  $\mu_A(x)$ , которая равна 1, если элемент принадлежит множеству  $A$ , или равна 0 в противном случае. В нечетких системах элемент может частично принадлежать любому множеству. Степень принадлежности элементу  $x$  множеству  $A$ , представляющая собой *обобщение* характеристической функции, называется функцией принадлежности  $\mu_A(x)$ , причем  $\mu_A(x) \in [0, 1]$ , и  $\mu_A(x) = 0$  означает отсутствие принадлежности  $x$  множеству  $A$ , а  $\mu_A(x) = 1$  -- полную принадлежность. Конкретное значение функции принадлежности называется степенью или коэффициентом принадлежности.

***Лингвистические переменные***

20. теории нечетких множеств, помимо переменных цифрового типа, существуют лингвистические переменные с приписываемыми им значениями.

Пусть  $x$  обозначает температуру. Можно определить нечеткие множества "отрицательная", "близкая к нулю", "положительная", характеризуемые функциями принадлежности  $\mu_{отриц}(x)$ ,  $\mu_{нул}(x)$ ,  $\mu_{полож}(x)$ . Лингвистическая переменная "температура" может принимать значения "отрицательная", "близкая к нулю", "положительная". Функция нечеткой принадлежности является непрерывным приближением пороговой функции точной принадлежности.

***Нечеткие правила вывода***

*Правило вывода*

если это  $A$ , то  $Y$  это  $B$

называется нечеткой импликацией  $A \rightarrow B$ , если  $A$  и  $B$  -- лингвистические значения (значения лингвистической переменной), идентифицированные нечетким способом через соответствующие функции принадлежности для переменных.

Часть "это  $A$ " называется условием (предпосылкой), а "это  $B$ " -- следствием (заключением).

Обобщение для  $N$ -мерного вектора  $x$  :

если  $x_1$  это  $A_1$  и  $x_2$  это  $A_2$  и... и  $x_N$  это  $A_N$ , то  $Y$  это  $B$ ,  $A_1, A_2, \dots, A_N, B$  обозначают величины соответствующих коэффициентов принадлежности  $\mu_A(x_i), i = 1, 2, \dots, N, \mu_B(y)$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Возможна интерпретация  $\mu_A(x)$



34. в форме логического произведения

$$\mu_A(x) = \min_{i=1, \dots, N} \mu_A(x_i)$$

35. в форме алгебраического произведения

$$\mu_A(x) = \prod_{i=1, \dots, N} \mu_A(x_i)$$

(агрегирование предпосылки).

Каждой импликации  $A \rightarrow B$  можно приписать значение функции принадлежности  $\mu_{A \rightarrow B}(x, y)$ :

22. форма логического произведения

$$\mu_{A \rightarrow B} = \min\{\mu_A(x), \mu_B(y)\}$$

23. форма алгебраического произведения

$$\mu_{A \rightarrow B} = \mu_A(x) \mu_B(y)$$

(агрегирование на уровне импликации).

### **Системы нечеткого вывода Мамдани-Заде**

Элементы теории нечетких множеств, правила импликации и нечетких рассуждений образуют систему нечеткого вывода. В ней можно выделить:

множество используемых нечетких правил;

базу данных, содержащую описания функций принадлежности;

механизм вывода и агрегирования, который формируется применяемыми правилами импликации.

16. случае технической реализации в качестве входных и выходных сигналов выступают измеряемые величины, однозначно сопоставляющие входным значениям соответствующие выходные значения.

Для обеспечения взаимодействия этих двух видов вводится нечеткая система с так называемым фазификатором (преобразователем множеств входных данных в нечеткое множество) на входе и дефазификатором (преобразователем нечетких множеств в конкретное значение выходной переменной) на выходе.

Фазификатор преобразует точное множество входных данных в нечеткое множество, определенное с помощью функции принадлежности, а дефазификатор решает обратную задачу - формирует однозначное решение относительно входной переменной на основании многих нечетких выводов, вырабатываемых исполнительным модулем нечеткой системы.

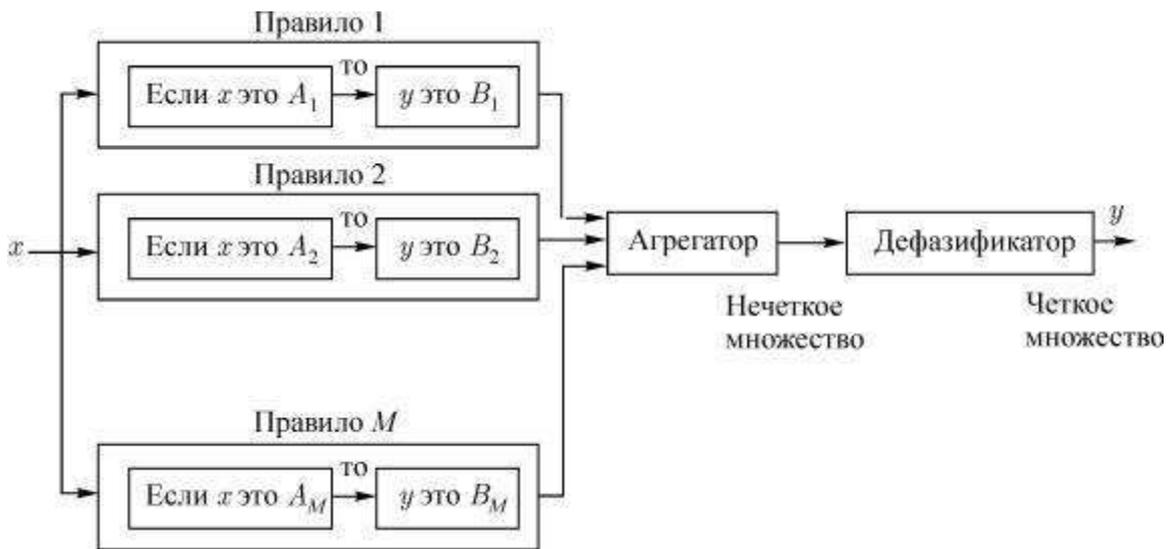


Рис. 1. Вывод в нечеткой системе при наличии M правил

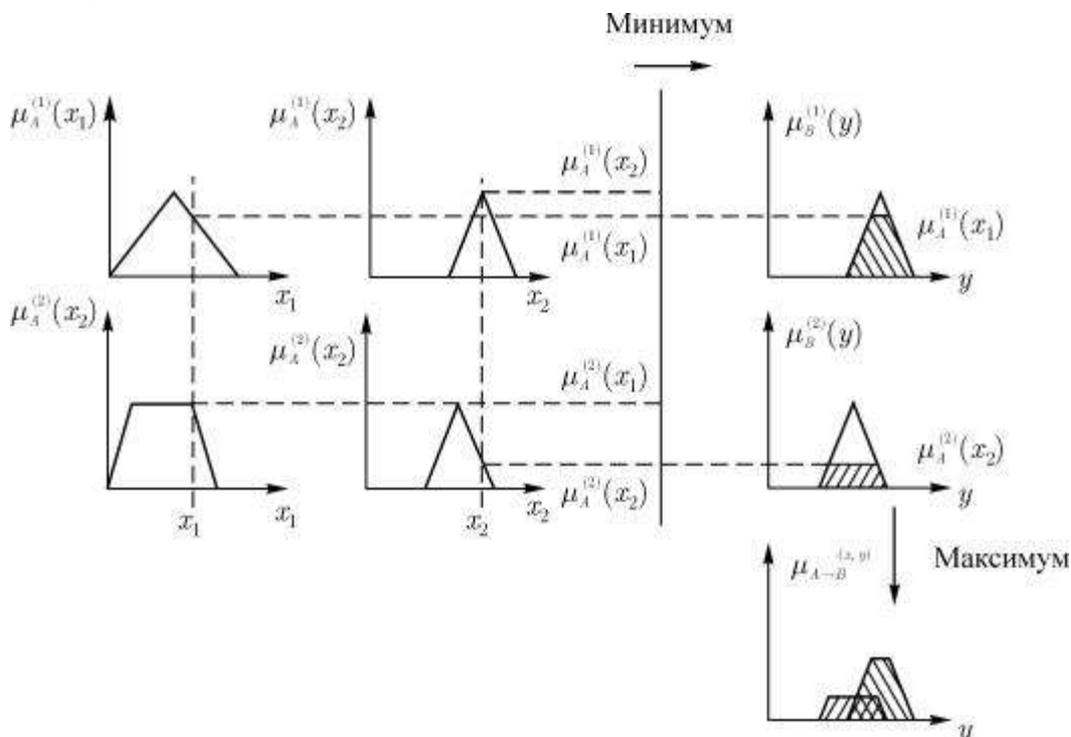
Выходной сигнал модуля вывода может иметь вид  $M$  нечетких множеств, определяющих диапазон изменения выходной переменной. Дефазификатор преобразует этот диапазон в одно конкретное значение, принимаемое в качестве выходного сигнала всей системы.

19. модели вывода Мамдани-Заде присутствуют следующие операторы:

оператор логического или арифметического произведения для определения результирующего уровня активации, в котором учитываются все компоненты вектора условия;  
оператор логического или арифметического произведения для определения значения функции принадлежности для всей импликации  $A \rightarrow B$  ;

оператор логической суммы как агрегатор равнозначных результатов импликации многих правил;

оператор дефазификации, трансформирующий нечеткий результат  $\mu(y)$  в четкое значение  $y$  .





**Рис. 2.** Пример системы вывода Мамдани-Заде

На рис. 2 представлено способ преобразования правил двух входных переменных  $x_1, x_2$ .

Логическое *произведение* (оператор  $\min$ ) используется как для агрегирования значений относительно конкретных переменных  $x_i, i = 1, 2$ , образующих вектор  $x$ , так и для импликации для одиночных правил вывода. Итерация правил импликаций, касающихся правил, проводится с использованием логической суммы (оператор логической суммы (оператор  $\max$ )).

**Фазификатор**

Фазификатор преобразует  $N$ -мерный вектор  $x = [x_1, x_2, \dots, x_N]$  в нечеткое множество  $A$ , характеризуемое функцией принадлежности  $\mu_A(x)$ .

Наибольшее применение находят функции гауссовского типа, треугольные и трапецеидальные функции:

♦7. Общая форма гауссовской функции

$$\mu_A(x) = \exp[-(x - c)^2 / \sigma^2]$$

$c$  - центр нечеткого множества,

$\sigma$  - коэффициент широты.

♦8. Симметричная треугольная функция

$$\mu_A(x) = \begin{cases} 1 - |x - c|/d, & \text{при } x \in [c - d, c + d]; \\ 0, & \text{для остальных } x \end{cases}$$

$c$  - центр,

$d$  - ширина.

♦9. Трапецеидальная функция

$$\mu_A(x) = \begin{cases} 0, & \text{при } x < g \text{ и } x > t; \\ 1, & \text{при } c - t/2 \leq x \leq c + t/2; \\ s(z - x), & \text{при } c + t/2 \leq x \leq z; \\ s(x - y), & \text{при } y \leq x \leq c - t/2; \end{cases}$$

$s$  - угол наклона.

При  $t = 0$  получаем треугольную функцию.



### Дефазификатор

Трансформировать нечеткое множество  $\mu(y) = \mu_{A \rightarrow B}(y)$  в точечное решение  $Y$  можно многими способами:

У Дефазификация относительно центра области

$$y_c = \int \mu(y) \cdot y \cdot dy / \int \mu(y) dy$$

или

$$y_c = \sum_i \mu(y_i) \cdot y_i / \sum_i \mu(y_i)$$

У Дефазификация относительно среднего центра

$$y_c = \sum_{i=1, M} \mu(c_i) \cdot c_i / \sum_{i=1, M} \mu(c_i)$$

где  $c_i$  - центр  $i$ -го нечеткого правила,

$\mu(c_i)$  - соответствующая функция принадлежности.

3. Дефазификация относительно среднего максимума

$$y_M = \sum_{i=1, m} y_i / m,$$

где  $m$  - количество точек, в которых  $\mu(y_i)$  достигает максимального значения. Если функция  $\mu(y)$  имеет максимальное значение только в одной точке, то

$$y_M = y_{max}.$$

4. выбирается минимальное из максимальных значений  $Y$ :

$y_s$  - наименьшее из  $Y$ , для которых  $\mu(y) = \max$ .



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

13. выбирается максимальное из максимальных значений:  $y_i$  - наибольшее из  $y$ , для которых  $\mu(y) = \max$ .



### Модель Мамдани-Заде как универсальный аппроксиматор

Модели нечеткого вывода позволяют описать выходной сигнал многомерного процесса как нелинейную функцию входных переменных  $x_i$ ,  $i = 1, 2, \dots, N$  и параметров нечеткой системы, например, при использовании в качестве агрегатора оператора алгебраического произведения с последующей дефазификацией относительно среднего центра. В модели Мамдани-Заде каждое из  $M$  правил определяется уровнем активации условия

$$\mu(y_i) = \prod_{j=1}^M \mu_{A_i}(x_j)$$

где  $y_i$  - значение  $y$ , при котором значение  $\mu(y_i)$  максимально. Пусть  $y_i$  — центр  $C_i$  нечеткого множества заключения  $i$ -го правила вывода. Тогда дефазификация относительно среднего центра дает

$$y = \left( \sum_{i=1}^M C_i \left[ \prod_{j=1}^N \mu_{A_i}(x_j) \right] \right) / \sum_{i=1}^M \prod_{j=1}^N \mu_{A_i}(x_j)$$

Приведенные формулы модели Мамдани-Заде имеют модульную структуру, которая идеально подходит для системного представления в виде многослойной структуры, напоминающей структуру классических нейронных сетей. Такие сети мы будем называть нечеткими нейронными сетями. Характерной их особенностью является возможность использования нечетких правил вывода для расчета выходного сигнала. Обучение таких сетей сводится к расчету параметров функции фазификации.

### Нечеткие сети TSK (Такаги-Сугено-Канга)

Схема вывода в модели TSK при использовании  $M$  правил и  $N$  переменных  $x_j$  имеет вид ( $i = 1, 2, \dots, M$ )

$$\text{if } (x_1 \text{ is } A_1^{(i)}) (x_2 \text{ is } A_2^{(i)}) \dots (x_N \text{ is } A_N^{(i)}) \\ \text{then } y_i = p_{i0} + \sum_{j=1}^N p_{ij} x_j.$$

Условие  $(x_i \text{ is } A_i)$  реализуется функцией фазификации

$$\mu_A(x_i) = 1 / (1 + ((x_i - c_i) / \sigma_i)^{2b_i}).$$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

При *M* правилах агрегированный выходной результат сети имеет вид



$$y(x) = \frac{\sum_{i=1}^M w_i y_i(x)}{\sum_{i=1}^M 1} \quad 1)$$

$$y_i(x) = p_{i0} + \sum_{j=1}^N p_{ij} x_j.$$

Веса  $w_i$  интерпретируются как *значимость* компонентов  $\mu_A^{(i)}(x)$ . Тогда формуле (1) можно поставить в соответствие многослойную нейронную сеть рис. 3.

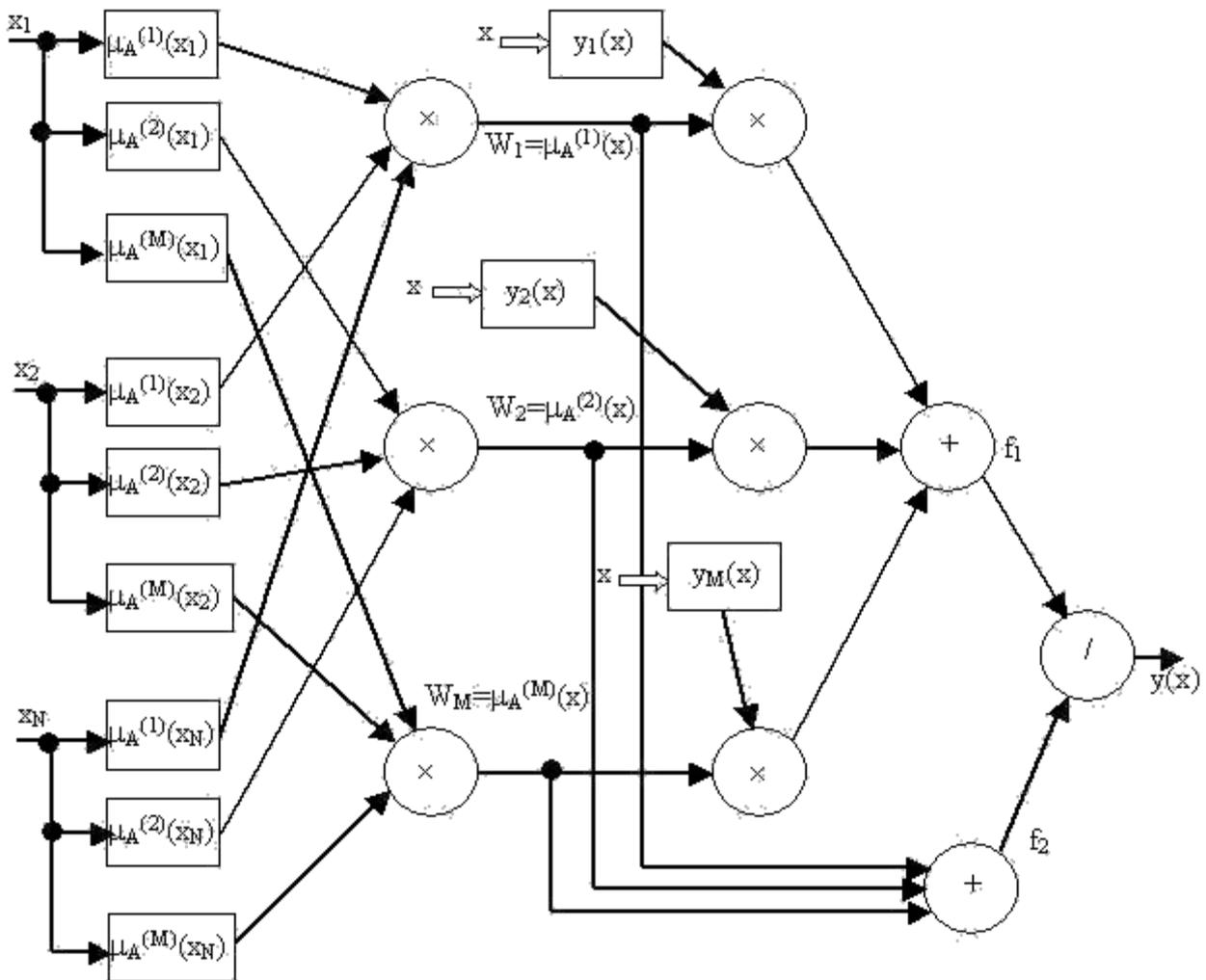


Рис. 3. Нечеткая нейронная сеть TSK

1. Первый слой выполняет фазификацию каждой переменной. Это параметрический слой с параметрами  $c_j^{(i)}, \sigma_j^{(i)}, b_j^{(i)}$ , подлежащими адаптации в процессе обучения.

2. Вторым слоем выполняется агрегирование отдельных переменных, определяющее результирующее значение коэффициента принадлежности  $w_i = \mu_A^i(x)$  для вектора  $x$  (непараметрический слой).



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

3. Третий слой - *генератор функции TSK*, рассчитывает значения



$$y_i(x) = p_{i0} + \sum_{j=1}^N p_{ij}x_j.$$

В этом слое также производится умножение  $y_i(x)$  на  $w_i$ , сформированные в предыдущем слое. Здесь адаптации подлежат веса  $p_{ij}, i = 1, 2, \dots, M, j = 1, 2, \dots, N$ , определяющие функцию следствия модели TSK.

4. Четвертый слой составляют два нейрона-сумматора, один из которых рассчитывает взвешенную сумму сигналов  $y_k(x)$  а второй - сумму весов  $w_i, i = 1, 2, \dots, M$  (непараметрический слой).

Пятый слой из одного нейрона - это нормализующий слой, в котором выходной сигнал сети агрегируется по формуле (1).

Таким образом, в процессе обучения происходит уточнение параметров только первого (нелинейного) и третьего (линейного) слоев.

#### ***Гибридный алгоритм обучения нечетких сетей***

Параметры, подлежащие адаптации, разделяются на две группы:

- первая состоит из параметров  $p_{ij}$  линейного третьего слоя;
- вторая состоит из параметров нелинейной функции принадлежности первого слоя.

Уточнение параметров проводится в два этапа.

На первом этапе при фиксации определенных значений параметров функции принадлежности путем решения системы линейных уравнений рассчитываются параметры  $p_{ij}$  полинома TSK.

При известных значениях функции принадлежности преобразование, реализуемое сетью, можно представить в виде

$$y(x) = \sum_{i=1}^M w_i (p_{i0} + \sum_{j=1}^N p_{ij}x_j).$$
$$w_i = \frac{[\prod_{j=1}^N \mu_A^{(i)}(x_j)]}{\sum_{k=1}^M [\prod_{j=1}^N \mu_A^{(k)}(x_j)]} = const.$$



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

При  $P$  обучающих выборках  $(x^{(l)}, d^{(l)})$ ,  $l = 1, 2, \dots, P$  и замене выходного сигнала сети *ожидаемым значением*  $d^{(l)}$  получим систему из  $P$  линейных уравнений вида

$$W \cdot P = d,$$



где

$$W = \begin{pmatrix} w'_{11} & w'_{11}x_1^{(1)} & \dots & w'_{11}x_N^{(1)} & \dots & w'_{1M} & w'_{1M}x_1^{(1)} & \dots & w'_{1M}x_N^{(1)} \\ w'_{21} & w'_{21}x_1^{(2)} & \dots & w'_{21}x_N^{(2)} & \dots & w'_{2M} & w'_{2M}x_1^{(2)} & \dots & w'_{2M}x_N^{(2)} \\ \dots & \dots \\ w'_{p1} & w'_{p1}x_1^{(p)} & \dots & w'_{p1}x_N^{(p)} & \dots & w'_{pM} & w'_{pM}x_1^{(p)} & \dots & w'_{pM}x_N^{(p)} \end{pmatrix}$$
$$P = \| \| p_{10} \dots p_{1N} \dots p_{M0} \dots p_{MN} \| \| ^T,$$

$w'_{ki}$  - уровень активации (вес)  $i$ -го правила при предъявлении  $k$ -го входного вектора  $x^{(k)}$ .

Размерность матрицы  $W$  равна  $p \times (N + 1)M$ , при этом обычно количество строк (количество выборок) значительно больше количества столбцов. Решение этой системы уравнений можно получить за один шаг при помощи псевдоинверсии матрицы  $W$ :

$$P = W^+d.$$

Псевдоинверсия матрицы заключается в решении задачи минимизации

$$\min \| W^+W - E \|,$$

где  $E$  - единичная матрица.

На втором этапе (линейные параметры  $p_{ij}, i = 1, \dots, M$  - фиксированы) рассчитываются фактические выходные сигналы  $y_k, k = 1, 2, \dots, p$ :

$$y = Wp,$$

вектор ошибки

$$\varepsilon = y - d,$$

градиент целевой функции  $E(n)$  по параметрам первого слоя. Если применяется метод наискорейшего спуска, то формулы адаптации принимают вид

$$c_j^{(i)}(n+1) = c_j^{(i)}(n) - \alpha_c \partial E(n) / \partial c_j^{(i)}$$
$$\sigma_j^{(i)}(n+1) = \sigma_j^{(i)}(n) - \alpha_\sigma \partial E(n) / \partial \sigma_j^{(i)}$$
$$b_j^{(i)}(n+1) = b_j^{(i)}(n) - \alpha_b \partial E(n) / \partial b_j^{(i)}$$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

где  $n$  обозначает номер очередной итерации.



После уточнения нелинейных параметров вновь запускается процесс адаптации линейных параметров TSK (первый этап) и нелинейных параметров (второй этап). Этот цикл повторяется вплоть до стабилизации всех параметров процесса.

### ***Мягкая экспертная система***

Рассмотрим архитектуру и основные структурно-функциональные решения мягкой экспертной системы (МЭС). Для определения МЭС сопоставим понятия нечеткой и мягкой экспертных систем. В описании архитектуры МЭС будем использовать три признака: способ *извлечения знаний*; *представление знаний*; *обработку знаний*. Перечисленные признаки создают общую "координатную" сетку описания.

### ***Определение мягкой экспертной системы. Сравнение нечеткой и мягкой экспертных систем***

*Нечеткие экспертные системы (ЭС)* используют *представление* знаний в форме нечетких продукций и *лингвистических переменных*. Основу представления *лингвистической переменной* составляет *терм* с функцией принадлежности. Способ обработки знаний в нечетких ЭС - это *логический вывод* по нечетким продукциям. Особенностью нечеткой ЭС является способ извлечения функций принадлежности, который сводится либо к статистическим методам построения, либо к методу экспертных оценок. Мягкой ЭС (МЭС) будем называть нечеткую ЭС, которая обладает следующими особенностями:

использует статистические данные, которые интерпретирует как обучающие выборки для нечетких нейронных сетей;

представляет знания в виде *лингвистических переменных* (функций принадлежности В ФП), нечетких продукций и обученных нейронных сетей. Редукция множества нечетких продукций, настройка ФП и базы правил выполняется с помощью генетических алгоритмов (ГА).

Мягкими называют вычисления, сочетающие теорию нечетких систем, нейронные сети, вероятностные рассуждения и *генетические алгоритмы*, и обладающие синергическим эффектом; следовательно, мягкой экспертной системой называют ЭС, сочетающую перечисленные теории ради того же эффекта взаимного усиления.

Рассмотрим возможные применения МЭС в автоматизированном проектировании. *Обобщенной моделью проектирования* является иерархически-блочный метод, сущность которого сводится к декомпозиции функций с последующим выделением иерархий систем и подсистем. Проектируемая система формируется с помощью синтеза таких подсистем. *Анализ* в ходе автоматизированного проектирования обычно заключается в том, что необходимо рассмотреть условия эксплуатации будущей системы или ее окружения, которое является сложной системой (например, для экономических информационных систем окружающая среда - это социально-экономическая среда). Кроме анализа окружающей среды в ходе проектирования приходится выполнять *анализ* результатов физических или численных экспериментов и имитационного моделирования. Можно выделить два основных принципа экспертной деятельности в ходе проектирования.

и Исходные данные для анализа представляются в виде качественного описания структурно-функционального решения и в виде совокупности временных рядов системных переменных окружения.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Принцип "конструктивной неопределенности" утверждает, что *точность* и смысл противоречат друг другу, начиная с некоторого момента анализа. Если в технике важными являются все более точные измерения, то в ходе анализа эксперт отказывается от точных цифр в



пользу нечетких, но содержательных оценок, которые осмыслены и позволяют принять проектное или управленческое решение.

Мягкая *экспертная система* должна предоставить инструментальную и информационную среду для экспертной деятельности в ходе проектирования. Инструменты для разработки МЭС должны представлять собой совокупность различных программных продуктов, объединенных логикой работы. Покажем, что МЭС, являющаяся инструментальной средой проектировщика, позволяет выполнить в автоматизированном режиме все этапы экспертной деятельности. Если рассматривать экспертную *деятельность* как управление объектом, то *инструментарий* экспертизы можно использовать как систему управления, а именно - нечеткий *контроллер*.

### ***Представление знаний в мягкой экспертной системе. Содержание баз знаний и данных мягкой экспертной системы***

Если использовать нечеткую НС на этапе *извлечения знаний*, то, кроме функций принадлежности и нечетких продукций, порождается совокупность обученных НС, которые входят в базу знаний МЭС. *Оптимизация (редукция) множества* извлеченных правил выполняется на основе генетического алгоритма.

*База знаний* МЭС должна содержать следующие части:

- функции принадлежности;
- нечеткие продукции;
- обученные нечеткие нейронные сети;
- процедуры интерпретации хромосом генетических алгоритмов;
- функции оптимальности.

Рассмотрим проблему представления перечисленных составных частей в компьютерных интеллектуальных системах. Если *функция* принадлежности характеризуется такими математическими свойствами, как непрерывность, выпуклость (унимодалность), то *функция* принадлежности может быть представлена параметризованной функцией формы. Наибольшее распространение получили четыре вида функций формы: треугольная, трапециевидная, колоколообразная и сигмоидальная, которые определяются тройкой, четверкой и двойкой параметров соответственно. Некоторые *операции* нечеткой алгебры сохраняют унимодалность при использовании трапециевидного представления функций принадлежности, поэтому результаты *операции* также являются четверкой параметров. *Представление* нечетких продукций упрощается в связи с тем, что порядок обработки нечетких продукций не важен и не влияет на ход вывода результата. *Представление* нечеткой нейронной сети является более сложной проблемой, так как описание структуры ННС не имеет смысла без нейроимитатора соответствующей архитектуры нечетких нейронных сетей, т.е. нейроимитатор определяется как составляющая часть механизма вывода мягкой ЭС. Для организации хранения знаний МЭС можно использовать как *СУБД*, так и специальные форматы.

### **Контрольные вопросы ЛР8(ПК-4):**

- Интеллектуальные информационные системы в условиях неопределенности и риска.
- Нечеткие множества.
- Лингвистические переменные.
- Нечеткие правила вывода.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Системы нечеткого вывода Мамдани-Заде.
- Фазификатор. Дефазификатор.



12. Модель Мамдани-Заде как универсальный аппроксиматор.
13. Распознавание рукописных текстов.
14. Состояние и тенденции развития искусственного интеллекта.
15. Успехи систем искусственного интеллекта и их причины
16. Моделирование бизнес-прецедентов.
17. Разработка модели бизнес-объектов.
18. Разработка концептуальной модели данных.
19. Разработка требований к системе.
20. Анализ требований и предварительное проектирование системы
21. Гибридный алгоритм обучения нечетких сетей.
22. Мягкая экспертная система.
23. Определение мягкой экспертной системы.
24. Сравнение нечеткой и мягкой экспертных систем.
25. Представление знаний в мягкой экспертной системе.
26. Содержание баз знаний и данных мягкой экспертной системы.
27. Синтезатор русской речи.
28. Язык формальной записи правил синтеза.
29. Интонационное обеспечение.
30. Аллофонная база данных. Лингвистический анализ.
31. Инструментарий синтеза русской речи.
32. Система распознавания речи.
33. Акустическая и лингвистическая модели.
34. Классификация систем распознавания. Системы машинного зрения.
35. Основные принципы или целостность восприятия.
36. Распознавание символов.
37. Состояние и тенденции развития искусственного интеллекта.
38. Успехи систем искусственного интеллекта и их причины

# Системы принятия решений

## Практическое занятие № 1. Применение классических критериев принятия решений.

*Решение задач. Построение матрицы решений. Выбор вариантов. Построение оценочной функции. Построение матрицы остатков. Поиск наивыгоднейшего распределения вероятностей на множестве вариантов*

### Теоретические сведения

Критерий принятия решений - это функция, выражающая предпочтения лица принимающего решения (ЛПР) и определяющая правило, по которому выбирается приемлемый или оптимальный вариант решения.

Задача принятия решений возникает тогда, когда возникает несколько конкурирующих вариантов решения. В противном случае ситуация предопределена. Варианты решений возникают в результате анализа проблемной ситуации, представленной в виде описательной модели. В классическом случае описание ситуации дается в виде матрицы, строки которой соответствуют вариантам решений, а столбцы - факторам, которые могут повлиять на результат, получаемый ЛПР. На пересечении столбцов и строк расположены либо проигрыши,



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

соответствующие реализациям решений  $E_i$  в соответствующих условиях  $F_j$ , либо, наоборот, выигрыши.

Рассмотрим простейший случай одностолбцовой матрицы.

Предположим, что у нас имеются варианты решений  $E_1, E_2, \dots, E_n$ , которые характеризуются некоторым результатом  $e_i$ .

Такой результат можно интерпретировать как выигрыш, полезность, надежность. Нам

необходимо найти  $\max_i (e_i)$ .

Таким образом, выбор оптимального варианта производится с помощью критерия

$$E_o = E_i \mid E_i \quad E_o \& e = \max_i (e_i) \quad (1)$$

Это правило выбора читается следующим образом: множество  $E_o$  оптимальных вариантов состоит из тех вариантов  $E_i$ , которые принадлежат множеству  $E$  всех вариантов и оценка  $e_i$  которых максимальна среди всех оценок  $\{e_i\}$ . (Логический знак  $\&$  читается как "и" и требует, чтобы оба связываемых им утверждения были истинны.)

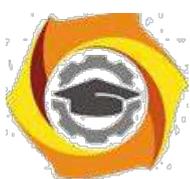
Такая постановка задачи, как было сказано выше, соответствует простому случаю.

ее более сложных структурах каждому допустимому варианту решения  $E_i$  по многим причинам могут соответствовать различные внешние условия (состояния)  $F_j$  и, как следствие, различные результаты  $e_{ij}$  реализации решений.

Под результатом решения  $e_{ij}$  здесь будем понимать численную оценку, соответствующую варианту  $E_i$  и условиям  $F_j$  и характеризующую экономический эффект (прибыль), полезность или потребительную стоимость. Будем называть такой результат эффективностью решения.

Таким образом, ситуация ПР описывается некоторой матрицей (Таблица 1). Размерность этой матрицы зависит от множества вариантов решений и множества рассматриваемых факторов или условий, влияющих на принятие решений.

Таблица 1 - Матрица решений  $e_{ij}$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

	F <sub>1</sub>	F <sub>2</sub>	...	F <sub>m</sub>
E <sub>1</sub>	<i>e</i> <sub>11</sub>	<i>e</i> <sub>12</sub>	...	<i>e</i> <sub>1m</sub>
E <sub>2</sub>	<i>e</i> <sub>21</sub>	<i>e</i> <sub>22</sub>	...	<i>e</i> <sub>2m</sub>
...	...	...	...	...
E <sub>n</sub>	<i>e</i> <sub>n1</sub>	<i>e</i> <sub>n2</sub>	...	<i>e</i> <sub>nm</sub>

В данном случае, так же как и в простейшем, описанном выше, ЛПР старается выбрать решение с наилучшим результатом. Однако, поскольку ему неизвестно, с какими условиями он столкнётся, он вынужден принимать во внимание все численные оценки  $e_{ij}$ ,

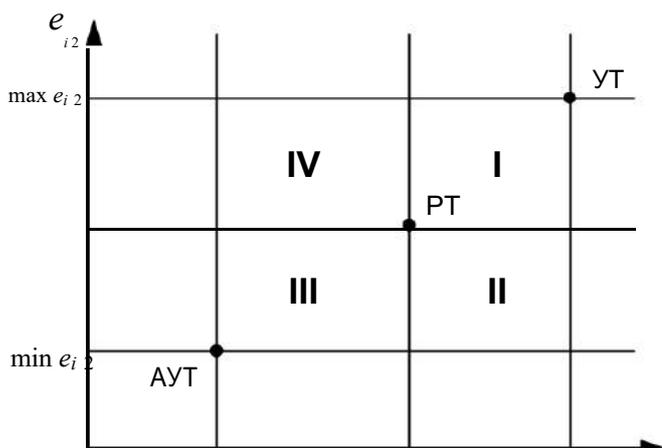
$$\max(e_i)$$

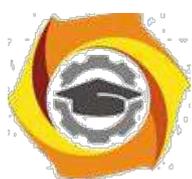
соответствующие варианту  $E_i$ . Первоначальная задача максимизации  $i$  согласно критерию (1) должна быть теперь заменена другой, подходящим образом учитывающей все последствия любого из вариантов решения  $E_i$ .

Для того, чтобы получить более ясную и наглядную интерпретацию перейдём к графическому представлению оценочных функций. Случай с двумя внешними условиями ( $m=2$ ) при  $n$  вариантах решений будет простейшим. Результат такого рассмотрения можно распространить на случай большего количества внешних факторов (условий).

Введём прямоугольную систему координат, откладывая по оси абсцисс значения результата  $e_{i1}$  решения  $E_i$ , соответствующего внешнему состоянию  $F_1$ , а по оси ординат - значения  $e_{i2}$ , соответствующего состоянию  $F_2$ ,  $i=1, \dots, n$ . При таких обозначениях каждый вариант решения  $E_i$  соответствует точке на плоскости с координатами  $(e_{i1}, e_{i2})$ ,  $i=1, \dots, n$  соответственно. Все точки образуют множество, которое можно вписать в прямоугольник со сторонами, параллельными осям координат, расположение которых соответствует максимальным и минимальным значениям среди всех элементов матрицы. Точку с координатами  $(\max(e_{i1}, i), \max(e_{i2}, i))$ , соответствующей верхнему правому углу, мы назовем утопической точкой (УТ). Смысл этого названия в том, что координаты всех точек  $(e_{i1}, e_{i2})$ ,  $i=1, \dots, n$ , соответствующих вариантам решений  $E_1, \dots, E_n$ , не могут быть больше, чем у точки УТ.

Аналогичное значение имеет и так называемая антиутопическая точка (АУТ), имеющая координаты  $(\min(e_{i1}, i), \min(e_{i2}, i))$ , соответствующая нижнему левому углу. Координаты всех точек  $(e_{i1}, e_{i2})$ ,  $i=1, \dots, n$ , соответствующих вариантам решений  $E_1, \dots, E_n$ , не могут быть меньше, чем у точки АУТ. Построенный прямоугольник называется полем полезности решений (рисунок 1).





**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

$\min e_{i1}$        $\max e_{i1}$        $e_{i1}$



Рис. 1 - Поле полезности решений

Рассмотрим это поле подробнее. Выберем произвольную точку на плоскости и назовем ее рабочей точкой (РТ) -  $E_{PT}$ . Обозначим ее координаты  $(e_{PT1}, e_{PT2})$ . С помощью прямых, параллельных координатным осям, разобьем плоскость на четыре части и обозначим их I, II, III и IV. В рассматриваемом нами двумерном случае каждая из этих частей имеет вид квадранта; в случае произвольной размерности они превращаются в так называемые конусы.

Все точки  $E_i$  из матрицы вариантов решений, лежащие в конусе I заведомо или гарантировано лучше, чем рассматриваемая точка РТ. Это преимущество решений из конуса I по отношению к решению, находящемуся в РТ не зависит от того, какой фактор -  $F_1$  или  $F_2$  реализуется, то есть не важно по какой координате это преимущество реализуется. Поэтому мы называем конус I конусом предпочтения.

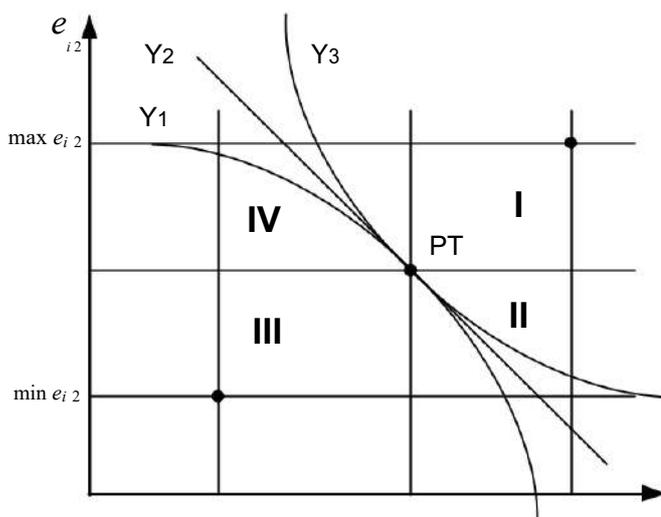
Соответственно все точки конуса III хуже точки РТ ( $e_{PT1} \geq e_{i1}$  и  $e_{PT2} \geq e_{i2}$ ), и мы будем называть область III антиконусом. Таким образом, оценка качества точек из этих двух конусов в сравнении с точкой РТ проста и однозначна.

Оценка же точек в отмеченных штриховкой конусах II и IV является неопределенной, так как соотношения их координат с РТ является противоречивым. Вследствие этого их называют областями неопределенности и варианты решений в этих конусах связаны с допущением некоторого риска принятия решений.

Таким образом, критериальные функции, лежащие внутри конуса I обеспечивают гарантированный результат и могут считаться безрисковыми (пессимистическими) критериями ПР. Линии, проходящие через II и IV квадранты, соответствуют критериям с риском.

И, наконец, линии, лежащие внутри III конуса соответствуют критерию азартного игрока или оптимистичной позиции. Рассмотрим линии, соответствующие трем группам критериев (рисунок 2).

Все точки из областей неопределенности, лежащие справа и выше каждой линии функций предпочтения лучше точек, лежащих слева и ниже. Всякая функция (линия) предпочтения объединяет все точки, соответствующие одному и тому же значению критерия (точки равной эффективности); справа и выше ее располагаются все лучшие точки, то есть точки более эффективных решений, а слева и ниже - худшие, то есть точки менее эффективных решений.





**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

$\min e_{i1}$

$\max e_{i1}$

$e_{i1}$



Рис. 2 - Функции предпочтения при принятии решений

$Y_1$  – азартного игрока (оптимистические);  $Y_2$  – критериев с риском (нейтральные);  $Y_3$  – безрисковые (пессимистические).

Таким образом, критериальная функция делит плоскость на две части. В соответствии с рассмотренным полем полезности, критериальные функции, проходящие ближе к границам I квадранта соответствуют безрисковой политике принятия решения или тенденции избегания риска, как например, вогнутая штриховая линия на рисунке 2. Линии, проходящие через квадрант III, соответствуют азартной стратегии с максимальным риском. Соответственно, прямая линия, проходящая через рабочую точку и квадранты II и IV, соответствует нейтральной или объективной стратегии в ПР.

Если выбор оценочной функции отдается на усмотрение ЛПР, то приходится считаться с возможностью различных результатов для одного и того же решения. Таким образом, принятие решения не есть чисто рациональный процесс. Опасность возникает в тех случаях, когда критериальные оценочные функции выбираются интуитивно, иногда даже без выяснения исходной позиции лица принимающего решения.

Всякое техническое или экономическое решение в условиях неполной информации - сознательно или неосознанно - принимается в соответствии с какой-либо оценочной функцией описанного выше типа. Как только это бывает признано явно, следствия соответствующих решений становятся лучше обозримыми, что позволяет улучшить их качество. При этом выбор оценочных функций всегда должен осуществляться с учетом количественных характеристик ситуации, в которой принимаются решения.

Следующие критерии относят к классическим:

$$Z_{MM} = \max_i \min_j (e_{ij})$$

– минимаксный

$$Z_{BL} = \max_{i,j=1}^m e_{ij}$$

– Байеса-Лапласа

$$Z_S = \min_i \max_j (\max_i (e_{ij}) - e_{ij})$$

– Сэвиджа

$$Z_{ME} = \max_{i=1, j=1}^{n, m} p$$

– расширенный минимаксный

$$Z_{AG} = \max_i \max_j (e_{ij})$$

– азартного игрока

## Минимаксный критерий принятия решений

### Аналитический метод расчета

Математическая интерпретация критерия выглядит следующим образом:



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

$$Z_{\max} \min_{j \in I} e_j \quad (29)$$

Процесс нахождения оптимального решения рассмотрим на примере.



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

8

E7	5
E8	4
E9	5
E10	6

Заданная матрица решений:

	F1	F2
E1	1	1
E2	3	3
E3	7	1
E4	2	2
E5	3	1
E6	4	4
E7	5	1
E8	4	2
E9	5	3
E10	6	2

Шаг 1.

Выбираем минимальное значение в каждой строке.

	F1	F2	$\min_j (e_{ij})$
E1	1	1	1
E2	3	3	3
E3	7	1	1
E4	2	2	2
E5	3	1	1
E6	4	4	4
E7	5	1	1
E8	4	2	2
E9	5	3	3
E10	6	2	2

Шаг 2.

Выбираем максимальное значение в добавленном столбце (

	F1	F2	$\min_j (e_{ij})$
E1	1	1	1
E2	3	3	3
E3	7	1	1
E4	2	2	2
E5	3	1	1
<b>E6</b>	<b>4</b>	<b>4</b>	<b>4</b>



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

$j$  )

$\min (e_{ij} )$





Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

	$F_1$	$F_2$	$e_{1i} q_j$	$e_{2i} q_j$	$e_{ij} q_j$ $j=1$
$E_1$	1	1	0,5	0,5	1



E2	3	3	1,5	1,5	3
<b>E3</b>	<b>7</b>	<b>1</b>	<b>3,5</b>	<b>0,5</b>	<b>4</b>
E4	2	2	1	1	2
E5	3	1	1,5	0,5	2
<b>E6</b>	<b>4</b>	<b>4</b>	<b>2</b>	<b>2</b>	<b>4</b>
E7	5	1	2,5	0,5	3
E8	4	2	2	1	3
<b>E9</b>	<b>5</b>	<b>3</b>	<b>2,5</b>	<b>1,5</b>	<b>4</b>
<b>E10</b>	<b>6</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>4</b>
					<b>4</b>

Шаг 3.

$$e_{ij}^m q_j$$

Находим максимальное значение в добавленном столбце  $j = 1$ .

$$e_{ij}^m q_j$$

Соответственно оптимальными решениями являются все решения, значения  $j = 1$  которых равны 4. В данном случае имеем четыре решения – E3, E6, E9, E10.

### Критерий Сэвиджа

#### Аналитический метод расчета

Математическая интерпретация критерия выглядит следующим образом:

$$Z_S = \min_i \max_j (\max_i (e_{ij}) - e_{ij})$$

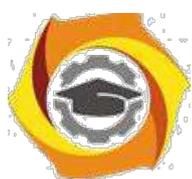
Процесс нахождения оптимального решения рассмотрим на примере.

Заданная матрица решений:

	F1	F2
E1	1	1
E2	3	3
E3	7	1
E4	2	2
E5	3	1
E6	4	4
E7	5	1
E8	4	2
E9	5	3
E10	6	2

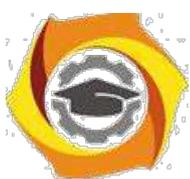
Шаг 1. Находим максимальное значение в каждом из столбцов

	F1	F2
--	----	----



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

E <sub>1</sub>	1	1
E <sub>2</sub>	3	3
E <sub>3</sub>	7	1



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

11

E4	2	2
E5	3	1
E6	4	4
E7	5	1
E8	4	2
E9	5	3
E10	6	2
	7	4

Шаг 2.

Каждый элемент матрицы решений вычитается из наибольшего результата соответствующего столбца. Эти разности образуют матрицу остатков.

	F1	F2
E1	6	3
E2	4	1
E3	0	3
E4	5	2
E5	4	3
E6	3	0
E7	2	3
E8	3	2
E9	2	1
E10	1	2
	7	4

Шаг 3.

Эта матрица пополняется столбцом наибольших разностей  $e_{ir} = \max_j \left( \max_i (e_{ij}) - e_{ij} \right)$

	F1	F2	$e_{ir}$
E1	6	3	6
E2	4	1	4
E3	0	3	3
E4	5	2	5
E5	4	3	4
E6	3	0	3
E7	2	3	3
E8	3	2	3
E9	2	1	2
E10	1	2	2
			2

Выбираются те решения  $E_i$ , в строках которых стоит наименьшее значение для этого столбца. Соответственно оптимальными решениями являются все решения, значения  $e_{ir}$  которых равны 2. В данном случае имеем два решения – E9 и E10.

**ЗАДАНИЕ**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Составьте и отладьте программу на одном из языков программирования (Lazarus, C, VBA for Excel, Java...), которая построение конуса предпочтения для решения по критерию,



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

12

соответствующему Вашему варианту. Номер варианта определяется как остаток от деления вашего номера по журналу на 5.

Вариант 1 - минимаксный  $Z_{MM} = \max_i \min_j (e_{ij})$  ;

Вариант 2 - Байеса-Лапласа  $Z_{BL} = \max_i \sum_{j=1}^m e_{ij} q_j$  ;

Вариант 3 - Сэвиджа  $Z_S = \min_i \max_j (\max_i (e_{ij}) - e_{ij})$  ;

Вариант 4 - расширенный минимаксный  $Z_{ME} = \max_p \min_{i,j} \sum_{i=1}^n \sum_{j=1}^m e_{ij} p_i q_j$  ;

Вариант 5 - азартного игрока  $Z_{AG} = \max_i \max_j (e_{ij})$  .

Исходные данные – матрица решений должна вводиться из текстового файла в кодировке UTF-8 следующей структуры, разделитель – табуляция, число исходов и решений не ограничено (пример):

	F1	F2
E1	1	1
E2	3	3
E3	7	1
E4	2	2
E5	3	1
E6	4	4
E7	5	1
E8	4	2
E9	5	3
E10	6	2

Программа должна включать диалог выбора файла исходных данных, запрос вероятностей и выдачу сообщения о выбранном варианте и возможных результатах.

### Контрольные вопросы

- В Опишите классический процесс принятия решения.
- В Что соответствует каждому варианту решения в классической схеме?
- В Запишите критерий выбора оптимального варианта решения.
- В Для чего предназначена оценочная функция?
- В Дайте определени матрице решений.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

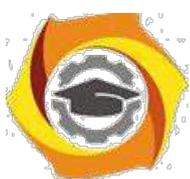
- В Что такое поле выбора решений?
- В Что такое конус предпочтения?
- В Какую позицию характеризует вогнутая кривая предпочтения?



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

13

35. Опишите минимаксный критерий принятия решения.
36. Опишите критерий Байеса – Лапласа.
37. Опишите критерий Сэвиджа.
38. Опишите расширенный минимаксный критерий.



## Практическое занятие № 2 Производные критерии принятия решений

Решение задач с использованием критериев Гурвица. Ходжа-Лемана. Гермейера. ВЛ (ММ), произведений.

### Теоретические сведения

Вводная часть находится в теоретических сведениях к первому практическому занятию.

### Критерий принятия решений Гурвица

Математическая интерпретация

$$Z_{HW} = \max_i (c \min_j (e_{ij}) + (1 - c) \max_j (e_{ij}))$$

### Аналитический метод расчета

Дана матрица решений:

	F1	F2
E1	1	1
E2	3	3
E3	7	1
E4	2	2
E5	3	1
E6	4	4
E7	5	1
E8	4	2
E9	5	3
E10	6	2

Шаг 1. Выбираем минимальное и максимальное значение в каждой строке.

	F1	F2	$\min_j (e_{ij})$	$\max_j (e_{ij})$
E1	1	1	1	1
E2	3	3	3	3
E3	7	1	1	7
E4	2	2	2	2
E5	3	1	1	3
E6	4	4	4	4
E7	5	1	1	5
E8	4	2	2	4
E9	5	3	3	5
E10	6	2	2	6

Шаг 2. Выбираем значение коэффициента  $c$ . В технических приложениях правильно выбрать множитель  $c$  бывает так же трудно, как правильно выбрать критерий. Вряд ли возможно найти количественную характеристику для тех долей оптимизма и пессимизма, которые присутствуют при принятии решения. Поэтому чаще всего весовой множитель  $c=0,5$



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

без возражений принимается в качестве некоторой «средней» точки зрения. Для нашего примера примем  $c=0.4$

Шаг 3. Домножаем  $\min_j (e_{ij})$  на  $c$ ,  $\max_j (e_{ij})$  на  $(1-c)$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

15

	F1	F2	$\min_j (e_{ij})$	$\max_j (e_{ij})$	$c \min_j (e_{ij})$	$(1 - c) \max_j (e_{ij})$
E1	1	1	1	1	0,4	0,6
E2	3	3	3	3	1,2	1,8
E3	7	1	1	7	0,4	4,2
E4	2	2	2	2	0,8	1,2
E5	3	1	1	3	0,4	1,8
E6	4	4	4	4	1,6	2,4
E7	5	1	1	5	0,4	3
E8	4	2	2	4	0,8	2,4
E9	5	3	3	5	1,2	3
E10	6	2	2	6	0,8	3,6

Шаг 4. Находим сумму  $c \min_j (e_{ij}) + (1 - c) \max_j (e_{ij})$

	F1	F2	$\min_j (e_{ij})$	$\max_j (e_{ij})$	$c \min_j (e_{ij})$	$(1 - c) \max_j (e_{ij})$	$c \min_j (e_{ij}) + (1 - c) \max_j (e_{ij})$
E1	1	1	1	1	0,4	0,6	1
E2	3	3	3	3	1,2	1,8	3
<b>E3</b>	<b>7</b>	<b>1</b>	<b>1</b>	<b>7</b>	<b>0,4</b>	<b>4,2</b>	<b>4,8</b>
E4	2	2	2	2	0,8	1,2	2
E5	3	1	1	3	0,4	1,8	2,2
E6	4	4	4	4	1,6	2,4	4
E7	5	1	1	5	0,4	3	3,4
E8	4	2	2	4	0,8	2,4	3,2
E9	5	3	3	5	1,2	3	4,2
E10	6	2	2	6	0,8	3,6	4,4
							<b>4,8</b>

Шаг 5.

Находим максимум из столбца  $c \min_j (e_{ij}) + (1 - c) \max_j (e_{ij})$ . Оптимальное решение: Все

решения для которых  $c \min_j (e_{ij}) + (1 - c) \max_j (e_{ij})$  максимально. В нашем случае это E3

### Геометрический метод расчета

Геометрическое решение можно найти используя преобразованную плоскость решений.

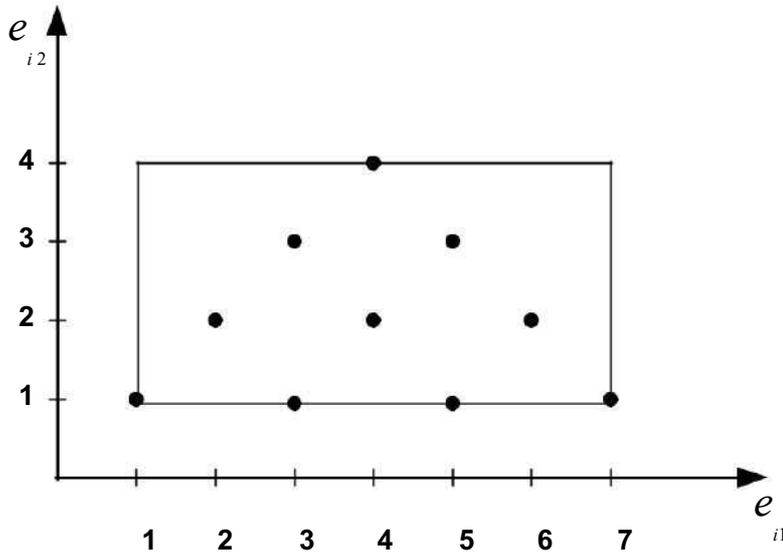


Рис.6 – Исходная плоскость множества решений  
 Решение на плоскости ищется следующим образом:  
 Шаг 1.

Строится новая плоскость решений, где осями будут не  $e_{i1}$  и  $e_{i2}$ , а  $c \min_j(e_{ij})$  и  $(1 - c) \max_j(e_{ij})$ . На этой плоскости строятся точки соответствующие решениями.

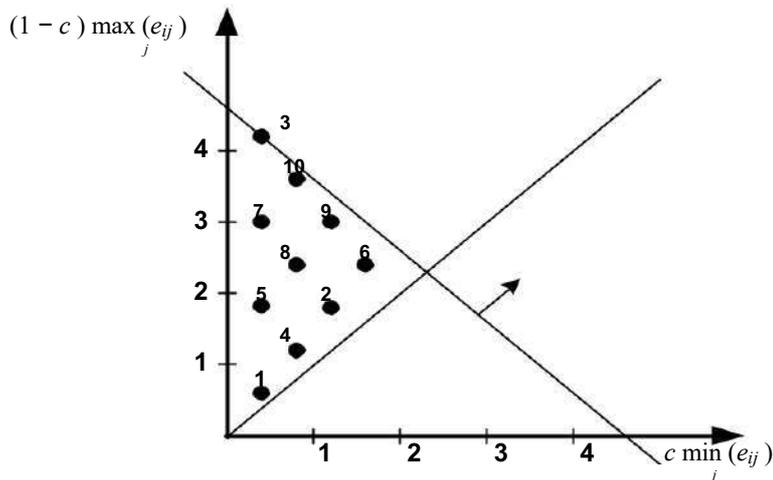


Рис.7 – Преобразованная плоскость решений

Шаг 2.

Строится направляющая – линия проведенная из начала координат под углом 45

Шаг 3.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Линия, соответствующая критерию прямой движется вдоль направляющей от начала координат до касания последней точки, которая и будет решением. В данном случае точка с координатами  $(0,4; 4,2)$ , т.е. ЕЗ



## Критерий принятия решений Гермейера

Математическая интерпретация

$$Z_G = \max_i (\min_j (e_{ij} q_j))$$

$q_j$  - вероятность условия  $F_j$ . Если о задаче ничего не известно, то принимаем  $q_1 = q_2 = 0,5$

### Аналитический метод расчета

Дана матрица решений:

	F <sub>1</sub>	F <sub>2</sub>
E <sub>1</sub>	1	1
E <sub>2</sub>	3	3
E <sub>3</sub>	7	1
E <sub>4</sub>	2	2
E <sub>5</sub>	3	1
E <sub>6</sub>	4	4
E <sub>7</sub>	5	1
E <sub>8</sub>	4	2
E <sub>9</sub>	5	3
E <sub>10</sub>	6	2

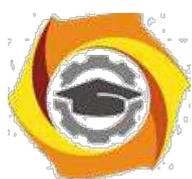
$q_1 = 0,4$  и  $q_2 = 0,6$

Шаг 1. Умножаем каждое  $e_{ij}$  на соответствующий ему  $q_j$ , т.е.

	F <sub>1</sub>	F <sub>2</sub>	$q_1 e_{i1}$	$q_2 e_{i2}$
E <sub>1</sub>	1	1	0,4	0,6
E <sub>2</sub>	3	3	1,2	1,8
E <sub>3</sub>	7	1	2,8	0,6
E <sub>4</sub>	2	2	0,8	1,2
E <sub>5</sub>	3	1	1,2	0,6
E <sub>6</sub>	4	4	1,6	2,4
E <sub>7</sub>	5	1	2	0,6
E <sub>8</sub>	4	2	1,6	1,2
E <sub>9</sub>	5	3	2	1,8
E <sub>10</sub>	6	2	2,4	1,2

Шаг 2. Выбираем минимум в каждой строке.

	F <sub>1</sub>	F <sub>2</sub>	$q_1 e_{i1}$	$q_2 e_{i2}$	$\min_j (e_{ij} q_j)$
E <sub>1</sub>	1	1	0,4	0,6	0,4
E <sub>2</sub>	3	3	1,2	1,8	1,2
E <sub>3</sub>	7	1	2,8	0,6	0,6
E <sub>4</sub>	2	2	0,8	1,2	0,8
E <sub>5</sub>	3	1	1,2	0,6	0,6
E <sub>6</sub>	4	4	1,6	2,4	1,6
E <sub>7</sub>	5	1	2	0,6	0,6
E <sub>8</sub>	4	2	1,6	1,2	1,2
E <sub>9</sub>	<b>5</b>	<b>3</b>	<b>2</b>	<b>1,8</b>	<b>1,8</b>
E <sub>10</sub>	6	2	2,4	1,2	1,2



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

E <sub>10</sub>	6	2	2,4	1,2	1,2
					<b>1,8</b>

Шаг 3.



Находим максимум в последнем столбце. Оптимальное решение: Все решения для которых  $\min_j (e_{ij} q_j)$  максимально. В нашем случае это Е<sub>9</sub>.

### Геометрический метод расчета

Геометрическое решение можно найти используя преобразованную плоскость решений.

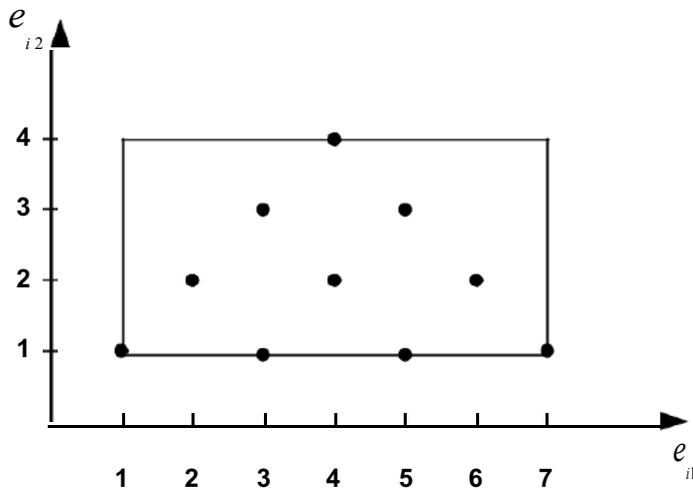


Рис.8 – Исходная плоскость множества решений

Решение на плоскости ищется следующим образом:

Шаг 1.

Строится новая плоскость решений, где осями будут не  $e_{i1}$  и  $e_{i2}$ , а  $q_1 e_{i1}$  и  $q_2 e_{i2}$ . На этой плоскости строятся точки соответствующие решениями.

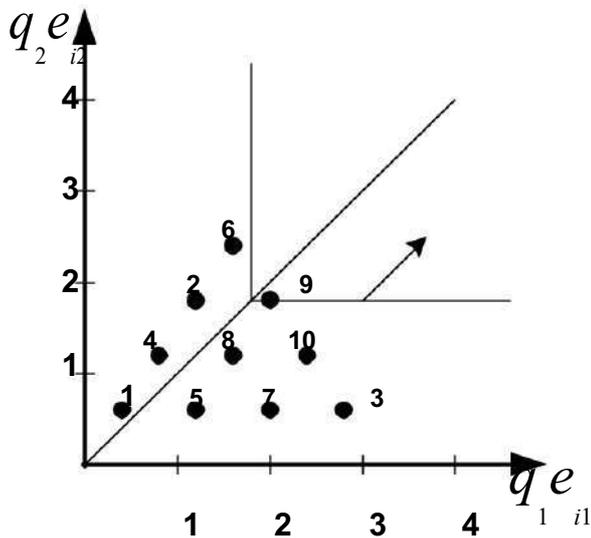


Рис.9 – Новая плоскость множества решений



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Шаг 2.

Строится направляющая – линия проведенная из начала координат под углом 45



Шаг 3.

Линия, соответствующая критерию (прямой угол) движется вдоль направляющей от начала координат до касания последней точки, которая и будет решением. В данном случае точка с координатами (2; 1,2), т.е. E9

## Критерий произведений

Математическая интерпретация

$$Z_p = \max_j \prod_{i=1}^m e_{ij}$$

### Аналитический метод расчета

Дана матрица решений:

	F1	F2
E1	1	1
E2	3	3
E3	7	1
E4	2	2
E5	3	1
E6	4	4
E7	5	1
E8	4	2
E9	5	3
E10	6	2

Шаг 1. Находим произведение  $e_{i1}$  и  $e_{i2}$

	F1	F2	$e_{i1}$	$e_{i2}$
E1	1	1	1	
E2	3	3	9	
E3	7	1	7	
E4	2	2	4	
E5	3	1	3	
<b>E6</b>	<b>4</b>	<b>4</b>	<b>16</b>	
E7	5	1	5	
E8	4	2	8	
E9	5	3	15	
E10	6	2	12	

Шаг 2. Находим максимум в последнем столбце.



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

	$F_1$	$F_2$	$e_{i1}$	$e_{i2}$
$E_1$	1	1	1	
$E_2$	3	3	9	



E <sub>3</sub>	7	1	7
E <sub>4</sub>	2	2	4
E <sub>5</sub>	3	1	3
<b>E<sub>6</sub></b>	<b>4</b>	<b>4</b>	<b>16</b>
E <sub>7</sub>	5	1	5
E <sub>8</sub>	4	2	8
E <sub>9</sub>	5	3	15
E <sub>10</sub>	6	2	12
			<b>16</b>

Оптимальное решение: Все решения для которых  $e_{i1} \cdot e_{i2}$  максимально. В нашем случае это E<sub>6</sub>

### Геометрический метод расчета

Геометрическое решение можно найти используя преобразованную плоскость решений.

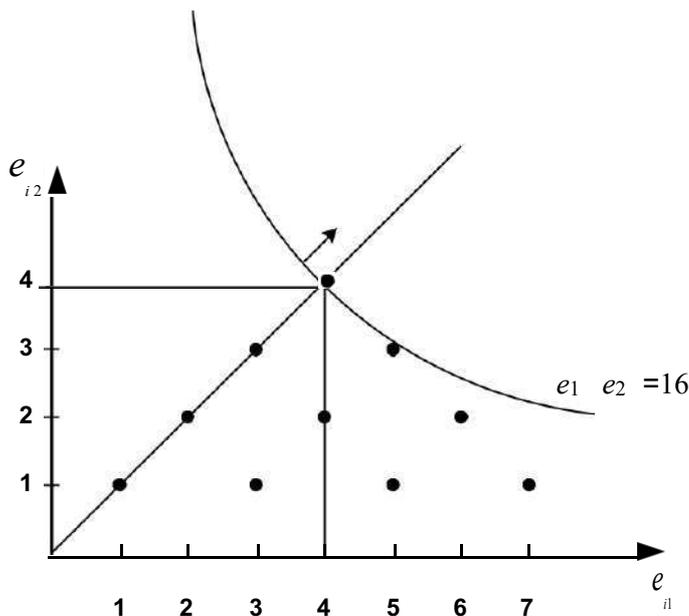


Рис.10. – Метод произведений

Решение на плоскости ищется следующим образом:

Шаг 1.

Строится новая плоскость решений, где осями будут  $e_{i1}$  и  $e_{i2}$ , а  $q_1 e_{i1}$  и  $q_2 e_{i2}$ . На этой плоскости строятся точки соответствующие решениями.

Шаг 2.

Строится направляющая –линия проведенная из начала координат под углом 45

Шаг 3.

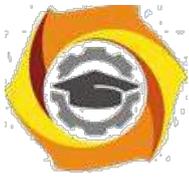
Линия, соответствующая критерию (парабола) движется вдоль направляющей от начала координат до касания последней точки, которая и будет решением. В данном случае точка с координатами (4; 4), т.е. E<sub>6</sub>

### ЗАДАНИЕ



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Составьте и отладьте программу на одном из языков программирования (Lazarus, C, VBA for Excel, Java...), которая выполняет принятие решения по критерию,



соответствующему Вашему варианту. Номер варианта определяется как остаток от деления вашего номера по журналу на 5 плюс 1.

Вариант 1 - критерий Гурвица  $Z_{HW} = \max_i c \min_j (e_{ij}) + (1 - c) \max_j (e_{ij})$ ;

Вариант 2 - Ходжа-Лемана  $Z_{HL} = \max_i v e_{ij} q_j + (1 - v) \min_j (e_{ij})$ ;

Вариант 3 - критерий Гермейера  $Z_G = \max_i \min_j (e_{ij} q_j)$ ;;

Вариант 4 - BL(ММ)-критерий

$$I_1 := i \mid i \in \{1, \dots, m\} \& e_{i0j} > 0 - \min_j (e_{ij} > 0) \text{ don } ,$$

$$I_2 := i \mid i \in \{1, \dots, m\} \& \max_j (e_{ij}) - \max_j (e_{i0j}) e_{i0j} > 0 - \min_j (e_{ij} > 0) ,$$

$$Z_{BL(ММ)} = \max_{1 \leq j \leq m} e_{ij} q_j ;$$

Вариант 5 - критерий произведений  $Z_P = \max_i \prod_{j=1}^m e_{ij} q_j$ .

Исходные данные – матрица решений должна вводиться из текстового файла в кодировке UTF-8 следующей структуры, разделитель – табуляция, число исходов и решений не ограничено (пример):

	F1	F2
E1	1	1
E2	3	3
E3	7	1
E4	2	2
E5	3	1
E6	4	4
E7	5	1
E8	4	2
E9	5	3
E10	6	2

Программа должна включать диалог выбора файла исходных данных, запрос вероятностей и выдачу сообщения о выбранном варианте и возможных результатах.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

99 В каких случаях применяются производные критерии?



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

22

22. Опишите критерий Гурвица (HW).
23. В каких случаях критерий HW превращается в критерий MM?
24. В каких случаях целесообразно применять критерий HW?
25. Опишите критерий Ходжа – Лемана.
26. На какие критерии опирается критерий Ходжа – Лемана (HL)?
27. В каких ситуациях целесообразно применять критерий Ходжа – Лемана (HL)
28. Когда HL критерий переходит в VL критерий?
29. Когда HL критерий переходит в MM критерий?
30. Опишите критерий Гермейера.
31. В каких ситуациях целесообразно применять критерий Гермейера?
32. В каких ситуациях целесообразно применять VL(MM) критерий?



## Практическое занятие № 3 Решение задач с учётом связей между критериями

*Критерии с прямоугольными конусами предпочтения. ММ-критерий. G-критерий. S-критерий. Функции предпочтения азартного игрока. Критерии с прямыми предпочтения.*

### Теоретические сведения

При введении и обсуждении критериев выбора решений стали заметны определенные сходства и различия между ними, которые теперь следует обсудить более подробно.

При этом мы хотим прояснить способ действия различных критериев как путем их взаимного сравнения, так и, насколько это возможно, с помощью соответствующих графических представлений.

При этом (в особенности с точки зрения последующих более общих построений) целесообразно сначала ввести в рассмотрение функцию  $e(y, x)$  вместо конечного числа значений  $e_{ij}$ . Здесь  $x$  представляет собой переменную для возможных состояний, а  $y$  — переменную для решений (мы используем при этом то 'положение, что в зависимости от заданного состояния  $x$  требуется выбрать подходящее значение переменной решения  $y$ ), которые принадлежат бесконечным, вообще говоря, областям  $F$  (для возможных состояний) и  $E$  (для переменных решения):  $e(y, x)$ ,

Случай конечных множеств вариантов решений и состояний, только и рассматривавшийся ранее, получается в этой постановке для конечных множеств  $E$  и  $F$ . Имея / виду наглядность графической интерпретации, ограничимся в дальнейшем двумя состояниями  $F_1$  и  $F_2$ , тогда как переменной решения  $y$  мы позволим принадлежать бесконечной, вообще говоря, области  $E$ .

## Минимаксный критерий принятия решений

### Геометрический метод расчета

Геометрический образ этого критерия представлен на рисунке 3.

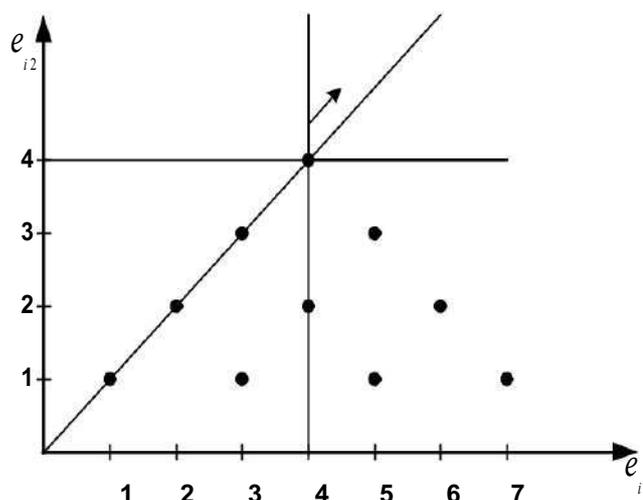


Рис.3. —Геометрическое представление минимаксного критерия Решение на плоскости ищется следующим образом:

Шаг 1.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Строится направляющая – линия проведенная из начала координат под углом 45  
Шаг 2. Линия, соответствующая критерию (прямой угол) движется вдоль  
направляющей от начала координат до касания последней точки, которая и



будет решением. В данном случае точка с координатами (4;4).

## Критерий Байеса–Лапласа

### Геометрический метод расчета

Геометрический образ этого критерия представлен на рисунке 4.

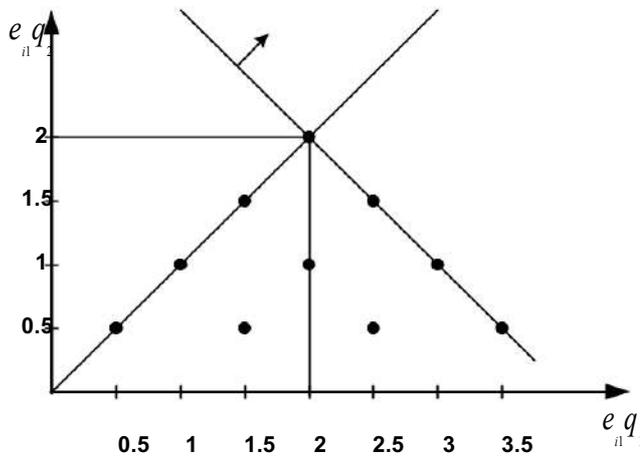


Рис.4. –Геометрическое представление критерия Байеса-Лапласа

Решение на плоскости ищется следующим образом:

Шаг 1.

Строим точки с координатами  $e_{1i} q_j$  и  $e_{2i} q_j$

	F <sub>1</sub>	F <sub>2</sub>	$e_{1i} q_j$	$e_{2i} q_j$	$\sum_{j=1}^m e_{ij} q_j$
E <sub>1</sub>	1	1	0,5	0,5	1
E <sub>2</sub>	3	3	1,5	1,5	3
<b>E<sub>3</sub></b>	<b>7</b>	<b>1</b>	<b>3,5</b>	<b>0,5</b>	<b>4</b>
E <sub>4</sub>	2	2	1	1	2
E <sub>5</sub>	3	1	1,5	0,5	2
<b>E<sub>6</sub></b>	<b>4</b>	<b>4</b>	<b>2</b>	<b>2</b>	<b>4</b>
E <sub>7</sub>	5	1	2,5	0,5	3
E <sub>8</sub>	4	2	2	1	3
<b>E<sub>9</sub></b>	<b>5</b>	<b>3</b>	<b>2,5</b>	<b>1,5</b>	<b>4</b>
<b>E<sub>10</sub></b>	<b>6</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>4</b>

Шаг 2.

Строится направляющая,—линия проведенная из начала координат под углом 45

Шаг 3.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Линия, соответствующая критерию – прямая перпендикулярная направляющей, движется вдоль направляющей от начала координат до касания последней точки, которая и будет решением. В данном случае это точки с координатами  $(2;2)$ ,  $(2,5;1,5)$ ,  $(3,5;0,5)$ ,  $(3,1)$ .



## Критерий Сэвиджа

### Геометрический метод расчета

Геометрический образ этого критерия представлен на рисунке 5.

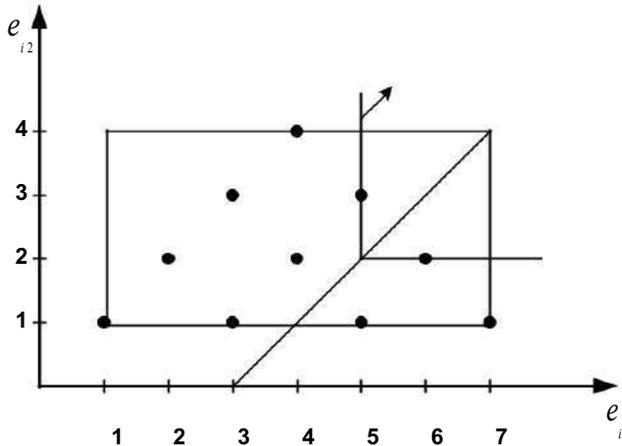


Рис.5 –Геометрическое представление критерия Сэвиджа

Решение на плоскости ищется следующим образом: Шаг

1.

Строим направляющую – прямую проходящую через утопическую точку под 45

Шаг 2.

Линия, соответствующая критерию – прямая прямой угол, движется вдоль направляющей от начала координат до касания последней точки, которая и будет решением. В данном случае это точки с координатами (5;3), (6;2), т.е. E<sub>9</sub> и E<sub>10</sub>

Составьте и отладьте программу на одном из языков программирования (Lazarus, C, VBA for Excel, Java...), которая выполняет принятие решения по критерию, соответствующему Вашему варианту. Номер варианта определяется как остаток от деления вашего номера по журналу на 5 плюс 1.

Вариант 1 - минимаксный  $Z_{MM} = \max_i \min_j (e_{ij})$ ;

Вариант 2 - Байеса-Лапласа  $Z_{BL} = \max_i \sum_{j=1}^m e_{ij} q_j$  ;

Вариант 3 - Сэвиджа  $Z_S = \min_i \max_j (\max_i (e_{ij}) - e_{ij})$  ;

Вариант 4 - расширенный минимаксный  $Z_{ME} = \max_p \min_{i,j} e_{ij} p_i q_j$  ;

Вариант 5 - азартного игрока  $Z_{AG} = \max_i \max_j (e_{ij})$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Исходные данные – матрица решений должна вводиться из текстового файла в кодировке UTF-8 следующей структуры, разделитель – табуляция, число исходов и решений не ограничено (пример):



	F1	F2
E1	1	1
E2	3	3
E3	7	1
E4	2	2
E5	3	1
E6	4	4
E7	5	1
E8	4	2
E9	5	3
E10	6	2

Программа должна включать диалог выбора файла исходных данных, запрос вероятностей и выдачу сообщения о выбранном варианте и возможных результатах.

### Контрольные вопросы

- / Начертите конус принятия решения для ММ критерия.
- / Начертите конус принятия решения для G критерия
- / Начертите конус принятия решения для S критерия
- / Начертите конус принятия решения для H критерия
- / Начертите конус принятия решения для VL критерия
- / Начертите конус принятия решения для HL критерия
- / Начертите конус принятия решения для P критерия
- / Начертите конус принятия решения для HW критерия
- / Начертите конус принятия решения для VL(ММ) критерия
- / В чем заключается принцип Бернулли?



## Практическое занятие 4 Принятие решения при наличии риска

*Решение задач. Оценка риска. Границы применимости критериев. Опорные величины для оценки риска. Опорные величины, зависящие от внешних факторов. Оценка значимости параметра.*

### Теоретические сведения

Полное устранение риска при принятии решений зачастую не требуется. Определенная степень риска даже вводится сознательно, так как принятие решения без риска может оказаться невыгодным. Но любой риск при этом должен учитываться по возможности полно (описываться количественно и ограничиваться) и не превышать уровня, при котором результат достигается с достаточной надежностью.

### Принятие решения при наличии риска

При оценке риска выбирают опорное для оценки значение  $Z$  (например, оптимум по минимаксному критерию  $Z_{MM}$ ). Это опорное значение используется для определения степени неоптимальности варианта решения  $E_i$  в виде так называемого дефекта варианта решения  $E_i$  относительно опорного значения оценочной функции:

$$e_i = Z - \min_{j} e_{ij} .$$

Максимальную разность дефектов при рассмотрении всех возможных вариантов решения  $E_i$ ,  $i=1, \dots, m$ , характеризуют как возможный риск:

$$e_{возм} = \max_i (Z - \min_{j} e_{ij}) - \min_j (Z - \min_{i} e_{ij}) .$$

Возможный риск независимо от информации о параметрах, имеющейся по результатам выборки, а также от числа реализаций процесса принятия решения представляет собой максимально возможную величину нереализуемой полезности решения.

В некоторых случаях имеет смысл выйти за рамки строгого следования минимаксному критерию, если принимающий решение готов пойти на некоторый риск, определяемый величиной  $e_{доп}$ .



Для некоторых внешних условий, имеющих большую вероятность реализации, могут получиться варианты решения, которые дают заметный выигрыш по сравнению с оптимальным опорным значением варианта решения (например, оптимальным решением по ММ-критерию). С целью оценки конкурентоспособности таких решений для каждого варианта  $E_i$  вводят специальную величину, равную сумме минимального результата  $\min_j e_{ij}$ ,  $j = 1, \dots, n$ , и эффекта риска:

...,  $n$ , и эффекта риска:

$$\min_j e_{ij} + i, \quad \text{где } i = \min(i_{\text{возм}}, \text{дон}).$$

Максимальный риск при рассмотрении всех вариантов решения  $E_i$ ,  $i = 1, \dots, m$ , равен  $\max_i i = \max_i \min_j (i_{\text{возм}}, \text{дон})$ .

Оценочная функция при этом принимает вид:

$$V(i) \sum_{j=1}^n e_{ij} h_j + (1 - V(i)) \min_j (e_{ij} + i),$$

где  $V(i)$  - один из доверительных факторов,  $h_j$  - частота (или, если известна, вероятность) возникновения внешнего условия  $F_j$ .

Множество всех вариантов решения  $E^*(i)$ , обеспечивающих максимум величины  $i$ :

$$E^*(i) = \{E_l \mid i_l = \max_i i\}.$$

Опорные величины для оценки риска

Риском считают реализацию случая, когда вариант решения  $E_i$  при внешнем условии  $F_j$  дает результат меньше ожидаемого. Такую ожидаемую величину и принимают в качестве опорной для оценки риска. Опорные величины разделяют на *зависящие* и *не зависящие* от внешних факторов.

Не зависящей от внешних факторов опорной величиной  $e_z$  может быть любая вещественная величина из диапазона  $\min_i \min_j e_{ij} \leq e_z \leq \max_i \max_j e_{ij}$ .

Величина  $i = e_z - \min_j e_{ij} = \max_j (e_z - e_{ij})$  называется *возможным дефектом выбора*

варианта решения  $E_i$ . Так как отрицательные значения  $i$  не могут считаться дефектом, то будем рассматривать величину  $i^+ = \max(i, 0)$  в качестве характеристики варианта решения  $E_i$ . Вариант решения  $E_i$  будет иметь дефект, когда  $i^+ > 0$ , и будет свободен от дефекта, когда  $i^+ = 0$ .

В качестве значений для независящих от внешних факторов опорных величин оценки риска используют значения известных критериев принятия решения (например, минимаксного критерия).

Опорную величину  $e_z$ , зависящую от внешних факторов, представляют в виде функции от всех  $m \cdot n$  значений результатов решения  $e_{ij}$ :  $e_z = e_z(e_{ij})$ , где  $i = 1, \dots, m$ ,  $j = 1, \dots, n$ . Тогда дефект, возможный при выборе варианта решения  $E_i$ ,



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

определяется как

$$i = (e_{ij}) - \min_j e_{ij} = \max_j (e_{ij}) - e_{ij}.$$

Выбор оптимального варианта  $E_i^*$  дает минимальный дефект  $\Delta^* = \Delta_i^* = \min_i \Delta_i$ .



Относительный риск при выборе соответствующего варианта решения  $E_i$  определяется как разность между дефектом  $i$  варианта решения  $E_i$  и минимальным дефектом  $*$ :

$$r_i = \frac{e_i - \min_j e_{ij}}{\min_j e_{ij}} = \frac{(e_i - \min_j e_{ij}) - \min_j [(e_i - \min_j e_{ij})]}{\min_j e_{ij}}$$

Опорные величины могут быть определены для каждого  $n$  внешних из

состояний  $F_1, \dots, F_n$  отдельно с помощью функции  $m$  переменных:  $e_{zj} = (e_{1j}, \dots, e_{mj}, \dots, e_{mj})$ , имеющей смысл в диапазоне  $\min_i e_{ij} \leq e_{zj} \leq \max_i e_{ij}$ .

Возможным дефектом выбранного варианта решения  $E_i$  или оценкой риска, сопутствующего такому решению, называется величина  $i = \max_j (e_{zj} - e_{ij})$ , у

которой учитываются только положительные значения  $i^+$ .

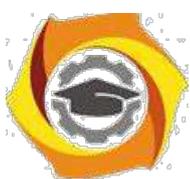
42. качество зависимых от внешних факторов опорных величин выбираются, например, граничные значения диапазона определения опорной величины:

$$e_{zj} := \max_{ij} e_{ij} \quad \text{и} \quad e_{zj} := \min_i e_{ij}, \quad \text{а также среднее значение} \quad e_{zj} := \frac{1}{m} \sum_{i=1}^m e_{ij}$$

Оптимальный выбор варианта решения  $E_i^*$  дает минимальный дефект  $i^* = \min_i i$ .

Относительный риск выбора варианта решения  $E_i$ :

$$r_i = \frac{e_i - \min_j e_{ij}}{\min_j e_{ij}} = \frac{\max_j (e_{zj} - e_{ij}) - \min_j [\max_j (e_{zj} - e_{ij})]}{\min_j e_{ij}}$$



**Задание.**

Для определенной Вами в лабораторной работе №1 матрицы решений выбрать определенные вариантом задания опорные величины для оценки риска и для каждого варианта решения рассчитать возможный и относительный дефект выбора этого варианта решения, определить оптимальный выбор с точки зрения минимального дефекта и с точки зрения, допускающей определенный риск доп (эпсилон ДОПУСТИМОЕ – допустимый риск).

**Варианты задания**

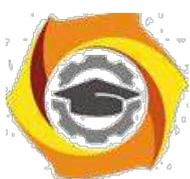
40. столбце  $e_z^{const}$  задан критерий выбора, определяющий опорную величину  $Z$ , не зависящую от внешних факторов;
41. столбце  $e_{zj}$  - определяется способ вычисления опорной величины, зависящей от внешних факторов, следующим образом:  
 max - означает вычисление по формуле  $e_{zj} := \max_i e_{ij}$  ,  
 min - по формуле  $e_{zj} := \min_i e_{ij}$  ,  
 среднее -  $e_{zj} := \frac{1}{m} \sum_{i=1}^m e_{ij}$  ;
59. столбце доп задан уровень допустимого риска в процентах от максимально возможного.

Номер варианта	Вид доверительного фактора		$e_{zconst}$	$e_{zj}$	доп %
1	эмпирический	0,05	MM	min	25
2	прогностический	0,05	S	max	10
3	эмпирический	0,1	BL	среднее	30
4	прогностический	0,1	G	min	40
5	эмпирический	0,2	HL	max	35
6	прогностический	0,2	HW	среднее	45
7	эмпирический	0,3	BL(MM)	min	50
8	прогностический	0,3	P	max	55
9	эмпирический	0,001	MM	среднее	5
10	прогностический	0,001	S	min	2
11	эмпирический	0,005	BL	max	10
12	прогностический	0,005	G	среднее	25
13	эмпирический	0,01	HL	min	40
14	прогностический	0,01	HW	max	30
15	эмпирический	0,02	BL(MM)	среднее	45
16	прогностический	0,02	P	min	35
17	эмпирический	0,025	MM	max	15
18	прогностический	0,025	S	среднее	50



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

19	эмпирический	0,03	BL	min	70
20	прогностический	0,03	G	max	20
21	эмпирический	0,04	HL	min	60
22	прогностический	0,04	HW	max	65
23	эмпирический	0,15	BL(MM)	среднее	25
24	прогностический	0,15	P	min	40



25	эмпирический	0,25	MM	max	10
26	прогностический	0,25	S	среднее	15
27	эмпирический	0,035	BL	min	45
28	прогностический	0,035	G	max	10
29	эмпирический	0,045	HL	среднее	20
30	прогностический	0,045	HW	min	10

### Контрольные вопросы

39. В чем разница между разумным риском и риском азартного игрока?
40. Какими характеристиками может быть количественно описан риск?
41. Как можно вычислит степенит неоптимальности решения?
42. Как определяется возможный риск?
43. Какой критерий целесообразно применять в случае малых выборок?
44. Какой критерий целесообразно применять в случае достаточно больших выборок?
45. Дайте определение опорной величине оценки риска.
46. В каких границах находится опорная величина оценки риска?
47. Как определяется возможный дефект выбранного варианта решения?
48. Как определить риск, сопутствующий принятому решению?



## Практическое занятие 5 Обработка экспертной информации

*Решение задач. Подготовка и проведение оценок. Обработка результатов экспертных оценок. Интерквартиль оцениваемой величины. Взвешивание результатов. Гибкий выбор при субъективной полезной информации*

### Теоретические сведения

**Экспертное ранжирование** *Ранг* – степень отличия по какому-либо признаку, № *ранжирование* – процесс определения рангов, относительных количественных оценок степеней отличий по качественным признакам. Например, студенты отличаются друг от друга и от «эталона-отличника» по признаку знаний материала дисциплин и опыта применения этих знаний в решении задач, а сам процесс ранжирования называется экзаменом. Качественными признаками в данном случае выступают наличие ответов по вопросам билета, глубина и полнота ответов, приведение примеров из практики.

Ранжирование применяется в случаях, когда невозможна или нецелесообразна непосредственная оценка. При этом «ранжирование объектов содержит лишь информацию о том, какой из них более предпочтителен, и не содержит информации о том, насколько или во сколько раз один объект предпочтительнее другого. Объект, расположенный в ранжировании эксперта, скажем, на третьем месте, может превосходить объект, расположенный на четвертом месте, в 1,01 раза, а может превосходить и в 101 раз».

### Алгоритм метода одномерного шкалирования

1 Вычисляют матрицу  $P = A^j / N$ , где  $A^j$  – ранжировка, данная экспертом. Элемент  $P_{ij}$  матрицы  $P$  интерпретируют как вероятность предпочтения объекта  $J$ -го.

65. Находят  $Z_{ij}$  по формуле

$$G(Z_{ij}) = p_{ij} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Z_{ij}} e^{-t^2/2} dt \quad (1)$$

с использование таблиц нормального распределения, исходя из известных  $p_{ij}$ . Величина  $Z_{ij}$  измеряется в единицах стандартного отклонения.

3 Образуют матрицу  $Z = (Z_{ij})$ . Подсчитывают сумму оценок  $Z_i = \sum_{j=1}^n Z_{ij}$  и среднее значение  $\bar{Z}_i = Z_i / n$ . Величину  $\bar{Z}_i$  принимают за искомую оценку объекта  $A_i$  ( $i = 1, n$ ).

47. Определяют величины  $P_i^- = G(\bar{Z}_i)$  по формуле (1), которые нормируют по формуле:

$$P_i^* = \frac{\bar{P}_i}{\sum_{j=1}^n \bar{P}_j} ;$$

$P_i^*$  называют показателями относительной важности объекта.

45. Осуществляют проверку на непротиворечивость. Для этого по формуле (1)



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

находят  $\bar{p}_{ij} = \bar{G}(Z_i - Z_j)$  и вычисляют разности  $_{ij}k$  (  $k$  – количество) между полученными значениями  $p_{ij}$  и исходными  $p_{ij}$ . Определяют среднее отклонение:



$$\left| \begin{matrix} i & j \\ i & j \end{matrix} \right| / k$$

если оно мало, то это свидетельствует о непротиворечивости полученных экспертных ранжировок.

### Пример выполнения задания

Дана матрица результатов оценивания  $m$  параметров информационной системы  $d$  экспертами –  $A_{dm}$ . Оценить относительную важность параметров информационной системы, используя одномерное шкалирование как метод обработки экспертной информации.

Матрица результатов опроса имеет вид:

$$A_{dm} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{matrix} 2 & 3 & 1 & 4 \\ 1 & 4 & 2 & 3 \\ 4 & 2 & 1 & 3 \end{matrix} \end{matrix}$$

**Решение.** Вычисляется матрица  $A$ , где  $A_j$  – ранжировка, данная  $j$ -м экспертом. Матрица квадратная, ее размерность соответствует количеству параметров.

Таблица – Матрица  $A$

	1	2	3	4
1	0	3	2	3
2	1	0	1	3
3	2	3	0	4
4	1	1	0	0

Строится матрица вероятностей предпочтения каждого параметра информационной системы экспертами:  $P_{ij}$  – вероятность предпочтения  $i$ -го параметра  $j$ -му.

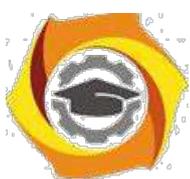
Таблица – Матрица  $P$

	1	2	3	4
1	0	0,75	0,5	0,75
2	0,25	0	0,25	0,75
3	0,5	0,75	0	1
4	0,25	0,25	0	0

Далее по формуле 1 строится матрица  $Z$ , используя таблицы функции обратной функции нормального распределения. Подсчитывается сумма оценок  $Z_i = \sum_{j=1}^n Z_{ij}$  и среднее значение  $Z_i = Z_{ij} / n$ .

Таблица – Матрица  $Z$

	1	2	3	4	$Z_i$	$Z_i$
1	0	0,67449	0	0,67449	1,34898	0,337245



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

<b>2</b>	-0,67449	0	-0,67449	0,67449	-0,67449	-0,16862
<b>3</b>	0	0,67449	0	3,174	3,84849	0,962123
<b>4</b>	-0,67449	-0,67449	-3,174	0	-4,52298	-1,13075



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Определяются величины  $P_i^- = G(Z_i^-)$  по формуле (1), которые нормируют по формуле  $P_i^* = \bar{P}_i / \sum_{j=1}^n \bar{P}_j$  :

Таблица – Относительная важность параметров

Parametr	$\bar{P}_i$	$P_i^*$
1	0,632	0,311945
2	0,433	0,213722
3	0,832	0,410661
4	0,129	0,063672
Summa	2,026	1

Далее осуществляется проверка на непротиворечивость оценок экспертов. Для этого по формуле 1 находятся значения  $\bar{P}_{ij} = G(Z_i - Z_j)$  и вычисляются разности  $ij$  ( $k$  – количество разностей) между полученными значениями  $\bar{P}_{ij}$  и исходными  $P_{ij}$  :

Таблица – Разности и отклонения

$\bar{Z}_i - \bar{Z}_j$	$\bar{P}_{ij}$	$P_{ij}$	$ij$	$ ij $	
Z1-Z2	0,505868	0,694	0,75	-0,056	0,056
Z1-Z3	-0,62488	0,266	0,5	-0,234	0,234
Z1-Z4	1,46799	0,929	0,75	0,179	0,179
Z2-Z3	-1,13075	0,129	0,25	-0,121	0,121
Z2-Z4	0,962123	0,832	0,75	0,082	0,082
Z3-Z4	2,092868	0,982	1	-0,018	0,018

$$\sqrt{\sum_{i,j=1}^n |ij|} / k = 0.115$$

Определяют среднее отклонение по формуле  $\sqrt{\sum_{i,j=1}^n |ij|} / k = 0.115$ . Так как  $11,5\% < 20\%$ , оценки, данные экспертами могут быть использованы для принятия решения о важности параметров информационной системы: наиболее важным является третий параметр, наименее – четвертый.

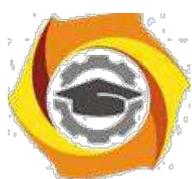
**Задание.**

Имеется матрица результатов оценивания  $m$  параметров информационной системы  $d$  экспертами. Оценить относительную важность параметров информационной системы, используя одномерное шкалирование как метод обработки экспертной информации.

Студент самостоятельно ранжирует объекты, выступая в роли экспертов (набор параметров выбирается студентом). Объекты ранжируются по степени важности.

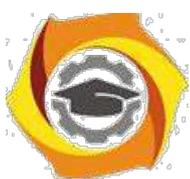
Таблица 1 – Данные результатов оценки

№ варианта	$m$	$d$
1	6	5
2	7	4
3	8	5
4	5	6



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

5	6	5
6	6	4
7	5	6
8	5	5



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

35

9	5	5
10	5	5
11	7	6
12	7	6
13	8	4
14	8	4
15	9	4
16	9	4
17	5	6
18	7	6
19	6	5
20	7	4
21	8	5
22	9	4
23	5	4
24	5	7
25	6	6

**Контрольные вопросы**

54. В чем заключаются недостатки субъективных оценок при принятии технических решений?
55. Когда допустимо применение субъективных оценок?
56. Что следует предпринять в случае необходимости применения субъективных оценок?
57. Перечислите этапы подготовки к субъективной оценке?
58. Как определить необходимое количество опрашиваемых специалистов – экспертов?
59. Какими качествами должен обладать специалист – эксперт?
60. Каким методом целесообразно обрабатывать результаты опроса экспертов?
61. Как ввести в результаты оценивания данные малой выборки?
62. В чем заключается опасность неконтролируемых умозаключений экспертов?
63. Какой критерий целесообразно применять в случае обработки субъективной экспертной информации?



## Практическое занятие № 6 Анализ ситуаций принятия решений

*Решение задач. Построение дерева решений и платёжной матрицы.*

### Теоретические сведения

Процесс принятия решений с помощью дерева решений в общем случае предполагает выполнение следующих пяти этапов.

**Этап 1. Формулирование задачи.** Прежде всего, необходимо отбросить не относящиеся к проблеме факторы, а среди множества оставшихся выделить существенные и несущественные. Это позволит привести описание задачи принятия решения к анализируемой форме. Должны быть выполнены следующие основные процедуры:

В определение возможностей сбора информации для экспериментирования и реальных действий;

В составление перечня событий, которые с определенной вероятностью могут произойти;

В установление временного порядка расположения событий, в исходах которых содержится полезная и доступная информация, и тех последовательных действий, которые можно предпринять.

**Этап 2. Построение дерева решений.**

**Этап 3. Оценка вероятностей состояний среды,** т.е. сопоставление шансов возникновения каждого конкретного события. Следует отметить, что указанные вероятности определяются либо на основании имеющейся статистики, либо экспертным путем.

**Этап 4. Установление выигрышей** (или проигрышей, как выигрышей со знаком минус) для каждой возможной комбинации альтернатив (действий) и состояний среды.

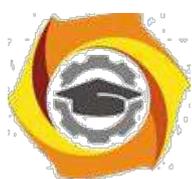
**Этап 5. Решение задачи.**

Прежде чем продемонстрировать процедуру применения дерева решений, введем ряд определений. В зависимости от отношения к риску решение задачи может выполняться с позиций так называемых «объективистов» и «субъективистов». Поясним эти понятия на следующем примере. Пусть предлагается лотерея: за 10 долл. (стоимость лотерейного билета) игрок с равной вероятностью  $p=0,5$  может ничего не выиграть или выиграть 100 долл. Один индивид пожалеет и 10 долл. за право участия в такой лотерее, т.е. просто не купит лотерейный билет, другой готов заплатить за лотерейный билет 50 долл., а третий заплатит даже 60 долл. за возможность получить 100 долл. (например, когда ситуация складывается так, что только имея 100 долл., игрок может достичь своей цели, поэтому возможная потеря последних денежных средств, а у него их ровно 60 долл., не меняет для него ситуации).

*Безусловным денежным эквивалентом* (БДЭ) игры называется максимальная сумма денег, которую ЛПР готов заплатить за участие в игре (лотерее), или, что то же, та минимальная сумма денег, за которую он готов отказаться от игры. Каждый индивид имеет свой БДЭ.

Индивида, для которого БДЭ совпадает с ожидаемой денежной оценкой (ОДО) игры, т.е. со средним выигрышем в игре (лотерее), условно называют объективистом, индивида, для которого БДЭ ОДО, – субъективистом. Ожидаемая денежная оценка рассчитывается как сумма произведений размеров выигрышей.

Основными обозначениями при построении дерева решений являются:



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

решение принимает игрок;



ООО - - решение «принимает» случай;

- отвергнутое решение.

## Основные определения концепции полезности

**Полезность  $W$**  – определенное число, приписываемое ЛПР каждому возможному результату (исходу). Полезность выражает степень удовлетворения, которое получает субъект в результате потребления товара или услуги.

**Функция полезности (функция Неймана-Моргенштерна)  $U/W$**  – показывает полезность, которую приписывает ЛПР каждому возможному результату в зависимости от его отношения к риску.

**Ожидаемая полезность события** – сумма произведений вероятностей возникновения

$$W = \sum_{i=1}^n p_i W_i$$

данного события на значение полезности последствий этих событий

Выбор ЛПР в условиях риска формализуется при помощи понятия потери, при этом ЛПР проявляет свои индивидуальные вкусы и склонность к риску. Решение ЛПР может быть найдено на основе следующего алгоритма:

1 Присваиваются произвольные значения полезности выигрышу для лучшего и худшего последствия, причем худшему из последствий ставится в соответствие меньшее значение полезности.

2 Игроку предоставляется выбор:

24. получить определенную гарантированную сумму  $W$ , которая находится в промежутке между худшим ( $s$ ) и лучшим ( $S$ ) значениями выигрышей  $s \leq W \leq S$ ;

44. принять участие в игре, т.е. получить с вероятностью  $p$  наибольшую денежную сумму  $S$  и с вероятностью  $(1-p)$  получить наименьшую денежную сумму  $s$ , при этом вероятность следует изменять (уменьшать или повышать) до тех пор пока ЛПР не станет безразличным к отношению выбора между гарантированной суммой и игрой.

$$W = p_0 U(S) + (1 - p_0) U(s), \text{ где } p_0 - \text{ заданная}$$

вероятность.

Функция полезности имеет вид:

**Безусловный денежный эквивалент (БДЭ)** – максимальная сумма денег, которую ЛПР готов заплатить за участие в игре (лотерее) или минимальная сумма денег, за которую он готов отказаться от игры.

**Ожидаемая денежная оценка (ОДО)** – средний выигрыш в игре.

**Вывод:** если БДЭ = ОДО ЛПР – объективист. Если БДЭ  $\neq$  ОДО ЛПР – субъективист (если БДЭ  $>$  ОДО ЛПР – склонен к риску; если БДЭ  $<$  ОДО ЛПР – не склонен к риску).

Основные функции полезности используются для изучения, анализа и оценки поведения субъектов риска:

$U(x) = a + bx; (b \geq 0)$  - функция, отражающая нейтральность к риску.

$U(x) = \log_a (x + b); (x \geq -b, a > 1)$  - функция, выражающая убывающую несклонность



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

42. риску.

$U(x) = -e_{-cx}$  ; (с > 0) - постоянная несклонность к риску.

$U(x) = -e_{cx}$  ; (с > 0) - постоянная склонность к риску.



5  $U(x) = a + bx - cx^2$ ; ( $c > 0$ ;  $x \leq \frac{b}{2c}$ ) - возрастающая несклонность к риску.

47.  $U(x) = -x^2$ ; ( $x > 0$ ) - возрастающая склонность к риску.

7  $U(x) = a + bx + cx^2$ ; ( $c > 0$ ;  $x \leq \frac{b}{2c}$ ) - убывающая склонность к риску.

8 Функция с интервальной нейтральностью к риску.

Одним из основных видов функции полезности, характеризующей финансовое поведение людей, является функция  $U(x) = \ln x$ , т.е. полезность бесконечно малого выигрыша прямо пропорциональна этому выигрышу и обратно пропорциональна денежной сумме, которой игрок обладает. Из этого следует, что если полезность описывается

функцией  $U(x) = \ln x$ , то потери более ощутимы, чем выигрыш.

**Детерминированный эквивалент лотереи**  $L$  – это гарантированная сумма  $x$ , получение которой эквивалентно участию в лотерее, т.е.  $x = L$ . Итак  $x$  определяется из уравнения

$$U(x) = M[U(x)], \text{ или } x = M^{-1} MU(x). \quad (1)$$

**Премия за риск** – это сумма, которой субъект готов пожертвовать из среднего выигрыша за то, чтобы избежать риска, связанного с лотерей. Премию за риск определяют таким образом:

$$r(x) = M[x] - x = x - x. \quad (2)$$

**Страховая сумма** – величина детерминированного эквивалента с противоположным знаком.

### Задание

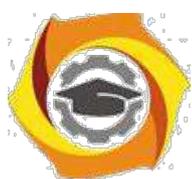
Инженер выбирает оптимальный технологический процесс выпуска новой продукции на крупном предприятии. Размер условного выигрыша, который предприятие может получить, зависит от благоприятного или неблагоприятного состояния среды (табл.1).

Таблица 1 – Исходные данные

Номер альтернативы	Действия инженера	Условный выигрыш, грн.	
		благоприятный исход	неблагоприятный исход
1	технологический процесс 1	$20+5*K$	$-(18+2*K)$
2	технологический процесс 2	$10+20*K$	$-(10+2*K)$
3	технологический процесс 3	$2*K$	$2*K$

Перед принятием решения руководство должно определить, заказывать ли дополнительное исследование среды или нет (стоимость услуги  $2*K$ , где  $K$  – номер варианта).

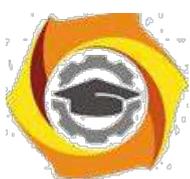
Возможности предприятия в виде условных вероятностей благоприятности и неблагоприятности среды представлены в табл. 3.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*Таблица 3 – Вероятности наступления прогнозных значений*

Прогноз	Фактически	
	Благоприятный	Неблагоприятный



Благоприятный	0,85	–
Неблагоприятный	–	0,65

Предположим, что фирма, которой заказали прогноз состояния среды, утверждает:

- ситуация будет благоприятной с вероятностью 0,45;
- ситуация будет неблагоприятной с вероятностью 0,55.

*Задание.* Построить программный модуль для выбора оптимальной альтернативы с помощью дерева решений, предусмотреть как максимизацию, так и минимизацию условного выигрыша. Рассчитать ценность точной информации без обращения за дополнительной информацией. Предусмотреть возможность введения исходных данных пользователем и вывод сообщения о выборе оптимальной альтернативы.

### Контрольные вопросы

22. На какие группы можно разделить решения исходя из затраченных средств для обработки данных?
23. Перечислите основные фазы процесса принятия решения.
24. Как влияет неопределённость на процесс принятия решения?
25. Какие виды неопределённостей могут встречаться при принятии технических решений?
26. Для чего применяется дерево событий? Как его построить?
27. Для чего применяется дерево решений? Как его построить?
28. Что понимают под стратегией принятия решения?
29. Какие методы анализа можно применить для оценки погрешности принятого решения?
30. Чем может быть вызвана погрешность при принятии решения?
31. Как влияет взаимное влияние мнений на качество принимаемого коллективного решения?



## Практическое занятие 7 Процесс принятия решения

*Решение задач. Построение процедуры принятия решения для заданного варианта. Одношаговые схемы принятия решения. Процессы выбора решения согласно классическим, производным и гибким критериям. Многошаговые решения. Поиск оптимальной стратегии.*

### Учебный материал

Важную роль среди инструментов, оказывающих помощь в принятии решений, играют компьютерные системы. В связи с тем, что процесс принятия решений состоит из этапов, каждый из которых имеет свою специфику, на рынке программных продуктов существуют и пользуются спросом программы, предназначенные для применения на отдельных этапах разработки управленческого решения. Например, система *Decision Explorer* (продукт компании Vanxia Software Ltd) предназначена для применения на стадии исследования проблемы и формирования альтернатив.

Рассмотрим названную программу «в действии», исследовав возможности ее применения в процессе решения задачи вывода предприятия из кризиса.

Объектом нашего исследования выступит отечественное предприятие с оптимистичным названием «Авангард» и проблемами, типичными для отечественного машиностроения.

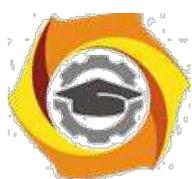
### Ситуация на ОАО «Авангард»

ОАО «Авангард» – компания среднего масштаба, изготавливающая двигатели, запчасти, комплектующие для тракторов и автомобилей, а также товары народного потребления (ТНП). На протяжении многих лет компания имела высокую репутацию благодаря превосходному качеству продукции и сервиса. Однако ситуация изменилась в середине 1990-х годов. Во-первых, резко сократилось количество заказов на основную продукцию предприятия – двигатели для сельскохозяйственных машин. При этом уменьшение платежеспособного спроса сопровождалось повышением требований заказчиков

42. приобретаемой продукции. Во-вторых, была осуществлена приватизация, и с этого момента развитие производства необходимо было осуществлять уже не за счет централизованных источников, а за счет самостоятельно заработанных средств. Как следствие, в течение последних двух лет компания отчитывается об убытках.

современном состоянии предприятия говорят следующие цифры: половина станочного парка эксплуатируется более 25 лет; номенклатура продукции завода обновляется ежегодно только на 1 - 2%; уровень затрат на 1 грн. продукции составляет 90 копеек при плане 75 коп.

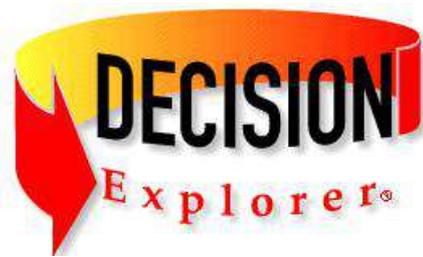
В текущем году руководство предприятия решительно приступило к формированию антикризисной программы развития производства и сбыта. Было создано экстренное совещание, на повестке дня которого стоял вопрос: «Как мы докатились до жизни такой?». Другими словами, руководство приступило к первому этапу процесса принятия решений – этапу, посвященному исследованию создавшейся проблемной ситуации, выявлению её причин и последствий. По настоянию начальника отдела компьютеризации, в качестве помощника в нелёгком деле диагностики проблемы выступила компьютерная система *Decision Explorer*.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Познакомимся с этой системой поближе.

**Decision Explorer**



*Decision Explorer* представляет собой инструмент построения и анализа дескриптивных нематематических моделей, описывающих набор элементов исследуемой системы (например, проблемы) и связи между ними. Примерами дескриптивных моделей выступают: структурная модель предметной области, диаграмма причинно-следственных связей, дерево целей.

Основные функции *Decision Explorer* – обработка качественной информации о предметной области, анализ структуры исследуемой системы, а также активизация и организация мышления ЛПР.

Работа с *Decision Explorer* включает две основные стадии: *синтез* и *анализ* (рис. 2). На первой стадии в процессе мозгового штурма формируется визуальная модель, отображающая совокупность элементов исследуемой системы и связи между ними. На второй стадии получившийся результат творчества подвергается анализу с помощью специализированного инструментария *Decision Explorer*. Затем полученный результат обсуждается, и исходная модель корректируется.

При применении *Decision Explorer* на первом этапе процесса принятия решения в качестве моделируемой системы выступает проблемная ситуация. В этом случае элементами модели являются причины и следствия проблемы, которые визуальным образом изображаются в виде прямоугольных или иных фигур с соответствующими надписями. Взаимосвязи между элементами модели визуальным образом изображаются в виде стрелок.

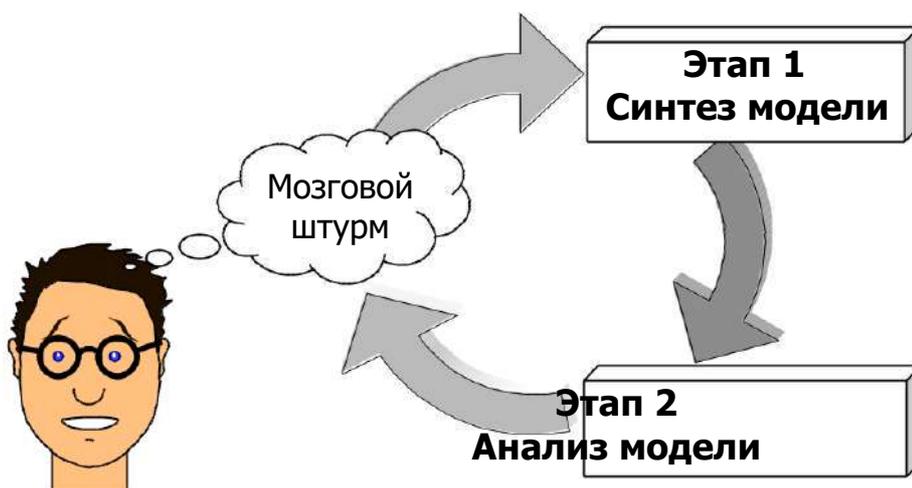


Рис. 2. Стадии работы с *Decision Explorer*

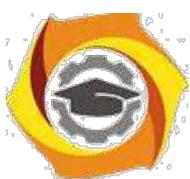
Для создания визуальной модели на стадии синтеза в среде *Decision Explorer* требуется ряд несложных действий:

- а) *двойной щелчок мыши* – для создания элементов модели;
- б) *операция Drag&Drop* – для создания связей между элементами модели (нужно указать на исходный элемент, нажать кнопку мыши и, удерживая ее, перенести курсор мыши на второй элемент).



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

Для выбора стиля оформления элементов модели служит команда **Property – Show style selector**, а также кнопка  на панели инструментов. Создать свой стиль оформления (например, стиль, позволяющий использовать в названиях элементов символы *кириллицы*)



можно с помощью команды **Property - Concept style properties** и кнопки . Масштаб изображения на экране задается командой **View-Display scale**. Переместить выбранный элемент модели можно с помощью команды **Move concept** контекстного меню. Для перемещения группы элементов следует дополнительно использовать клавишу **Shift** на клавиатуре.

Модель проблемной области, сформированная специалистами ОАО «Авангард», представлена на рис. 3.

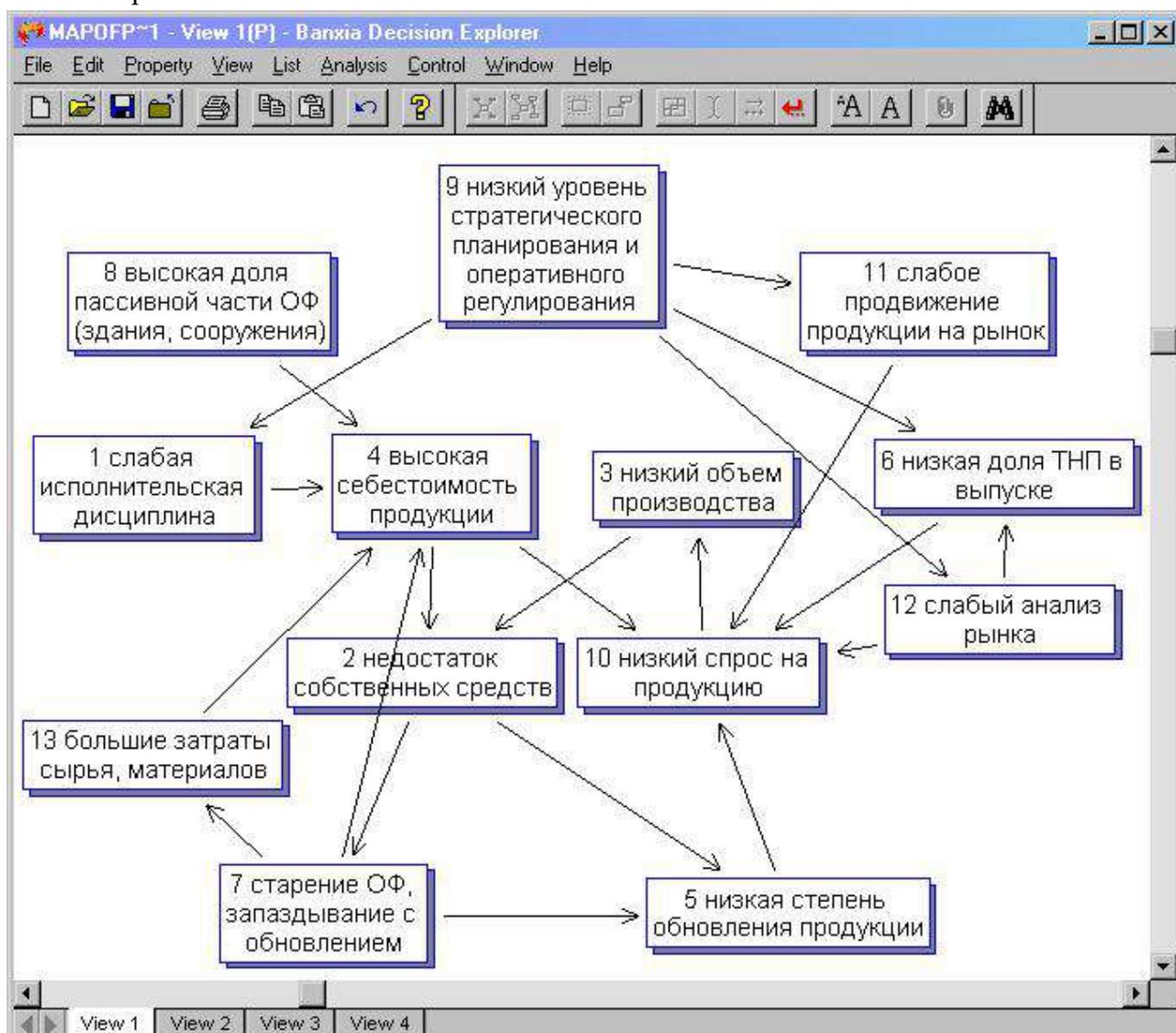


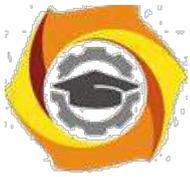
Рис. 3. Карта проблемы, построенная в среде *Decision Explorer*

На стадии анализа полученная визуальная модель подвергается исследованию с помощью следующих процедур:

с помощью команды **Central** и **Domain** определяются «ключевые» элементы модели – то есть элементы, имеющие наибольшее количество связей;

с помощью команды **Cluster** выявляются относительно независимые подсистемы элементов модели (если они есть);

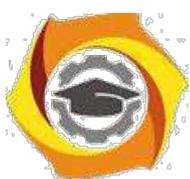
команда **Map concept** строит иерархию «подчиненных» элементов для выделенного звена модели;



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

-команда **Loop** выявляет петли положительной и отрицательной обратной связи между элементами модели.

Эти и другие команды вводятся с клавиатуры в командную строку либо выбираются в списке меню **Analysis**. Командная строка **DE** появляется внизу экрана автоматически при



попытке ввода текста с клавиатуры. Другой способ отобразить командную строку – нажать на клавиатуре клавишу управления курсором  ↑.

Посмотрим, к каким результатам привело использование инструментария *Decision Explorer* специалистами предприятия «Авангард».

Применение команды **Central 3** (синтаксис команды - **Central n**, где **n** – заданное пользователем количество *ключевых элементов*) позволило выявить три ключевых звена системы причинно-следственных связей (рис. 4). Полученная информация полезна тем, что она дает возможность сформировать 3 основные задачи антикризисного управления предприятием:

30. увеличение спроса на продукцию,
31. снижение себестоимости,
32. повышение эффективности стратегического планирования и оперативного регулирования.

Заметим, что если ваше интуитивное представление о ключевых элементах проблемы не совпало с результатами применения команды **Central**, это означает, что построенная модель нуждается в уточнении: ключевые элементы проблемы должны быть дополнены новыми причинными и следственными связями.

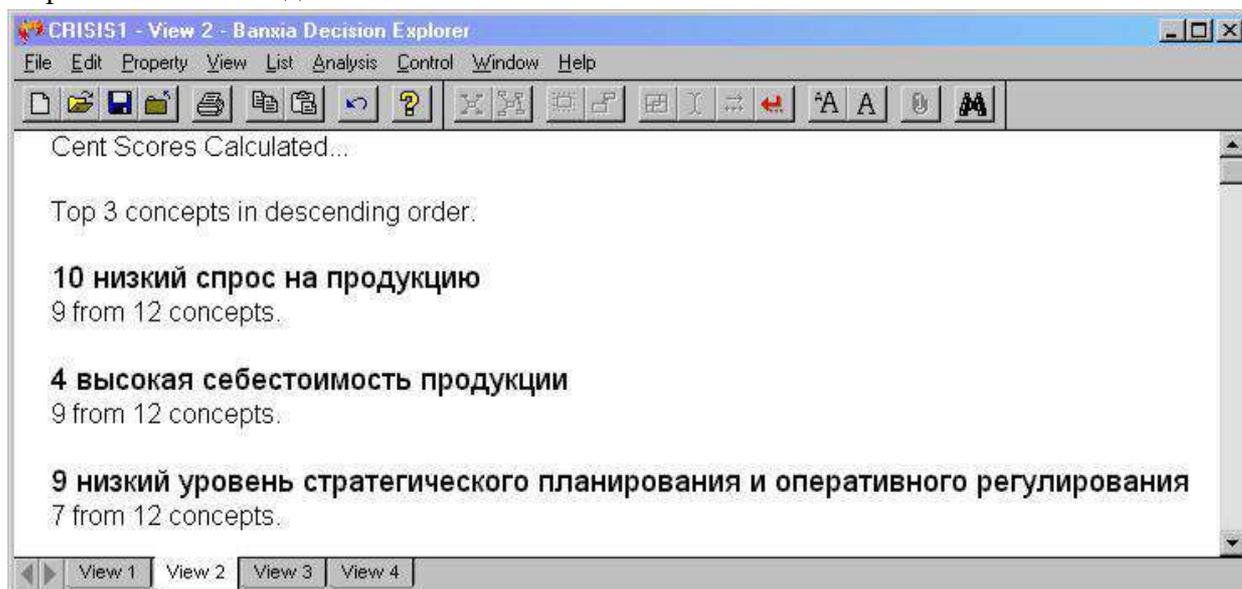


Рис. 4. Результат выполнения команды **Central 3**

Применение команды **Cluster** позволило выявить две подсистемы кризисообразующих факторов (рис. 5, 6), что привело к формированию на ОАО «Авангард» двух подсистем антикризисного управления:

23. подсистемы совершенствования менеджмента,
24. подсистемы совершенствования техники и технологии.

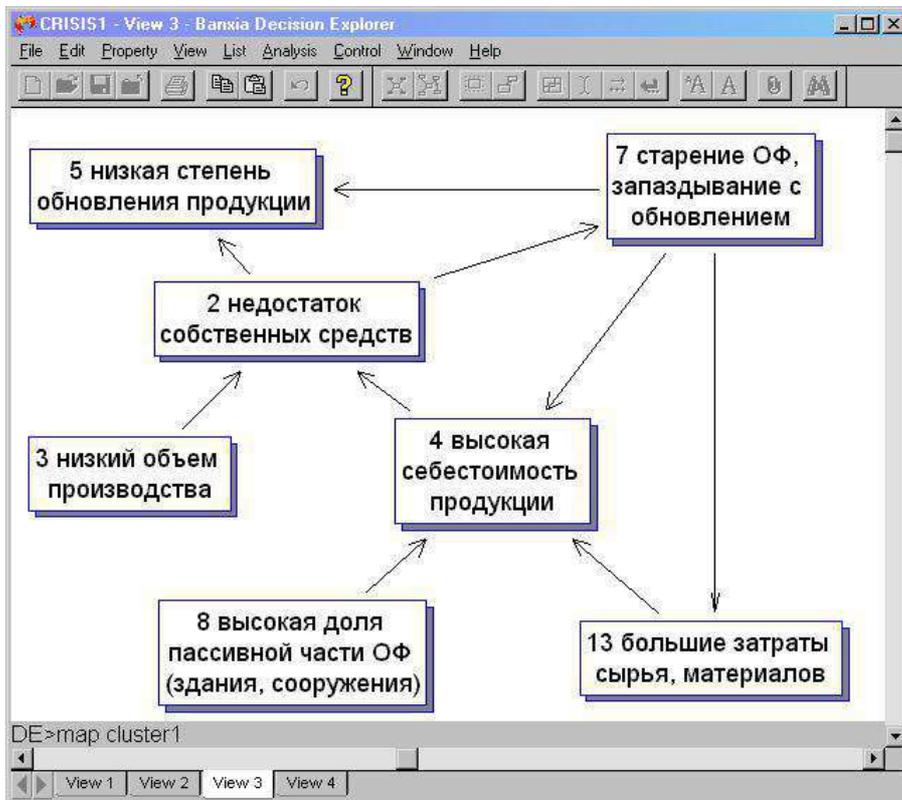


Рис. 5. Результат выполнения команды Cluster: подсистема кризисобразующих факторов в сфере техники и технологии



Рис. 6. Результат выполнения команды Cluster: подсистема кризисобразующих факторов в сфере менеджмента



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Команда **Loop** позволила выявить ряд петель обратной связи между кризисными факторами (рис. 7).



Рис. 7. Одна из петель мультипликатора, выявленная командой Loop

Для выяснения причин недостаточности собственных финансовых средств предприятия к элементу № 2 модели была применена команда **Map concept**, позволяющая увидеть иерархию «подчиненных» звеньев данного элемента модели (рис. 8). При этом вид и параметры иерархии были заданы в диалоговом окне **Mapping options** (рис. 9.)

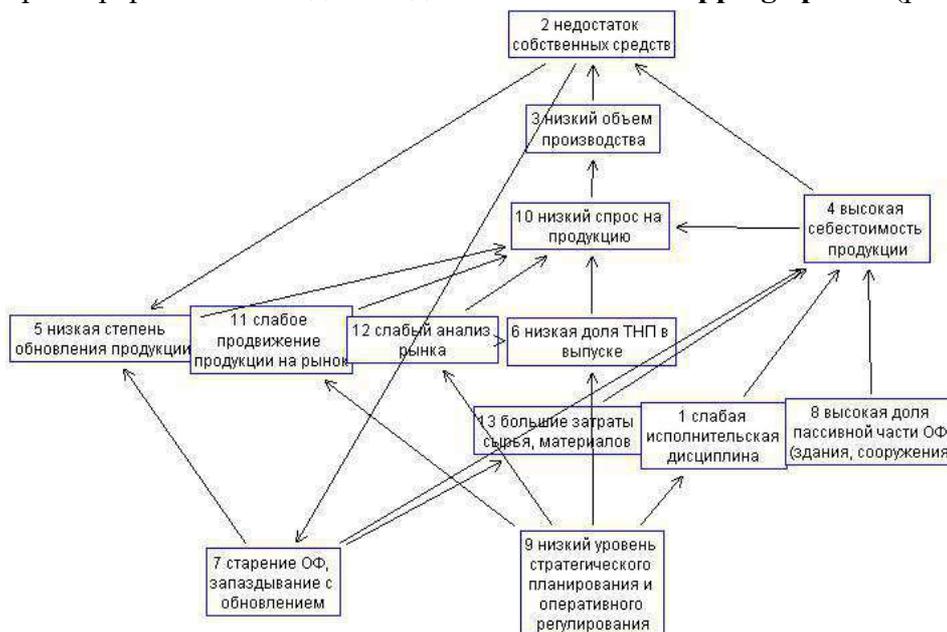


Рис. 8. Иерархия причин недостаточности собственных средств предприятия, выявленная с помощью команды Map concept

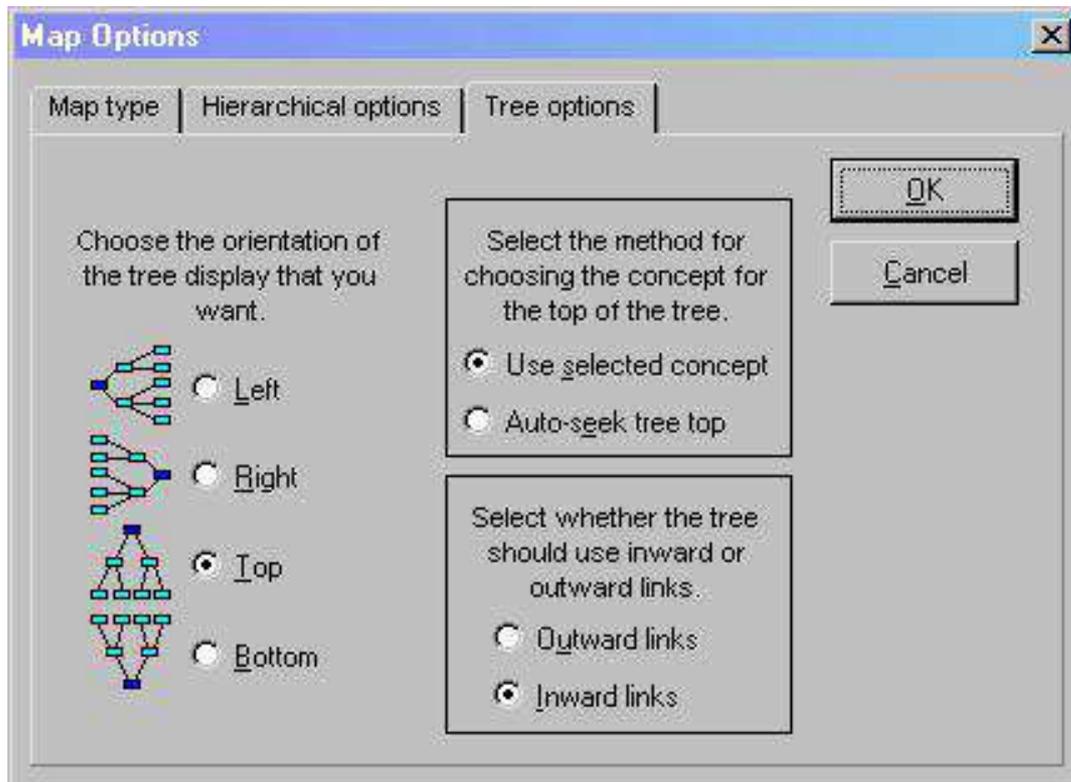


Рис. 9. Параметры иерархии элементов, выявляемой с помощью команды **Map concept**

Как видим, инструментарий *Decision Explorer* облегчил процесс структурирования проблемы на ОАО «Авангард», позволил сформировать относительно независимые подсистемы антикризисного управления, а также помог выявить первоочередные задачи, стоящие перед предприятием на текущем этапе развития.

### ЗАДАНИЕ

Выбрать для исследования некоторую проблемную ситуацию и разработать ее логически обоснованное решение.

Для достижения этой цели необходимо выполнить следующие задания:

25. Сформулировать постановку задачи.
26. В среде *Decision Explorer* построить карту проблемы, отражающую основные взаимосвязи причин и следствий проблемной ситуации.
27. С помощью инструментария *Decision Explorer* провести анализ карты проблемы, в том числе:
  - выявить два-три ключевых элемента модели;
  - построить иерархию причин для одного из элементов модели;
  - выявить петли положительной и отрицательной обратной связи;
  - сделать выводы о том, для каких целей может быть использована полученная информация.
28. Разработать альтернативные варианты решения проблемы (3 альтернативы).
29. Сформировать перечень критериев оценки альтернатив (3 шт.). Множество критериев должно быть согласовано с картой проблемы.
30. Дать субъективные оценки альтернатив по каждому из критериев по 10-бальной шкале.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

31. Для выявления наилучшего варианта решения проблемы рассчитать суммарные оценки каждой из альтернатив по всему множеству критериев.
32. Сформулировать итоговые выводы.
33. По результатам проделанной работы составить отчет.



### **ПРИМЕРЫ ПРОБЛЕМНЫХ СИТУАЦИЙ**

20. проблема низкого спроса на продукцию;
21. проблема отсутствия финансовых ресурсов предприятия;
22. проблема низкой эффективности руководства;
23. проблема слабой мотивации персонала;
24. проблема высокой себестоимости продукции;
25. проблема убыточности внешнеэкономической деятельности;
26. проблема дебиторской задолженности;
27. проблема отсутствия инноваций;
28. проблема низкой квалификации сотрудников.
29. высокая текучесть кадров
30. низкий уровень использования информационных технологий
31. отсутствие взаимодействия с учебными заведениями, выпускающими специалистов соответствующего профиля
32. высокая вероятность банкротства
33. низкая температура в помещениях зимой
34. воровство на предприятии
35. сезонные колебания спроса на продукцию
36. проблема брака
37. снижение доли рынка вследствие появления конкурента
38. сотрудники не склонны к самообразованию и обмену профессиональными знаниями между собой
39. высокие цены и низкое качество обслуживания в заводской столовой
40. низкая результативность инновационной деятельности вследствие отсутствия творческих идей о путях усовершенствования продукции
41. недостаток молодых кадров
42. отрицательный имидж предприятия у покупателей
43. предприятие находится далеко от маршрутов городского транспорта



## ДОПОЛНИТЕЛЬНЫЕ ИНСТРУКЦИИ К ВЫПОЛНЕНИЮ ЗАДАНИЯ ПО АНАЛИЗУ КАРТЫ ПРОБЛЕМЫ

34. Прежде всего, лист с исходной визуальной моделью следует защитить от изменений с помощью команды **View – Protect View**. Работу с командами, предназначенными для анализа карты проблемы, следует выполнять на пустых листах

35. Для построения иерархии причин для ключевого элемента модели необходимо выполнить следующие действия:

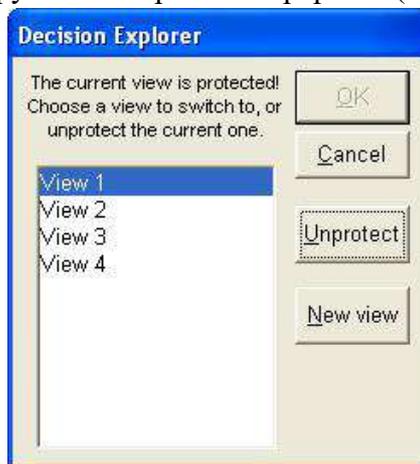
18. Выделить на карте проблемы тот ключевой элемент, для которого будет строиться иерархия причин, и вызвать команду **Analysis – Map concept**.

19. В появившемся диалоговом окне «**Map Options**» задать следующие параметры иерархии (если окно «**Map Options**» не появилось, его можно вызвать с помощью команды **Analysis – Mapping options**):

- на вкладке «**Map type**» задать тип иерархии – «**Tree**» («дерево»);
- на вкладке «**Tree options**» задать параметры так, как изображено на рис. 9 в методичке:
  - **inward links** (входящие связи)
  - **use selected concept** (в качестве вершины иерархии использовать выбранный элемент)
  - ориентацию дерева выбрать на свое усмотрение.

Затем нажать «**ОК**».

20. В появившемся диалоговом окне нужно выбрать тот пустой лист, на котором планируется отобразить иерархию (так как лист с исходной моделью защищен от изменений).



И После применения команды **loop** нужно отобразить визуальные модели найденных петель (как на рис. 7 в методичке) с помощью команд **map loop1**, **map loop2** ... и т.д. в зависимости от количества найденных петель.

## НЕКОТОРЫЕ ПОЛЕЗНЫЕ КОМАНДЫ DECISION EXPLORER

Команды, набираемые в командной строке

**cls** – очистить лист

**mapall** – отображение на листе всей визуальной модели (карты)

**map** ... – отображение на листе части визуальной модели, в т.ч.:



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

**map loop1** – отображение первой петли (эта команда доступна после выполнения команды loop)

Команды меню

**View – Protect View** – защита листа от изменений.



**View – Display scale – Select Zoom** – выбор масштаба изображения

### Контрольные вопросы

- Из каких стадий состоит процесс принятия решения по классическим критериям?
- Из каких стадий состоит процесс принятия решения по производным критериям?
- Из каких стадий состоит процесс принятия решения по гибкому критерию?
- В чем заключается принцип минимальной задержки?
- В чем сущность OL – стратегий?
- В чем сущность OL – стратегий?
- В чем разница между OL и CL стратегиями?
- Опишите применение принципа обратной связи.
- В чем сущность OLFB стратегии?
- В каком случае нет разницы между результатами применения OL и FB стратегиями?



## Практическое занятие №8 Многоцелевые решения

*Решение задач. Многомерные цели. Реализация целей. Выбор внутри эффективных множеств..*

где  $x = \{x_1, \dots, x_m\}$   
( $q, k=1$ ) вектор  
которое будет

### Теоретические сведения

Под ситуацией многоцелевых решений понимают пару  $\{x, F\}$ , множество решений субъектов управления,  $F = \{F^1, F^2, \dots, F^Q\} = \{f_k^q\}_{Q, m}$ , функционала оцениваний. Необходимо выбрать единственное решение, оптимальным по критерию свертки с учетом влияния факторов ( $v, w, u$ ).

· – метод нормализации;  $u$  – соотношение приоритета;  $w$  – критерий свертки.

**Метод нормализации** – это функция перехода  $F$ , как однозначного отображения  $R^Q$  в  $R^1$ , нормализация используется для перехода к сравнительным шкалам в значениях функционала оценивания.

**Метод приоритета** – вектор оценок ( $u_1, \dots, u_Q$ ) на компонентах  $F = \{F^1, F^2, \dots, F^Q\}$ .

**Критерий свертки** – принцип принятия оптимальных решений или функция отображения  $R^Q$  в  $R^1$ .

Таблица – Методы нормализации

Методы нормализации	Математическая запись
1. Замена ингредиентов	$(-f_k^q), \frac{1}{f_k^q}$
2. Относительная нормализация	$f_k^q / \max_k f_k^q; f_k^q / \min_k f_k^q$
3. Сравнительная нормализация	$f_k^q - \min_k f_k^q; \max_k f_k^q - f_k^q$
4. Естественная нормализация	$(f_k^q - \min_k f_k^q) / (\max_k f_k^q - \min_k f_k^q)$
5. Севиджа	$(\max_k f_k^q - f_k^q) / (\max_k f_k^q - \min_k f_k^q)$

Таблица – Принцип построения приоритетов

Принципы	Математическая запись
1. Линейный	$u_q f_k^q$
2. Показательный	$(f_k^q)^{u_q}$

Таблица – Критерий свертки ( $F^+$ )

Критерий свертки	Математическая запись
1. Гарантированный результат	$\min_q J_k^q$
2. Доминирующий результат	$\max_q f_k^q$
3. Равенство	$f_k^1 = f_k^2 = \dots = f_k^Q$
4. Суммарная эффективность	$f_k^q$
5. Равномерность	$f_k^q$



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

**Пример выполнения задания**

Имеем  $l=2$ ,  $X = x_1, \dots, x_5$ ,  $\begin{matrix} = \\ 1 \end{matrix} \begin{matrix} ' \\ 2 \end{matrix} \begin{matrix} ' \\ 3 \end{matrix}$ ,  $F = F^+$ ,  $F_1$  и  $F_2$  заданы в виде матриц:



$$F_1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{matrix} 2 & 4 & 1 \\ 4 & 3 & 4 \\ 6 & 7 & 5 \\ 10 & 10 & 8 \\ 2 & 10 & 9 \end{matrix} \end{matrix} \quad F_2 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{matrix} 0 & 12 & 7 \\ 4 & 0 & 8 \\ 7 & 3 & 9 \\ 4 & 2 & 5 \\ 10 & 1 & 2 \end{matrix} \end{matrix}$$

Принять решение в поле пятой информационной ситуации. Субъект управления задает приоритеты с такими весовыми коэффициентами:  $u_1 = 0.35$ ;  $u_2 = 0.65$ .  
Нормализация природная, критерий принятия решений – критерий Вальда, принцип учета приоритетов – показательный, свертка – гарантированный результат.

**Решение.** Для природной нормализации для матрицы  $F_1$  имеем:

– для состояния среды 1 :

$$\min_{k, k1} f_{k1}^1 = 2 \quad \max_{k, k1} f_{k1}^1 = 10 \quad \max_{k, k1} f_{k1}^1 - \min_{k, k1} f_{k1}^1 = 8$$

– для состояния среды 2 :

$$\min_{k, k2} f_{k2}^1 = 3 \quad \max_{k, k2} f_{k2}^1 = 10 \quad \max_{k, k2} f_{k2}^1 - \min_{k, k2} f_{k2}^1 = 7$$

– для состояния среды 3 :

$$\min_{k, k3} f_{k3}^1 = 1 \quad \max_{k, k3} f_{k3}^1 = 9 \quad \max_{k, k3} f_{k3}^1 - \min_{k, k3} f_{k3}^1 = 8$$

Отсюда имеем

$$F_1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{matrix} 0 & 1/7 & 0 \\ 2/8 & 0 & 3/8 \\ 4/8 & 4/7 & 4/8 \\ 1 & 1 & 7/8 \\ 0 & 1 & 1 \end{matrix} \end{matrix}$$

Для природной нормализации для матрицы  $F_2$  имеем:

– для состояния среды 1 :

$$\min_{k, k1} f_{k1}^2 = 0 \quad \max_{k, k1} f_{k1}^2 = 10 \quad \max_{k, k1} f_{k1}^2 - \min_{k, k1} f_{k1}^2 = 10$$

– для состояния среды 2 :

$$\min_{k, k2} f_{k2}^2 = 0 \quad \max_{k, k2} f_{k2}^2 = 12 \quad \max_{k, k2} f_{k2}^2 - \min_{k, k2} f_{k2}^2 = 12$$

– для состояния среды 3 :



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

$$\min_{k^2} f_{k^2} = 2 \quad \max_{k^2} f_{k^2} = 9 \quad \max_{k^2} f_{k^2} - \min_{k^2} f_{k^2} = 7$$

Отсюда имеем



$$F_2 = \begin{matrix} & & x_1^1 & x_2^2 & x_3^3 \\ x_1 & & 0 & 1 & 5/7 \\ x_2 & & 4/10 & 0 & 6/7 \\ x_3 & & 7/10 & 3/12 & 1 \\ x_4 & & 4/10 & 2/12 & 3/7 \\ x_5 & & 1 & 1/12 & 0 \end{matrix}$$

Используя показательный принцип учета приоритетов, получим:

$$\tilde{F}^1 = (f_k^1)_{k=1}^n, \quad \tilde{F}^2 = (f_k^2)_{k=1}^n,$$

то есть

$$\tilde{F}^1 = \begin{matrix} & & x_1^1 & x_2^2 & x_3^3 \\ x_1 & & 0 & 0 & 0 \\ x_2 & & 0.62 & 0.51 & 0.71 \\ x_3 & & 0.78 & 0.82 & 0.78 \\ x_4 & & 1 & 1 & 0.95 \\ x_5 & & 0 & 1 & 1 \end{matrix}, \quad \tilde{F}^2 = \begin{matrix} & & x_1^1 & x_2^2 & x_3^3 \\ x_1 & & 0 & 1 & 0.8 \\ x_2 & & 0.55 & 0 & 0.9 \\ x_3 & & 0.79 & 0.41 & 1 \\ x_4 & & 0.55 & 0.31 & 0.58 \\ x_5 & & 1 & 0.2 & 0 \end{matrix}$$

И

Используя гарантированный критерий свертки, получим общий функционал оценивания:

$$F = \begin{matrix} & & x_1^1 & x_2^2 & x_3^3 \\ x_1 & & 0 & 0 & 0 \\ x_2 & & 0.55 & 0.51 & 0.71 \\ x_3 & & 0.78 & 0.41 & 0.78 \\ x_4 & & 0.55 & 0.31 & 0.58 \\ x_5 & & 0 & 0.2 & 0 \end{matrix}$$

По критерию Вальда оптимальной является стратегия  $x^* = x_3$ , так как  $x^* = \max_k \min_j f_k^j = \max_k 0; 0.41; 0.31; 0 = 0.41$ .

### Задание.

Пусть субъект управления имеет  $Q(Q > 0)$  ситуаций принятия решений

$X_1, F_1, X_2, F_2, \dots, X_Q, F_Q$ , которые отличаются функционалом оценивания в заданной информационной ситуации  $I$ . Необходимо определить оптимальное решение для всех  $Q$  ситуаций принятия решений одновременно.  $X = x_1 \dots x_k$ , множество возможных

Использование основных факторов  $V, U, W$  принятия многоцелевых решений позволяет получить ситуацию принятия решений с одним скалярным функционалом оценивания для заданной информационной ситуации  $I$  и критерия принятия решений.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Заданы множество решений органа управления –  
ситуаций –  $\{x_1, \dots, x_q\}$ , тип функционала оценивания и информационная ситуация.



Определить в соответствии с исходными данными (таблица 1) оптимальное решение.

Функционалы оценивания строятся таким образом, чтобы не было идентичного повторения матриц, используя генератор случайных чисел в пределах заданного диапазона.

Использовать при определенном типе информационной ситуации:

-  $I_1$  – критерий Байеса (при  $q=2$   $p_1 = 0.45$  ; при  $q=3$   $p_2 = 0.25, p_1 = 0.35$  ; при  $q=4$   $p_2 = 0.2, p_1 = 0.25, p_3 = 0.3$  );

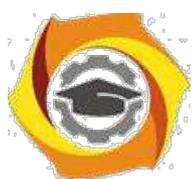
-  $I_4$  – критерий Лапласа;

-  $I_5$  – критерий Вальда;

-  $I_6$  – критерий Гурвица (  $= 0.6$  ).

Приоритет задается студентом самостоятельно с помощью соответствующих весовых коэффициентов.

Построить программный модуль для принятия многоцелевых решений, предусмотреть возможность ввода исходных данных пользователем, информативность алгоритма, вывод по результатам оценки альтернатив по критерию.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*Таблица 1 – Исходные данные*

№ варианта	$q = l$	$k$	Метод нормализации $\nu$	Соотношение приоритетов $u$	Критерий свертки $W$	Тип функционалов оценивания $F^+(F^-)$	Элементы матриц $f_k^q$		$I$
							$min$	$max$	
1	2	6	относительной нормализации	линейный	гарантированного результата	+	0	15	1
2	3	7	естественной нормализации	показательный	доминирующего результата	+	1	9	4
3	4	8	сравнительной нормализации	линейный	равномерности	-	0	6	5
4	2	5	Севиджа	показательный	суммарной эффективности	+	0	14	6
5	2	9	относительной нормализации	линейный	равномерности	-	0	12	5
6	4	6	естественной нормализации	показательный	гарантированного результата	-	0	16	6
7	4	7	сравнительной нормализации	линейный	доминирующего результата	-	1	9	4
8	3	7	Севиджа	показательный	равномерности	+	1	10	1
9	3	5	относительной нормализации	линейный	суммарной эффективности	+	1	15	1
10	2	6	естественной нормализации	показательный	равномерности	+	0	13	4
11	3	6	сравнительной нормализации	линейный	гарантированного результата	-	1	11	5
12	2	7	Севиджа	показательный	доминирующего результата	+	1	12	5
13	3	8	относительной нормализации	линейный	суммарной эффективности	-	1	8	4
14	4	9	естественной нормализации	показательный	суммарной эффективности	+	1	7	6
15	4	9	сравнительной нормализации	линейный	равномерности	-	0	5	6
16	4	8	Севиджа	показательный	гарантированного результата	+	0	6	6
17	2	8	Севиджа	линейный	доминирующего результата	-	0	7	1
18	3	7	относительной нормализации	показательный	суммарной эффективности	-	0	8	1
19	3	7	естественной нормализации	линейный	суммарной эффективности	-	0	10	4
20	2	6	сравнительной нормализации	показательный	равномерности	+	1	7	5
21	2	6	Севиджа	линейный	гарантированного результата	+	1	8	6
22	4	5	естественной нормализации	показательный	доминирующего результата	-	1	9	6
23	3	5	сравнительной нормализации	линейный	доминирующего результата	-	1	10	4
24	4	8	естественной нормализации	показательный	суммарной эффективности	+	1	10	5
25	2	6	относительной нормализации	показательный	равномерности	+	0	4	5

**Контрольные вопросы**

**В** В каких отношениях могут находиться между собой многомерные цели?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- В В каком случае можно использовать лексикографический метод принятия решения?
- В Как организовать процесс принятия решения в случае конкурирующих целей?
- В В каких связях могут находиться цели принятия решения?
- В Как организовать процесс принятия решения в случае дизъюнктивной связи целей?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

55

- в Как организовать процесс принятия решения в случае конъюнктивной связи целей?
- в Как организовать процесс принятия решения в случае смешанной связи целей?
- в Опишите свойства множества Парето.
- в Как используется теория нечётких множеств для поиска решения при многомерных целях?
- в Как устанавливаются характеристические множества?



## Практическое занятие № 9 Принятие решений с применением теории нечётких множеств

*Решение задач с применением нечётких множеств.*

### Теоретические сведения

теоретические сведения

Основой систем нечеткой логики является математическая теория нечетких множеств, которая берет свое начало со статьи американского ученого Лотфи Заде, опубликованной под названием "Fuzzy Sets" (нечеткие множества) в 1965 году в журнале Information and Control. В настоящее время эта теория получила достаточно глубокое развитие, а системы на базе нечеткой логики нашли широкое применение в промышленности от производства фотоаппаратов, стиральных машин, микроволновых печей до управления крупными промышленными производствами и систем поддержки принятия решения.

Идея, лежащая в основе теории нечетких множеств, заключается в том, что человек в своей повседневной жизни мыслит и принимает решения на основе нечетких понятий. Создание теории нечетких множеств - это попытка формализовать человеческий способ рассуждений. Развитие вычислительной техники позволяет в настоящее время создавать на базе теории нечетких множеств системы нечеткой логики, которые копируют способ рассуждений человека.

к большинству решаемых человеком задач не требуется высокая точность. Наоборот, часто приходится находить разумный компромисс между понятиями "точность" и "важность" при общении с реальным миром. Так, для принятия решения о переходе улицы человек не оценивает скорость приближающегося автомобиля с точностью до десятых долей метров в секунду. Он определяет для себя скорость автомобиля как "очень быструю", "быструю", "медленную" и т. д., т. е. использует для обозначения скорости лингвистические переменные.

*Определение лингвистической переменной (формальное)*

Лингвистической переменной называется пятерка  $(x, T(x), X, G, M)$ , где  $x$  - имя переменной;  $T(x)$  - множество имен лингвистических значений переменной  $x$ , каждое из которых является нечетким множеством на множестве  $X$ ;  $G$  есть синтаксическое правило для образования имен значений  $x$ ;  $M$  есть семантическое правило для ассоциирования каждой величины значения с ее понятием.

Это определение может вызвать ощущение, что лингвистическая переменная - очень сложное понятие, но на самом деле это не так. Цель концепции лингвистической переменной состоит в том, чтобы формальным образом сказать, что переменная может принимать в качестве значений слова из естественного языка. Например, если мы говорим "быстрая скорость", то переменная "скорость" должна пониматься как лингвистическая переменная, но это не означает, что переменная "скорость" не может принимать реальные значения.

*Определение лингвистической переменной (интуитивное)*

Если переменная может принимать значения слов в естественном языке (например, "маленький", "быстрый" и т. п.), то эта переменная определяется как лингвистическая переменная. Слова, значения которых принимает лингвистическая переменная, обычно обозначают собой нечеткие множества. Лингвистическая переменная может принимать своими значениями либо слова, либо числа.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Определение. Нечеткое множество - это такое множество, которое образуется путем введения обобщенного понятия принадлежности, т. е. расширения двухэлементного



множества значений функции принадлежности  $\{0,1\}$  до отрезка  $[0,1]$ . Это означает, что переход от полной принадлежности объекта множеству к его полной непринадлежности происходит не скачком, как в обычных "четких" множествах, а плавно, постепенно, причем степень принадлежности элемента множеству выражается числом из интервала  $[0,1]$ .

Таким образом, нечеткое множество  $A = \{(x, \mu_A(x))\}$  определяется математически как совокупность упорядоченных пар, составленных из элементов  $x$  множества  $X$  и соответствующих им степеней принадлежности  $\mu_A(x)$  или непосредственно в виде функции  $\mu_A: X \rightarrow [0,1]$ . Рассмотрим пример нечеткого множества.

*Пример 1.* Множество высоких людей

Пусть  $x$  есть лингвистическая переменная, обозначающая "рост человека", а ее функция принадлежности к множеству высоких людей  $\mu_A: X \rightarrow \{0,1\}$ , где  $X$  - множество, включающее в себя все возможные значения роста человека, задана следующим образом:

$$\mu_A(x) = \begin{cases} 1, & \text{если } x \geq 180, \\ 0, & \text{если } x < 180. \end{cases}$$

Тогда множество "высоких людей" задается выражением  $A = \{x | \mu_A(x) = 1\}$ ,  $x \in X$ . Графически это представлено на рисунке 1 (сплошная линия).

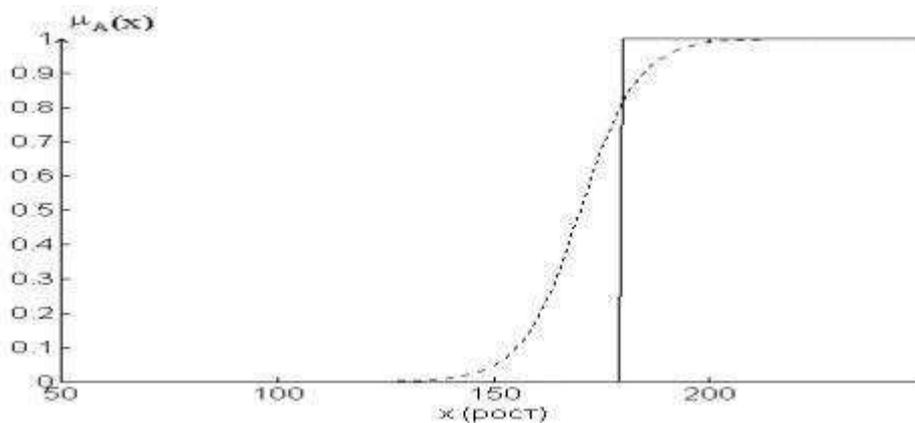


Рисунок 1

Однако в реальной жизни такое понятие, как "рост высокого человека", является субъективным, т. е. зависит от индивидуума, делающего оценку. Одни считают, что высокий человек должен быть ростом более 170 см, другие - более 180 см, третьи - более 190 см. Нечеткие множества позволяют учесть такую размытость оценок. Так, пусть функция принадлежности  $\mu_A: X \rightarrow \{0,1\}$  имеет вид, представленный на рисунке 1 пунктирной линией. Тогда множество  $A$  ("высоких людей") задается множеством пар  $A = \{x, \mu_A(x)\}$ ,  $x \in X$  и является нечетким множеством.

Таким образом, человек ростом 145 см будет принадлежать множеству  $A$  со степенью принадлежности  $\mu_A(145) = 0$ , человек ростом 165 см -  $\mu_A(165) = 0,3$ , ростом 185 см -  $\mu_A(185) = 0,9$ , а ростом 205 см -  $\mu_A(205) = 1$ .

Можно сказать, что в нечеткой логике правдивость каждого утверждения рассматривается в рамках степени правдивости, которая может также рассматриваться как степень уверенности эксперта, делающего оценку, в том, что элемент  $x$  принадлежит множеству



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Из вышесказанного можно сделать следующие выводы:

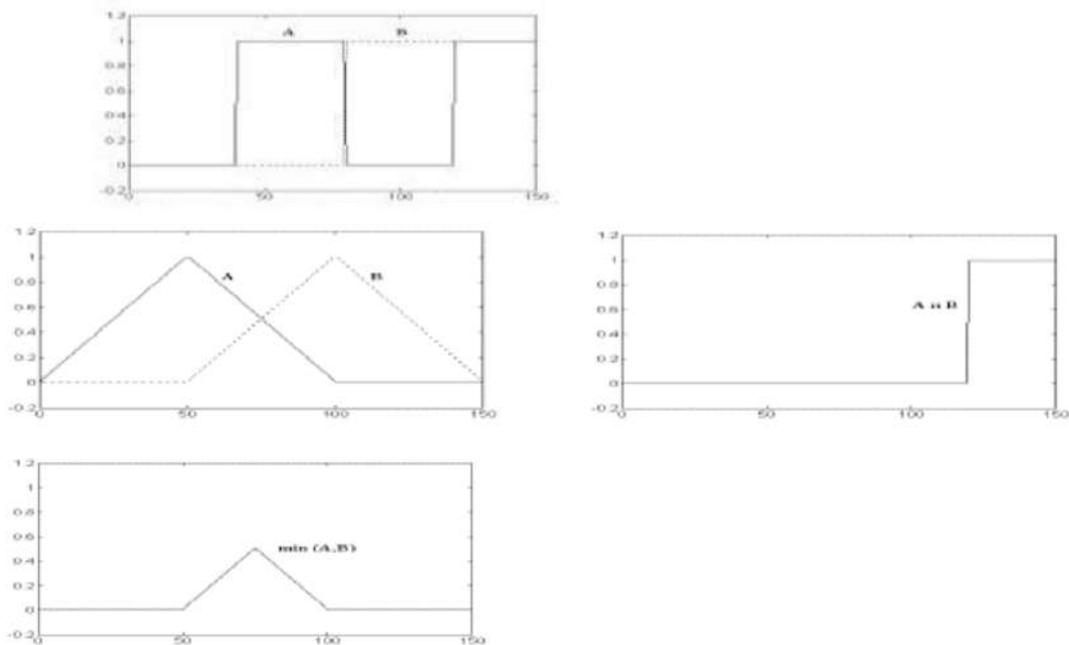
О нечеткие множества описывают неопределенные понятия (быстрый бегун, горячая вода, жаркая погода);



в нечеткие множества допускают возможность частичной принадлежности к ним (пятница - частично выходной день (укороченный), погода скорее жаркая);

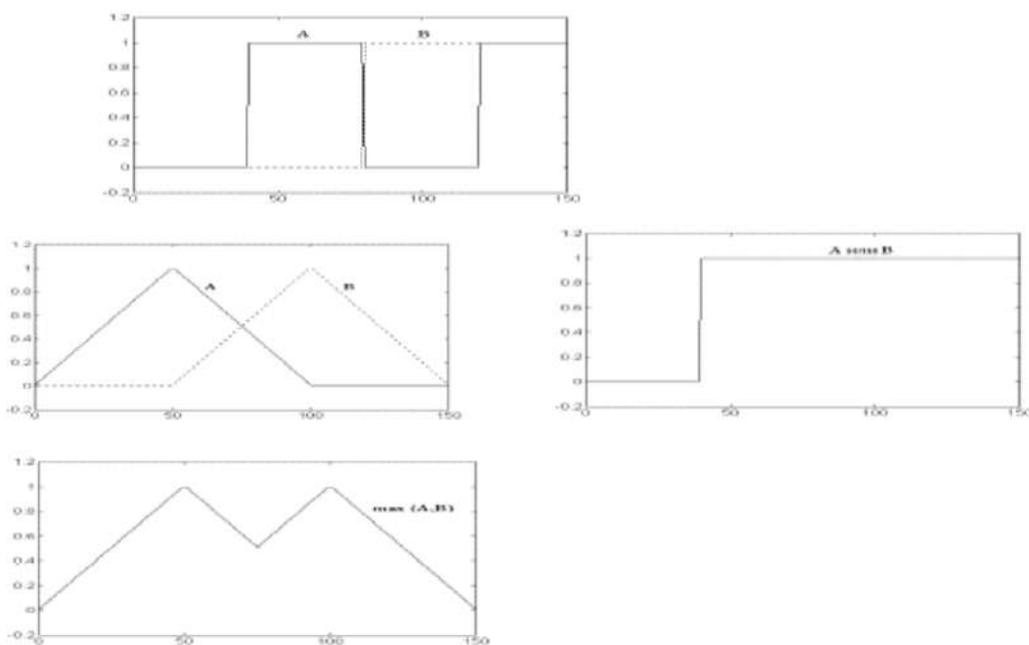
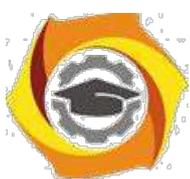
в степень принадлежности объекта к нечеткому множеству определяется соответствующим значением функции принадлежности на интервале  $[0,1]$  (пятница принадлежит к выходным дням со степенью принадлежности 0,3);  
В функция принадлежности ставит в соответствие объекту (или логической переменной) значение степени его принадлежности к нечеткому множеству.

*Логические операции с нечеткими множествами.* Одной из важных вещей для понимания механизма нечеткой логики является то, что она может рассматриваться как расширение стандартной, хорошо известной булевой логики (подобно тому, как нечеткие множества являются расширением обычных множеств). Другими словами, если рассматривать нечеткие значения в их экстремумах, т. е. в 1 (абсолютная правда) и в 0 (абсолютная ложь), то можно применять стандартные логические операции: И, ИЛИ, НЕ. В нечеткой логике истинность каждого утверждения рассматривается в определенной степени. В нечеткой логике по аналогии с операциями И и ИЛИ вводятся операции  $\min(A, B)$  и  $\max(A, B)$ , а операции НЕ соответствует операция  $1-A$ . Графически это представлено на рисунках 2 - 4.



а	б
---	---

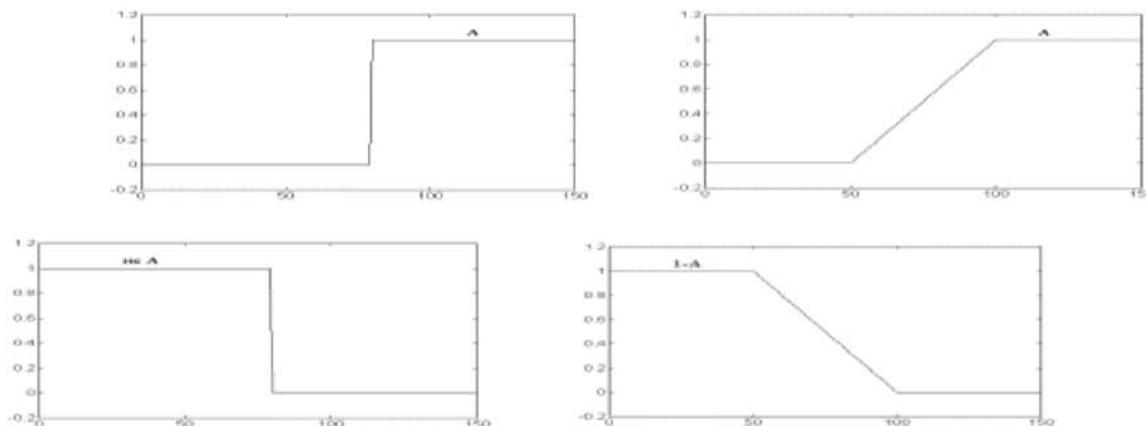
Рисунок 2 - Операция  $\min(A, B)$ : а - "четкая" логика; б - "нечеткая" логика



а

б

Рисунок 3 - Операция  $\max(A, B)$ : а - "четкая" логика; б - "нечеткая" логика



а

б

Рисунок 4 - Операция  $1-A$ : а - "четкая" логика; б - "нечеткая" логика

Рассмотрим формальные определения.

**Определение 1.** Пересечением  $A \cap B$  двух нечетких множеств  $A$  и  $B$  на множестве  $X$  является нечеткое множество на множестве  $X$  с функцией принадлежности, определяемой

для любого  $x \in X$  как  $\mu_{A \cap B} = \min\{\mu_A(x), \mu_B(x)\}$ .

**Определение 2.** Объединением  $A \cup B$  двух нечетких множеств  $A$  и  $B$  на множестве  $X$  является нечеткое множество на множестве  $X$  с функцией принадлежности, определяемой

для любого  $x \in X$  как  $\mu_{A \cup B} = \max\{\mu_A(x), \mu_B(x)\}$ .

**Определение 3.** Дополнением  $\bar{A}$  нечеткого множества  $A$  на множестве  $X$  является нечеткое множество на множестве  $X$  с функцией принадлежности, определяемой для любого  $x \in X$  как

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Определения 1 и 2 представляют собой один из возможных выборов оператора пересечения  
13) объединения нечетких множеств. Выбор оператора ведет к определенной интерпретации понятий пересечение и объединение. В литературе вводится понятие треугольных норм и треугольных конорм для обозначения операторов пересечения и объединения нечетких



множеств.

*Определение 4.* Треугольной нормой (t-нормой) называется двухместная действительная функция  $T: [0,1] \times [0,1] \rightarrow [0,1]$ , удовлетворяющая следующим условиям:

- 1)  $T(0, 0) = 0$ ;  $T(\mu_A, 1) = T(1, \mu_A) = \mu_A$  (ограниченность);
  - 2)  $T(\mu_A, \mu_B) \leq T(\mu_C, \mu_D)$ , если  $\mu_A \leq \mu_C, \mu_B \leq \mu_D$  (монотонность);
  - 3)  $T(\mu_A, \mu_B) = T(\mu_B, \mu_A)$  (коммутативность);
- к  $T(\mu_A, T(\mu_B, \mu_C)) = T(T(\mu_A, \mu_B), \mu_C)$  (ассоциативность). Простыми случаями t-норм являются операции пересечения, алгебраического произведения (пересечение II), ограниченного произведения (пересечение III), сильного произведения

(пересечение IV). *Определение 5.* Алгебраическим произведением  $A * B$  двух нечетких множеств  $A$  и  $B$  на множестве  $X$  является нечеткое множество на множестве  $X$  с функцией принадлежности,

определяемой для любого  $x \in X$  как  $A * B(x) = \mu_A(x) \cdot \mu_B(x)$ .

*Определение 6.* Ограниченным произведением  $A \overset{\circ}{*} B$  двух нечетких множеств  $A$  и  $B$  на множестве  $X$  является нечеткое множество на множестве  $X$  с функцией принадлежности, определяемой для любого  $x \in X$  как

$$\mu_{A \overset{\circ}{*} B}(x) = \max(0, \mu_A(x) + \mu_B(x) - 1).$$

*Определение 7.* Сильным произведением  $A \overset{*}{*} B$  двух нечетких множеств  $A$  и  $B$  на множестве  $X$  является нечеткое множество на множестве  $X$  с функцией принадлежности, определяемой для любого  $x \in X$  как

$$\mu_{A \overset{*}{*} B}(x) = \begin{cases} \mu_A(x), & \text{если } \mu_B(x) = 1; \\ \mu_B(x), & \text{если } \mu_A(x) = 1; \\ 0 & \text{— в других случаях} \end{cases}$$

Для этих t-норм справедливо неравенство

$$A \cdot B \subseteq A \overset{\circ}{*} B \subseteq A \overset{*}{*} B \subseteq A \cup B.$$

*Определение 8.* Треугольной конормой (t-конормой) называется двухместная действительная функция

$\perp: [0,1] \times [0,1] \rightarrow [0,1]$ , удовлетворяющая следующим условиям:

- 1)  $\perp(1, 1) = 1$ ;  $\perp(0, \mu_A) = \perp(\mu_A, 0) = \mu_A$  (ограниченность);
- 2)  $\perp(\mu_A, \mu_B) \geq \perp(\mu_C, \mu_D)$ , если  $\mu_A \geq \mu_C, \mu_B \geq \mu_D$  (монотонность);
- 3)  $\perp(\mu_A, \mu_B) = \perp(\mu_B, \mu_A)$  (коммутативность);
- 4)  $\perp(\mu_A, \perp(\mu_B, \mu_C)) = \perp(\perp(\mu_A, \mu_B), \mu_C)$  (ассоциативность).

Простыми случаями t-конорм являются операции объединения, алгебраической суммы (объединение II), ограниченной суммы (объединение III) и сильной суммы (объединение IV).

*Определение 9.* Алгебраической суммой  $A \oplus B$  двух нечетких множеств  $A$  и  $B$  на множестве  $X$  является нечеткое множество на множестве  $X$  с функцией принадлежности, определяемой

для любого  $x \in X$  как  $A \oplus B(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*Определение 10.* Ограниченной суммой  $A \dot{+} B$  двух нечетких множеств  $A$  и  $B$  на множестве  $X$  является нечеткое множество на множестве  $X$  с функцией принадлежности, определяемой



для любого  $x \in X$  как

$$\mu_{A+B}(x) = \min(1, \mu_A(x) + \mu_B(x)).$$

*Определение 11.* Сильной суммой  $A \hat{+} B$  двух нечетких множеств  $A$  и  $B$  на множестве  $X$  является нечеткое множество на множестве  $X$  с функцией принадлежности, определяемой для любого  $x \in X$  как

$$\mu_{A \hat{+} B}(x) = \begin{cases} \mu_A(x), & \text{если } \mu_B(x) = 0; \\ \mu_B(x), & \text{если } \mu_A(x) = 0; \\ 1 & - \text{ в других случаях.} \end{cases}$$

Для этих t-конорм справедливо неравенство

$$A \hat{+} B \geq A + B \geq A \oplus B \geq A \cup B.$$

Следует отметить, что в теории нечетких множеств оператор дополнения также не является единственным. Помимо оператора  $1-A$  существует целый набор операторов отрицания.

Наиболее общее определение функции отрицания в нечетких множествах  $C: [0,1] \rightarrow [0,1]$  предполагает, что выполняются, по крайней мере, два следующих свойства:

1)  $C(0)=1,$   $C(1)=0;$

2)  $C$  - невозрастающая функция, т. е. если  $\mu_A \leq \mu_B$ , то  $C(\mu_A) \geq C(\mu_B)$ . *Определение 12.* Разностью  $A-B$  двух нечетких множеств  $A$  и  $B$  на множестве  $X$  называется нечеткое множество на множестве  $X$  с функцией принадлежности, определяемой для любого  $x \in X$  как

*Определение 13.* Концентрированием нечеткого множества  $A^2$  на множестве  $X$  называется

нечеткое множество на множестве  $X$  с функцией принадлежности следующего вида:

$$\mu_{A^2}(x) = \mu_A^2(x) \quad \text{для} \quad \text{любого} \quad x \in X.$$

Результатом применения операции концентрирования к нечеткому множеству  $A$  является уменьшение степени принадлежности элементов к этому множеству, и происходит оно в

квадратичной зависимости, т. е. если  $\mu_A(x) \approx 1$ , то это уменьшение мало, а если  $\mu_A(x) \approx 0$ , то уменьшение велико.

В естественном языке применение операции концентрирования к значению лингвистической

переменной соответствует использованию усиления "очень".

*Определение 14.* Растяжением нечеткого множества  $A^{1/2}$  называется операция, противоположная концентрированию. В естественном языке ее можно сравнить с использованием слов "достаточно" или "более-менее" ("более-менее близкий"). *Системы*



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*нечеткой логики.* Системами нечеткой логики называются системы, которые оперируют с нечеткими понятиями, такими как нечеткие множества, лингвистические переменные и т. п., и используют при этом нечеткую логику. Системы нечеткой логики

могут быть классифицированы по трем основным типам:

- 1) простые системы нечеткой логики (pure Fuzzy Logic Systems);
- 2) нечеткие системы Такаги и Суджено (Takagi and Sugeno);
- 3) системы нечеткой логики с фаззификатором и дефаззификатором.

*Простые системы нечеткой логики.* Базовая конфигурация простой системы нечеткой логики представлена на рисунке 5. Базис нечетких правил содержит набор нечетких IF-THEN (ЕСЛИ-ТО) правил, а механизм нечеткого вывода на основе принципов нечеткой логики использует эти IF-THEN-правила



для отображения нечетких множеств из входящего множества высказываний  $X$  в нечеткие множества из множества высказываний  $Y$  на выходе системы. Нечеткие IF-THEN-правила выглядят следующим образом:  
 $R(p): \text{IF } x_1 \text{ есть } F_{1p} \text{ И } \dots \text{ И } x_n \text{ есть } F_{np} \text{ THEN } y \text{ есть } G_p, (1)$   
где  $F_{1p}$  и  $G_p$  - нечеткие множества;  $X = (X_1, \dots, X_n)^T$  и  $Y$  - входные и выходная

лингвистические переменные соответственно;  $p = \overline{1, m}$

### Как заставить SciLAB 4.1.2 работать?

Для работы с fuzzy logic нужно загрузить модуль *SciFLT* в среду SciLAB. Сделать это можно открыв файл loader.sce из этого пакета и выполнив команду *execute-> load into SciLAB*.

**Используемые в системе SciLAB типы функций принадлежности** Инструментарий нечеткой логики (ИНЛ) в составе пакета SciFLT содержит 11 встроенных типов функций принадлежности (ФП), формируемых на основе кусочно-линейных функций, распределения Гаусса, сигмоидной кривой, квадратичных и кубических полиномиальных кривых.

□ наиболее простым ФП (и в этом их главное достоинство) можно отнести треугольную в трапецевидную. Наименование треугольной ФП - *trimf* (*triangle membership function*). В параметрическом виде она представляет собой ни что иное, как набор трех точек, образующих треугольник.

Описание функции:

$$y = \text{trimf}(x, [a \ b \ c]).$$

Вектор  $x$  обозначает базовое множество, на котором определяется ФП.

Величины  $a$  и  $c$  задают основание треугольника,  $b$  - его вершину.

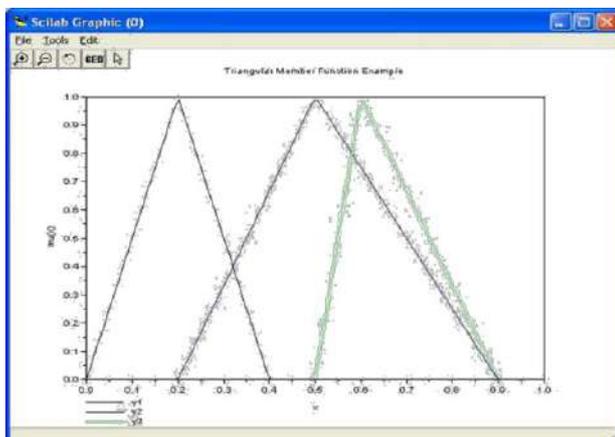


Рис. 1.1. Треугольные функции

Далее рассмотрим примеры использования различных ФП в системе *SciFLT*. Примеры представляют собой фрагменты программ и комментариев на языке пакета *SCILAB*.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Пример 1.1. Программа использования ФП `trimf` (результат на рис.1.1)  
`x=linspace(0,1,100)'; //задаем множество x`  
`y1=trimf(x,[0 0.2 0.4]); //и три треугольных функции на нем`



```
y2=trimf(x,[0.2 0.5 0.9]);
y3=trimf(x,[0.5 0.6 0.9]);
xbasc();
```

```
//строим функции принадлежности plot2d(x,[y1 y2 y3],leg="y1@y2@y3");
//подписываем рисунок
```

```
xtitle("Triangular Member Function Example","x","mu(x)");
```

Трапециевидная ФП, *trapmf* (*trapezoid membership function*), отличается от предыдущей функции лишь тем, что имеет верхнее основание.

Описание функции:

$$y = \text{trapmf}(x, [a \ b \ c \ d]),$$

где параметры  $a$  и  $d$  задают нижнее основание,  $b$  и  $c$  – верхнее основание трапеции (рис. 1.2).

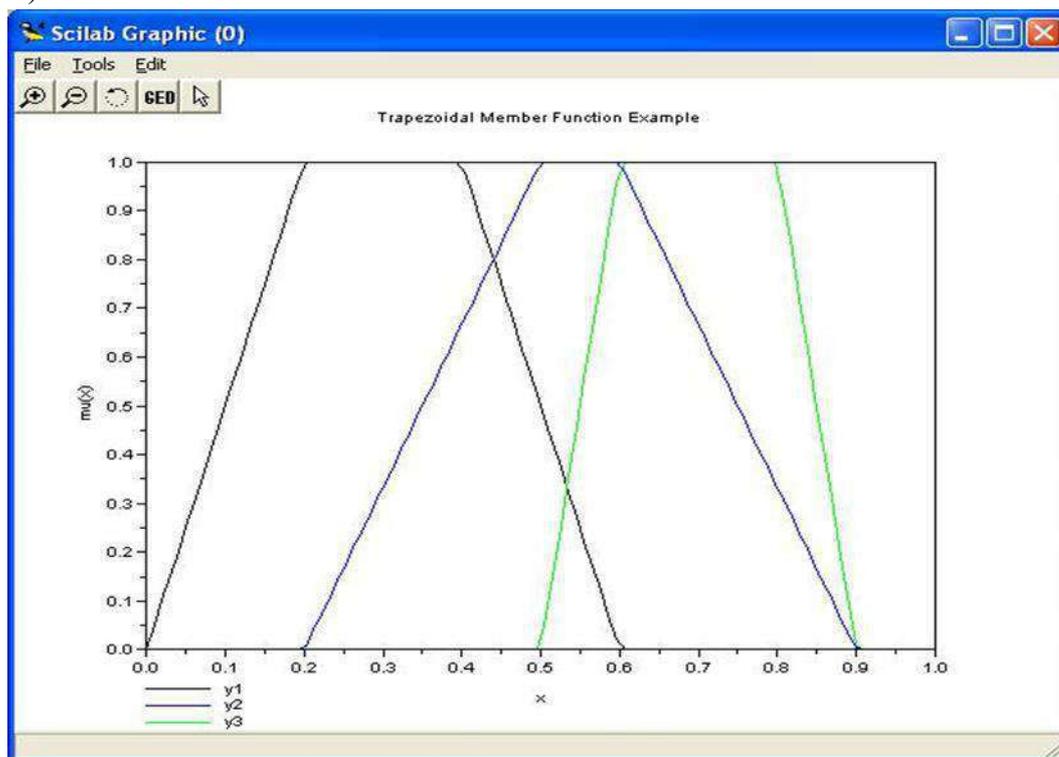


Рис.1.2. Трапециевидные функции принадлежности

Пример 1.2. Программа использования ФП *trapmf* (результат на рис. 1.2)

```
x=linspace(0,1,100)';
y1=trapmf(x,[0 0.2 0.4 0.6]);
y2=trapmf(x,[0.2 0.5 0.6 0.9]);
y3=trapmf(x,[0.5 0.6 0.8 0.9]);
xbasc(); plot2d(x,[y1 y2 y3],leg="y1@y2@y3"); xtitle("Trapezoidal Member Function
Example","x","mu(x)");
```



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

На основе функции распределения Гаусса в *SCIFLT* можно построить ФП двух видов: простую ФП Гаусса и двухстороннюю ФП, образованную с помощью различных функций распределения Гаусса. Первая из них обозначается – *gaussmf*, а вторая – *gauss2mf*.



Описание функции:

$$y = \text{gaussmf}(x, [a, b, c]).$$

Последнее выражение является комбинацией двух различных функций распределения Гаусса.

На рис. 1.3. представлены графики кривых *gaussmf* и *gauss2mf*, заданных в программе из примера 1.3.

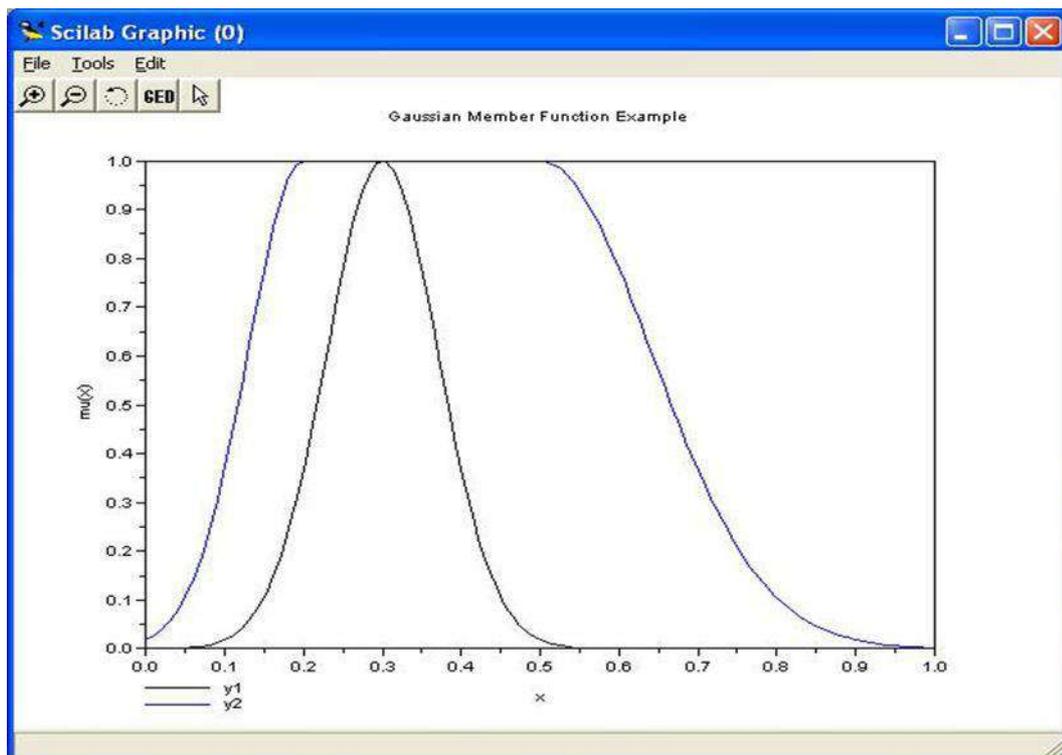


Рис. 1.3. Гауссовы кривые

Пример 1.3. Программа использования ФП *gaussmf*

```
x=linspace(0,1,100);
y1=gaussmf(x,[0.3 0.1]);
y2=gauss2mf(x,[0.2 0.1 0.5 0.2]);
xbaso();
plot2d(x,[y1 y2],leg="y1@y2");
```

`xtitle("Gaussian Member Function Example","x","mu(x)");`Символ “ ’ ” в строке определения базового множества *x* показывает транспонированность базового множества.

Следующей функцией, которая позволяет представлять нечеткие субъективные предпочтения, является ФП «обобщенный колокол». Она обозначается *gbellmf* (*bell shape membership function*). Ее отличие от рассмотренных ранее ФП заключается в добавлении



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

третьего параметра, что позволяет осуществлять плавный переход от нечеткого множества к четкому.



Описание функции:

$$y = gbellmf(x, [a \ b \ c]).$$

где  $c$  определяет расположение центра ФП,  $a$  и  $b$  оказывают влияние на форму кривой (рис. 1.4).

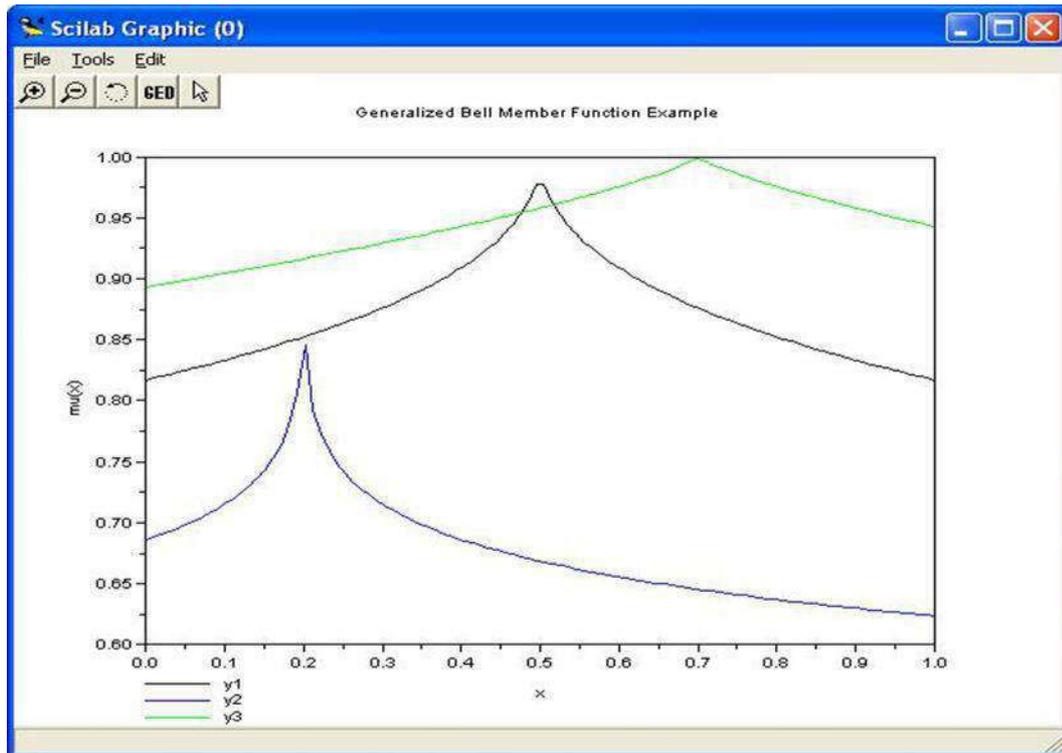


Рис. 1.4. Колоколообразные функции принадлежности

Пример 1.4. Программа использования `gbellmf`

```
x=linspace(0,1,100);
y1=gbellmf(x,[0.5 10 0.5]);
y2=gbellmf(x,[0.2 10 0.2]);
y3=gbellmf(x,[0.7 10 0.8]);
```

```
xbasc();
plot2d(x,[y1 y2 y3],leg="y1@y2@y3");
```

```
xtitle("Generalized Bell Member Function Example","x","mu(x)");
```

ФП на основе функции распределения Гаусса и ФП «обобщенный колокол» отличаются гладкостью и простотой записи и являются наиболее используемыми при описании нечетких множеств.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Но несмотря на то , что гауссовы и колоколообразные ФП обладают свойством гладкости, они не позволяют формировать асимметричные ФП. Для этих целей в *SciFLT*



существует набор сигмоидных функций, которые могут быть открыты либо слева, либо справа в зависимости от типа функции. Симметричные и закрытые функции синтезируются с использованием двух дополнительных сигмоид. Основная сигмоидная ФП обозначается *sigmf*, а дополнительные – *dsigmf* и *psigmf*.

Описание основной сигмоидной функции:

$$y = \text{sigmf}(x, [a \ c]).$$

24. зависимости от знака параметра *a* рассматриваемая ФП будет открыта справа или слева (рис. 1.5), что позволит применять ее при описании таких нечетких понятий, как «очень большой», «крайне отрицательно» и т.д.

Описание дополнительной сигмоидной функции:

$$y = \text{dsigmf}(x, [a_1 \ c_1 \ a_2 \ c_2]).$$

ФП *dsigmf* зависит от четырех параметров  $a_1, c_1, a_2$  и  $c_2$  и определяется

как разность двух сигмоидных функций:  $f_1(x, a_1, c_1) - f_2(x, a_2, c_2)$ .

Описание дополнительной сигмоидной функции:

$$y = \text{psigmf}(x, [a_1 \ c_1 \ a_2 \ c_2]).$$

ФП *psigmf* также как и предыдущая функция зависит от четырех параметров  $a_1, c_1, a_2$  и  $c_2$  и определяется как произведение двух сигмоидных функций:  $f_1(x, a_1, c_1) \cdot f_2(x, a_2, c_2)$ .

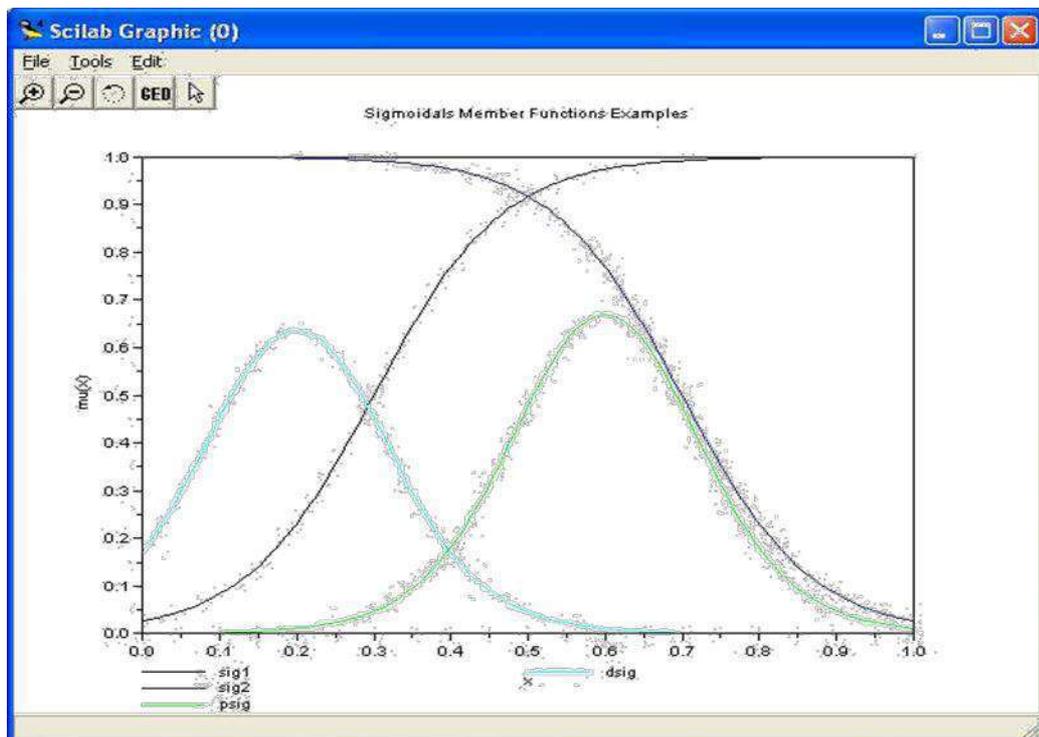


Рис. 1.5. Сигмоидные функции принадлежности



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Пример 1.5. Программа использования сигмоидных функций



```
x=linspace(0,1,100);
sig1=sigmf(x,[12 0.3]);
sig2=sigmf(x,[-12 0.7]);
```

```
psig=psigmf(x,[15 0.5 -15 0.7]);
dsig=dsigmf(x,[15 0.1 15 0.3]);
xbasc();
```

```
plot2d(x,[sig1 sig2 psig dsig],leg="sig1@sig2@psig@dsig"); xtitle("Sigmoidals Member
Functions Examples", "x", "mu(x)");
```

Инструментарий нечеткой логики *SciFLT* предоставляет возможность формирования ФП на основе полиномиальных кривых. Соответствующие функции называются *Z*-функции (*zmf*), *PI*-функции (*pimf*) и *S*-функции (*smf*). Функция *zmf* представляет собой асимметричную полиномиальную кривую, открытую слева, функция *smf* – зеркальное отображение функции *zmf*. Функция *pimf* равна нулю в правом и левом пределах и принимает значение, равное единице в середине некоторого отрезка (рис. 1.6).

Описание функции:

$$17. \quad = zmf(x, [a \ b]).$$

Параметры  $a$  и  $b$  определяют экстремальные значения кривой.

Описание функции:

$$y = pimf(x, [a \ b \ c \ d]).$$

Параметры  $a$  и  $d$  задают переход функции в нулевое значение, параметры  $b$  и  $c$  – в единичное.

Описание функции:

$$y = smf(x, [a \ b]).$$

Параметры  $a$  и  $b$  определяют экстремальные значения кривой.

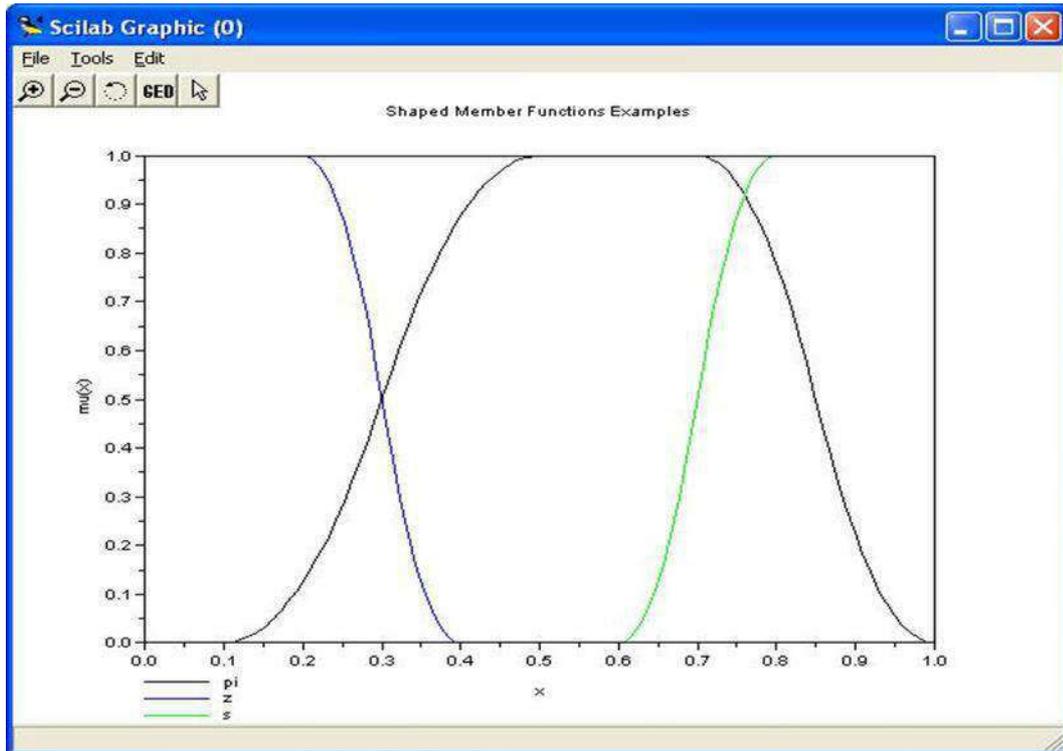


Рис. 1.6. Полиномиальные кривые

```
Пример 1.6. Программа использования полиномиальных кривых
x=linspace(0,1,100)';
pi=pimf(x,[0.1 0.5 0.7 1.0]);
z=zmf(x,[0.2 0.4]);
s=smf(x,[0.6 0.8]);
xbasc();
plot2d(x,[pi z s],leg="pi@z@s");
```

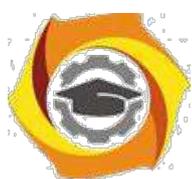
```
xtitle("Shaped Member Functions Examples","x","mu(x)");
```

Помимо рассмотренных выше функций, позволяющих представлять нечеткие множества, в *SciFLT* можно формировать собственные ФП или модифицировать встроенные.

### ***Операции с нечеткими множествами в модуле SciFLT***

Выделяют три основные логические операции с нечеткими множествами: конъюнкцию, дизъюнкцию и логическое отрицание.

модуле *SciFLT* существует возможность определять конъюнктивные и дизъюнктивные операторы различными методами.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Определение функции:



$y = \text{tnorm}(x, \text{class} [\text{class\_par}])$ ,  
 $x$  – матрица, размерностью  $[m,n]$ .  
 $y$  – матрица, размерностью  $[m,1]$ .

**class** – строка, задает вид T-нормы. Может принимать следующие значения: «**dubois**» для T-норма Дюбуа- Прада, «**yager**» для T-норм Ягера, «**dprod**» для вероятностного И, «**eprod**» для произведения Энштейна, «**aproduct**» для алгебраического произведения и «**min**» для операции нахождения минимума.

**class\_par** – скалярная величина, которая используется в T-нормах «**dubois**» и «**yager**».

Определение функции:

$y = \text{snorm}(x, \text{class} [\text{class\_par}])$

20. – матрица, размерностью  $[m,n]$ .

21. – матрица, размерностью  $[m,1]$ .

**class** – строка, задает вид S-конормы. Может принимать следующие значения: «**dubois**» для S- конормы Дюбуа-Прада, «**yager**» для S-конормы Ягера, «**dsum**» для вероятностного ИЛИ, «**esum**» для суммы Энштейна, «**asum**» для алгебраической суммы и «**max**» для операции нахождения максимума.

**class\_par** – скалярная величина, которая используется в S-конормах «**dubois**» и «**yager**»

Пример 1.7. Программа использования операций **min** и **max**

```
x=[0:0.1:10];
y1=gaussmf(x,[3 1.2]);
```

```
y2=gaussmf(x,[7 1]);
yy1=tnorm([y1 y2],'min');
yy2=snorm([y1 y2],'max');
yy3=tnorm([y1 y2],'dprod');
yy4=snorm([y1 y2],'dsum');
xbasf();
subplot(3,1,1);
```

```
plot2d(x,[y1 y2],leg='mf1@mf2',rect=[0 -0.1 10 1.1]); xtitle('Member Function
Evaluation','x','mu(x)'); subplot(3,1,2);
```

```
plot2d(x,[yy1 yy3],leg='min@dprod',rect=[0 -0.1 10 1.1]); xtitle('AND
OPERATION','x','and(mf1,mf2)'); subplot(3,1,3);
```

```
plot2d(x,[yy2 yy4],leg='max@dsum',rect=[0 -0.1 10 1.1]); xtitle('OR
OPERATION','x','or(mf1,mf2)');
```

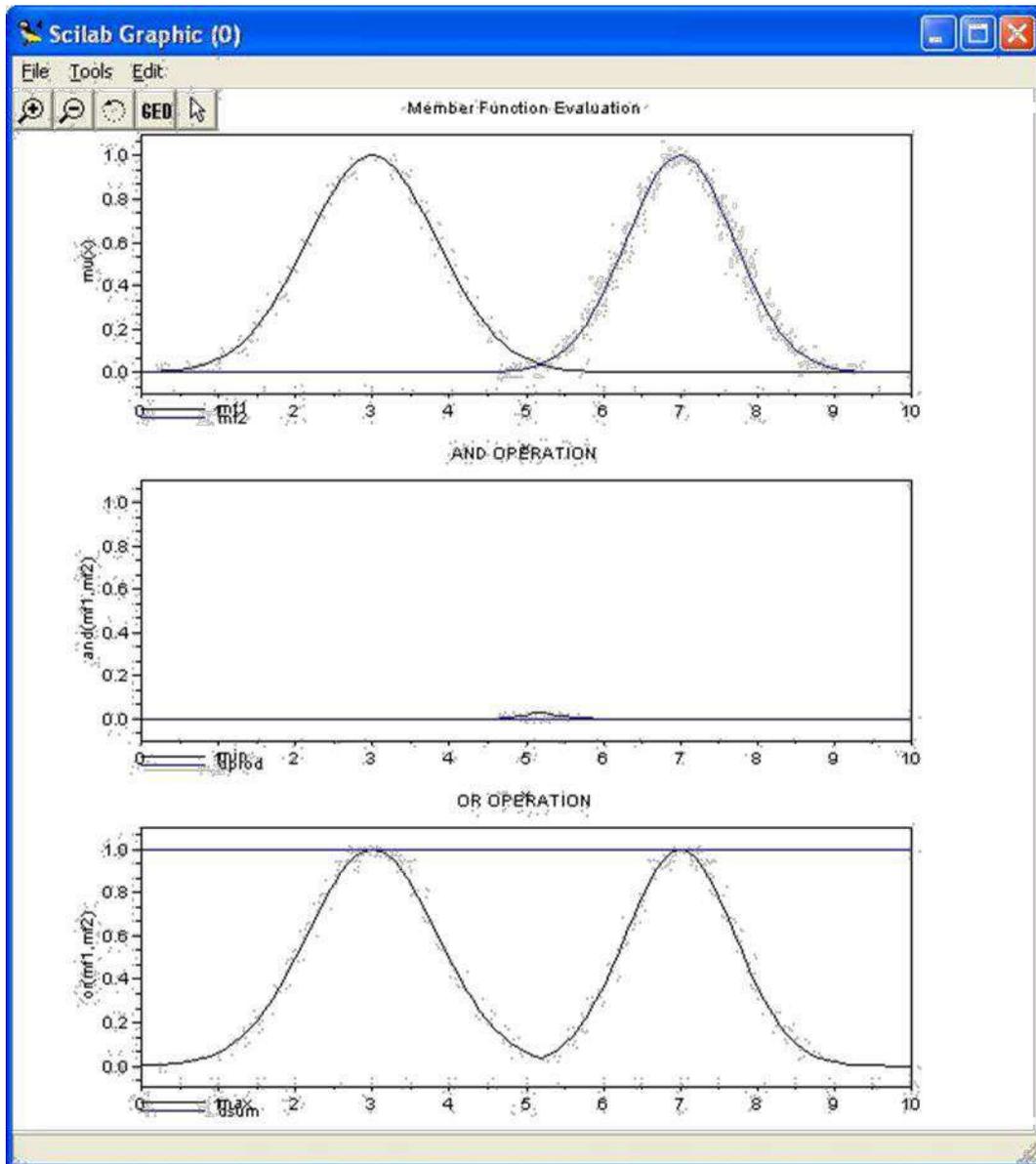


Рис. 1.7. Результаты работы программы из примера 1.7.

Минимаксная интерпретация является наиболее распространенной при построении нечетких систем. Тем не менее, на практике довольно часто используется альтернативная вероятностная интерпретация конъюнктивных и дизъюнктивных операторов.

Дополнение нечеткого множества есть не что иное, как математическое представление вербального выражения “НЕ  $A$ ”, где  $A$  – нечеткое множество, описывающее некоторое размытое суждение.

Описание функции дополнения:

$y = \text{complement}(x, \text{class} [, \text{class\_par}])$ , где  $x, y$  – матрицы, размерностью  $[m, n]$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**class** – строка, задает вид оператора дополнения. Может принимать следующие значения: «**one**» для обычного дополнения, «**sugeno**» для дополнения по формуле Сугено и «**yager**» для формулы Ягера.



`class_par` – скалярная величина, которая используется в S-конормах «`sugeno`» и «`yager`».

Пример 1.8. Программа использования операции дополнения

```
x=[0:0.1:10];
```

```
y1=gaussmf(x,[3 1.2]);
```

```
y2=complement(y1,'one');
```

```
xbasc();
```

```
plot2d(x,[y1 y2],leg='mf1@Not_mf1',rect=[0 -0.1 10 1.1]); xtitle('Member Function and
Inverse','x','mu(x)');
```

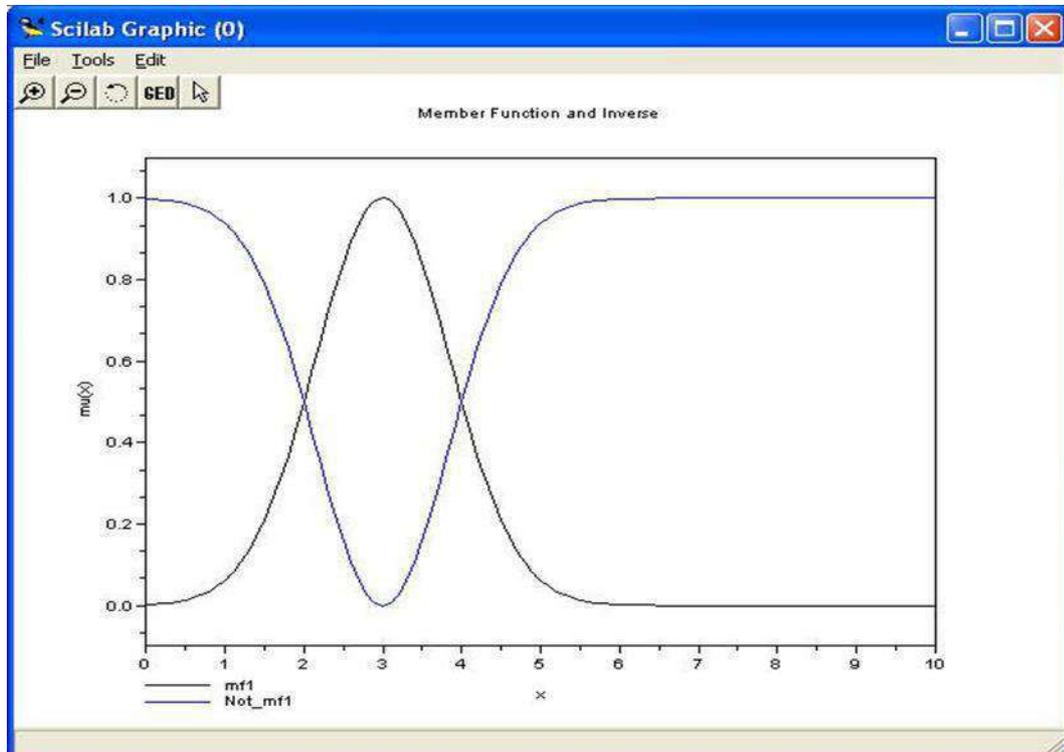


Рис.1.8. Функция дополнения

## Контрольные вопросы

20. В чем особенность метода нечётких множеств?
21. Что задаёт характеристическая функция (функция принадлежности)?
22. В чем особенность нормальной характеристической функции (функции принадлежности)?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

23. В чем общность и различие между вероятностными моделями и моделями нечётких множеств?



- У Каким образом может быть построена характеристическая функция (функция принадлежности)?
- У Какие операции могут быть применены к нечётким множествам?
- У В каких случаях может быть применён метод принятия решения на основе нечётких множеств?
- У Какими аналитическими функциями может быть описана характеристическая функция (функция принадлежности)?
- У Перечислите и охарактеризуйте типовые функции принадлежности нечётких множеств.
- У Что понимается под нечёткой моделью?

## СОЦИОЛОГИЯ

### Методические указания по практическим занятиям

#### *Практическое занятие 1*

Тема: Предмет социологии.

15. Определение понятия социологии как науки. Социология и обществознание.
16. Социология и социологическое знание. Уровни социологического знания.
17. Законы, категории, функции и методы социологии.

Социология (от лат. общество и греч. - слово, понятие, учение) наука о закономерностях становления, функционирования и развития общества, социальных отношений и социальных общностей. В настоящее время социология трактуется как одна из составных частей обществознания. Если считать, что социология - наука об обществе, социальных общностях и отношениях и механизма их формирования и их деятельности.

Большинство социологов различают три уровня социологического знания: общесоциологическая теория, специальные (частные) социологические теории, эмпирические социологические исследования.

Социальные законы - необходимые, существенные, устойчивые, повторяющиеся отношения между социальными явлениями и процессами.

Социологические категории - наиболее общие понятия социологии, обращающие объективную социальную реальность в обобщенном виде, в ее появлении, развитии и функционировании. Метод в социологии (от греч. - путь к чему-либо) - способ построения и обоснования социологического знания, совокупность приемов, процедур и операций эмпирического и теоретического познания социальной реальности. Функция (от лат. - исполнение, совершение) - роль, которую выполняет тот или иной элемент социальной системы в ее организации как целого, в осуществлении интересов социальных групп. В своих ответах приведите примеры социальных законов, категорий, методов и функций.

#### **Темы рефератов и докладов**

- Исторические предпосылки возникновения социологии.
- Социологические взгляды Огюста Конта.
- Учение М. Вебера об идеальных типах.
- Что изучает современная социология.
- Социология в системе обществознания.
- Роль социологии в профессиональной подготовке специалиста и руководителя.

#### *Практическое занятие 2*



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Тема: История социологии

10. Этапы развития социальной мысли:

а) Становление статистики (Политическая арифметика В. Петти. Университетская статистика А. Конринга и социальная статистика А. Кетле) .

б) Становление социологии: Социология Маркса. Социология О. Конта, Г. Спенсера и Э. Дюркгейма.

11. Современная социология:

а) Позитивистская социология. Социология структурно-функционального анализа.

б) Фрейдистская социология. "Понимающая социология".

в) Русская классическая социология



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Тематика рефератов и докладов**

О. Конт о взаимосвязи эволюции общества и развития отдельного индивида. Органическая социология Г. Спенсера Субъективная социология П. Л. Лаврова Социологические взгляды П. А. Кропоткина Теория социокультурной динамики П. Сорокина Марксизм и современная социологическая мысль

Психоаналитическая ориентация социальных идей З. Фрейда

***Практическое занятие 3***

Тема: Социология общества и личности

19. Общество как система. Социальная структура. Социологическая характеристика современного российского общества.

20. Сущность социальных институтов. Человек как субъект и объект общественных отношений.

Общество - исторически развивающаяся совокупность отношений между людьми, складывающихся в процессах их совместной деятельности. Социальная система (система от греч. - составленное из частей, соединение) - определенное целостное образование, основными элементами которого являются люди, их нормы и связи. Сфера (от греч. - шар) - область, пределы распространения чего-либо; общественное окружение, среда. Кроме социальной сферы, какие сферы жизнедеятельности общества Вы знаете?

Социальная структура (от лат. - строение) - совокупность взаимосвязанных социальных групп, а также социальных институтов и отношений между ними. Социальная связь - понятие, обозначающее любые социокультурные обязательства индивидов (групп индивидов) по отношению друг к другу. Приведите примеры социальных связей в студенческой группе.

Студенческая группа относится к малым группам.

Малая группа - такая социальная группа, между всеми членами которой имеются непосредственные контакты. Бывают и большие социальные группы (классы, слои, сословия, нации, народности и т.д.) Между социальными группами устанавливаются социальные отношения - отношения между группами людей, занимающими разное положение в обществе, принимающими неодинаковое участие в его жизни, различающимися менталитетом, уровнем и источниками доходов, структурой потребления.

Кроме общества и его составных частей, социология изучает также социальные институты. Социальный институт (институт от лат. - установление, учреждение) - исторически сложившиеся, устойчивые формы организации совместной деятельности людей. Перечислите социальные институты.

Социология, кроме общества и социальных институтов, изучает и человека как субъекта и объекта общественных отношений. В каком аспекте философия, психология и другие науки исследуют человека?

**Темы рефератов и докладов**

Современные проблемы развития общества и личности.

Общество и место личности специалиста и руководителя в нем.

Смысл жизни в зеркале социологии.

Социология и общество: пути и цели использования социологических знаний.

Место и роль социальных институтов в жизни общества.

Современная Россия: переходное общество.

Социальный кризис в современной России и пути его разрешения.

Социальные конфликты в современной России.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

***Практическое занятие 4***

Тема: Основные методы эмпирических социологических исследований. Методы и операции сбора данных, подлежащих количественному анализу. Количественные методы исследования: наблюдение, контент-анализ, опросы. Сравнительный анализ количественных и качественных методов, основные отличия.

13. Методы социологических исследований.

14. Методы и операции сбора данных в социологии.

Социальную информацию можно условно разделить на массовую и специальную. Массовая социальная информация — это коммуникационные (информационные) сообщения о знаниях, эмоциональных настроениях, волевых воздействиях, которые необходимы или важны, полезны или интересны для всех членов общества. Массовая информация адресована и доступна любому человеку, поэтому она называется массовой.

Специальная социальная информация всегда адресована каким-либо целевым группам. Соответственно видов специальной информации столько, сколько существует целевых групп в данном обществе. К специальной социальной информации относится экономическая, медицинская, политическая, научно-техническая, военная информация и др.

Социологическая информация — информация, полученная в ходе проведения социологического исследования.

Социологическая информация делится на первичную и вторичную.

Социологическое исследование — «исследование социальных объектов, отношений, процессов, направленное на получение новой информации и выявление закономерностей общественной жизни на основе теорий, методов и процедур, принятых в социологии».

Методический блок программы включает расчет выборки, обоснование выбора тех или иных методов исследования, разработку инструментария, определение методов обработки и анализа данных, составление графика и сметы исследования.

Расчет выборки — расчет количества респондентов, подлежащих исследованию, для достижения максимальной репрезентативности исследования. 2.

Обоснование методов исследования — производится с учетом специфики объекта, предмета, целей и задач исследования, достоинств и ограничений применения того или иного метода исследования. Необходимо учитывать также временные и материальные возможности данного проекта, так как различные методы требуют различных затрат.

Разработка инструментария — разработка анкеты, карты наблюдения, контент-анализа, гайда и т. п., то есть документа, с помощью которого информация будет собрана в процессе исследования.

Определение методов обработки и анализа данных — создание логической схемы обработки и анализа данных, определение как общетеоретических, так и специальных социологических методик анализа.

Процедурная часть программы — составление графика исследования, сметы, формирование и обучение группы сбора первичной информации, написание компьютерных программ обработки и анализа собранной информации.

Задание 1.

Студентам предлагается в групповом режиме обсудить и решить следующую логическую задачу:

Среди специалистов по социологическим и маркетинговым исследованиям известны так называемые «Законы статистики Уолкапа»:

Закон № 1. «Все, что угодно, коррелируется с чем угодно, особенно когда подлежащие корреляции переменные определяет один и тот же индивидуум».

Закон № 2. «Очень мало проку в обнаружении хорошей корреляции между переменной, которая вас интересует, и какой-то другой переменной, которую вы не осмыслите в результате этого хотя бы немного лучше».



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Закон № 3. «Пока вы не можете придумать логически здоровое объяснение тому, почему две переменные должны быть связаны как причина и следствие, мало проку в поиске корреляции между ними. В городе Колумбус, штат Огайо, количество дождливых дней по месяцам прекрасно коррелирует с числом букв в названии месяца».

Необходимо привести конкретные примеры, иллюстрирующие три данных закона в практической работе социолога. Для этого необходимо сначала обсудить с группой первый вопрос и сделать совместные заключительные выводы. После обсуждения первого закона нужно последовательно обсудить и проанализировать все предложенные в списке законы и также сделать заключительные выводы.

Задание 2. Напишите эссе на любую из предложенных тем:

15. Отличия социальной и социологической информации.
16. Суть биологического подхода к определению информации.
17. Роль массовой информации в обыденной жизни человека.
18. Роль специальной социологической информации и обыденной жизни человека.
19. Чем отличается суть понятия «информация» сейчас и 100 лет назад?
20. Почему средства массовой информации называют четвертой властью? Задание 3.

Разработайте программу социологического исследования. Тему исследования нужно выбрать самостоятельно, но она должна соответствовать специальности.

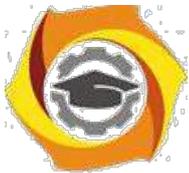
Работа выполняется в группах по 2–3 человека. Результат выполнения задания — презентация программы исследования. Программа должна содержать два раздела, все необходимые пункты плана. Презентация должна включать план, слайды, соответствующие пунктам плана программы исследования, список использованных источников.

Примерные темы исследования:

9. Социальные проблемы города.
10. Социальное самочувствие населения города.
11. Государственный служащий глазами молодежи.
12. Отношение молодежи к муниципальной власти.
13. Электоральное поведение современного студенчества.
14. Актуальные проблемы управления недвижимостью в Российской Федерации (в целом и/или по отдельным регионам).
15. Государственное регулирование демографических процессов в Российской Федерации (в целом и/или по отдельным регионам).
16. Миграционные процессы в Российской Федерации (в целом и/или по отдельным регионам).
17. Проблемы регулирования социально-трудовых отношений.
18. Государственная политика в области образования.
19. Государственная служба: социальный портрет персонала.
20. Кадровая политика в государственной и муниципальной службе: проблемы и перспективы.
21. Мотивация и стимулирование персонала государственных (муниципальных) органов управления.
22. Анализ и оценка системы пенсионного обеспечения.
23. Управление системой образования в регионе.
24. Оценка качества и уровня жизни населения.
25. Формирование рынка рабочей силы и управление им.

## *Практическое занятие 5*

Тема: Социологические теории классов в России. Современные теории социальной стратификации.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Данное занятие проходит частично в формате традиционного семинара, а на протяжении двух часов – в форме деловой игры (автор методики С.В.Творогова).

**Деловая игра.** Деловая игра посвящена обобщению и усвоению материалов по истории учений по теоретическим проблемам социальной стратификации. Основной целью использования игрового метода в ходе обучения является выработать навыки критического восприятия теорий видных социологов, вдумчивого отношения к литературе по изучаемой теме, накопление опыта аргументированного обоснования своей позиции в полемике с оппонентами.

**Тренировочная игра.** Для ознакомления студентов с принципами игрового взаимодействия проводится тренировочная игра по облегченной схеме, предметом рассмотрения которой являются прочитанные студентами работы классиков социологии об обществе как социальной структуре (См. выше – План практического занятия по теме 1). В ходе данного занятия студенческая группа разбивается на подгруппы по 4 – 6 человек, каждая из которых должна подготовить сообщение по работе одного из авторов, проработанной дома. После этого каждая группа выступает с небольшим сообщением о принципах рассмотрения общества, которых придерживался тот или иной автор. Следующий этап работы - групповой разбор изложенных подходов: в каждой из подгрупп рассматриваются достоинства и недостатки соответствующих концептуальных подходов, которые потом выносятся на обсуждение всей аудитории.

Деловая игра по теме «К. Маркс, М. Вебер и их последователи о социальной стратификации». За неделю до проведения игры студенты разделяются на две группы: сторонников теоретических представлений современных последователей К. Маркса и М. Вебера (неомарксисты и неовеберянцы). Каждая такая группа ответственна за более детальное изучение современных теорий в рамках парадигм К. Маркса и М. Вебера соответственно. Каждая группа получает следующее задание:

подготовить краткий обзор теоретических воззрений по соответствующему направлению (от основоположника данного подхода до современных теоретиков, работающих в рамках данных представлений);

проанализировать любую реальную ситуацию с позиций рассматриваемой теоретической концепции;

рассмотреть основные положения альтернативной концепции и подготовить замечания к ним (с позиций «собственной» концепции).

На занятии работа организуется следующим образом:

## Вступление

До начала игры оговариваются основные принципы ее проведения, относящиеся ко всем присутствующим и всем высказываемым точкам зрения:

равенство

активность

корректность

доверие и сотрудничество

рациональность

Выделяется группа экспертов (3-5 человек), задача которой заключается в оценке деятельности обеих групп.

## Часть 1. Обзор теоретических представлений

Задачи каждой из групп в данной части игры – изложить основные положения той концепции, которую она представляет. Основные требования – четкость представления и системность изложения материалов. Эта часть относительно непродолжительна по времени – на выступление отводится по 5 минут, на подготовку – от 5 до 10 минут.

## Часть 2. Анализ реальной ситуации

30. этой части студенты демонстрируют понимание изученных материалов, способность применять предложенный авторами категориальный аппарат в собственных разработках, умение



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

вычленять фрагменты действительности, формулировать проблему и анализировать ее. На подготовку отводится 15-20 минут, на выступление – до 10 минут.

## Релаксация

Для того, чтобы участники игры сохраняли высокий уровень активности на всем протяжении игры, им необходимо определенное переключение, отдых. Предотвращая распыление активности на переменах и препятствуя перерывам в игре, мы включаем в схему игры данный этап. Требования к его проведению: все участники игры должны быть задействованы; тип деятельности должен отличаться от того, который является основным в ходе игры; процесс должен быть достаточно увлекательным и по возможности неожиданным. Продолжительность этого этапа игры не превышает 10 минут.

## Часть 3. «Дискуссия»

данной части игры происходит обмен вопросами и ответами между группами. Вопросы могут касаться тем, затронутых в ходе двух предыдущих частей, или быть с ними не связанными.

10. В любом случае вопросы должны затрагивать концептуальные моменты рассматриваемых теоретических подходов и отражать не личные представления группы, а позиции школы, выразителями идей которой группа является на время игры.

Взаимодействие организуется следующим образом: группы готовят вопросы, обмениваются первой парой вопросов и готовят ответы на них, поочередно представители каждой группы отвечают на сформулированные вопросы, и процедура повторяется. В этом случае на подготовку вопросов отводится 15-20 минут, на подготовку ответов 5-10 минут.

Для подготовленной аудитории данный этап игры может быть проведен в виде серии 5-минутных встреч-раундов между всеми поочередно участниками обеих групп. В этом случае каждый из участников имеет право задать вопрос своему визави, получить на него ответ и ответить на аналогичный вопрос другой стороны. Вопросы разрабатываются группой заранее в ходе подготовительного этапа, продолжительность которого составляет примерно 20-25 минут. В ходе раундов в случае необходимости группа может приходить на помощь своему товарищу, но только в случае его обращения за такой помощью – сама группа не должна прерывать ход дискуссии.

Содержательный этап игры кончен – надо подвести некоторые итоги, посмотреть на проделанную работу в более широком контексте.

Эксперты дают свою оценку деятельности обеих групп, возможными критериями оценки могут быть такие показатели как активность группы, уровень подготовки к игре, профессиональная грамотность выступлений, качество представления материала, скоординированность действий группы и уровень группового единства, оригинальность высказанных соображений и т.п.

качестве темы для размышления формулируются вопросы типа: «Что было сделано?», «Что мы вынесли из сегодняшней встречи?», «Что делал каждый из нас?», «Что делала каждая группа?», «Чего мы достигли в итоге?». На размышление по этим вопросам отводится 3-5 минут, после чего желающие могут высказаться на эту тему (5-10 минут).

## Темы рефератов и докладов

17. Власть как основной критерий стратификации, признаки власти.
18. Собственность и социальный престиж как критерии социальной стратификации.
19. Теория классов К.Маркса и становление стратификационной теории.
20. Макс Вебер: классический этап становления социологии неравенства.
21. Развитие У.Муром и К.Девисом функциональной концепции социального неравенства.
22. Основные типы стратификационных систем (по Д.Грусскому).
23. Касты и сословия в социологической теории.
24. Социальные классы и слои в социологической теории.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

***Практическое занятие 6***

Тема: Определение социальной мобильности. Типы мобильности: вертикальная, горизонтальная.

20. Типы мобильности: вертикальная, горизонтальная.
21. Географическая мобильность, индивидуальная и групповая, организованная, структурная мобильность.
22. Измерение мобильности.
23. Объем, масштаб, коэффициент, степень, диапазон, дистанция мобильности.

Задание 1. Выберите верные суждения о видах социальной мобильности и запишите цифры, под которыми они указаны.

28. Внутрипоколенная мобильность предполагает изменение социального статуса от одного поколения к другому.
29. Наряду с индивидуальной мобильностью существует и групповая мобильность.
30. Повышение человека в должности служит примером горизонтальной восходящей мобильности.
31. С переходом к индустриальному обществу социальная мобильность возрастает.
32. Неоднократное изменение человеком на протяжении жизни своего социального статуса служит проявлением межпоколенной мобильности.

Задание 2.

Выберите верные суждения о видах и особенностях социальной мобильности и запишите цифры, под которыми они указаны.

32. Перемещение человека из одной социальной группы в другую без изменения статуса служит проявлением горизонтальной социальной мобильности.
33. Социальная мобильность выражается в изменении индивидом своего общественного положения.
34. Вертикальная социальная мобильность всегда ведет к повышению социального статуса личности.
35. Различают два вида социальной мобильности: горизонтальную и восходящую.
36. Средневековому обществу была присуща низкая социальная мобильность.

Задание 3.

Аналитик коммерческого банка А. уехал из Москвы в Париж на двухлетнюю стажировку в известной французской финансовой компании. А. будет во Франции работать в той же должности, что и в Москве, получать примерно такую же зарплату. Выберите в приведенном ниже списке характеристики социальной мобильности, относящихся к данному примеру, и запишите эти цифры, под которыми они указаны.

14. Индивидуальная.
15. Восходящая.
16. Миграция.
17. Горизонтальная.
18. Нисходящая.
19. Межпоколенная.

Задание 4.

Назовите любые три лифта вертикальной социальной мобильности и проиллюстрируйте примером каждый из них. (Каждый пример должен быть сформулирован развернуто).

Задание 5.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Выберите верные суждения о каналах вертикальной социальной мобильности и запишите цифры, под которыми они указаны.

- \* Каналы вертикальной социальной мобильности стабильны и не зависят от развития общества.
- \* Основным каналом вертикальной социальной мобильности в современном обществе является образование.
- \* В качестве каналов вертикальной социальной мобильности могут выступать политические партии.
- \* Каналы вертикальной социальной мобильности позволяют людям достигать более высокого социального статуса независимо от личных усилий.
- \* Функционирование каналов социальной мобильности снижает чувство социальной несправедливости и обеспечивает стабильность общества.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

***Заочная форма обучения***

***Практическое занятие 1***

Тема: Предмет социологии.

в Определеение понятия социологии как науки. Социология и обществознание.

в Социология и социологическое знание. Уровни социологического знания.

в Законы, категории, функции и методы социологии.

Социология (от лат. общество и греч. - слово, понятие, учение) наука о закономерностях становления, функционирования и развития общества, социальных отношений и социальных общностей. В настоящее время социология трактуется как одна из составных частей обществозна-ния. Если считать, что социология - наука об обществе, социальных общностях и отношениях и механизма их формирования и их деятельности.

Большинство социологов различают три уровня социологического знания: общесоциологи-ческая теория, специальные (частные) социологические теории, эмпирические социологические исследования.

Социальные законы - необходимые, существенные, устойчивые, повторяющиеся отноше-ния между социальными явлениями и процессами.

Социологические категории - наиболее общие понятия социологии, обращающие объек-тивную социальную реальность в обобщенном виде, в ее появлении, развитии и функционирова-нии. Метод в социологии (от греч. - путь к чему-либо) - способ построения и обоснования социо-логического знания, совокупность приемов, процедур и операций эмпирического и теоретического познания социальной реальности. Функция (от лат. - исполнение, совершение) - роль, которую выполняет тот или иной элемент социальной системы в ее организации как целого, в осуществле-нии интересов социальных групп. В своих ответах приведите примеры социальных законов, кате-горий, методов и функций.

**Темы рефератов и докладов**

Исторические предпосылки возникновения социологии.

Социологические взгляды Огюста Конта.

Учение М. Вебера об идеальных типах.

Что изучает современная социология.

Социология в системе обществознания.

Роль социологии в профессиональной подготовке специалиста и руководителя.

***Практическое занятие 2***

Тема: История социологии

и Этапы развития социальной мысли:

а) Становление статистики (Политическая арифметика В. Петти. Университетская стати-стика А. Конринга и социальная статистика А. Кетле) .

б) Становление социологии: Социология Маркса. Социология О. Конта, Г. Спенсера и Э.

Дюркгейма.

\* Современная социология:

а) Позитивистская социология. Социология структурно-функционального анализа.

б) Фрейдистская социология. "Понимающая социология".

в) Русская классическая социология



Методические указания  
по выполнению контрольной работы  
по дисциплине

## Схемотехника

### Электронные усилители

**Принципы построения усилителей.** Линейный и нелинейный, стационарный и переходной режимы работы усилителя. Основные показатели и характеристики, определяющие эффективность работы усилителя. Входные и выходные параметры усилителя. Частотные и нелинейные искажения

**Обратная связь в электронных устройствах.** Основные понятия теории устойчивости линейных систем. Оценка устойчивости усилителя на основе физических представлений (баланс амплитуд и фаз). Критерий Найквиста. Запасы устойчивости. Применение положительной ОС в генераторах сигналов

**Каскады предварительного усиления.** Требования, предъявляемые к каскадам предварительного усиления и особенности их анализа. Схемотехнические решения по обеспечению начальных условий работы усилительного каскада

**Апериодические усилительные каскады,** их принципиальные схемы. Применение эквивалентных схем для анализа каскадов предварительного усиления. Модели усилительных элементов, используемые при этом анализе. Коэффициенты усиления, амплитудно-частотные и переходные характеристики

#### **Вопросы для самопроверки**

и Какая из возможных схем включения усилительных приборов позволяет получить наибольший коэффициент усиления по мощности?

и Какой каскад необходимо применить на выходе усилителя для согласования с малым сопротивлением нагрузки?



- 20) Каковы основные достоинства каскада с ОБ?
- 21) Что такое граничная частота транзистора? Как она зависит от схемы включения транзистора?
- 22) Сравните режимы работы А и В по коэффициенту полезного действия.
- 23) В каких случаях целесообразно использовать режимы А, В, С и D?
- 24) Почему нагрузочная прямая постоянного тока в трансформаторном каскаде проходит более круто, чем нагрузочная прямая переменного тока, а в резисторном каскаде наоборот?
- 25) Какие существуют способы межкаскадной связи? Назовите их достоинства и недостатки.
- 26) Какие существуют способы стабилизации тока покоя?
- 27) Что такое обратная связь, петля обратной связи?
- 28) Дайте определение возвратной разности и возвратного отношения (петлевого усиления и глубины обратной связи).
- 29) Как зависит выходное сопротивление усилителя от способа снятия ООС?
- 30) Как зависит входное сопротивление усилителя от способа ввода ООС?
- 31) Расскажите правило определения способа снятия ООС (по току и напряжению).
- 32) Дайте определение сквозного коэффициента усиления. Как он меняется при введении цепи ООС?
- 33) Дайте определение стабильности коэффициента усиления, Как ООС влияет на него?
- 34) Поясните на примере возможность уменьшения нелинейных искажений с помощью ООС.
- 35) Как изменяются частотные искажения усилителя, охваченного цепью ООС?
- 36) Дайте определение частотно-зависимой ООС.

**Многокаскадные усилители.** Цепи связи между каскадами и их схемы замещения. Частотные характеристики многокаскадных усилителей. Широкополосные (импульсные) усилительные каскады. Основные виды и характеристики импульсных сигналов и их последовательностей. Анализ переходных процессов при смене состояний усилителя. Особенности АЧХ широкополосных трактов. Применение высокочастотной и низкочастотной коррекции для обеспечения требуемой формы выходного импульса

**Оконечные усилительные каскады.** Особенности ручного и компьютерного анализа каскадов. Однотактный трансформаторный усилитель. Построение выходных нагрузочных характеристик по постоянному и переменному токам. Основные энергетические соотношения. Двухтактный трансформаторный усилитель. Основные схемотехнические решения. Построение выходных процессов по току и напряжению. Основные энергетические соотношения. Двухтактные бестрансформаторные оконечные каскады. Осо-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

бенности работы и свойства двухтактных каскадов. Режимы работы транзисторов и нелинейные искажения выходного сигнала. Основные разно-



видности бестрансформаторных двухтактных каскадов. Способы повышения энергетической эффективности оконечных каскадов

**Вопросы для самопроверки**

- с Что такое эквивалентный генератор тока (напряжения)?
- с Нарисуйте принципиальную схему резисторного каскада на транзисторе. Покажите пути, по которым проходят постоянная и переменная составляющие коллекторного тока, а также пути протекания постоянной составляющей тока базы.
- с Нарисуйте эквивалентную схему резисторного каскада на транзисторе для области средних частот. Определите коэффициенты усиления тока, напряжения, мощности.
- с Нарисуйте эквивалентную схему резисторного каскада в области нижних частот. Поясните причину возникновения частотных искажений.
- с Нарисуйте эквивалентную схему резисторного каскада в области верхних частот. Поясните причину возникновения частотных искажений.
- с Какие искажения возникают при усилении резисторным каскадом импульсных сигналов
- с Для какой цели ставятся блокирующие конденсаторы в цепи эмиттера?
- с Что такое площадь усиления?
- с На каком принципе основана работа схем высокочастотной коррекции с помощью корректирующих катушек?
- 10. Как можно корректировать АЧХ с помощью цепи  $K_f C_f$ ? Нарисуйте эквивалентную схему каскада.
- 13) Приведите пример использования частотно-зависимых цепей ООС с целью коррекции АЧХ резисторных каскадов.
- 14) Каковы достоинства и недостатки трансформаторных каскадов?
- 15) Нарисуйте эквивалентную схему трансформаторного каскада и объясните причины возникновения частотных искажений.
- 16) Каково назначение входного трансформатора?
- 17) Дайте определение каскада мощного усиления.
- 18) Как строится нагрузочная прямая однотактного трансформаторного каскада мощного усиления на транзисторе, работающем в режиме А?
- 19) Дайте определение оптимальной нагрузки каскада.
- 20) Почему точку покоя каскада, работающего в режиме В, выбирают не на оси абсцисс выходных характеристик, а несколько смещают вверх?
- 21) Двухтактный каскад в режиме А. Основные достоинства и недостатки.
- 22) Расскажите об основных достоинствах и недостатках трансформаторных каскадов усиления мощности.
- 23) Как строится нагрузочная прямая резисторного каскада мощного усиления (режим А)?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

24) Нарисуйте схему бестрансформаторного усилителя на комплементарных транзисторах с разделительным конденсатором на выходе. Покажите пути прохождения тока через нагрузку.



13. Нарисуйте схему бестрансформаторного усилителя на комплементарных транзисторах с непосредственной связью с нагрузкой. Покажите пути прохождения тока через нагрузку.

14. Нарисуйте схему бестрансформаторного усилителя на транзисторах одинаковой структуры с последовательным возбуждением. Поясните принцип работы схемы.

15. Нарисуйте схемы фазоинверсных каскадов. Сравните их по качественным показателям.

### **Электронные схемы на основе усилителей**

**Операционные усилители.** Схемотехника функциональных узлов операционных усилителей. Инвертирующие и неинвертирующие усилители, повторители напряжения входного сигнала. Устройства на ОУ, осуществляющие суммирование и вычитание. Базовые функциональные узлы аналоговой схемотехники на основе ОУ. Устройства для аналоговых вычислений. Устройства на ОУ, осуществляющие линейные математические операции над сигналами: дифференцирование и интегрирование, и нелинейные математические операции: логарифмирование, антилогарифмирование (потенцирование). Компаратор напряжения на ОУ. Принцип функционирования и упрощенная схема компаратора на ОУ без цепей ОС. Сквозная передаточная характеристика компаратора. Быстродействие и погрешности компаратора. Применение положительной ОС в компараторах на ОУ

**Активные фильтры.** Основные типы фильтров. Способы аппроксимации амплитудно-частотных характеристик фильтров. Активные фильтры на основе ОУ. Применение частотно-зависимых цепей на входе и/или в тракте глубокой отрицательной обратной связи. Примеры построения схем активных RC-фильтров первого и второго порядков. Фильтры нижних и верхних частот, полосовые, режекторные (заграждающие), фазовые фильтры

### **Генераторы периодических сигналов.**

41. Генераторы прямоугольных импульсов. Общие сведения и основные оп-ределения. Функциональная схема и принцип работы мультивибратора. Особенности построения несимметричного мультивибратора. Функциональная схема и принцип работы ждущего мультивибратора (одновибратора). Блокинг-генераторы и особенности их применения.

42. Генераторы линейно изменяющегося напряжения. Общие принципы формирования линейно изменяющегося напряжения. Способы повышения точности формирования линейно изменяющегося напряжения. Основные типы генераторов линейно изменяющегося напряжения. Функциональные схемы и принципы работы генераторов линейно изменяющегося напряжения с интегрирующей RC-цепочкой и компенсирующей ОС

### **Вопросы для самопроверки**

10. Какие существуют виды ИС? В чем заключается их отличие?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

11. Нарисуйте частотную, переходную и амплитудную характеристики уси-лителя постоянного тока.
12. Что такое дрейф нуля? Как можно уменьшить дрейф нуля?



- В Нарисуйте схему дифференциального каскада. Расскажите о ее достоинствах.
- В Дайте определение операционного усилителя. Как он может быть использован для решения конкретных задач?
- В Перечислите свойства идеального ОУ.
- В Какова структура ОУ?
- В Для достижения какой цели питание ОУ осуществляется от двухполярного источника?
- В Нарисуйте схемы включения инвертирующего и неинвертирующего усилителя, повторителя на ОУ.
- В Как рассчитать коэффициент усиления с включенной цепью ОС?
- В Нарисуйте схему суммирующего усилителя на ОУ.
- В Нарисуйте схемы простейших активных фильтров нижних и верхних частот. Поясните принцип их работы.
- В Нарисуйте схему избирательного усилителя с двойным Т-образным мостом в цепи ОС. Поясните принцип его работы.
- В Дайте определение активного фильтра. Объясните его достоинства по сравнению с фильтром на LC элементах.
- В Произведите классификацию видов регулировок. Приведите конкретные применения отдельных видов регулировок.
- В Почему в регуляторах громкости используются резисторы со степенной характеристикой зависимости сопротивления от угла поворота подвижного контакта?
- В Почему в широкополосных усилителях для регулировки усиления используются низкоомные резисторы?
- В Нарисуйте возможную схему включения регулятора тембра в области низких частот. Поясните принцип ее работы.
- В Нарисуйте возможную схему включения регулятора тембра в области высоких частот. Поясните принцип ее работы.
- В Для какой цели применяются ступенчатые регуляторы усиления?

### **Основы построения базовых элементов цифровой техники**

**Элементная база цифровой техники.** Функционально полные системы функций алгебры логики. Синтез логических устройств в различных базах. Физическое представление логических уровней. Схемная реализация элементов НЕ, И-НЕ, ИЛИ-НЕ в технологиях ТТЛ, nМОП, КМОП. Серии интегральных схем, их функциональный состав. Основные параметры, характеристики логических элементов (коэффициенты объединения по входу, разветвления по выходу, передаточная характеристика). Активные и пассивные логические уровни элементов И-НЕ, ИЛИ-НЕ. Состояния выхода цифровых элементов и узлов. Триггеры, комбинационные схемы, регистры, счетчики



**Компьютерный анализ и проектирование электронных устройств.** Математическое моделирование электронных устройств. Особенности применения методов компьютерного моделирования. Математические компьютерные модели основных элементов и функциональных узлов электронных устройств. Основные сведения о численных методах расчета электронных устройств по эквивалентным схемам. Возможности современных систем математического моделирования электронных устройств. Основные программные пакеты и системы для математического моделирования электронных устройств и особенности их использования (Micro-Cap, MathCad, MathLab, Design Center, Design Lab и пр.)

**Вопросы для самопроверки**

- е Изобразите схему базового элемента ТТЛ-серии и поясните ее работу.
- е Изобразите схему базового элемента МОП-серии и поясните ее работу.
- е Преимущества цифровой формы сигналов.
- е Форматы представления данных.
- е Переводы чисел в позиционных системах счисления (2-16-10).
- е Арифметические операции в прямом и дополнительном кодах.
- е Способы задания логических функций. Базисы.
- е Минимизация логических функций.
- е Классификация ИМС по общепринятым признакам.
- е Система условных обозначений. Обозначения цифровых ИМС на схемах
- е Параметры и характеристики логических элементов
- е Расширители входов. Назначение. Обозначение. Характеристики
- е Элементы с общим коллектором. Назначение. Обозначение. Характеристики
- е Элемент с тремя состояниями выходов. Назначение. Обозначение. Характеристики
- е Элементы с мощным выходом. Назначение. Обозначение. Характеристики
- е Этапы синтеза цифровых схем
- е Проблемы оптимизации сложных логических функций

**В Лабораторные работы и практические занятия**

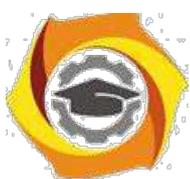
**Практическое занятие №1. Аналитический расчет каскадов предварительного усиления.** Алгоритмы расчета резисторных цепей. Выбор емкостей при синтезе частотных характеристик каскада. Формирование основных параметров каскадов предварительного усиления

**Практическое занятие №2. Анализ частотных и переходных характеристик усилительных каскадов.** Анализ методики синтеза амплитудно-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

частотных и фазо-частотных характеристик (АЧХ и ФЧХ) усилительных каскадов. Анализ методики синтеза переходных характеристик усилитель-



ных каскадов. Синтез АЧХ и ФЧХ типовых каскадов.

**Лабораторная работа №1. Исследование интегральных коммутаторов.** Экспериментальное исследование принципов работы ИМС—коммутаторов аналоговых сигналов

**Лабораторная работа №2. Исследование аналоговых устройств на основе операционных усилителей.** Анализ основных вариантов использования операционных усилителей

**Лабораторная работа №3. Исследование интегральных компараторов напряжения.** Экспериментальное исследование принципов работы ИМС—компараторов аналоговых сигналов

**Практическое занятие 3. Анализ основных типов триггеров.** Моделирование работы интегральных микросхем триггеров и анализ динамики их поведения

## Теория вероятностей и математическая статистика

### Методические указания по практическим занятиям ЗАНЯТИЕ № 1

**Непосредственное вычисление вероятности по классическому определению:  
и задачи без применения комбинаторики,  
задачи с применением комбинаторики.**

**Цель занятия:**

Выработать умения и навыки работы с вычислением вероятности непосредственно и вычислением ее с применением формул комбинаторики. Научить студентов применять формулы комбинаторики.

**Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников 37. с.17-26, а также (2) с.8-12.

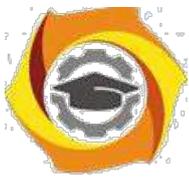
**36. Задание:**

Решить примеры: (2) №№ 1-25 (выборочно, по рекомендации преподавателя).

**30) Порядок выполнения работы:**

- 4.1. Записать классическое определение вероятности случайного события.
- 4.2. Записать формулы комбинаторики: число перестановок без повторений, число размещений, число сочетаний.
- 4.3. Сформулировать правило суммы и правило произведения событий.
- 4.4. Решить примеры см. п.3.

**В Отчет:**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- 5.1. Случайное событие.
- 5.2. Виды событий.
- 5.3. Классическое определение вероятности события.
- 5.4. Число перестановок.
- 5.5. Число сочетаний.
- 5.6. Число размещений.
- 5.7. Решенные примеры.

**39. Список литературы:**

- 100** Хуснутдинов Р.Ш. Теория вероятностей. Учебник. М.: НИЦ ИНФРА-М, 2013
- 101** Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике: учебное пособие. М.: Юрайт 2009



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**ЗАНЯТИЕ № 2**

**Решение основных задач теории вероятностей с помощью теорем сложения и умножения.**

**33. Цель занятия:**

Научить студентов применять к решению задач теоремы сложения вероятностей и умножения вероятностей. Научить студентов применять к решению задач о разрыве электрической цепи теоремы сложения вероятностей и умножения вероятностей.

**/ Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников  
**10** с.27-47, а также **(2)** с.12-30.

**/ Задание:**

Решить примеры: **(2)** №№ 26 - 88 (выборочно, по рекомендации преподавателя).

**/ Порядок выполнения работы:**

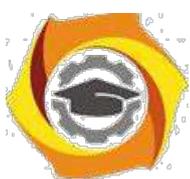
- 4.1. Записать теорему сложения вероятностей несовместных событий.
- 4.2. Записать теорему умножения вероятностей.
- 4.3. Записать теорему умножения для независимых событий.
- 4.4. Записать теорему сложения вероятностей совместных событий.
- 4.5. Рассмотреть задачу на примере последовательного соединения двух и более элементов.
- 4.6. Рассмотреть задачу на примере параллельного соединения двух и более элементов.
- 4.7. Рассмотреть задачи, описывающие сложные цепи.
- 4.8. Решить примеры см.п.3.

**В Отчет:**

- 5.1. Теорема сложения вероятностей несовместных событий.
- 5.2. Противоположные события.
- 5.3. Условная вероятность.
- 5.4. Теорема умножения вероятностей.
- 5.5. Теорема умножения для независимых событий.
- 5.6. Теорема сложения вероятностей совместных событий.
- 5.7. Сложные цепи, составленные из последовательно и параллельно соединенных элементов.
- 5.8. Решенные примеры.

**\endash      Список литературы:**

- 43.** Хуснутдинов Р.Ш. Теория вероятностей. Учебник. М.: НИЦ ИНФРА-М, 2013
- 44.** Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике: учебное пособие. М.: Юрайт 2009



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**ЗАНЯТИЕ № 3**

**Формулы полной вероятности и Бейеса. Формула Бернулли.  
Контрольная работа №1**

**42. Цель занятия:**

Познакомить студентов с понятием полной группы событий и гипотезы. Научить применять к решению задач формулу полной вероятности и формулу Бейеса. Выработать умения и навыки по применению формулы Бернулли.

**60. Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников с.50-53, а также (2) с.31-37.

**№ Задание:**

Решить примеры: (2) №№ 89-96 (выборочно, по рекомендации преподавателя).

**66. Порядок выполнения работы:**

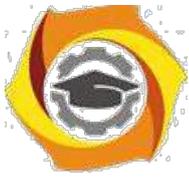
- 4.1. Записать определение полной группы событий.
- 4.2. Записать формулу полной вероятности.
- 4.3. Записать формулу Бейеса.
- 4.4. Записать формулу Бернулли.
- 4.5. Записать формулу наиболее вероятного числа появлений события.
- 4.6. Решить примеры см.п.3.

**48. Отчет:**

- 5.1. Полная группа событий.
- 5.2. Теорема о полной вероятности события.
- 5.3. Формула Бейеса.
- 5.4. Формула Бернулли.
- 5.5. Решенные примеры.

**46. Список литературы:**

64. Хуснутдинов Р.Ш. Теория вероятностей. Учебник. М.: НИЦ ИНФРА-М, 2013
65. Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике: учебное пособие. М.: Юрайт 2009



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**ЗАНЯТИЕ № 4**

**Нахождение законов распределения дискретных случайных величин (ДСВ). Функция распределения ДСВ. Нахождение функции и плотности распределения непрерывной случайной величины (НСВ).**

**В Цель занятия:**

Выработать умения и навыки по определению закона распределения ДСВ. Научить студентов находить функцию распределения ДСВ и строить ее график. Выработать умения и навыки по нахождению функции распределения и плотности распределения вероятностей непрерывной случайной величины.

**ООО Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников **25.** с.64-100, а также **(2)** с.52-82.

**45. Задание:**

Решить примеры: **(2)** №№ 164-185 (выборочно, по рекомендации преподавателя).

**43. Порядок выполнения работы:**

- к Дать определение ДСВ.
- к Дать определение закона распределения ДСВ.
- к Дать определение функции распределения ДСВ.
- к Записать определение НСВ.
- к Записать определение функции распределения НСВ.
- к Сформулировать свойства функции распределения НСВ.
- к Записать определение плотности распределения НСВ.
- к Сформулировать свойства плотности распределения НСВ.
- к Решить примеры см.п.3.

**48. Отчет:**

- 5.1. Определение ДСВ.
- 5.2. Законы распределения ДСВ.
- 5.3. Определение функция распределения ДСВ.
- 5.4. Определение НСВ.
- 5.5. Формула нахождения функции распределения.
- 5.6. Формула нахождения плотности распределения вероятностей.
- 5.7. Решенные примеры.

**21. Список литературы:**

- 32.** Хуснутдинов Р.Ш. Теория вероятностей. Учебник. М.: НИЦ ИНФРА-М, 2013
- 33.** Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике: учебное пособие. М.: Юрайт 2009



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**ЗАНЯТИЕ № 5**

**Вычисление числовых характеристик ДСВ и НСВ.**

**43. Цель занятия:**

Выработать умения и навыки по вычислению числовых характеристик дискретной и непрерывной случайной величины. Познакомить с понятиями математическое ожидание, дисперсия, среднее квадратическое отклонение, мода, медиана, асимметрия и эксцесс.

**24. Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников

**33.** с.64-100, а также **(2)** с.52-82.

**25. Задание:**

Решить примеры: **(2)** №№ 205-235 (выборочно, по рекомендации преподавателя).

**34. Порядок выполнения работы:**

- 4.1. Дать определение случайной величины.
- 4.2. Записать закон распределения дискретной случайной величины.
- 4.3. Записать определение математического ожидания ДСВ.
- 4.4. Записать определение дисперсии и среднего квадратического отклонения ДСВ.
- 4.5. Записать определение моды и медианы.
- 4.6. Записать определение асимметрии и эксцесса.
- 4.7. Записать формулы для вычисления математического ожидания.
- 4.8. Записать формулы для вычисления дисперсии.
- 4.9. Записать формулу для вычисления среднего квадратического отклонения.
- 4.10. Решить примеры см.п.3.

**44. Отчет:**

- 5.1. Математическое ожидание.
- 5.2. Математическое ожидание суммы, разности, произведения.
- 5.3. Определение дисперсии и среднего квадратического отклонения.
- 5.4. Дисперсия суммы, разности, произведения.
- 5.5. Определение моды и медианы.
- 5.6. Формулы вычисления математического ожидания.
- 5.7. Формулы вычисления дисперсии НСВ и ДСВ.
- 5.8. Формула нахождения среднего квадратического отклонения.
- 5.9. Решенные примеры.

**36. Список литературы:**

- 21.** Хуснутдинов Р.Ш. Теория вероятностей. Учебник. М.: НИЦ ИНФРА-М, 2013
- 22.** Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике: учебное пособие. М.: Юрайт 2009



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**ЗАНЯТИЕ № 6**

**Основные законы распределения непрерывных случайных величин (равномерное распределение, показательное распределение, нормальное распределение).**

**Контрольная работа №2**

**И Цель занятия:**

Выработать умения и навыки по применению равномерного, нормального и показательного законов распределения к решению задач. Получить навыки в построении графиков основных законов распределения.

**· Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников с.122-155, а также (2) с.106-114.

**· Задание:**

Решить примеры: (2) №№ 322-372 (выборочно, по рекомендации преподавателя).

**17) Порядок выполнения работы:**

- и Записать определение НСВ.
- и Записать равномерное распределение, а также основные характеристики этого закона.
- и Записать нормальное распределение.
- и Записать показательное распределение.
- и Записать основные характеристики этих законов.
- и Решить примеры см.п.3.

**· Отчет:**

- 5.1. Равномерное распределение и его характеристики.
- 5.2. Показательное распределение и его характеристики.
- 5.3. Нормальное распределение и его характеристики.
- 5.4. Вероятность попадания нормально распределенной СВ в заданный интервал.
- 5.5. Правило трех сигм.
- 5.6. Решенные примеры.

**· Список литературы:**

- В Хуснутдинов Р.Ш. Теория вероятностей. Учебник. М.: НИЦ ИНФРА-М, 2013
- В Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике: учебное пособие. М.: Юрайт 2009



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

***Заочная форма обучения***

**ЗАНЯТИЕ № 1**

**Решение основных задач теории вероятностей с помощью теорем сложения и умножения. Определение вероятности разрыва электрической цепи. Формула полной вероятности и формула Байеса. Формула Бернулли**

**- Цель занятия:**

Выработать умения и навыки по применению теорем сложения вероятностей и умножения вероятностей. Научить студентов применять эти теоремы к решению задач о разрыве электрической цепи. Выработать умения и навыки по применению формулы полной вероятности, формул Байеса и Бернулли.

**В Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников:  
в с.27-69, а также (2) с.12-46.

**к Задание:**

Решить примеры: (2) №№ 26 -110 (выборочно, по рекомендации преподавателя).

**О Порядок выполнения работы:**

- 4.1 Записать теорему сложения вероятностей несовместных событий.
- 4.2. Записать теорему умножения вероятностей.
- 4.3. Записать теорему умножения для независимых событий.
- 4.4. Записать теорему сложения вероятностей совместных событий.
- 4.5. Записать формулу полной вероятности.
- 4.6. Записать формулы Байеса и Бернулли.
- 4.7. Решить примеры см.п.3.

**в Отчет:**

- 5.1. Теорема сложения вероятностей несовместных событий..
- 5.2. Теорема умножения вероятностей.
- 5.3. Теорема умножения для независимых событий.
- 5.4. Теорема сложения вероятностей совместных событий.
- 5.5. Формула полной вероятности.
- 5.6. Формулы Байеса и Бернулли.
- 5.7. Решенные примеры.

**В Список литературы:**

- 14) Хуснутдинов Р.Ш. Теория вероятностей. Учебник. М.: НИЦ ИНФРА-М, 2013
- 15) Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике: учебное пособие. М.: Юрайт 2009



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**ЗАНЯТИЕ № 2**

**Нахождение законов распределения дискретных случайных величин (ДСВ).  
Функция распределения ДСВ. Нахождение функции и плотности  
распределения непрерывной случайной величины (НСВ). Вычисление  
числовых характеристик ДСВ и НСВ.**

**к Цель занятия:**

Выработать умения и навыки по определению закона распределения дискретной случайной величины, по вычислению числовых характеристик ДСВ. Научить студентов находить функцию распределения ДСВ и строить ее график.

**22. Краткие теоретические сведения:**

Справочный материал и разобранные примеры приведены в любом из следующих источников

в с.64-100, а также (2) с.52-82.

**37. Задание:**

Решить примеры: (2) №№ 164-185, 205-235 (выборочно, по рекомендации преподавателя).

**25. Порядок выполнения работы:**

18. Дать определение ДСВ.
19. Дать определение закона распределения ДСВ.
20. Дать определение функции распределения ДСВ.
21. Записать определение математического ожидания ДСВ.
22. Записать определение дисперсии и среднего квадратического отклонения ДСВ.
23. Записать определение функции распределения НСВ.
24. Записать определение плотности распределения НСВ.
25. Записать формулы для вычисления математического ожидания.
26. Записать формулы для вычисления дисперсии и среднего квадратического отклонения.
27. Решить примеры см.п.3.

**22. Отчет:**

- 5.1. Определение ДСВ.
- 5.2. Законы распределения ДСВ.
- 5.3. Определение функция распределения ДСВ.
- 5.4. Определение математического ожидания.
- 5.5. Определение дисперсии и среднего квадратического отклонения.
- 5.6. Формула нахождения функции распределения.
- 5.7. Формула нахождения плотности распределения вероятностей.
- 5.8. Формулы вычисления математического ожидания.
- 5.9. Формулы вычисления дисперсии НСВ и ДСВ.
- 5.10. Формула нахождения среднего квадратического отклонения.
- 5.11. Решенные примеры.

**24. Список литературы:**

- У Хуснутдинов Р.Ш. Теория вероятностей. Учебник. М.: НИЦ ИНФРА-М, 2013
- У Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике: учебное пособие. М.: Юрайт 2009



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

по дисциплине

## Технологии баз данных

### - Основы проектирования БД

**1.1. Модели и типы данных.** БД и СУБД. Этапы проектирования БД. Модели структур данных. Основные элементы реляционной модели. Типы взаимосвязей различных сущностей.

**1.2. Метод нормальных форм.** Основы реляционной алгебры. Нормализация. Технология приведения к нормальным формам. Средства ускоренного доступа к данным. Индексирование.

Вопросы и упражнения для самоконтроля:

Каковы этапы разработки проекта?

Что называется концептуальной моделью данных?

В чем отличие концептуальной и логической модели?

Какие типы связей предполагаются в БД?

В чем суть метода нормальных форм при проектировании БД?

Сформулируйте суть трех первых нормальных форм.

В каком порядке выполняются нормальные формы?

Какой материал является основой для проектирования?

### ее СУБД реляционных БД

**2.1. Архитектура СУБД.** Основные объекты СУБД. Задачи СУБД. Создание таблиц в различных режимах. Операции с таблицами. Виды соединений в таблицах. операции с пакетом. Создание таблиц с помощью конструктора. Типы данных. Режимы разработки таблиц. Связывание таблиц, установка типов связей. Сортировка и поиск данных. Создание простого и составного первичного ключа. Вторичный ключ. Контроль правильности ввода данных, добавление условий на значение поля

**2.2. Манипуляции с данными в БД.** Конструирование запросов. Типы запросов. Формы и отчеты. Технология разработки. Средства ускоренного доступа к данным. Сортировка и фильтрация. Изменение данных средствами запроса

**2.3. Разработка приложений пользователей.** Среда VBA. Переменные, типы данных и константы. Стандартные функции и выражения. Инструкции VBA. Операторы языка. Модули. Макросы. Операторы присваивания, цикла, безусловного перехода. Управление выполнением программы

**2.4. Обеспечение информационной безопасности приложения.** Принципы управления доступом. Формы для управления безопасностью Вопросы и упражнения для самоконтроля:

В Что такое таблица БД?

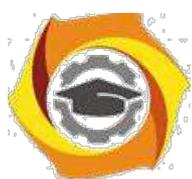
В Каковы задачи СУБД?

В Какие типы данных может содержать БД?

В Что такое ключ и для каких целей он создается?

В Какие связи образуются в БД?

В Какова технология импорта данных из электронных таблиц?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



38. Какова технология импорта данных из текстового файла?
39. Что такое запрос к БД?
40. Что означают групповые операции?
41. Как реализуется сортировка данных?
42. Для каких целей используется запрос?
43. Каковы преимущества запроса с параметром?
44. Что такое VBA? Для каких целей он применяется?
14. .Что такое форма? Для чего они создаются?
  - 1) Какие типы форм используются?
  - 2) Что такое отчет?
  - 3) Какие средства редактирования отчетов доступны?

### 31) **Распределенная обработка данных**

**3.1. Структурированный язык запросов.** Основы SQL. Создание и редактирование таблиц. Операторы и функции языка SQL. Создание простых запросов. Фильтрация данных. Операторы SELECT, INSERT, UPDATE, DELETE. Взаимодействие с языками программирования. Применение предложений FROM, WHERE, ORDERBY, HAVING, GROUPBY, предикатов DISTINCT, TOP, манипулирование данными

**3.2. Модели распределенных систем.** Совместный доступ к БД с помощью SQL Server. Режимы работы с БД в сети. Распределение функций в архитектуре "клиент-сервер". Установка связи с источником данных посредством интерфейса OLE DB Создание проекта с использованием существующей базы данных. Средства языка Transact-SQL. Организация транзакций, их свойства. Форматы транзакций. Инструкции SQLBEGINTRANSACTION, COMMIT, предложение ROLLBACK. Механизм хранимых процедур. Инструкция CREATE PROC. Создание процедур в БД. Средства языка Transact-SQL. Вызов сохраненных процедур

**3.3. Разработка БД для Интернета.** Язык HTML. Преобразование объектов БД в Web-страницы. Создание страниц доступа к БД. Добавление на страницу элементов ActiveX. Публикация данных с помощью Web-страниц. Структура документа HTML. Основные теги: HTML, HEAD, BODY, TITLE, <H1>, <P>, ALIGN, <UL>, <TABLE>, CAPTION, BORDER, ALIGN, WIDTH, BGCOLOR, CELLPADDING, CELLSPACING, <TR>, <A>, NAME

Вопросы и упражнения для самоконтроля:

- В Какие задачи реализуются с помощью языка SQL?
- В В чем заключается назначение механизма ODBC?
- В Какие операции над множествами может реализовать оператор SELECT?
- В Что такое индексы БД? Как они создаются?
- В Что такое транзакция?
- В Какие свойства имеют транзакции?



40. Чем отличаются динамические Интернет страницы от статических?
  41. Укажите причины, достоинства и недостатки обработки данных XML-документа на сервере.
  42. Поясните особенности клиент-серверной технологии?
  43. Какие параметры могут быть использованы в операторе SELECT?
- Курсовое проектирование. Разработка БД по вариантам*

## 102 Лабораторные работы и практические занятия

*Практическое занятие №1. Проектирование БД.* Анализ предметной области. Определение последовательности выполнения задач. Построение концептуальной и логической модели.

*Практическое занятие №2. Разработка таблиц БД.* Создание таблиц с помощью конструктора. Задание типов данных. Ввод и редактирование данных. Организация связей между таблицами.

*Практическое занятие №3. Экспорт и импорт данных в БД.* Импорт электронных таблиц. Импорт и присоединение текстовых файлов. Исправление ошибок после экспорта.

*Лабораторная работа №1. Создание простых запросов.* Создание запроса в режиме конструктора. Запросы с параметром. Вычисляемые поля.

*Лабораторная работа №2. Создание сложных запросов.* Конструирование сложных условий отбора. Создание вычисляемых полей. Использование функций в вычисляемых полях. Запросы к связанным таблицам.

*Лабораторная работа №3. Разработка форм.* Создание форм с помощью мастера и автоматически. Режим конструктора. Редактирование формы. Оформление формы.

*Лабораторная работа №4. Разработка отчетов.* Создание отчетов автоматическое и с помощью мастера. Редактирование и форматирование отчета. Экспорт отчетов. Преобразование форматов.

*Практическое занятие №4. Управление БД с помощью VBA.* Написание процедур создания БД и ее заполнение, выполнения обработки данных, создание простых запросов.

*Лабораторная работа №5. Работа с БД на языке SQL.* Разработка БД и запросов к ней на языке структурированных запросов.

*Лабораторная работа №6. Разработка структуры данных на сервере.* Создание клиентского соединения. Создание структуры данных. Нормализация. Разработка структуры индексов.

*Лабораторная работа №7. Гипертекстовые БД.* Структура документа HTML. Разработка структуры гипертекстовой БД. Исследование и модернизация программы получения данных из БД.

## 5. Задание на курсовой проект



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Разработать базу данных по индивидуальному варианту. Перечень работ, подлежащих выполнению определяется заданием, утвержденным заведующим кафедрой. Типовой вариант предполагает:

- / обоснование концептуальной модели базы данных;
- / выполнение нормализации базы данных;
- / конструирование таблиц базы данных и заполнение полей;
- / оформление схемы данных;
- / разработку запросов к базе данных. Запросы должны быть следующих

типов:

5. по одному условию;
6. по двум и более условиям;
7. с использованием оператора Like для поиска по фрагменту данных;
8. с параметром;

-с групповой операцией.

Должно быть разработано **не менее пяти запросов**. Конкретные запросы необходимо выбрать из примеров индивидуального задания, а при отсутствии запросов требуемых типов - разработать самостоятельно. Желательно в отдельных запросах применить функции времени. В случае необходимости для организации требуемых запросов разработчику следует изменить состав информации, хранящейся в базе данных;

- разработку форм базы данных. Форма должна предусматривать компактное размещение материала на основе элемента управления "Вкладки". Вкладок необходимо создать не менее трех на основе подчиненных форм. Отредактировать на них все необходимые надписи, использовать графические элементы (фото) из разработанных таблиц хранения данных;

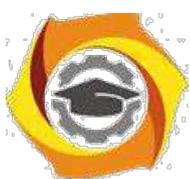
- разработку отчета. Отчет должен быть приближен к реальным документам (по теме индивидуального задания). В отчете необходимо использовать уровни группировки данных по выбранным полям, текущую дату и подписи должностных лиц.

Работа может быть выполнена в доступной для автора среде. В качестве инструмента разработки допустимо использовать СУБД MSAccess, OpenOfficeBase, MSSQLServer, MySQL, PostgreSQL и пр.

Вариант индивидуального задания выбирается по последним двум цифрам зачетной книжки (табл.1). Для определения варианта в диапазоне значений 31-60 предварительно необходимо вычесть 30, в диапазоне 61-90 предварительно вычесть 60, для остальных - вычесть 90.

Таблица 1. Индивидуальные варианты заданий на курсовой проект

Вариант и тема	Обязательный перечень форм и запросов
1. Модель «Деканат» должна содержать информацию о списках групп, начислен-	- получить полную информацию обо всех сданных экзаменах и зачетах в выбранной группе;



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

ных стипендиях, зачетах,  
оценках, полученных на эк-  
заменах

- получить список всех студентов, не сдавших зачет по выбранной дисциплине;
- получить список студентов, не сдавших зачет более чем по 2-м предметам;
- получить список студентов, не получивших стипендию;
- получить список студентов, получивших ту же оценку на экзамене по выбранному предмету, что и выбранный студент;
- получить список предметов, по которым получено одинаковое количество зачетов;
- получить список групп, в которых студенты получили заданное количество оценок «пять»



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

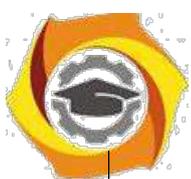
по выбранному предмету

2. Модель «Страховая компания» должна содержать информацию о клиентах, договорах страховки, объектах страховки, страховых случаях

- получить список клиентов с заданным объектом страховки;
- получить список клиентов с заданным объектом страховки и одинаковой страховой суммой;
- получить список всех договоров, по которым произошел заданный страховой случай;
- получить список всех страховых случаев, происшедших с даты по дату;
- получить список клиентов, у которых тот же объект страховки, что и у заданного клиента;
- для заданного объекта страховки получить пары (имя клиента и номер договора), по которым произошел страховой случай

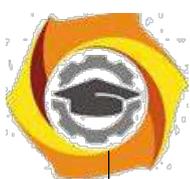
3. Модель «Поликлиника» должна содержать информацию о пациентах, докторах, диагнозах

- получить список всех пациентов с заданным диагнозом;
- получить список пациентов, которые обслуживаются тем же доктором, что и заданный пациент;
- получить список докторов-однофамильцев с заданным пациентом;
- получить список докторов, у которых лечатся пациенты не меньше заданного количества с заданным диагнозом;
- получить список пациентов, имеющих те же симптомы болезни, что и у заданного пациента;
- для заданного доктора «А» получить список его пациентов, имеющих тот же диагноз, что и выбранный пациент, лечащийся у доктора «В»



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

<p>В Модель «Отдел кадров»- должна содержать информацию о сотрудниках, подразделениях организации, должностях, официальных документах (паспорт, диплом, военный билет и т.п.)</p>	<ul style="list-style-type: none"><li>- получить список всех сотрудников, работающих в заданном подразделении;</li><li>- получить список сотрудников, работающих под руководством того же начальника, что и у выбранного сотрудника;</li><li>- получить список всех однофамильцев заданного начальника подразделения;</li><li>- для заданного подразделения получить пары (Ф.И.О. сотрудника, № военного билета), сотрудников, работающих в заданной должности;</li><li>- получить список всех подразделений, в которых работают сотрудники с ученой степенью (на основании диплома);</li><li>- получить список сотрудников, состоящий из троек (Ф.И.О. сотрудника, № паспорта, должность), пенсионного возраста. (Обратить внимание на то, что у мужчин и женщин пенсионный возраст разный)</li></ul>
<p>5. Модель «Фильмотека» должна содержать информацию о кинофильмах, актерах, киностудиях и контрактах между актерами и киностудиями</p>	<ul style="list-style-type: none"><li>- получить список всех актеров, снимающихся на заданной киностудии;</li><li>- получить список кинофильмов, в которых снимаются те же актеры, что и в заданном кинофильме;</li><li>- получить пары (Ф.И.О. актера, № контракта), занятых в фильмах, выпущенных на заданной киностудии;</li><li>- получить список киностудий, в которых были сняты фильмы по заданной тематике и в заданном году;</li><li>- получить список актеров, у которых имеется контракт с киностудией, расположенной в том же городе, в каком проживают эти актеры;</li><li>- получить список кинофильмов, в которых заданный актер не снялся ни разу</li></ul>
<p>6. Модель «Роддом» должна содержать информацию о родившихся детях, их матерях, докторам, курирующих ребенка</p>	<ul style="list-style-type: none"><li>- получить список всех матерей, у которых родились девочки;</li><li>- получить список матерей – однофамилиц, с заданным врачом;</li><li>- выбрать докторов, у которых пациентами являются дети, родившиеся в тот же день, что и у выбранной матери;</li><li>- получить список докторов, у которых пациентами являются близнецы;</li><li>- получить список матерей, у которых роди-</li></ul>



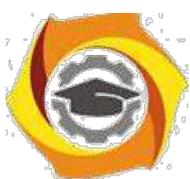
**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

	<p>лись мальчики, на определенную дату; — для заданного доктора получить пары (мать, ребенок), у которых пол совпадает</p>
<p>7. Модель «Заказ билетов» должна содержать информацию о пассажире, билете, рейсе самолета</p>	<ul style="list-style-type: none"><li>- получить список всех пассажиров заданного рейса;</li><li>- получить список пассажиров, вылетающих рейсами до заданного города и имеющих билет на заданное место;</li><li>- получить список рейсов, на которых вылетали пассажиры с заданными Ф.И.О. и датой вылета;</li><li>- получить список, состоящий из пар (№ рейса, № места, на которые не проданы билеты);</li><li>- получить список пассажиров, вылетавших до заданного города в заданном интервале дат;</li><li>- получить список типов самолетов, летающих по тому же маршруту, что и самолет, на котором летел пассажир с заданным номером билета</li></ul>
<p>8. Модель «Склад» должна содержать информацию о складе, товаре, кладовщике, документе (накладная на прием товара, акт на отпуск товара)</p>	<ul style="list-style-type: none"><li>- получить список всех товаров, находящихся на всех складах;</li><li>- получить список кладовщиков, принимавших заданный товар;</li><li>- получить пары (название товара, № акта), отпущенные с заданного склада в заданный день товаров;</li><li>- получить список складов, на которые был осуществлен прием тех же товаров, что и на заданный склад;</li><li>- получить номера всех накладных на прием тех же товаров, что и принимал заданный кладовщик;</li><li>- получить список (наименование товара, номер склада, Ф.И.О. кладовщика), принятых по заданной накладной</li></ul>



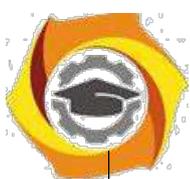
**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

<p>9. Модель «Предприятие» должна содержать информацию о поставщиках, производимых изделиях (составе изделия), потребителях продукции, товарообороте</p>	<ul style="list-style-type: none"><li>- получить список поставщиков заданной детали;</li><li>- получить список всех изделий, в которое входит заданная деталь от заданного поставщика;</li><li>- получить список потребителей изделия, в которые входит та же деталь, что и производится заданным поставщиком;</li><li>- получить список изделий, у которых поставщик деталей находится в том же городе, что и потребитель;</li><li>- получить список потребителей изделия, в которое входит одно и то же количество деталей, производимых разными поставщиками;</li><li>- получить все номера деталей, которые поставляются только одним поставщиком</li></ul>
<p>10. Модель «Издательство» должна содержать информацию об авторах, изданиях, видах рукописей</p>	<ul style="list-style-type: none"><li>- получить список всех изданий, выпускаемых заданным издательством;</li><li>- получить список авторов, которые публикуют заданный вид рукописи в заданном издательстве;</li><li>- получить список изданий, выпущенных в том же издательстве, в котором публиковался автор, проживающий в том же городе, что и автор с заданной фамилией;</li><li>- получить список издательств и авторов, расположенных в одном и том же городе;</li><li>- получить список авторов, опубликовавших одну и ту же рукопись в разных издательствах;</li><li>- получить пары (автор, издательство) для рукописей, изданных за заданный период времени</li></ul>
<p>11. Модель «Плановый отдел» должна содержать информацию о материальных ценностях, подразделениях организации, заявках, поставщиках материальных ценностей</p>	<ul style="list-style-type: none"><li>- получить список всех мат.ценностей (МЦ), требуемых заданному подразделению;</li><li>- получить список всех подразделений, которым требуется та же МЦ, что и заданному подразделению;</li><li>- получить список пар (МЦ, поставщик) таких, у которых стоимость МЦ не больше заданной;</li><li>- получить список пар (МЦ, количество), требуемых подразделениям организации, находящимся в том же городе, что и заданное подразделение;</li><li>- получить номера заявок от подразделений, которым требуется заданная МЦ;</li></ul>



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

	<ul style="list-style-type: none"><li>- получить список поставщиков МЦ, находящихся в той же стране, что и подразделение организации, которой эта МЦ требуется</li></ul>
12. Модель «Канцелярия» должна содержать информацию о типах документов (входящие, исходящие), собственно о документе, адресате и корреспонденте	<ul style="list-style-type: none"><li>- получить список всех исходящих документов за период;</li><li>- получить список всех корреспондентов, которым пришли письма в тот же день, что и заданному корреспонденту;</li><li>- получить номера входящих документов от корреспондентов, проживающих в том же городе, что и адресат;</li><li>- получить пары (номер входящего документа, номер исходящего), у которых темы одинаковы;</li><li>- получить список всех адресатов, которым были направлены письма по той же теме, что и заданному адресату;</li><li>- получить список (корреспонденты, номер входящего письма), ответы которым должны быть подготовлены к заданной дате.</li></ul>
13. Модель «Налоговая инспекция» должна содержать информацию о зарегистрированных налогоплательщиках, видах налогов, начисленных и перечисленных суммах налогов	<ul style="list-style-type: none"><li>- получить список всех налогоплательщиков, начисляющих заданный вид налога;</li><li>- получить список налогоплательщиков, переплативших по заданному виду налога;</li><li>- получить список налогов, по которым произошла недоплата;</li><li>- получить суммы начисленных налогов налогоплательщиками, которые перечислили деньги в тот же день, что и заданный налогоплательщик;</li><li>- получить список налогоплательщиков, не перечисливших деньги к заданной дате;</li><li>- получить список налогоплательщиков, у которых начисленная сумма по заданному виду налога лежит в заданном диапазоне.</li></ul>
14. Модель «Филателистическое агентство» должна содержать информацию о видах филателистической продукции, управлениях федеральной почтовой связи (УФПС), собственно о филателистической продукции, заявках от УФПС	<ul style="list-style-type: none"><li>- получить список всех УФПС, которым необходим заданный вид филат.продукции;</li><li>- получить список УФПС, которые заказали ту же фил.продукцию, что и заданное УФПС;</li><li>- получить номера заявок от УФПС, которым требуется заданное количество заданной фил.продукции;</li><li>- получить пары (вид фил.продукции, количество) для УФПС, расположенного в заданном</li></ul>



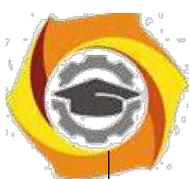
Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

	регионе
15. Модель «Завод» должна содержать информацию о поставщиках, деталях и проектах	<ul style="list-style-type: none"><li>- получить все номера деталей, поставляемых поставщиком из заданного города;</li><li>- получить номера деталей, поставляемых для всех проектов, обеспечиваемых поставщиком из того же города, где размещен проект;</li><li>- получить список всех поставщиков, поставляющих одну и ту же деталь для всех проектов;</li><li>- получить список проектов, использующих, по крайней мере, одну деталь, производимую заданным поставщиком;</li><li>- получить список троек (номер детали, количество, поставщик), аналогичных деталям, поставляемым заданным поставщиком;</li><li>- получить список проектов, которые обеспечиваются деталями от одного поставщика</li></ul>
16. Модель «Театр» должна содержать информацию о театре, спектакле, ведущих артистах в спектакле	<ul style="list-style-type: none"><li>- получить список ведущих артистов всех театров;</li><li>- получить список спектаклей, в которых занят заданный артист;</li><li>- получить список театров, в которых играют однофамильцы;</li><li>- получить пары (название театра, город), в котором идут те же спектакли, что и спектакль, в котором занят заданный артист;</li><li>- получить список спектаклей, в которых участвуют артисты, живущие в том же городе, что и театр;</li><li>- получить пары (название театра, спектакль), в которых играют артисты с той же фамилией, что и автор пьесы</li></ul>



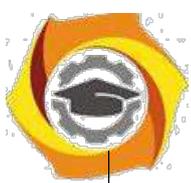
**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

<p>45. Модель «Собачий питомник» должна содержать информацию о собаке, ее родителях, ее хозяевах</p>	<p>- получить список собак заданной породы; - получить адреса хозяев собаки с заданными родителями; - получить список хозяев собак заданной породы и заданного окраса; - получить пары (имя хозяев, телефон) всех собак женского пола, у которых «папа» был тем же самым, что и у заданной собаки; - получить список пород собак, хозяева которых проживают на одной улице; - получить список имен собак заданной породы, у которых родители были чемпионами в заданном году</p>
<p>18. Модель «Швейная компания» должна содержать информацию о моделях одежды, об используемых тканях, фурнитуре, о поставщиках ткани и фурнитуры</p>	<p>- получить список всех поставщиков, производящих заданную ткань; - получить список моделей одежды, у которых поставщики ткани и фурнитуры расположены в одном городе; - получить пары(номер модели, номер изделия фурнитуры), входящих в модель одежды, сшитую из заданной ткани; - получить список всех поставщиков, поставляющих ткань и фурнитуру для моделей заданного сезона; - получить пары (наименование фурнитуры, количество), входящих в модели заданного сезона и заданной ткани; - получить список номеров моделей одежды, сшитых из ткани заданного цвета</p>
<p>19. Модель «Торговая компания» должна содержать информацию о товарах, магазинах, поставщиках, покупателях</p>	<p>- получить список всех товаров, продаваемых в заданном магазине; - получить список товаров, поставщик которых находится в том же городе, что и магазин, их продающий; - получить список всех потребителей, купивших товары стоимостью не больше заданной; - получить список поставщиков, которые поставили заданный товар в магазин с заданным адресом; - получить пары (номер товара, название поставщика), продаваемых через магазины, в которых обслуживается заданный потребитель; - получить список пар (магазин, адрес), в которых продается заданный товар от заданного</p>



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

	поставщика
<p>20. Модель «ГАИ» должна содержать информацию о зарегистрированных автомобилях, их владельцах, выданных документах на автомобиль</p>	<ul style="list-style-type: none"><li>- получить список всех автомобилей, зарегистрированных в заданном году;</li><li>- получить список владельцев автомобиля заданной марки;</li><li>- получить список номеров свидетельств о регистрации АМТС той же марки и того же цвета, что и автомобиль, принадлежащий заданному владельцу;</li><li>- получить список пар (марка автомобиля, гос.номер), принадлежащих заданному владельцу и зарегистрированному не позднее заданного года;</li><li>- получить список троек (Ф.И.О. владельца, марка автомобиля, гос.номер) автомашин, выпущенных после заданного года и зарегистрированных в заданном отделении ГАИ;</li><li>- получить список адресов владельцев автомобилей, зарегистрированных в том же отделении ГАИ и в том же году, что и автомобиль заданного владельца</li></ul>
<p>21. Модель «Политклуб» должна содержать информацию о членах клуба; образовании; членстве в общественных организациях и представительных органах гос.власти и местного самоуправления; мероприятиях клуба; выполнении персональных поручений</p>	<ul style="list-style-type: none"><li>- получить список членов политклуба по определённой форме;</li><li>- получить список членов клуба, участвующих в заданном исполнительном органе государственной власти;</li><li>- получить список членов клуба, имеющих ту же награду, что и заданный член клуба;</li><li>- определить общую сумму взносов за заданный период времени;</li><li>- получить список невыполненных поручений и ответственных за них</li></ul>



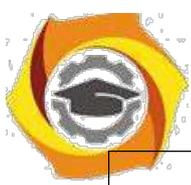
**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

<p>43. Модель «Салон мобильных телефонов» должна содержать информацию о товарах и складах</p>	<ul style="list-style-type: none"><li>- список товаров на складе в заданном районе;</li><li>- список товаров на складе заданном ценовом диапазоне;</li><li>- список товаров на складе, которые будут в наличии в течение 5 дней;</li><li>- список складов с минимальной ценой определенного товара;</li><li>- список складов с максимальной ценой определенного товара;</li><li>- список товаров в магазине, приобретенных на определенном складе</li></ul>
<p>23. Модель «Сервис» должна содержать информацию об услугах и клиентах</p>	<ul style="list-style-type: none"><li>- перечень имеющихся услуг в заданном ценовом диапазоне;</li><li>- перечень оказанных услуг в определённом районе (городе);</li><li>- список клиентов, заказавших определённую услугу;</li><li>- список услуг, которые оказываются в настоящее время;</li><li>- минимальное и максимальное количество оказанных услуг в определённый промежуток времени;</li><li>- список услуг, предоставленных в заданный период времени</li></ul>
<p>24. Модель «Заводы» должна содержать информацию о заводе, основном виде деятельности по Общероссийскому классификатору видов экономической деятельности(ОКВЭД), продукции, покупателях</p>	<ul style="list-style-type: none"><li>- список заводов в заданном районе;</li><li>- список заводов по ОКВЭД;</li><li>- список товаров, приобретенных на определенном заводе;</li><li>- список организаций-покупателей, покупающих определенные товары;</li><li>- список заводов, производящих продукцию на экспорт</li></ul>
<p>25. Модель «Отель» должна содержать информацию о номерах, стоимости, клиентах, периоде проживания</p>	<ul style="list-style-type: none"><li>- перечень свободных номеров, в том числе по классам;</li><li>- список гостей, проживающих в номерах VIP класса;</li><li>- список гостей, приехавших из определенной страны/города;</li><li>- список и количество номеров, выдаваемых каждый день;</li><li>- минимальная и максимальная количество свободных/не свободных номеров в определенный период времени;</li></ul>



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

	<ul style="list-style-type: none"><li>- стоимость проживания в номере в зависимости от класса</li></ul>
26. Модель «Кабельное телевидение» должна содержать информацию об услуге, клиентах, сотрудниках, стоимости	<ul style="list-style-type: none"><li>- перечень имеющихся услуг в заданном ценовом диапазоне;</li><li>- перечень услуг в заданном районе;</li><li>- список клиентов о сопровождающих сотрудниках;</li><li>- доход за определенный период;</li><li>- популярность отдельных услуг</li></ul>
27. Модель «Больница» должна содержать информацию о пациентах, докторах, диагнозах, загруженности стационара	<ul style="list-style-type: none"><li>- список всех пациентов с заданным диагнозом;</li><li>- загруженность стационара по отделениям;</li><li>- список пациентов, которые находятся на лечении больше заданного времени;</li><li>- список лечащих врачей, закреплённых за пациентами с заданными диагнозами;</li><li>- список докторов, у которых лечатся пациенты больше заданного времени;</li><li>- список пациентов, имеющих те же симптомы болезни, что и у заданного пациента;</li><li>- для заданного доктора «А» список его пациентов, имеющих тот же диагноз, что и выбранный пациент, лечащийся у доктора «В»</li></ul>
28. Модель "Строительно-монтажное предприятие" должна содержать информацию о работах, заказчиках, сроках работ, бригадах, нормативах, стоимости работ	<ul style="list-style-type: none"><li>- продолжительность работ на конкретном объекте;</li><li>- задержки в выполнении работ;</li><li>- занятости бригад на конкретном объекте;</li><li>- перечень бригад, выполнивших работы за период;</li><li>- затратах стройматериалов;</li><li>- доход по конкретному объекту</li></ul>
29. Модель "Ресторан" должна содержать сведения о расчете затрат приготовления блюд из ингредиентов, прибыли, финансовом мониторинге	<ul style="list-style-type: none"><li>- заказы, которые были выполнены или выполняются сегодня;</li><li>- расход продуктов по всем выполненным заказам;</li><li>- затраты на продукты по всем выполненным заказам за период времени;</li><li>- общая стоимость всех блюд в продажах;</li><li>- прибавочная стоимость без вычета средств на зарплату, накладных расходов и налогов и другие запросы</li></ul>



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

61. Модель "Нефтепереработка" должна содержать сведения о цехах, продукции, персонале, выработанной продукции, выбросах

- 49. выработанной продукции за период;
- 50. количестве полученного сырья;
- 51. проданной продукции;
- 52. доходе за период;
- 53. пропорции продукции по наименованиям;
- 54. начальнике смены, который ответственен за конкретную партию топлива



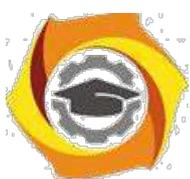
**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Методические рекомендации для студентов по организации и выполнению самостоятельной работы по модулю "Методы повышения эффективности производства".

Самостоятельная работа является неотъемлемой составляющей образовательного процесса. Самостоятельная работа магистрантов включает изучение теоретического курса и подготовку к практическим и лабораторным занятиям, выполнение домашнего задания, подготовка к зачетам. Настоящие Методические рекомендации для студентов по организации и выполнению самостоятельной работы по модулю "Методы повышения эффективности производства" относятся к виду учебной работы «Изучение теоретического курса и подготовка к зачету». Самостоятельная работа магистрантов также включает все виды текущей аттестации.

***Тематика самостоятельной работы***

№	Наименование работы
26.	Изучение материалов лекции: Модели и методы принятия решений. Принятие решений в условиях неопределенности. Принятие решений в условиях риска. Модели и методы принятия решений при нечеткой информации. Теоретические основы выбора альтернатив. Марковские модели принятия решений. Метод анализа иерархий.
27.	Подготовка к практическому занятию: Анализ решений (в условиях неопределенности). Анализ решений (в условиях риска).
28.	Подготовка к практическому занятию: Операции над нечеткими отношениями и множествами. Определение оптимального решения при конечном горизонте планирования методом итераций по стратегиям. Метод анализа иерархий. Решение задач принятия решений на языке бинарных отношений
29.	Подготовка к зачету по теме: Принятие инженерных решений
30.	Изучение материалов лекции: Введение в управление проектом. Управление временем проекта. Анализ хода работ, управление стоимостью проекта. Управление рисками проекта. Оценка и обоснование проекта.
31.	Подготовка к практическому занятию: Понятийный аппарат в управлении проектами. Модель «дуга – работа» (activity-on-arrow). Модель «узел – работа» (activity-on-node). Расчёт сетевой модели.
32.	Подготовка к практическому занятию: Проекты, ограниченные по времени. Проекты, ограниченные по ресурсам. Минимизация затрат. Формулировка задачи линейного программирования для данной модели. Имитационное моделирование. Средняя норма прибыли на инвестиции. Метод оценки по периоду окупаемости (payback method).
33.	Подготовка к зачету по теме: Проектное управление
34.	Изучение материалов лекции: Кадровый менеджмент в системе управления производственной организацией. Методы и стили управления производственным коллективом. Роль корпоративной (организационной) культуры в управлении производственным коллективом.
31.	Подготовка к практическому занятию: Функции кадрового менеджмента в управлении производственной организацией. Кадровая политика и кадровые процедуры производственной организации Подготовка к практическому занятию: Административные методы управления
32.	Обеспечение безопасности труда. СОУТ. Социально-психологические методы управления. Экономические методы управления. Стили управления производственным коллективом. Профилактика производственных конфликтов. Подготовка



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

	персонала к внедрению инноваций. Управление корпоративной (организационной) культурой
11.	Подготовка к зачету по теме: Управление производственным коллективом

***Самостоятельная работа № 1***

*Тема:* Изучение материалов лекции: Модели и методы принятия решений. Принятие решений в условиях неопределенности. Принятие решений в условиях риска. Модели и методы принятия решений при нечеткой информации. Теоретические основы выбора альтернатив. Марковские модели принятия решений. Метод анализа иерархий.

*Цель:*

- повторение материалов лекции.
- подготовка к выполнению кейсов.

*Изучение литературных и электронных источников:*

24. Модели и методы принятия решений.
25. Принятие решений в условиях неопределенности.
26. Принятие решений в условиях риска.
27. Модели и методы принятия решений при нечеткой информации.
28. Теоретические основы выбора альтернатив.
29. Марковские модели принятия решений.
30. Метод анализа иерархий.

*Практическое задание.* Выполнение домашнего задания

***Самостоятельная работа № 2***

*Тема:* Подготовка к практическому занятию: Анализ решений (в условиях неопределенности). Анализ решений (в условиях риска).

*Цель:*

- отработка навыка решения практических задач.
- подготовка к выполнению кейсов.

*Изучение литературных и электронных источников:*

- Анализ решений (в условиях неопределенности).
- Анализ решений (в условиях риска).

*Практическое задание.* Выполнение домашнего задания

***Самостоятельная работа № 3***

*Тема:* Подготовка к практическому занятию: Операции над нечеткими отношениями и множествами. Определение оптимального решения при конечном горизонте планирования методом итераций по стратегиям. Метод анализа иерархий. Решение задач принятия решений на языке бинарных отношений.

*Цель:*

- отработка навыка решения практических задач.
- подготовка к выполнению кейсов.

*Изучение литературных и электронных источников:*

- Операции над нечеткими отношениями и множествами.
- Определение оптимального решения при конечном горизонте планирования методом итераций по стратегиям.
- Метод анализа иерархий.
- Решение задач принятия решений на языке бинарных отношений.

*Практическое задание.* Выполнение домашнего задания



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

***Самостоятельная работа № 4***

*Тема:* Подготовка к зачету по теме: Принятие инженерных решений *Цель:*

- отработка навыка решения практических задач.
- подготовка к выполнению зачетного задания.

*Изучение литературных и электронных источников:*

- Модели и методы принятия решений.
- Принятие решений в условиях неопределенности.
- Принятие решений в условиях риска.
- Модели и методы принятия решений при нечеткой информации.
- Теоретические основы выбора альтернатив.
- Марковские модели принятия решений.
- Метод анализа иерархий.
- Анализ решений (в условиях неопределенности).
- Анализ решений (в условиях риска).
- Операции над нечеткими отношениями и множествами.
- Определение оптимального решения при конечном горизонте планирования методом итераций по стратегиям.
- Метод анализа иерархий.
- Решение задач принятия решений на языке бинарных отношений *Практическое задание.*  
Выполнение заданий по теме " Принятие инженерных решений "

***Самостоятельная работа № 5***

*Тема:* Изучение материалов лекции: Введение в управление проектом. Управление временем проекта. Анализ хода работ, управление стоимостью проекта. Управление рисками проекта. Оценка и обоснование проекта.

*Цель:*

- повторение материалов лекции.
- подготовка к выполнению кейсов.

*Изучение литературных и электронных источников:*

- Введение в управление проектом.
- Управление временем проекта.
- Анализ хода работ, управление стоимостью проекта.
- Управление рисками проекта.
- Оценка и обоснование проекта.

*Практическое задание.* Выполнение домашнего задания

***Самостоятельная работа № 6***

*Тема:* Подготовка к практическому занятию: Понятийный аппарат в управлении проектами. Модель «дуга – работа» (activity-on-arrow). Модель «узел – работа» (activity-on-node). Расчёт сетевой модели.

*Цель:*

- отработка навыка решения практических задач.
- подготовка к выполнению кейсов.

*Изучение литературных и электронных источников:*

- Понятийный аппарат в управлении проектами.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Модель «дуга – работа» (activity-on-arrow).
- Модель «узел – работа» (activity-on-node).
- Расчёт сетевой модели.

*Практическое задание.* Выполнение домашнего задания

***Самостоятельная работа № 7***

*Тема:* Подготовка к практическому занятию: Проекты, ограниченные по времени. Проекты, ограниченные по ресурсам. Минимизация затрат. Формулировка задачи линейного программирования для данной модели. Имитационное моделирование. Средняя норма прибыли на инвестиции. Метод оценки по периоду окупаемости (payback method).

*Цель:*

- отработка навыка решения практических задач.
- подготовка к выполнению кейсов.

*Изучение литературных и электронных источников:*

- Проекты, ограниченные по времени.
- Проекты, ограниченные по ресурсам.
- Минимизация затрат.
- Формулировка задачи линейного программирования для данной модели.
- Имитационное моделирование.
- Средняя норма прибыли на инвестиции.
- Метод оценки по периоду окупаемости (payback method).
- Решение задач принятия решений на языке бинарных отношений.

*Практическое задание.* Выполнение домашнего задания

***Самостоятельная работа № 8***

*Тема:* Подготовка к зачету по теме: Проектное управление

*Цель:*

- отработка навыка решения практических задач.
- подготовка к выполнению зачетного задания.

*Изучение литературных и электронных источников:*

- Введение в управление проектом.
- Управление временем проекта.
- Анализ хода работ, управление стоимостью проекта.
- Управление рисками проекта.
- Оценка и обоснование проекта.
- Понятийный аппарат в управлении проектами.
- Модель «дуга – работа» (activity-on-arrow).
- Модель «узел – работа» (activity-on-node).
- Расчёт сетевой модели.
- Проекты, ограниченные по времени.
- Проекты, ограниченные по ресурсам.
- Минимизация затрат.
- Формулировка задачи линейного программирования для данной модели.
- Имитационное моделирование.
- Средняя норма прибыли на инвестиции.
- Метод оценки по периоду окупаемости (payback method).
- Решение задач принятия решений на языке бинарных отношений.

*Практическое задание.* Выполнение заданий по теме "Проектное управление "



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

***Самостоятельная работа № 9***

*Тема:* Изучение материалов лекции: Кадровый менеджмент в системе управления производственной организацией. Методы и стили управления производственным коллективом. Роль корпоративной (организационной) культуры в управлении производственным коллективом.

*Цель:*

- повторение материалов лекции.
- подготовка к выполнению кейсов.

*Изучение литературных и электронных источников:*

- Кадровый менеджмент в системе управления производственной организацией.
- Методы и стили управления производственным коллективом.
- Роль корпоративной (организационной) культуры в управлении производственным коллективом.

*Практическое задание.* Выполнение домашнего задания

***Самостоятельная работа № 10***

*Тема:* Подготовка к практическому занятию: Функции кадрового менеджмента в управлении производственной организацией. Кадровая политика и кадровые процедуры производственной организации

*Цель:*

- отработка навыка решения практических задач.
- подготовка к выполнению кейсов.

*Изучение литературных и электронных источников:*

- Функции кадрового менеджмента в управлении производственной организацией
- Кадровая политика и кадровые процедуры производственной организации

*Практическое задание.* Выполнение домашнего задания

***Самостоятельная работа № 11***

*Тема:* Подготовка к практическому занятию: Административные методы управления. Обеспечение безопасности труда. СОУТ. Социально-психологические методы управления. Экономические методы управления. Стили управления производственным коллективом. Профилактика производственных конфликтов. Подготовка персонала к внедрению инноваций. Управление корпоративной (организационной) культурой

*Цель:*

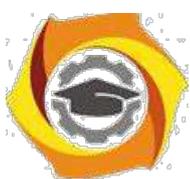
- отработка навыка решения практических задач.
- подготовка к выполнению кейсов.

*Изучение литературных и электронных источников:*

- Административные методы управления.
- Обеспечение безопасности труда. СОУТ.
- Социально-психологические методы управления.
- Экономические методы управления.
- Стили управления производственным коллективом.
- Профилактика производственных конфликтов.
- Подготовка персонала к внедрению инноваций.
- Управление корпоративной (организационной) культурой

*Практическое задание.* Выполнение домашнего задания

***Самостоятельная работа № 12***



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*Тема:* Подготовка к зачету по теме: Управление производственным коллективом  
*Цель:*

- отработка навыка решения практических задач.
- подготовка к выполнению зачетного задания.

*Изучение литературных и электронных источников:*

- Кадровый менеджмент в системе управления производственной организацией.
- Методы и стили управления производственным коллективом.
- Роль корпоративной (организационной) культуры в управлении производственным коллективом.
- Функции кадрового менеджмента в управлении производственной организацией
- Кадровая политика и кадровые процедуры производственной организации
- Административные методы управления.
- Обеспечение безопасности труда. СОУТ.
- Социально-психологические методы управления.
- Экономические методы управления.
- Стили управления производственным коллективом.
- Профилактика производственных конфликтов.
- Подготовка персонала к внедрению инноваций.
- Управление корпоративной (организационной) культурой

*Практическое задание.* Выполнение заданий по теме "Управление производственным коллективом "

Методические рекомендации для студентов по организации и выполнению самостоятельной работы по модулю " Факультативный модуль Английский язык для специалистов и руководителей энергослужб предприятий".

Самостоятельная работа является неотъемлемой составляющей образовательного процесса. Самостоятельная работа магистрантов включает изучение теоретического курса и подготовку к практическим и лабораторным занятиям, выполнение домашнего задания, подготовка к зачетам, экзаменам. Настоящие Методические рекомендации для студентов по организации и выполнению самостоятельной работы по модулю " Факультативный модуль Английский язык для специалистов и руководителей энергослужб предприятий" относятся к виду учебной работы «Изучение теоретического курса и подготовка к экзамену». Самостоятельная работа магистрантов также включает все виды текущей аттестации.

***Тематика самостоятельной работы***

№	Наименование работы
1	Изучение практического и теоретического материала, выполнение домашних заданий для расширения лексики и грамматических упражнений, предусмотренных в данном семестре (3)
2	Подготовка к зачету
3	Изучение практического и теоретического материала, выполнение домашних заданий для расширения лексики и грамматических упражнений, предусмотренных в данном семестре (4)
4	Подготовка к зачету
5	Изучение практического и теоретического материала, выполнение домашних заданий для расширения лексики и грамматических упражнений, предусмотренных в данном семестре (5)
6	Подготовка к зачету
7	Изучение практического и теоретического материала, выполнение домашних заданий для расширения лексики и грамматических упражнений, предусмотренных в



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

	данном семесте (6)
8	Подготовка к зачету
9	Изучение практического и теоретического материала, выполнение домашних заданий для расширения лексики и грамматических упражнений, предусмотренных в данном семесте (7)
10	Подготовка к зачету
11	Изучение практического и теоретического материала, выполнение домашних заданий для расширения лексики и грамматических упражнений, предусмотренных в данном семесте (8)
12	Подготовка к зачету
13	Изучение практического и теоретического материала, выполнение домашних заданий для расширения лексики и грамматических упражнений, предусмотренных в данном семесте (9)
в	Подготовка к экзамену

***Самостоятельная работа № 1,3,5,7,9,11,13***

*Тема:* Подготовка к практическому занятию: Знакомство, вводная презентация + Noun and its plural – существительное и множественное число. Личные качества и компетенции ин-женера (Traits and competencies of an engineer) + Артикли a/an и the. Инженерное образование



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

(An engineer's education) + Nouns: countables/uncountables, some/any, little/few, much/many, a lot of. Презентация информации (Presenting information) + Present Simple – настоящее простое время;

Подготовка к практическому занятию: Таблицы и графики (Tables and graphs) + There is/are. Система измерения и чертежи (System of units, dimensions and drawings) + Making comparison – сравнение. Числа (Working with numbers) + Present Continuous – настоящее продолженное время. Материалы и Свойства (Material and properties) + Сопоставление Present Simple и Present Continuous. Инструменты (Tools) + Глаголы, не имеющие форм группы Continuous.;

Подготовка к практическому занятию: Техника безопасности (Safety Precautions) + Past Simple – прошедшее простое время. Рынок, работа с заказчиками и производителями, оформление заказа (Markets, customers and producers, making an order) + Past Continuous. Сопостовление (Interview) + Сопоставление Past Continuous и Past Simple. Деловая переписка (Business correspondence) + Present Perfect – настоящее совершенное время.;

Подготовка к практическому занятию: Деловая поездка (Business trip) + Сопоставление Past Simple и Present Perfect. Деловые переговоры (Business meeting) + Future Simple – будущее простое время. Составление отчетов, решение проблем (Reporting and problem solving) + Be Going To. Вопросы защиты окружающей среды (Protecting the environment) + Modal verbs – модальные глаголы. Будущее энергетики (The future of energy) + The Passive – страдательный залог *Практическое задание*. выполнение упражнений на закрепление грамматического материала

Примечание: Примерные упражнения для самостоятельных работ с 1-18 представлены в сборнике «Методические указания по грамматике английского языка» (Приложение 1)

***Самостоятельная работа № 1,3,5,7,9,11,13***

*Тема:* Выполнение домашних работ по теме: Английский язык для специалистов и руководителей энергослужб предприятий

Темы домашних работ:

16. Подготовить краткую презентацию о себе и своей компании на английском языке, выучить новые слова по теме First Business Card, Business Card, прочитать и перевести статью Working in a foreign country
17. Подготовить на английском языке сообщение о качествах идеального инженера, выучить новые слова по теме: What is a perfect engineer? и цитаты. Прочитать и перевести статью Top qualities of a great engineer.
29. Подготовить на английском языке сообщение о командной работе, прочитать и перевести статью Teams with bright ideas, выучить новые слова по теме Teamwork.
30. Проработать пример составления резюме, подготовить на английском языке сообщение по теме “Проведение собеседования с кандидатом на вакансию специалиста энергослужбы, принятие решения и объяснение своего выбора». Выучить новые слова и выражения.
31. Подготовить на английском языке сообщение по теме «Presenting information», выучить новые слова.
32. На примере выступлений бизнес-кейса круглый стол по Устойчивой энергетике Форума по изменению климата в Париже, декабрь 2015 г. подготовить вопросы на английском языке к выступающим.
33. Подготовить на английском языке описание графика, тренда или диаграммы, выучить новые слова по теме “Tables and graphs. Discussing readings and trends”, прочитать и перевести статью “Production Progress Report”.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

34. Подготовить на английском языке сообщение о геометрических формах и размерах определенного объекта или компонента оборудования, выучить новые слова по теме “Describing



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

shapes and qualities” на примере различных стандартных конфигураций вилок и розеток в разных странах. Научиться задавать вопросы и отвечать о размерах и формах предметов, объектов и оборудования. Прочитать и перевести статью “The Roman Coliseum”.

Подготовить на английском языке сообщение о размещении оборудования на общем плане цеха, о габаритных размерах, высоте отметок, поперечных размерах, используемых единицах измерения, выучить новые слова по теме “System of units, dimensions and drawings”.

Научиться правильно использовать числа в речи и на письме, выучить валюты различных стран, выучить новые слова и выражения по теме “Business costs”, прочитайте и переведите статью «The business of giving”.

Подготовить на английском языке сообщение о принятии решения о выборе материалов оборудования и инженерных коммуникаций. Выучить слова по теме “Materials and properties”, прочитайте и переведите статью “Construction Basics: Understanding Your Materials”.

Подготовить на английском языке сообщение об учете энергоресурсов значимыми энерго-носителями. Выучить новые слова по теме «Instruments and Tools”.

Выучить пиктограммы и их названия, предупреждающие фразы по теме “Safety Precautions”.

Составить письмо-заказ на оборудование. Выучить слова по теме «Markets, customers and producers, making an order». Прочитайте и переведите статью «Unusual markets. Playtime.”

Написать письмо иностранному партнеру в соответствии с конкретным для каждого студента бизнес-кейсом. Выучить слова и выражения по теме “Business correspondence”.

Составить три диалога на английском языке по темам: «Airport», «Hotel», «Cafe». Выучить новые слова и выражения по теме «Business trip».

Подготовить на английском языке отчет к приезду иностранной делегации в соответствии с конкретным для каждого студента бизнес-кейсом. Выучить новые слова и выражения по теме “Business meeting”. Прочитайте и переведите статью “Project analysis”.

Подготовить на английском языке краткий отчет о проведении энергоаудита, выучить слова и выражения по теме “Reporting and problem solving”. Прочитайте и переведите статью “Lighting up the world”.

Подготовить на английском языке краткое сообщение о том, какой вклад каждый из нас может внести в вопрос защиты окружающей среды. Выучить новые слова и выражения по теме “Protecting the environment”. Прочитайте и переведите статью “The Kyoto Protocol”.

Бизнес-кейс по теме “The future of energy”.

***Самостоятельная работа № 2, 4, 6, 8, 10, 12***

*Тема:* Подготовка к зачету

*Цель:*

- отработка навыка решения практических задач.
- подготовка к выполнению зачетного задания.

*Изучение литературных и электронных источников:*

с Английский язык для специалистов и руководителей энергослужб предприятий.

*Практическое задание.* Выполнение заданий по темам модуля "Факультативный модуль Английский язык для специалистов и руководителей энергослужб предприятий"

***Самостоятельная работа № 14***

*Тема:* Подготовка к зачету

*Цель:*

- отработка навыка решения практических задач.
- подготовка к выполнению экзаменационного задания.

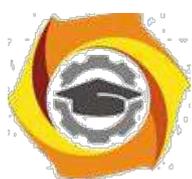
*Изучение литературных и электронных источников:*

42. Английский язык для специалистов и руководителей энергослужб предприятий. *Практическое задание.* Выполнение заданий по темам модуля

"Факультативный модуль Английский язык для специалистов и руководителей энергослужб предприятий"



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

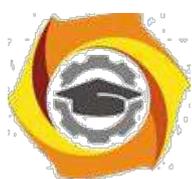
ПРИЛОЖЕНИЕ 1

**Методические указания по грамматике английского языка**

**Верхняя Пышма 2015**

**Содержание**

Введение.....	4
Unit 1: Present Simple.....	5
Unit 2: Present Continuous.....	9
Unit 3: Сопоставление Present Simple и Present Continuous.....	13
Unit 4: Глаголы, не имеющие форм группы Continuous.....	15
Unit 5: Past Simple.....	17
Unit 6: Past Continuous.....	22
Unit 7: Сопоставление Past Continuous и Past Simple.....	25
Unit 8: Present Perfect.....	27
Unit 9: Сопоставление Past Simple и Present Perfect.....	32
Unit 10: Future Simple.....	36
Unit 11: Be Going To.....	41
Unit 12: The Passive.....	44
Список литературы.....	52



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

## Введение

Учебно-методический комплекс дисциплины английский язык предназначен для практических занятий направления подготовки магистров 13.04.02 «Электроэнергетика и электротехника» программы «Управление и устойчивое развитие энергохозяйства предприятий».

Одним из компонентов содержания курса иностранного языка для неязыковых вузов являются средства общения, которые предусматривают такие языковые явления как грамматические формы и конструкции. В соответствии с программой УМО в области лингвистики данные методические указания направлены на развитие и совершенствование грамматических навыков (в том числе приобретенных в школе).

Практическая цель заключается в формировании у студента способности и готовности к межкультурной коммуникации, что предполагает развитие умений опосредованного письменного (чтение, письмо) и непосредственного устного (говорение, аудирование) иноязычного общения. В связи с выше сказанным методические указания ориентированы на развитие грамматических навыков, как для чтения, так и для устного общения.

Теоретический материал излагается в виде таблиц, использования функциональных примеров. Упражнения для закрепления новых структур способствуют активному усвоению материала.

Данные методические указания могут быть использованы как для самообразования, так и для занятий под руководством преподавателя, в качестве грамматического дополнения к любому курсу английского языка соответствующего уровня.

## Unit 1: Present Simple – настоящее простое время

### Образование

Настоящее простое время образуется с помощью *подлежащего* (существительного или местоимения в именительном падеже) и *смыслового глагола*. В утверждениях к смысловому глаголу в 3-м лице единственного числа обычно прибавляется окончание *-s*. В вопросах и отрицаниях с местоимениями *I, you, we, they* используется вспомогательный глагол *do/don't*, а с местоимениями *he, she, it* – *does/doesn't*. Если смысловой глагол употребляется с *does/doesn't*, окончание *-s* к нему не прибавляется.

### Утверждение

I/You/We/They	read.
He/She/It	reads.

### Вопрос

Do	I/you/we/they	read?
Does	he/she/it	

### Отрицание

	Полная форма	Краткая форма	
I/You/We/They	do not	don't	
He/She/It	does not	doesn't	read.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*Употребление*

**Present Simple употребляется для выражения:**

- В** постоянных состояний,  
**В** повторяющихся и повседневных действий (часто со следующими наречиями: always, never, usually и т.д.),  
Mr. Gibson **is** a businessman. He **lives** in New York. (постоянное состояние)  
He **usually starts** work at 9 a.m. (повседневное действие)  
He **often stays** at the office until late in the evening. (повседневное действие)  
**В** непреложных истин и законов природы,  
The moon **moves** round the earth.  
**В** действий, происходящих по программе или по расписанию (движение поездов, автобусов и т.д.).  
The bus **leaves** in ten minutes.

*Правописание*



Большая часть глаголов в 3-м лице единственного числа приобретает окончание –s. I read – he reads



К глаголам, оканчивающимся на –ss, –sh, –ch, –x и –o, прибавляется –es.

I kiss – he kisses, I brush – he brushes, I teach – he teaches, I fix – he fixes, I go – he goes

**В** В глаголах, оканчивающихся на **согласный + у**, опускается у и прибавляется –ies. I try – he tries, I fly – he flies

**В** В глаголах, оканчивающихся на **гласный + у**, прибавляется окончание –s.

I buy – he buys

*Маркеры*

Маркерами Present Simple являются: *usually, always* и т.п., *every day/week/month/year* и т.д., *on Mondays/Tuesdays* и т.д., *in the morning/afternoon/evening, at night/the weekend* и т.д.

***Present Simple: exercises***

**1.1 Complete the sentences using one of the following:**

cause(s) close(s) drink(s) live(s) open(s) speak(s) take(s) place

- 12) Ann **speaks** German very well.
- 13) I never --- coffee.
- 14) The swimming pool --- at 9 o'clock and --- at 18.30 every day.
- 15) Bad driving --- many accidents. **causes**
- 16) My parents --- in a very at small flat. **live**
- 17) The Olympic Games --- every four years. **Take place**

**1.2 Put the verb into the correct form.**

9. Jane **doesn't drink** (not/drink) tea very often.
10. What time --- (the banks/close) in Britain? **Do the banks close**
11. 'Where --- (Martin/come) from?' 'He's Scottish.' **Does Martin come**
12. 'What --- (you/do)?' 'I'm an electrical engineer.' – **do you do**
13. It --- (take) me an hour to get to work. How long --- (it/take) you? – **take, does it take**
14. I --- (play) the piano but I --- (not/play) very well. **Play, don't play**
15. I don't understand this sentence. What --- (this word/mean)? **Does this word mean**

**1.3 Use one of the following verbs to complete these sentences. Sometimes you need the negative:**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

believe eat flow go grow make rise tell translate

15. The earth **goes** round the sun.
16. Rice **doesn't grow** in Britain.
17. The sun --- in the east. **rises**
18. Bees --- honey. - **make**
19. Vegetarians --- meat. – **don't eat**
20. An atheist --- in God. **doesn't believe**
21. An interpreter --- from one language into another. **translates**
22. A liar is someone who --- the truth. **Does not tell**
23. The River Amazon --- into the Atlantic Ocean. **flows**

**1.4 Ask Liz questions about herself and her family.**

27. You know that Liz plays tennis. You want to know how often. Ask her. How often **do you play tennis?**
28. Perhaps Liz's sister plays tennis too. You want to know. Ask Liz. --- your sister --- (- **does your sister play tennis?**)
31. You know that Liz reads a newspaper every day. You want to know which one. Ask her. --- ( **Which newspaper do you read everyday?**)
32. You know that Liz's brother works. You want to know what he does. Ask Liz. - **what does your brother do?**
35. You know that Liz goes to the cinema a lot. You want to know how often. Ask her. **How often do you go to the cinema?**
36. You don't know where Liz's mother lives. Ask Liz. --- **Where does your mother live ?**

**1.5 Complete using one of the following.**

- I apologise I insist I promise I recommend I suggest
- 13) It's a nice day. **I suggest** we go out for a walk.
  - 14) I won't tell anybody what you said. ---. **I promise**
  - 15) (in a restaurant) You must let me pay for the meal. ---. – **I insist**
  - 16) --- for what I said about you. It wasn't true and I shouldn't have said it. – **I apologize**
  - 17) The new restaurant in Hill Street is very good --- it. **I recommend**

**Unit 2: Present Continuous – настоящее продолженное время**

*Образование*

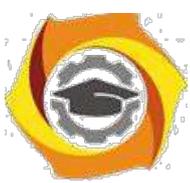
Настоящее продолженное время образуется с помощью вспомогательного глагола *to be* и смыслового глагола, к которому добавляется *-ing*.

**Утверждение**

	<b>Полная форма</b>	<b>Краткая форма</b>	
I	am	'm	
He/She/It	is	's	reading.
You/We/They	are	're	

**Вопрос**

Am	I	
Is	he/she/it	reading?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Are

you/we/they

**Отрицание**

	<b>Полная форма</b>	<b>Краткая форма</b>	
I	am not	'm not	
He/She/It	is not	isn't	reading.
You/We/They	are not	aren't	

*Употребление*

**Present Continuous употребляется для выражения:**

- C** действий, происходящих в момент речи,  
He **is reading** a book right now.
- C** временных действий, происходящих в настоящий период времени, но не обязательно в момент речи,  
She **is practicing** for a concert these days. (В данный момент она не играет. Она отдыхает.)
- C** действий, происходящих слишком часто и по поводу которых мы хотим высказать раздражение или критику (обычно со словом *always*),  
"You're **always interrupting** me!" (раздражение)
- C** действий, заранее запланированных на будущее.  
He **is flying** to Milan in an hour. (это запланировано)

*Правописание*



В глаголах, оканчивающихся на **-e**, обычно опускается **-e** и прибавляется **-ing**.  
dance – dancing, **НО** agree - agreeing



В глаголах, оканчивающихся на **ударный слог с кратким гласным** между двумя согласными, конечная согласная буква удваивается и прибавляется **-ing**.  
refer – referring, run – running, get – getting, **НО** open – opening



В глаголах, оканчивающихся на **-l**, эта буква **l** удваивается и прибавляется **-ing**. travel - travelling



В глаголах, оканчивающихся на **-ie**, опускается **-ie** и прибавляется **-y + -ing**. lie – lying, die - dying

*Маркеры*

Маркерами Present Continuous являются: *now, at the moment, these days, at present, always, tonight, still* и т.д.

**Present Continuous: exercises**

**2.1 Complete the sentences with one of the following verbs in the correct form:**

come get happen look make start stay try work

8. 'You **'re working** hard today.' 'Yes, I have a lot to do.'

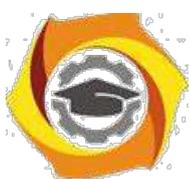
9. I --- for Christine. Do you know where she is?

10. It --- dark. Shall I turn on the light?

They haven't got anywhere to I've at the moment. They --- with friends until they find somewhere.

'Are you ready, Ann?' 'Yes, I ---.'

6. Have you got an umbrella? It --- to rain. **Is starting**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

15. You --- a lot of noise. Could you be quieter? I --- to concentrate. **Are making, am trying**  
Why are all these people here? What ---? **Is happening**

**2.2 Use the words in brackets to complete the questions.**

10. **'Is Colin working** this week?' 'No, he's on holiday.' (Colin/work)  
11. Why --- at me like that? What's the matter? (you/look) **are you looking**  
12. 'Jenny is a student at university.' 'Is she? What ---?' (she/study) **is she studying**  
13. --- to the radio or can I turn it off? (anybody/listen) **Is anybody listening**  
How is your English? --- better? (it/get) **Is it getting**

**2.3 Put the verb into the correct form. Sometimes you need the negative (I'm not doing etc.).**

13. I'm tired. **I'm going** (go) to bed now. Goodnight!  
14. We can go out now. it **isn't raining** (rain) any more.  
15. 'How is your new job?' 'Not so good at the moment. I --- (enjoy) it very much.' **'m not enjoying**  
16. Catherine phoned me last night. She's on holiday in France. She --- (have) a great time and doesn't want to come back. **'s having**  
17. I want to lose weight, so this week I --- (eat) lunch. **'m not eating**  
18. Angela has just started evening classes. She --- (learn) German. **'s learning**  
I think Paul and Ann have had an argument. They --- (speak) to each other. **Aren't speaking**

**2.4 Read this conversation between Brian and Sarah. Put the verbs into the correct form.** SARAH: Brian! How nice to see you! What (1) --- (you/do) these days?

- BRIAN: I (2) --- (train) to be a supermarket manager.  
SARAH: Really? What's it like? (3) --- (you/enjoy) it?  
BRIAN: It's all right. What about you?  
SARAH: Well, actually I (4) --- (not/work) at the moment.  
I (5) --- (try) to find a job but it's not easy.  
But I'm very busy. I (6) --- (decorate) my flat.  
BRIAN: (7) --- (you/do) it alone?  
SARAH: No, some friends of mine (8) --- (help) me.

**2.5 Complete the sentences using one of these verbs: get change rise fall increase**

You don't have to use all the verbs and you can use a verb more than once.

12. The population of the world **is rising** very fast.  
13. Ken is still ill but he --- better slowly.  
14. The world ---. Things never stay the same.  
15. The cost of living ---. Every year things are more expensive.  
16. The economic situation is already very bad and it --- worse.

**Unit 3: Сопоставление Present Simple и Present Continuous**

*Present Simple*



Present Simple употребляется для выражения постоянных состояний, повторяющихся и повседневных действий.

Claire Bryan **works** as a secretary.  
She **starts** work at eight o'clock every day.  
She **types** Mr. Moore's letters.

*Present Continuous*



Present Continuous употребляется для выражения временных действий, происходящих в момент речи или в настоящий период времени.

It's nine o'clock. Claire is still at home because she is ill.  
She is wearing her pajamas and she is sitting on her bed.  
She isn't working today.



**Present Simple or Present Continuous: exercises**

**3.1 Are the underlined verbs right or wrong? Correct the verbs that are wrong.**

6. Water boils at 100 degrees celsius. RIGHT
7. The water boils. Can you turn it off? WRONG: is boiling
8. Look! That man tries to open the door of your car. --- wrong, is trying
9. Can you hear those people? What do they talk about? --- wrong, are they talking
10. The moon goes round the earth. ---right
11. I must go now. It gets late. --- wrong, is getting
12. I usually go to work by car. --- right
13. 'Hurry up! It's time to leave.' 'OK, I come.' --- wrong, am coming
14. I hear you've got a new job. How do you get on? --- wrong, are you getting on

**3.2 Put the verb in the correct form, present continuous or present simple.**

- Let's go out. It isn't raining (not/rain) now.  
Julia is very good at languages. She speaks (speak) four languages very well.  
Hurry up! Everybody --- (wait) for you. – is waiting  
'--- (you/listen) to the radio?' 'No, you can turn it off.' – Are you listening  
'--- (you/listen) to the radio every day?' 'No, just occasionally.' – Do you listen  
The River Nile --- (flow) into the Mediterranean. - flows  
Look at the river. It --- (flow) very fast today - much faster than usual. – is flowing  
We usually --- (grow) vegetables in our garden but this year we --- (not/grow) any. –grow, are not growing  
'How is your English?' 'Not bad. It --- (improve) slowly.' Is improving  
Ron is in London at the moment. He --- (stay) at the Park Hotel. He --- (always/stay) there when he's in London. – is staying, always stays  
Can we stop walking soon? I --- (start) to feel tired. – am starting  
'Can you drive?' 'I --- (learn). My father --- (teach) me.' – am learning, is teaching  
Normally I --- (finish) work at 5.00, but this week I --- (work) until 6.00 to earn a bit more money.  
– finish, am working  
My parents --- (live) in Bristol. They were born there and have never lived anywhere else. Where  
- -- (your parents/live)? Live, do your parents live  
7. Sonia --- (look) for a place to live. She --- (stay) with her sister until she finds somewhere.  
- is looking , is staying  
8. 'What --- (your father/do)?' 'He's an architect but he --- (not/work) at the moment.'  
- does your father do, is not working  
и (at a party) Usually I --- (enjoy) parties but I --- (not/enjoy) this one very much.  
- enjoy, am not enjoying  
и The train is never late. It --- (always/leave) on time.  
  
в Jim is very untidy. He --- (always/leave) his things all over the place. - is always leaving

**3.3 Finish B's sentences. Use always ~ing (see Section B).**

1. A: I'm afraid I've lost my key again.  
к Not again! You're always losing your key.
2. A: The car has broken down again.  
That car is useless! It ---  
Is always breaking down
3. A: Look! You've made the same mistake again.



B: Oh no, not again! I ---  
Am always making the same mistake  
4. A: Oh, I've left the lights on again.  
B: Typical! You --- are always leaving the lights on

#### Unit 4: Глаголы, не имеющие форм группы Continuous

**Во временах группы Continuous обычно не употребляются глаголы:**



выражающие восприятия, ощущения (see, hear, feel, taste, smell),  
This cake **tastes** delicious.



выражающие мыслительную деятельность (know, think, remember, forget, recognize, believe,  
understand, notice, realize, seem, sound),

**I don't know** his name.

B

выражающие эмоции, желания (love, prefer, like, hate, dislike, want),  
Shirley loves jazz music.

B

include, matter, need, belong, cost, mean, own, appear, have (когда выражает принадлежность) и  
т.д.

That jacket **costs** a lot of money.

#### *Глаголы, не имеющие форм группы Continuous: exercises*

##### **4.1 Are the underlined verbs right or wrong? Correct the ones that are wrong.**

I'm seeing the manager tomorrow morning. RIGHT

I'm feeling hungry. Is there anything to eat? --- wrong, I feel hungry

Are you believing in God? --- wrong, do you believe in God? no I'm an atheist

This sauce is great. It's tasting really good. --- - wrong, it tastes really good.

I'm thinking this is your key. Am I right? --- wrong, I think

##### **4.2 Look at the pictures. Use the words in brackets to make sentences. (You should also study Unit 3 before you do this exercise.)**

K (you/not/seem/very happy today) You don't seem very happy today.

K (what/you/do?) ---what are you doing?

B (who/this umbrella/belong to?) --- To whom does this umbrella belong ?

I've no idea.

B (the dinner/smell/good) ---The dinner smells good.

B Excuse me. (anybody/sit/here?) ---Is anybody sitting here?

No, it's free

6. Can you ring me back in half an hour? (I/have/dinner) ---I'm having dinner

##### **4.3 Put the verb into the correct form, present continuous or present simple.**

- Are you hungry? Do you want something to eat? (you/want)
- Jill is interested in politics but she --- to a political party. (not/belong) does not belong to
- Don't put the dictionary away. I --- it. (use) I'm using it
- Don't put the dictionary away. I --- it. (need) I need it.
- Who is that man? What ---? (he/want) does he want
- Who is that man? Why --- at us? (he/look) is he looking
- George says he's 80 years old but nobody --- him. (believe) - believes
- She told me her name but I --- it now. (not/remember) – don't remember



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

16. I --- of selling my car. (think) Would you be interested in buying it? Am thinking  
I --- you should sell your w  
car. (think) You --- it very often. (not/use) think, don't use  
17. I used to drink a lot of coffee but these days I --- tea. (prefer) prefer  
18. Air --- mainly of nitrogen and oxygen. (consist) consists

**4.4 Complete the sentences using the most suitable form of be. Sometimes you must use the simple (am/is/are) and sometimes the continuous is more suitable (am/is/are being).**

I can't understand why he's being so selfish. He isn't usually like that.

- / Jack --- very nice to me at the moment. I wonder why. – was being
- / You'll like Jill when you meet her. She --- very nice. - is
- / Normally you are very sensible, so why --- so silly about this matter? Are you being
- / Why isn't Sarah at work today? --- ill? – Is she being

**Unit 5: Past Simple – прошедшее простое время**

*Образование*

**Правильные глаголы**

Прошедшее простое время правильных глаголов образуется путем прибавления *-ed* к смысловому глаголу. Вопросы и отрицания строятся с помощью вспомогательного глагола *did/did not (didn't)* и смыслового глагола без *-ed*.

**Неправильные глаголы**

Прошедшее простое время неправильных глаголов образуется не путем прибавления *-ed*, а другими способами. Например: leave – left, cut – cut, swim – swam (вторая форма неправильных глаголов).

Вопросы и отрицания строятся с помощью вспомогательного глагола *did/did not (didn't)* и корневой (основной) формы смыслового глагола. Например:

She watched TV last night. – **Did she watch** TV last night? – She **did not watch** TV last night.  
They left. – **Did they leave?** – They **didn't leave**.

**Утверждение**

I  
He/She/It finished (left).  
You/We/They

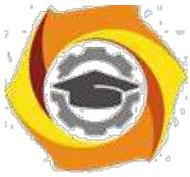
**Вопрос**

Did I  
he/she/it finish (leave)?  
you/we/they

**Отрицание**

**Полная форма                      Краткая форма**

I  
He/She/It did not didn't finish (leave). You/We/They



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

Употребление

**Past Simple употребляется для выражения:**

В действий, произошедших в прошлом в определенное указанное время, то есть нам известно, **когда** эти действия произошли,  
They **graduated four years ago**. (Когда они закончили университет? Четыре года назад. Мы знаем время.)

В повторяющихся в прошлом действий, которые более не происходят. В этом случае могут использоваться наречия частоты (*always, often, usually* и т.д.)  
He **often played** football with his dad when he was five. (Но теперь он уже не играет в футбол со своим отцом.)

В действиях, следовавших непосредственно одно за другим в прошлом.  
They **cooked the meal first. Then they ate** with their friends.

В Past Simple употребляется также, когда речь идет о людях, которых уже нет в живых.  
Princess Diana **visited** a lot of schools.

Правописание

К глаголам, оканчивающимся на **-e**, прибавляется только **-d**. dance – danced

В глаголах, оканчивающихся на **согласную + y**, опускается **y** и прибавляется **-ied**. try – tried

К глаголам, оканчивающимся на **гласную + y**, прибавляется **-ed**.  
play – played

В глаголах, оканчивающихся на **ударный слог с кратким гласным** между двумя согласными, конечная согласная буква удваивается и прибавляется **-ed**.  
plan – planned, **НО** open – opened

В глаголах, оканчивающихся на **-l**, эта буква **l** удваивается и прибавляется **-ed**. travel – travelled, quarrel – quarrelled

Маркеры

Маркерами Past Simple являются: *yesterday, last night/week/month/year/Monday* и т.д., *two days/weeks/months/years ago, then, when, in 1992* и т.д.

**Past Simple: exercises**

**5.1 Read what Sharon says about a typical working day:**

SHARON

I usually get up at 7 o'clock and have a big breakfast. I walk to work, which takes me about half an hour. I start work at 8.45. I never have lunch. I finish work at 5 o'clock. I'm always tired when I get home. I usually cook a meal in the evening. I don't usually go out. I go to bed at about 11 o'clock. I always sleep well.

Yesterday was a typical working day for Sharon. Write she did or didn't do yesterday.

- 15) She got up at 7 o'clock.
- 16) She --- a big breakfast.
- 17) She ---.
- 18) It --- to get to work.
- 19) --- at 8.45.
- 20) --- lunch.
- 21) --- at 5 o'clock.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- --- tired when --- home.
- --- a meal yesterday evening.
- --- out yesterday evening.
- --- at 11 o'clock.
- --- well last night.

**5.2 Put one of these verbs in each sentence:**

buy catch cost drink fall hurt sell spend teach throw win write

14. Mozart wrote more than 600 pieces of music.
15. 'How did you learn to drive?' 'My father --- me.'
16. We couldn't afford to keep our car, so we --- it.
17. I was very thirsty. I --- the water very quickly.
18. Paul and I played tennis yesterday. He's much better than me, so he --- easily.
19. Don --- down the stairs this morning and --- his leg.
20. Jim --- the ball to Sue, who --- it.
21. Ann --- a lot of money yesterday. She --- a dress which --- 1100.

**5.3 A friend has just come back from holiday. You ask him about it. Write your questions.**

- 8 (where/go?) Where did you go?
- 9 (go alone?) ---
- 10 (food/good?) ---
- 11 (how long/stay there?) ---
- 12 (stay/at a hotel?) ---
- 13 (how/travel?) ---
- 14 (the weather/fine?) ---
- 15 (what/do in the evenings?) ---
- 16 (meet anybody interesting?) ---

**5.4 Complete the sentences, Put the verb into the correct form, positive or negative.**

14. It was warm, so I took off my coat. (take)
15. The film wasn't very good. I didn't enjoy it very much. (enjoy)
16. I knew Sarah was very busy, so I --- her. (disturb)
17. I was very tired, so I --- to bed early. (go)
18. The bed was very uncomfortable. I --- very well. (sleep)
19. Sue wasn't hungry, so she --- anything. (eat)
20. We went to Kate's house but she --- at home. (be)
21. It was a funny situation but nobody --- (laugh)
22. The window was open and a bird --- into the room. (fly)
23. The hotel wasn't very expensive. It --- very much. (cost)
24. I was in a hurry, so I --- time to phone you. (have)

**Unit 6: Past Continuous – прошедшее продолженное время**

1

2

*Образование*

Прошедшее продолженное время образуется с помощью *was/were* (формы past simple глагола to be) и *смыслового глагола*, к которому добавляется *-ing*. В вопросах *was/were* ставятся перед подлежащим. В отрицаниях *not* следует после *was/were*.

w

a

sI

He/She/It

hYou/We/They

a

r

d

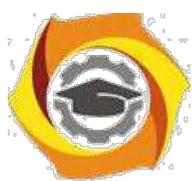
**Утверждение**

was

was

were

studying.



W  
o

**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Вопрос**

Was	I	
Was	he/she/it	studying?
Were	you/we/they	

**Отрицание**

	<b>Полная форма</b>	<b>Краткая форма</b>	
I	was not	wasn't	
He/She/It	was not	wasn't	studying.
You/We/They	were not	weren't	

*Употребление*

**Past Continuous употребляется для выражения:**



временного действия, продолжавшегося в прошлом в момент, о котором мы говорим. Мы не знаем, когда началось и когда закончилось это действие,

At three o'clock yesterday afternoon Mike and his son **were washing** the dog. (Мы не знаем, когда они начали и когда закончили мыть собаку.)



временного действия, продолжавшегося в прошлом (longer action) в момент, когда произошло другое действие (shorter action). Для выражения второго действия мы употребляем past simple, He **was reading** a newspaper **when** his wife **came**. (was reading = longer action; came = shorter action)



двух и более временных действий, одновременно продолжавшихся в прошлом.

The people **were watching while** the cowboy **was riding** the bull.



Past Continuous употребляется также для описания обстановки, на фоне которой происходили события рассказа (повествования).

The sun **was shining** and the birds **were singing**. Tom **was driving** his old truck through the forest.

*Правописание*

**В**

В глаголах, оканчивающихся на **-e**, обычно опускается **-e** и прибавляется **-ing**. dance – dancing, **НО** agree - agreeing

**В**

В глаголах, оканчивающихся на **ударный слог с кратким гласным** между двумя согласными, конечная согласная буква удваивается и прибавляется **-ing**.

refer – referring, run – running, get – getting, **НО** open - opening

**В**

В глаголах, оканчивающихся на **-l**, эта буква **l** удваивается и прибавляется **-ing**. travel - travelling

**В**

В глаголах, оканчивающихся на **-ie**, опускается **-ie** и прибавляется **-y + -ing**. lie – lying, die - dying

*Маркеры*

Маркерами Past Continuous являются: *while, when, as, all day/night/morning* и т.д.



when/while/as + past continuous (longer action)



when + past simple (shorter action)

**Past Continuous: exercises**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**6.1 What were you doing at the following times? Write one sentence as in the examples. The past continuous is not always necessary (see the second example).**

1. (at 8 o'clock yesterday evening)  
I was having dinner with some friends.
2. (at 5 o'clock last Saturday)  
I was on a train on my way to London.
- (at 10.15 yesterday morning)
- (at 4.30 this morning)
- (at 7.45 yesterday evening)
- (half an hour ago)

**6.2 Use your own ideas to complete these sentences. Use the past continuous.**

13. Tom burnt his hand while he was cooking the dinner.
14. The doorbell rang while I ---
15. We saw an accident while we ---
16. Mary fell asleep while she ---
17. The television was on but nobody ---

**6.3 Put the verbs into the correct form, past continuous or past simple.**

- 12) I saw (see) Sue in town yesterday but she --- (look) the other way.
- 13) I --- (meet) Tom and Ann at the airport a few weeks ago. They --- (go) to Berlin and I --- (go) to Madrid. We --- (have) a chat while we --- (wait) for our flights.
- 14) I --- (cycle) home yesterday when suddenly a man --- (step) out into the road in front of me. I -- (go) quite fast but luckily I --- (manage) to stop in time and --- (not/hit) him.

**6.4 Put the verbs into the correct form, past continuous or past simple.**

- 15) Jane was waiting (wait) for me when I arrived (arrive).
- 16) 'What --- (you/do) this time yesterday?' 'I was asleep.'
- 17) '--- (you/go) out last night?' 'No, I was too tired.'
- 18) 'Was Carol at the party last night?' 'Yes, she --- (wear) a really nice dress.'
- 19) How fast --- (you/drive) when the accident --- (happen)?
- 20) John --- (take) a photograph of me while I --- (not/look).
- 21) We were in a very difficult position. We --- (not/know) what to do.
- 22) I haven't seen Alan for ages. When I last --- (see) him, he --- (try) to find a Job in London.
- 23) I --- (walk) along the street when suddenly I --- (hear) footsteps behind me. Somebody --- (follow) me. I was frightened and I --- (start) to run.
- 24) When I was young, I --- (want) to be a bus driver.

**Unit 7: Сопоставление Past Continuous и Past Simple**

*Past Continuous*

Past Continuous употребляется для выражения:

**В** временного действия, продолжавшегося в прошлом в момент, о котором мы говорим,

At nine o'clock yesterday morning, the plane **was flying** to Tahiti from New York. (Мы не знаем, когда начался полет и когда он закончился.)

*Past Simple*

Past Simple употребляется для выражения:

 действия, которое произошло (завершилось) в прошлом в установленное время,

The plane **landed** at the airport at eight o'clock yesterday morning. (Время установлено. Действие завершилось. Самолет приземлился.)

**И** действий, следовавших непосредственно одно за другим в прошлом.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

двух временных действий, одновременно продолжавшихся в прошлом.

He **was listening** carefully while they **were explaining** the plan to him.

First she **read** the advertisement and then she **called** the company.

**Past Simple or Past Continuous: exercises**

**7.1 Put the verbs in brackets into past simple or past continuous. Which is the longer action in each sentence?**

11. As I .....*was doing*.....(do) the washing-up, I .....*broke*..... (break) a glass. “*Doing the washing-up is the longer action.*”
12. We ..... (walk) in the woods when the storm ..... (begin).
13. John ..... (repair) his motor bike when his mother .....(arrive).
14. I ..... (eat) my lunch when the phone ..... (ring).
15. He .....(ride) his bicycle to school when he .....(drop) his bag.
16. We ..... (see) a bad accident as we ..... (drive) to the airport.
17. Tom ..... (watch) the match when the TV ..... (break down).
18. We ..... (talk) when she ..... (come) into the room.

**7.2 Put the verbs into the correct form, past continuous or past simple.**

<p>1. I ....<i>saw</i>.... (was seeing/saw) Sue in town yesterday but she .....(wasn't seeing/ didn't see) me. She ..... (was looking/ looked) the other way.</p>	<p>2. I .....(was meeting/ met) Tom and Ann at the airport a few weeks ago. They ..... (were going/went) to Berlin and I ..... (were going/went) to Madrid. We ..... (were having/ had) a chat while we (were waiting/waited) for our flights.</p>	<p>3. I .....(was cycling/ cycled) home yesterday when suddenly a man ..... (was stepping/stepped) out into the road in front of me. I..... (was going/went) quite fast but luckily I ..... (was managing/ managed) to stop in time and (wasn't hitting/hit) him.</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**7.3 Complete these texts using past continuous or past simple of the verbs in brackets.**

Beethoven .....*wrote*.... (write) nine symphonies, he ....*was writing*.... (write) another symphony when he died.

16. Last Saturday Tom wanted to make two salads. He ..... (make) the first one in five minutes. He ..... (make) the second one when his guests ..... (arrive), and they ..... (help) him to finish it.
17. The artist Gaudi ..... (design) several houses in Barcelona, Spain. Later he ..... (start) work on a church. He ..... (work) on the church when he ..... (die).



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

13. Last month a bank robber ..... (escape) while the police ..... (take) him to prison. Later they ..... (catch) him again, and this time they ..... (lock) him up without any problem.
14. Philip's football team were lucky last Saturday. After twenty minutes they ..... (lose), but in the end they ..... (win) the game by four goals to two.
15. John Lennon ..... (sing) and ..... (play) on many records with the Beatles. After that he ..... (record) several songs without the Beatles. He ..... (prepare) a new record when Mark Chapman ..... (shoot) him.
16. The evening was getting darker; the street lights ..... (come) on. People ..... (hurry) home after work. I ..... (stand) in a queue at the bus stop. Suddenly somebody ..... (grab) my bag.

**Unit 8: Present Perfect – настоящее совершенное время**

*Образование*

Настоящее совершенное время образуется с помощью вспомогательного глагола *have/has* и *причастия прошедшего времени* (past participle). Причастие прошедшего времени правильных глаголов образуется путем добавления к глаголу окончания – *ed*. Например: *play – played*.

Причастие прошедшего времени неправильных глаголов образуется иначе. Например: *see – seen*. (Смотрите список неправильных глаголов.)

Вопросы строятся путем постановки *have/has* перед подлежащим. Например: *Have they read the book?* Отрицания строятся путем постановки *not* между *have/has* и *причастием прошедшего времени*. Например: *He has not/hasn't repaired the TV yet.*

**Утверждение**

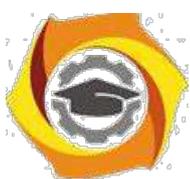
	<b>Полная форма</b>	<b>Краткая форма</b>	
I	have	've	
He/She/It	has	's	finished/left.
You/We/They	have	've	

**Вопрос**

Have	I	
Has	he/she/it	finished/left?
Have	you/we/they	

**Отрицание**

	<b>Полная форма</b>	<b>Краткая форма</b>	
I	have not	haven't	
He/She/It	has not	hasn't	finished/left.
You/We/They	have not	haven't	



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*Употребление*

**Present Perfect употребляется для выражения:**



действий, которые произошли в прошлом в неопределенное время. Конкретное время действия не важно, важен результат,

Kim **has bought** a new mobile phone. (Когда она его купила? Мы это не уточняем, поскольку это не важно. Важно то, что у нее есть новый мобильный телефон.)



действий, которые начались в прошлом и все еще продолжаются в настоящем,

He **has been** a car salesman since 1990. (Он стал продавцом автомобилей в 1990 году и до сих пор им является.)



действий, которые завершились совсем недавно и их результаты все еще ощущаются в настоящем.

They **have done** their shopping. (Мы видим, что они только что сделали покупки, поскольку они выходят из супермаркета с полной тележкой.)



Present Perfect употребляется также со словами *today, this morning/afternoon* и т.д., когда обозначенное ими время в момент речи еще не истекло.

He **has made** ten pots *this morning*. (Сейчас утро. Указанное время не истекло.)

*Маркеры*

Маркерами Present Perfect являются: *for, since, already, just, always, recently, ever, how long, yet, lately, never, so far, today, this morning/afternoon/week/month/year* и т.д.

**В утверждениях**



for: I have known them **for** six years.



since: She has been ill **since** Monday.



already: We have **already** eaten our lunch.



just: I have **just** posted the letter.



always: She has **always** wanted to travel abroad.



recently: He has **recently** published a book.

**В вопросах**



ever: Have you **ever** met anybody famous?



how long: **How long** have you lived here?



yet: Has Paul left **yet**?



lately: Have you seen any good films **lately**?

**В отрицаниях**



for: I haven't talked to him **for** days.



since: They haven't been abroad **since** 1990.



yet: She hasn't answered my letter **yet**.



lately: I haven't seen John **lately**.



never: They have **never** worked abroad.

***Present Perfect: exercises***

**8.1 You are writing a letter to a friend. In the letter you give news about yourself and other people. Use the words given to make sentences. Use the present perfect.**

Dear Chris,

Lots of things have happened since I last wrote to you.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

11. I/buy/a new  
car I've bought a new  
car.
12. my father/start/a new job
13. I/give up/smoking
14. Charles and Sarah/go/to Brazil
15. Suzanne/have/a baby

**8.2 Read the situations and write sentences. Choose one of the following:**

arrive break go up grow improve lose

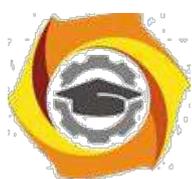
- > Mike is looking for his key. He can't find it. He has lost his key.
- > Margaret can't walk and her leg is in plaster. She ---
- > Maria's English wasn't very good. Now it is much better. ---
- > Tim didn't have a beard last month. Now he has a beard. ---
- > This morning I was expecting a letter. Now I have it. ---
- > Last week the bus fare was 80 pence. Now it is 90. ---

**8.3 Complete Bs sentences. Use the verb in brackets + just/already/yet (as shown).**

1. A: Would you like something to eat?
8. No, thanks. I've just had lunch. (just/have)
2. A: Do you know where Julia is?
11. Yes, I --- her. (just/see)
3. A: What time is David leaving?
- C He --- (already/leave)
4. A: What's in the newspaper today?
- l) I don't know. I --- (not/read/yet)
5. A: Is Ann coming to the cinema with us?
- C No, she --- the film. (already/see)
6. A: Are your friends here yet?
- 19) Yes, they --- (just/arrive)
7. A: What does Tim think about your plan?
- 12) I --- (not/tell/yet)

**8.4 Read the situations and write sentences with just, already or yet.**

1. After lunch you go to see a friend at her house. She says 'Would you like something to eat?' You say: No, thank you. I've just had lunch. (have lunch)
  2. Joe goes out. Five minutes later, the phone rings and the caller says 'Can I speak to Joe?' You say: I'm afraid --- (go out)
  3. You are eating in a restaurant. The waiter thinks you have finished and starts to take your plate away. You say: Wait a minute! --- (not/finish)
  4. You are going to a restaurant this evening. You phone to reserve a table. Later your friend says 'Shall
- 
23. You know that a friend of yours is looking for a job. Perhaps she has been successful. Ask her. You say: ---? (find)
  24. Ann went to the bank, but a few minutes ago she returned. Somebody asks 'Is Ann still at the bank?'



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

You say: No, --- (come back)

**8.5 Put in been or gone.**

в Jim is on holiday. He's gone to Italy.

в Hello! I've just --- to the shops. I've bought lots of things.

в Alice isn't here at the moment. She's --- to the shop to get a newspaper.

в Tom has. --- out. He'll be back in about an hour.

в 'Are you going to the bank?' 'No, I've already --- to the bank.'

**Unit 9: Сопоставление Past Simple и Present Perfect**

*Past Simple*

Мы употребляем Past Simple для выражения действия, которое:

В произошло в прошлом в указанное время,

Ed Prior **won** his first gold medal in 1992. (Когда? В 1992 году. Время указано.)

и началось и закончилось в прошлом.

Annette had a cold for two days. (Она больше не простужена.)

*Present Perfect*

Мы употребляем Present Perfect для выражения действия, которое:

В произошло в прошлом в неопределенное время,

Ed Prior **has won** a lot of medals. (Когда? Мы не знаем. Время не указано.)

В началось в прошлом и продолжается сейчас.

Lucy **has had** a cold for two days. (Она до сих пор простужена.)

***Past Simple or Present Perfect: exercises***

**9.1 What has happened in these situations?**

- Jack had a beard. Now he hasn't got a beard. He has shaved off his beard.
- Linda was here five minutes ago. Now she's in bed. She ---
- The temperature was 25 degrees. Now it is only 17. The temperature ---
- The light was off. Now it is on. Somebody ---
- The tree was only three metres high. Now it is four. The tree ---
- The plane was on the runway a few minutes ago. Now it is in the air. The plane ---

**9.2 Put the verbs in brackets in the correct form, present perfect or past simple.**

В 'Where's your key?' 'I don't know. I've lost it.' (lose)

В I was very tired, so I lay down on the bed and went to sleep. (be)

В Mary --- to Australia for a while but she's back again now. (go)

В 'Where's Ken?' 'He --- out. He'll be back in about an hour.' (go)

В I did German at school but I --- most of it. (forget)

В I meant to phone Diane last night but I --- (forget)

В I --- a headache earlier but I feel fine now. (have)

В Look! There's an ambulance over there. There --- an accident. (be)

В They're still building the new road. They --- it. (not/finish)



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

13. 'Is Helen still here?' 'No, she --- out.' (just/go)
14. The police --- three people but later they let them go. (arrest)
15. Ann --- me her address but I'm afraid I --- it. (give, lose)
16. Where's my bike? It --- outside the house. It --- (be, disappear)
17. What do you think of my English? Do you think I ---? (improve)

**9.3 Are the underlined parts of these sentences right or wrong? Correct the ones that are wrong.**

10. Do you know about Sue? She's given up her job. RIGHT
11. The Chinese have invented printing. WRONG: The Chinese invented
12. How many plays has Shakespeare written? ---
13. Have you read any of Shakespeare's plays? ---
14. Aristotle has been a Greek philosopher. ---
15. Ow! I've cut my finger. It's bleeding. ---
16. My grandparents have got married in London. ---
17. Where have you been born? ---
18. Mary isn't at home. She's gone shopping. ---
19. Albert Einstein has been the scientist who has developed the theory of relativity. ---

**9.4 (Section C) Put the verb into the most suitable form, present perfect or past simple.**

12. A: Look! Somebody has split (spill) coffee on the carpet.

B: Well, it wasn't (not/be) me. I didn't do (not/do) it.

13. A: Ben --- (break) his leg.

и Really? How --- (that/happen)?

• He --- (fall) off a ladder.

3. A: Your hair looks nice. --- (you/have) a haircut?

• Yes.

• Who --- (cut) it? --- (you/go) to the hairdresser?

B: No, a friend of mine --- (do) it for me.

**9.5 Are the underlined parts of these sentences right or wrong? Correct the ones that are wrong.**

и I've lost my key. I can't find it anywhere. RIGHT

и Have you seen the news on television last night? WRONG: Did you see

и I've bought a new car. Do you want to see it? ---

и I've bought a new car last week. ---

и Where have you been yesterday evening? ---

и Jenny has left school in 1991. ---

и I'm looking for Mike. Have you seen him? ---



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- I'm very hungry. \_ I haven't eaten\_ anything today. ---
- Diane \_hasn't been\_ at work yesterday. ---
- When \_has this book been\_ published? ---

**9.6 Make sentences from the words in brackets. Use the present perfect or past simple.**

- (it/not/rain/this week) It hasn't rained this week.
- (the weather/be/cold/recently) The weather ---
- (it cold/last week) It ---
- (I not/read/a newspaper yesterday) I ---
- (I not/read/a newspaper today)
- (Ann/earn/a lot of money/this year)
- (she not/earn/so much/last year)
- (you have/a holiday recently?)

**9.7 Put the verb into the correct form, present perfect or past simple.**

- C I don't know where Amy is. Have you seen (you/see) her?
- C When I --- (get) home last night, I --- (be) very tired and I --- (go) straight to bed.
- C Your car looks very clean --- (you/wash) it?
- C George --- (not/be) very well last week.
- C Mr Clark --- (work) in a bank for 15 years. Then he gave it up.
- C Molly lives in Dublin. She --- (live) there all her life.
- 7 --- (you/go) to the cinema last night?' 'Yes, but it --- (be) a mistake. The film (be) awful.
- C My grandfather --- (die) 30 years ago. I --- (never/meet) him.
- C I don't know Carol's husband. I --- (never/meet/him).
- C A: Is your father at home?

- When exactly --- (he/go) out? B: About ten minutes ago.  
A: Where do you live?
- In Boston.
- How long --- (you/live) there?
- Five years.
- Where --- (you/live) before that?
- In Chicago.
- And how long --- (you/live) in Chicago?
- Two years.

**9.8 Write sentences about yourself using the ideas in brackets.**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- (something you haven't done today) I haven't eaten any fruit today.
- (something you haven't done today)
- (something you didn't do yesterday)
- (something you did yesterday evening)
- (something you haven't done recently)
- (something you've done a lot recently)

**Unit 10: Future Simple – будущее простое время**

*Образование*

Будущее простое время образуется с помощью *will* и *инфинитива без частицы to*. *Will* используется для всех лиц. Вопросы образуются путем постановки *will* перед подлежащим. В отрицаниях после *will* ставится *not*.

**Утверждение**

	<b>Полная форма</b>	<b>Краткая форма</b>	
I	will	'll	
He/She/It	will	'll	phone.
You/We/They	will	'll	

**Вопрос**

Will	I	
Will	he/she/it	phone?
Will	you/we/they	

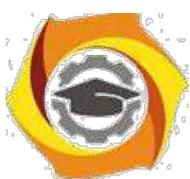
**Отрицание**

	<b>Полная форма</b>	<b>Краткая форма</b>	
I	will not	won't	
He/She/It	will not	won't	phone.
You/We/They	will not	won't	

*Употребление*

**Future simple употребляется:**

- для обозначения будущих действий, которые, возможно, произойдут, а возможно, и нет, **We'll visit Disney World one day.**
- для предсказаний будущих событий (predictions), **Life will be better fifty years from now.**
- для выражения угроз или предупреждений (threats/warnings), **Stop or I'll shoot.**
- для выражения обещаний (promises) и решений, принятых в момент речи (on-the-spot decisions), **I'll help you with your homework.**



**В** с глаголами *hope, think, believe, expect* и т.п.,  
с выражениями *I'm sure, I'm afraid* и т.п.,  
II. также с наречиями *probably, perhaps* и т.п.  
I **think** he **will** support me. He  
**will probably** go to work.

### Маркеры

Маркерами Future Simple являются: *tomorrow, the day after tomorrow, next week/month/year, tonight, soon, in a week/month/year* и т.д.

**Примечание:** Future Simple не употребляется после слов *while, before, until, as soon as, after, if* и *when* в придаточных предложениях условия и времени. В таких случаях используется Present Simple.

Например: I'll make a phone call **while** I **wait** for you. Please phone me **when** you **finish** work.

17. дополнительных придаточных предложениях после «when» и «if» возможно употребление Future Simple.

Например: I don't know **when/if** Helen **will** be back.

### Future Simple: exercises

#### 10.1. What will life be like in 50 years? Make sentences using *will* or *won't*.

- |                                      |   |
|--------------------------------------|---|
| 1. People/live longer.               | + |
| People <i>will live</i> longer.      |   |
| 2. Robots/do most of the work.       | + |
| 3. People/use electric cars.         | - |
| 4. Pollution/disappear.              | + |
| 5. People/die of serious diseases.   | - |
| 6. Children/stop going to school.    | - |
| 7. People/go on holiday to the moon. | + |

#### 10.2 Put in *will* ('ll) or *won't*.

- Can you wait for me? I *won't* be very long.
- There's no need to take an umbrella with you. It --- rain.
- If you don't eat anything now, you --- be hungry later.
- I'm sorry about what happened yesterday. It --- happen again.
- I've got some incredible news! You --- never believe what's happened.
- Don't ask Margaret for advice. She --- know what to do.

#### 10.3 Complete the sentences with *I'll* + a suitable verb (get, go, do, switch, have, show, send, stay, give).

И "I'm too tired to walk home. I think .....*I'll get*..... a taxi."

И "It's a bit cold in this room." – "Is it? ..... on the heating then."



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- o “We haven’t got any milk.” – “Oh, haven’t we? ..... and get some.”
- o “Do you want me to do the washing-up?” – “No, it’s all right. .... it.”
- o “I don’t know how to use this computer.” – “OK, ..... you.”
- o “Would you like tea or coffee?” – “..... coffee, please.”
- o “Goodbye! Have a nice holiday.” – “Thanks. .... you a postcard.”
- o “Thank you for lending me your camera. .... it back to you on Monday, OK?”
- o “Are you coming with us?” – “No, I think ..... here.”

**10.4. Use the verbs in the box with *will* or *won’t* to complete these dialogues.**

<i>have</i>	<i>take</i>	<i>phone</i>	<i>finish</i>	<i>be</i>	<i>be</i>	<i>win</i>	<i>make</i>
-------------	-------------	--------------	---------------	-----------	-----------	------------	-------------

B Are you coming to the cinema on Sunday?  
B I’m not sure. I *will phone* you on Saturday.

B A: Don’t change your clothes now. We ..... late.  
No, we *won’t*. We ..... a taxi.

B A: George is going to have a party at the weekend.  
Why?  
It’s his birthday. He ..... thirty on Saturday.

B A: She ..... the tennis match tomorrow.  
Why not?

B She ..... mistakes. She always makes mistakes in important matches.

C A: ..... Steve ..... the work tonight?  
No, he *won’t finish*. He ..... time.

**10.5 Complete the sentences with *will* ('ll) + one of these verbs:**

be be come get like look meet pass

- B Don't worry about your exam. I'm sure you I'll pass.
- B Why don't you try on this jacket? It --- nice on you.
- B You must meet George sometime. I think you --- him.
- B It's raining. Don't go out. You --- wet.
- B They've invited me to their house. They --- offended if I don't go.
- B Goodbye. I expect we --- again before long.
- B I've invited Sue to the party but I don't think she ---.
- B I wonder where I --- 20 years from now.

**10.6 Write questions using *do you think ... will ...?* + one of these verbs:**

be back cost finish get married happen like rain



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- B I've bought Mary a present. Do you think she'll like it?  
B The weather doesn't look very good. Do you ---  
B The meeting is still going on. When do you ---  
B My car needs to be repaired. How much ---  
B Sally and David are in love. Do ---  
B 'I'm going out now.' 'OK. What time ---'  
B The future situation is uncertain. What ---

***Will/Shall: exercises***

**Мы употребляем:**

- B **Will you...?**, когда просим сделать что-нибудь для нас (**request**),  
Например: **Will you** post these letters for me, please? (=Can you post these letters for me, please?)  
и **Shall I...?**, когда предлагаем сделать что-нибудь для кого-то (**offer**),  
Например: **Shall I** help you clean your room? (=Do you want me to help you clean your room?)  
11 **Shall we...?**, когда предлагаем кому-то сделать что-либо вместе (**suggestion**).  
Например: **Shall we** go to the theatre tonight? (=Why don't we go to the theatre tonight?)

**10.4 Ask questions using the prompts, as in the example.**

- 28 The garden is very untidy. (I/cut/the grass) *Shall I cut the grass?*  
29 It's a lovely evening. (we/go for/a walk)  
30 I need a hot drink. (I/make/some tea)  
31 It's very quiet in here. (I/turn on/ the radio)  
32 The Smiths are back. (we/visit/them)  
33 I've cut my finger. (I/get/a plaster)

**10.5. Put *Shall I* or *Shall we* in the gaps in the dialogues.**

- B A: I'm hungry. Are you going to the shops?  
Yes. ....*Shall I*.... get you something to eat?
- B A: We need a holiday.  
What a good idea! ..... go to Florida?
- B A: I'm going to get some tickets for the concert next week. .... buy you one?  
Yes, please. I'd love to come.
- B A: ..... go to a restaurant tonight?  
OK, but I don't have any money. Will you pay for me?
- B A: I want to go to Italian classes, but I've never learnt a foreign language before.  
..... come with you?
- A: That's very kind of you.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

13. A: Where is your meeting?  
At John's office on Baker Street.  
..... walk or take a taxi?

\quote A: You look thirsty. .... get you a drink?  
Yes, please. Can I have an orange juice or some water?

\quote A: It's a beautiful day! ..... have a picnic?  
Wonderful idea! Who shall we invite?

**Unit 11: Be Going To**

*Образование*

**Утверждение**

	<b>Полная форма</b>	<b>Краткая форма</b>	
I	am	'm	
He/She/It	is	's	going to leave.
You/We/They	are	're	

**Вопрос**

Am	I	
Is	he/she/it	going to leave?
Are	you/we/they	

**Отрицание**

	<b>Полная форма</b>	<b>Краткая форма</b>	
I	am not	'm not	
He/She/It	is not	isn't	going to leave.
You/We/They	are not	aren't	

*Употребление*

**Be going to употребляется для:**

**В** выражения заранее принятых планов и намерений на будущее,  
Bob **is going to drive** to Manchester tomorrow morning.

**В** предсказаний, когда уже есть доказательства того, что они сбудутся в близком будущем.  
Look at that tree. It **is going to fall down**.

***Be Going To: exercises***

**11.1 Answer the questions. You are going to do all these things but you haven't done them yet.**

**Use going to and the word(s) in brackets.**

17. Have you cleaned the car? (tomorrow) Not yet. I'm going to clean it tomorrow.

18. Have you phoned Sally? (later) Not yet. ---



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Have you done the shopping? (this afternoon) Not yet. ---
- Have you read the paper? (after dinner) Not ---
- Have you had dinner? (just) ---

**11.2 Write a question with going to for each situation.**

B Your friend has won some money. You ask:

(what/do with it?) What are you going to do with it?

B Your friend is going to a party tonight. You

ask: (what/wear?)

B Your friend has just bought a new table. You

ask: (where/put it?)

B Your friend has decided to have a party. You

ask: (who/invite?)

**11.3 Read the situations and complete the dialogues. Use going to.**

1. You have decided to write some letters this evening.

FRIEND: Are you going out this evening? You: No, I'm going to write some letters.

- You are a smoker but you have decided to give it up

soon. FRIEND: Smoking is very bad for you.

YOU: I know. ---

- You have been offered a job but you have decided not to take it. FRIEND: I hear you've been offered a job.

YOU: That's right, but ---

- You are in a restaurant. The food is awful and you've decided to complain. FRIEND: This food is awful, isn't it?

YOU: Yes, it's disgusting. ---

**11.4 What is going to happen in these situations? Use the words in brackets.**

12. There are a lot of black clouds in the sky. (rain) It's going to rain.

13. It is 8.30. Jack is leaving his house. He has to be at work at 8.45 but the journey takes 30 minutes. (late) He ---

14. There is a hole in the bottom of the boat. A lot of water is coming in through the hole. (sink) The boat ---

15. Emma is driving. There is very little petrol left in the tank. The nearest petrol station is a long way away. (run out) She ---

**11.5 Complete the sentences with was/were going to + one of these verbs:**

give up have phone play travel



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

13. We were going to travel by train but then we decided to go by car instead.
14. We --- tennis yesterday but it rained all day.
15. I --- Jim, but I decided to write him a letter instead.
16. When I last saw Tim, he --- his job but in the end he decided not to.
17. We --- a party last week but some of our friends couldn't come, so we cancelled it.

**Unit 12: The Passive – страдательный залог**

*Образование*

Страдательный залог образуется с помощью глагола *to be* и причастия прошедшего времени (past participle) смыслового глагола.

**to be + past participle (pp)**

■ Временные формы страдательного залога:

<i>Present Simple:</i> The office <b>is cleaned</b> twice a week.	<b>am/is/are + pp</b>
<i>Present Continuous:</i> The office <b>is being cleaned</b> now.	<b>am/is/are being + pp</b>
<i>Past Simple:</i> The office <b>was cleaned</b> last week.	<b>was/were + pp</b>
<i>Past Continuous:</i> The office <b>was being cleaned</b> when the boss arrived.	<b>was/were being + pp</b>
<i>Present Perfect:</i> The office <b>has not been cleaned</b> yet.	<b>have/has been + pp</b>
<i>Past Perfect:</i> The office <b>had been cleaned</b> by two o'clock.	<b>had been + pp</b>
<i>Future Simple:</i> The office <b>will be cleaned</b> tomorrow.	<b>will be + pp</b>

- В вопросах *to be* ставится перед подлежащим. Например: **Is your car being serviced?**
- Отрицания образуются с помощью *not*. Например: The furniture **has not been delivered** yet.
- Инфинитив в форме страдательного залога:

**to be + past participle**

Например: He wants **to be told** the truth.

- Модальные глаголы в страдательном залоге:

**modal verb + be + past participle**

Например: The roof of the house **must be repaired**.

*Употребление*



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Мы употребляем страдательный залог, когда хотим показать, что само действие важнее, чем лицо, которое его совершает.

The missing climber **was found** yesterday. (Не важно, кто именно нашел потерявшегося альпиниста, важно, что его нашли.)

- Лицо, совершающее действие («агент» действия), вводится предлогом *by* и упоминается лишь тогда, когда это важно или необходимо.

Australia **was discovered by Captain Cook**. (*by Captain Cook* – личность «агента» действия важна)

- «Агент» действия не упоминается, когда: а) он неизвестен,

Jim's car **was stolen** last night. (Мы не знаем, кто украл его машину.) б) это не важно,

Champagne **is made** in France. («Агент» действия не важен.) в) он очевиден.

The bank robbers **are being chased**. (Ясно, что их преследует полиция.)

### *The Passive: exercises*

#### **12.1 Complete the sentences using one of these verbs in the correct form:**

cause damage hold include invite make overtake show translate write

- Many accidents are caused by dangerous driving.
- Cheese --- from milk.
- The roof of the building --- in a storm a few days ago.
- There's no need to leave a tip. Service --- in the bill.
- You --- to the wedding. Why didn't you go?
- A cinema is a place where films ---
- In the United States, elections for President --- every four years.
- Originally the book --- in Spanish and a few years ago it
- We were driving along quite fast but we --- by lots of other cars.

#### **12.2 Write questions using the passive. Some are present and some are past.**

и Ask about the telephone. (when/invent?)

When was the telephone invented?

и Ask about glass. (how/make?) How ---

и Ask about Australia. (when/discover?)

и Ask about silver. (what/use for?)

и Ask about television. (when/invent?)

#### **12.3 Put the verb into the correct form, present simple or past simple, active or passive.**

\endash It's a big factory. Five hundred people are employed (employ) there.

\endash Water --- (cover) most of the Earth's surface.

\endash Most of the Earth's surface --- (cover) by water.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- K The park gates --- (lock) at 6.30 p.m. every evening.
- K The letter --- (post) a week ago and it --- (arrive) yesterday.
- K The boat --- (sink) quickly but fortunately everybody --- (rescue).
- K Ron's parents --- (die) when he was very young. He and his sister --- (bring) up by their grandparents.
- K I was born in London but I --- (grow) up in the north of England.
- K While I was on holiday, my camera --- (steal) from my hotel room.
- K While I was on holiday, my camera --- (disappear) from my hotel room.
- K Why --- (Sue/resign) from her job? Didn't she enjoy it?
- K Why --- (Bill/sack) from his job? What did he do wrong?
- K The company is not independent. It --- (own) by a much larger company.
- K I saw an accident last night. Somebody --- (call) an ambulance but nobody --- (injure) so the ambulance --- (not/need).
- K Where --- (these photographs/take)? In London? --- (you/take) them?

**12.4 Rewrite these sentences. Instead of using 'somebody/they/people' etc. write a passive sentence.**

- B Somebody cleans the room every day. \_The room is cleaned every day.\_
- B They cancelled all flights because of fog. All ---
- B People don't use this road very often. ---
- B Somebody accused me of stealing money. I ---
- B How do people learn languages? How ---
- B People advised us not to go out alone. ---

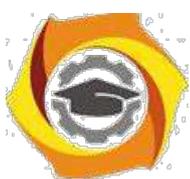
**12.5 Rewrite these sentences. Instead of using 'somebody' or 'they', write a passive sentence.**

- \endash Somebody has cleaned the room. \_The room has been cleaned.\_
- \endash They have postponed the concert. The ---
- \endash Somebody is using the computer at the moment. The computer ---
- \endash I didn't realise that somebody was recording our conversation. I didn't realise that ---
- \endash When we got to the stadium we found that they had cancelled the game. When we got to the stadium, we found that ---
- \endash They are building a new ring road round the city. ---
- \endash They have built a new hospital near the airport. ---

**12.6 When were they born? Choose five of these people and write a sentence for each. (Two of them were born in the same year.)**

Beethoven Galileo Elvis Presley 1452 1869 1929

Agatha Christie Mahatma Gandhi Leonardo da Vinci 1564 1891 1935



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Walt Disney Martin Luther King William Shakespeare 1770 1901

9. \_Walt Disney was born in 1901.\_

10. ---

11. ---

12. ---

13. ---

14. ---

15. And you? I ---

**12.7. Put the verb into the correct form, *present simple* or *past simple*, *active* or *passive*.**

9. It's a big factory. Five hundred people .... *are employed*.... (employ) there.

10. Water ..... (cover) most of the Earth's surface.

11. Most of the Earth's surface ..... (cover) by water.

12. The park gates ..... (lock) at 6.30 p.m. every evening.

13. The letter ..... (post) a week ago and it ..... (arrive) yesterday.

14. The boat ..... (sink) quickly but fortunately everybody ..... (rescue).

15. Ron's parents ..... (die) when he was very young. He and his sister ..... (bring) up by their grandparents.

16. I was born in London but I ..... (grow) up in the north of England.

17. While I was on holiday, my camera ..... (steal) from my hotel room.

18. While I was on holiday, my camera ..... (disappear) from my hotel room.

19. Why ..... (Sue/resign) from her job? Didn't she enjoy it?

20. Why ..... (Bill/sack) from his job? What did he do wrong?

21. The company is not independent. It ..... (own) by a much larger company.

22. I saw an accident last night. Somebody ..... (call) an ambulance but nobody ..... (injure), so the ambulance ..... (not/need).

**Tenses**

**The Active Voice**

	Present V/V-s	Past V-ed/V(2)	Future will/shall+V
Simple	I <b>write</b> letters to my mother every day.	I <b>wrote</b> two letters yesterday.	I <b>will write</b> it tomorrow.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

<b>Continuous</b>	<b>am/is/are+V-ing</b> I <b>am writing</b> a letter (now).	<b>was/were+V-ing</b> I <b>was writing</b> a letter when you called.	<b>will/shall be+V-ing</b> I <b>will be writing</b> letters at 5 o'clock.
<b>Perfect</b>	<b>have/has+V(3)</b> I <b>have</b> already <b>written</b> 5 letters.	<b>had+V(3)</b> I <b>had written</b> 5 letters by the time you came.	<b>will/shall have+V(3)</b> I <b>will have written</b> the letters by noon.

*The Passive Voice*

	Present	Past	Future
<b>Simple</b>	<b>am/is/are+V(3)</b> The letters <b>are written</b> every week.	<b>was/were+V(3)</b> The letters <b>were written</b> last week.	<b>will be+V(3)</b> The letter <b>will be written</b> next week.
<b>Continuous</b>	<b>am/is/are being+V(3)</b> The letters <b>are being written</b> at this moment.	<b>was/were being+V(3)</b> The letters <b>were being written</b> at 5 o'clock yesterday.	_____
<b>Perfect</b>	<b>have/has been+V(3)</b> The letters <b>have</b> already <b>been written</b> .	<b>had been+V(3)</b> The letters <b>had been written</b> by 5 o'clock.	<b>will have been+V(3)</b> The letters <b>will have been written</b> by noon.

Методические рекомендации для студентов по организации и выполнению самостоятельной работы по модулю " Факультативный модуль Эмоциональный интеллект ".

Самостоятельная работа является неотъемлемой составляющей образовательного процесса. Самостоятельная работа магистрантов включает изучение теоретического курса и подготовку к практическим и лабораторным занятиям, выполнение домашнего задания, подготовка к зачетам. Настоящие Методические рекомендации для студентов по организации и выполнению самостоятельной работы по модулю " Факультативный модуль Эмоциональный интеллект " относятся к виду учебной работы «Изучение теоретического курса и подготовка к зачету». Самостоятельная работа магистрантов также включает все виды текущей аттестации.

*Тематика самостоятельной работы*

№	Наименование работы
---	---------------------



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

1	Изучения теоретического материала и подготовки к практическим занятиям: 1. Профессиональная деятельность как предмет научного изучения психологии 2. Психологические основы профессиональной деятельности 3. Влияние профессиональной деятельности на индивидуально-психологические особенности личности 4. Влияние особенностей личности на эффективность профессиональной деятельности 5. Социально-психологические проблемы профессиональной деятельности
2	Выполнение домашней работы
3	Изучения теоретического материала и подготовки к практическим занятиям: Деловые конфликты. Деловое общение и профессиональная деятельность. Психология профессионального отбора и подбора персонала. Технологии социально-профессионального саморазвития.
4	Подготовка к зачету

***Самостоятельная работа № 1***

- Изучения теоретического материала и подготовки к практическим занятиям:
18. Профессиональная деятельность как предмет научного изучения психологии
  19. Психологические основы профессиональной деятельности
  20. Влияние профессиональной деятельности на индивидуально-психологические особенности личности
  21. Влияние особенностей личности на эффективность профессиональной деятельности
  22. Социально-психологические проблемы профессиональной деятельности

*Тема. Профессиональная деятельность как предмет научного изучения психологии* Контрольные вопросы

11. Профессиональная и трудовая деятельности как факторы развития личности. Основные психологические проблемы изучения профессиональной деятельности. Особенности изучения профессиональной деятельности в рамках различных научных направлений психологии.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Отрасли психологии, изучающие профессиональную деятельность: акмеология, психология профессий, профессиоведение, психология профессионального образования и др.

Инженерная психология как отрасль фундаментального и прикладного научного знания. Узловые проблемы инженерной психологии.

История изучения психологических особенностей профессиональной деятельности  
12. отечественной и зарубежной психологии. Современное состояние психологических исследований профессиональной деятельности.

Методологические основы психологического изучения профессиональной деятельности. Методы исследования психологических закономерностей взаимодействия профессиональной деятельности и работника

***Задания для самоподготовки.***

Составить схему всех известных вам классификаций функций общения. Указать взаимосвязь между ними.

Подобрать литературные источники по изучению одной из психологических проблем в профессиональной деятельности.

Составить схему методов исследования в психологии и раскрыть сущность методов в форме глоссария.

Подобрать примеры из жизни успешных менеджеров в горной промышленности и металлургии.

*Тема. Психологические основы профессиональной деятельности* Контрольные вопросы

15. Проблема деятельности в психологии. Труд, профессия, специальность, квалификация. Классификация профессий. Психологические основы профессиоведения. Содержание и структура профессиональной деятельности.

16. Модели профессионального развития личности. Стадии профессионального становления личности. Карьера как стратегия профессиональной жизни. Критерии поиска работы. Классификации и виды карьеры. Достоинства и недостатки. Место карьеры в системе жизненных ценностей работника.

17. Понятие успеха в профессиональной деятельности. Динамика профессионального успеха. Психотехника профессионального успеха. Прогнозирование и профессиональный успех. Эффективный темп профессионального успеха.

18. Профессиональные кризисы. Факторы, детерминирующие кризисы профессионального развития. Типология профессиональных кризисов. Стратегии преодоления профессиональных кризисов. Методики изучения профессиональных кризисов.

19. Профессионализм, компетентность, квалификация. Стороны и критерии профессионализма. Мастерство.

***Задания для самоподготовки.***

Сравнительный анализ периодизаций профессионального становления личности. Прочитать притчи «О Мастере и карандаше», «О Мастере и ученике», «О Мастере чайной церемонии» и другие и составить перечень личностных характеристик Мастера.

*Тема. Влияние профессиональной деятельности на индивидуально-психологические особенности личности*

Контрольные вопросы

21. Эффективность выполнения различных видов профессиональной деятельности и типологические особенности личности.

22. Профессиональная адаптация. Виды адаптации в трудовом коллективе. Этапы адаптации. Правила адаптации.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

35. Негативное влияние профессиональной деятельности на социальное, личностное и профессиональное развитие личности. Профессиональные деформации. Проявление профессиональных деформаций у представителей различных профессий. Виды профессиональных деформаций. Причины и механизмы появления профессиональной деформации личности.

36. Понятие о профессиональном стрессе. Динамика профессионального стресса. Стрессовые сценарии. Саморегуляция работника в условиях профессионального стресса. Коммуникативный стресс в условиях профессиональной деятельности. Профессиональный стресс достижения. Профессиональный стресс конкуренции. Профессиональный стресс успеха. Деньги и профессиональный стресс. Стресс подчинения. Технологии оптимизации стрессоустойчивости.

37. Психологические особенности профессий типа «человек-техника», «человек-чело-век». Психологические особенности инженеров

***Задания для самоподготовки.***

Познакомиться с законодательными и нормативными документами по формированию не-зависимой системы оценки и сертификации квалификаций, составить их перечень.

Проанализировать образовательные компетенции и соотнести их с требованиями профессии.

*Тема. Влияние особенностей личности на эффективность профессиональной деятельности*

Контрольные вопросы

33. Профессионально важные качества личности и способности. Профессиональная компетентность. Ключевые квалификации и компетенции. Профессиональная культура.

34. Индивидуальный профессиональный стиль. Стили профессиональной деятельности и личностные особенности.

35. Прогнозирование эффективности профессиональной деятельности по типологическим особенностям личности. Изучение ошибочных действий человека в профессиональной деятельности.

36. Гендерные особенности в профессиональной деятельности. Пол и склонность к профессиям. Представленность мужчин и женщин в различных профессиях. Пол и профессиональная карьера. Различия в удовлетворенности трудом мужчин и женщин. Пол и эффективность групповой деятельности Пол и организационное поведение.

***Задания для самоподготовки.***

Написать эссе «Корпоративные ценности УГМК как средство развития профессионала»

Подберите из литературы и видео материалов примеры высокого служения человека своему профессиональному делу

Подготовить сообщение на тему «Влияние ценностных установок личности на психическое здоровье»

Подберите из литературы и киноматериалов примеры влияния ценностей на эффективность профессиональной деятельности

Составить сообщение и электронную презентацию на тему «Психологическая безопасность в профессиональной деятельности».

Подобрать примеры ошибочных действий работников на производстве и проанализировать их причины и следствия.

*Тема. Социально-психологические проблемы профессиональной деятельности* Контрольные вопросы:

12. Совместная профессиональная деятельность как объект социально-психологического исследования. Мотивация и эффективность профессиональной деятельности. Стимулирование профессиональной деятельности. Мотивация трудовой деятельности. Приемы мотивирования.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Начало профессиональной деятельности. Уровень мотивации работника и профессиональный успех.

Психологические особенности успешных руководителей. Стили руководства. Эффективность различных стилей руководства. Влияние стиля руководства на эффективность профессиональной деятельности трудового коллектива. Приемы взаимодействия руководителя и членов трудового коллектива.

Роль организационных ценностей, традиций и ритуалов в регуляции поведения членов трудового коллектива. Влияние профессиональных ролей на психологические особенности личности. Феномены межличностных отношений в трудовом коллективе. Исследование межличностных отношений в трудовом коллективе.

Психологическая совместимость как фактор эффективности взаимодействия в профессиональной деятельности. Виды психологической совместимости. Критерии оценки совместимости. Различные подходы к определению социально-психологического климата. Факторы, определяющие социально-психологический климат коллектива. Управление социально-психологическим климатом.

Лидерство и руководство: общие черты и отличия этих понятий. Типы лидерства. Ситуации, способствующие выдвижению лидера. Требования к личности лидера. Руководитель и лидер – необходимость взаимодействия между ними.

### ***Задание для самоподготовки***

Составить презентацию на тему «Стимулирование и мотивация трудовой деятельности»

Проанализируйте мотивацию работников в притче. «На стойке работали три человека. Занимались все одним и тем же, но когда их спросили, что они делают, то ответы оказались разными. Один сказал: «Я кладу кирпичи». Другой: «Я зарабатываю себе и своей семье на жизнь». 25. третий ответил: «Я строю храм, который простоит века».

Составить схему всех известных вам классификаций функций общения. Указать взаимосвязь между ними.

Подобрать примеры проявления различных феноменов межличностных отношений в трудовом коллективе и на их основе разработать ролевые ситуации для обыгрывания их на тренинге.

Составьте структурно-логическую схему по теме «Лидерство и руководство» Подберите упражнения для развития лидерских качеств.

## ***Самостоятельная работа № 2***

*Тема:* Выполнение домашней работы

*Цель:*

- отработка навыка решения практических задач.
- подготовка к выполнению кейсов.

*Перечень примерных тем домашних работ.*

33. Сопряженность профессиональных способностей и успешности профессиональной деятельности.

34. Профессионально значимые свойства личности работника и динамика их развития.

35. Профессионально важные и значимые качества личности инженера в горно-металлургической компании.

36. Особенности формирования индивидуального стиля профессиональной деятельности.

37. Техническая эргатическая система как объект психологического исследования.

38. Влияние функциональных состояний человека на эффективность профессиональной деятельности.

39. Динамика работоспособности человека и производственное утомление в условиях горно-металлургической промышленности.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

37. Пути и способы профилактики производственного утомления.
38. Психологическая готовность к профессиональной деятельности: проблема и пути решения.
39. Оптимизация функциональных состояний работников в производственных условиях.
40. Отечественные и зарубежные теории профессионального развития.
41. Характеристика профессиональной пригодности инженера и ее диагностика.
42. Значение процессуальных теорий профессиональной мотивации в практике управления персоналом.
43. Программы повышения мотивации профессиональной деятельности на промышленном предприятии.
44. Удовлетворенность профессиональной деятельностью и ее диагностика у работников.
45. Психологическая характеристика надежности профессиональной деятельности.
46. Психологическая характеристика причин возникновения аварийных ситуаций на производстве.
47. Психологическое обеспечение безопасности труда на производстве.
48. Пути и способы снижения конфликтогенных факторов на предприятии.
49. Профессиональное саморазвитие личности как фактор успешной карьеры.
50. Гендерные особенности построения профессиональной карьеры.
51. Копинг-стратегии как фактор успешной профессиональной деятельности.
52. Выученная беспомощность и ее преодоление в профессиональной деятельности.
53. Профессиональная идентичность личности: структура и условия становления
54. Развитие внутренней позиции профессионала в ходе профессионального обучения.

***Самостоятельная работа № 3***

Изучения теоретического материала и подготовки к практическим занятиям:

Деловые конфликты.

Деловое общение и профессиональная деятельность.

Психология профессионального отбора и подбора персонала.

Технологии социально-профессионального саморазвития.

*Тема. Деловые конфликты.*

Контрольные вопросы

20. Функциональное значение деловых конфликтов. Причины деловых конфликтов. Типы деловых конфликтов и коммуникативные тактики в конфликтных ситуациях. Модель делового конфликта. Последствия деловых конфликтов. Стратегии поведения в конфликтных ситуациях.

21. Моббинг как технология психологического притеснения. Моббинг по вертикали и горизонтали. Преодоление негативных влияний моббинга. Улаживание конфликта при помощи административных способов воздействия.

22. Психологические способы регуляции деловых конфликтов. Переговоры в конфликтной ситуации. Конфликтный человек в работающей команде. Коммуникативная рефлексия как способ регулирования деловых конфликтов. Правила поведения и общения в конфликтной ситуации.

***Задания для самоподготовки.***

Провести мини исследование по определению стратегий поведения в конфликте в трудовом коллективе.

Подобрать примеры конфликтных ситуаций и на основе их анализа разработать пути и способы выхода из них.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

*Тема. Деловое общение и профессиональная деятельность*

Контрольные вопросы

\* Понятие коммуникации в организации, и ее основные характеристики. Основные элементы коммуникационной цепи. Виды коммуникаций по направлению и путям передачи информации. Особенности информационных потоков в организации, их регулирование. Информированность работников как фактор эффективности управления. Слухи как результат недостатка информации.

\* Коммуникативная компетентность как условие успешной профессиональной деятельности и межличностного взаимодействия. Составляющие коммуникативной компетентности. Коммуникативные способности. Коммуникативные умения и способы их формирования. Коммуникативные навыки в практической деятельности руководителя.

\* Психологический контакт с собеседником. Позиции в общении. Этапы общения. Дело-вое общение, его виды и формы. Деловой этикет.

\* Формы общения на производстве: деловое совещание, деловая беседа, деловые переговоры, публичное выступление.

***Задание для самоподготовки***

Составить структурно-логическую схему по теме занятия «Общение как межличностное взаимодействие. Интерактивная сторона общения».

Подобрать упражнения для развития умений подстройки и ведения собеседника в процессе общения, противостояния манипулированию

Составить структурно-логическую схему по теме занятия «Правила делового этикета». Разработать сообщение по технике безопасности на рабочем месте.

*Тема. Психология профессионального отбора и подбора персонала*

Контрольные вопросы

в Профессиональный отбор и подбор. Необходимость профессионального отбора. Психология профессиональной пригодности. Система профессиональной сертификации и квалификации. Профессиональные стандарты.

в Методы измерения и оценки профессионального развития персонала. Валидность, надежность, трудоемкость методов. Психологическое тестирование. Тесты общих и специальных способностей. Психомоторные тесты. Личностные опросники.

в Методики определения профессионально важных качеств. Диагностика профессиональной пригодности, профессиональной направленности, профессиональной мобильности, профессиональной готовности, карьерных ориентаций.

в Методики профессионального отбора на различные виды инженерной деятельности. Методики отбора менеджеров и руководителей в области промышленного производства. Методики изучения стилей деятельности и руководства.

***Задание для самоподготовки***

Составить резюме и подготовить портфолио для проведения игры.

Подобрать методики диагностики профессионально важных и профессионально значимых качеств личности работников.

*Тема. Технологии социально-профессионального саморазвития*

Контрольные вопросы:

и Саморазвитие личности. Приемы саморазвития личности. Профессиональное самосовершенствование на разных этапах профессионального становления. Роль обучения в профессиональном развитии личности.

и Виды обучения: с отрывом и без отрыва от производства. Методы обучения (лекция, кейс-стади, ролевая игра, деловая игра, групповая дискуссия, брейнсторминг).



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

\* Тренинги профессионального самосовершенствования личности. Стратегии преодоления психологических барьеров профессионального развития.

***Задания для самоподготовки***

Составить «банк» способов и приемов работы над собой известных людей (Демосфен, А.В. Суворов, М.И. Кутузов, Л.Н. Толстой, Г.К. Жуков, Эйзенхауэр и др.)  
Составить программу саморазвития (цель, препятствия, ресурсы, способы, время)

***Самостоятельная работа № 4***

*Тема:* Подготовка к зачету

*Цель:*

– подготовка к выполнению зачетного задания.

*Изучение литературных и электронных источников:*

– Профессиональная деятельность как предмет научного изучения психологии

– Психологические основы профессиональной деятельности

– Влияние профессиональной деятельности на индивидуально-психологические особенности личности

– Влияние особенностей личности на эффективность профессиональной деятельности

– Социально-психологические проблемы профессиональной деятельности

– Деловые конфликты

– Деловое общение и профессиональная деятельность

– Психология профессионального отбора и подбора персонала

– Технологии социально-профессионального саморазвития

*Практическое задание.* Выполнение тестовых заданий по модулю «Факультативный модуль Эмоциональный интеллект»

## **Физическая культура**

### **Методические указания по практическим занятиям «Профессиональной подготовке студентов»**

#### **Занятие 1**

#### **ЦЕННОСТНЫЕ ОРИЕНТАЦИИ И ОТНОШЕНИЕ СТУДЕНТОВ К ФИЗИЧЕСКОЙ КУЛЬТУРЕ.**

*Цель:* формирование ценностных ориентаций к физической культуре у студентов.

*Содержание занятия:*

37) Основные понятия о ценностных отношениях студентов к физической культуре.

38) Сущность понятия физическая культура человека.

39) Социальные функции физической культуры в вузе.

40) Организация методико-практического занятия.

*Ход занятия:*

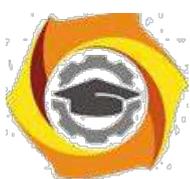
1. Преподаватель сообщает цель, задачи, содержание занятия, знакомит с ценностями физической культуры по качественному критерию.

с Рассматриваются мотивационные варианты выбора студентами направления регулярных занятий спортом.

с Разбираются методики в которых содержатся ценности физической культуры, социальные функции физической культуры в вузе.

с Обсуждаются результаты методико-практического занятия.

*Оборудование:* Схемы, плакаты, рисунки, протокол занятия, мультимедийное оборудование, презентации.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

**Домашнее задание:** провести социологическое исследование с использованием анкет, включающие вопросы и касающиеся выяснения отношения студентов одногруппников к ценностям здорового образа жизни, формам проведения досуга. В результате данного занятия студенты должны:

*знать* процесс формирования ценностных ориентаций на физическую культуру, мотивационные варианты выбора направления регулярных занятий спортом.

*уметь* достичь цель физического воспитания, сформировать физическую культуру личности, решить воспитательные, образовательные, развивающие и оздоровительные задачи.

## ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Под **ценностями** понимаются предметы, явления и их свойства, необходимые обществу и личности в качестве средств удовлетворения потребностей.

**Ценностные ориентации** - важнейший компонент структуры личности, который определяет ее поведение и отношение к окружающему миру.

**Материальные ценности** - условия занятий, качество спортивной экипировки, льготы со стороны общества;

**Физические ценности** - здоровье, телосложение, двигательные умения и навыки, физические качества, физическая подготовленность;

**Социально- психологические ценности** - отдых, развлечение, удовольствие, трудолюбие, навыки поведения в коллективе, чувства долга, чести, совести, благородства, средства воспитания и социализации, рекорды, победы, традиции; **Психически ценности** - эмоциональные переживания, черты характера, свойства и качества личности, творческие задатки;

**Культурные ценности** - познание, самоутверждение, самоуважение, чувство собственного достоинства, эстетические и нравственные качества, общение, авторитет.

**Тема «Социально-биологические основы адаптации организма человека к физической и умственной деятельности, факторам среды обитания»**

**Занятие 2**

**МЕТОДЫ ОЦЕНКИ ФУНКЦИОНАЛЬНОГО СОСТОЯНИЯ И ФИЗИЧЕСКОГО РАЗВИТИЯ ОРГАНИЗМА. (4 ЧАСА)**

**Цель:** освоить методы оценки телосложения, физического развития, функционального состояния.

**Содержание занятия**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

1. Понятие о контроле и самоконтроле
2. Методика оценки функционального состояния сердечно - сосудистой системы в условиях покоя.
- 25) Методика оценки функционального состояния сердечно-сосудистой системы с помощью функциональных проб.
- 26) Методика оценки функционального состояния дыхательной системы.
- 27) Методика оценки функционального состояния нервной системы
- 28) Методы оценки антропометрических признаков
7. Организация и содержание методико-практического занятия.

**Оборудование:** секундомеры, метроном, динамометр, тумба (высота от 30 до 50 см), сантиметровая лента, таблицы, протокол занятия.

**Ход занятия.**

1. Преподаватель кратко сообщает цель, задачи, структуру занятия.
2. Студенты одновременно выполняют функциональные пробы для оценки дыхательной и нервной системы:
  - а) пробу задержки дыхания на вдохе. После пробы результат заносят в протокол;
  - б) одновременный подсчет частоты дыхания в течение одной минуты. Результат заносят в протокол;
  - в) пробу задержки дыхания на выдохе;
  - г) пробу Ромберга;
  - д) пробу Яроцкого.Между пробами задержки дыхания на вдохе и выдохе должно быть время для отдыха. Поэтому после первой пробы сначала подсчитывают частоту дыхания, заносят результат в протокол (т.е. дают время для восстановления дыхания) и только потом выполняют пробу на выдохе, и результат заносят в протокол.
3. Студенты одновременно выполняют действия для оценки функционального состояния сердечно-сосудистой:
  - а) подсчет пульса в состоянии покоя.
  - б) измерение артериального давления. Работу можно выполнять в парах, тройках под руководством и инструкцией последовательности действий преподавателя, результаты заносят в протокол;
  - в) проведение функциональной пробы Руфье. Студенты ведут наблюдение за техникой выполнения, участвуют в математических расчетах результатов тестирования. Полученные данные записывают в свой протокол.
4. Студенты одновременно выполняют действия для оценки антропометрических показателей используя методы индексов.
  - а) измерение роста и массы тела. Производят расчет индекса Брока и Кетле.
  - б) измерение окружности грудной клетки. Определяют показатель крепости телосложения.
  - в) проведение динамометрии о подсчёт силового индекса.Полученные данные записывают в свой протокол.
5. Сопоставляются все полученные величины исследования. Дается оценка функционального состояния систем, физического развития, телосложения.
6. Обсуждаются результаты методико-практического занятия.

**Домашнее задание**

Провести самостоятельно ортостатическую функциональную пробу и гарвардский степ тест. Результаты занести в протокол занятия, при необходимости обсудить с преподавателем. В результате проведенного занятия студенты должны:  
*знать* простые методы самоконтроля за функциональным состоянием организма;  
*уметь* проводить функциональные пробы и анализировать реакцию организма на выполненную физическую нагрузку.

**ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ:**

**Функциональное состояние** — это относительно устойчивая структура актуализируемых субъектом средств деятельности в конкретной ситуации, которая отражает специфику



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

сложившихся на текущий момент времени механизмов регуляции деятельности и определяет эффективность решения трудовых задач.

**Физическое развитие** - это процесс изменения форм и функций организма человека под влиянием условий жизни и воспитания. **Самоконтроль при занятиях физкультурой и спортом** — регулярное наблюдение занимающегося физкультурой и спортом за состоянием своего здоровья, физического развития, а также за влиянием на организм физических нагрузок.

**Функциональные пробы** позволяют оценивать общее состояние организма, его резервные возможности, особенности адаптации различных систем к физическим нагрузкам, которые в ряде случаев имитируют стрессорные воздействия.

**Тема: «Социально-биологические основы адаптации организма человека к физической и умственной деятельности, факторам среды обитания»**

## **Занятие 3**

### **ОСНОВЫ МАССАЖА И САМОМАССАЖА**

**Цель:** освоить методику проведения массажа и самомассажа.

**Содержание занятия:**

1 Основные понятия массажа Противопоказания к массажу.

2. Приемы массажа и самомассажа.

16. Методика выполнения приемов массажа.

17. Гигиенические требования к массажу.

**Ход занятия**

43. Преподаватель кратко сообщает цель, задачи, структуру занятия.

44. Обучение основным массажным приемам:

13. 1 Рассказ о технике выполнения каждого из массажных приемов и показ на практике;

В2 Самостоятельное выполнение студентами массажных приемов;

е 3 Разбор основных ошибок при выполнении массажных приемов;

В4 Разбор способов устранения ошибок при выполнении массажных приемов.

В Обучение различным методикам массажа:

с 1 Разбор методики выполнения массажа на примере одной из представленных;

- 2 Массаж под наблюдением преподавателя;

16. 3 Разбор основных ошибок при выполнении массажа;

36. 4 Самостоятельное выполнение массажа.

37. Обучение методике самомассажа:

в 1 Разбор методики выполнения самомассажа;

В2 Студенты выполняют приемы самомассажа под наблюдением преподавателя;

с 3 Разбор основных ошибок при выполнении самомассажа.

**Оборудование:** кушетка, тальк, массажный крем.

**Домашнее задание**

и Выполнить косметический (профилактический) массаж.

и Тренировка в выполнении разминающих и выжимающих приемов

и Массаж при различных травмах верхних и нижних конечностей.

**ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ:**

**Массаж** - метод профилактики и лечения, представляющий собой совокупность приемов дозированного механического воздействия на различные участки тела человека, производимого руками массажиста или специальными аппарата

При массаже применяются определенные **приемы**, их можно подразделить на пять основных групп. К ним относятся:

поглаживание;

растирание;

выжимание,



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

разминание;  
вибрация.

Существуют следующие **виды массажа**:

1. Спортивный. 2. Лечебный. 3. Гигиенический. 4. Косметический.

Применяются следующие **формы массажа**:

х Общий массаж. 2. Локальный массаж. 3. Самомассаж. 4. Парный массаж. 5.  
Взаимный массаж.

в зависимости от способа выполнения массажа можно выделить следующие его **методы**:

1. Ручной массаж 2. Аппаратный массаж. 3. Комбинированный массаж. 4. Ножной массаж.

**Тема: «Образ жизни и его отражение в профессиональной деятельности»**

**Занятие 4**

## **МОДЕЛИРОВАНИЕ ИНДИВИДУАЛЬНОЙ ЗДОРОВЬЕСБЕРЕГАЮЩЕЙ ЖИЗНЕДЕЯТЕЛЬНОСТИ.**

**Цель:** формирование, развитие и сохранение здоровья.

**Содержание занятия:**

20 Формирование ценностного отношения к здоровью.

21 Внедрение здоровьесберегающих технологий в физкультурно-оздоровительном процессе.

22 Определение оптимально физкультурно-оздоровительной деятельности.

**Ход занятия:**

16. Преподаватель сообщает цель, задачи, содержания занятия, знакомит с реализацией индивидуальных физкультурно-оздоровительных программ физической подготовки.

17. Разбираются развитие когнитивных способностей, формирование физического здоровья.

18. Обсуждаются результаты методико-практического занятия.

**Оборудование:** рисунки, плакаты, схемы, протокол занятия, мультимедийное оборудование.

**Домашнее задание:**

провести диагностику своего здоровья и образа жизни друзей, знакомых, близких.

В результате данного занятия студенты должны:

*знать* систему урочных форм занятий по физической подготовке, систему внеурочных форм физического воспитания, систему вузовских спортивно-оздоровительных мероприятий;  
*уметь* оценить эффективность физкультурно-оздоровительного процесса.

## **ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ**

**1. Здоровьесберегающая образовательная технология** - функциональная система организационных способов управления учебно - познавательной и практической деятельности учащихся, научно и инструментально обеспечивающая сохранение и укрепление их здоровья.

**27. Гипокинезия** - состояние недостаточной двигательной активности организма с ограничением темпа и объёма движений.

**63. Гиподинамия** - нарушение функций организма (опорно-двигательного аппарата, кровообращения, дыхания, пищеварения) при ограничении двигательной активности, снижении силы сокращения мышц.

**Тема: «Общая физическая, специальная и спортивная подготовка в системе физического воспитания»**

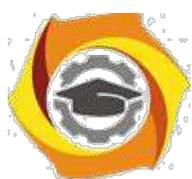
**Занятие 5**

## **МЕТОДИКА РАЗВИТИЯ ФИЗИЧЕСКИХ КАЧЕСТВ**

**Цель :** освоить методику развития основных физических качеств.

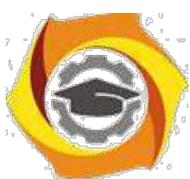
**Содержание занятия:**

- Основные понятия физических качеств.
- Методика развития выносливости.
- Методика развития силы.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Методика развития быстроты.
- Методика развития гибкости.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

6. Методика развития ловкости.

**Ход занятия:**

- Преподаватель сообщает цель, задачи, содержание занятия, знакомит с основами методики развития физических качеств: выносливости, силы, быстроты, гибкости, ловкости.
- Разбираются методики развития отдельных физических качеств (приложение 1). Используя данные методики (схемы, таблицы, рисованные комплексы упражнений для развития того или иного физического качества), студенты под руководством преподавателя разрабатывают содержание тренировочных занятий, построенных по тому или иному методу тренировки. Физические упражнения для развития ловкости, гибкости могут быть продемонстрированы как с использованием рисованных комплексов, так и подготовленным студентом.
- Записывается в протокол (приложение 2) ряд физических упражнений для развития физического качества (по выбору) тем или иным методом тренировки.
- Обсуждаются результаты методико-практического занятия.

Приложение 1

**Протокол занятия**

Тема: методика развития физических качеств

Студент

Спортивная специализация

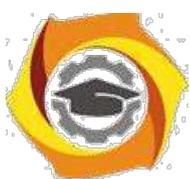
Физическое качество

Задачи

Методы развития физических качеств	Содержание	Продолжительность	Темп или скорость	Кол-во повторений	Кол-во серий	Интервалы отдыха
Равномерный непрерывный	1. 2. 3.					
Равномерный переменный	1. 2. 3.					
Интервальный	1. 2. 3.					
Круговая тренировка	1. 2. 3.					

Приложение 2

№ п/п	Физические качества	Методы	Характеристики
1.	Сила	Максимальных усилий	>90% максимальных усилий. 1-3 повторений, 5-6 серий. Отдых 4-8 мин (до восстановления). 1 раз в 1-2 недели
		Повторный	От 40 до 80% от максимальных усилий. 15-25 повторений, 3-6 серий. Отдых до восстановления
		Интервальный	От 40 до 80% от максимальных усилий. 5-10 повторений. Отдых 2-4 мин
		Круговой	Серийное выполнение упражнений по схеме на определенном месте «станции». 4-5 «станций», 2-3 круга, от 40 до 80% от максимальных усилий. 5-10 повторений. Отдых 2-4 мин



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

2.	Выносливость	Изометрический	Максимальное напряжение 4-6 сек, 3-5 серий
		Повторный	До 30% от рекордного. 15-25 повторений, 4-6 серий. Отдых до восстановления
		Интервальный	До 30% от рекордного. 15-25 повторений, 4-6 серий. Отдых 2-4 мин
		Круговой	Серийное выполнение упражнений по схеме на определенном месте «станции», 5-6 «станций», 4-5 кругов, до 30% от максимальных усилий. 15-25 повторений. Отдых 2-4 мин
		Переменный	Выполнение упражнений с ритмичными и неритмичными колебаниями интенсивности (с разной скоростью, темпом, длительностью, амплитудой, величиной усилий, сменой техники движений и т. д.) от 10 мин до 1 часа
	Равномерный	Выполнение упражнений непрерывно с относительно постоянной интенсивностью, от 15 до 90 мин и более	
3.	Быстрота	Повторный	Максимальные частота и темп выполнения, короткая продолжительность (3-10 сек, 30-100 мин), отдых до полного восстановления
		Круговой	Серийное выполнение упражнений по схеме на определенном месте «станции», 4-5 станций», 2-3 круга, до 90% от максимальных усилий (3-10 сек, 30-100 мин). Отдых до полного восстановления
		Игровой	Выполнение разнообразных упражнений с максимально возможной скоростью в условиях проведения подвижных и спортивных игр
		Соревновательный	применяется в форме различных тренировочных состязаний (прикидки, эстафеты, гандикапы — уравнивательные соревнования) и соревнований
4.	Гибкость	Повторный	Упражнения на гибкость до появления болевых ощущений 15-30 сек x 5-6 раз, отдых 30-60 сек
		Изометрический	Максимальное напряжение в статических позах 15-30 сек X 5-6 раз, отдых 30-60 сек
5.	Ловкость	Повторный	Повторение упражнений со сложной координацией движений 5-10-минутными сериями, 2-3 серии
		Игровой	Разучивание новых упражнений, выполнение упражнений в меняющихся, нестандартных условиях 15-20-минутными сериями, 2-3 серии
		Соревновательный	Выполнение упражнений в меняющихся, нестандартных условиях

**Оборудование:** спортивный инвентарь для развития быстроты, силы, гибкости и ловкости, секундомеры, ориентиры, рисунки, плакаты, схемы, протокол занятия.

**Домашнее задание:** самостоятельно разработать и составить методику совершенствования своего наименее развитого физического качества. В результате данного занятия студенты должны:

*знать* упражнения и виды спорта, развивающие физические качества, методы совершенствования отдельных физических качеств;  
*уметь* индивидуально подбирать средства и методы направленного развития и совершенствования отдельных физических качеств.

#### ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

**1. Физические (двигательные) качества** - определенные стороны двигательных способностей человека: сила, выносливость, быстрота (скоростные возможности), гибкость, ловкость (координационные возможности)

и **Сила** - способность преодолевать внешнее сопротивление или противодействовать ему посредством мышечных напряжений. Развитие силы сопровождается утолщением и ростом мышечных волокон.

**3. Выносливость** - это способность противостоять физическому утомлению в процессе мышечной деятельности. Различают два вида выносливости: общую и специальную. Общая выносливость - способность выполнять работу с невысокой интенсивностью в течение продолжительного



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

времени за счет аэробных источников энергообеспечения. Специальная выносливость - это способность эффективно выполнять работу в определенной трудовой или спортивной деятельности. 14

**4.Быстрота** - комплекс функциональных свойств организма, определяющих скоростные характеристики движений, а также двигательной реакции.

**5.Гибкость** - способность выполнять упражнения с большой амплитудой.

**6.Координационные способности** (ловкость, устойчивость равновесия, пространственная ориентация) - это способности быстро, точно, целесообразно, экономно решать двигательные задачи.

**Тема «Общая физическая, специальная и спортивная подготовка в  
системе физического воспитания»**

**Занятие 6**

**ОСНОВЫ ПОСТРОЕНИЯ И СТРУКТУРА ЗАНЯТИЯ (УРОКА)  
ПО ФИЗИЧЕСКОЙ КУЛЬТУРЕ**

**Цель:** сформировать у студентов знания о структуре и построении занятия (урока) по физической культуре.

**Содержание занятия**

ее Теоретические основы построения занятий(урока) по физической культуре 1.1 Формы занятий по физической культуре 1.2 Структура и построение урока (занятия) по физической культуре

В 3 Задачи учебного урока (занятия)

45. 4 Методы организации занимающихся на уроке (занятие)

2.Методика обучения построения занятия (урока) по физической культуре

46. 1 Разбор порядка построения подготовительной части занятия

37. 2 Разбор порядка построения основной части занятия

32) 3.Разбор порядка построения заключительной части занятия

В 4 Разбор порядка организации и методики проведения урока (занятия)

**Ход занятия:**

44. Преподаватель сообщает цель, задачи, содержание занятия, знакомит с теоретическими основами (формами) построения занятий по физической культуре.

45. Разбирают структуру и построения занятия, используя данные методики. Составляют совместно с преподавателям комплекс упражнений для подготовительной части занятия.

46. Составляют совместно с преподавателям план учебно-тренировочного занятия по выбору студента.

47. Обсуждаются результаты методико-практического занятия

**Оборудование:** спортивный инвентарь для развития быстроты, силы, гибкости и ловкости, секундомеры, ориентиры, рисунки, плакаты, схемы, протокол занятия.

**Домашнее задание:** самостоятельно разработать и составить план учебно-тренировочного занятия по аэробике или силовой подготовке

В результате данного занятия студенты должны:

*Знать* теоретические основы о структуре занятия и методики ее проведения

*Уметь* практически составлять комплексы подготовительной, основной и заключительной части занятия, а также проведения учебного занятия по избранному виду спорту.

**ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ**

**103 построении каждого урока различаются три части:** - первая часть – организующая, подготавливающая учащихся к выполнению наиболее трудных задач урока, условно называется вводной; - вторая часть – направленная на выполнение основных задач урока,



**Негосударственное частное образовательное  
учреждение  
высшего образования  
«Технический университет»**

условно называется основной; - третья, во время которой организовано завершается урок, условно называется заключительной.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Учебные занятия** - основная форма физического воспитания. Учебные занятия могут быть: теоретические, практические, контрольные, элективные практические занятия (по выбору) и факультативные; индивидуальные и индивидуально-групповые дополнительные занятия (консультации); самостоятельные занятия по заданию и под контролем преподавателя.

**Тема «Методические основы самостоятельных занятий  
физическими упражнениями и самоконтроль в процессе занятий»**

**Занятие 7**

**МЕТОДИКА ОЦЕНКИ И САМООЦЕНКИ РАБОТОСПОСОБНОСТИ. ПРИМЕНЕНИЕ  
СРЕДСТВ ФИЗИЧЕСКОЙ КУЛЬТУРЫ ДЛЯ ПОВЫШЕНИЯ ФУНКЦИОНАЛЬНЫХ  
СПОСОБНОСТЕЙ ОРГАНИЗМА**

**Цель:** овладеть самостоятельным, регулярным наблюдениям простыми и доступными способами за физическим развитием, состоянием своего организма, влиянием на него физических упражнений или вида спорта.

**Содержание занятия**

34. Психофизиологические компоненты работоспособности

2. Виды работоспособности

/ Стадии работоспособности

/ Способы оценки работоспособности

/ Производственное утомление и способы его снятия средствами физической культуры

6. Дневник самоконтроля

**Ход занятия:**

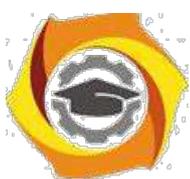
/ Преподаватель сообщает цель, задачи, содержание занятия, напоминает, что включают в себя компоненты работоспособности. Доводит до сведения тесты, которые будут апробироваться на занятии:

- степ-тест

- оценка умственной работоспособности по Э. Крепелину;

Определение PWC170 методом степэргометрии. Испытуемый в течение 3 минут совершает подъемы на ступень высотой 35 см с частотой 20 подъемов в минуту (частота метронома 80 ударов в минуту). На один удар метронома совершается одно движение. По окончании нагрузки считают пульс в течение 10 с (P1). Далее выполняется вторая нагрузка с частотой 30 подъемов в минуту (120 уд/мин). По окончании второй нагрузки снова считают пульс (P2). Затем определяют показатель PWC170 с помощью таблицы 3.5. На горизонтальной линии находят ЧСС после первой нагрузки, а на вертикальной, соответственно, после второй. Пересечение двух показателей дает величину относительного PWC170 в пересчете на 1 кг веса тела. Общая работоспособность рассчитывается следующим образом: PWC170 (кгм/мин) = А \* М, где: А - величина относительного PWC170 М - масса тела испытуемого.

Пульс за 10 с при подъеме на ступеньку														
2 нагрузка (P <sub>2</sub> )	1-я нагрузка (P <sub>1</sub> )													
	15	16	17	18	19	20	21	22	23	24	25	26	27	28
18	22,7													
19	18,9	21,9												
20	16,6	18,2	20,7											
21	15,0	16,0	17,3	19,2										
22	13,8	14,5	15,3	16,2	18,0									
23	13,0	13,5	13,9	14,4	15,3	16,8								
24	12,4	12,7	12,9	13,2	13,7	14,4	15,6							
25	11,9	12,1	12,2	12,3	12,6	13,0	13,5	14,4						
26	11,4	11,6	11,7	11,7	11,8	11,9	12,7	12,6	13,2					
27	11,2	11,2	11,2	11,2	11,2	11,3	11,4	11,5	11,7	12,0				
28	10,8	10,8	10,8	10,8	10,8	10,8	10,8	10,8	10,8	10,8	10,8			
29	10,5	10,5	10,4	10,4	10,4	10,4	10,3	10,2	10,2	10,1	9,6	9,6		
30	10,3	10,3	10,2	10,2	10,1	10,1	9,9	9,9	9,7	9,6	9,4	9,0	8,4	
31	10,1	10,1	10,0	9,9	9,8	9,8	9,7	9,6	9,4	9,2	9,0	8,6	8,1	7,2
32	10,0	9,9	9,8	9,7	9,6	9,6	9,4	9,1	9,0	8,7	8,4	7,9	7,6	7,2
33	9,8	9,8	9,6	9,6	9,5	9,4	9,3	9,1	9,0	8,6	8,5	8,2	7,8	7,2



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

Тест Э. Крепелина «Когда я дам команду "Начинайте!", вы как можно быстрее, и без ошибок должны начать складывать пары чисел, напечатанные на бланке. Если сумма больше 10, то, отбросив десятки, пишете только единицы. Каждый раз, когда я буду говорить "Черта!", на том месте, где вас застанет этот сигнал, вы должны поставить вертикальную черту и сразу же продолжать работу, пока я не дам команду: "Стоп! Переверните лист"». Ведущий дает команду «Начинайте!». Каждые 15 секунд говорит: «Черта!». Дает команду «Стоп! Переверните лист» и останавливает секундомер, когда будет закончен бланк при индивидуальном исследовании или поднята рука – при групповом.

2. Разбираются методики проведения каждой пробы (теста). Используя данные методики, назначенные студенты под руководством преподавателя, проводят поочередно данные тесты со студентами группы в качестве руководителя и организатора.

3. Рассматриваются степени утомления при выполнении умственной и физической работы. Составляются комплексы упражнений для снятия возникшего утомления при умственной и физической нагрузке. После обсуждения данных комплексов, автор наиболее понравившегося проводит данный комплекс с группой.

/ Данные самонаблюдений записываются в дневник самоконтроля

## 6. ДНЕВНИК САМОКОНТРОЛЯ

№	Показатели самоконтроля	День тренировки	День отдыха
1.	Пульс (ЧСС) утром лежа за 15 сек.		
2.	Пульс утром стоя за 15 сек.		
3.	Масса тела до тренировки		
4.	Масса тела после тренировки		
5.	Жалобы		
6.	Самочувствие		
7.	Сон		
8.	Аппетит		
9.	Болевые ощущения		
10.	Желание тренироваться		
11.	Потоотделение		
12.	Кистевая динамометрия - до тренировки - после тренировки		
13.	Настроение		
14.	Функция кишечника		
15.	Работоспособность		
16.	Спортивные результаты		
17.	Нарушение режима		
18.	Содержание тренировки		
19.	Перенос нагрузки		

4. Обсуждаются результаты методико-практического занятия.

**Оборудование:** ступень высотой 35 см, секундомер, спирометр, метроном, маты гимнастические, измерительная лента, дневник самоконтроля.

**Домашнее задание:**

/ Коррекция содержания и методики занятий физическими упражнениями по результатам контроля.

/ Методики самооценки личности.

## ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

**Работоспособность** — потенциальная возможность индивида выполнять целесообразную деятельность на заданном уровне эффективности в течение определенного времени.

Работоспособность рассматривается:

как способность человека выполнять определенную работу на определенном уровне качества и надежности

как нечто тождественное функциональному состоянию организма,



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

как способность обеспечивать определенный заданный уровень деятельности, эффективность работы (что тождественно понятию производительности труда), как предельные возможности организма.

Традиционно выделяют следующие **стадии работоспособности**:

вработываемость,  
оптимальная работоспособность,  
компенсируемое и некомпенсируемое утомление,  
конечный «порыв»,  
прогрессивное снижение продуктивности

**Тема «Особенности занятий избранным видом спорта»**

**Занятие 8**

**МЕТОДИКА СОСТАВЛЕНИЯ ИНДИВИДУАЛЬНОГО ЗАНЯТИЯ ПО ИЗБРАННОМУ ВИДУ СПОРТА**

**Цель:** ознакомиться с методикой составления и проведения самостоятельных занятий с гигиенической и тренировочной направленностью на примере занятия по легкой атлетике.

**Содержание занятия:**

- В Подготовка к проведению занятий
- В Структура и методика проведения подготовительной части занятия
- В Структура и методика проведения основной части занятия
- В Проведение заключительной части занятия
- В Составление плана-конспекта проведения занятий

**Ход занятия:**

- \endash Преподаватель кратко сообщает цель, задачи, структуру занятия, знакомит с простейшими формами самостоятельных занятий физическими упражнениями гигиенической и тренировочной направленности, методикой составления и проведения плана занятия.
- \endash Разбирается содержание подготовительной части занятия. Один из студентов с группой проводит подготовительную часть. Контроль за правильностью выполнения, соблюдение соответствующей последовательности выполнения упражнений осуществляет преподаватель. Студенты активно включаются в обсуждение содержания упражнений.
- \endash Разбираются возможные разделы легкой атлетике по которым целесообразно проводить занятие. Подбираются средства и методы для тренировки двигательных навыков и физических качеств. 2-3 студента проводят тренировочные упражнения для развития быстроты и выносливости
- \endash Обсудив содержание заключительной части занятия, один из студентов её проводит.
- \endash Раскрывается структура написания плана-конспекта занятия.

**ПЛАН**

**учебно-тренировочного занятия по легкой атлетике**

<i>Содержание занятия</i>	<i>Время, мин.</i>	<i>Организационно-методические указания</i>
<b>Подготовительная часть занятия (15 мин.)</b>		
<b>Основная часть занятия (65 мин.)</b>		
<b>Заключительная часть занятия (10 мин.)</b>		

6. Обсуждение результатов методико-практического занятия.



# Негосударственное частное образовательное учреждение высшего образования «Технический университет»

**Оборудование:** спортивное ядро легкоатлетического стадиона, бланки планов проведения занятий.

**Домашнее задание:**

46. Составить план-конспект практического занятия по заданной тематике

47. Ознакомиться с новыми популярными видами спорта:

-сквош

-черлидинг

-паркур

-петанк

В результате проведенного методико-практического занятия студенты должны:

**знать** особенности форм, содержания и структуры самостоятельных занятий физическими упражнениями;

**уметь** составить и провести самостоятельно занятие гигиенической или тренировочной направленности.

## ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

При разработке конспекта урока по физической культуре рекомендуется придерживаться следующей последовательности:

44. Определить основные и частные задачи.

45. Выявить средства и методические приемы для решения каждой задачи обучения двигательным действиям, а также последовательность и преемственность расположения упражнений (подводящих и основных) в каждой части урока.

46. Наметить теоретические сведения, которые следует сообщить учащимся на данном уроке.

47. Спрогнозировать место проведения и уточнить пути решения запланированных задач обучения, физического развития и укрепления здоровья.

48. Распределить в уроке учебные задания с учетом динамики работоспособности занимающихся.

49. Разработать методы организации учебной деятельности и наметить необходимое оборудование и инвентарь.

50. Определить критерии и формы оценки.

51. Определить педагогически оправданные методические приемы для каждой части урока, руководствуясь при этом задачами урока, особенностями занимающихся и конкретными условиями работы.

Вначале рекомендуется разработать основную часть урока, затем - вводную (подготовительную) и заключительную.

## Философия

### Методические указания по практическим занятиям

#### I. Введение

##### 1. Цели изучения дисциплины

Целью изучения дисциплины «Философия» является осознание студентами круга философско-антропологических и социально-философских проблем, формирование у них навыков самостоятельного аналитического и диалектического мышления в сфере гуманитарного знания, понимание принципов рационального философского подхода к процессам и тенденциям современного информационного общества.

##### 2. Планируемые результаты обучения

Изучение дисциплины направлено на формирование у выпускника способности решать профессиональные задачи в соответствии с **проектным** видом профессиональной деятельности:

Результатом освоения дисциплины являются сформированные у выпускника следующие компетенции:



**Компетенции выпускника, формируемые в результате освоения дисциплины (в части, обеспечиваемой дисциплиной)**

**Указания к практическим занятиям**

- **Методические указания к практическим занятиям очной формы обучения**

**Практическое занятие 1. Философия как вид знания**

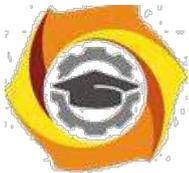
Целью проводимого занятия является формирование знаний: предмета философии, ее места в культуре; структуры философского знания; понятия мировоззрения, общее представление о мифологическом, философском, религиозном и научном мировоззрении; понятия философской рефлексии; умения анализировать мировоззренческие проблемы.

**Форма проведения:** контрольная работа по заранее известным вопросам. Обсуждение вопросов по теме.

**Вопросы к контрольной работе:**

38. Место и функции философии в культуре.
39. Мифология как исторический тип мировоззрения.
40. От мифологии к философии.
41. Специфика философского знания.
42. Основные темы философского познания.
43. Философия в понимании Платона и Аристотеля.
44. Гегель о значении философии.
45. Философия как строгая наука (Гуссерль).

**Дополнительные вопросы для обсуждения и самопроверки:**



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- в Как возникла необходимость в философии?
- в В чем специфическая особенность философского мышления?
- в Какие особенности имеет философское мировоззрение?
- в Какие основные темы философского познания?
- в Как представлялась наука философия в понимании Платона и Аристотеля?
- в Что говорил Гегель о значении философии?
- в Что означает понимание философии как строгой науки в феноменологии Гуссерля?
- в Обладает ли философия своим языком? В чем состоит его особенность?
- в Каковы отличительные признаки философского текста?
- в В чем выражается значение философии в жизни человека?

**Литература:**

- В Губин В.Д. Основы философии: учеб. пособие / В.Д. Губин. — 4-е изд. М.: ФОРУМ : ИНФРА-М, 2019. — 288 с. <https://znanium.com/catalog/product/1009582>
- В Аналитика человеческого бытия: введение в опыт самопознания. Систематический очерк : монография / С.Е. Ячин. — М. : ИНФРА-М, 2019. — 210 с. <https://znanium.com/catalog/product/989370>
- В Философия : учебник / В.А. Канке. — М. : ИНФРА-М, 2019. — 291 с. <https://znanium.com/catalog/product/977818>
- В Социальная философия и философия истории : учебное пособие / под ред. д-ра филос. наук К.Х.Момджяна. — Москва : ИНФРА-М, 2019. — 478 с. <https://znanium.com/catalog/product/947030>
- В Философия : учеб. пособие / И.А. Карпенко. — Москва : ИНФРА-М, 2019. — 190 с. <https://znanium.com/catalog/product/1039315>

**Практическое занятие 2. Античная философия. Ранняя христианская философия. схоластика.**

Целью проводимого занятия является формирование знаний: специфики раннего этапа античного философского исторического развития; антропологического периода древнегреческой философии; основных идей философии Платона и Аристотеля; средневекового этапа в истории развития философии; основных религиозно-философских (христианских) понятий и проблем человеческого бытия; общего представления о средневековой религиозной картине мироздания, сущности, назначении и смысле жизни человека.

**Форма проведения:** контрольная работа по заранее известным вопросам. Обсуждение во-просов по теме.

**Вопросы к контрольной работе:**

- с Особенности греческого мировоззрения.
- с Проблемы философии в до-сократический период.
- с Поиск первоосновы мира в античной философии.
- с Атомистическое учение Демокрита.
- с Софисты.
- с Сократ.
- с Платон.
- с Аристотель.
- с Школы поздней античности: стоицизм, эпикуреизм, скептицизм.

**Дополнительные вопросы для обсуждения и самопроверки:**

- и Каковы особенности греческого мировоззрения?
- и В чем заключались проблемы философии в до-сократический период?



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- x Какие выдвигались идеи о первооснове мира в античной философии?
- x В чем суть атомистического учения Демокрита?
- x В чем особенность взглядов софистов?
- x Каков образ личности и мышления Сократа?
- x Назовите главные идеи философии Платона?
- x Обозначьте ключевые интуиции философии Аристотеля?
- x Охарактеризуйте школы поздней античности: стоицизм, эпикуреизм, скептицизм.
- x Человек и мир в философии бл. Августина.
- x Роль философских и нравственных идей Библии в развитии мировой духовной культуры.
- x Взаимоотношения веры и знания.
- x Патристика: суть учения и основные представители.
- x Теоцентризм средневековой философии.
- x Спор между номиналистами и реалистами в средневековой философии.
- x Схоластика и ее роль в философии Средневековья
- x В чем состоят главные идеи интерпретации человека и мира в философии бл. Августина?
- x Какова роль философских и нравственных идей Библии в развитии мировой духовной культуры?

**Литература:**

- в Губин В.Д. Основы философии: учеб. пособие / В.Д. Губин. — 4-е изд. М.: ФОРУМ : ИНФРА-М, 2019. — 288 с. <https://znanium.com/catalog/product/1009582>
- в Аналитика человеческого бытия: введение в опыт самопознания. Систематический очерк : монография / С.Е. Ячин. — М. : ИНФРА-М, 2019. — 210 с. <https://znanium.com/catalog/product/989370>
- в Философия : учебник / В.А. Канке. — М. : ИНФРА-М, 2019. — 291 с. <https://znanium.com/catalog/product/977818>
- в Социальная философия и философия истории : учебное пособие / под ред. д-ра филос. наук К.Х.Момджяна. — Москва : ИНФРА-М, 2019. — 478 с. <https://znanium.com/catalog/product/947030>
- в Философия : учеб. пособие / И.А. Карпенко. — Москва : ИНФРА-М, 2019. — 190 с. <https://znanium.com/catalog/product/1039315>

**Практическое занятие 3. Философия Нового времени**

Целью проводимого занятия является формирование знаний: о начальном периоде научной революции; основных положений философии Ф. Бэкона и Р. Декарта; соотношения истины и заблуждения, особенностей функционирования знания в Новое время; проблемы соотношения духовного и телесного начал в человеке.

**Форма проведения:** контрольная работа по заранее известным вопросам. Обсуждение во-просов по теме.

**Вопросы к контрольной работе:**

- 23 Общая характеристика новоевропейской философии.
- 24 Эмпиризм и рационализм.
- 25 Учение Ф. Бэкона о разуме и о науках.
- 26 Декарт Р. Сомнение и способ установления достоверного знания. Интуиция и дедукция. Учения о врожденных идеях
- 27 Теория Т. Гоббса.
- 28 Учение Д. Локка.
- 29 Сравнительный учений Г. Лейбница и Д. Беркли.
- 30 Д. Юм: возможна ли наука.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

**Дополнительные вопросы для обсуждения и самопроверки:**

19. Дайте общую характеристику новоевропейской философии.
20. Обоснуйте главные идеи эмпиризма и рационализма?
21. В чем специфика учения Ф. Бэкона о разуме и о науках?
22. Изложите идеи Декарта о сомнении, способе установления достоверного знания, интуиции и дедукции, учения о врожденных идеях.
23. В чем суть теории Т. Гоббса.
24. Покажите элементы философского учения Д. Локка.
25. Проведите сравнительный анализ учений Г. Лейбница и Д. Беркли.
26. Как вы считаете, возможна ли наука согласно учению Д.Юма?

**Литература:**

28. Губин В.Д. Основы философии: учеб. пособие / В.Д. Губин. — 4-е изд. М.: ФОРУМ : ИНФРА-М, 2019. — 288 с. <https://znanium.com/catalog/product/1009582>
29. Аналитика человеческого бытия: введение в опыт самопознания. Систематический очерк : монография / С.Е. Ячин. — М. : ИНФРА-М, 2019. — 210 с. <https://znanium.com/catalog/product/989370>
30. Философия : учебник / В.А. Канке. — М. : ИНФРА-М, 2019. — 291 с. <https://znanium.com/catalog/product/977818>
31. Социальная философия и философия истории : учебное пособие / под ред. д-ра филос. наук К.Х.Момджяна. — Москва : ИНФРА-М, 2019. — 478 с. <https://znanium.com/catalog/product/947030>
32. Философия : учеб. пособие / И.А. Карпенко. — Москва : ИНФРА-М, 2019. — 190 с. <https://znanium.com/catalog/product/1039315>

**Практическое занятие 4. Немецкая философия. Позитивизм и экзистенциализм XIX – XX вв.**

Целью проводимого занятия является формирование знаний: основных положений учений о человеке, его активной деятельности и свободе И. Канта, И.Г. Фихте, Ф.В.Й. Шеллинга, Г.В.Ф. Гегеля; структуры и методов диалектического познания; развитие современной философии; особенность научной философии; формирование экзистенциальной философии; отношение современной культуры и философских учений.

**Форма проведения:** контрольная работа по заранее известным вопросам. Обсуждение вопросов по теме.

**Вопросы к контрольной работе:**

64. Философские идеалы эпохи Просвещения.
65. И. Кант: основные философские идеи.
66. Г. Гегель: система диалектики.
67. Антропологический материализм Л. Фейербаха.
68. Философия А. Шопенгауэра и Ф. Ницше.
69. Принцип историзма, разработанный в немецкой классической философии
70. Философская концепция К.Маркса
71. Человек в «философии жизни»
72. Философские аспекты учений психоанализа и неопрецидизма
73. Смысл и задачи позитивной философии.
74. Психологическое обоснование логики в первом позитивизме.
75. Аналитическая философия.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

- Феноменология Э.Гуссерля.
- Проблемы истины в неопозитивизме.
- Философские воззрения С. Кьеркегора.
- Экзистенциальная проблематика в творчестве Ф. Достоевского.
- Культура и человек в учении М. Хайдеггера.

**Дополнительные вопросы для обсуждения и самопроверки:**

- Каковы философские идеалы эпохи Просвещения?
- В чем основные философские идеи И. Канта?
- Изложите элементы учения Г. Гегеля.
- В чем особенность антропологического материализма Л. Фейербаха?
- Какие главные идеи философия А. Шопенгауэра и Ф. Ницше?
- Изложите структуру философской концепции К.Маркса.
- Как понимается человек в «философии жизни»?
- Каков смысл и задачи позитивной философии?
- В чем заключается идея и смысл психологического обоснования логики в первом позитивизме?
- Изложите идеи феноменологии Э.Гуссерля.
- Поясните проблематику истины в неопозитивизме?
- В чем особенность философских воззрений С. Кьеркегора?
- Поясните экзистенциальную проблематику в творчестве Ф. Достоевского.
- Как понимается культура и человек в учении М. Хайдеггера?

**Литература:**

и Губин В.Д. Основы философии: учеб. пособие / В.Д. Губин. — 4-е изд. М.: ФОРУМ : ИНФРА-М, 2019. — 288 с. <https://znanium.com/catalog/product/1009582>

и Аналитика человеческого бытия: введение в опыт самопознания. Систематический очерк : монография / С.Е. Ячин. — М. : ИНФРА-М, 2019. — 210 с.

<https://znanium.com/catalog/product/989370>

и Философия : учебник / В.А. Канке. — М. : ИНФРА-М, 2019. — 291 с. <https://znanium.com/catalog/product/977818>

и Социальная философия и философия истории : учебное пособие / под ред. д-ра филос. наук К.Х.Момджяна. — Москва : ИНФРА-М, 2019. — 478 с.

<https://znanium.com/catalog/product/947030>

и Философия : учеб. пособие / И.А. Карпенко. — Москва : ИНФРА-М, 2019. — 190 с. <https://znanium.com/catalog/product/1039315>

**Практическое занятие 5. Русская философия**

Целью проводимого занятия является формирование знаний: особенностей и проблематики русской философии; проблемы путей исторического развития России; назначения и смысла жизни человека; анализировать мировоззренческие, социально и личностно значимые философские проблемы; приемами ведения дискуссии, полемики и диалога.

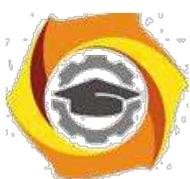
**Форма проведения:** контрольная работа по заранее известным вопросам. Обсуждение во-просов по теме.

**Вопросы к контрольной работе:**

ее Славянофилы и западники.

ее Учение Вл. Соловьева о всеединстве.

ее Проблема человека у Н. Бердяева



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

В Разум и вера в понимании Льва Шестова.  
В Специфика русской философии.

**Дополнительные вопросы для обсуждения и самопроверки:**

47. Укажите основных представителей и идеи славянофилов и западников.
48. Каков характер учения Вл. Соловьева?
49. Как решалась проблема человека у Н. Бердяева?
50. Как понимается отношение разума и веры в философии Льва Шестова?

**Литература:**

38. Губин В.Д. Основы философии: учеб. пособие / В.Д. Губин. — 4-е изд. М.: ФОРУМ : ИНФРА-М, 2019. — 288 с. <https://znanium.com/catalog/product/1009582>
39. Аналитика человеческого бытия: введение в опыт самопознания. Систематический очерк : монография / С.Е. Ячин. — М. : ИНФРА-М, 2019. — 210 с. <https://znanium.com/catalog/product/989370>
40. Философия : учебник / В.А. Канке. — М. : ИНФРА-М, 2019. — 291 с. <https://znanium.com/catalog/product/977818>
41. Социальная философия и философия истории : учебное пособие / под ред. д-ра филос. наук К.Х.Момджяна. — Москва : ИНФРА-М, 2019. — 478 с. <https://znanium.com/catalog/product/947030>
42. Философия : учеб. пособие / И.А. Карпенко. — Москва : ИНФРА-М, 2019. — 190 с. <https://znanium.com/catalog/product/1039315>

**Практическое занятие 6. Проблема Бытия**

Целью проводимого занятия является формирование знаний: уяснение понятий «бытие», «сущность», «природа», «субстанция»; выяснение ценностной иерархии бытия; характеристика человеческого бытия; проблема познания бытия; бытия и язык.

**Форма проведения:** контрольная работа по заранее известным вопросам. Обсуждение во-просов по теме.

**Вопросы к контрольной работе:**

- 33) Онтология как учение о бытии.
- 34) Монистические и плюралистические концепции.
- 35) Реальность объективная и субъективная.
- 36) Понятия «мир», «реальность» и «картина мира».
- 37) Условия формирования картины мира и ее изменения.
- 38) Понятие идеального. Идеальное и реальное, идеальное и идеал.

**Дополнительные вопросы для обсуждения и самопроверки:**

- В Что представляет собой онтология как учение о бытии?  
В Обозначьте специфику монистических и плюралистических концепций бытия.  
В В чем принципиальная разница между объективной и субъективной реальностью?  
В Дайте определения понятиям «мир», «реальность» и «картина мира».  
В В чем заключаются условия формирования картины мира и ее изменения?  
В Поясните смысл понятия идеального.

**Литература:**



48. Губин В.Д. Основы философии: учеб. пособие / В.Д. Губин. — 4-е изд. М.: ФОРУМ : ИНФРА-М, 2019. — 288 с. <https://znanium.com/catalog/product/1009582>
49. Аналитика человеческого бытия: введение в опыт самопознания. Систематический очерк : монография / С.Е. Ячин. — М. : ИНФРА-М, 2019. — 210 с. <https://znanium.com/catalog/product/989370>
50. Философия : учебник / В.А. Канке. — М. : ИНФРА-М, 2019. — 291 с. <https://znanium.com/catalog/product/977818>
51. Социальная философия и философия истории : учебное пособие / под ред. д-ра филос. наук К.Х.Момджяна. — Москва : ИНФРА-М, 2019. — 478 с. <https://znanium.com/catalog/product/947030>
52. Философия : учеб. пособие / И.А. Карпенко. — Москва : ИНФРА-М, 2019. — 190 с. <https://znanium.com/catalog/product/1039315>

### Практическое занятие 7. Современные концепции бытия

Целью проводимого занятия является формирование знаний: современные концепции мироустройства; современные физические и космологические концепции; теории томизма; проблемы социального бытия.

**Форма проведения:** контрольная работа по заранее известным вопросам. Обсуждение во-просов по теме.

#### Вопросы к контрольной работе:

- 104 Проблематика Бытия в философском учении М. Хайдеггера
- 105 Понимание Бытия в современном неотолизме
- 106 Бытие в качестве сознания в феноменологии Э.Гуссерля
- 107 Экзистенциальные модели бытия человека.
- 108 Бытие в фокусе постмодернистской мысли.

#### Дополнительные вопросы для обсуждения и самопроверки:

35. Изъясните проблематику Бытия в философском учении М. Хайдеггера.
36. Как понимается Бытие в современном неотолизме?
37. В чем особенность онтологического представления сознания в феноменологии Э.Гуссерля?
38. Укажите экзистенциальные модели бытия человека.
39. В чем суть проблемы Бытия в фокусе постмодернистской мысли?

#### Литература:

- / Губин В.Д. Основы философии: учеб. пособие / В.Д. Губин. — 4-е изд. М.: ФОРУМ : ИНФРА-М, 2019. — 288 с. <https://znanium.com/catalog/product/1009582>
- / Аналитика человеческого бытия: введение в опыт самопознания. Систематический очерк : монография / С.Е. Ячин. — М. : ИНФРА-М, 2019. — 210 с. <https://znanium.com/catalog/product/989370>
- / Философия : учебник / В.А. Канке. — М. : ИНФРА-М, 2019. — 291 с. <https://znanium.com/catalog/product/977818>
- / Социальная философия и философия истории : учебное пособие / под ред. д-ра филос. наук К.Х.Момджяна. — Москва : ИНФРА-М, 2019. — 478 с. <https://znanium.com/catalog/product/947030>
- / Философия : учеб. пособие / И.А. Карпенко. — Москва : ИНФРА-М, 2019. — 190 с. <https://znanium.com/catalog/product/1039315>



### Практическое занятие 8. Основы философской антропологии

Целью проводимого занятия является формирование знаний: многообразия рационального и иррационального в человеческой жизнедеятельности; сущности сознания, его взаимоотношения с бессознательным; роли сознания и самосознания в поведении; специфики мышления человека и природы творчества; понятий духовного бытия и духовной жизни человека и общества.

**Форма проведения:** контрольная работа по заранее известным вопросам. Обсуждение во-просов по теме.

#### Вопросы к контрольной работе:

- / Биологическое, духовное и социальное в человеке
- / Человек, индивид, личность.
- / Личность и массы.
- / Жизнь, смерть и бессмертие.
- / Природное и социальное в человеке.
- / Свобода и ответственность как условия существования личности.

#### Дополнительные вопросы для обсуждения и самопроверки:

- / Как понимать биологическое, духовное и социальное в человеке?
- / В чем разница в понятиях «человек», «индивид», «личность».
- / Объясните отношение личности и массы.
- / Как вы понимаете проблему жизни, смерти и бессмертия?
- / Как можно интерпретировать природное и социальное в человеке?
- / Как понимать отношение свободы и ответственности как условий существования личности?

#### Литература:

- / Губин В.Д. Основы философии: учеб. пособие / В.Д. Губин. — 4-е изд. М.: ФОРУМ : ИН-ФРА-М, 2019. — 288 с. <https://znanium.com/catalog/product/1009582>
- / Аналитика человеческого бытия: введение в опыт самопознания. Систематический очерк : монография / С.Е. Ячин. — М. : ИНФРА-М, 2019. — 210 с. <https://znanium.com/catalog/product/989370>
- / Философия : учебник / В.А. Канке. — М. : ИНФРА-М, 2019. — 291 с. <https://znanium.com/catalog/product/977818>
- / Социальная философия и философия истории : учебное пособие / под ред. д-ра филос. наук К.Х.Момджяна. — Москва : ИНФРА-М, 2019. — 478 с. <https://znanium.com/catalog/product/947030>
- / Философия : учеб. пособие / И.А. Карпенко. — Москва : ИНФРА-М, 2019. — 190 с. <https://znanium.com/catalog/product/1039315>

### Практическое занятие 9. Социальная философия. Философия глобальных проблем

Целью проводимого занятия является формирование знаний: понятия общества; единства и многообразия культур; отличия культуры от цивилизации; основных закономерностей социального историко-культурного развития человека и общества; сценариев будущего развития системы образования; роли науки в развитии цивилизации; соотношения науки и техники и связанные с ними глобальные проблемы современности; возможные «модели» человека и альтернативные подходы к пониманию его роли в обществе; содержания современных фило-



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

софских дискуссий по проблемам взаимодействия цивилизаций и сценариев будущего общественного развития.

**Форма проведения:** контрольная работа по заранее известным вопросам. Обсуждение вопросов по теме.

**Вопросы к контрольной работе:**

- В Проблема теоретической модели общества
- В Структура общества
- В Культура и цивилизация.
- В Понятия «общество» и «социальные отношения».
- В Система социальных отношений.
- В Проблема типологии исторического процесса (О.Шпенглер, К.Маркс, А.Тойнби).
- В Культура и цивилизация: многообразие их форм, особенности взаимодействия.
- В Понятие духовной жизни общества.
- В Закономерности формирования и функционирования духовной жизни общества
- В Современная общепланетарная цивилизация.
- В Комфорт как ценность техногенной цивилизации.
- В Информационное общество.
- В Глобальные проблемы.
- В Техника: ее специфика и закономерности развития.
- В Техника и технические науки.
- В Философия техники.
- В Понятие «информация». Информатизация: тенденции в изменении сознания, мышления, культуры.
- В Глобальные проблемы, причины их возникновения и основные признаки. Классификация глобальных проблем.

**Дополнительные вопросы для обсуждения и самопроверки:**

- \endash Какие известны проблемы теоретической модели общества?
- \endash Какова структура общества?
- \endash В чем разница между культурой и цивилизацией?
- \endash Что вам известно о проблеме типологии исторического процесса (О.Шпенглер, К.Маркс, А.Тойнби)?
- \endash В чем проявляет себя духовная жизнь общества?
- 48. Какие существуют закономерности формирования и функционирования духовной жизни общества?
- 49. Каковы черты современной цивилизации?
- 50. Какова ценность комфорта как ценности техногенной цивилизации?
- 51. Каковы особенности информационного общества?
- 52. Объясните основные глобальные проблемы.
- 53. В чем сущность техники, ее специфика и закономерности развития.
- 54. Техника и технические науки.
- 55. Изъясните проблемы философии техники?
- 56. Каково понятие «информация». Информатизация: тенденции в изменении сознания, мышления, культуры.

**Литература:**

- 52. Губин В.Д. Основы философии: учеб. пособие / В.Д. Губин. — 4-е изд. М.: ФОРУМ : ИН-ФРА-М, 2019. — 288 с. <https://znanium.com/catalog/product/1009582>



62. Аналитика человеческого бытия: введение в опыт самопознания. Систематический очерк : монография / С.Е. Ячин. — М. : ИНФРА-М, 2019. — 210 с.  
<https://znanium.com/catalog/product/989370>
63. Философия : учебник / В.А. Канке. — М. : ИНФРА-М, 2019. — 291 с. <https://znanium.com/catalog/product/977818>
64. Социальная философия и философия истории : учебное пособие / под ред. д-ра филос. наук К.Х.Момджяна. — Москва : ИНФРА-М, 2019. — 478 с.  
<https://znanium.com/catalog/product/947030>
65. Философия : учеб. пособие / И.А. Карпенко. — Москва : ИНФРА-М, 2019. — 190 с. <https://znanium.com/catalog/product/1039315>

## 2. Методические указания к практическим занятиям заочной формы обучения

### Практическое занятие 1. Философия как вид знания

Целью проводимого занятия является формирование знаний: предмета философии, ее места 55.роли в культуре; структуры философского знания; понятия мировоззрения, общее представление о мифологическом, философском, религиозном и научном мировоззрении; понятия философской рефлексии; умения анализировать мировоззренческие проблемы.

**Форма проведения:** контрольная работа по заранее известным вопросам. Обсуждение вопросов по теме.

#### Вопросы к контрольной работе:

- № Место и функции философии в культуре.
- № Мифология как исторический тип мировоззрения.
- № От мифологии к философии.
- № Специфика философского знания.
- № Основные темы философского познания.
- № Философия в понимании Платона и Аристотеля.
- № Гегель о значении философии.
- № Философия как строгая наука (Гуссерль).

#### Дополнительные вопросы для обсуждения и самопроверки:

- 67. Как возникла необходимость в философии?
- 68. В чем специфическая особенность философского мышления?
- 69. Какие особенности имеет философское мировоззрение?
- 70. Какие основные темы философского познания?
- 71. Как представлялась наука философия в понимании Платона и Аристотеля?
- 72. Что говорил Гегель о значении философии?
- 73. Что означает понимание философии как строгой науки в феноменологии Гуссерля?
- 74. Обладает ли философия своим языком? В чем состоит его особенность?
- 75. Каковы отличительные признаки философского текста?
- 76. В чем выражается значение философии в жизни человека?

#### Литература:

49. Губин В.Д. Основы философии: учеб. пособие / В.Д. Губин. — 4-е изд. М.: ФОРУМ : ИН-ФРА-М, 2019. — 288 с. <https://znanium.com/catalog/product/1009582>



47. Аналитика человеческого бытия: введение в опыт самопознания. Систематический очерк : монография / С.Е. Ячин. — М. : ИНФРА-М, 2019. — 210 с.  
<https://znanium.com/catalog/product/989370>
48. Философия : учебник / В.А. Канке. — М. : ИНФРА-М, 2019. — 291 с. <https://znanium.com/catalog/product/977818>
49. Социальная философия и философия истории : учебное пособие / под ред. д-ра филос. наук К.Х.Момджяна. — Москва : ИНФРА-М, 2019. — 478 с.  
<https://znanium.com/catalog/product/947030>
50. Философия : учеб. пособие / И.А. Карпенко. — Москва : ИНФРА-М, 2019. — 190 с. <https://znanium.com/catalog/product/1039315>

## Практическое занятие 2. Основы философской антропологии

Целью проводимого занятия является формирование знаний: многообразия рационального и иррационального в человеческой жизнедеятельности; сущности сознания, его взаимоотношения с бессознательным, роли сознания и самосознания в поведении; специфики мышления человека и природы творчества.

**Форма проведения:** контрольная работа по заранее известным вопросам. Обсуждение вопросов по теме.

### Вопросы к контрольной работе:

66. Биологическое, духовное и социальное в человеке
67. Человек, индивид, личность.
68. Личность и массы.
69. Жизнь, смерть и бессмертие.
70. Природное и социальное в человеке.
71. Свобода и ответственность как условия существования личности.

### Дополнительные вопросы для обсуждения и самопроверки:

- В Как понимать биологическое, духовное и социальное в человеке?
- В В чем разница в понятиях «человек», «индивид», «личность».
- В Объясните отношение личности и массы.
- В Как вы понимаете проблему жизни, смерти и бессмертия?
- В Как можно интерпретировать природное и социальное в человеке?
- В Как понимать отношение свободы и ответственности как условий существования личности?

### Литература:

- ООО Губин В.Д. Основы философии: учеб. пособие / В.Д. Губин. — 4-е изд. М.: ФОРУМ : ИНФРА-М, 2019. — 288 с. <https://znanium.com/catalog/product/1009582>
- ООО Аналитика человеческого бытия: введение в опыт самопознания. Систематический очерк : монография / С.Е. Ячин. — М. : ИНФРА-М, 2019. — 210 с.  
<https://znanium.com/catalog/product/989370>
- ООО Философия : учебник / В.А. Канке. — М. : ИНФРА-М, 2019. — 291 с. <https://znanium.com/catalog/product/977818>
- ООО Социальная философия и философия истории : учебное пособие / под ред. д-ра филос. наук К.Х.Момджяна. — Москва : ИНФРА-М, 2019. — 478 с.  
<https://znanium.com/catalog/product/947030>



26. Философия : учеб. пособие / И.А. Карпенко. — Москва : ИНФРА-М, 2019. — 190 с. <https://znanium.com/catalog/product/1039315>

### Практическое занятие 3. Познание, его возможности и границы. Знание и веры

Целью проводимого занятия является формирование знаний: многообразия форм исторического человеческого опыта и знания; соотношения истины и заблуждения, знания и веры; познавательных способностей человека; основных методов философского познания; понятия истины и концепций истины в истории философии и умения критически оценивать подходы и высказанные суждения в системе гуманитарных и социальных знаний.

**Форма проведения:** контрольная работа по заранее известным вопросам. Обсуждение во-просов по теме.

#### Вопросы к контрольной работе:

46. Понятие познания. Компоненты познания.
47. Роль и значение понимания в структуре познания.
48. Истина, оценка, ценность.
49. Знание и вера.
50. Основные этапы и формы процесса познания.
51. Истина и заблуждения.
52. Достоверность знания.
53. Критерии истинности.

#### Дополнительные вопросы для обсуждения и самопроверки:

44. Обоснуйте понятие познания. Какие есть компоненты познания?
45. В чем роль и значение понимания в структуре познания?
46. Каков смысл категорий «истина», «оценка», «ценность»?
47. Как понимать отношение знания и веры?
48. Каковы основные этапы и формы процесса познания?
49. Какие есть критерии истинности?

#### Литература:

- к Губин В.Д. Основы философии: учеб. пособие / В.Д. Губин. — 4-е изд. М.: ФОРУМ : ИН-ФРА-М, 2019. — 288 с. <https://znanium.com/catalog/product/1009582>
- к Аналитика человеческого бытия: введение в опыт самопознания. Систематический очерк : монография / С.Е. Ячин. — М. : ИНФРА-М, 2019. — 210 с. <https://znanium.com/catalog/product/989370>
- к Философия : учебник / В.А. Канке. — М. : ИНФРА-М, 2019. — 291 с. <https://znanium.com/catalog/product/977818>
- к Социальная философия и философия истории : учебное пособие / под ред. д-ра филос. наук К.Х.Момджяна. — Москва : ИНФРА-М, 2019. — 478 с. <https://znanium.com/catalog/product/947030>
- к Философия : учеб. пособие / И.А. Карпенко. — Москва : ИНФРА-М, 2019. — 190 с. <https://znanium.com/catalog/product/1039315>



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Курс по философии рассчитан на изучение в течение одного семестра, включает лекционные, практические занятия и самостоятельную работу студентов.

Для контроля уровня сформированности компетенций, качества знаний, умений и навыков особое внимание уделяется стимулированию самостоятельной работы студентов.

Содержание дисциплины «Философия» разделено на два модуля, по окончании изучения которых осуществляется текущий контроль усвоения учебного материала. В течение семестра проводятся модульно-рейтинговые мероприятия, в том числе, и в форме компьютерного тестирования для проверки самостоятельной работы студентов.

Изучение философии является трудоемкой задачей, в связи с чем, предусматривается обязательное посещение лекционных занятий для получения системы целостных знаний. Рекомендованные учебники и учебные пособия не всегда являются достаточными источниками знаний, так как, во-первых, учебный курс носит междисциплинарный характер, во-вторых, лекционный материал непосредственно дает квинтэссенцию ответов на экзаменационные вопросы.

После лекций проводятся практические занятия, они, прежде всего, предназначены для закрепления теоретических знаний и выработки таких практических навыков, как, аргументированное отстаивание собственного видения рассматриваемых проблем, владение приемами ведения дискуссии, полемики и диалога. Специфика курса заключается в том, что практически все темы требуют дискуссионного обсуждения. На практические занятия выносятся материал, дополняющий и углубляющий лекционный, а также, материал, не вошедший в лекционный курс, поэтому при подготовке к практическим занятиям необходимо самостоятельное прочтение и осмысление рекомендуемых работ.

Темы курса рассчитаны на закрепление и систематизацию лекционного материала, самостоятельное освоение дополнительной литературы, а также, нацелены на формирование навыков по практическому применению полученных знаний в сфере образования и общественной коммуникации.

Дисциплина «Философия» призвана сформировать мировоззренческие основы будущего специалиста, а также заложить методологические и духовно-нравственные установки для его формирования.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ПРОВЕДЕНИЯ ПРАКТИЧЕСКИХ  
ЗАНЯТИЙ И ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ**

**«Функциональное программирование»**

**Практическое занятие №1.**

Цель занятия: Приводить выражения к нормальной и СЗНФ в  
системе редукций  $\lambda$  - исчисления, отличать понятия чистого  
и расширенного

- исчисления.

**2 Представление выражений в  $\lambda$  - исчислении.**

Функциональные языки программирования появились в качестве средства для написания программ, не содержащих в явном виде понятий ячеек памяти для хранения значений (переменных) и последовательности вычислений как процесса изменения состояния памяти.

Основа для создания таких языков была предложена ещё в середине 1930-х гг. Алонзо Черчем и Стефаном Клини. Теория Черча, названная им  $\lambda$  -



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

исчисление, рассматривается в качестве теоретической основы и «минимального» функционального языка программирования. Любую программу, написанную на любом функциональном языке программирования, можно свести к формуле  $\lambda$  - исчисления.

Черч и Клини не создавали язык программирования. Они занимались формализацией понятия вычислимой функции и создания универсального математического аппарата для определения и работы с вычислимыми функциями. Черч сумел построить такую систему, при которой базовых функций нет, а вместо способов построения сложных функций из простых используются правила преобразований. В результате Черчу удалось создать исчисление чистых функций, в котором все математические понятия, такие как числа, арифметика и др., сводятся к понятию функции.

Процесс преобразования формул в лямбда – исчислении Черча (*редукции*) напоминает процесс вычисления функции, происходящий при исполнении программы.

Основным понятием в  $\lambda$  - исчислении является понятие выражения или формулы. В формулах переменные обычно обозначают аргументы функций, задаваемых  $\lambda$  - выражениями, однако сама по себе переменная является простейшим видом выражения.

$\therefore \epsilon \nu \delta \alpha \sigma \eta$  - выражение имеет вид  $(\lambda x . e)$ , где  $x$  – имя переменной, а  $e$  – выражение. Семантически такое выражение обозначает функцию с аргументом  $x$  и телом  $e$ . Применение функции записывается в виде  $(e1 e2)$ , где  $e1$  и  $e2$  – выражения ( $e1$  – функция, а  $e2$  – её аргумент).

Примеры:

$\lambda x . x$  – простейшая функция, выдающая свой аргумент; скобки опущены, поскольку это не вызывает неоднозначности;

и  $\lambda f . \lambda x . f x$  – функция с двумя аргументами, применяющая свой первый аргумент ко второму. Строго говоря, надо было бы расставить скобки, чтобы выражение приняло вид  $\lambda f . (\lambda x . (f x))$ . Однако принято соглашение, по которому «операция» применения функции к аргументу имеет более высокий приоритет, чем «операция» образования  $\lambda$  - выражения. Функции при этом применяются в порядке слева направо. Выражение  $f x u$  понимается как  $(f x) u$  – применение функции  $f$  к аргументу  $x$  и применение полученного результата к аргументу  $u$ . Образование же безымянной функции (операция «лямбда») наоборот применяется справа налево, так что выражение  $\lambda x . \lambda y . e$  понимается как  $\lambda x . (\lambda y . e)$ .

Следует учитывать, что при применении стандартных функций используется только префиксная запись операций, т. е. вместо привычного  $3+5$  записывается выражение  $+ 3 5$ . Естественно, бессмысленным будет выражение  $/ 3 0$ , поскольку результат деления числа 3 на число 0 не



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

определен. Бессмысленным будет выражение  $+ \text{TRUE } 0$ , поскольку невозможно выполнить сложение логического значения TRUE с числом 0. Условное выражение **if B then E1 else E2** может быть представлено применением стандартной функции IF с тремя аргументами B, E1, и E2. Кортежи представляются в виде применения стандартных функций кортежирования TUPLE – n, где n – произвольное натуральное число. Такая функция будет иметь n аргументов, и выдавать в результате работы кортеж, содержащий эти аргументы.

Обратная функция извлечения элемента кортежа по заданному номеру будет представлена стандартной функцией INDEX, аргументами которой будет номер элемента кортежа и сам этот кортеж.

Пример:

ф Если составить кортеж из чисел 2,4 и 10, а затем извлечь из него второй элемент, то должны в результате получить число 4. Соответствующая конструкция расширенного  $\lambda$  - исчисления будет выглядеть так:  
INDEX 1 (TUPLE- 3 2 4 10). Так как элементы кортежа нумеруются с нуля, так что второй элемент кортежа будет иметь номер 1.

41)  $\lambda$  - исчислении определена система *редукций*, т.е. набор правил преобразования выражений. С их помощью можно переходить от одних выражений к другим, эквивалентным им.

с Нормальная форма.

Вхождение переменной в некоторое выражение будет *связанным*, если оно находится внутри некоторого  $\lambda$  - выражения, в заголовке которого эта переменная упомянута в качестве аргумента. Если переменная не является связанной в некотором выражении, то она в нём будет *свободной*.

Рассматриваются четыре вида редукций в  $\lambda$  - исчислении. Первое называется переименованием переменных, или  $\alpha$ -преобразованием.

Его смысл состоит в том, что суть функции не меняется, если заменить имя её формального аргумента.

Пример:

29) выражении  $\lambda x . \lambda f . f x$  у можно заменить переменную x на переменную z.

Два следующих правила редукций – это «вычислительные» преобразования, которые приводят к упрощению выражения.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Преобразование  $\delta$ -редукция имеет следующий вид: пусть имеется выражение  $f \in e_1 e_2 \dots e_k$ , где  $f$  – константа, представляющее имя «встроенной» функции с  $k$  аргументами, а  $e_1, e_2, \dots, e_k$  – значения, могущие служить аргументами этой функции. Тогда такое выражение можно заменить на эквивалентное ему выражение.

Пример:

Если константа  $+$  представляет функцию арифметического сложения целых, а константа  $OR$  – функцию логического «ИЛИ», то в результате  $\delta$ -редукции выражение  $+ 1 4$  может быть преобразовано к выражению  $5$ .

Преобразование  $\beta$  - редукция соответствует применению функции, представленной  $\lambda$  - выражением, к аргументу.

Пример:

Выражение  $((\lambda x . + x x) 3)$  в результате применения  $\beta$  - редукции будет преобразовано в  $(+ 3 3)$ , которое, в свою очередь, может быть преобразовано в  $6$  с помощью применения  $\delta$ -редукции.

Если некоторое выражение содержит в себе подвыражение, к которому можно применить одну из редукций, то такое подвыражение называется *редуцируемым* или сокращенно *редексом*. Таким образом процесс

преобразования выражения сводится к применению  $\beta$  и  $\delta$ -редукций к *редексам*, содержащимся в исходном выражении.

18. - преобразование выражает тот факт, что две функции, которые при применении к одному и тому же аргументу дают один и тот же результат, эквивалентны.

Основное назначение  $\lambda$  - исчисления состоит в том, чтобы показать, что любая вычислимая функция может быть представлена в виде  $\lambda$  - выражения.

Пример:

Функция  $(\lambda x . * x x)$  представляет собой функцию возведения в квадрат. Функция действительно выдаст результат  $36$ , если применить её к аргументу  $6$ . Для этого составим выражение  $((\lambda x . * x x) 6)$  и выполним те редукции, которые можно применить к этому выражению. Сначала в результате применения  $\beta$  - редукции получится выражение  $(* 6 6)$  к которому можно применить  $\delta$ -редукцию и окончательно получить значение  $36$ .

Редукции можно применять до тех пор, пока в выражении имеется хотя бы один редекс. Если ни одного редекса в выражении нет, то говорят, что выражение находится в *нормальной форме*.



к Слабая заголовочная нормальная форма (СЗНФ).

45. результате последовательного применения  $\beta$  и  $\delta$ -редукций можно не получить нормальной формы выражения, а получим довольно близкий её эквивалент – так называемую слабую заголовочную нормальную форму (СЗНФ).

Выражение находится в СЗНФ, если оно имеет один из следующих видов:

14. константа  $c$  (в том числе – одна из встроенных функций);
15. переменная  $x$ ;
16.  $\lambda$  - выражение  $\lambda x . e$ , где  $e$  – выражение, находящееся в нормальной форме;
17.  $f e_1 e_2 \dots e_k$ , где  $f$  – встроенная функция с  $n$  аргументами,  $e_1, e_2, \dots, e_k$  – произвольные выражения, причём  $k < n$ .

Если записать последнее выражение в эквивалентном виде  $\lambda$  - выражения, то можно видеть, что нормальная форма отличается от СЗНФ



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

только тем, что в СЗНФ не производится приведение к нормальной форме выражений, находящихся внутри тела  $\lambda$  - выражения.

Контрольные вопросы к выполнению практического занятия

Практическое занятие №.1 (ОПК-2, ПК-2)

- В Кем и когда была предложена математическая основа для создания функциональных языков программирования.
- В Какую систему в  $\lambda$ (лямбда) –исчислении построил Алонзо Черч.
- В Основное понятие в  $\lambda$ (лямбда) – исчисления Черча.
- В Представление выражений в  $\lambda$ (лямбда)-исчислении.
- В Место и роль различных *редукций* в  $\lambda$ (лямбда)-исчислении .
- В Преобразование исходного выражения к его нормальной форме.
- В Слабая заголовочная нормальная форма выражения (СЗНФ).
- В Рекурсия в  $\lambda$ (лямбда)-исчислении.
- В Чистое  $\lambda$ (лямбда)-исчисление.

Задание №1: Определить, какие из переменных в заданном выражении  $\lambda$  - исчисления связаны, а какие свободны?

$X (\lambda x. \lambda y. y (\lambda z. x) z) (\lambda x. y);$

Задание №2:

На языке Haskell даны определения следующих двух взаимно – рекурсивных функций:

$f\ x = x : g\ (2*x)$

$g\ y = y : f\ (y+1)$

Составить выражения для этих функций в расширенном  $\lambda$ (лямбда) –исчислении.

Выполнение заданий практического занятия оформляется в виде отчёта с ответами на контрольные вопросы и с пояснениями выполнения заданий.



Практическое занятие №2 (ОПК-2, ПК-2).

Цель занятия: Решить задачи на языке Java или Паскаль в императивном стиле и с использованием функционального стиля.

Функциональное программирование – это ветвь программирования, при котором программирование ведётся с помощью определения функций.

Алгоритмы, записанные с помощью языков функционального программирования, во- первых, допускают сравнительно простой анализ и формальное преобразование программ. Во – вторых, отдельные части программ могут исполняться независимо друг от друга. Такими замечательными свойствами и обладают языки функционального программирования.

Описание алгоритмов в функциональном стиле сосредоточено не на том, как достичь нужного результата ( в какой последовательности выполнять шаги алгоритма), а больше на том, что должен представлять собой этот результат.

Отличительными особенностями функций, определяемых любым функциональным языком программирования, являются следующие:

- е каждая функция в программе выдаёт один и тот же результат на одном и том же наборе входных данных (аргументов функции), т.е. результат работы функции является «повторяемым»;
- е вычисление функции не может повлиять на результат работы других функций, т.е. функции являются «чистыми». Если программа представляет собой набор чистых детерминированных функций , то она будет функциональной независимо от того, написана ли она на специальном языке функционального программирования Haskell или на традиционном Java.

Пример: Функция вычисления суммы элементов числового списка.

```
double sumList (List<Double> list) {
 double sum = 0;
 for (Double element : list) {
 sum += element;
 }
 return sum;
}
```

Эта функция детерминированная и «чистая», она выдаёт всегда один и тот же результат на одинаковых входных данных и не влияет на поведение других функций.

Однако всё же с точки зрения функционального стиля имеет одну



«неправильность». В функции определяется и используется локальная переменная *sum*, которая нужна для запоминания промежуточных значений суммы.

В чисто функциональных программах присваиваний вообще нет, как нет и понятия последовательного вычисления. Каждая функция должна представлять собой суперпозицию обращений к другим функциям.

Представим приведенную программу в функциональном стиле. **double** sumList (List<Double> list) {

**final int** size = list. size ()

; **final int** mid = size / 2;

**return**

size == 0 ? list . get (0) :

sumList (list. subList (0, mid ) ) +

sumList (list. subList (mid, size ) ) ;

Вместо цикла использована рекурсия, а вместо условного оператора в этой программе применяются условные выражения, которые соединяют условиями не отдельные части последовательно выполняющейся программы, а отдельные подвыражения. Это является характерной особенностью функционального стиля программирования.

Контрольные вопросы к выполнению практического занятия

- с Отличительные особенности функций, определяемых функциональным языком программирования.
- с Особенности функционального стиля программирования.
- с Недостатки и преимущества функционального стиля по отношению к императивному стилю программирования.
- с «Чистые» и «нечистые» функции.
- с Что представляет собой каждая функция в функциональном программировании.
- с Что используется вместо цикла в функциональной программе.

Задание: Составить программы решения задач на языке Java в привычном императивном стиле и в функциональном стиле программирования

1. Приближенное вычисление числа *e* по формуле для



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

разложения  $e^x$  в ряд Тейлора.

- Вычисление приближённого значения корня уравнения  $\cos x = x$  методом бисекции.

Выполнение заданий практического занятия оформляется в виде отчёта с ответами на контрольные вопросы и с пояснениями выполненных заданий.

Лабораторная работа №1. (ОПК-2, ПК-2)

Цель работы: установка системы Haskell Platform.

Для того чтобы начать работу с функциональным языком программирования, нужно загрузить на компьютер систему программирования на языке Haskell, в состав которой входят, в частности, компилятор и интерпретатор программ, написанных на этом языке, а также простая оболочка для операционной системы Microsoft Windows. Систему Haskell Platform можно установить с сайта <https://www.haskell.org>, перейдя по ссылке <https://www.haskell.org/platform/>.

После установки системы можно сразу же запустить интерпретатор выражений, записанных на языке Haskell, GHCi (интерпретатор командной строки) или (в системе Windows) WinGHCi.GHC (Glasgow Haskell Compiler) – это популярный компилятор программ на языке Haskell.

Все примеры работы интерпретатора будут приводиться непосредственно в 18. тексте. Если в интерпретаторе набрать выражение  $2*2$ , то он выведет результат вычисления – 4. В тексте это будет показано следующим образом:

```
2*2
```

```
4
```

Команды, подаваемые интерпретатору, - это выражения для вычисления или собственно команды, с помощью которых можно управлять работой интерпретатора, а также получать всевозможную информацию о текущем контексте вычислений, получать отладочную информацию, устанавливать режимы работы, загружать программы и программные модули и т.д.

Полный список команд можно получить, набрав команду: “ : ? “

Компилятор GHC умеет вычислять тип поданного ему выражения с помощью команды **: type**, которую можно задавать в сокращенном виде - **: t**:

Посмотреть на конкретный тип вычисленного выражения можно, включив опцию компилятора с помощью команды “ : set + t”. После



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

включения этой опции интерпретатор после каждого вычисления будет показывать тип вычисленного значения

Задания к выполнению лабораторной работы.

46. Система программирования Haskell Platform:
47. Установка системы программирования Haskell Platform;
48. Запуск интерпретатора выражений на языке Haskell – GHCi (интерпретатор командной строки)
49. Получение и просмотр полного списка команд;
50. Установление различных режимов работы интерпретатора.

Выполнение заданий лабораторной работы оформляется в виде отчёта с ответами на контрольные вопросы и с пояснениями выполненных заданий.

Лабораторная работа №2. (ОПК-2, ПК-2)

Цель работы: Подготовить, загрузить и исполнить программы на языке Haskell с помощью установленного интерпретатора.

Программа на языке Haskell – это набор определений значений и функций. Загрузив программу в интерпретатор, можно затем выполнять действия по вычислению результатов, просто вызывая функции, определения которых заданы в программе. Для примера, чтобы показать, каким образом можно подготовить, загрузить и исполнить программу на языке Haskell с помощью установленного интерпретатора, определим функцию с тремя аргументами, которая выдаст результат вычислений по известной формуле Герона.

Определим в программе функцию *triangle* с аргументами *a*, *b* и *c*, которая сначала введёт обозначение *p* для значения полупериметра треугольника, а затем вычислит его площадь, используя как вычисленное значение полупериметра, так и аргументы – длины сторон треугольника.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

`triangle a b c = let p = (a + b + c) / 2 in sqrt (p * (p-a) * (p-b) * (p-c))`

Запишем программу в текстовый файл с именем `triangle.hs` (расширение имени файла `.hs` используется для текстов программ на языке Haskell).

Пусть, например, полное имя файла в системе Windows будет `C:\Haskell\triangle.hs`

Теперь запустим интерпретатор и подадим команду на загрузку и проверку программы:

```
>> : load C:\Haskell\triangle.hs
```

Если имя файла записано правильно и не содержит пробелов и букв, отличных от букв латинского алфавита, то программа будет загружена, и интерпретатор проверит правильность синтаксиса. Если все правильно, то на экране увидим результат работы интерпретатора.

Теперь можно попробовать вычислить площадь какого-либо треугольника, записав выражение:

```
>> triangle 3 4 5
```

что приведёт к запуску функции и выдаче ожидаемого результата `6.0`.

Задания к выполнению лабораторной работы.

- в Составление простой программы (например, вычисление площади треугольника по формуле Герона).
- в Запись программы в текстовый файл с именем `triangle.hs` (полное имя файла в системе Windows будет `C:\Haskell\triangle.hs`).
- В Запуск интерпретатора и подача команды на загрузку и проверку программы.
- В Загрузка выражения для применения функции.
- В Проверить правильность работы функции (программы) при нескольких вычислениях.
- В Для подготовки программ изучить следующие вопросы:

элементарные типы данные Haskell;

Задания на выполнение лабораторной работы:

- с подготовить и исполнить программу вычисления площади треугольника при различных аргументах;



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

с подготовить программу нахождения чисел Фибоначчи.

Выполнение заданий лабораторной работы оформляется в виде отчёта с ответами на контрольные вопросы и с пояснениями выполненных заданий.  
Практическое занятие №3. (ОПК-2, ПК-2)

Цель занятия: Изучение типов и классов в языке программирования Haskell.

Haskell – строго типизированный язык. Такими же свойствами обладают большинство современных языков программирования, такие как Паскаль, Java, C++ и многие другие. Лисп – язык без строгой типизации.

Основу системы типов Haskell составляют типы: целые (Integer и Int); вещественные (Float и Double); логические (Bool); символьные (Char). Все идентификаторы в Haskell чувствительны к регистру букв, так что integer, Integer, INTEGER – это три разных идентификатора.

Регистр первой буквы идентификатора определяет встроенный (заглавная буква) или определённый программистом тип или класс. С заглавных букв начинаются также имена модулей и пакетов. Идентификаторы объектов – значения простых и сложных типов, в том числе функций – начинаются со строчной буквы или символа подчёркивания. Идентификаторы строятся из букв, подчёркиваний и цифр и символа апострофа. Ни апостроф, ни цифра не могут быть первым символом идентификатора.

и программе можно вводить собственные идентификаторы для имеющихся значений.

Классы в языке Haskell имеют примерно тот же смысл, что и других языках программирования: они определяют набор операций (функций), которые можно производить над объектами разных типов, принадлежащих этому классу. Например, полный список разрешенных операций для класса *Num* можно получить, набрав команду : *info*, в результате выполнения которой получим примерно следующий результат:

```
x : info Num
Class Num a where
(+) :: a -> a -> a
(*) :: a -> a -> a
(-) :: a -> a -> a
negate :: a -> a
abs :: a -> a
```



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

signum :: a -> a

fromInteger :: Integer -> a

в данном описании указаны все функции и операции, разрешённые для типов из этого класса, так что, если выполнить операцию сложения над двумя значениями, каждое из которых имеет тип, принадлежащий классу Num, то в результате получим значение того же типа, что и типы операндов.

Задания контрольные вопросы к выполнению заданий практического занятия.

Различие между типами и классами в языке программирования Haskell.

Особенности базовых типов Haskell, непривычных по другим языкам программирования.

Объединение произвольных типов в кортежи.

Особенности языка Haskell касающиеся изображения чисел.

Операции и стандартные функции в языке программирования Haskell.

Две особенности записи выражений.

Объединение произвольных типов в кортежи.

Пустой кортеж

Выполнение заданий практического занятия оформляется в виде отчёта с ответами на контрольные вопросы и с пояснениями выполнения заданий.

Лабораторная работа №3(ОПК-2, ПК-2)

Цель работы: Определение функций с помощью уравнений. Подготовить, загрузить и исполнить программы на языке Haskell с помощью установленного интерпретатора.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Для того, чтобы написать сколь-нибудь полезную программу, необходимо помимо новых идентификаторов и правильного написания выражений уметь определять новые функции.

Каждая функция в Haskell – это тоже некоторое значение, для которого определён тип. В типе функции указывают типы её аргументов и тип значения функции, при этом типы аргументов и значения функции отделяются друг от друга символами “ $\rightarrow$ ”. Так что, например, тип функции одного вещественного аргумента с вещественным результатом (такой, как, например, функция вычисления синуса) можно записать в виде

`Double  $\rightarrow$  Double` ,

31 тип функции с двумя целыми аргументами и одним логическим результатом (такой, как, например, операция сравнения двух целых значений по величине) – в виде `Int  $\rightarrow$  Int  $\rightarrow$  Bool`.

Саму функцию можно определить с помощью «уравнения», в котором выясняется, как функция должна себя вести, если ей задать значение аргумента. Например, определим функцию удвоения, которая удваивает значение своего вещественного аргумента и выдаёт получившееся значение. Для этого задаётся идентификатор функции *twice*, задаётся её тип и записывается уравнение, в котором показывается, что вызов этой функции с заданным значением аргумента эквивалентен выражению, в котором это значение умножается на 2:

`twice :: Double  $\rightarrow$  Double`

`twice x = 2 * x`

Теперь можно загрузить и скомпилировать функцию для удвоения вещественных чисел.

Если функция *twice* определена, то вычисление выражения `twice 5.5` можно представить следующим образом. Сначала происходит сопоставление фактического значения аргумента 5.5 с формальным аргументом *x*. Затем вызов функции заменяется правой частью уравнения, в которой вместо формального аргумента используется сопоставленное с ним значение фактического аргумента. Таким образом, после сопоставления и замены вместо выражения `twice 5.5` получаем выражение `2*5.5`, которое после вычисления даёт значение 11. Процесс преобразования выражения можно записать следующим образом:

`twice 5.5  $\rightarrow$  2 * 5.5  $\rightarrow$  11`

Преобразование выражений, подобное приведенному, называют *редукциями*.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Таким образом, вычисление выражений в Haskell (исполнение программы) осуществляется с помощью последовательных редукций исходного выражения.

27.       целом процесс вычислений можно представить следующим образом. Сначала имеется некоторое исходное выражение. В этом выражении выбирается некоторый вызов функции( более точно – применение функции к аргументам). Затем рассматривается уравнение, определяющее эту функцию, и осуществляется сопоставление аргумента с формальным параметром функции. После этого правая часть уравнения, в которой все формальные параметры заменены на фактический аргумент, подставляется на место вызова. Так продолжается до тех пор, пока в выражении больше не останется применений функций к аргументам. Тогда говорят, что исходное выражение приведено к нормальной форме.

Задание к выполнению лабораторной работы.

Составить программу вычисления факториала натурального числа:

- 33.определить значение функции при задании отрицательного значения аргумента;
- 34.при выдаче интерпретатором сообщения об аварийном завершении работы для определения функциизаписатьнесколькоуравнений, определяющих поведение функции при различных значениях аргумента;
- 35.определить ту же функцию, задав аргументу вызова нулевое значение;
- 36.проверить правильность работы функции при нескольких вычислениях.

Выполнение заданий лабораторной работы оформляется в виде отчёта с ответами на контрольные вопросы и с пояснениями выполненных заданий.

Практическое занятие №4. (ОПК-2, ПК-2)

Цель занятия: Рекурсивное определение функций в языке программирования Haskell.

Функции «встроенные» или «примитивные», работа которых не определяется уравнениями, «встроены» в язык. Например, примитивными



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

функциями являются все арифметические операции. Редукция выражений, содержащих примитивные операции, осуществляются за один шаг, без сопоставления и подстановки.

Определение функции может быть рекурсивным, т.е. в правой части уравнения может быть вызов определяемой функции. В этом случае в процессе преобразования выражения, содержащего вызов рекурсивной функции, может получиться выражение, также содержащее вызов той же самой функции. Для того чтобы процесс вычисления мог закончиться, необходимо использовать условные выражения, которые приводят к выбору одной из двух альтернатив при вычислении сложных выражений. Условное выражение имеет вид:

**if** <условие> **then** <выражение – «то»> **else** <выражение – «иначе»>

Зададим определение простой рекурсивной функции, предназначенной для вычисления факториала заданного целого числа. Определение этой функции может выглядеть следующим образом:

```
factorial : : Integer -> Integer
factorial n = if n == 0 then 1 else n * factorial (n-1)
```

Вместо символа  $\rightarrow$ , который использовался для того, чтобы показать один шаг редукции, можно просто записывать результаты последовательных редукций в последовательных строках текста.

```
Factorial 3
if 3 == 0 then 1 else 3 * factorial 2
if False then 1 else 3 * factorial 2
3 * factorial 2
3 * if 2 == 0 then 1 else 2 * factorial 1
3 * 2 * factorial 1
3 * 2 * if 1 == 0 then 1 else 1 * factorial 0
3 * 2 * 1 * factorial 0
3 * 2 * 1 * if 0 == 0 then 1 else 0 * factorial (n-1)
3 * 2 * 1 * 1
6
```

Вместо применения условного выражения можно использовать более наглядную запись уравнения с «охраняющими условиями» (guards).

Контрольные вопросы и задания.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

76. Редукция выражений, работа которых не определяется уравнениями, содержащих примитивные встроенные операции (арифметические операции).
77. **Рекурсивное** определение функции.
78. Вычисление условного выражения.
79. Использование записи уравнения с «охраняющими условиями» вместо применения условного выражения
80. Процесс преобразования выражения при вычислении факториала числа.

Выполнение заданий практического занятия оформляется в виде отчёта с ответами на контрольные вопросы и с пояснениями выполнения заданий

Лабораторная работа №4 (ОПК-2, ПК-2)

Цель работы: Техника работы со списками.

- традиционных языках программирования для построения объектов, более сложных, чем элементарные числа или логические значения, используют различные встроенные механизмы для построения сложных объектов из более простых.

- языке Haskell имеется способ, позволяющий строить структуры данных произвольной степени сложности из более простых объектов – списки. Списки напоминают и списковые структуры традиционных языков программирования, и обычные массивы.

Список - это последовательность объектов одного и того же типа, состоящая из произвольного числа объектов (возможно, ни одного - список называется пустым). Тип списка определяется типом его компонент и обозначается [T]. Таким образом, [Integer] – тип компонент целые числа, [[Char], Double]-компонентами являются кортежи из двух полей, причём первым полем в каждом кортеже будет список символов, а вторым полем вещественное число. Сами объекты списки задаются перечислением своих компонент в квадратных скобках через запятую.

Список, состоящий из символов, можно записать в виде строки, заключив составляющие его символы в двойные кавычки. Например, строка «Haskell» обозначает в программе список из семи символьных значений. Для списков символов можно использовать идентификатор типа **String** вместо



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

[Char].

Второе средство построения списков - это задание списка чисел, образующих арифметическую прогрессию. Когда шаг прогрессии равен единице, то указывается только первый и последний элементы списка [1..10]. Если шаг прогрессии не равен единицы, то указываются два первых элемента и последний элемент. [3, 5..11].

Для того, чтобы писать программы обработки списков, нужно научиться извлекать отдельные элементы списка. Это можно делать с помощью встроенных операций, позволяющих извлечь отдельные составные части списка, либо посредством аппарата сопоставления с образцом.

Важнейшие встроенные операции обработки списков:

- head, last – функции, которые по заданному аргументу-списку выдают его первый и последний элементы соответственно;
- tail – функция, выдающая остаток списка, полученный отбрасыванием первого элемента;
- !! – операция, которая по списку и заданному номеру элемента выдаёт соответствующий элемент списка. Список должен быть непустым, а номер должен лежать в пределах от нуля (первый элемент списка имеет номер ноль) до количества элементов списка без единицы. Например вычисление конструкции [1, 2, 3, 4] !! 2 приведёт к выдаче в качестве результата числа 3 – элемента списка, имеющего номер 2;
- null – функция проверки пустоты списка. Если в качестве аргумента этой операции будет задан пустой список, то функция выдаст значение True, в противном случае – False;
- length – функция, вычисляющая количество элементов в списке;
- ++ - операция соединения двух списков.

Задания к выполнению лабораторной работы:

и Определить функцию (составить программу), которая по заданным натуральным числам определяет их наибольший общий делитель согласно алгоритму Евклида.

2. Вызвать стандартную функцию *g c d* для проверки результата.

ее Определить функцию для проверки того, является ли заданное натуральное число простым:

– кроме основной функции *p r i m e* применить вспомогательную функцию *p r i m e'*, так как рекурсию по основному аргументу функции (проверяемому числу) организовать трудно.;



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

- использовать идентификатор *otherwise* в качестве условия при написании уравнения для функции *prime*’;
  - проверить правильность работы функции при нескольких вычислениях.
- ее     Подготовить программу суммирования элементов целочисленного списка.

Выполнение заданий лабораторной работы оформляется в виде отчёта с ответами на контрольные вопросы и с пояснениями выполненных заданий.

### Практическое занятие №5 (ОПК-2, ПК-2)

Цель занятия: Уметь пользоваться стандартными функциями высших порядков для массовой обработки данных.

В языке Haskell имеется большое количество стандартных функций, которые имеют функциональные аргументы и результаты. Рассмотрим стандартную функцию обработки списков **map** (отображение), которая, получив в качестве аргументов список и функцию преобразования элементов этого списка, выдаёт в качестве результата список из преобразованных элементов, полученных применением функционального параметра к каждому элементу списка. Если, например, функция *sqr* возводит свой аргумент в квадрат, то с помощью функции **map** можно, имея список целых чисел, получить список их квадратов: `map sqr [1, 2, 5, -2] → [1, 4, 25, 4]`.

Функция, подобная **map**, которая в качестве аргумента получает другую функцию или выдаёт другую функцию в качестве результата, называется *функцией высшего порядка*, или *функционалом*.

стандартной библиотеке языка Haskell имеются две функции высшего порядка, одна из которых применяет заданную бинарную операцию к элементам списка от начала списка к его концу, а вторая начинает обработку элементов с конца списка, применяя заданную в качестве аргумента операцию последовательно, двигаясь по направлению к началу списка. Первая из этих операций называется *foldl*, вторая – *foldr*. Общее название для этих операций - операции свёртки списка.

Мощным средством обработки списков является сочетание функций отображения и свёртки. Их последовательное применение позволяет сначала подготовить элементы списка к последовательной обработке, причём эта подготовка будет производиться параллельно, а затем уже будет получен окончательный результат.



Например, пусть имеется таблица, в которой некоторым символам приписан номер (класс символа). К различным классам отнесём гласные буквы, согласные буквы, цифры, знаки препинания и все прочие символы. Таблица должна представлять собой список строк, в которой в каждой строке собраны символы одного класса. Программа сбора статистики по тексту представлена следующим образом.

```
43. Таблица классов
символов classTable :: [
 [Char]]
classTable = ["aeiouy", "bcdfghjklmnpqrstvwxyz", "0123456789", "
(),,:;?!_"]
```



Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»

39) Функция определения класса

символа `getClass :: Char -> [[Char]] ->`

`Int getClass c [] = 0`

`getClass c (s: rest) | c `elem` s = 1 | restClass == 0 = 0 |  
otherwise = restClass + 1`

Where `restClass = getClass c rest`

.....

Контрольные вопросы и задания.

В Понятия функций высшего порядка .

В Понятие **λ-выражений** и методы их  
использования в функциях высших порядков.

В Объединение нескольких уравнений в одно  
выражение с помощью специальной  
конструкции для выбора по образцам.

4. Операции свёртки списка *foldl* и *foldr*.

53. Уравнения функций свёртки списков.

54. Рекурсивные вызовы внутри функции *foldl*.

55. Обработка списков с помощью функций  
высших порядков.

56. Дописать программу статистики символов по  
заданному тексту

Выполнение заданий практического занятия оформляется в виде отчёта с ответами на контрольные вопросы и с пояснениями выполнения заданий.

Лабораторная работа №5 (ОПК-2, ПК-2)

Цель работы: Использовать навыки применения функций высших порядков в программировании на языке Haskell.

Конструктор для формирования списков имеет обозначение – двоеточие. Конструкторы и функции имеют много общих черт. И конструктор, и функция служат для образования новых значений из



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

имеющихся. Разница состоит в том, что функция для образования нового значения требуются правила, с помощью которых это значение и образуется, 109 конструктору никаких правил не нужно – аргументы собираются конструктором в единый объект нового типа механически.

качестве простого примера функции обработки двоичных деревьев рассмотрим функцию вычисления высоты дерева:

40. функция для определения высоты заданного двоичного дерева

`Height :: Tree a -> Int`

`height Null`

`height (Tree _ t1 tr) = 1 + max (height t1) (height tr)`

Что касается свёртки, то для дерева можно определить много вариантов прохода по его узлам, соответственно, можно будет несколько вариантов функции свёртки дерева согласно алгоритмам его прохождения. Имея такую функцию, можно легко вычислить сумму значений узлов, хранящихся в дереве, или определить максимальное значение узла для непустого дерева:

`tsum :: Num a => Tree a -> a`

`tsum t = tfoldr (+) 0 t`

`tmax :: Ord a => Tree a -> a`

`tmax t@ (Tree root _ _) = tfoldr max root t`

Контрольные вопросы и задания для выполнения лабораторной работы

- / Составить матрицы смежности для заданного графа.
- / Представить обход графа в ширину, построив дерево обхода.
- / Программировать обход графа.
- / Составить функцию (программу) построения пути между двумя заданными вершинами в графе.
- / Проверить правильность работы функции (программы)

## **Цифровая обработка сигналов**



*Исследование спектров ДПФ различных сигналов.*

Задача 1:

Определить дискретный спектр периодического дискретного сигнала  $x(n) = (0, 1, 1, 0)$ .

$$c_k = \frac{1}{4} \sum_{n=0}^3 x(n) e^{-j \frac{\pi}{2} kn}, \quad k = 0, 1, \dots, 3.$$

Задача 2:

Определить дискретный спектр периодического дискретного сигнала  $x(n) = (1, 1, 0, 0, 1, 1)$ .

$$c_k = \frac{1}{6} \sum_{n=0}^5 x(n) e^{-j \frac{\pi}{2} kn}, \quad k = 0, 1, \dots, 5.$$



ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №2

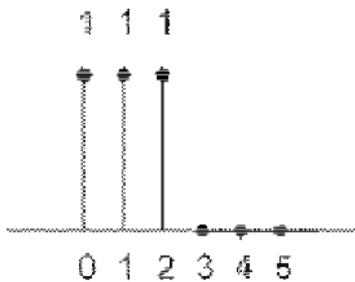
**Численные методы осуществления преобразования Фурье**

Задача 1:

Дискретный сигнал периодически задан шестью равноотстоящими отсчётами

$$\{x_k\} = \{1, 1, 1, 0, 0, 0\}$$

Найти коэффициенты ДПФ этого сигнала



Задача 2: Вычислите дискретное преобразование Фурье (ДПФ) сигнала  $x(n) = (2, 2, 2, 2, 0, 0, 0)$ .

Задача 3: Вычислите дискретное преобразование Фурье (ДПФ) сигнала  $x(n) = (1, 1, 1, 0)$ .

Задача 4: Рассчитать линейную и круговую свертки сигналов  $(1, 2, 1, 2)$  и  $(1, 2, 3, 4)$  непосредственным вычислением и с помощью алгоритма цифровой фильтрации в частотной области.



ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №3

**Исследование нерекурсивных цифровых фильтров 1-го и 2-го порядков**

Задача 1. На вход нерекурсивного цифрового фильтра (НЦФ) с импульсной характеристикой (ИХ)  $\{g(nT)\} = (-1,2; 2; 0,5)$  подается сигнал  $\{x(kT)\} = (0,25; 1; 0,75)$ .

25. Найти системную функцию  $H(z)$  фильтра.
26. Записать алгоритм цифровой фильтрации.
27. Изобразить схему фильтра.

Задача 2: На вход нерекурсивного цифрового фильтра (НЦФ) с импульсной характеристикой (ИХ)  $\{g(nT)\} = (-1,8; 2; 0,2)$  подается сигнал  $\{x(kT)\} = (0,05; 1; 0,95)$ .

14. Найти системную функцию  $H(z)$  фильтра.

Системная функция  $H(z)$  фильтра определяется формулой:

$$H(z) = b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + \dots + b_mz^{-m},$$

где  $b_i$  – весовые коэффициенты данного НЦФ.

$$H(z) = -1,8 + 2z^{-1} + 0,2z^{-2}.$$

15. Записать алгоритм цифровой фильтрации.

Алгоритм цифровой фильтрации представляет собой дискретную свёртку двух сигналов, один из которых импульсная характеристика, а другой входной дискретный сигнал:



=

-

3. Изобразить схему фильтра.

#### ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №4

#### Исследование рекурсивных цифровых фильтров 1-го и 2-го порядков

Задача 1. На вход рекурсивного цифрового фильтра (РЦФ) с системой функций  $H(z)$  подается сигнал  $\{x(kT)\} = (0,1; 1; 0,9)$ .

$a_0$	$a_1$	$a_2$	$b_1$	$b_2$
0,2	1	0,2	0,6	-0,08

#### РЕШЕНИЕ

Записать алгоритм цифровой фильтрации.

23. помощью системной функции запишем алгоритм цифровой фильтрации РЦФ:

$$H(z) = \frac{0,2 + z^{-1} + 0,2z^{-2}}{1}$$

$$y(nT) = a_0x(nT) + a_1x(nT-T) + \dots + b_1y(nT-T) + \dots + b_{L-1}y(nT-(L-1)T),$$

$$y(nT) = 0,2x(nT) + x(nT-T) + 0,2x(nT-2T) + 0,6y(nT-T) - 0,08y(nT-2T).$$

12. Изобразить схему фильтра в прямой и канонической формах.



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Прямая форма реализации структурной схемы РЦФ изображена на рис.2. Необходимо заметить, что существуют и другие структурные схемы РЦФ. В контрольной работе требуется представление канонической схемы реализации предложенного РЦФ 2-го порядка. Её преимущество – это минимальное число элементов задержки, равное порядку фильтра. Каноническая структура представлена на рисунке 3.

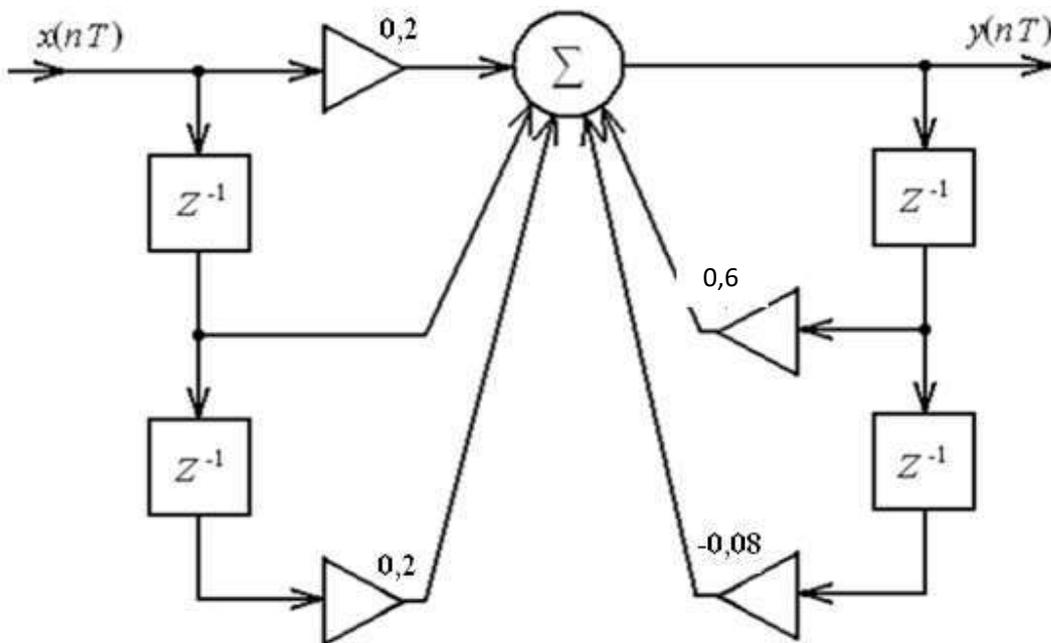


Рисунок 2 Структура ЦФ прямая.

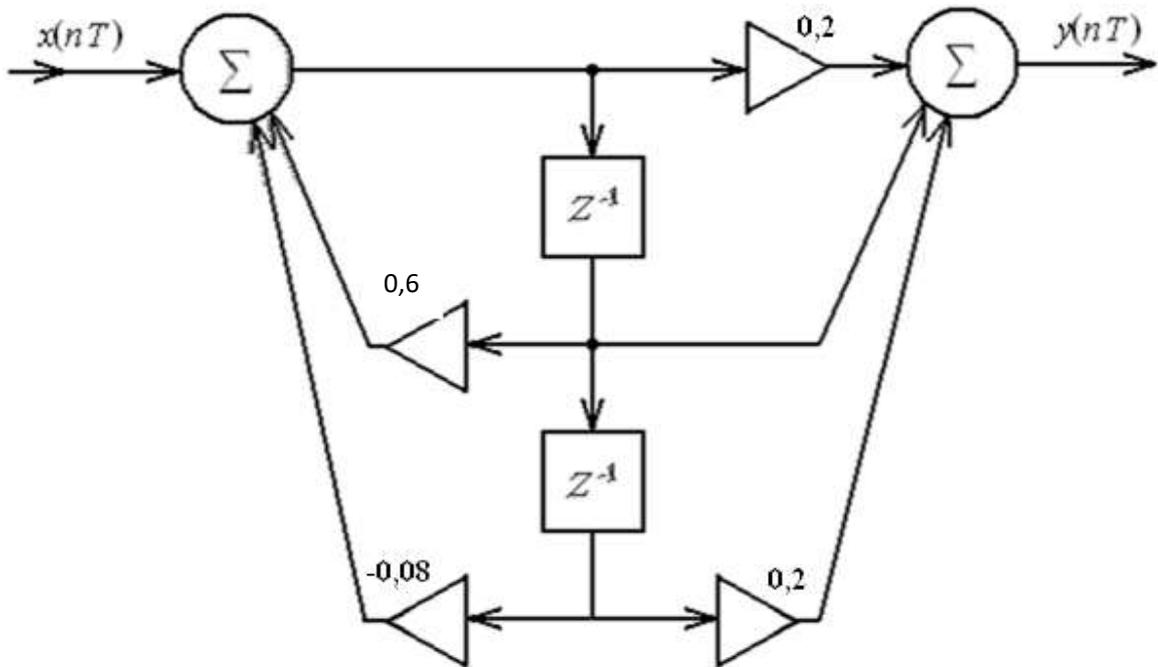


Рисунок 3 Структура ЦФ каноническая.

13. Определить сигнал на выходе фильтра ( $n = 0, 1, \dots, 6$ ). Сигнал  $\{y(kT)\}$  на выходе НЦФ можно рассчитать путем непосредственного использования разностного уравнения, описывающего данный фильтр:

$$y(nT) = 0,2x(nT) + x(nT-T) + 0,2x(nT-2T) + 0,6y(nT-T) - 0,08y(nT-2T).$$

$y(0)$	0,03
$y(T)$	0,418
$y(2T)$	1,5384
$y(3T)$	1,9896
$y(4T)$	1,250688



$y(5T)$	0,591245
$y(6T)$	0,254692

4. Определить, устойчив ли фильтр.

Фильтр является устойчивым, если все полюсы  $H(z)$  располагаются внутри единичной окружности в плоскости  $z$ . Нули могут быть расположены в любой точке комплексной  $z$  – плоскости. Это означает, что для устойчивости фильтра необходимо и достаточно выполнения условия

$$|z_k| < 1, k = 1, 2, \dots, N,$$

где  $z_k$  – полюс функции передаточной функции.

Найдем полюсы  $z_{1,2}$  системной функции РЦФ:

$$z^2 - 0,6z + 0,08 = 0,$$

$$z_1 = 0,4; z_2 = 0,2; |z| < 1.$$

Условие устойчивости РЦФ выполняется, поэтому исследуемый РЦФ устойчив.

5. Найти частотную характеристику  $H(j\omega)$  фильтра. Для описания аналоговых фильтров используют частотную характеристику

$$H(j\omega) = \frac{Y(j\omega)}{X(j\omega)}.$$

Для нерекурсивного дискретного фильтра

$$H(j\omega) = \sum_{k=0}^M h(kT) e^{-j\omega kT}. \text{ Эта функция является периодической с периодом } \frac{2\pi}{T}.$$

$$H(j\omega) = H(z) \Big|_{z=e^{j\omega T}} = \frac{0,2 + e^{-j\omega T} + 0,2e^{-2j\omega T}}{1 - 0,6e^{-j\omega T} + 0,08e^{-2j\omega T}}$$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

6. Найти АЧХ  $A(\omega)$  фильтра.

АЧХ  $A(\omega)$  фильтра будем находить, используя частотную характеристику  $H(j\omega)$  РЦФ.

$$A(\omega) = \sqrt{\frac{\sum_{k=0}^M b_k \cos k\omega T \quad \quad \quad \sum_{k=1}^M b_k \sin k\omega T}{1 - \sum_{k=1}^M a_k \cos k\omega T \quad \quad \quad \sum_{k=0}^M a_k \sin k\omega T}}$$

$$|H(j\omega)| = \left| \frac{0,2 + e^{-j\omega T} + 0,2e^{-2j\omega T}}{1 - 0,6e^{-j\omega T} + 0,08e^{-2j\omega T}} \right| =$$

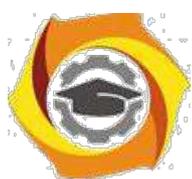
$$21. \sqrt{\frac{(0,2 + \cos \omega T + 0,2\cos(2\omega T))^2 + (\sin \omega T + 0,2\sin(2\omega T))^2}{(1 - 0,6\cos \omega T + 0,08\cos(2\omega T))^2 + (0,6\sin \omega T - 0,08\sin(2\omega T))^2}}$$

7. Найти ФЧХ  $\phi(\omega)$  фильтра.

Аргумент частотной характеристики является ФЧХ заданного НЦФ:

$$20. \quad \phi(\omega) = -\arctg \frac{[\operatorname{Im}[H(j\omega)]]}{[\operatorname{Re}[H(j\omega)]]}$$

$$\phi(\omega) = -\arctg \frac{\sin \omega T + 0,2\sin(2\omega T)}{0,2 + \cos \omega T + 0,2\cos(2\omega T)} - \arctg \frac{0,6\sin \omega T + 0,08\sin(2\omega T)}{1 - 0,6\cos \omega T + 0,08\cos(2\omega T)}$$



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**



Задача 2.

На вход рекурсивного цифрового фильтра (РЦФ) с системой функцией  $H(z)$  подается сигнал  $\{x(kT)\} = (0,05; 1; 0,95)$ .

$a_0$	$a_1$	$a_2$	$b_1$	$b_2$
0,1	1	0,1	2,0	-0,96

Записать алгоритм цифровой фильтрации.

23. помощью системной функции запишем алгоритм цифровой фильтрации РЦФ:

$$H(z) = \frac{0,1 + z^{-1} + 0,1z^{-2}}{1}$$

$$y(nT) = a_0x(nT) + a_1x(nT-T) + \dots + b_1y(nT-T) + \dots + b_{L-1}y(nT-(L-1)T),$$

38. Изобразить схему фильтра в прямой и канонической формах.

Задача 3. На вход согласованного цифрового фильтра (СЦФ) поступает сигнал  $\{x(kT)\} = (0,2; 1; 0,8)$  и белый шум с дисперсией  $\sigma_{вх}^2 = 0,1$ .

Требуется:

37. Определить импульсную характеристику  $\{h(nT)\}$  СЦФ.

38. Найти системную функцию  $H(z)$  фильтра.

39. Записать алгоритм цифровой фильтрации.

40. Изобразить схему фильтра.

41. Определить сигнал на выходе фильтра.

42. Определить максимальную пиковую мощность сигнала на выходе СЦФ  $P_c$  в момент времени  $MT$ .

43. Определить мощность шума на выходе СЦФ  $P_{ш}$ .



8. Найти наибольшую величину отношения сигнал-шум на выходе СЦФ

$$\frac{P_c}{P_{ш}}$$

в момент времени  $mT$ . Определить, возможно ли обнаружение сигнала  $\{x(kT)\}$ .

### РЕШЕНИЕ

13. Если импульсная характеристика согласованного цифрового фильтра будет иметь вид  $h(mT) = x((N - 1)T - mT)$ , то среди всех КИХ-фильтров СЦФ обеспечивает наибольшую величину отношения сигнал-шум. Порядок СЦФ равен  $N=3$  и его ИХ  $h(nT) = x((2 - n)T) = (0,8; 1; 0,2)$ .

14. Z-преобразование ИХ является системной функцией СЦФ:

$$H(z) = \sum_0^{\infty} h(kT) z^{-k},$$

$$H(z) = 0,8 + z^{-1} + 0,2z^{-2}.$$

3. Линейная свертка входного сигнала и ИХ позволяет записать алгоритм цифровой фильтрации

$$y(kT) = \sum_{n=0}^5 x(nT) h(kT - nT)$$

$$y(kT) = 0,8x(kT) + x(kT - T) + 0,2x(kT - 2T).$$

4. Схема фильтра изображена на рис.4

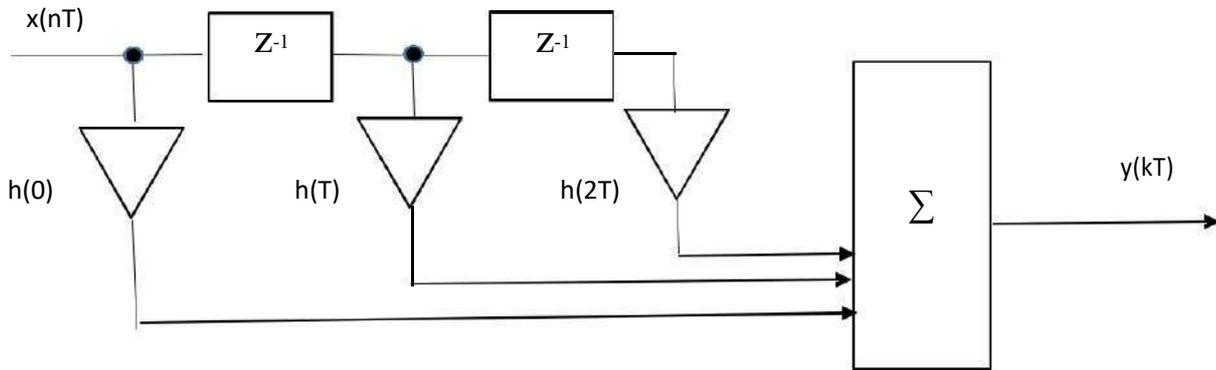


Рисунок 4. Структурная схема СЦФ.

5. Сигнал на выходе  $\{y(kT)\} = (0,16; 1,00; 1,68; 1,00; 0,16)$ .

$x(0)$	$x(T)$	$x(2T)$	$x(3T)$	$x(4T)$
0,20	1,00	0,80	0,00	0,00
$y(0)$	$y(T)$	$y(2T)$	$y(3T)$	$y(4T)$
0,16	1,00	1,68	1,00	0,16

26. В момент времени  $MT$  максимальная пиковая мощность сигнала на выходе СЦФ  $P_c$ :

$$P_c = \sum_{l=0}^5 x(lT) h((2-l)T) =$$

$$31. \quad x(0)h(2T) + x(T)h(T) + x(T)h(T) + x(T)h(T) + x(2T)h(0) + x(3T)h(-T) + x(4T)h(-2T) + x(5T)h(-3T) = 1,$$

68.



7. Мощность шума на выходе СЦФ  $P_{ш}$ :

$$P_{ш} = \sigma_{ex}^2 \sum_{n=0}^{N-1} h^2(nT) = 0,168.$$

40. Наибольшая величина отношения сигнал-шум на выходе СЦФ в момент времени  $MT$ :

$$k_{ш} = \frac{P_c}{P_{ш}} = \frac{1,68}{0,168} = 10 > 1.$$

Сигнал  $\{x(kT)\}$  обнаруживается.

Задача 2: На вход согласованного цифрового фильтра (СЦФ) поступает сигнал  $\{x(kT)\} = (0,05; 1; 0,95)$  и белый шум с дисперсией  $\sigma_{ex}^2 = 0,2$ . Требуется:

55. Определить импульсную характеристику  $\{h(nT)\}$  СЦФ.

56. Найти системную функцию  $H(z)$  фильтра.

57. Записать алгоритм цифровой фильтрации.

58. Изобразить схему фильтра.

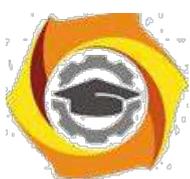
59. Определить сигнал на выходе фильтра.

60. Определить максимальную пиковую мощность сигнала на выходе СЦФ  $P_c$  в момент времени  $MT$ .

15. Определить мощность шума на выходе СЦФ  $P_{ш}$ .

16. Найти наибольшую величину отношения сигнал-шум на выходе СЦФ  $\frac{P_c}{P_{ш}}$

23. момент времени  $MT$ . Определить, возможно ли обнаружение сигнала  $\{x(kT)\}$ .



**Негосударственное частное образовательное учреждение  
высшего образования  
«Технический университет»**

Методические указания по проведению практических занятий по дисциплине «Экономика» составлены в соответствии с рабочей программой по данной дисциплине.

Практические занятия направлены на достижение следующих целей:

41. формирование у студента профессиональных знаний и умения ориентироваться в экономической проблематике и на этой основе принимать эффективные решения.
42. закрепление знаний по определенной теме, приобретенных в процессе контактной работы и самостоятельного изучения источников, применение этих знаний в практической деятельности.
43. приобретение навыков анализа и обобщения собранного материала, грамотного и последовательного изложения содержания темы в письменной форме.

Методика проведения практических занятий предусматривает:

61. Проведение индивидуального и группового опросов, в ходе которых проверяется степень и качество усвоения изучаемого материала.
62. Выполнение студентами практических заданий по каждой теме с составлением отчетов, которые подлежат защите на практических занятиях.



## **2.1. Практическое занятие № 1: Экономика: основные характеристики (2 часа)**

### **2.1.1. Цель занятия**

Приобрести первичные умения использовать экономические категории для описания процессов производства, распределения, обмена и потребления.

### **2.1.2. Основные понятия:**

экономика, хозяйственная деятельность, экономические субъекты, предпринимательские фирмы, домашние хозяйства, производство, распределение, обмен, потребление, общественные блага, ресурсы, издержки упущенных возможностей, кривая производственных возможностей.

### **2.1.3. Задания**

24. Проанализировать и сравнить различные трактовки предмета теоретической экономики, выделить объект и предмет изучения;
25. Используя условные данные, построить кривую производственных возможностей для экономики, производящей два вида благ;
26. Рассчитать альтернативные издержки обучения студента в вузе на основе следующих данных: обучаясь в вузе, студент получает стипендию 1200 руб. Оставив учебу, он может работать продавцом-консультантом в магазине и получать зарплату 20000 руб., или стать курьером и получать 8000 руб.

### **2.1.4. Рекомендации к выполнению заданий:**

На основе изучения рекомендованных источников

- \* Выяснить смысл ограниченности ресурсов и необходимости выбора рационального способа распределения этих ресурсов между альтернативными целями;
- \* Определить цену (издержки) выбора как упущенную возможность получения более высокого дохода по сравнению с выбранным вариантом.

### **2.1.5. Контрольные вопросы**

- в Чем руководствуются экономические субъекты, принимая различные хозяйственные решения?
- в Что такое экономическое благо? Каковы признаки общественных благ?
- в Чем обусловлена необходимость выбора между альтернативными возможностями использования ресурсов?
- в В чем заключается ограниченность ресурсов?
- в Почему кривая производственных возможностей имеет отрицательный наклон?
- в Что собой представляют издержки упущенных возможностей? Почему они имеют тенденцию к росту?



## **2.2. Практическое занятие № 2: Рыночная система. Основы теории спроса и предложения (4 часа).**

### **2.2.1. Цель занятия**

Изучить основы функционирования и механизм рыночной экономики, овладеть навыками хозяйственного поведения в условиях рынка.

### **2.2.2. Основные понятия:**

рынок, конкуренция, конкурентность, совершенная конкуренция, монополистическая конкуренция, олигополия, монополия, спрос, величина спроса, функция спроса, эластичность спроса, функция предложения, равновесие рынка.

### **2.2.3. Задания**

- и Рассчитать индекс Херфиндаля-Хиршмана для рынка, на котором действуют три фирмы. Удельный вес в общих продажах у фирмы А – 40%, у фирмы В – 30% и у фирмы С – 30%. Оценить степень монополизации рынка.
- и Рассчитать, на сколько процентов изменится величина спроса, если цена товара повысится на 4%, а доход увеличится на 5%. Эластичность спроса по цене составляет 1,6, а по доходу – 0,8.
- и Построить бюджетную линию для потребителя, имеющего еженедельный доход 5 тыс.руб. и тратящего его на еду и развлечения, если цена единицы еды 100 руб., цена единицы развлечений 500 руб. Как изменится наклон бюджетной линии, если цена развлечений составит 700 руб.?

### **2.2.4. Рекомендации к выполнению заданий**

- \* Изучить учебную литературу.
- \* Индекс Херфиндаля-Хиршмана рассчитывается как сумма квадратов долей фирм в общем объеме продаж. Чем выше значение индекса, тем выше степень монополизации рынка.
- \* Эластичность спроса на товар показывает степень чувствительности спроса к воздействию на него различных факторов, например, цены или дохода. Коэффициент эластичности показывает, на сколько процентов увеличится или уменьшится спрос на товар в случае изменения его цены или дохода потребителя на один процент.
- \* Бюджетная линия – это геометрическое место точек, представляющих собой доступные потребителю наборы товаров  $x$  и  $y$  при неизменных ценах и фиксированном доходе. В случае изменения цены на один их товаров, бюджетная линия изменяет свой наклон.

### **2.2.5. Контрольные вопросы**

- В В чем различие конкуренции и конкурентности рынка?
- В Чем отличается совершенная конкуренция от несовершенной?
- В Чем отличается монополистическая конкуренция от монополии?
- В От каких факторов зависит величина спроса? Величина предложения?
- В Возможен ли абсолютно неэластичный спрос?
- В Как влияет фактор времени на эластичность спроса и предложения?
- В В чем заключается парадокс Гиффена?



2 Как графически отображается равновесие потребителя?

## 2.3. Практическое занятие № 3: Система планирования на предприятии (4 часа)

### 2.3.1. Цель занятия

Изучить сущность и виды планирования, ознакомиться с структурой, содержанием и назначением бизнес-планов.

### 2.3.2. Основные понятия:

внешняя и внутренняя среда предприятия, стратегическое, тактическое, оперативное планирование, принципы и методы планирования, миссия предприятия, цели предприятия, бизнес-планирование.

### 2.3.3. Задания

- Описать виды и функции планирования на предприятии.
- Изучить стандарты и методики разработки бизнес-планов, технико-экономических обоснований и инвестиционных меморандумов.
- Сформулировать понятие миссии и требования, предъявляемые к ее разработке.
- Проанализировать миссии российских компаний, сделать выводы о полноте и адекватности отображения в них предназначения и смысла деятельности соответствующих организаций.

### 2.3.4. Рекомендации к выполнению заданий

- и Изучить учебную литературу.
- и Ознакомиться с официальными сайтами российских компаний на предмет изучения их миссии и стратегии.
- и Бизнес-план разрабатывается в целях реализации нового проекта, новой бизнес-идеи в отличие от традиционного планирования, охватывающего все стороны текущей и перспективной деятельности предприятия.
- и Основными целями бизнес-плана являются разработка стратегии, привлечение инвестиций, привлечение потенциальных партнеров.

### 2.3.5. Контрольные вопросы

- 42) Каковы критерии выбора стратегии организации?
- 43) Каково назначение бизнес-планов?
- 44) Какова структура компонентов, определяющих содержание миссии?
- 45) В чем различия миссии и целей организации?
- 46) Чем вызвана необходимость разработки «дерева» целей организации?
- 47) Каковы критерии классификации планов организации?
- 48) Каковы базовые стратегии роста организации?
- 49) Каковы этапы разработки бизнес-плана?



## **2.4. Практическое занятие № 4: Макроэкономическое равновесие (4 часа)**

### **2.4.1. Цель занятия**

Изучить основные макроэкономические показатели, характеризующие результаты национальной экономики.

### **2.4.2. Основные понятия:**

система национальных счетов, ВВП, ВНД, НД, индекс цен, дефлятор ВВП, личный доход, располагаемый доход, благосостояние, качество жизни, модель AD-AS, кейнсианская модель равновесия.

### **2.4.3. Задания**

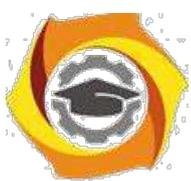
- с Определить величину чистого валового продукта, если заработная плата составила 159 млрд.руб., рентные доходы – 200 млрд.руб., процентные платежи – 166 млрд.руб., прибыль 407 млрд.руб., косвенные налоги – 260 млрд.руб., амортизация – 80 млрд.руб.
- с Определить реальный доход в ценах базисного года, если номинальный доход составил 119,9 ден. ед., а индекс потребительских цен 110%.

### **2.4.4. Рекомендации к выполнению заданий**

- 30) Изучить учебную литературу:
- 31) Чистый валовой продукт определяется как разница между ВВП и амортизацией; величина ВВП может быть определена путем суммирования доходов. Индексы цен показывают изменение во времени общего уровня цен на товары и услуги. При этом в базисном году номинальный доход равен реальному доходу.

### **2.4.5. Контрольные вопросы**

- 19. Для чего необходима разработка системы национальных счетов?
- 20. Чем ВВП отличается от ВНД?
- 21. Каковы методы подсчета ВВП?
- 22. Что такое добавленная стоимость?
- 23. В чем различия номинального и реального ВВП?
- 24. Возможно ли снижение реального ВВП при одновременном росте его номинального объема?
- 25. Является ли ВВП показателем уровня благосостояния в обществе?
- 26. Какими показателями можно оценить качество жизни в стране?
- 27. В чем различия классического и кейнсианского подходов к проблеме равновесия экономики?



## **2.5. Практическое занятие № 5: Государственный бюджет и фискальная политика (2 часа)**

### **2.5.1. Цель занятия**

Рассмотреть структуру доходов и расходов государства и основные направления фискальной политики.

### **2.5.2. Основные понятия:**

государственные финансы, бюджет, дефицит и профицит бюджета, налоговая система, прямые и косвенные налоги, кривая Лаффера, дискреционная и недискреционная фискальная политика, государственный долг, реструктуризация долга.

### **2.5.3. Задания**

46. Определить размер профицита государственного бюджета, если расходы государства на закупку товаров и услуг составляют 1000 млрд. ден.ед., процент по государственно-му долгу, достигшему 500 млрд. ден.ед., - 10% годовых, трансферты – 200 млрд. ден.ед., налоги – 1300 ден.ед.
47. Определить, на какую сумму государству необходимо разместить облигации на внутреннем рынке для покрытия дефицита государственного бюджета, если доходы бюджета составляют 420 млрд. ден.ед., а на внешнем рынке могут быть заимствованы 200 млрд. ден.ед.

### **2.5.4. Рекомендации к выполнению заданий**

18. Изучить учебную литературу:
19. Профицит государственного бюджета образуется в случае, если налоговые поступления превышают планируемые расходы государства. Дефицит бюджета образуется, если расходы государства не покрываются за счет налоговых поступлений и других платежей в бюджет. Заимствования на внутреннем и внешнем рынках являются одной из возможностей финансирования государственных расходов, однако приводят к возникновению и росту государственного долга.

### **2.5.5. Контрольные вопросы**

- В В чем заключается сущность и значение государственных финансов?
- В Следует ли государству ежегодно добиваться сбалансированного или профицитного бюджета?
- В Какова структура расходной части федерального бюджета РФ?
- В Кто утверждает и исполняет федеральный бюджет РФ?
- В Приведет ли повышение налогов к увеличению поступлений в бюджет?
- В Каковы способы урегулирования внешнего долга?
- В В чем различия дискреционной и недискреционной фискальной политики?
- В В чем различия прогрессивного и пропорционального налогообложения?



## **2.6. Практическое занятие № 6: Денежный рынок и денежно-кредитная политика государства (2 часа)**

### **2.6.1. Цель занятия**

Рассмотреть структуру денежного рынка и роль банковской системы в формировании денежного предложения, в том числе функции центрального банка как регулятора на денежном рынке.

### **2.6.2. Основные понятия:**

деньги, ликвидность, денежные агрегаты, спрос на деньги, предложение денег, равновесие на денежном рынке, денежная эмиссия, коммерческий банк, центральный банк, банковский мультипликатор, обязательные и избыточные резервы банка, операции на открытом рынке, ставка рефинансирования, ключевая ставка.

### **2.6.3. Задания**

- e Рассчитать денежный агрегат M1, если наличные деньги в обращении составляют 130 млрд. ден.ед., средства на расчетных, текущих и специальных счетах предприятий и населения – 4,45 млрд. ден.ед., вклады населения в сберегательном банке до востребования 2 млрд. ден.ед., депозиты населения и предприятий в коммерческих банках – 5 млрд. ден.ед.
- e Сравнить различные инструменты денежно-кредитной политики, выделить наиболее эффективные.

### **2.6.4. Рекомендации к выполнению заданий**

- В Изучить учебную литературу:
- В Денежные агрегаты используются для характеристики денежного предложения. M1 является наиболее узким денежным агрегатом и включает деньги для сделок, т.е. наиболее ликвидные средства такие, как наличные деньги, средства на расчетных и текущих счетах предприятий и населения, вклады до востребования и т.п.
- В Наиболее распространенными средствами денежно-кредитной политики являются операции на открытом рынке, изменение резервной нормы, изменение ставки рефинансирования. Перечисленные инструменты оказывают воздействие на величину денежного предложения.

### **2.6.5. Контрольные вопросы**

- с Что такое деньги? Тожественны ли понятия «деньги» и «ликвидность»?
- с Какова структура и источники предложения денег?
- с Чем определяется количество денег в обращении и их покупательная способность?
- с От каких факторов зависит спрос на деньги?
- с В чем заключаются активные операции коммерческих банков?
- с Каковы функции центрального банка?
- с В чем значение обязательных резервов коммерческого банка?
- с В чем заключается мультипликационный эффект на денежном рынке?



### **3.1. Практическое занятие № 1: Система планирования на предприятии (2 часа)**

#### **3.1.1. Цель занятия**

Изучить сущность и виды планирования, ознакомиться с структурой, содержанием и назначением бизнес-планов.

#### **3.1.2. Основные понятия:**

внешняя и внутренняя среда предприятия, стратегическое, тактическое, оперативное планирование, принципы и методы планирования, миссия предприятия, цели предприятия, бизнес-планирование.

#### **3.1.3. Задания**

- Описать виды и функции планирования на предприятии.
- Изучить стандарты и методики разработки бизнес-планов, технико-экономических обоснований и инвестиционных меморандумов.
- Сформулировать понятие миссии и требования, предъявляемые к ее разработке.
- Проанализировать миссии российских компаний, сделать выводы о полноте и адекватности отображения в них предназначения и смысла деятельности соответствующих организаций.

#### **3.1.4. Рекомендации к выполнению заданий**

19. Изучить учебную литературу.
20. Ознакомиться с официальными сайтами российских компаний на предмет изучения их миссии и стратегии.
21. Бизнес-план разрабатывается в целях реализации нового проекта, новой бизнес-идеи в отличие от традиционного планирования, охватывающего все стороны текущей и перспективной деятельности предприятия.
22. Основными целями бизнес-плана являются разработка стратегии, привлечение инвестиций, привлечение потенциальных партнеров.

#### **3.1.5. Контрольные вопросы**

51. Каковы критерии выбора стратегии организации?
52. Каково назначение бизнес-планов?
53. Какова структура компонентов, определяющих содержание миссии?
54. В чем различия миссии и целей организации?
55. Чем вызвана необходимость разработки «дерева» целей организации?
56. Каковы критерии классификации планов организации?
57. Каковы базовые стратегии роста организации?
58. Каковы этапы разработки бизнес-плана?



## **3.2. Практическое занятие № 2: Госбюджет и фискальная политика (2 часа)**

### **3.2.1. Цель занятия**

Рассмотреть структуру доходов и расходов государства и основные направления фискальной политики.

### **3.2.2. Основные понятия:**

государственные финансы, бюджет, дефицит и профицит бюджета, налоговая система, прямые и косвенные налоги, кривая Лаффера, дискреционная и недискреционная фискальная политика, государственный долг, реструктуризация долга.

### **3.2.3. Задания**

- в Определить размер профицита государственного бюджета, если расходы государства на закупку товаров и услуг составляют 1000 млрд. ден.ед., процент по государственно-му долгу, достигшему 500 млрд. ден.ед., - 10% годовых, трансферты – 200 млрд. ден.ед., налоги – 1300 ден.ед.
- в Определить, на какую сумму государству необходимо разместить облигации на внутреннем рынке для покрытия дефицита государственного бюджета, если доходы бюджета составляют 420 млрд. ден.ед., а на внешнем рынке могут быть заимствованы 200 млрд. ден.ед.

### **3.2.4. Рекомендации к выполнению заданий**

- В Изучить учебную литературу.
- В Профицит государственного бюджета образуется в случае, если налоговые поступления превышают планируемые расходы государства. Дефицит бюджета образуется, если расходы государства не покрываются за счет налоговых поступлений и других платежей в бюджет.
- В Заимствования на внутреннем и внешнем рынках являются одной из возможностей финансирования государственных расходов, однако приводят к возникновению и росту государственного долга.

### **3.2.5. Контрольные вопросы**

- с В чем заключается сущность и значение государственных финансов?
- с Следует ли государству ежегодно добиваться сбалансированного или профицитного бюджета?
- с Какова структура расходной части федерального бюджета РФ?
- с Кто утверждает и исполняет федеральный бюджет РФ?
- с Приведет ли повышение налогов к увеличению поступлений в бюджет?
- с Каковы способы урегулирования внешнего долга?
- с В чем различия дискреционной и недискреционной фискальной политики?
- с В чем различия прогрессивного и пропорционального налогообложения?

### **3.3. Практическое занятие № 3: Денежный рынок и денежно-кредитная политика государства (2 часа)**

#### **3.3.1. Цель занятия**

Рассмотреть структуру денежного рынка и роль банковской системы в формировании де-нежного предложения, в том числе функции центрального банка как регулятора на денежном рынке.

#### **3.3.2. Основные понятия:**

деньги, ликвидность, денежные агрегаты, спрос на деньги, предложение денег, равновесие на денежном рынке, денежная эмиссия, коммерческий банк, центральный банк, банковский мультипликатор, обязательные и избыточные резервы банка, операции на открытом рынке, ставка рефинансирования, ключевая ставка.

#### **3.3.3. Задания**

- и Рассчитать денежный агрегат M1, если наличные деньги в обращении составляют 130 млрд. ден.ед., средства на расчетных, текущих и специальных счетах предприятий и населения – 4,45 млрд. ден.ед., вклады населения в сберегательном банке до востребования 2 млрд. ден.ед., депозиты населения и предприятий в коммерческих банках – 5 млрд. ден.ед.
- и Сравнить различные инструменты денежно-кредитной политики, выделить наиболее эффективные.

#### **3.3.4. Рекомендации к выполнению заданий**

- х Изучить учебную литературу:
- х Денежные агрегаты используются для характеристики денежного предложения. M1 является наиболее узким денежным агрегатом и включает деньги для сделок, т.е. наиболее ликвидные средства такие, как наличные деньги, средства на расчетных и текущих счетах предприятий и населения, вклады до востребования и т.п.
- х Наиболее распространенными средствами денежно-кредитной политики являются операции на открытом рынке, изменение резервной нормы, изменение ставки рефинансирования. Перечисленные инструменты оказывают воздействие на величину денежного предложения.

#### **3.3.5. Контрольные вопросы**

- в Что такое деньги? Тождественны ли понятия «деньги» и «ликвидность»?
- в Какова структура и источники предложения денег?
- в Чем определяется количество денег в обращении и их покупательная способность?
- в От каких факторов зависит спрос на деньги?
- в В чем заключаются активные операции коммерческих банков?
- в Каковы функции центрального банка?
- в В чем значение обязательных резервов коммерческого банка?
- в В чем заключается мультипликационный эффект на денежном рынке?