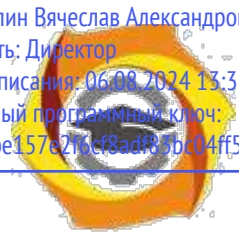


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Лапин Вячеслав Александрович
Должность: Директор
Дата подписания: 06.08.2024 13:32:58
Уникальный программный ключ:
df48b51be157e21ccf8adf87bc04ff59a6aeacac



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

ЗАДАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ

Направление подготовки	<i>09.04.01 Информатика и вычислительная техника</i>
Направленность (профиль)	<i>Бизнес-информатика</i>
Уровень высшего образования	<i>магистратура</i> <i>(бакалавриат, специалитет, магистратура)</i>
Квалификация выпускника	<i>магистр</i>



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 применении 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

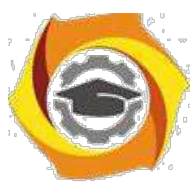


должно быть основано на формализации определения понятия «неясности» [BGI].

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

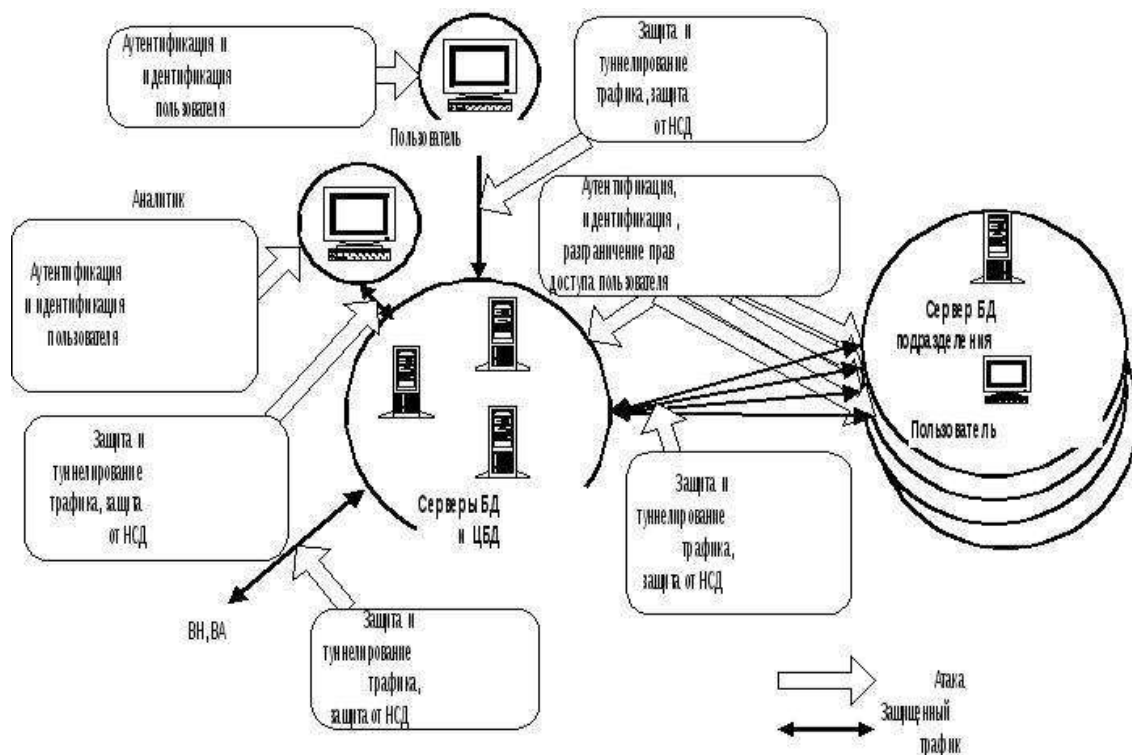


Рис. 1 - Методы компенсации угроз информационной безопасности

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).

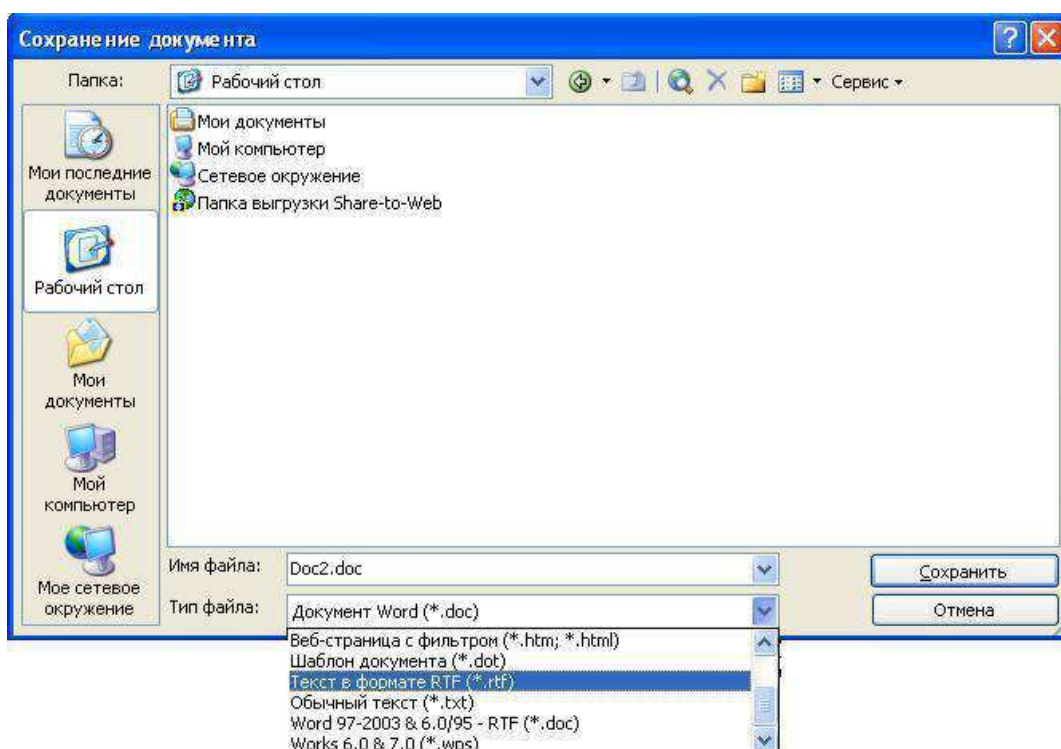


Рис. 7

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.

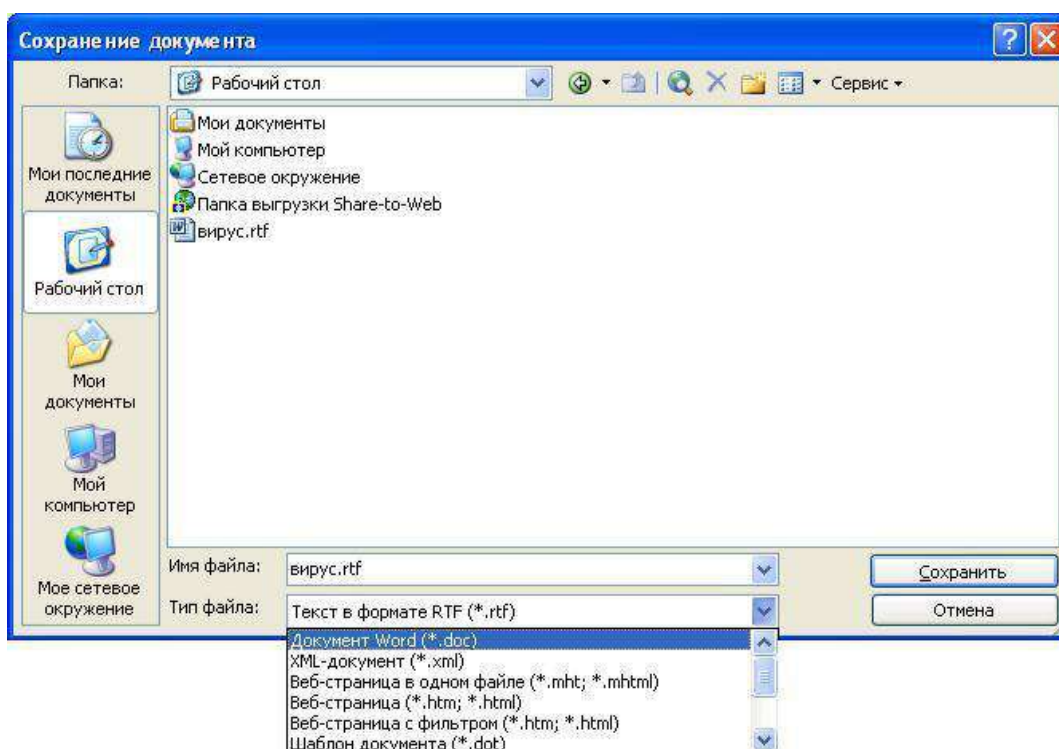
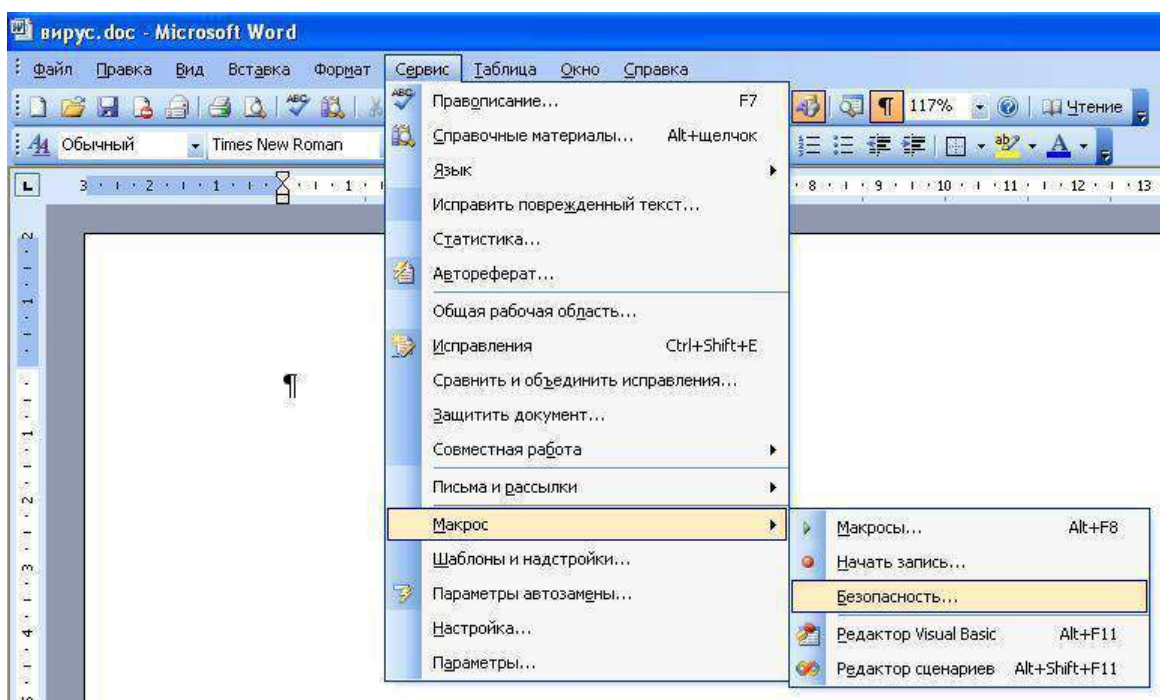


Рис. 8

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).

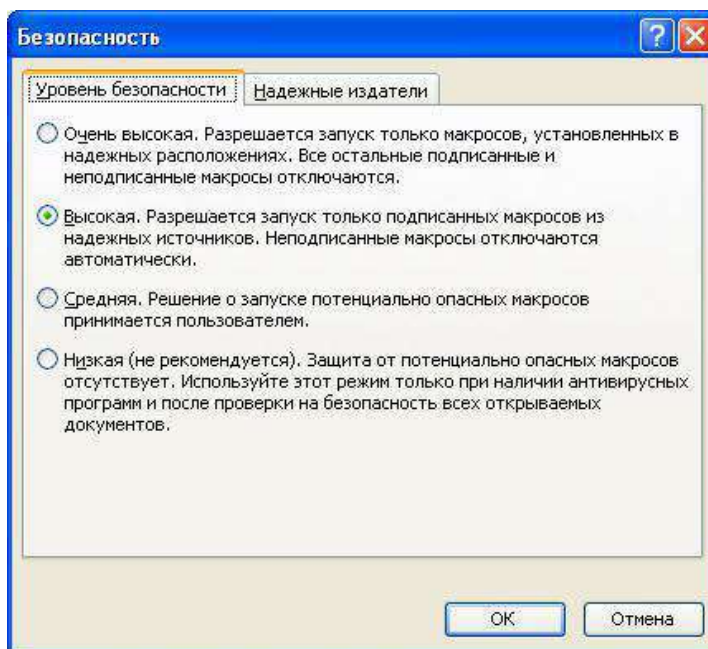


Рис. 10

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 гроху-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).

5. В 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.

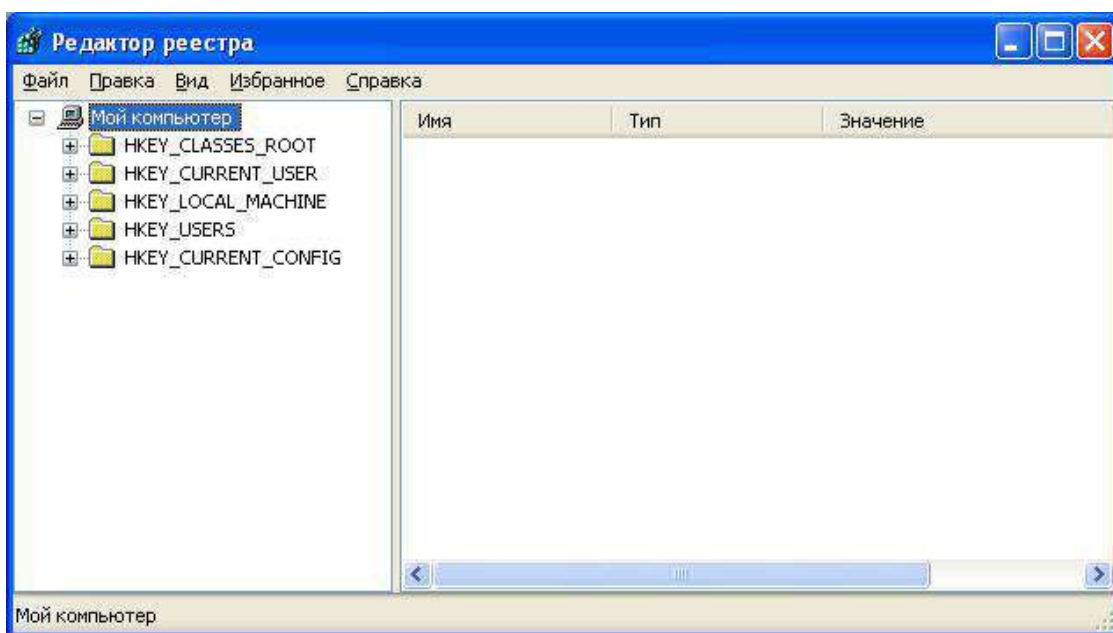


Рис. 11

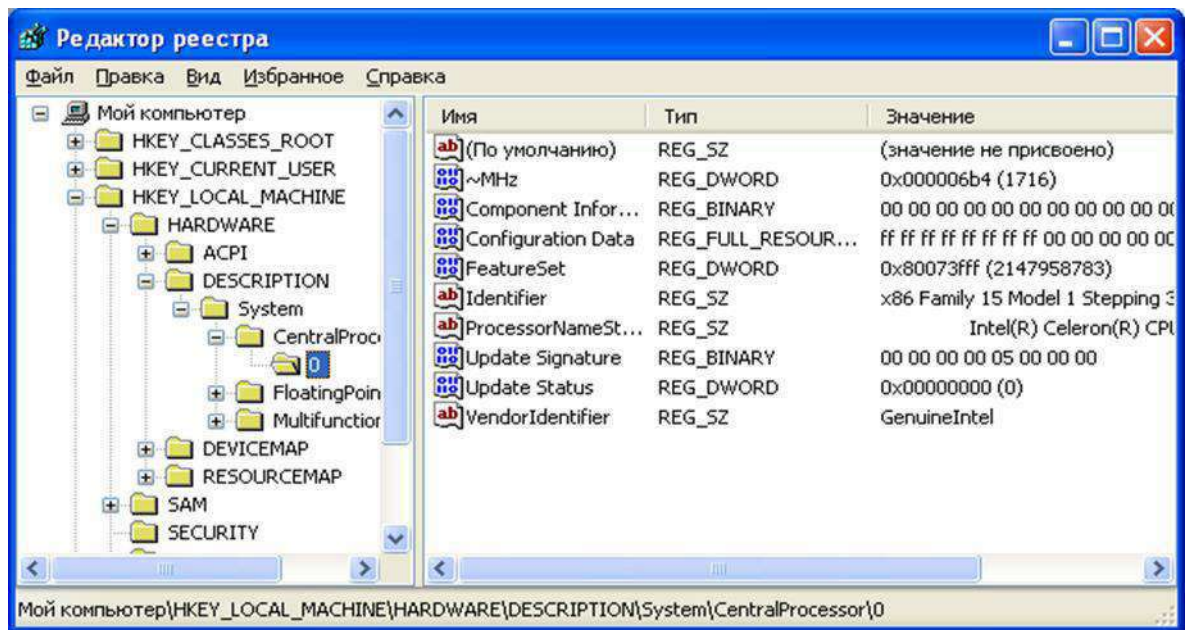


Рис. 12

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).

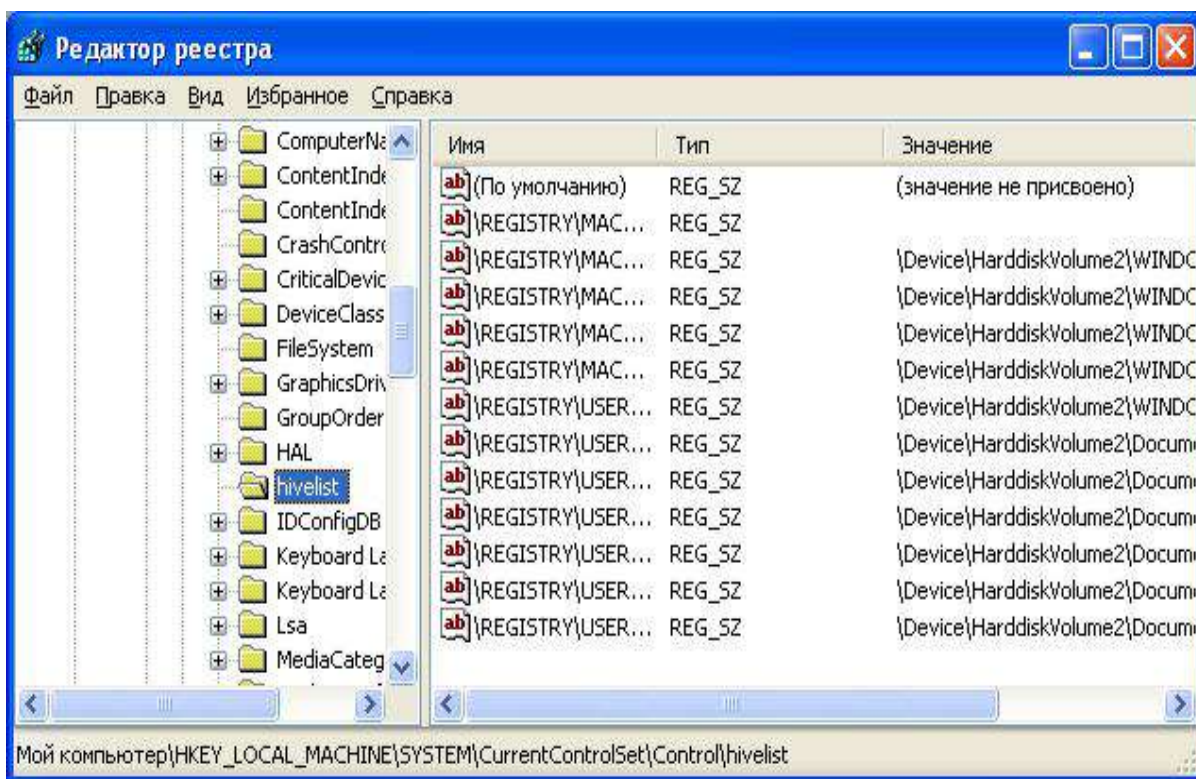


Рис. 13

/ 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).

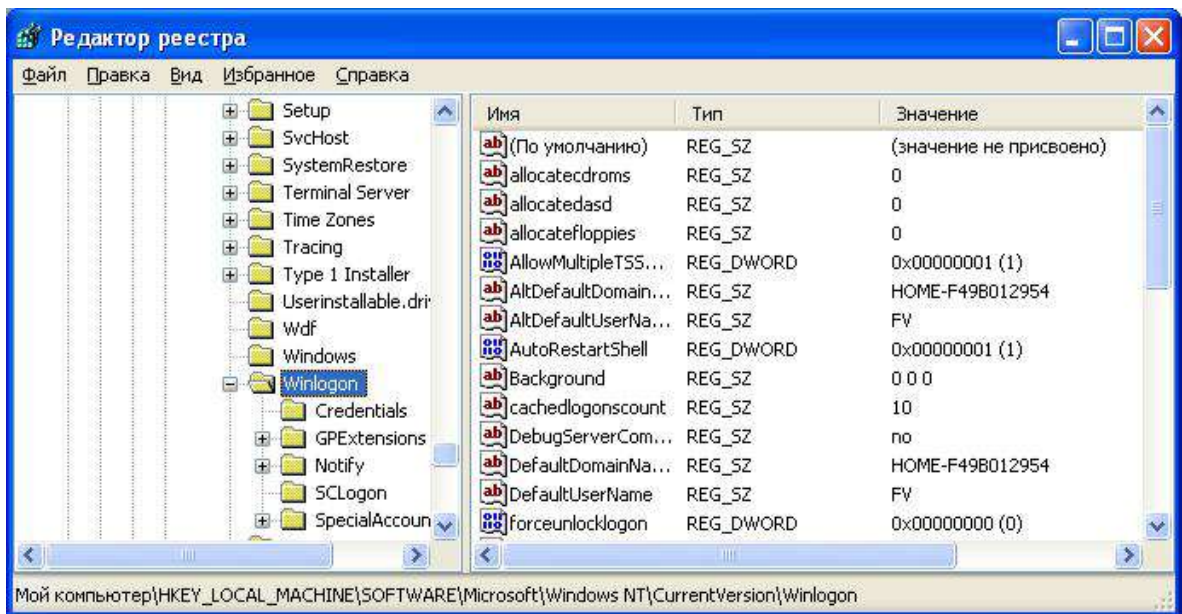


Рис. 14

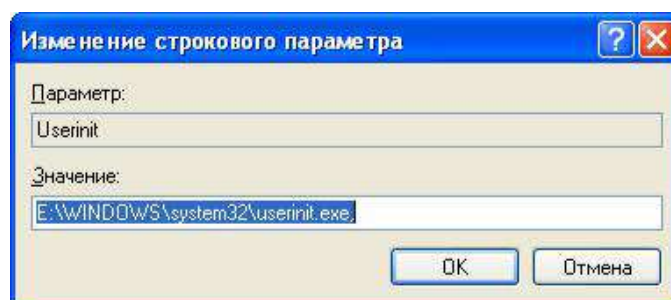
1) Если в указанном ключе содержатся дополнительные записи, то это могут быть «троянские программы».

2) В этом случае проанализируйте место расположения программы, обратите внимание на время создания файла и сопоставьте с Вашими действиями в это время.

1) Если время создания файла совпадает со временем Вашей работы в Интернете, то возможно, что в это время Ваш компьютер был заражен «троянским конем».

1. Для удаления этой записи необходимо дважды щелкнуть на названии ключа (или при выделенном ключе выбрать команду **Изменить** из меню **Правка** программы **regedit.exe**).

2. В открывшемся окне в поле **Значение** (рис. 15) удалите ссылку на подозрительный файл.





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

Рис. 15



4. Закройте программу **regedit.exe**.
5. Перейдите в папку с подозрительным файлом и удалите его.
6. Перезагрузите операционную систему и выполните пункты задания 1-4.

1) 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.

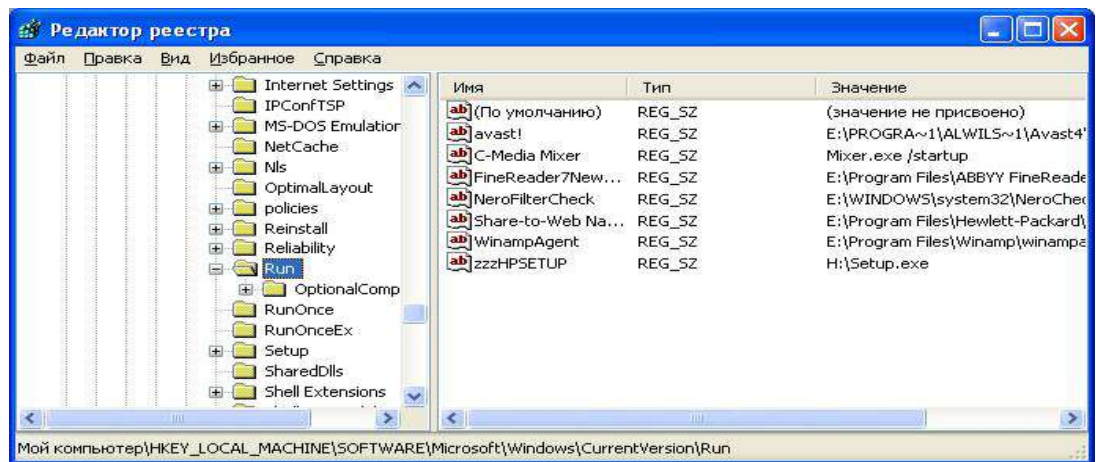


Рис. 16

3. В рассматриваемом примере автоматически запускается резидентный антивирус и его планировщик заданий, а также утилита, относящаяся к программе Nero (запись на CD).

1. Если в указанном разделе есть записи вызывающие подозрения, то выполните пункты 6-14 предыдущего задания.

Задания для самостоятельной работы

1 Проверьте содержимое ключа **HKEY_LOCAL_MACHINE\ Software\ Microsoft\ WindowsNT\ CurrentVersion\ Winlogon\ System (REG_SZ)**.



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

45

2. Зафиксируйте этапы работы, используя команду PrintScreen клавиатуры.
1. 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:

1. личности, общества, государства от воздействия недоброкачественной информации;
2. информации и информационных ресурсов от неправомерного и несанкционированного воздействия посторонних лиц;
3. 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

2. 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).

2) 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 (специальные 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 4.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:

7. *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;

1. *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;

2. 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.

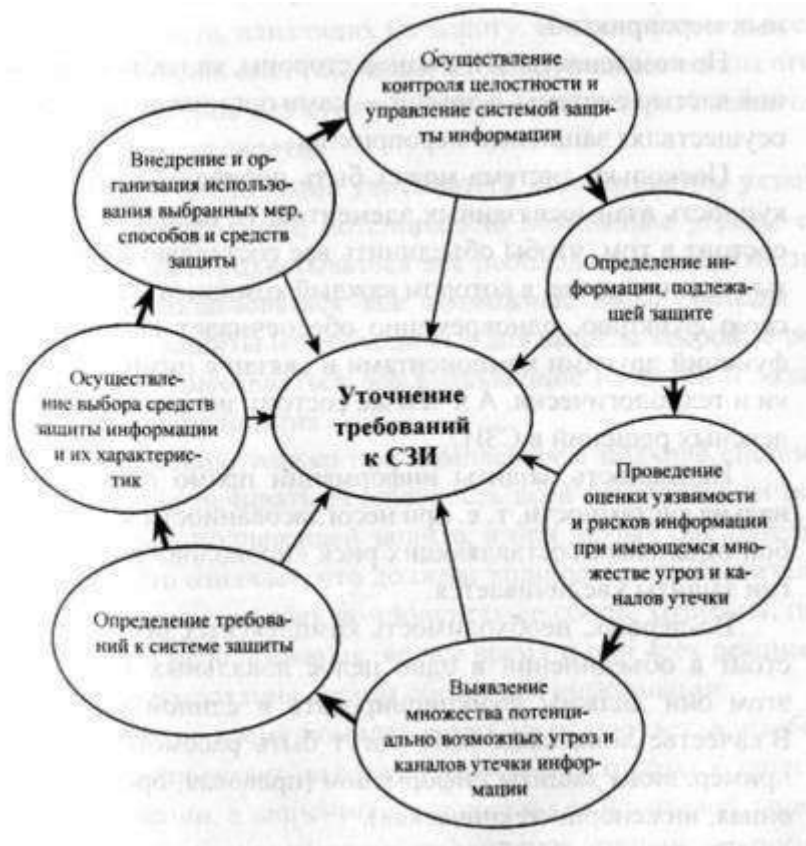


Рис. 1. Непрерывный цикл создания СЗИ

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.

1. 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

а) 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.

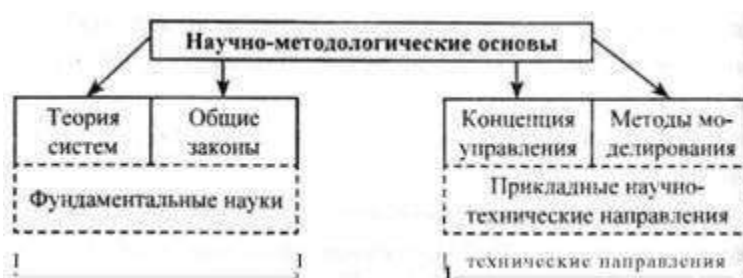


Рис. 2. Состав научно-методологических основ КСЗИ

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 представляющих 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).

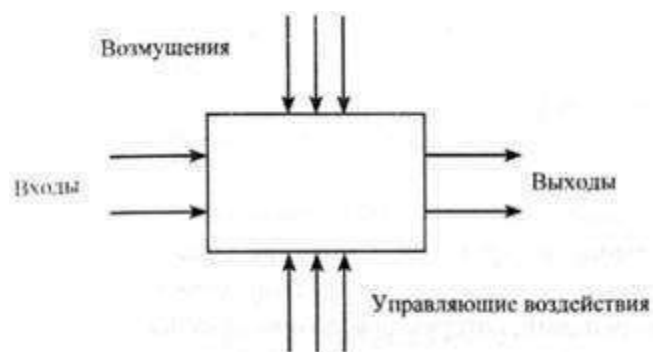


Рис. 3. Обобщенное представление системы



**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.

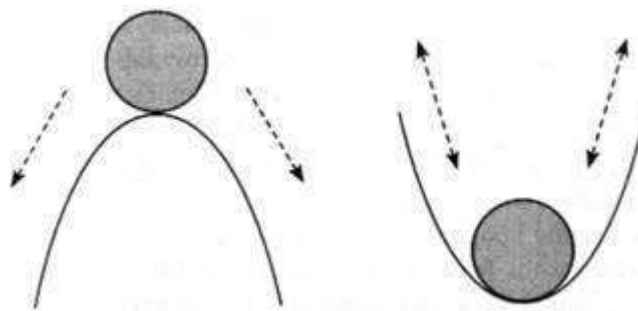


Рис. 4. Система в устойчивом состоянии (справа) и неустойчивом (слева)

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

1) 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.

3) 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]:

1. 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 микроскопического 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.

1. Настроить брандмауэр на работу с **Веб-сервером** (HTTP), FTP-сервером и зафиксировать соответствующее окно для отчета (PrinScreen).
1. Включить **журнал безопасности**.
2. После выполнения задания 1 и 2 подключиться к Интернету и посетить любой веб-сервер.
3. Просмотреть **журнал безопасности**.
4. 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.

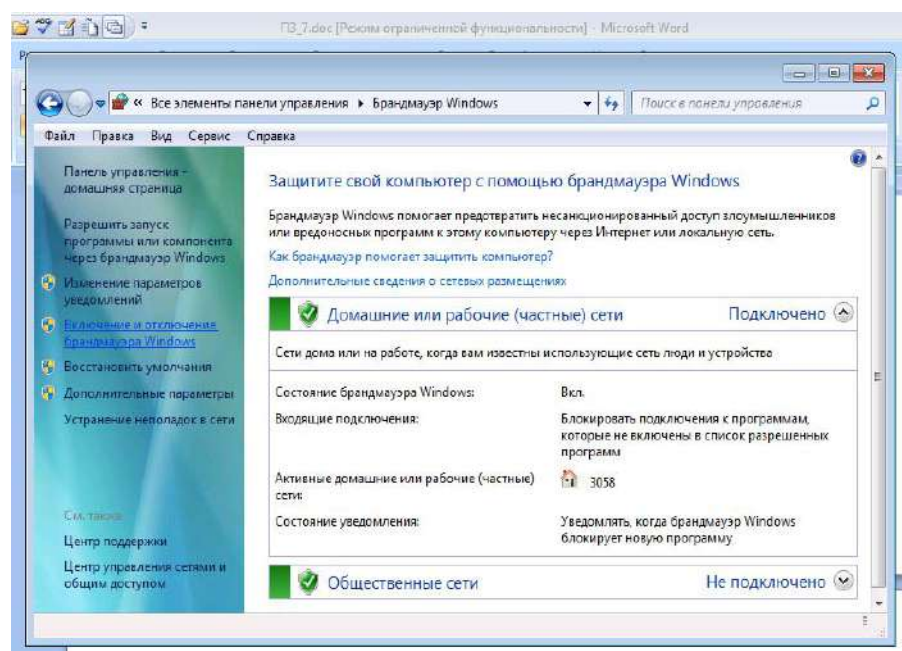
Алгоритм выполнения работы.

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 **Включение брандмауэра Windows** под каждым сетевым размещением (рисунок 2), которое следует защитить, и нажмите кнопку **ОК**.

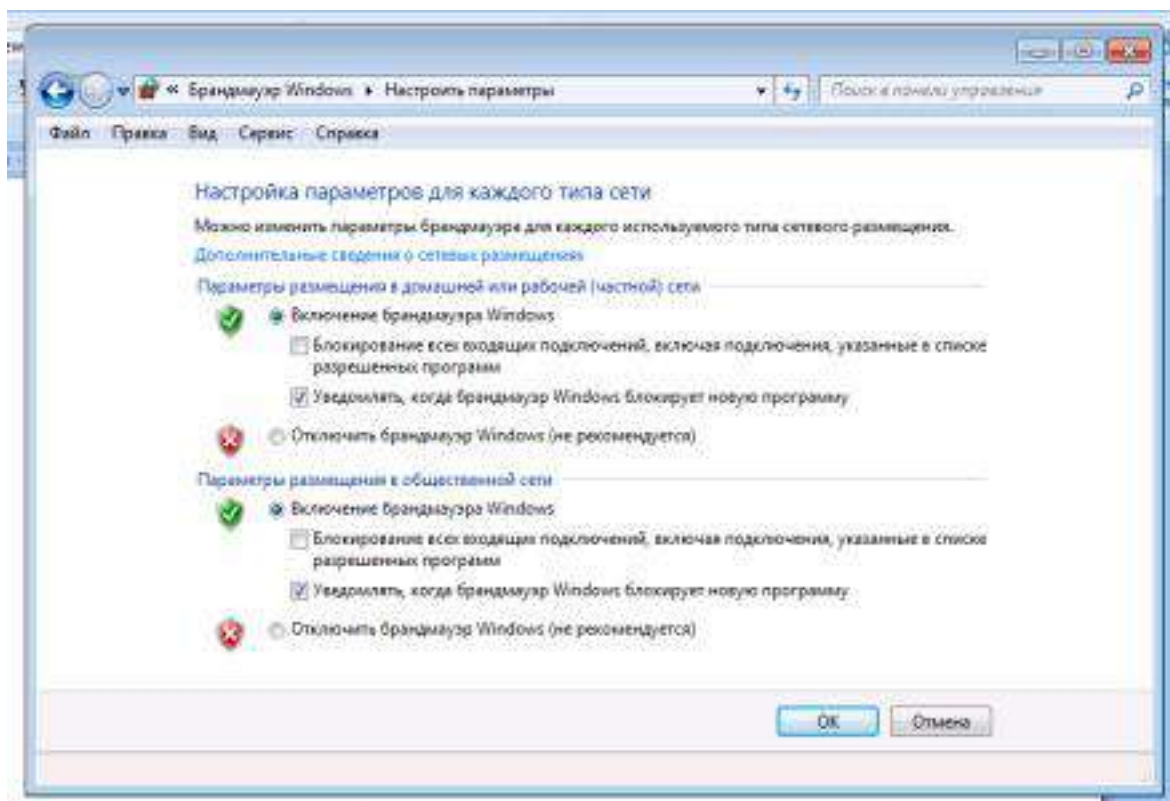


Рисунок 2 – Окно включения/отключения брандмауэра

- 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).

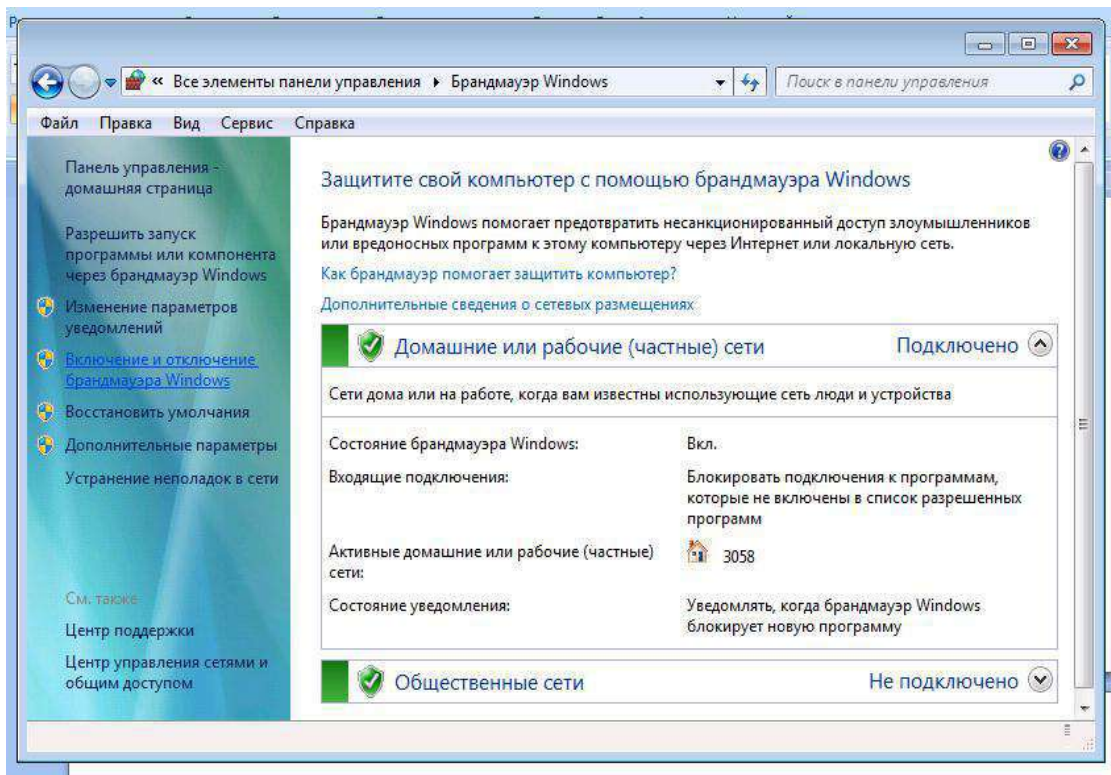


Рисунок 3 – Окно выбора дополнительных параметров брандмауэра

- 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.

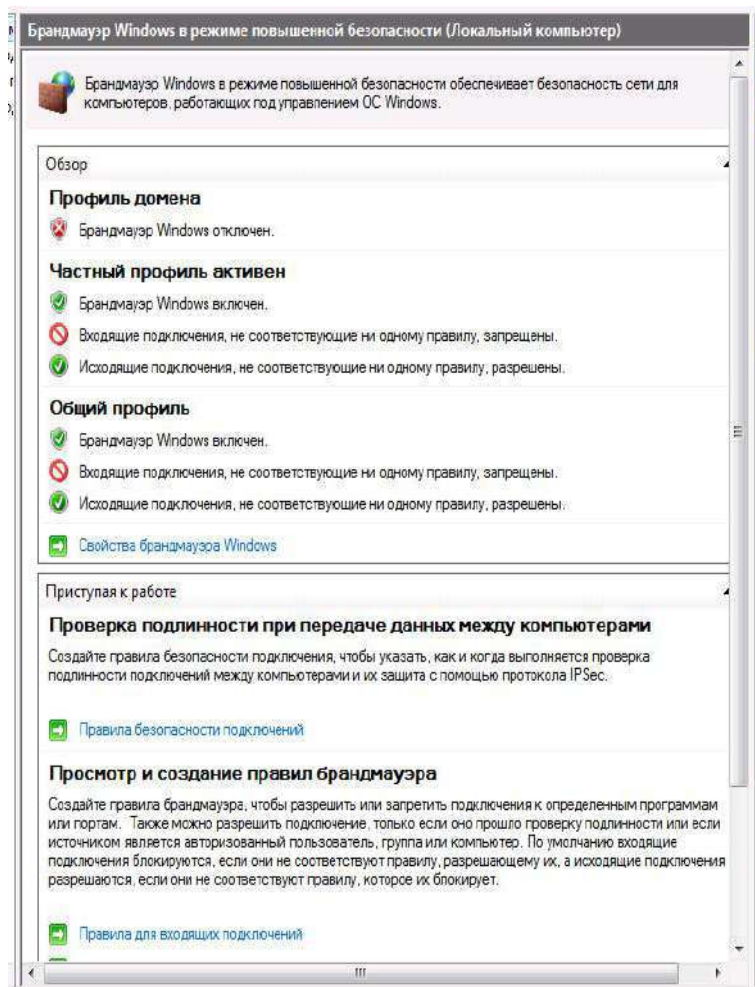


Рисунок 4 – Окно брандмауэра в режиме повышенной безопасности

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**.

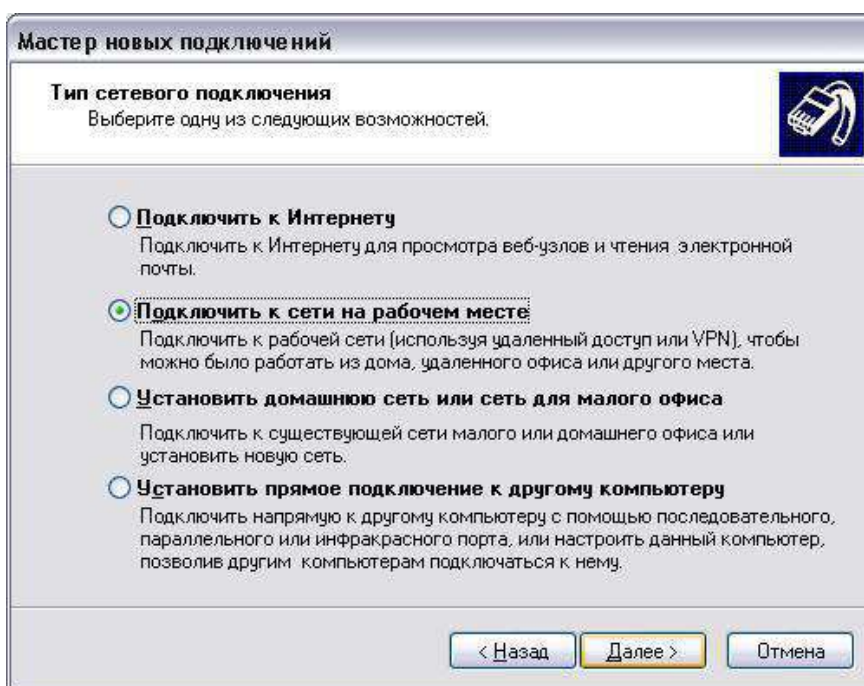


Рис. 5

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.

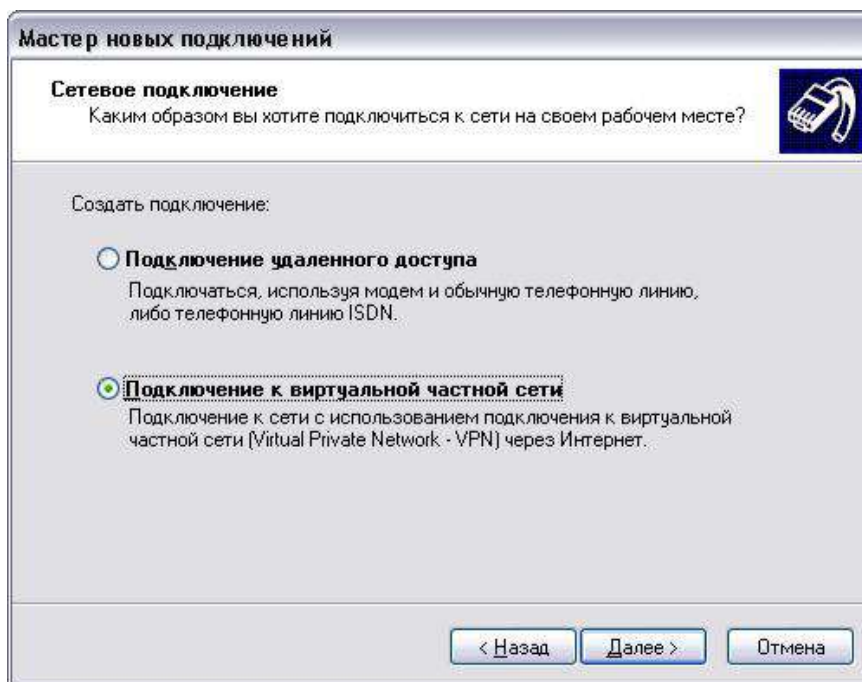
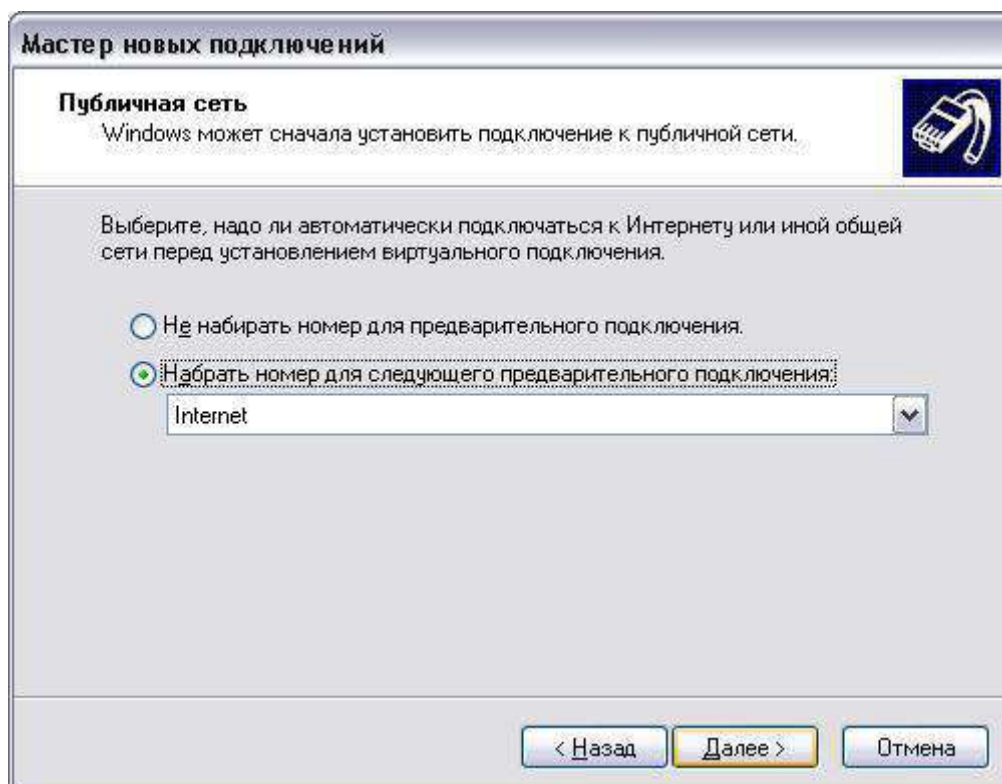


Рис. 6





**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).

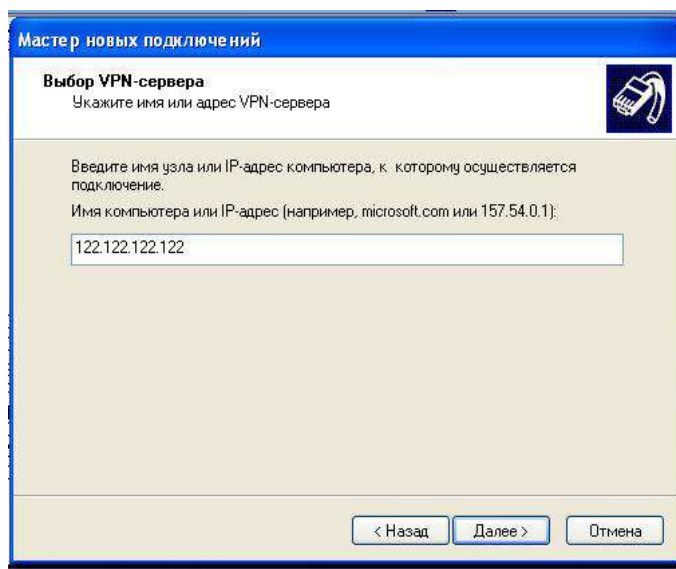
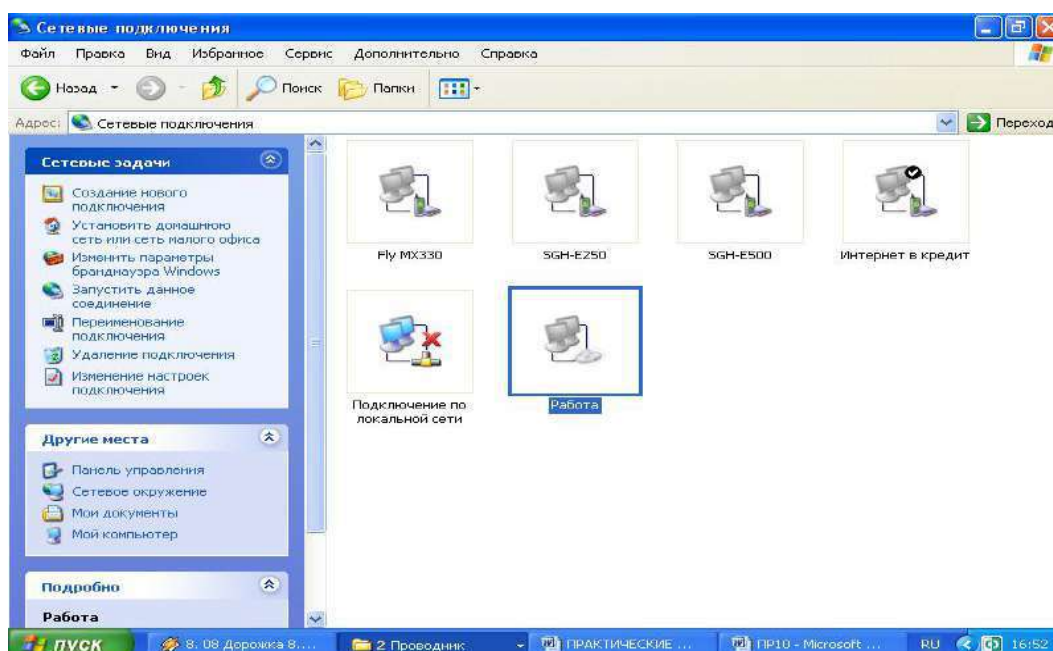


Рис. 8

- 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

2. Какие механизмы безопасности используются при реализации VPN-подключения?

1. Что такое «туннель» и в чем состоит принцип «туннелирования»?
2. 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).

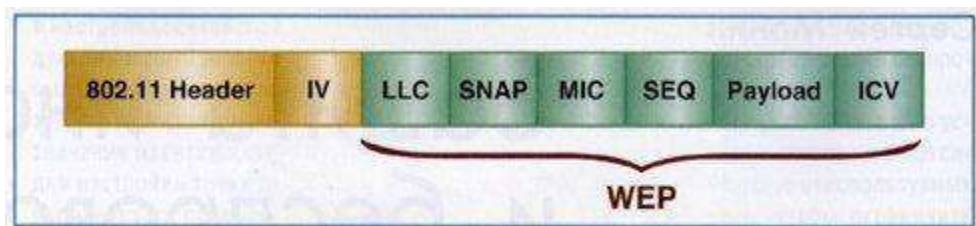


Рисунок 2 - Структура пакета 802.11x при использовании TKIP-PPK, MIC и шифрации по WEP.

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



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, 1. 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 постулов 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 $(o) >$ 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, независящих 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



1. 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.

б. 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)**

3) Режим пакетной обработки

4) 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**.

2) **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.

3) **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.

4) 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



3. зависимости от того, в каком порядке при мультипрограммном режиме выполняются программы пользователей, различают режимы пакетной обработки задач и коллективного доступа.

4. режиме пакетной обработки задачи выстраиваются в одну или несколько очередей и последовательно выбираются для их выполнения.

Режим коллективного доступа

7. 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:

2) **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.

2. 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**.

2 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.

3. 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

2. Щелчком правой кнопки мыши по свободному месту Рабочего стола откройте контекстное меню и выберите команду **Свойства**.

3. В открывшемся диалоговом окне **Свойства: Экран** перейдите на вкладку **Заставка**.

4. В раскрывающемся списке **Заставка** выберите какую-либо заставку.

5. 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:

4. иметь длину не менее семи символов (наиболее надежные пароли состоят из 7 - 14 символов);

5. 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 `~!@#\$\$%^&*()_+={}|[\:";'<>

?,./

3. содержать хотя бы один специальный символ на позициях со второй по шестую;
4. существенно отличаться от ранее использованных паролей;
5. не содержать фамилии или имени пользователя;
6. 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 определённому объекту идентификации. Пароли для защиты дисков делятся:

по типу объектов идентификации: пароли пользователей, ресурсов
5. 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.

5. 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.

2. в 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 характеризующей метке 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.

1. в 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.

2. в 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)

3. В 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
7. 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 2. 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.

4. *заключении* диссертации излагают итоги выполненного исследования, рекомендации, перспективы дальнейшей разработки темы, внедрения проекта. Рекомендованный объем 1-2 страницы. Стиль 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 2.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), \geq (больше или равно), \leq (меньше или равно), \neq (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

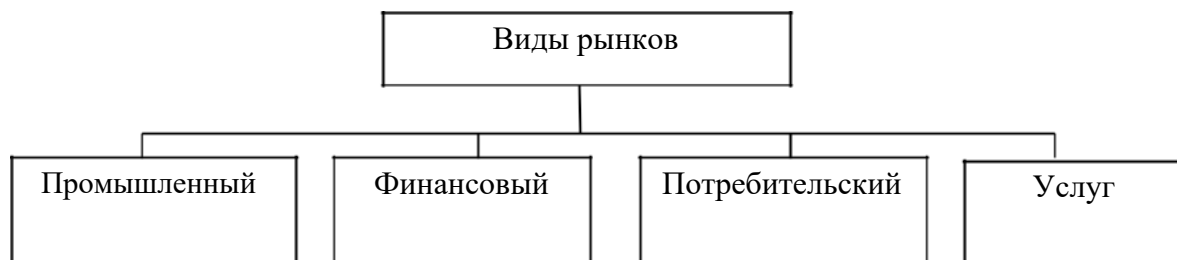


Рисунок 1.2 – Классификация рынков

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-



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.: одобрен 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																													
10.	ТЗ на выполнение маги- 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»**

8. Л.М. Гальчук Английский язык в научной среде: практикум устной речи: Учебное пособие / - 2изд. Москва: Вузовский учебник, НИЦ ИИНФРА-М, 2017г.

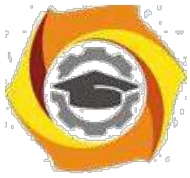
9. Маньковская З.В. Английский язык в ситуациях повседневного делового общения: учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

10. З.В. Маньковская Деловой английский язык: ускоренный курс : учеб. пособие (Высшее образование: Бакалавриат). Москва: ИИНФРА-М, 2017г.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 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.

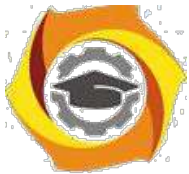
Один студент, по желанию, будет выполнять роль председателя, ведущего конференции и оценит доклад каждого участника. Его задача правильно начать, направлять ход конференции и закончить событие.



**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:

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.

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:

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 № 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:

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.



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

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.

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:

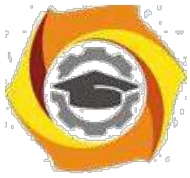
Изучить справочный материал в [1] с. 51 – 55, (3) с. 30, [2] с. 115 – 118.

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

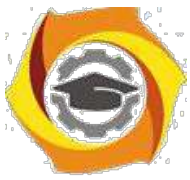
1. Цели занятия:

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:

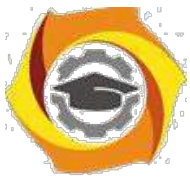
- 3.1. Выполнить грамматические упражнения: 1 – устно, 2, 3 – письменно.
- 3.2. Прочитать и перевести текст «Computer graphics»; ответить на вопросы к тексту.
- 3.3. Дать развернутый устный ответ на один из контрольных вопросов по теме текста.

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...)

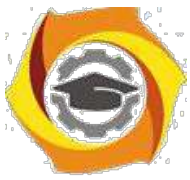
2 – обоснование (на чем вы основываетесь, довод в поддержку вашей позиции) - ...потому, что...(Because...)

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-четырёх 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.



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.



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

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

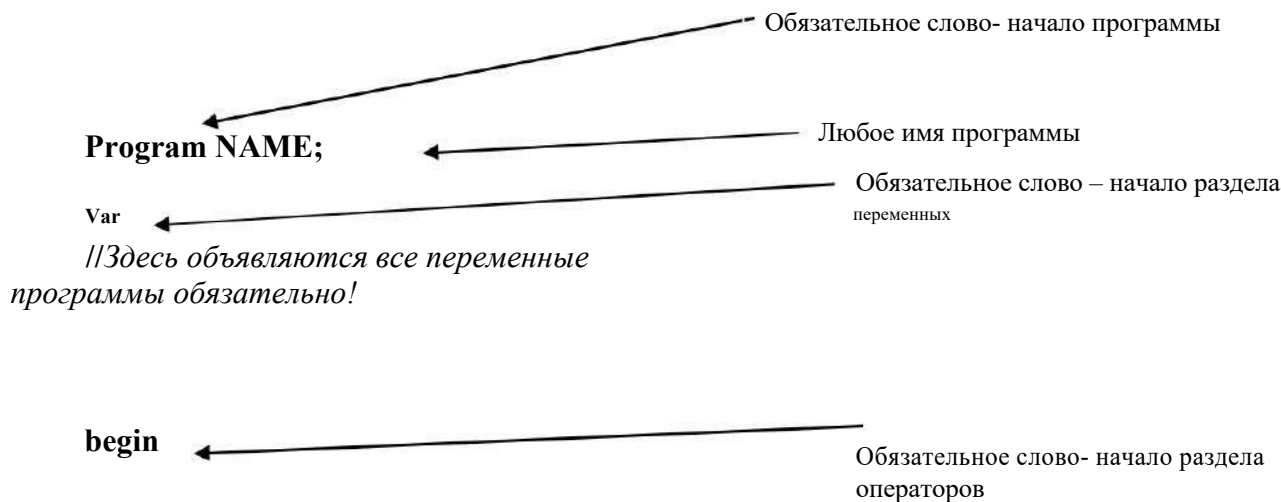
Общие сведения о языке Pascal

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»

4 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

6. 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.

7. 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:

$$= a^n = e^{\ln a^n}, \text{ тогда } y = \exp(n * \ln(a)).$$

4. 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;

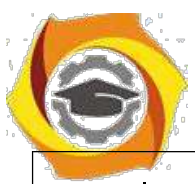
и для округления числа (в таблице 1 эта операция обозначается символами " $()$ ") использовать встроенную функцию **ROUND**;

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.	$z = \frac{\pi i}{2} - \sqrt{2x} - \frac{x + y^2}{0.75 \operatorname{tg} x + y }$ $m = (z)$ $n = [z]$	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.

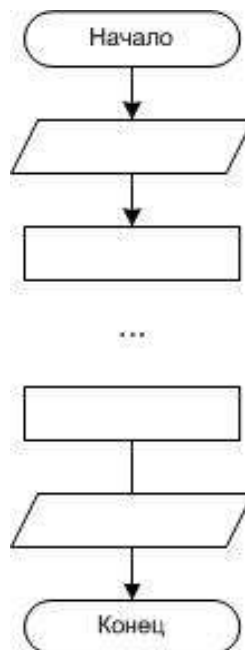


Рисунок 1 – Программа алгоритма линейной структуры

7. Скриншот окна среды программирования с введенным текстом программы представлен на рисунке 2.

```
File Edit Search Run Compile Debug Tools Options Window Help
\TEST1_1.PAS
Program test1_1;
Uses crt;
VAR
i,j: integer;
c,x,y: real;
Begin
clrscr;
Writeln ('Введите числа x и y (для разделения чисел используйте пробел, для
ReadLn (x,y);

c:=(y-pi)*cos(pi/4)/(1+ln(abs(1-y)))+exp((ln((x*y)/(1+0.3*x)))/3);
i:=round(c);
j:=trunc(c);
Writeln ('*****');
Writeln ('* x=',x:6:3,' y=',y:6:3,' *');
Writeln ('* c=',c:8:4,' *');
Writeln ('* i=',i,' j=',j,' *');
Writeln ('*****');
Writeln ('Для завершения работы программы нажмите Enter. ');
ReadLn;
End.
```

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;

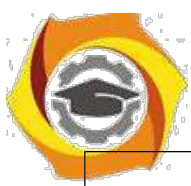
В после выполнения программы и проверки полученного результата сделать скриншоты: окна среды программирования с введенным текстом программы и окна запущенной программы.



**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.

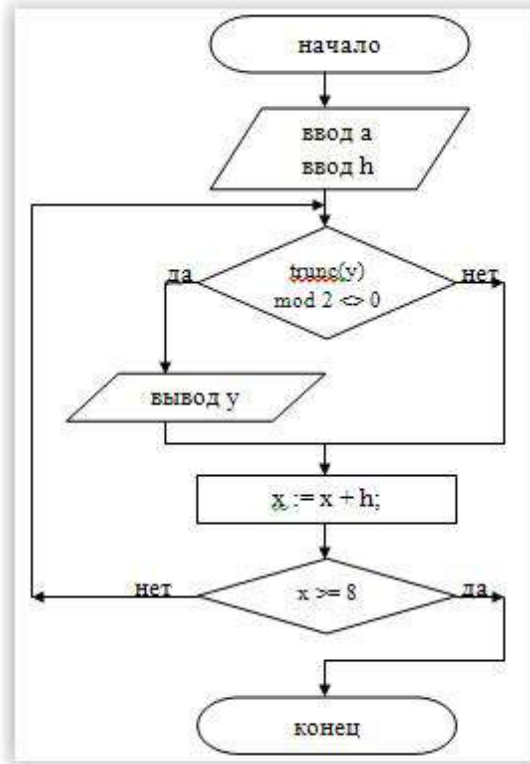


Рисунок 4 – Программа алгоритма разветвляющейся структуры

/ Скриншот окна среды программирования с введенным текстом программы приведен на рисунке 5.

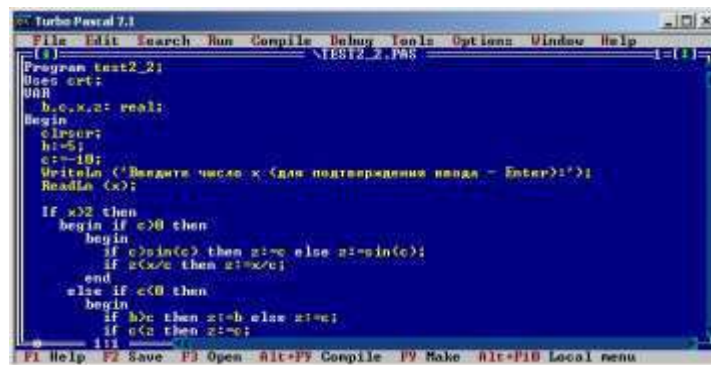


Рисунок 5 – Исходный код программы 2



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

Скриншот окна запущенной программы представлен на рисунке 6.

```
с:\ 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;

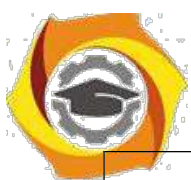
7. использовать в качестве параметра цикла переменную целого типа (счётчик числа повторений цикла);

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 $\epsilon =$ _____, воспользовавшись разложением в ряд:</p> $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$
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 $\epsilon = 10^{-5}$ 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 $\epsilon = 10^{-5}$ константу Эйлера (основание натурального логарифма), воспользовавшись разложением в ряд, $x=1.1$:</p> $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$ <p>Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.</p>
11	<p>Вычислить $\cos 0.7$ с точностью до $\epsilon = 10^{-5}$, 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 $\epsilon = 0.001$, 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 $\epsilon = 10^{-5}$, 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. $y = \frac{1}{\cos^2 x + \ln \operatorname{tg} B^A} + Cx;$</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. $y = \frac{Ae^{\sin x} + \cos(B+C)}{C}; \sin(x+B)$</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	-15,42	0,372431031	0,20271	-0,53871	-0,36899
b	3	-10,25	0,378481296	0,20271	0,61856	0,79434
	6	-5,08	0,447021348	0,20271	0,84017	1,08449
c	5	0,09	0,598017063	0,20271	0,95012	1,34543
	1	5,26	0,866255348	0,20271	0,99653	1,66008
h	7	10,43	1,282997321	0,20271	0,98937	2,06966
	5,17	15,6	1,828356173	0,20271	0,92732	2,55297
		20,77	2,373283466	0,20271	0,7953	2,96587
		25,94	2,695032797	0,20271	0,5076	2,99992
		31,11	2,625193416	0,20271	-0,63962	1,78286
		36,28	2,202447729	0,20271	-0,84979	1,14995
		41,45	1,635291969	0,20271	-0,95489	0,47769
		46,62	1,125219822	0,20271	-0,99767	-0,07516
		51,79	0,760242177	0,20271	-0,98707	-0,42953
		56,96	0,535902455	0,20271	-0,92114	-0,58795
		62,13	0,416040478	0,20271	-0,78327	-0,56994
		67,3	0,369924463	0,20271	-0,47199	-0,30478
		72,47	0,383629844	0,20271	0,65924	0,84016
		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;G109;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:



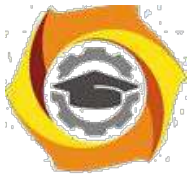
**Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»**

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<=-\$1\$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		
	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	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 ПРЯВЫСИТИ ДОПУСТИМОЕ ЧИСЛО ПОПЫТОК!» «ПРОЩАНИЕ!» (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.

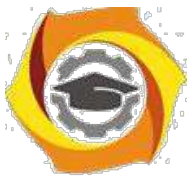
Линейные процессы имеют место, например, при вычислении арифметических выражений.

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
Базовые типы данных	Логический boolean	Символьный (литерный) char	Целые	Вещественные	Нетипизированный указатель pointer
			Конструируемые типы		
		Интервал (диапазон) budni = mo .. fr;			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:



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

```
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.

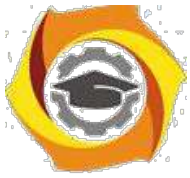
Список ввода - это последовательность имен переменных, разделенных запятыми.

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
```

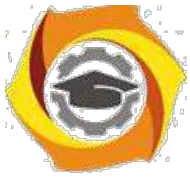
Особенно важен формат при выводе вещественных переменных. К примеру, если не указать формат, то число 10.5 будет выведено как 1.0500000000E+0001. Такой формат называется записью с плавающей точкой.

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+уууу`). 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 $p = \frac{a+b+c}{2}$

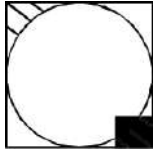
$$h_b = \frac{2}{b} \sqrt{p(p-a)(p-b)(p-c)} \quad h_c = \frac{2}{c} \sqrt{p(p-a)(p-b)(p-c)}$$

И Даны координаты вершин некоторого треугольника. Вычислить его периметр.

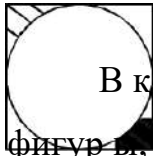


Негосударственное частное образовательное учреждение
высшего образования
«Технический университет»

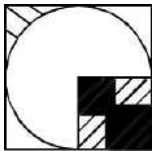
В квадрат вписана окружность. Определить площадь заштрихованной части фигуры, если известна длина сторон квадрата.



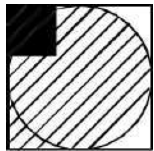
В квадрат вписана окружность. Определить площадь заштрихованной части фигуры, если известен радиус окружности.



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:

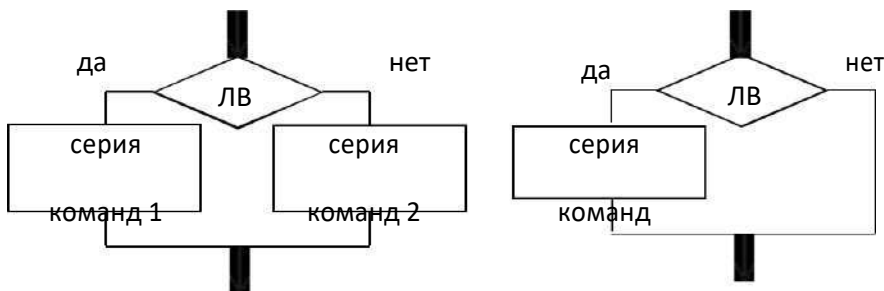
```
if i>0 then if s>2  
    then s:=1  
    else  
    else s:=-1;
```

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.

2. 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. $c^2 = a^2 + b^2$.

3. Составить программу, печатающую значение `true`, если указанное высказывание является истинным, и `false` в противном случае: число c является средним арифметическим чисел a и b .

4. Составить программу, печатающую значение `true`, если указанное высказывание является истинным, и `false` в противном случае: число c является средним геометрическим чисел a и b .

5. 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»**

3. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: произведение натуральных чисел a и b кратно числу c .
4. 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 .
5. 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.
6. 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 .
5. 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.
6. 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.
7. Определите, попадает ли точка $M(x, y)$ в круг радиусом R с центром в точке (x_0, y_0) .
7. Заданы площади круга и квадрата. Определите, поместится ли квадрат в круге.
8. 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.



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

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, к-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.



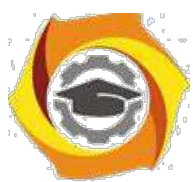
```
Program error2;  
  
Var   Vect: array[1..5] of integer;  
  
      Sum, i : integer;  
  
Begin  
  
      I:=1;  
  
      While i<5 DO begin      sum:=sum+vect[i];  
  
                              I:=i+1;  
  
      End;  
  
End.
```

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() {...};
```

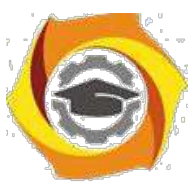
```
int B() {...};
```

```
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.

6. 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.

7. 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.

8. 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.



9. 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;

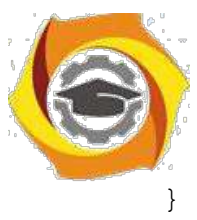
выражение 2 (условие) - до и после каждого прогона цикла.

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 скрин-шот 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 $z = \operatorname{tg}(2x) - \sin(x)$ 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 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.

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 следующем 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 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



**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 1. 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 2. 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:

4. Искусство деловых отношений в Европе;
5. 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 : «по-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.

4. 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»**

4. 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, 5. 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. 2. 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 24 Тажд-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 (руссск.) 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- волчьей 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

1. Этносы - субъекты исторического развития.
2. 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:

1. Дайте определение этикета, речевого этикета и ситуативного этикета (используйте учебную литературу). 27

2. 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.

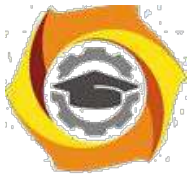
М.Монтень : « Мы берем на хранение чужие мысли и знания, только и всего. Нужно, однако, сделать их собственными. Мы уподобляемся человеку, который, нуждаясь в огне, отправился за ним к соседу, и , найдя у него прекрасный, яркий огонь, стал греться у чужого очага, забыв о своем намерении разжечь очаг 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 1. 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 2. 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 она 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: «порок-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.

1. Сформулируйте свое отношение к нравственным проблемам современности путем градации их по степени важности с вашей точки зрения. (Список проблем смотри ниже).

2. Напишите сочинение – миниатюру на тему «Самая актуальная нравственная проблема современности».

Нравственные проблемы современности: -
проблема взаимоотношения поколений;



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

1. проблема патриотического воспитания, отсутствия патриотизма;
2. проблема однополюсной любви; - насилие в обществе и семье;
3. духовная деградация общества; - терроризм;
4. алкоголизм и наркомания; - проституция и порнография;
5. кризис семьи и брака;
6. деньги – мировой эквивалент всего?
7. проблема самореализации личности;
8. проблема эвтаназии; - проблема абортотворения;
9. проблема лжи и лицемерия;
10. агрессия СМИ, проблема интервенции сознания в самосознание;
11. аномия; - безнравственность современного бизнеса;
12. проблема веры и безверия;
13. проблема ранних браков;
14. проблема одиночества;
- проблема экологии;
15. 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»**

4. 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.

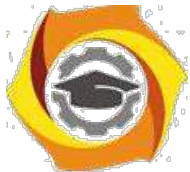
Форма проведения: контрольная работа по заранее известным вопросам. Дискуссионное обсуждение вопросов по теме.

Вопросы к контрольной работе:

11. Проблемы этногенеза славян.
12. Происхождение Российского государства.
13. Хроника первых веков
14. Феодалная раздробленность и иностранная интервенция на Русь в XII-XV вв.
15. Монголо-татары и Русь: проблемы взаимовлияния. Было ли иго? Современные оценки.
16. Политика Московских князей.
17. Личность и деятельность Ивана III.
18. Воссоединение земель вокруг Москвы в XIV- первой половине XV вв.
19. Московское царство в XVI в. Иван IV.
20. Причины и ход Смутного времени.
21. Россия в XVII веке.
22. 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:

- 5 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>
- 6 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>
- 7 История: учеб. пособие / Ю.А. Шестаков. — М. : РИОР : ИНФРА-М, 2019. — 248 с.
<https://znanium.com/catalog/product/1014722>
- 8 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>
- 9 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:

- 7. Внутренняя и внешняя политика царизма в нач. XIX в.
- 8. Отечественная война 1812 г.
- 8. Кризис крепостнической системы.
- 9. Кавказская война.
- 10. Культура в 1-й четверти XIX века.
- 11. Политическая реакция и реформы в период правления Николая I.
- 12. Крымская война.
- 13. Реформы Александра II.
- 10. Развитие капитализма в пореформенной России.
- 5. Общественные движения России 70-х гг. XIX в.
- 6. Официальная идеология самодержавия и общественная мысль России второй четверти и середины XIX в.
- 7. Причины свертывания реформаторского процесса.
- 8. 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.



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:

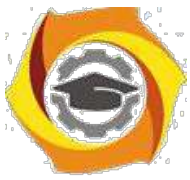
- В СССР в 20-30-е гг. 20 века.
- В Великая Отечественная война.
- В СССР в 50-80-е гг. 20 века.
- В Современный этап развития русской истории.

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:

10 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.



9. 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.

14. 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.

9. 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,



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 ± 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 "x".

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 Р53724-2009 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).

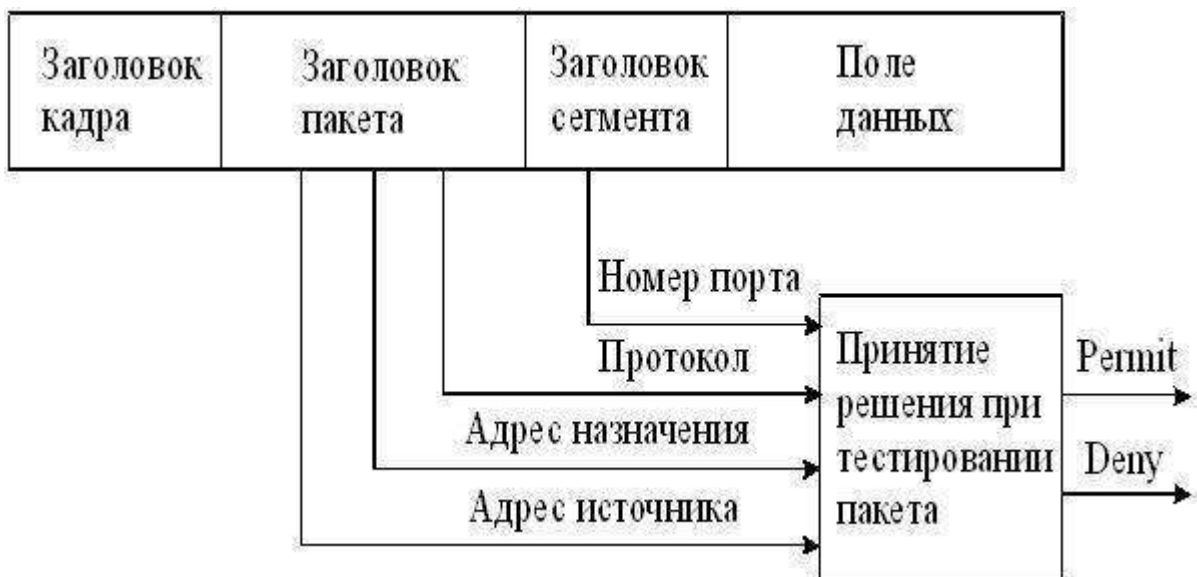


Рисунок 1.1 – Анализ заголовков пакета



**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

11 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:

15. Создание списка доступа в режиме глобального конфигурирования.

16. 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.

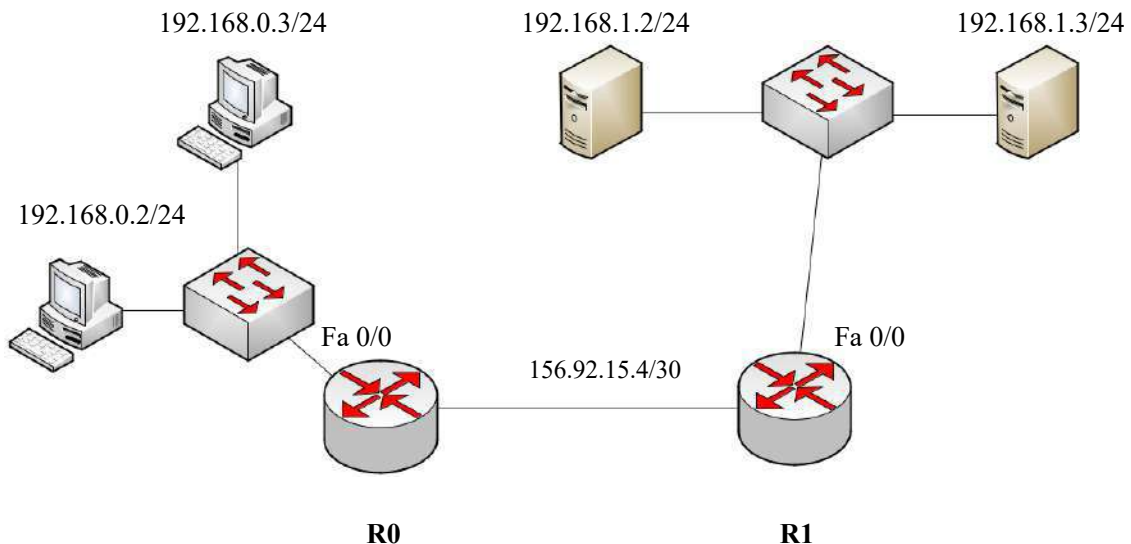


Рисунок 1.2 – Схема сети

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.

10. 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»

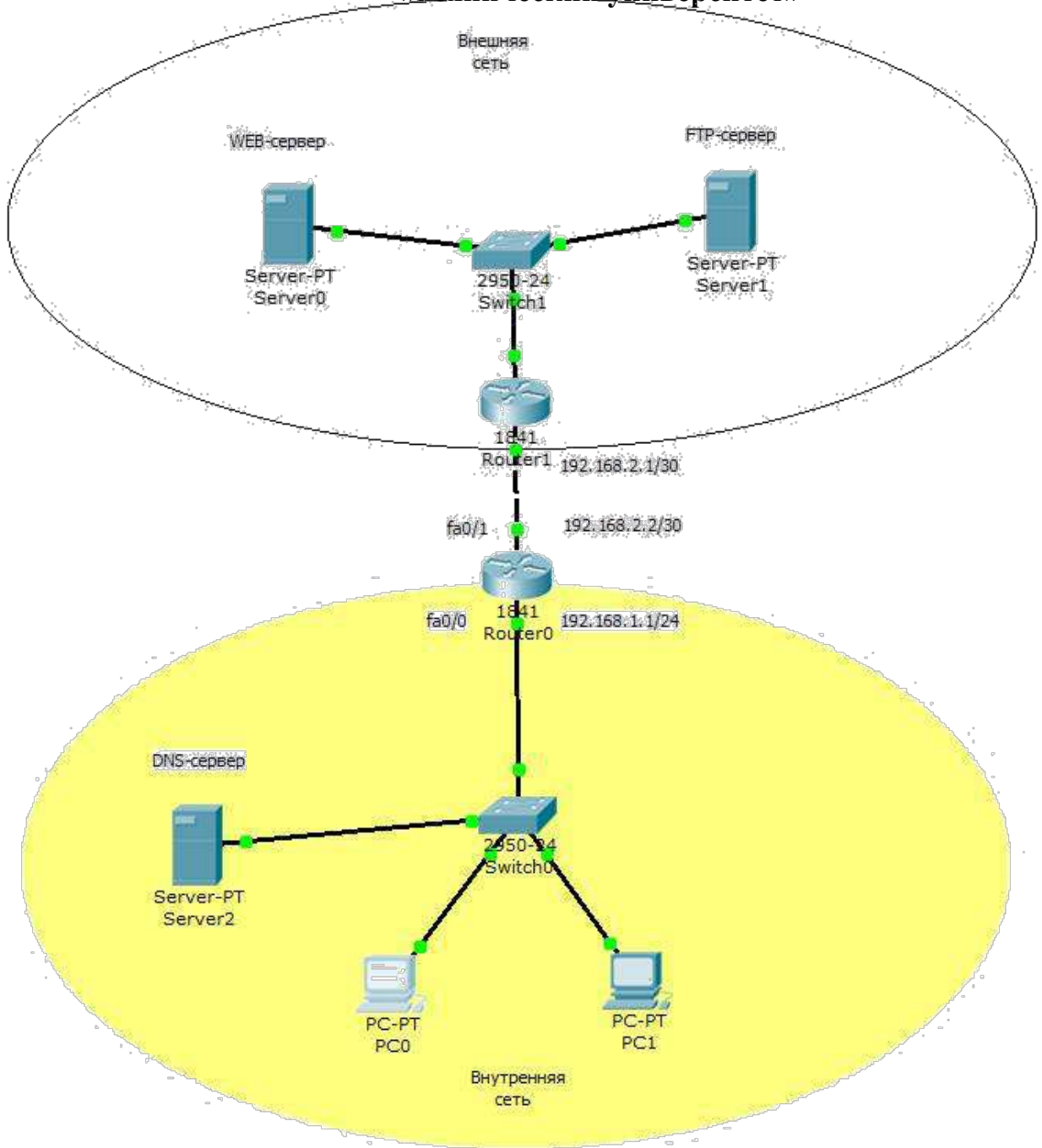


Рисунок 2.1 – Пример сети

```
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).

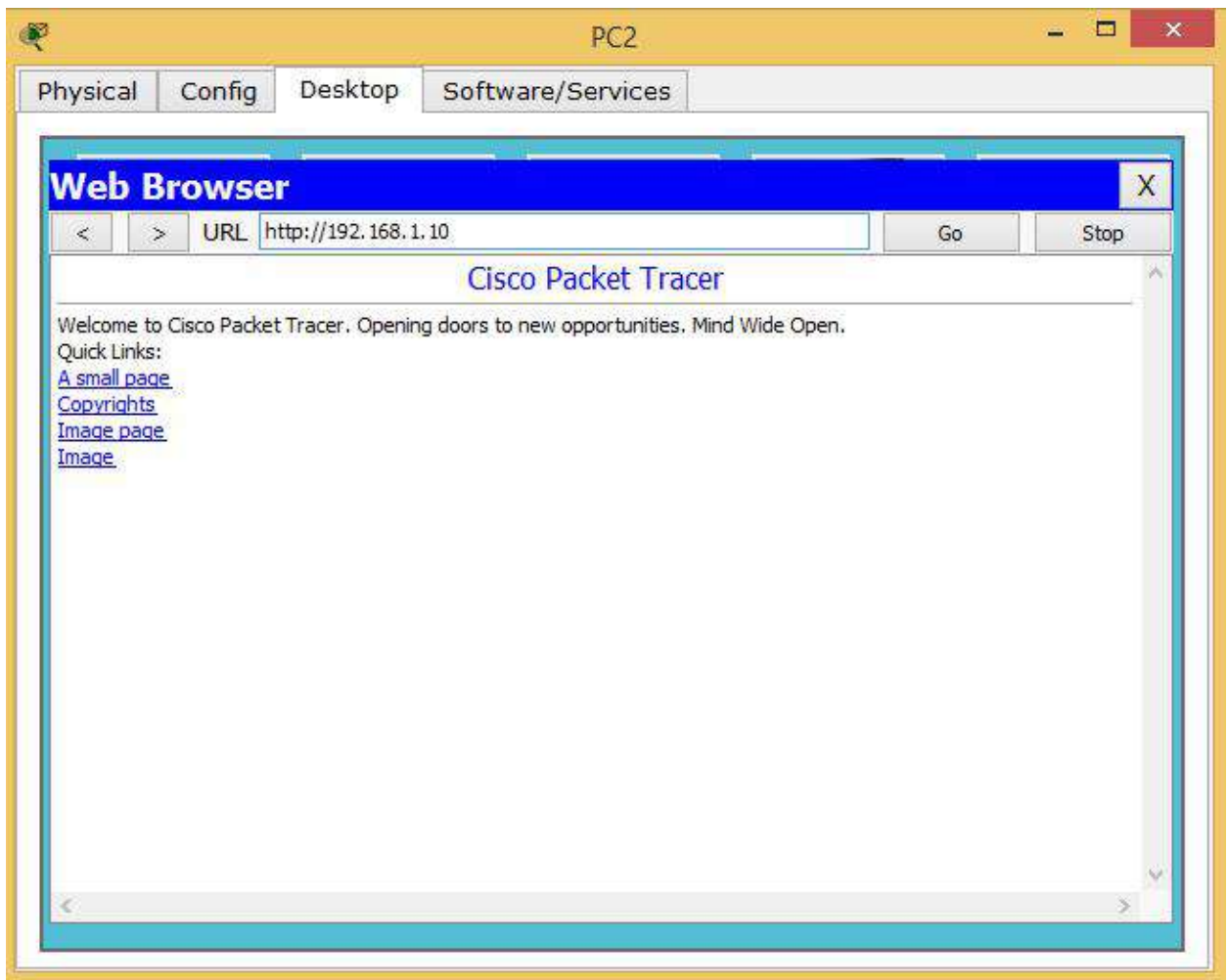


Рисунок 2.3 – Соединение с сервером внутренней сети по протоколу HTTP

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).

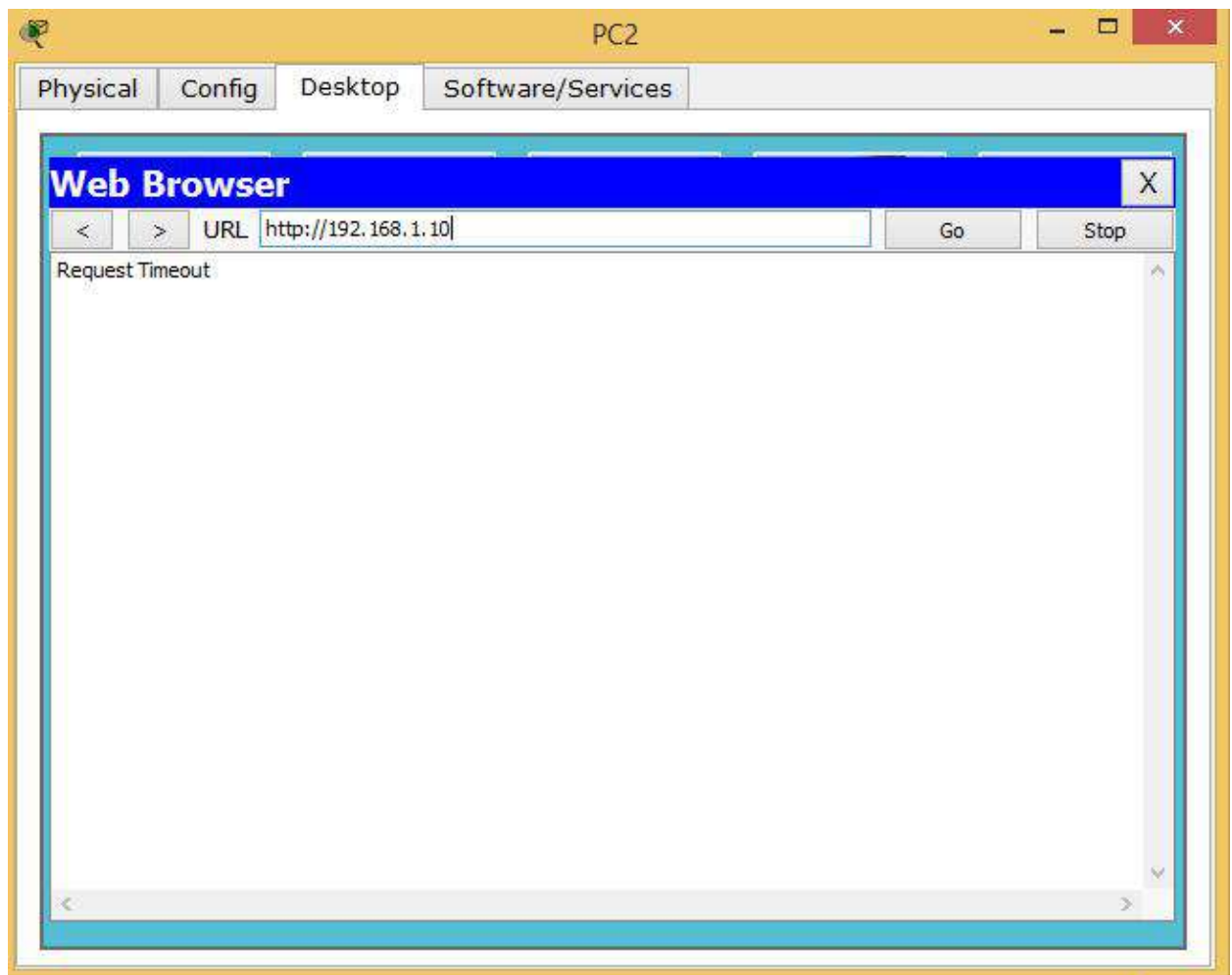


Рисунок 2.5 – Отсутствие соединения с сервером внутренней сети по протоколу HTTP



**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

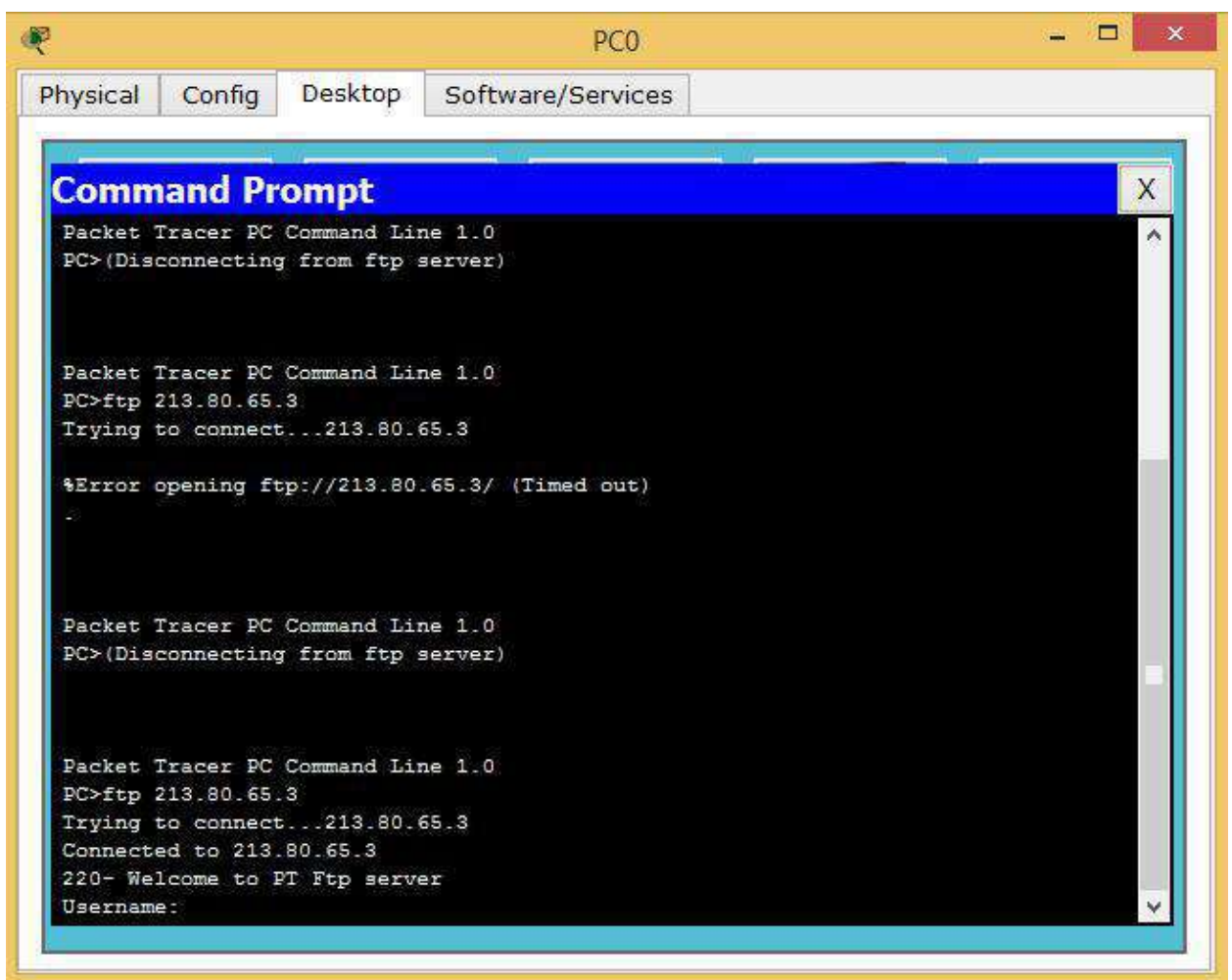


Рисунок 2.8 – Доступ к ftp-серверу после настройки инспектирования TCP-трафика



**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»

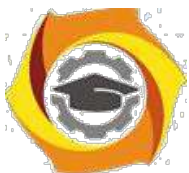
**zone-pair security <имя пары> source <имя зоны> destination <имя зоны>
service-policy type inspect <имя policy-map >**

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»

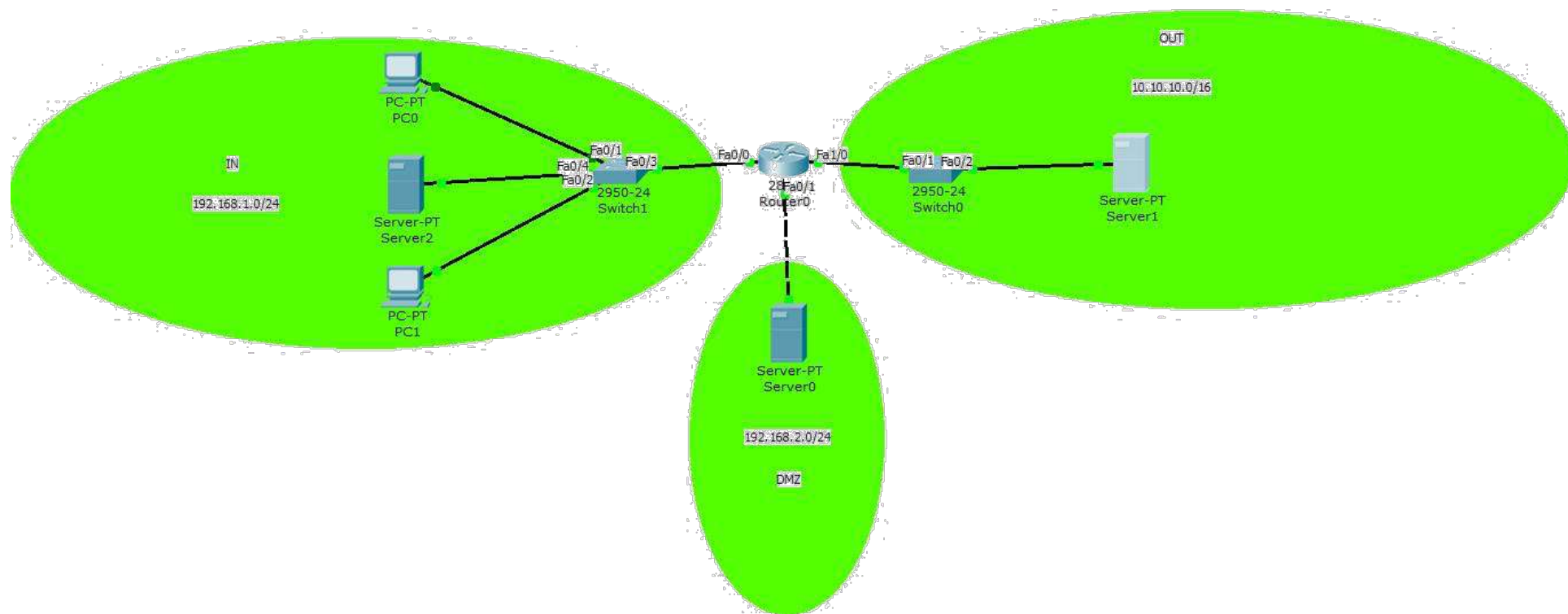


Рисунок 3.1 – Пример сети



**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-mpar-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

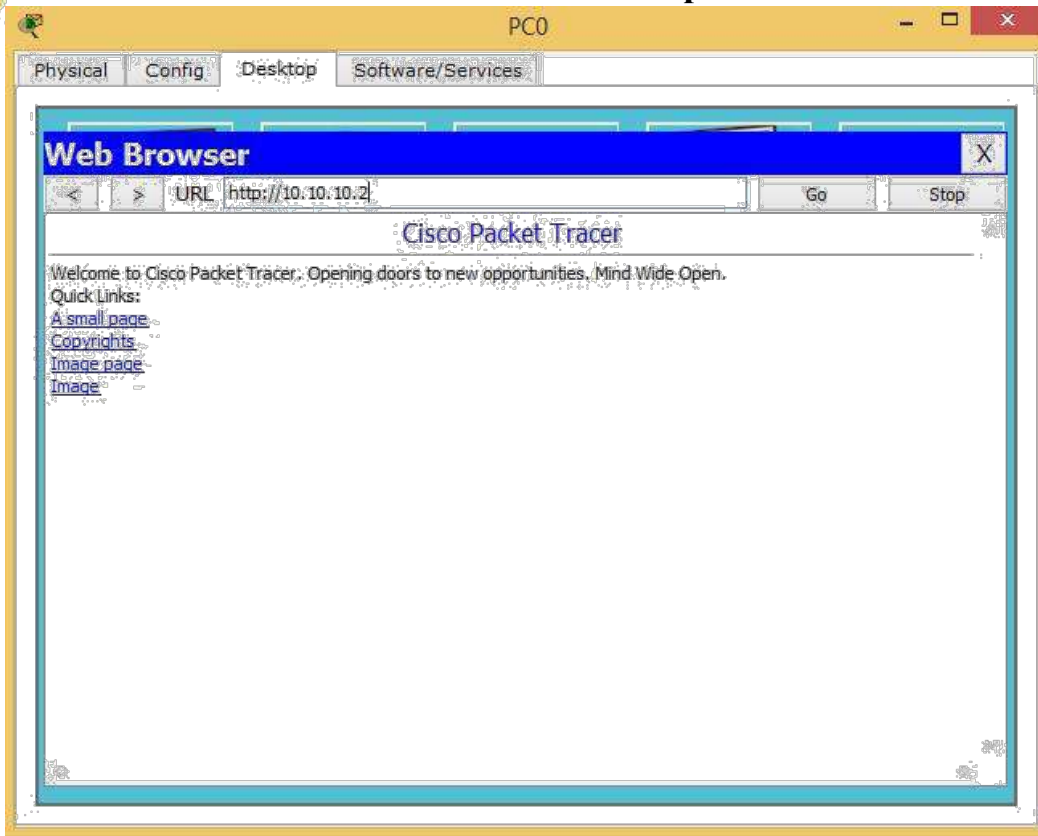


Рисунок 3.2 – Получение ответа при запросе из внутренней сети

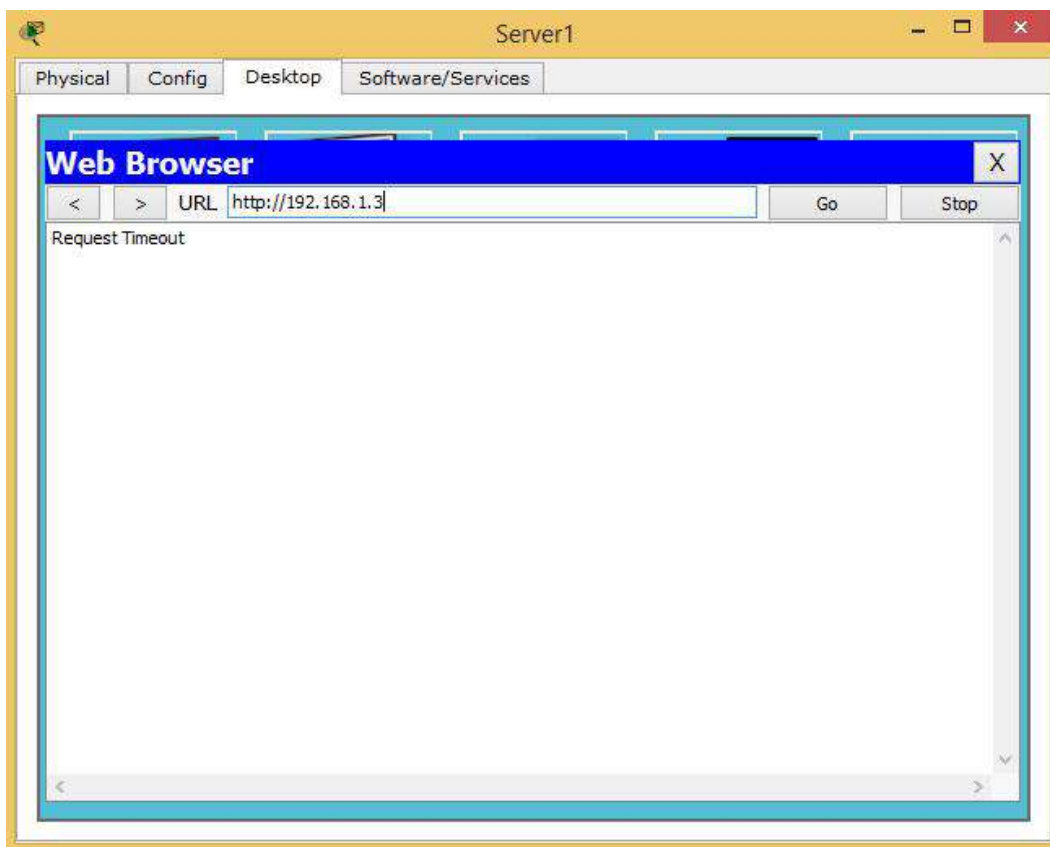


Рисунок 3.3 – Отсутствие ответа при запросе из внешней сети



**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;

– 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»

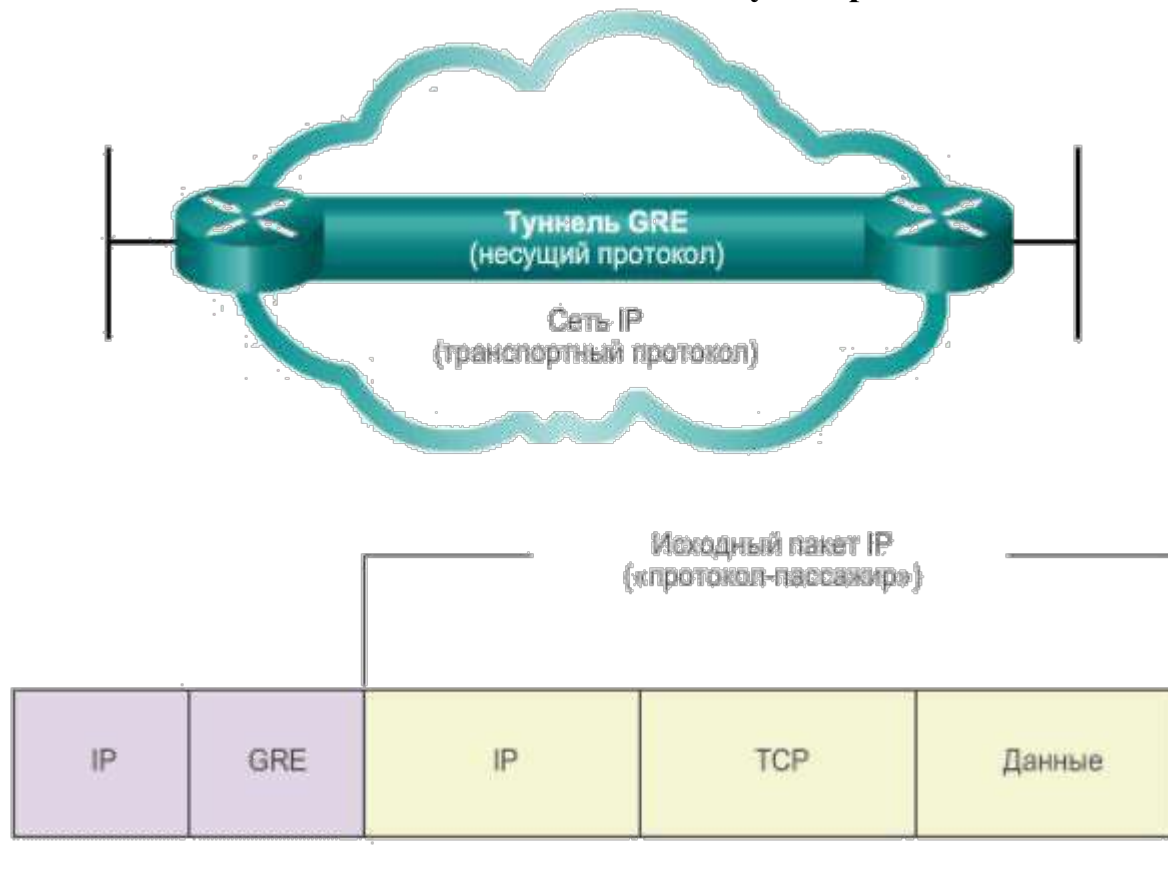


Рисунок 4.1 – Инкапсуляция с использованием GRE

Рассмотрим пример, заимствованный из [8].

Имеется общедоступная сеть, к которой выполнено подключение 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.

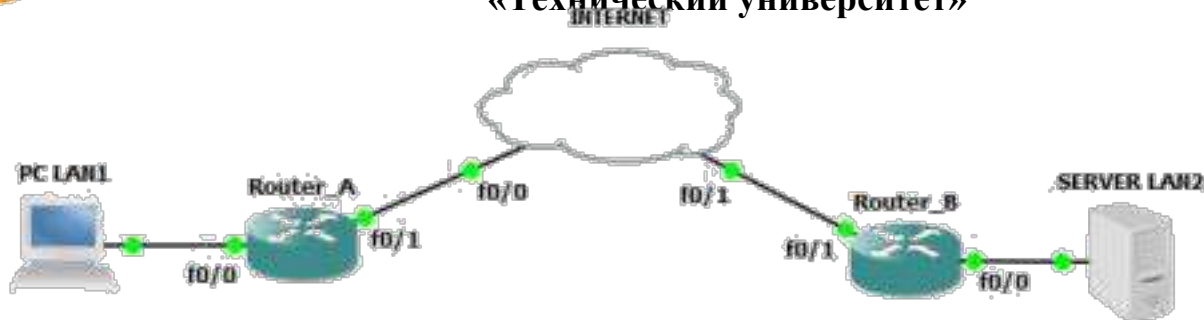


Рисунок 4.2 – Пример объединения двух частных сетей

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.

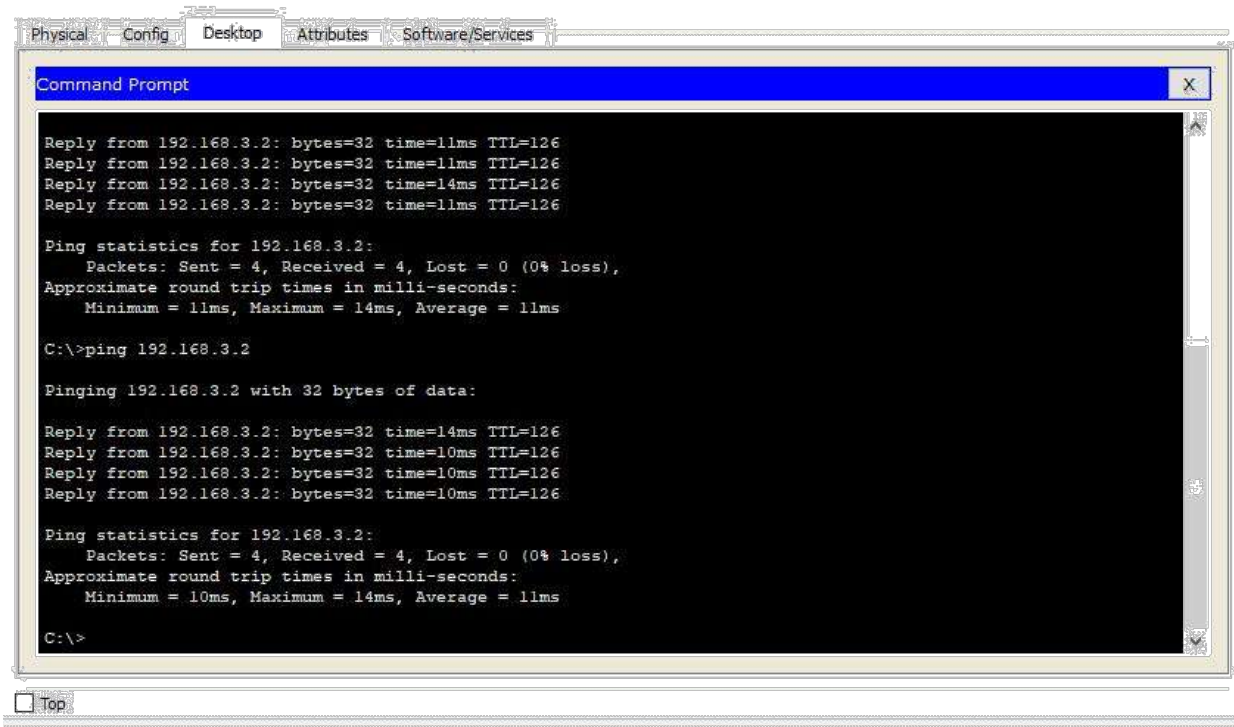


Рисунок 4.3 – Доступ ко второй локальной сети из первой с использованием туннеля

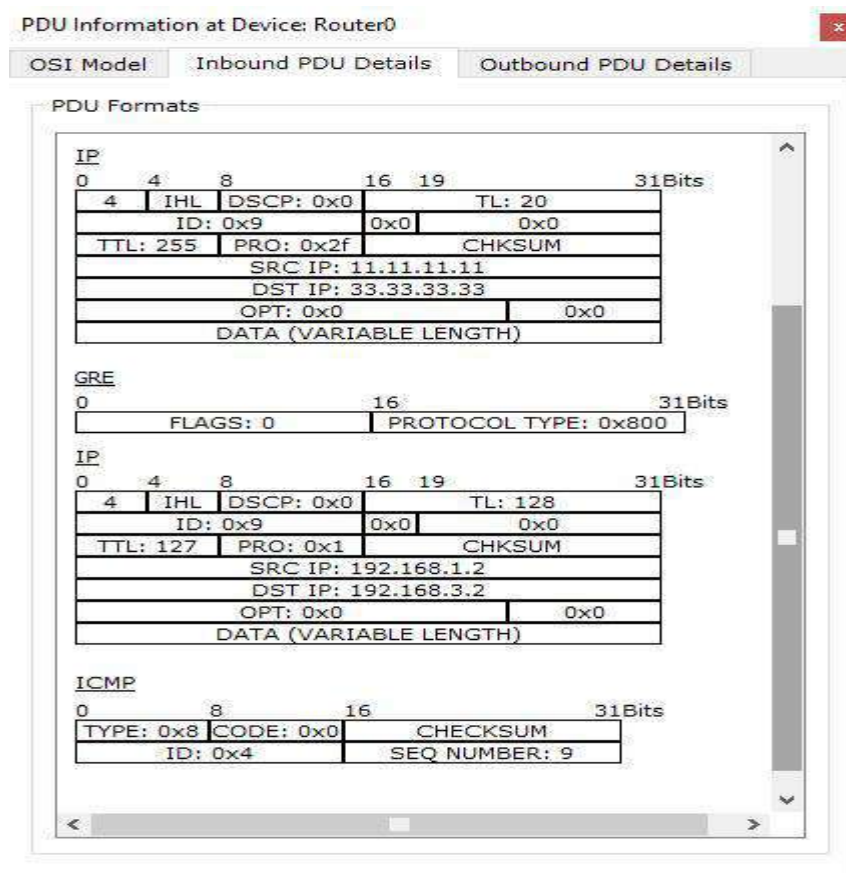


Рисунок 4.4 – Анализ содержимого пакета, передаваемого по туннелю



**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.

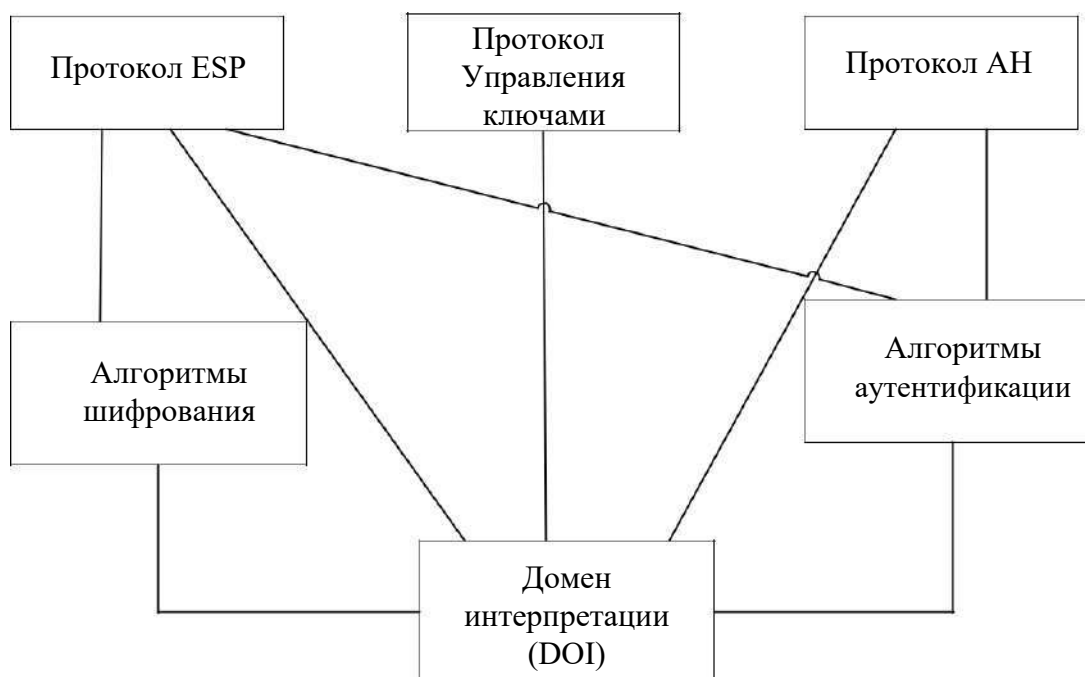


Рисунок 4.6 – Архитектура IPsec

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.

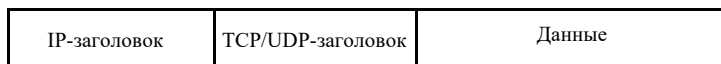
Так же, как и АН, ESP может работать в транспортном и туннельном режимах. Исходный пакет, пакеты после обработки протоколом ESP в транспортном и туннельном режимах показаны на рисунке 4.9. На рисунке серым фоном отмечены поля пакета, подлежащие шифрованию.



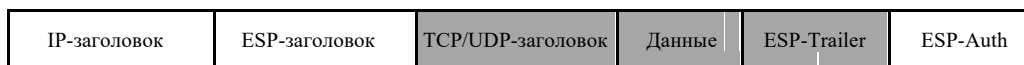
**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**



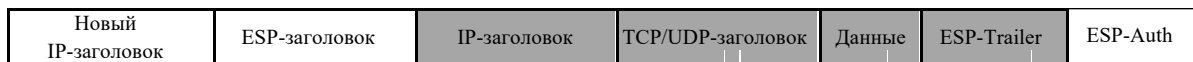
**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**



а) исходный пакет



б) пакет после обработки в транспортном режиме

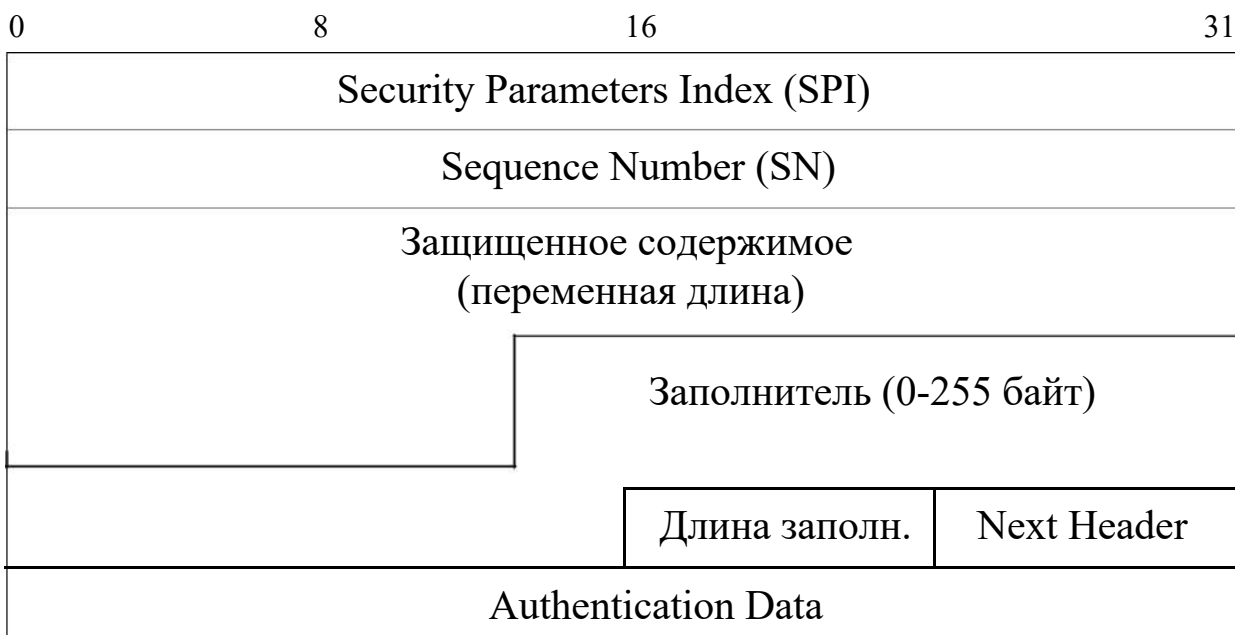


в) пакет после обработки в туннельном режиме

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.

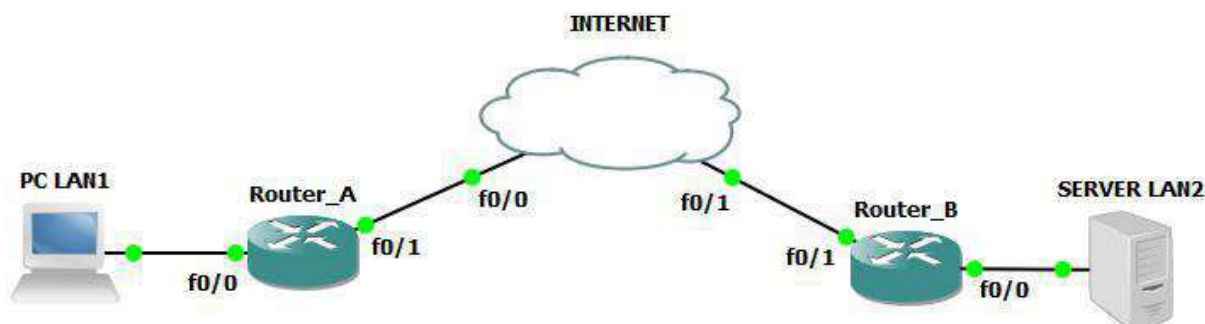


Рисунок 4.11 – Пример объединения двух частных сетей

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 DH [9]:

Алгоритм DH-1 - ключ 768 бит

Алгоритм DH-2 - ключ 1024 бит

Алгоритм DH-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].

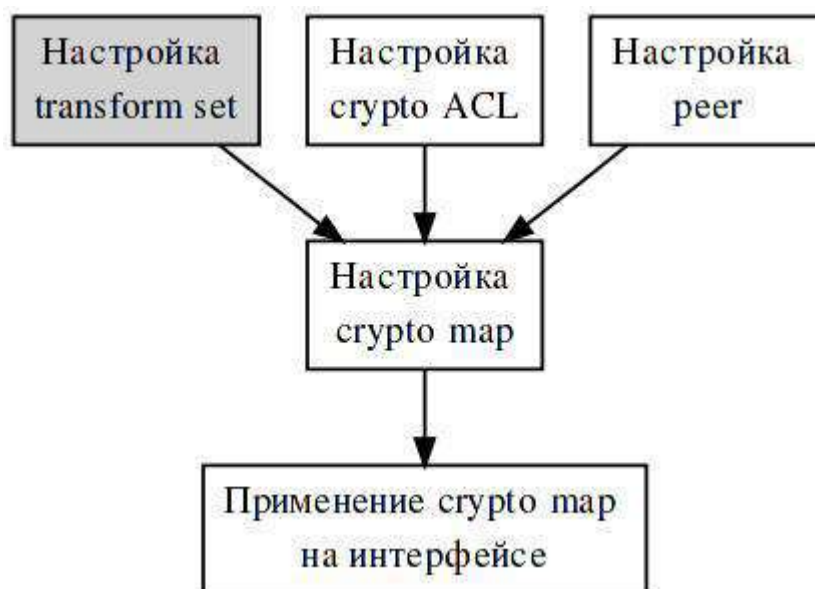


Рисунок 4.12 – Принцип конфигурирования transform-set

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»**

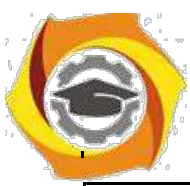
туннелю, используемый ACL, используемый transform-set.
Заключительным этапом является привязка созданной криптокарты на конкретном интерфейсе.

**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.

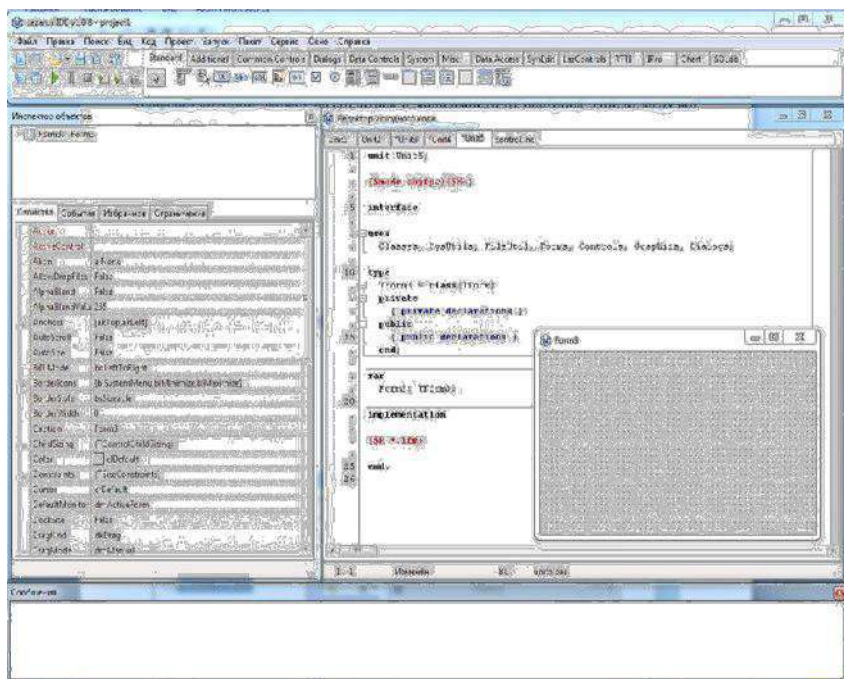


Рис. 1.1. Основные окна среды программирования Lazarus
Главное окно Lazarus содержит:

15. главное меню;
16. палитру компонентов.



Главное меню содержит обширный набор команд для доступа к функциям Lazarus.

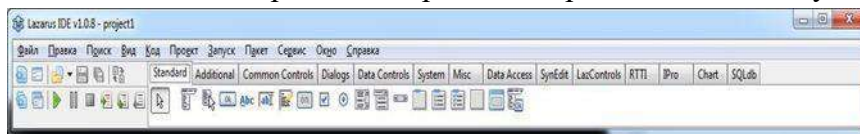


Рис. 1.2. Главное меню среды программирования Lazarus

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.

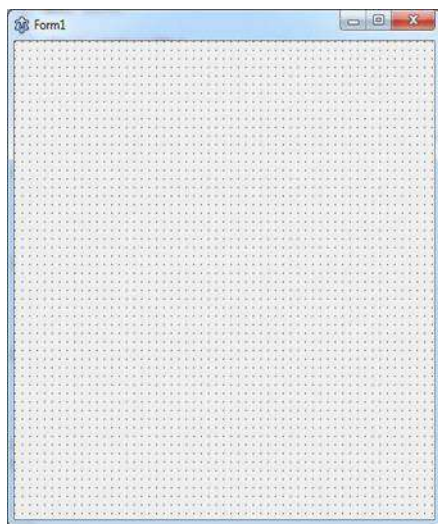


Рис. 1.3. Окно формы (или Конструктора формы)

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.

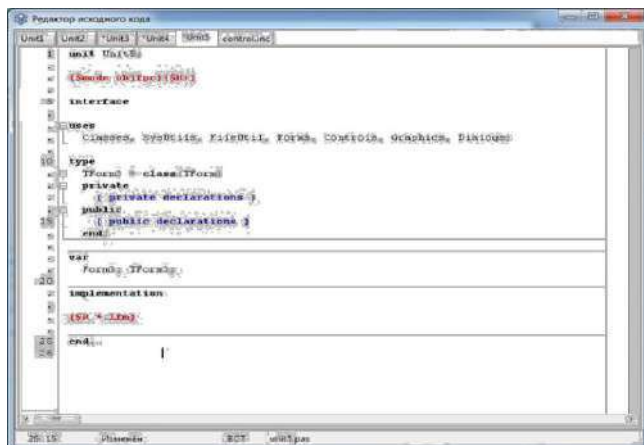


Рис. 1.4. Окно Редактора кода

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.

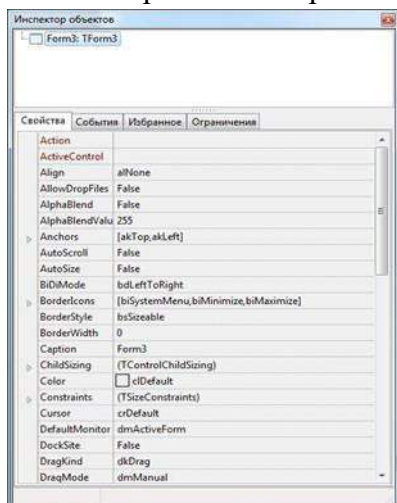


Рис. 1.5. Окно Инспектора объектов

Главное меню программы – имеет древовидную структуру и содержит доступ ко всем элементам программы от «Файл» до «Справка».

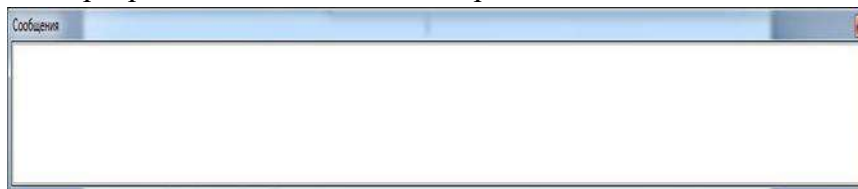


Рис. 1.6. Окно ошибок и подсказок

Управление проектом: назначение кнопок управление проектом (рис.1.7).

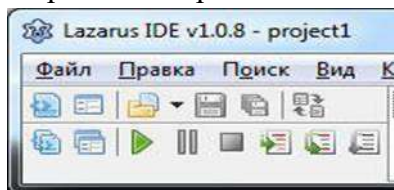
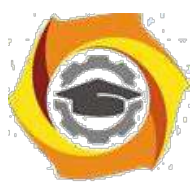


Рис. 1.7. Основные кнопки управления проектом



**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 вкладок.

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).

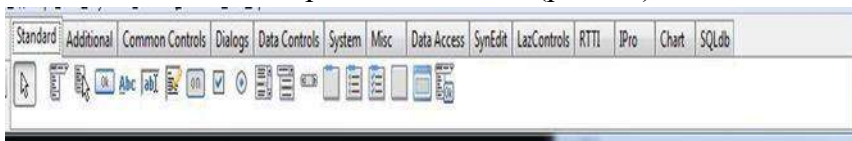


Рис. 1.8. Палитра Компонент

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. 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:

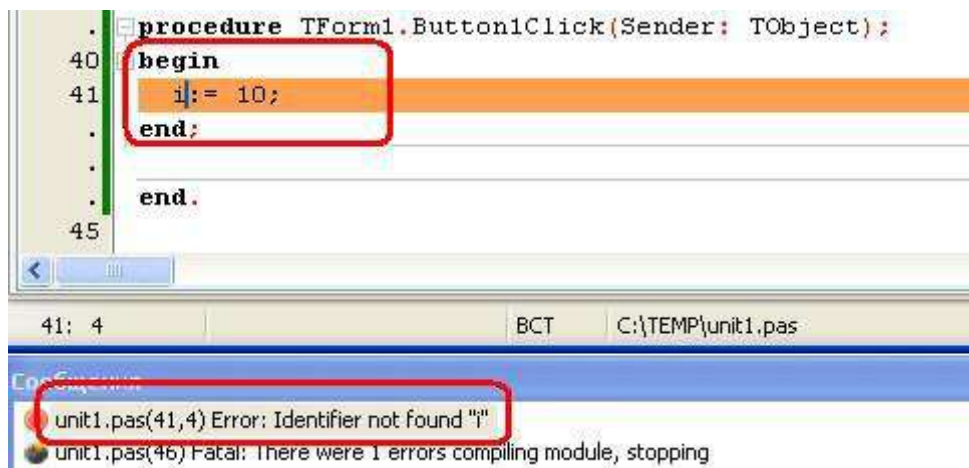


Рис. 2.1. Найденная компилятором синтаксическая ошибка - нет объявления переменной i



**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:

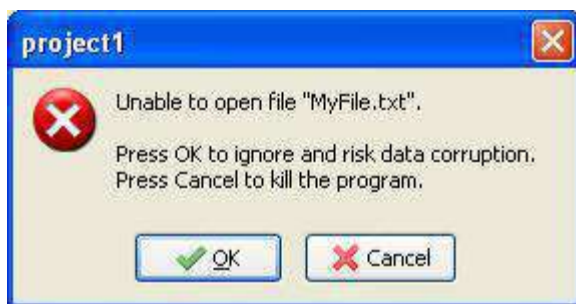


Рис. 2.2. Сообщение Lazarus об ошибке времени выполнения

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 **LP2**. Имена проекта, формы, модуля и кнопки изменять не нужно, оставьте имена, данные по умолчанию.

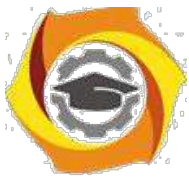
Далее, сгенерируйте событие OnClick для кнопки, в котором напишите следующий код:

```
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:

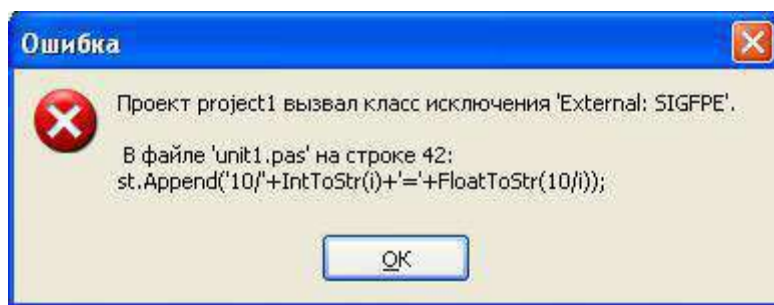


Рис. 2.3. Сообщение Lazarus об ошибке

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));
```

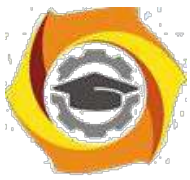
Ну что, ничего криминального тут нет, почему же произошла ошибка? Код верный и должен был нормально выполняться... Когда вы заходите в подобный *тупик*, помочь вам может *здравый смысл* и встроенный отладчик. *Здравый смысл* говорит, что ошибка произошла где-то в теле *цикла* `for`. А чтобы воспользоваться 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":

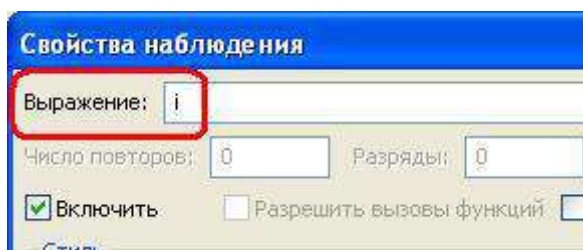


Рис. 2.6. Установка наблюдения за переменной

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:

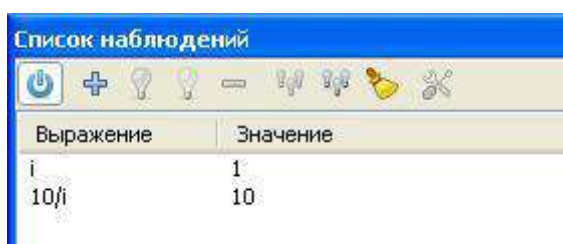




Рис. 2.7. Окно Списка наблюдений

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:

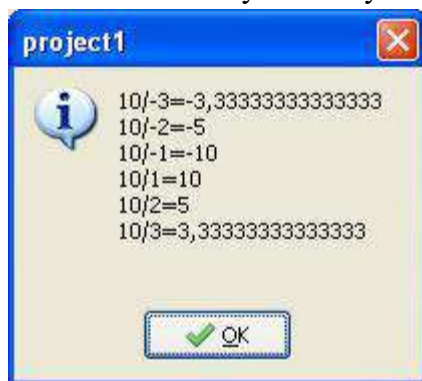


Рис. 2.8. Результирующее сообщение программы

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:

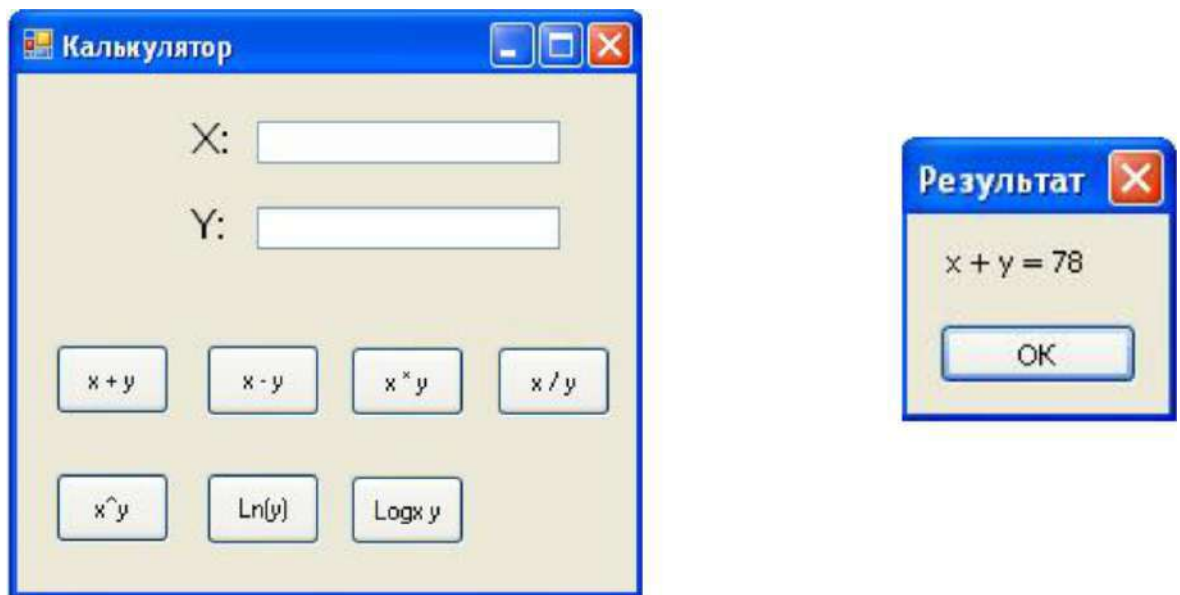


Рис. 3. 1



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:

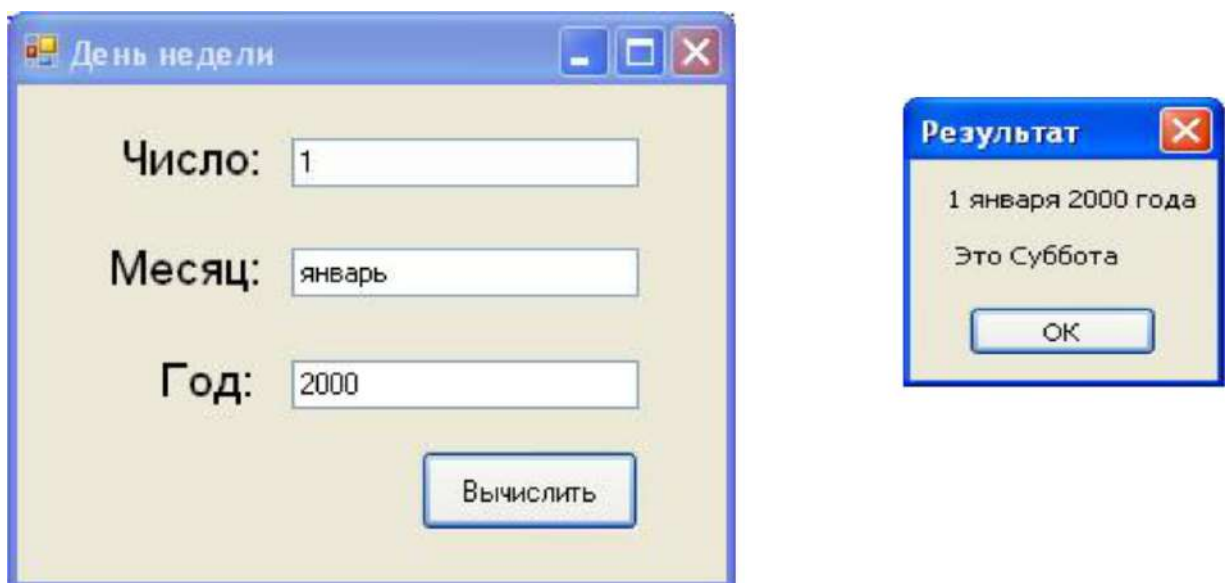


Рис. 3. 2



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:

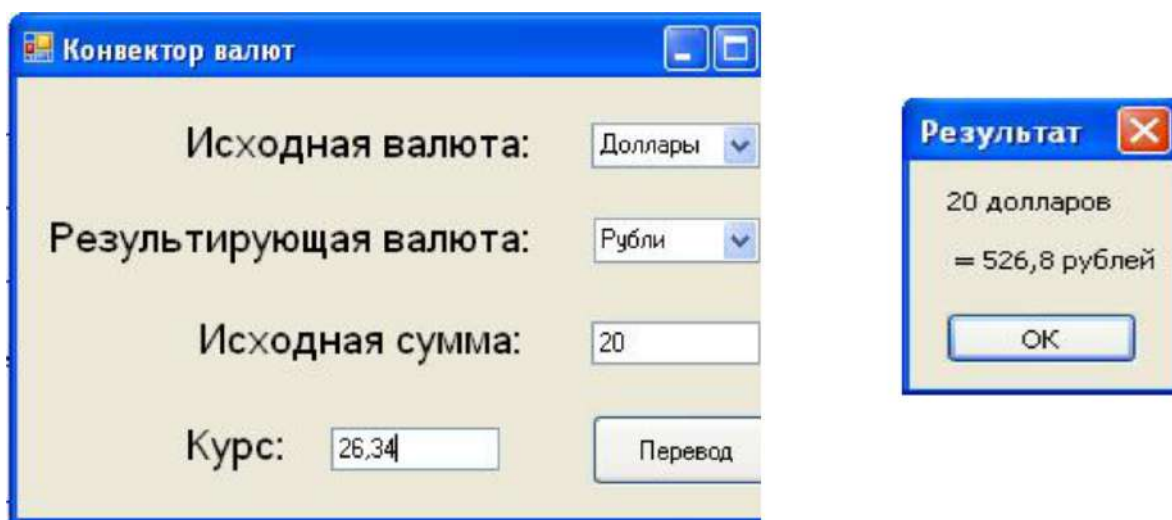


Рис. 3. 3



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 видны: 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:

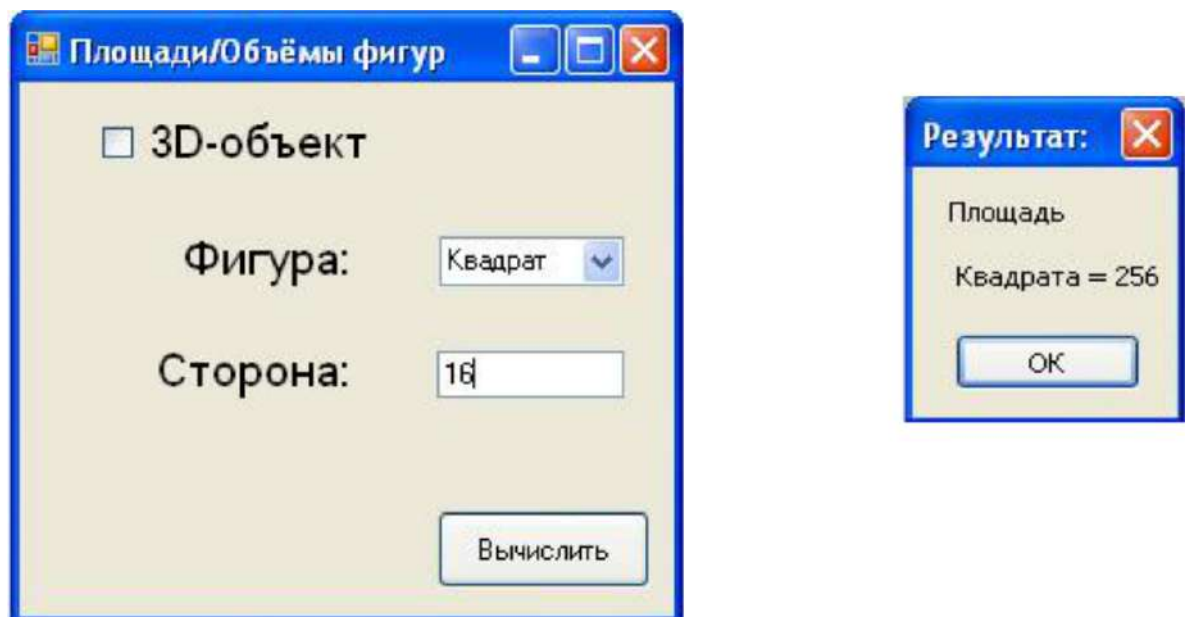


Рис. 3. 4



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:

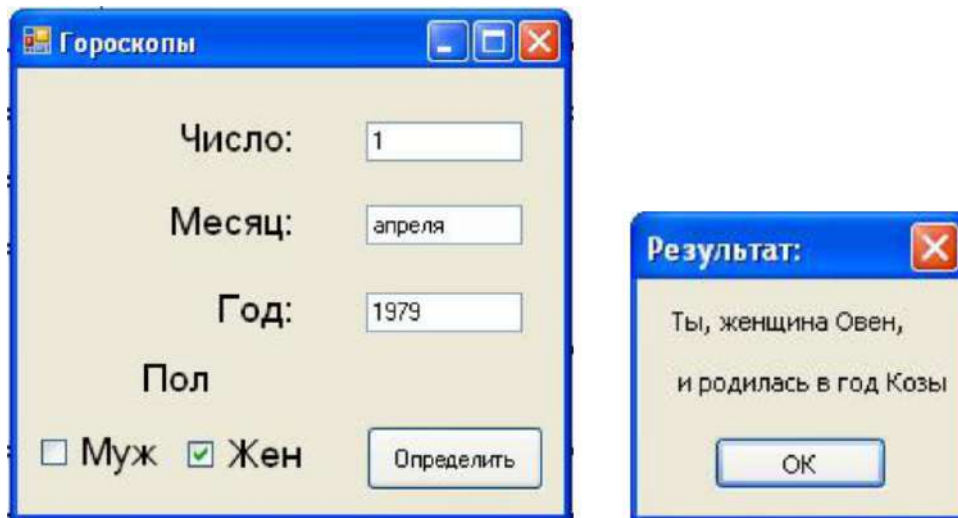


Рис. 3. 5

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, (\$)	Неиспользуемые 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:

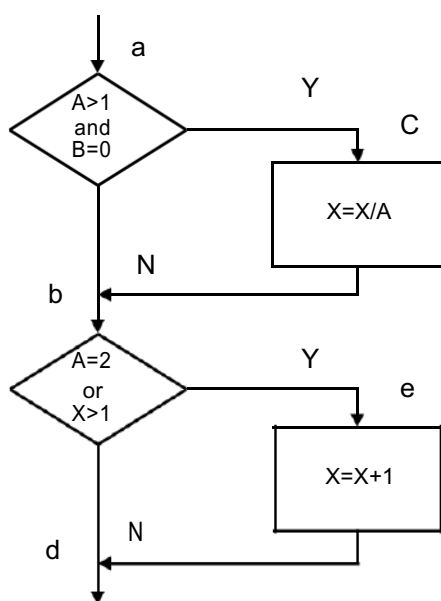


Рисунок 5.1

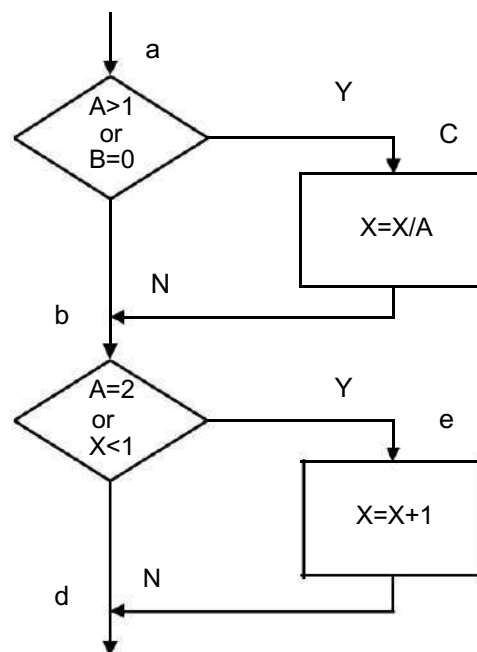


Рисунок 5.2



**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 NILP (тест $A = 2, B = 0, X = 4$), NIMKT (тест $A = 3, B = 1, X = 0$), NJKT (тест $A = 0, B = 0, X = 0$), NJKR (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.

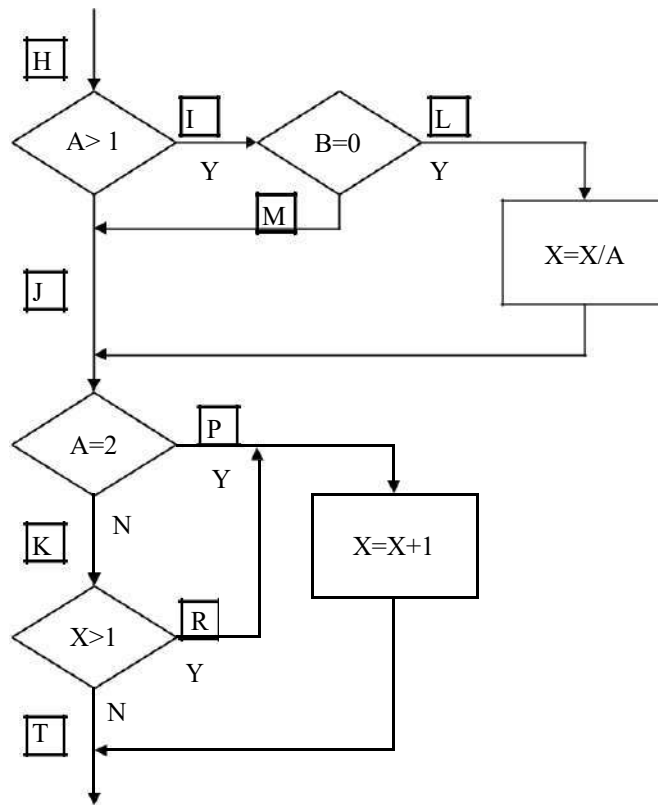


Рисунок 5.3

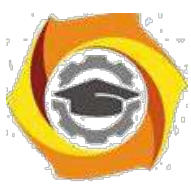
2.5 Метод комбинаторного покрытия условий.

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};



- $A=2, B=1, X=1$ {покрывает б, е};
- $A=0,5, B=0, X=2$ {покрывает в, ж};
- $A=1, B=0, X=1$ {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
$A=0,5, B=0, X=2$	$X=3$	$X=4$	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:

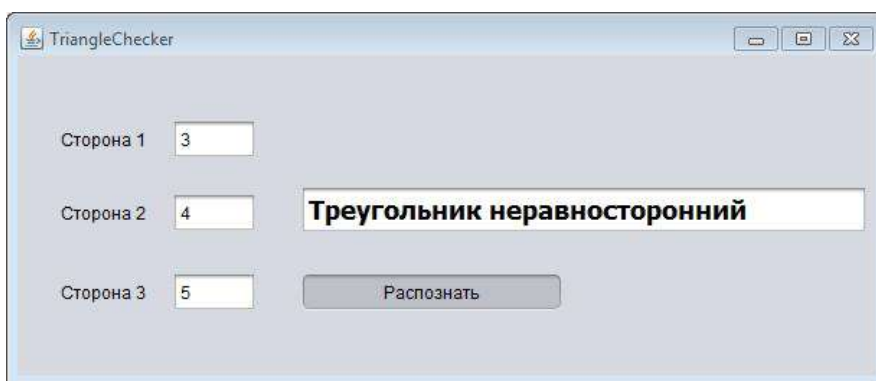


Рис. 7. 3

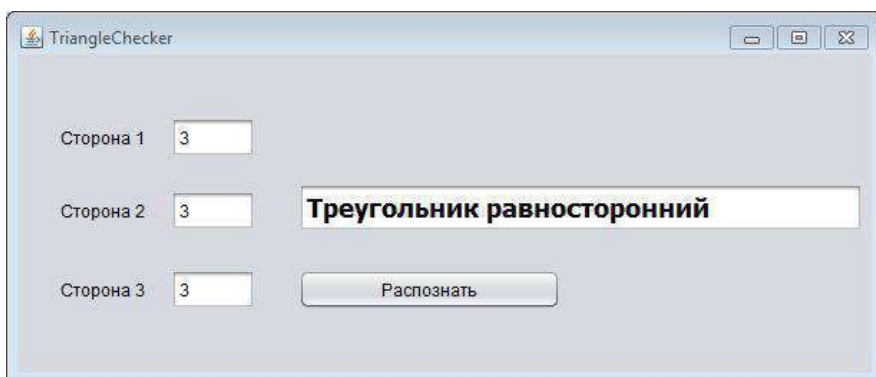


Рис. 7.1

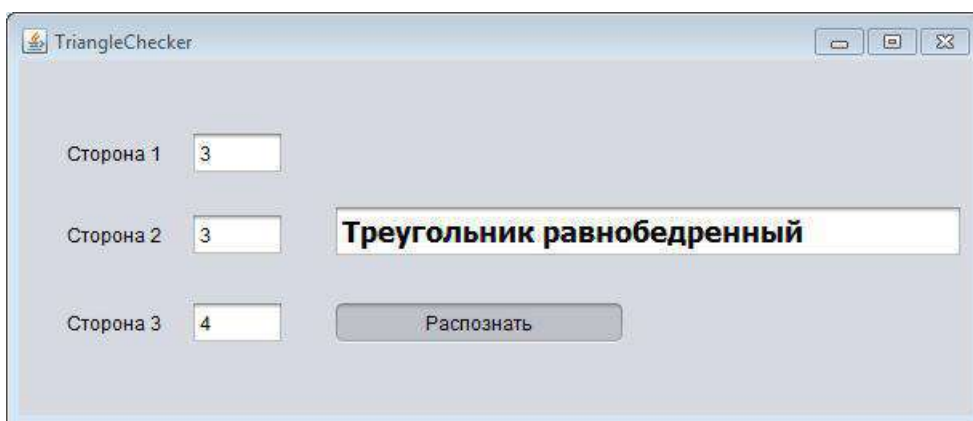


Рис. 7.2

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:

7. некорректность или отсутствие функций;
8. ошибки интерфейса;
9. ошибки в структурах данных;
10. ошибки машинных характеристик (нехватка памяти и др.);
11. ошибки инициализации и завершения.

Техника тестирования ориентирована:



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

8. на сокращение необходимого количества тестовых вариантов;
9. 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 операторов¹.

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:

9. некоторое конкретное значение; определяется один допустимый и два недопустимых класса эквивалентности: заданное значение, множество значений меньше заданного, множество значений больше заданного;
10. 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;
11. 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:

10. 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;

¹См. 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

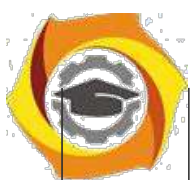
- ' Что такое тестирование ПС?
- ' Чем тестирование отличается от отладки ПС?
- ' Для чего проводится функциональное тестирование?
- ' Что такое комплексное тестирование?
- ' Каковы правила тестирования программы «как черного ящика»?
- ' Как проводится тестирования программы по принципу «белого ящика»?
- ' 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
6.	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:

2. Ознакомиться с причинами стрессовых ситуаций в профессиональной деятельности.
3. Методика определения стрессоустойчивости и социальной адаптации Холмса и Раге.
4. Методика «Инвентаризация симптомов стресса»
5. Обсуждение результатов и формулирование выводов
6. 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.

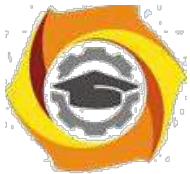
-Модель «дуга – работа» (activity-on-arrow)

-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:

3. Задачи кадрового менеджмента на современном этапе.
4. Универсальные функции управления персоналом: планирование, организация, координация, регулирование, контроль.
5. 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

3. 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.

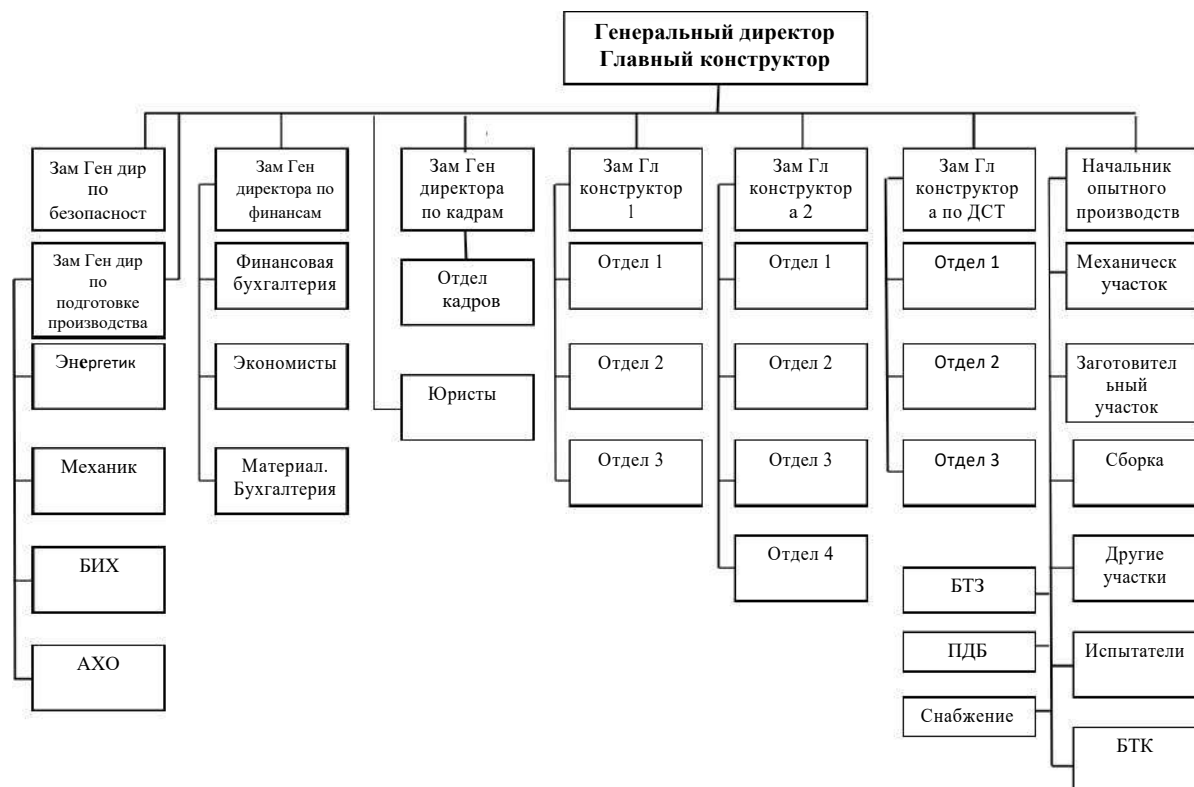


Схема 1- Организационная структура АО «УКБТМ»

Кадровая политика и кадровые процедуры производственной организации.

Устные вопросы по теме практического занятия:

3. Основные группы кадровых процедур: приемы на работу; переводы на другую работу; увольнения работников; предоставление отпусков работникам; привлечение к ответственности и др.

4. Общероссийский классификатор управленческой документации (ОКУД) ОК 011-93.

5. 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:

16. Для чего предназначены унифицированные формы первичной учетной документации по учету труда и его оплаты?

17. Можно ли вносить изменения в унифицированные формы, и какие именно изменения?



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

1. Что такое приказ по личному составу?
2. Какие процедуры движения кадров документируют приказами по личному составу?
3. Для чего составляется штатное расписание? Что оно включает в себя?
4. Какие документы нужны, чтобы составить штатное расписание?
5. 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:

1. Унифицированная форма N Т-1. Утверждена Постановлением Госкомстата России от 05.01.2004 №1.
2. Унифицированная форма № Т-5. Утверждена постановлением Госкомстата России от 05.01.2004 № 1.
3. Унифицированная форма № Т-5а. Утверждена постановлением Госкомстата России от 05.01.2004 № 1.
4. Унифицированная форма № Т-6. Утверждена постановлением Госкомстата России от 05.01.2004 № 1.
5. Унифицированная форма № Т-7. Утверждена постановлением Госкомстата России от 05.01.2004 № 1.

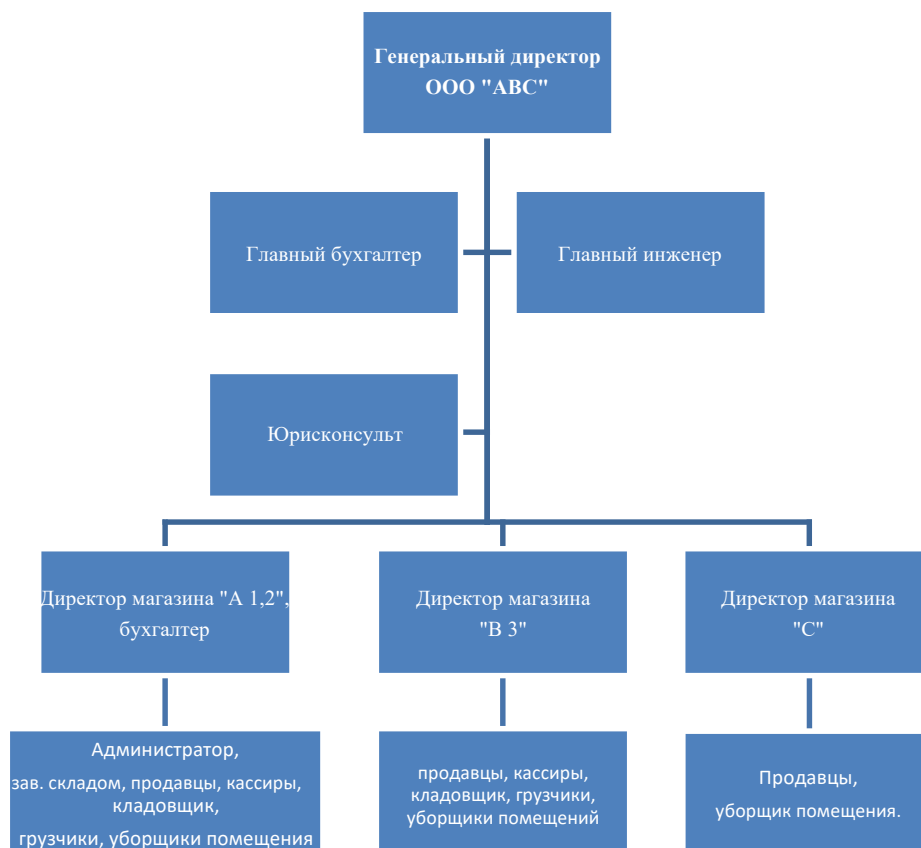


Схема 2 - Организационная структура ООО «ABC».

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», 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.

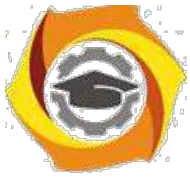
Содержание практического занятия

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:

- 1) 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.
- 2) 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).

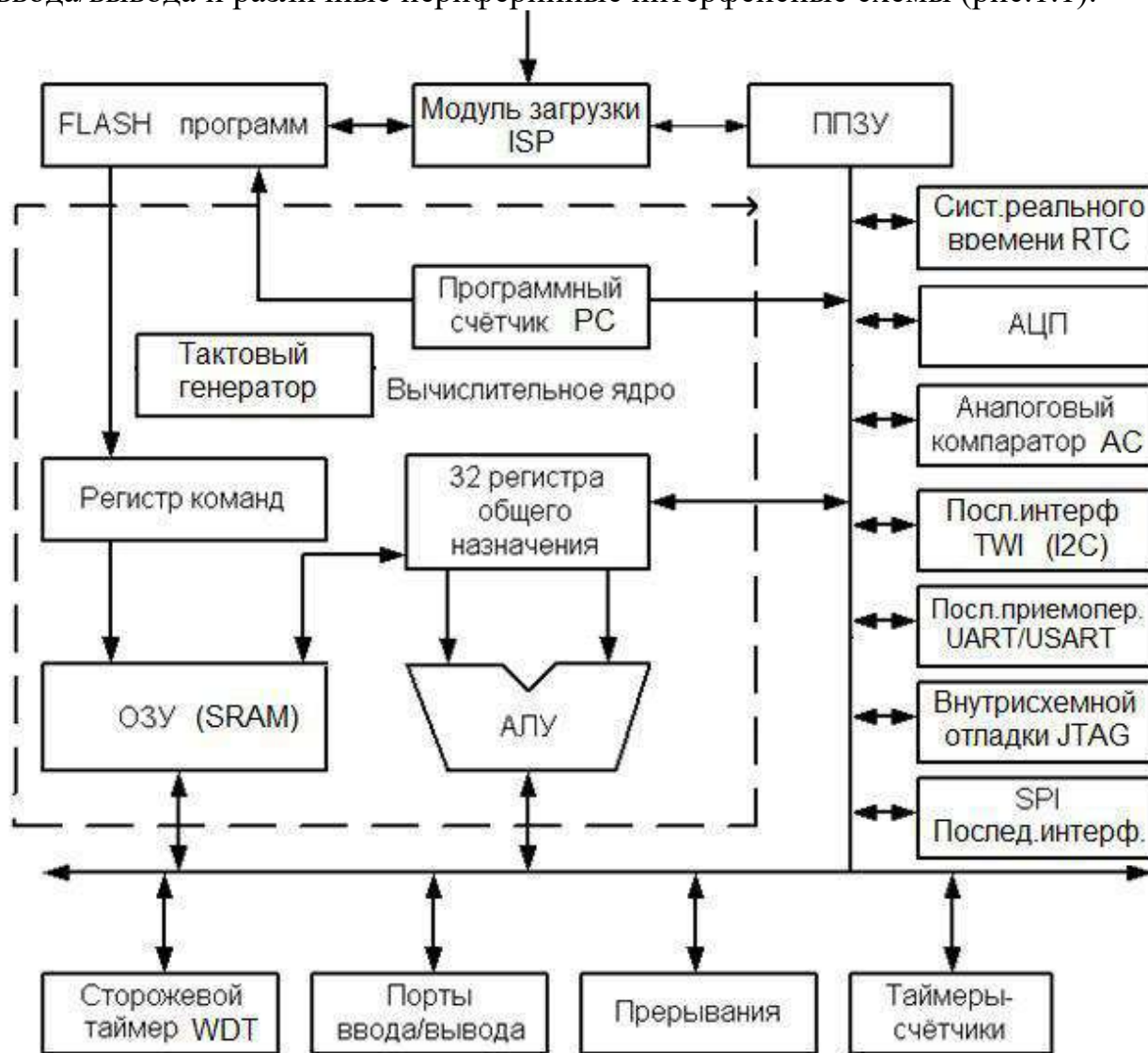


Рис.1.1. Структурная схема МК AVR

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 записывается 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.

в 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.

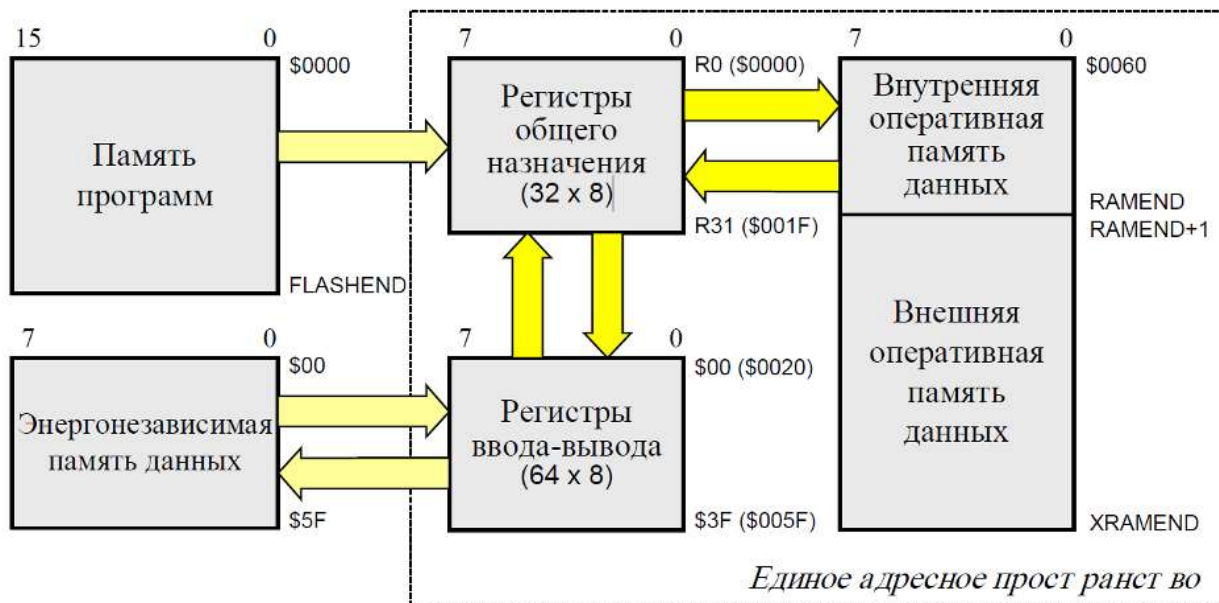


Рис.1.2. Программная модель AVR контроллера

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

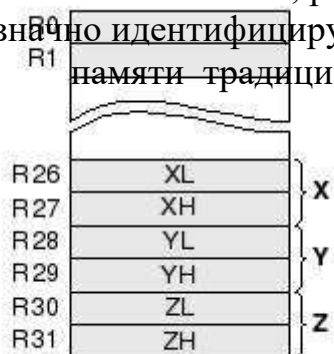


Рис.1.3. Регистры микроконтроллера AVR

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»

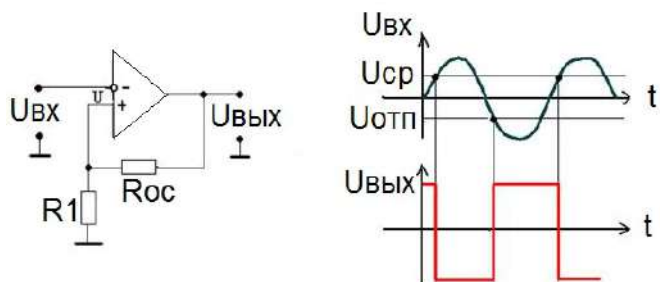


Рис.2.1. Диаграммы работы
триггера Шмитта



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 **5** 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 ссылаться, 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.

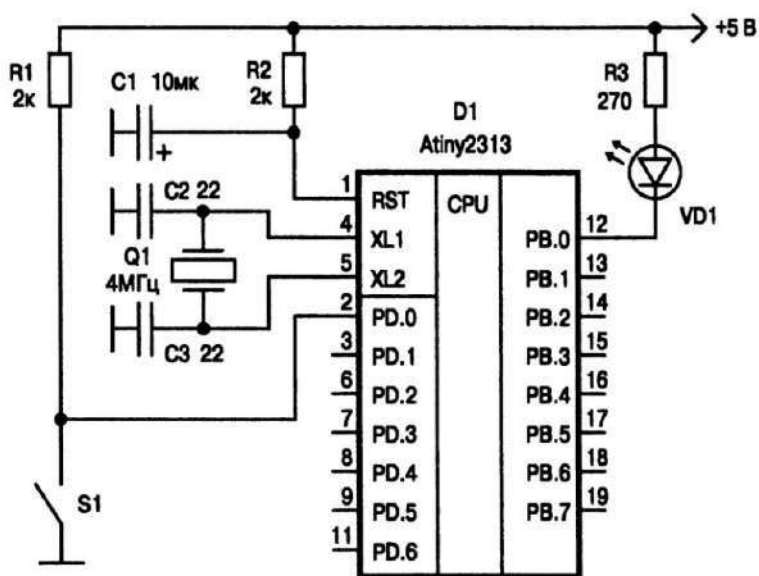


Рис.2.3. Схема подключения светодиода и кнопки

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.

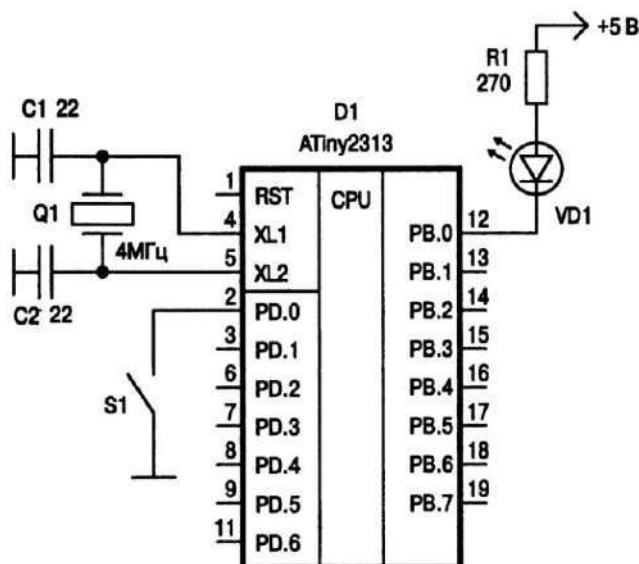


Рис.2.4. Упрощенная схема подключения кнопки и светодиода



```
#####  
##          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 BB 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; INT01 – 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.

2. 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;

3. Проверить инициализацию стека без использования явных команд инициализации.



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

4. Промоделировать работу программы в среде Proteus. Зафиксировать использованные элементы схемы.



7. 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

- 5) Составить схему алгоритма.
- 6) Разработать программу и добиться правильной компиляции.
- 7) Проверить работоспособность программы на эмуляторе AVR Studio.
- 8) 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.

5) отчете представить:

1. Схему алгоритма задачи;
2. Текст программы на языке программирования (только основной



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

цикл);



5. 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:

8. Схему алгоритма задачи;
9. Текст программы на языке программирования (только основного цикла);
10. 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
- 3) выключить светодиод;
- Перейти к началу основного цикла (п.1).

Алгоритм мигания:

3. Зажечь светодиод;
4. Выдержать паузу;
5. Потушить светодиод;



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

6. Выдержать паузу;



3 Перейти к п.1.

В отчете представить:

4. Схему алгоритма задачи;
5. Текст программы на языке программирования (с указанием только основного блока);
6. 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:

6. Схему алгоритма задачи;
7. Текст программы на заданном языке программирования (с указанием только основного блока);
8. 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 ?
6. Назначение и использование регистров X, Y и Z.
7. Как осуществляется инициализация стека на языке ассемблера и языке C?



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

8. 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()` ?



7. УПРАВЛЕНИЕ СЧЕТЧИКАМИ МИКРОКОНТРОЛЛЕРА

Цель:

- 1) Углубить и закрепить теоретические знания по принципам построения и работы специализированных контроллеров;
- 2) Приобрести практические навыки самостоятельного управления счетчиками контроллера;
- 3) 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

- 3) Белов А.В. Создаем устройства на микроконтроллерах. - СПб.: Наука и техника, 2007. - 304 с.
- 4) Евстифеев А.В. Микроконтроллеры AVR семейства Tiny. Руководство пользователя. - М.: Издательский дом Додэка-XXI, 2007. - 432с.
- 5) Ревич Ю.В. Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. СПб.: БХВ-Петербург, 2011. - 352с.
- 6) 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

6. Как работает счетчик? Какие режимы реализует счетчик?
7. Какие существуют методы построения делителей?
8. Какие сигналы выдает схема сравнения?
9. На что влияет разрядность счетчика?



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

10. 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:

4. Программную модель счетчиков с указанием всех портов и их адресов;
6. Структуру регистра управления и таблицы задания режимов работы счетчиков;
7. 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:

6. измерение и задание временных интервалов;
 7. подсчет количества внешних событий (импульсов);
 8. формирование одиночных и периодических импульсов с постоянными и переменными параметрами (частота, длительность импульса, количество импульсов, фазы);
 9. преобразование временного интервала в код для АЦП;
 10. фиксация состояния счетчика по внешнему сигналу (захват) и т.д.
8. микроконтроллерах бывает до десяти счетчиков. В ATtiny2313 имеется два счетчика: однобайтный T0 и двухбайтный T1.



Программная модель счетчиков AVR

3. 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.

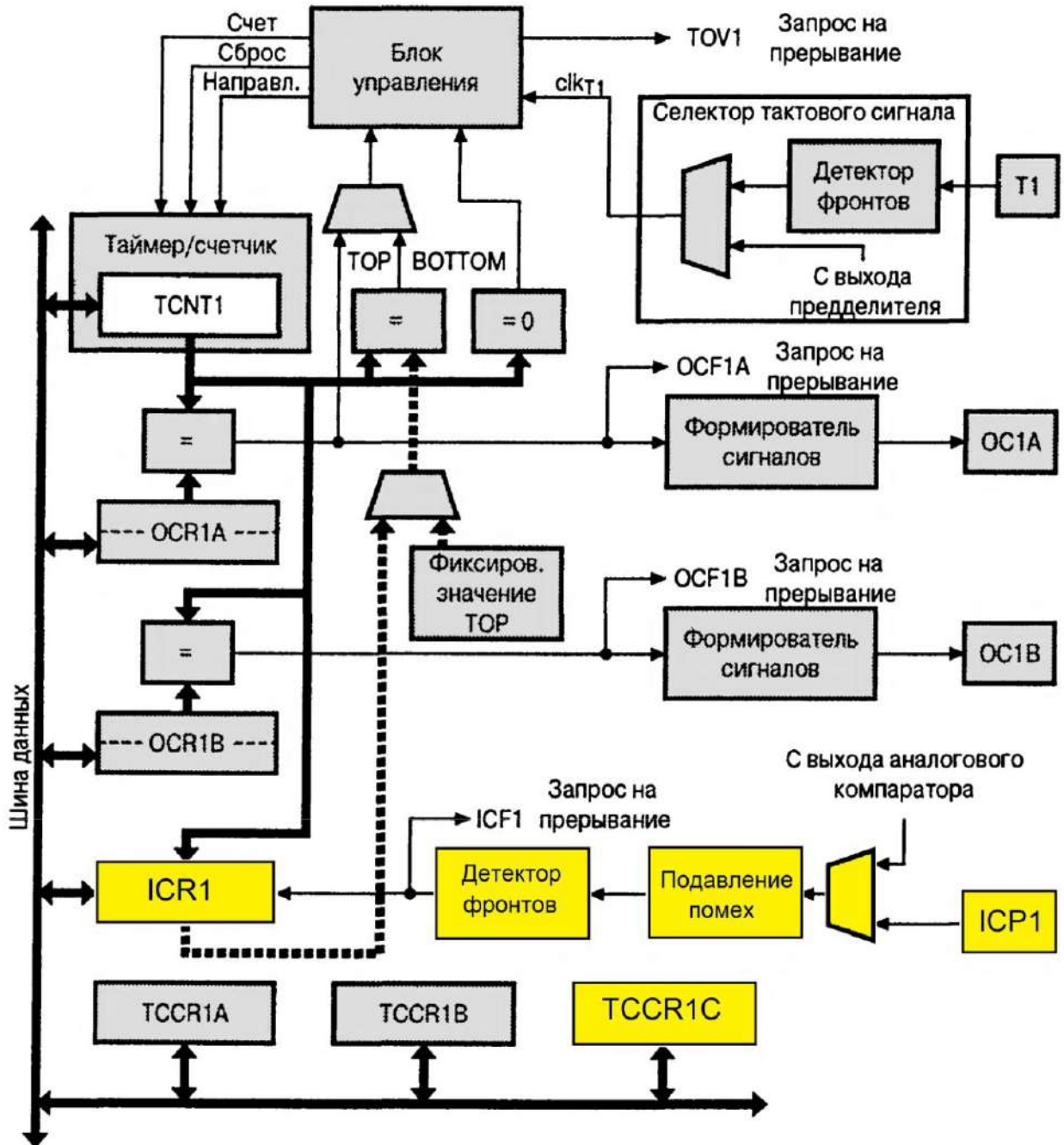


Рис.3.1. Структурная схема счетчика T1 ATtiny2313



**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)

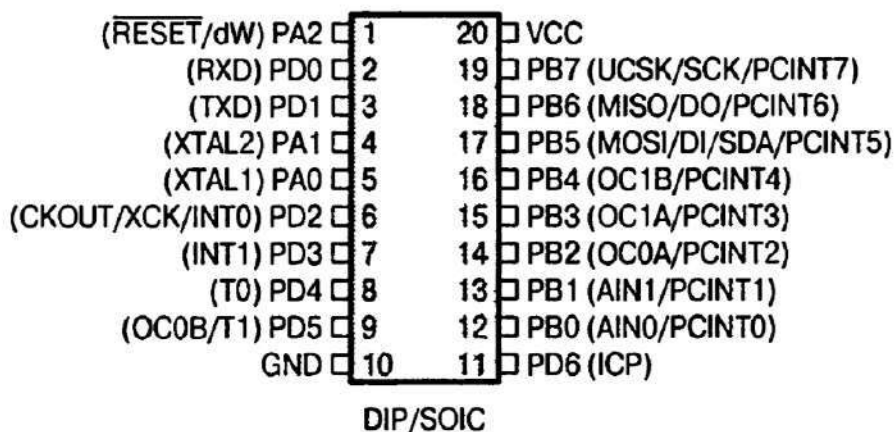


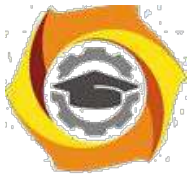
Рис.3.2. Выводы микроконтроллера ATtiny2313

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 <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 <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) <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) <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 <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);

- 3. 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;
2. - коэффициент деления предделителя;
OCR0A - код в регистре сравнения.

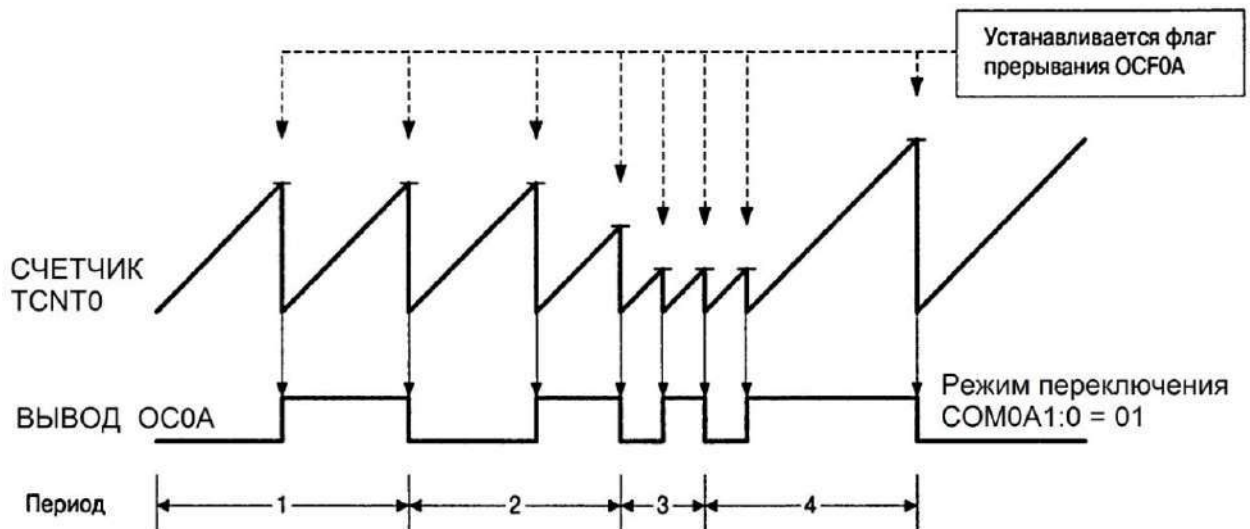


Рис.3.5. Временные диаграммы работы в режиме CTC

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:

1. 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);

2. при переполнении (переход через $\$FF$) устанавливается флаг переполнения TOV0;

3. при равенстве содержимого счетчика и регистра сравнения устанавливается флаг совпадения OCF0A или OCF0B;

4. при равенстве содержимого счетчика и регистра сравнения еще происходит управление выводами в соответствии с таблицей 3.4.



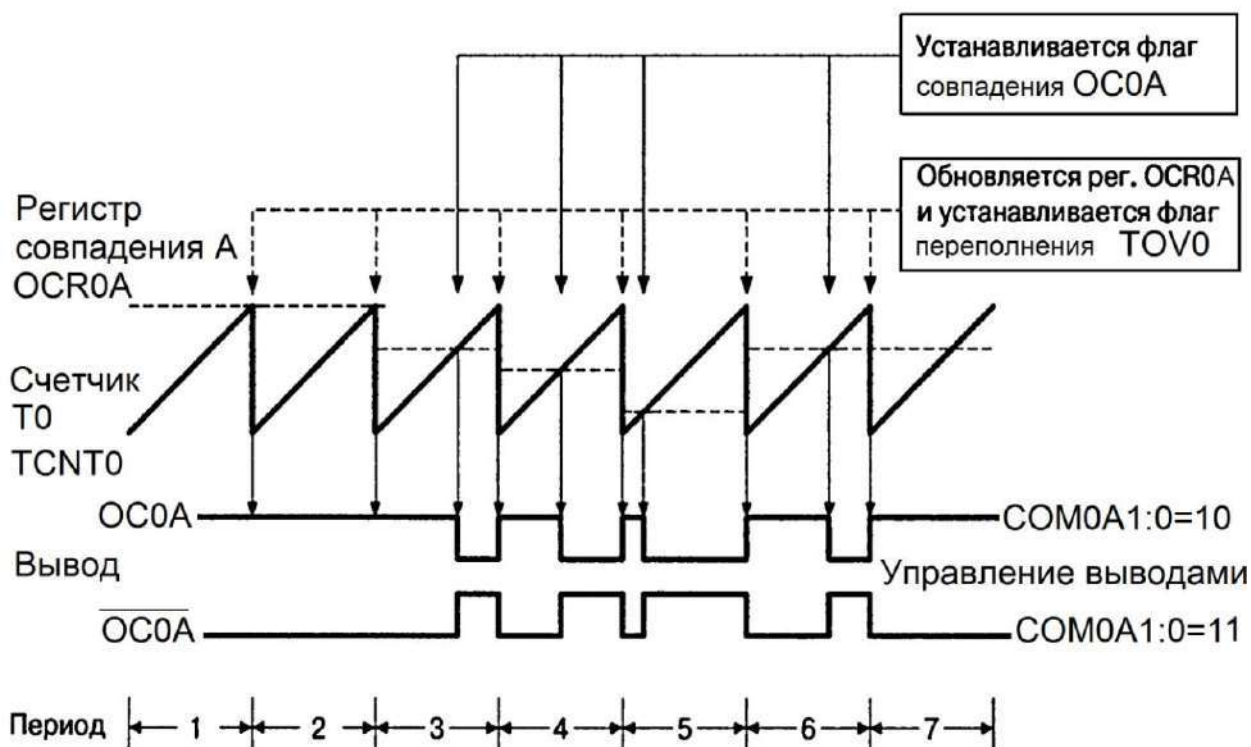
Таблица 3.4

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	OC0A: WGM02 = 0 — таймер/счетчик T0 отключен от вывода OC0A; WGM02 = 1 — состояние вывода меняется на противоположное при равенстве регистров TCNT0 и OCR0A. OC0B: 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_{CLK} - 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;

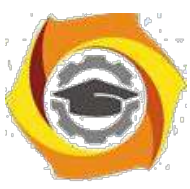
С при совпадении с регистром сравнения происходит изменение сигнала на выходе в соответствии с таблицей 3.5.

Таблица 3.5

Управление выводами, OC0A, OC0B в режиме Fast Correct PWM

COM0A1 (COM0B1)	COM0A0 (COM0B0)	Описание
0	0	Таймер/счетчик T0 отключен от вывода OC0A (OC0B)
0	1	OC0A: WGM02 = 0 — таймер/счетчик T0 отключен от вывода OC0A; WGM02 = 1 — состояние вывода меняется на противоположное при равенстве регистров TCNT0 и OCR0A. OC0B: 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)

б) 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.

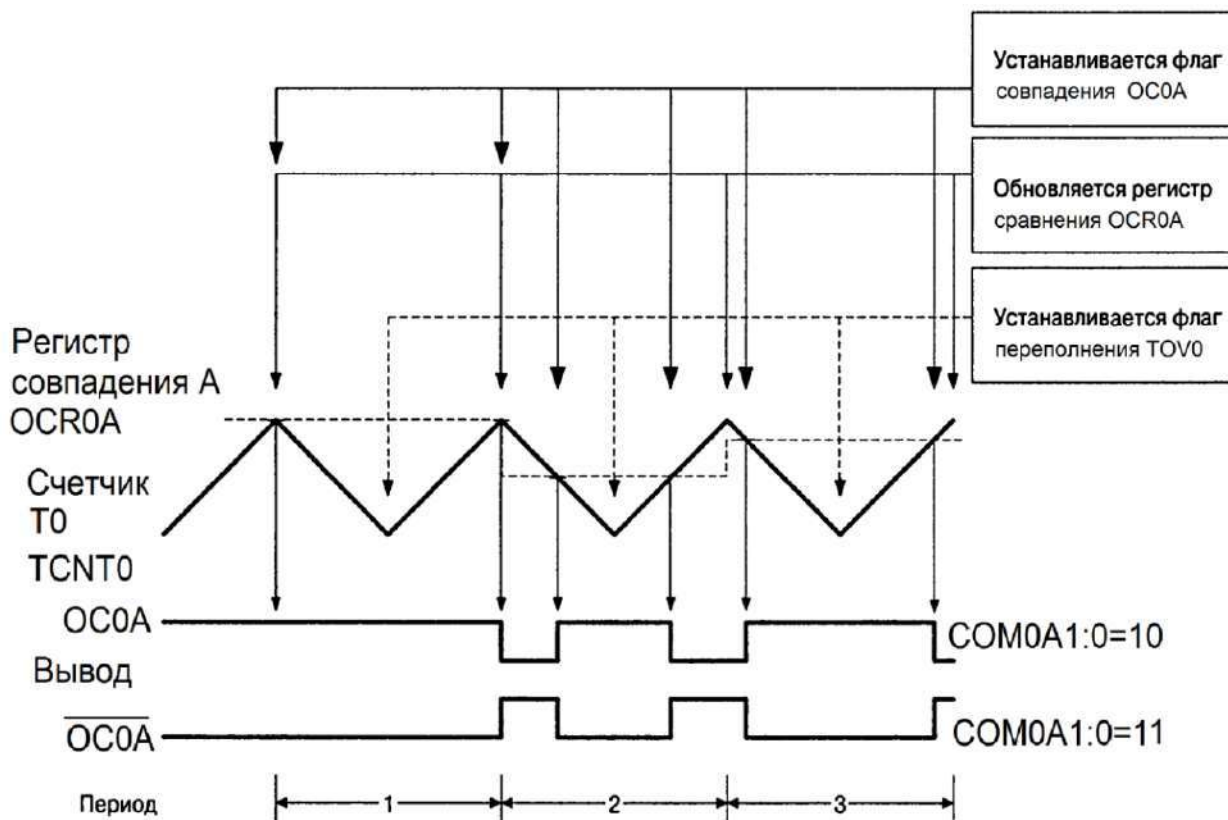


Рис.3.7. Диаграмма сигналов в режиме Fast Correct PWM

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.

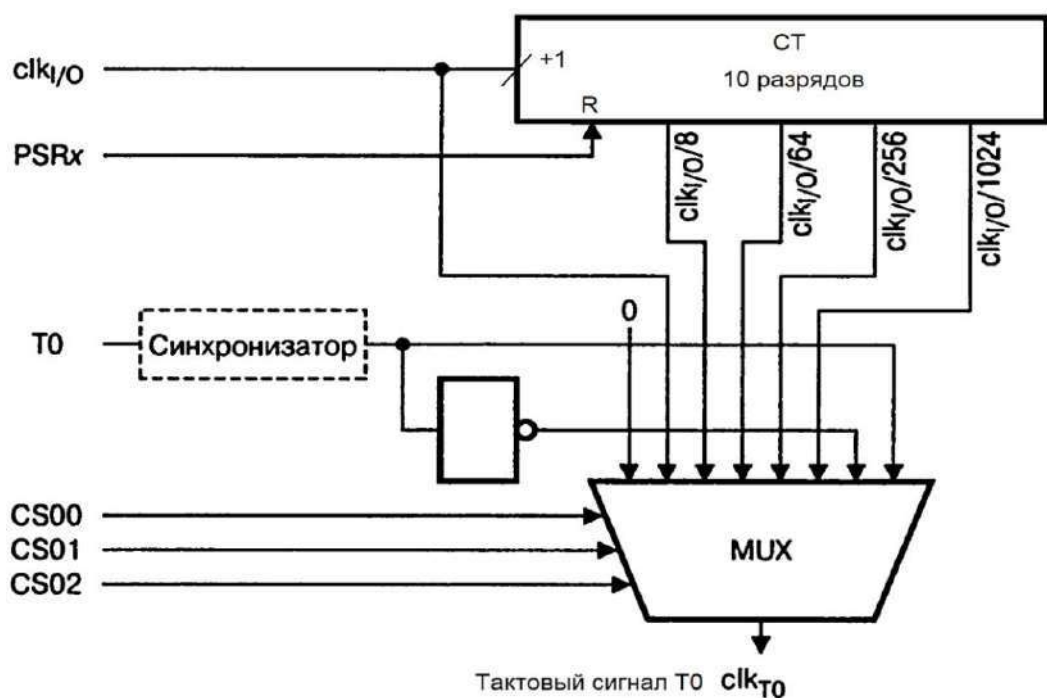
Задание 3.2. Задание параметров счетного сигнала

Изучить программную модель управления частой работы таймеров-счетчиков, научиться программировать требуемые параметры тактовых сигналов счетчиков-таймеров.

В отчете представить:

- 2) Адреса и имена регистров ввода-вывода, обеспечивающих управление предделителем;
- 3) Диапазон коэффициентов деления предделителя;
- 4) 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 _{I/O}
0	1	0	clk _{I/O} /8
0	1	1	clk _{I/O} /64
1	0	0	clk _{I/O} /256
1	0	1	clk _{I/O} /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.

4) 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):

$TCCR0A \& = \sim (1 \ll WGM01) | (1 \ll WGM00);$ // По умолчанию

$TCCR0B \& = \sim (1 \ll WGM02);$

2. задание режима работы выводов счетчика в TCCR0A (выводы COM0A и COM0B), например, в режим "изменение состояния на противоположное":

$TCCR0A \& = \sim (1 \ll COM0A0) | (1 \ll COM0B0);$

3. установка цифровых портов "на выход" для подключения выводов счетчика OC0A и OC0B (регистры DDRB, DDRD):

$DDRB |= (1 \ll PB2)$ // для вывода OC0A

$DDRD |= (1 \ll PD5)$ // для вывода OC0B;

- установка начального значения счетчика TCNT0 (TCNT1) или регистров сравнения OCR0 (OCR1):

$TCNT0 = \text{const};$ // Или OCR0A=... OCR0B= ...;

в сброс флагов счетчика-таймера T0 в регистре TIFR:

$TIFR |= (1 \ll OCF0B) | (1 \ll TOV0) | (1 \ll OCF0A);$

в разрешение прерываний счетчика-таймера T0 в TIMSK:

$TIMSK |= (1 \ll TOIE0) | (1 \ll OCIE0B) | (1 \ll OCIE0A);$

в установка предделителя счетчика-таймера T0 в TCCR0B, то есть старт таймера:

$TCCR0B |= (1 \ll CS02) | (0 \ll CS01) | (1 \ll CS00).$

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 ОС0А и ОС0В 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 ОС0А и ОС0В. 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 ОС0А или ОС1А. 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 Т1 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:



$$61 / 64 = 0,954 \text{ Гц};$$

1. Соответственно, для получения точно 1 Гц следует уменьшить количество отсчетов счетчика в 0,954 раза:

$$65536 * 0,954 = 62521;$$

2. 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:

- Выражения 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
      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

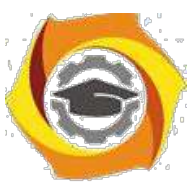
- С Белов А.В. Создаем устройства на микроконтроллерах. - СПб.: Наука и техника, 2007. - 304 с.
- С Евстифеев А.В. Микроконтроллеры AVR семейства Tiny. Руководство пользователя. - М.: Издательский дом Додэка-XXI, 2007. - 432с.
- С Ревич Ю.В. Практическое программирование микроконтроллеров Atmel AVR на 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	Переполнение 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 записываются 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.

3. 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)  
{  
    3. 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 <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 <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 <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 <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 <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 ISC11:ISC10	Определяют условие генерации внешних прерываний INT0 и INT1 следующим образом:		
	ISCn1	ISCn0	Условие
	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 <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.

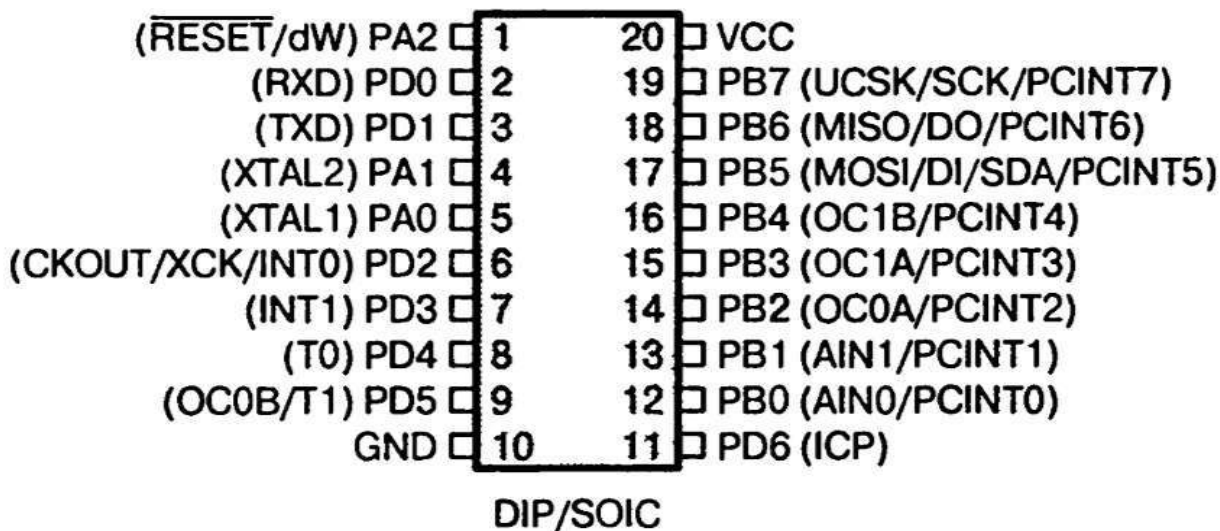


Рис.4.6. Выводы микроконтроллера ATtiny2313

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»**

отпуская. 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.

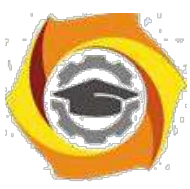
Если коэффициент делителя выбрать 64, то частота тактирования счетчика TCNT1 вычисляется: $4\text{МГц}/64=62,5\text{КГц}$, что соответствует периоду 16 мкс .

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:

3. Схему подключения светодиода;
4. Расчетные соотношения для периода 1 сек;
5. Схему алгоритма программы;
6. 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 отчета 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?



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

77

- 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 С?
- 12. Как написать обработчик на языке С?
- 1 Что представляет собой реакция микроконтроллера на сигнал преры-вания?
- 14. Каким образом процессор завершает выполнение обработчика? Что при этом выполняется?
- 4 Каким образом с помощью таймера организовать задержку большой длительности (например, минуты)?
- 5 Какие настройки выполняются для организации прерываний?
- 6 Какую роль выполняют флаги прерываний?
- 7 Как определяется приоритет прерываний микроконтроллера?
- 8 Как можно уточнить зарезервированные имена обработчиков;
- 9 Что нужно сделать, чтобы обработчик не выполнялся?
- 10 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.

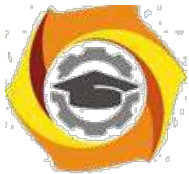
Студенты заочной формы обучения, работающие по профилю подготовки, проходят практику по месту работы по согласованию с администрацией предприятия. Студенты, не работающие по направлению подготовки, проходят практику по месту жительства на предприятиях (организациях), профиль работы которых соответствует 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, ρ кг/м³, вычисляются по формуле:

$$\rho = \frac{m}{V}, \quad (1)$$

где m – масса образца, кг,

V – объем образца, м³.

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 (подрисовочный текст). Точка после подрисовочного 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 заголовки 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.

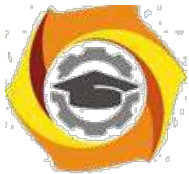
Отчетность по практике

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, ρ кг/м³, вычисляют по формуле:

$$\rho = \frac{m}{V}, \quad (1)$$

где m – масса образца, кг,

V – объем образца, м³.

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 (подрисовочный текст). Точка после подрисовочного 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 подрисовочная 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 заголовки 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:

CRT -	Cathode	Ray	Tube;
LCD -	Liquid	Crystal	Display;
DLP -	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



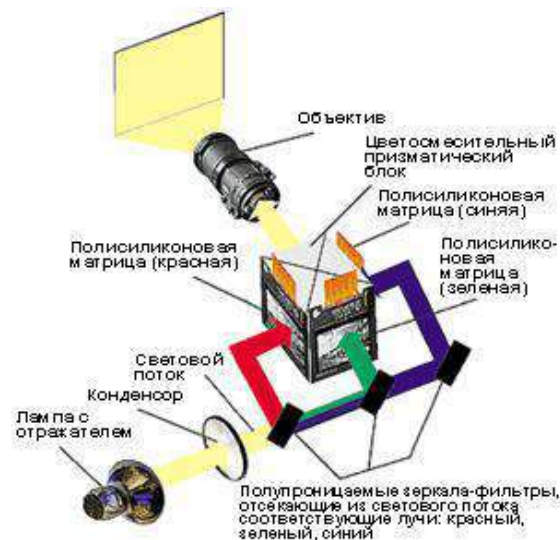
Негосударственное частное образовательное учреждение высшего образования «Технический университет»

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.

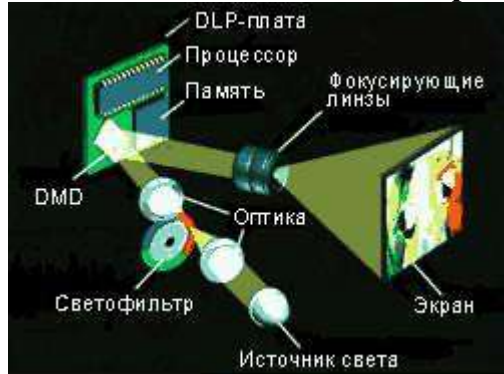
Устройство DLP-проектора



**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

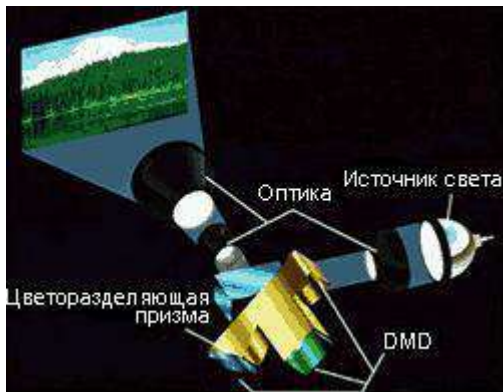


В двухматричных DLP-проекторах вращающийся цветной фильтр имеет два сектора пурпурного (смесь красного с синим) и желтого (смесь красного и зеленого) цветов. Дихроичные призмы разделяют световой поток на составляющие, при этом поток красного цвета в каждом случае направляется на одну из DMD-матриц. На 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

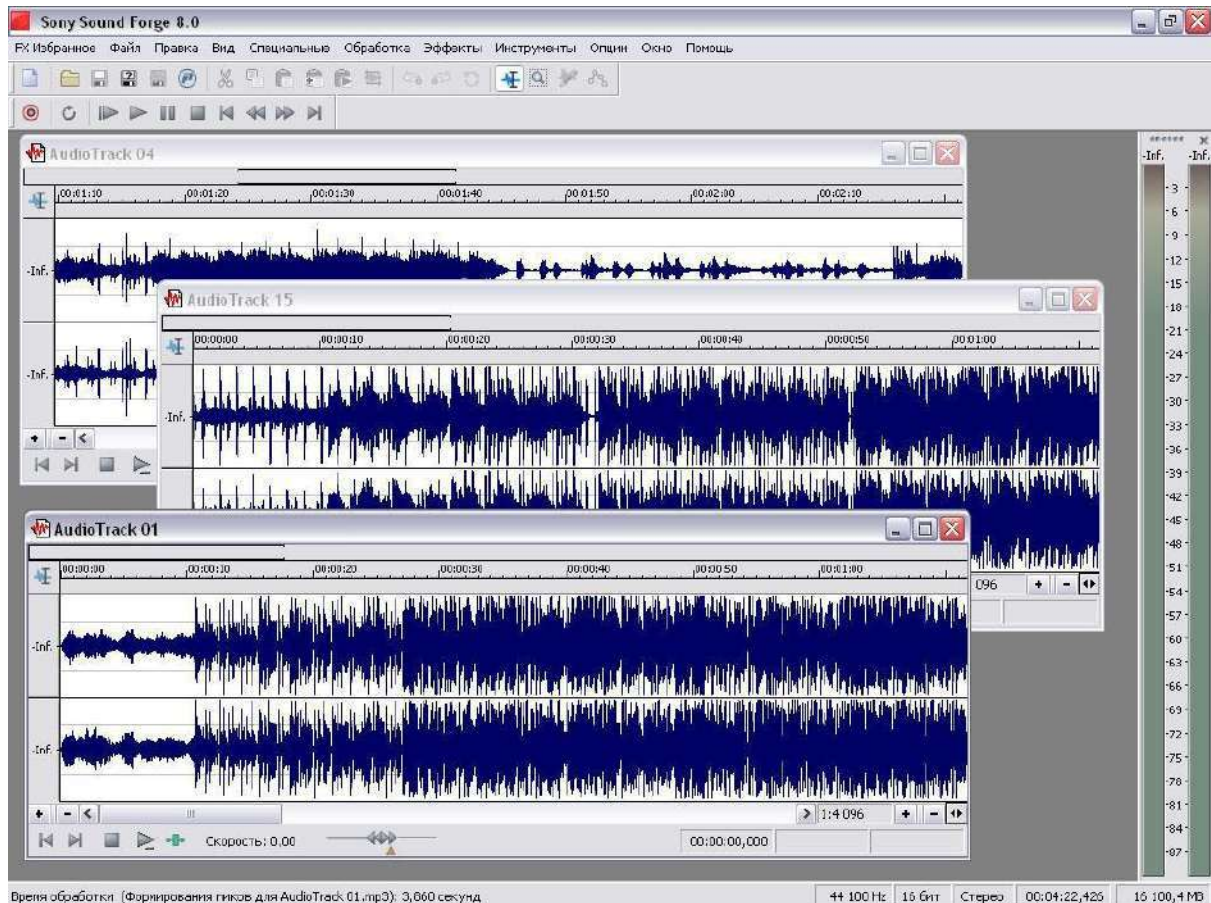
· **Цель работы**

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





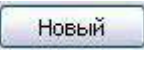
Sound Forge 8.0 фирмы Sonic Foundry, Inc - это одна из самых популярных и действительно полезных программ, предназначенных для обработки аудио на профессиональном уровне. С ее помощью можно обрабатывать аудиосигнал, изменяя его до неузнаваемости или же редактировать неудачно записанную партию какого-либо 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 I_R или I_L .




**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.

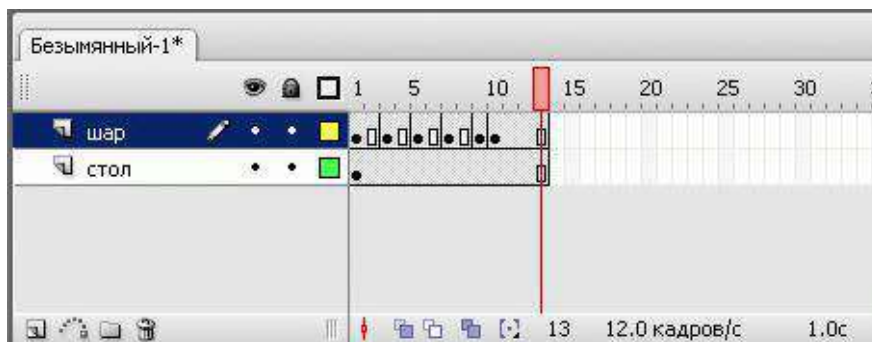
Чтобы просмотреть анимацию "глазами пользователя", т. е. как она будет выглядеть в итоге, нажмите комбинацию клавиш CTRL+Enter или Shift+CTRL+Enter (в меню "Управление" - "Тестировать ролик" или "Тестировать сцену").

Просмотрев полученный мультик, мы имеем два вывода:

- Наконец-то! У меня на экране что-то движется.
- 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;



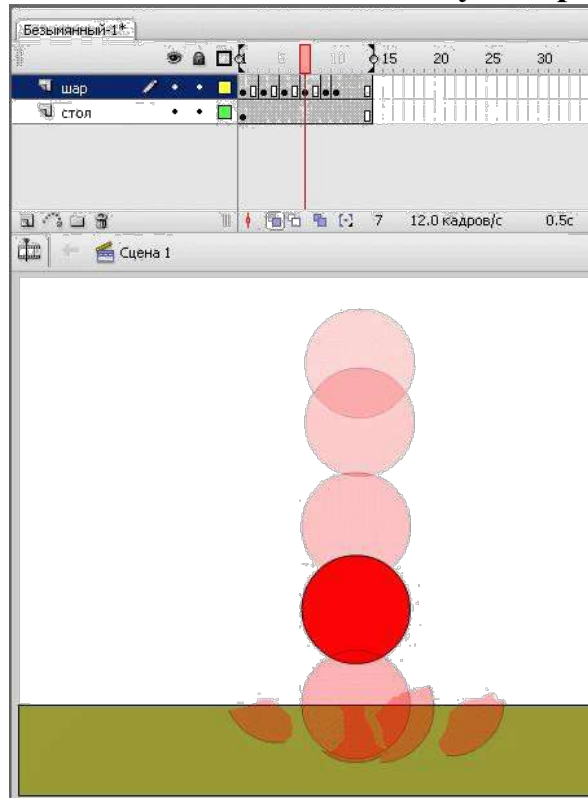
- В добавим к анимации фон - поверхность, о которую разбивается наш шар. Для этого создадим еще один слой в панели временной линейки, перетащим его вниз (наш "стол" должен быть под шариком). По умолчанию при создании нового слоя сразу создается пустой ключевой кадр и промежуточные по длине уже готовой 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

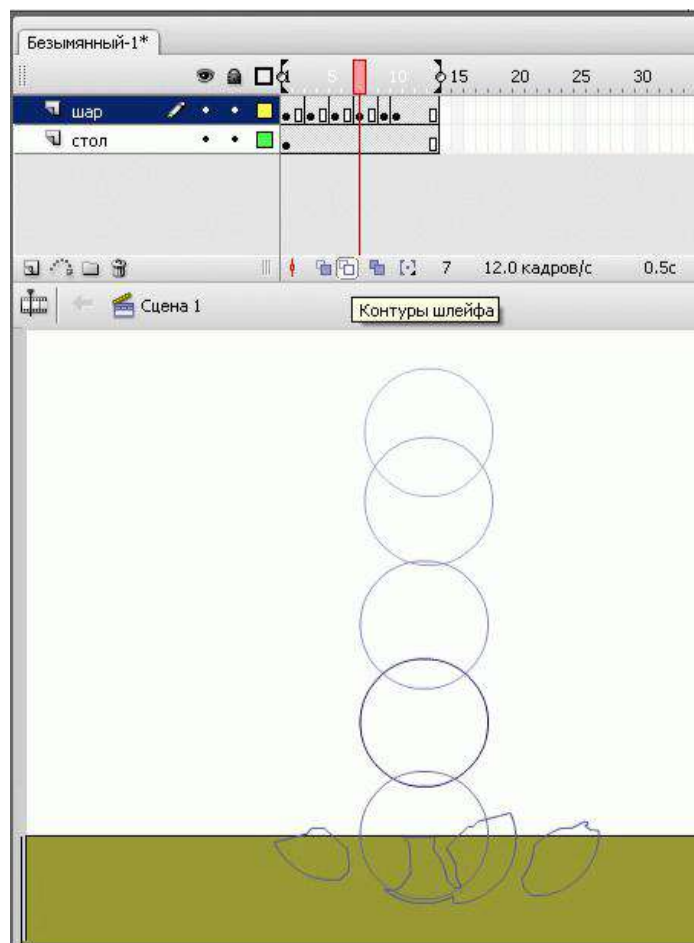
- 7) пяти кнопок, расположенных в низу линейки кадров. а)

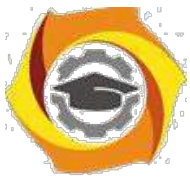


Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»



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.

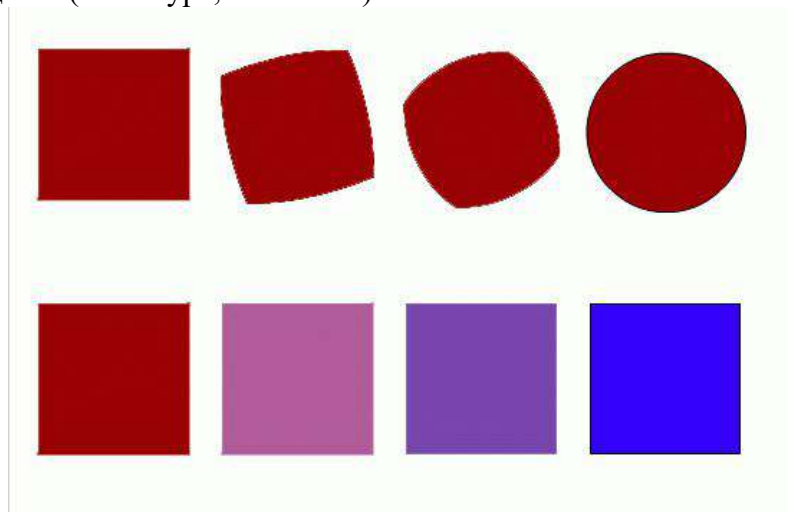
Итак, первая анимация готова. У вас в руках есть все возможности ее редактирования (кадры, слои, объекты).

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).



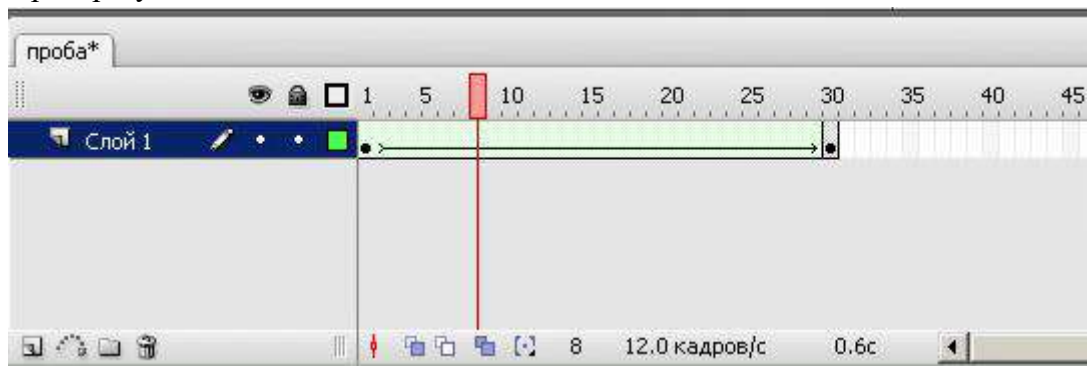
Рассмотрим анимацию формы на примере.

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:

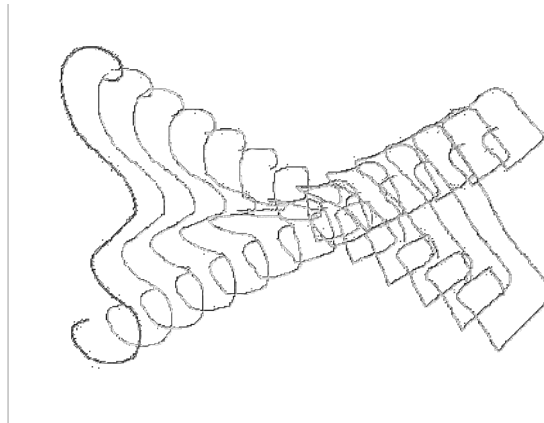
- 28. скорость изменений; по умолчанию изменения протекают с постоянной скоростью, однако вы можете управлять ею, изменяя значения параметра "Замедлить": отрицательные значения этого параметра (от -1 до -100) означают, что изменения будут постепенно ускоряться, а положительные значения (от 1 до 100), 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.

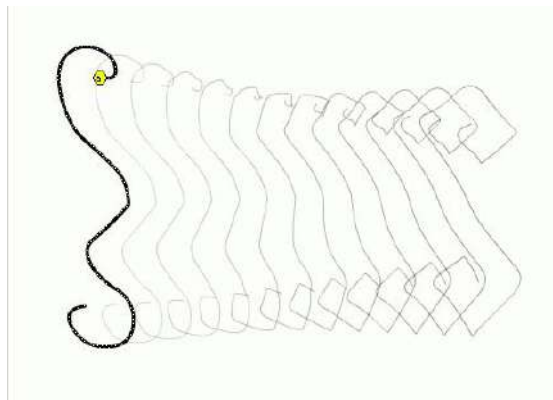
а)



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:

12. Создайте анимацию формы между двумя ключевыми кадрами.
13. Щелкните левой кнопкой мыши в ячейке ключевого кадра, соответствующего исходному изображению (первого ключевого кадра).
14. 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".
15. 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.
16. 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.
17. 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»

10. Вернитесь на первый ключевой кадр анимированной последовательности и убедитесь, что цвет метки изменился на желтый.
11. 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.

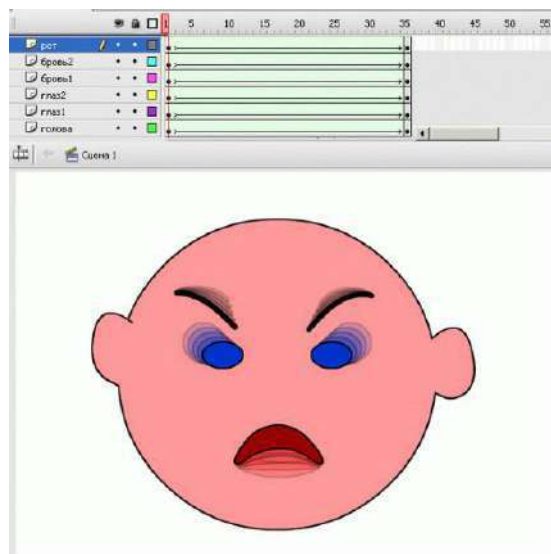
Анимация формы просто завораживает подростков - они могут по несколько минут любоваться самыми простыми преобразованиями. При этом, как показывает опыт, они в большинстве своем не видят нелогичности и "неправильности" преобразований, когда все находится в движении.

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.

12. 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**

11. 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;



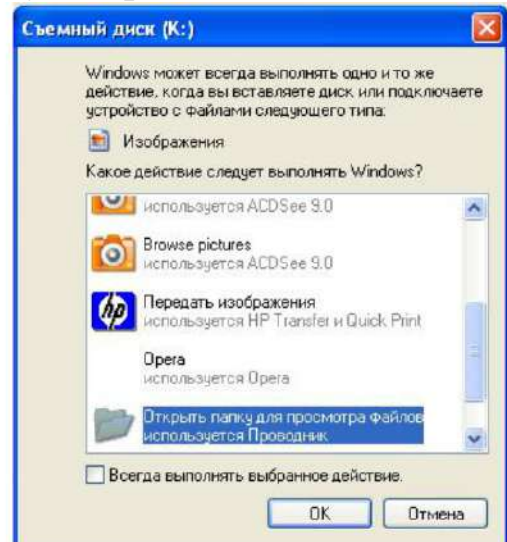
разрешение экрана от 1024x768 с глубиной цвета 32 бит.

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.

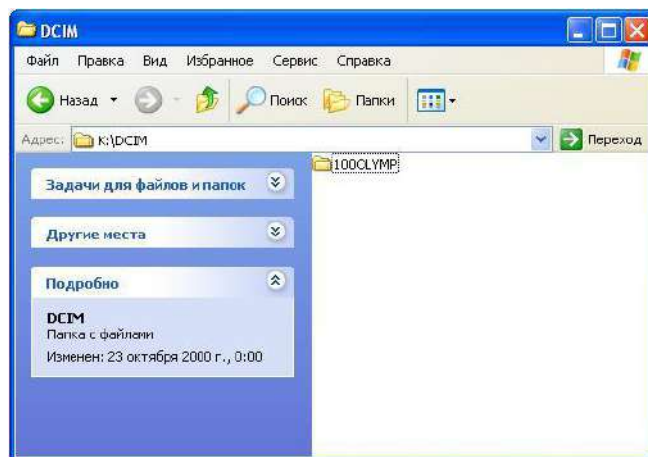
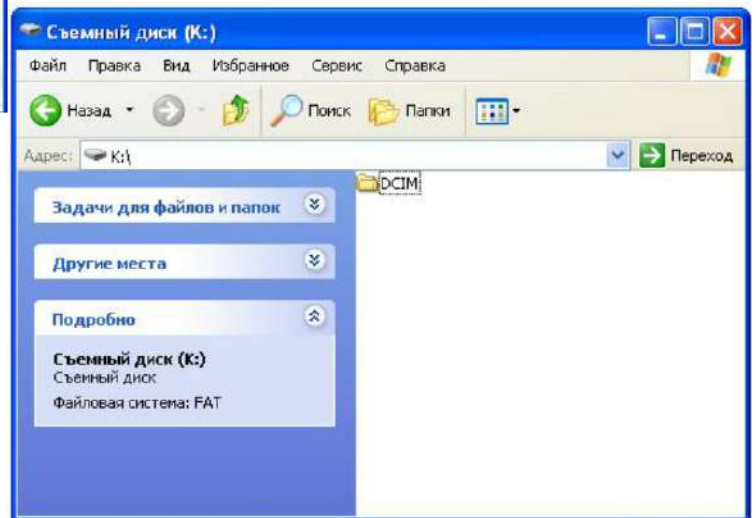
При подключении к компьютеру фотоаппарат распознается как внешний накопитель. Если используется Windows XP, то проблем с драйверами не возникает и все работает через 10-30 секунд после подключения.



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;

- б) 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



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»

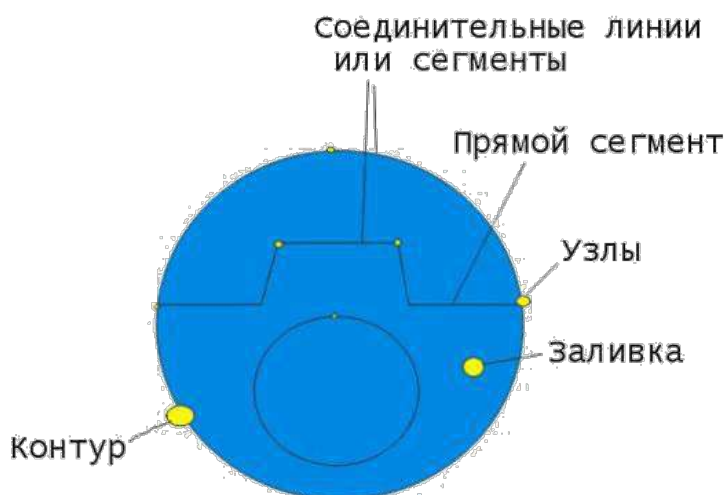


Рис. 1.1. Пример объекта

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

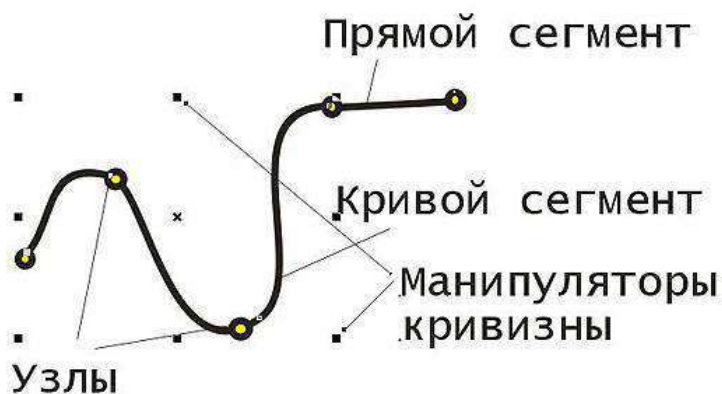


Рис. 1.2. Кривая Безье

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»**

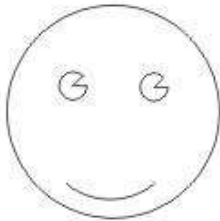
программы. Теперь, когда рабочее окно занимает весь экран, работать с CorelDRAW станет удобнее.

2.2 Перечень используемого оборудования

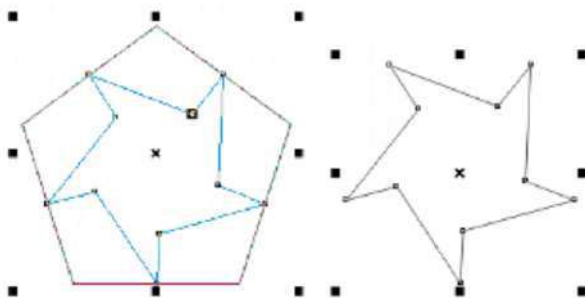
Персональный компьютер

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.

б) 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.



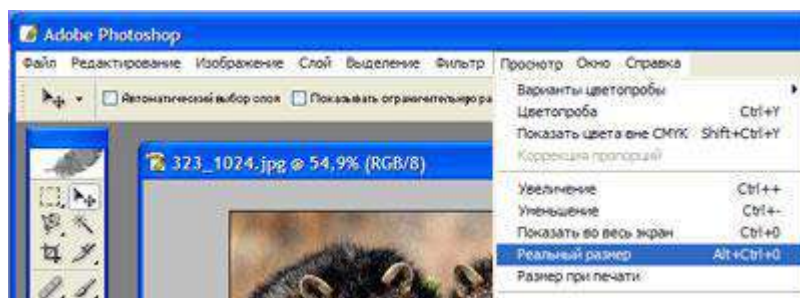
**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Размер изображения

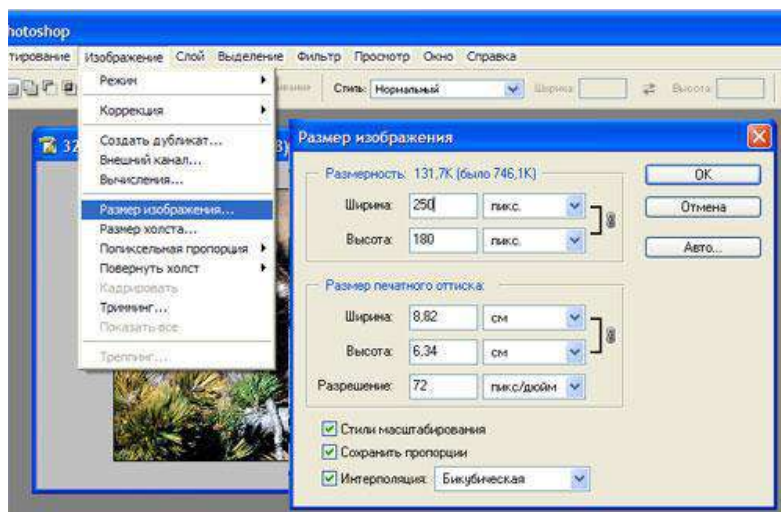
Когда говорят о размере фотографии, имеют в виду либо размер файла фотографии, либо размер самого изображения.

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

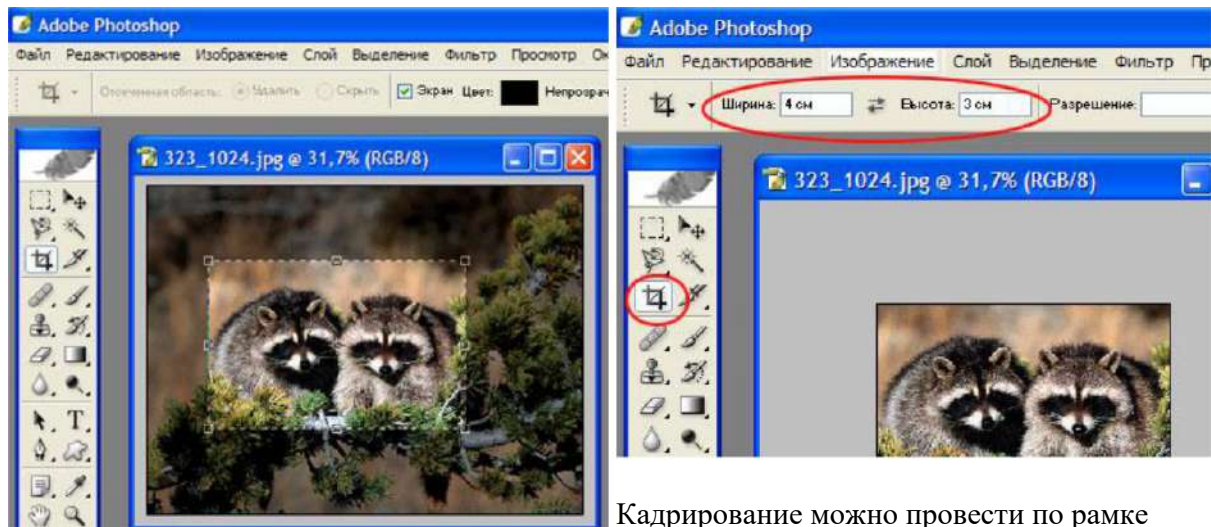
Иногда требуется кадрировать изображение, т.е. выделить фрагмент (Кадр), который нужен, а остальное - отрезать (например, после вращения остаются белые поля по углам).

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.

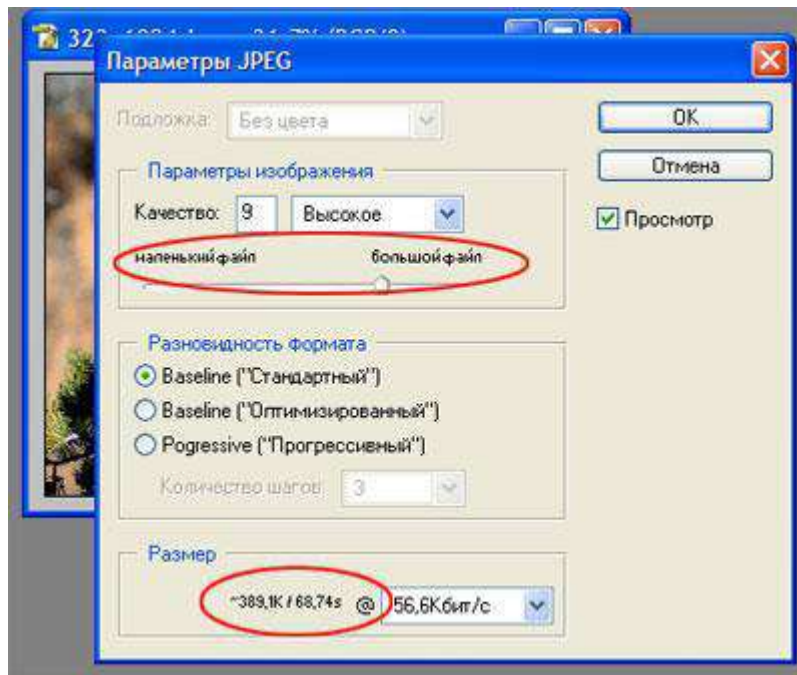
JPEG - Это формат, позволяющий создать минимальный по размерам файл с наименьшей потерей качества изображения. Поэтому он получил большую популярность - это стандарт для изображений в Интернете, в нем сохраняют многие фотоаппараты и фотографии в JPEG принимаются на печать в лабораториях. Минус у этого 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.

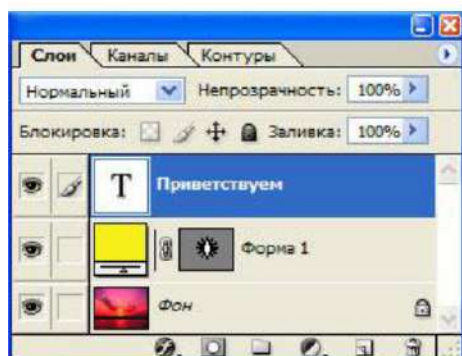


Рис. 2.1 Палитра Слои

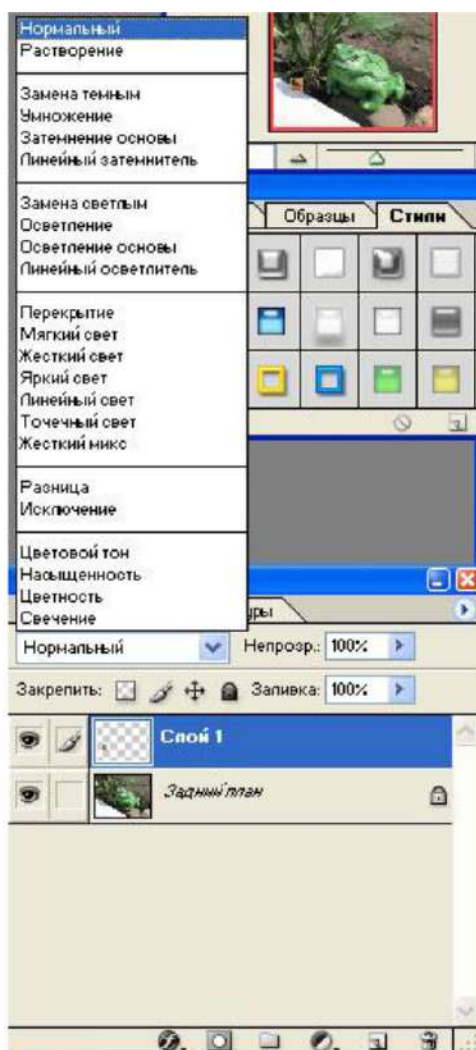


Рис. 2.2 Режимы наложения



**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»

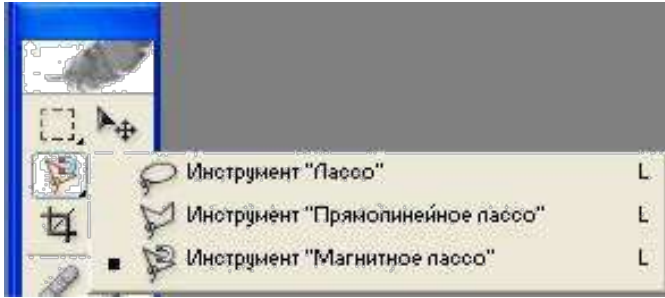



Рис. 2.3 Инструмент Лассо

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»

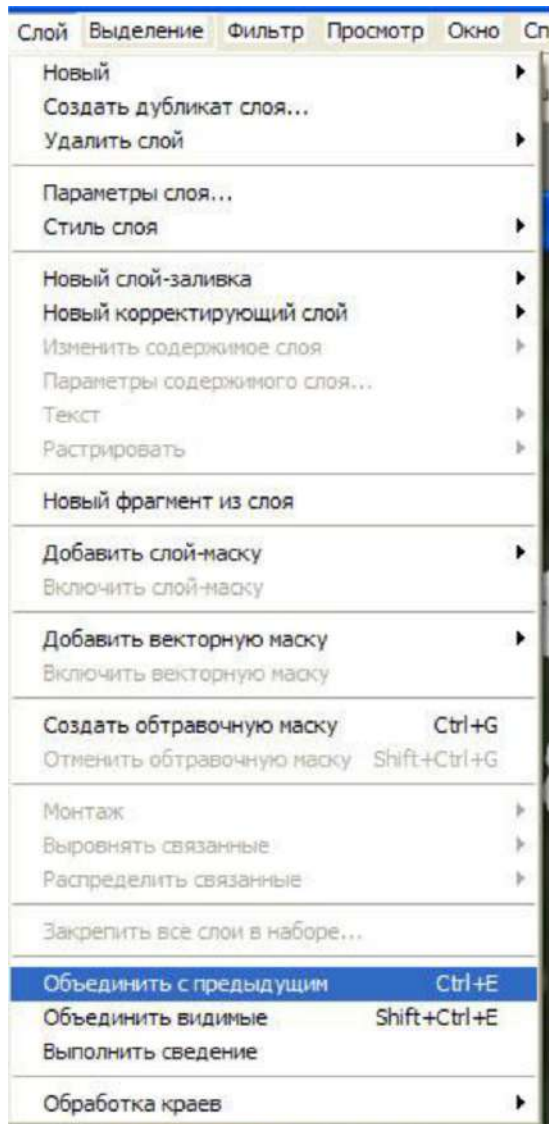


Рис. 2.5 Замкнутый контур выделения

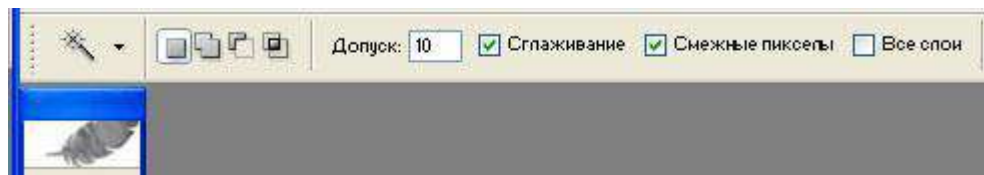


Рис. 2.6 Панель свойств инструмента "Волшебная палочка"

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.

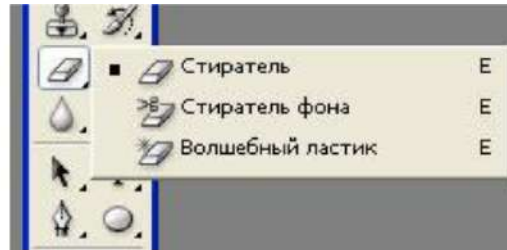
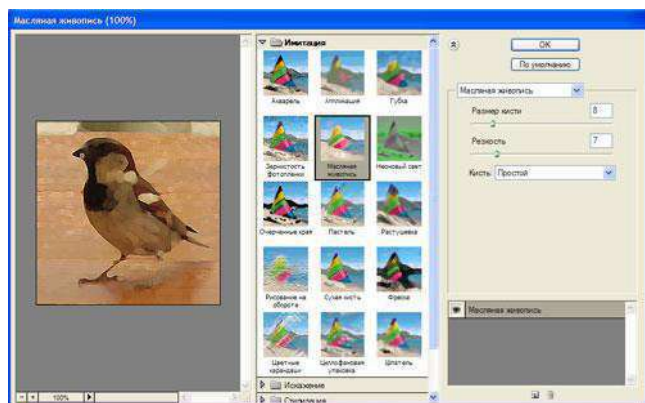


Рис.2.8 Инструмент Ластик

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»

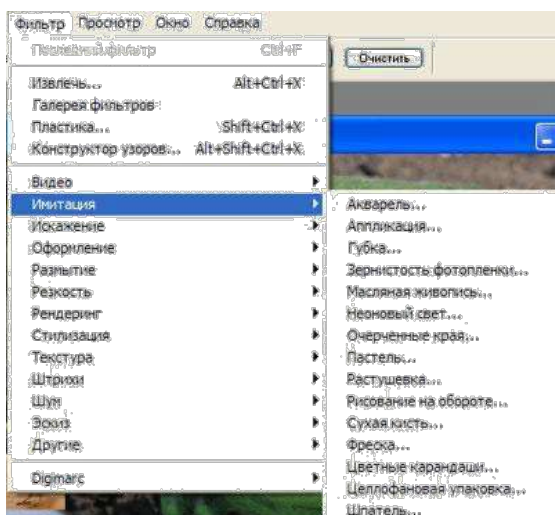


Рис. 2.10 Окно фильтра

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 ϕ 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.

5. Изменение внешнего вида текста

Выберите текст. В меню **Формат** выберите пункт:

8. **Шрифт**, чтобы изменить шрифт или его размер, цвет и стиль начертания;
9. **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**.

9) 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...»**.

б) 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**.

6. Перемещение рисунка

Установите указатель на рисунке. Когда указатель примет вид крестообразной стрелки перетащите рисунок.

11. 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 HF** на панели инструментов **Рисунок**. Установите указатель на черный маркер обрезки и перетащите маркер.

4) Поворот рисунка

Щелкните по рисунку. Установите указатель на зеленый

маркер поворота. Когда указатель примет вид кружка (C£, поверните рисунок.

- Управление режимами обтекания текстом

Нажмите кнопку **Обтекание** на панели инструментов **Текст** **Рисунок**.

7. Вывод на экран панели инструментов Рисунок

4 меню **Вид** выберите пункт **Панели инструментов**, затем выберите пункт **Рисунок**.

7. 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**.

9. 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**.

9. 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.

8. 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

4) 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

7) 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. Обоснование выбора методов исследования.
2	Составление отчета по темам практики: Проведение экспериментальной части исследований. Обработка результатов экспериментов.
3	Составление отчета по темам практики: Анализ результатов теоретической части исследования. Анализ результатов практической части исследования. Оформление научных результатов в формате магистерской диссертации.

Самостоятельная работа № 1,2,3

Составление отчета по НИР

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ	2
1. ОБЩИЕ ПОЛОЖЕНИЯ	3
2. СТРУКТУРА ОТЧЕТА	3
3. ТРЕБОВАНИЯ К СТРУКТУРНЫМ ЭЛЕМЕНТАМ ОТЧЕТА	3
4. ПРАВИЛА ОФОРМЛЕНИЯ ОТЧЕТА	7
ЗАДАНИЕ	14



2. 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.

3. СТРУКТУРА ОТЧЕТА

Структурными элементами отчета о НИР являются:

6. **титульный лист;**
7. список исполнителей;
8. **реферат;**
9. **содержание;**
10. перечень сокращений, условных обозначений, символов, единиц и тер-минов;
11. **введение;**
12. **основная часть;**
13. **заключение;**
14. список использованных источников;
15. приложения.

Обязательные структурные элементы выделены полужирным шрифтом. Остальные структурные элементы включают в отчет по смотрению исполнителя НИР с учетом требований разд.3 и 4.

4. 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:

2. наименование организации - исполнителя НИР;
3. индекс УДК;
4. коды Высших классификационных группировок Общесоюзного классификатора промышленной и сельскохозяйственной продукции для НИР, предшествующих постановке продукции на производство;
5. номера, идентифицирующие отчет;
6. грифы согласования и утверждения;



6. наименование работы;
7. наименование отчета;
8. вид отчета (заключительный, промежуточный);
9. номер (шифр) темы;
10. должности, ученые степени, ученые звания, фамилии и инициалы руководи-телей организации - исполнителя НИР, руководителей НИР;
11. 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:

18. сведения об объеме отчета, количестве иллюстраций, таблиц, приложений, количестве книг отчета, количестве использованных источников;
19. перечень ключевых слов;
20. 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;

6. объект исследования или разработки;
7. цель работы;
8. метод исследования и аппаратуру;
9. полученные результаты и их новизну;
10. основные конструктивные, технологические и технико-эксплуатационные характеристики;
11. степень внедрения;
12. рекомендации по внедрению или итоги внедрения результатов НИР;
13. область применения;
14. экономическую эффективность или значимость работы;
15. 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:

б. 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 3) 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

№ п.п.	Разделы (этапы) практики	Всего дней
1	Выездная практика на предприятии	10
1.1	Инструктаж по соблюдению правил внутреннего трудового распорядка, требований охраны труда, пожарной безопасности и ТБ	1
1.2	Проведение научно-исследовательской работы на предприятии, направленной на управление и устойчивое развитие электрохозяйства в соответствии с ТЗ на ВКР	7
1.3	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:

- 2) формуляр отчета по практике (приложение 3);
- 3) отзыв руководителя практики от предприятия (приложение 4);
- 4) справка о проведении инструктажей обучающихся по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, правилами внутреннего трудового распорядка (приложение 1);

Приложение 1

СПРАВКА

1.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. с 9-00 до 9-30	Требования охраны труда: 1. 2.	Иванов И.А., мастер АО «__»	Сидоров А.А.
03.07.2017 г. с 9-30 до 10-00	Требования техники безопасности: 1. 2.	Иванов И.А., мастер АО «__»	Сидоров А.А.
03.07.2017 г. с 10-00 до 10-30	Требования пожарной безопасности: 1. 2.	Иванов И.А., мастер АО «__»	Сидоров А.А.
03.07.2017 г. с 10-30 до 11-00	Правилами внутреннего трудового распорядка: 1. 2..	Петров А.С. специалист кадрового управления АО «__»	Сидоров А.А.

Руководитель практики от предприятия: _____ И.О.Фамилия
Руководитель по персоналу
организации УГМК: _____ И.О.Фамилия

М.П.



Структура отчета по практике

1. *Титульный лист*
2. *Содержание* (включает названия разделов с указанием страниц, с которых они начинаются)
3. *Введение* (включает цели, задачи, сроки практики, краткую характеристику предприятия, технологических процессов, оборудования в соответствии с направлением подготовки)
4. *Основная часть* (включает описание выполненной на практике работы, с подробным раскрытием видов работ, оборудования, приборов, инструментов, технологий и т.д.)
5. *Заключение* (включает выводы, рациональные предложения, инновации сделанные во время прохождения практики)
6. *Список использованной литературы и нормативных документов*
7. *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	_____	_____	курса
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



ОТЗЫВ РУКОВОДИТЕЛЯ ПРАКТИКИ ОТ ПРЕДПРИЯТИЯ

Название практики

2. 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 производственных

условиях, соблюдение трудовой дисциплины и др.

ОБЩЕЕ ЗАКЛЮЧЕНИЕ:

ЗАМЕЧАНИЯ

Руководитель практики от предприятия:

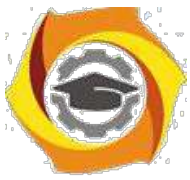
Фамилия, имя, отчество:

Должность:

Подпись

Дата: _____ 20__

(печать организации)



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.

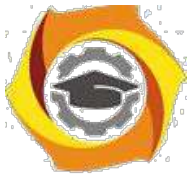
Класс используется для создания объектов. Основная форма имеет вид:

```
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;  
}
```

11. 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.

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.

- 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:

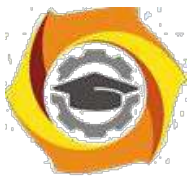


```
void set_i(int n);  
int get_i();  
};  
  
/ Задание производного класса  
class derived : public base  
{  
    int j;  
    public:  
    void set_j(int n);  
    int mul();  
};  
  
/ Установка значения i в базовом классе  
void base::set_i(int n)  
{  
    i = n;  
}  
  
/ Возврат значения 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.

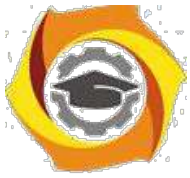
40. Варианты заданий:

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.

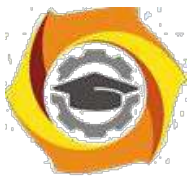


5. Даны числа S, T . Получить с использованием функции пользователя с параметрами $G(1, \sin(S)) + 2G(T * S, 24) - G(5, -S)$, где $G(A, B) = (2A + B * B) / (A * B * 2 + B * 5)$.
24. Составить программу для расчета значений гипотенузы треугольника, определив функцию, выполняющую этот расчет. Катеты передаются 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 передается 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»**

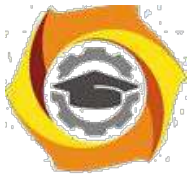
19

- 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 $F(x, y) = (2x^3 - 4x^2 + x + 1) / (9y^3 + y + 4) + 3y^2 + 5y$.



**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.



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:

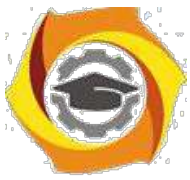
```
ob1 = 0, 0
ob2 = 30, 40
```



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

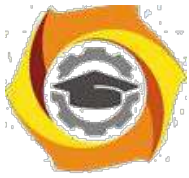
- в Составить программу для перевода длины в метрах в длину в сантиметрах, определив функцию, выполняющую это преобразование и передав длину в метрах в качестве параметра.
- в Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив функцию, выполняющую 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.

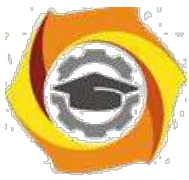


- Найти периметр десятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, которые передаются функции в качестве параметров из основной программы.
- Найти периметр шестиугольника, координаты вершин которого заданы. 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 «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 тоже 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.

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.

- 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).

ее 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»**

{



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 устанавки 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.

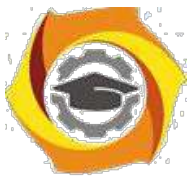


5. Даны числа S, T . Получить с использованием функции пользователя с параметрами $G(1, \sin(S)) + 2G(T * S, 24) - G(5, -S)$, где $G(A, B) = (2A + B * B) / (A * B * 2 + B * 5)$.
35. Составить программу для расчета значений гипотенузы треугольника, определив функцию, выполняющую этот расчет. Катеты передаются 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 $G(A, B) = (2 * A + B * B) / (A * B * 2 + B * 5)$.
- ООО Определите функцию, определяющую, какой целой степенью числа 2 является ее аргумент (если число не является степенью двойки - выдать соответствующее сообщение).
- ООО Определите функцию, подсчитывающую сумму N первых элементов целочисленного массива A. N и массив A передать в качестве параметров.
- ООО Вычислить количество простых чисел, не превосходящих заданного N. Описать функцию логического типа, возвращающую значение true, если число простое и false в противном случае.
- ООО 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.



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

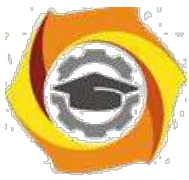
- в Составить программу для перевода длины в метрах в длину в сантиметрах, определив функцию, выполняющую это преобразование и передав длину в метрах в качестве параметра.
- в Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив функцию, выполняющую 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:

- с) Приобретение навыков установки и создания виртуальных машин в Oracle VM VirtualBox
- д) 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.

5) 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.

5) 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), 4. 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.

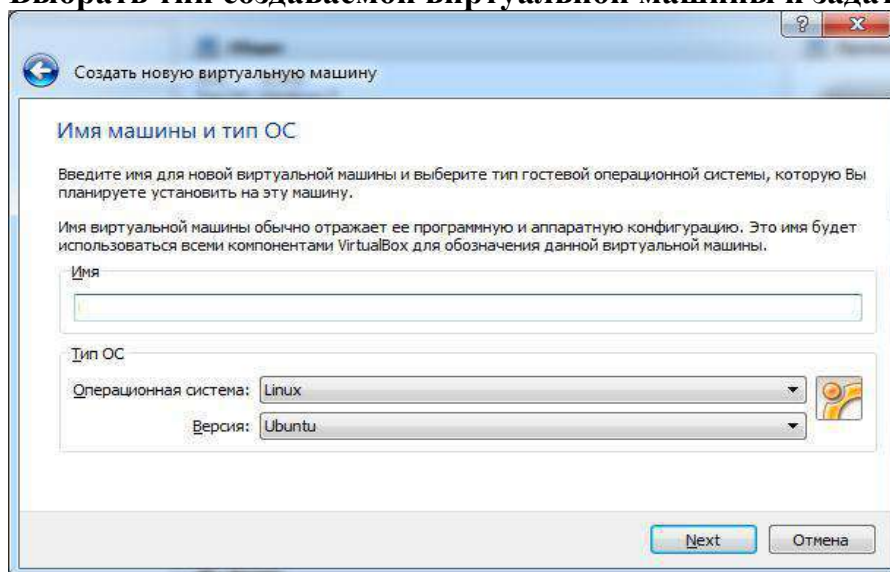
1.3.Ход работы:

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

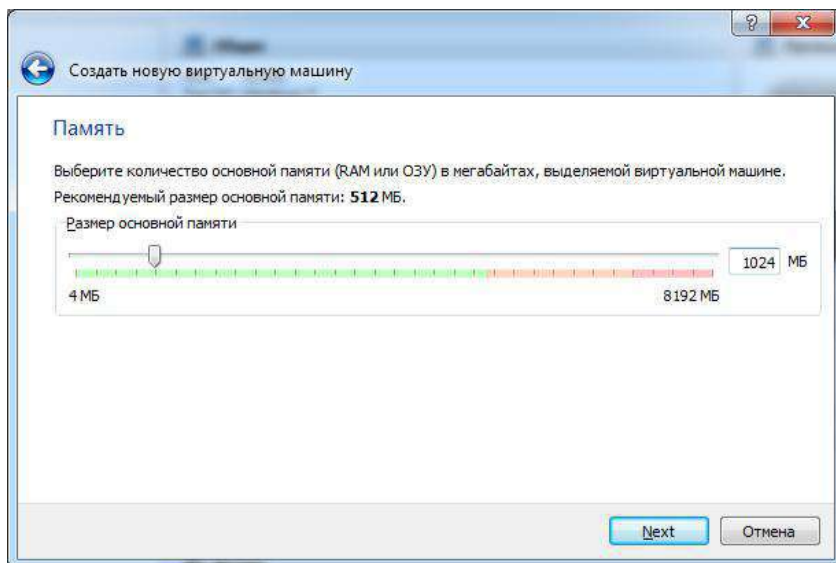
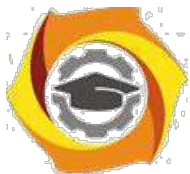
Установка Oracle VM VirtualBox производится с настройками по умолчанию. После установки виртуальной машины установить пакет расширения (Oracle_VM_VirtualBox_Extension_Pack).

1)Создание виртуальной машины:

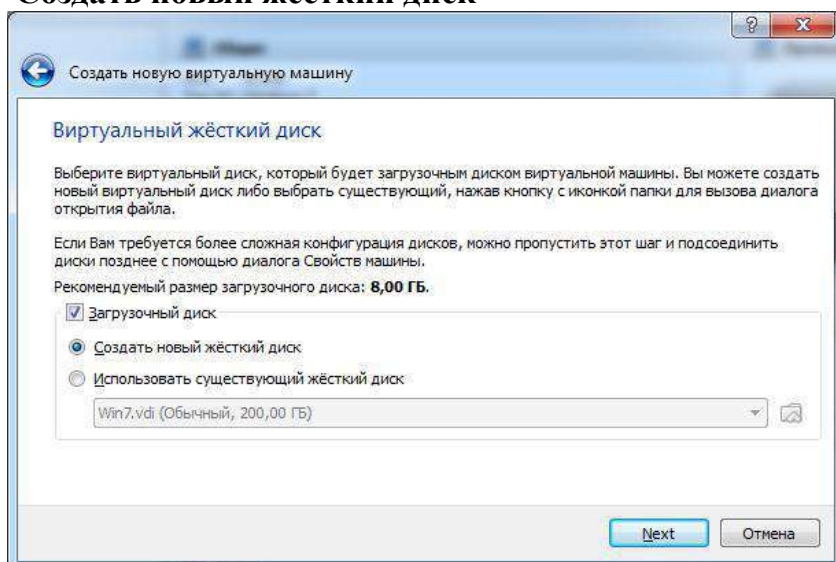
Выбрать тип создаваемой виртуальной машины и задать имя.



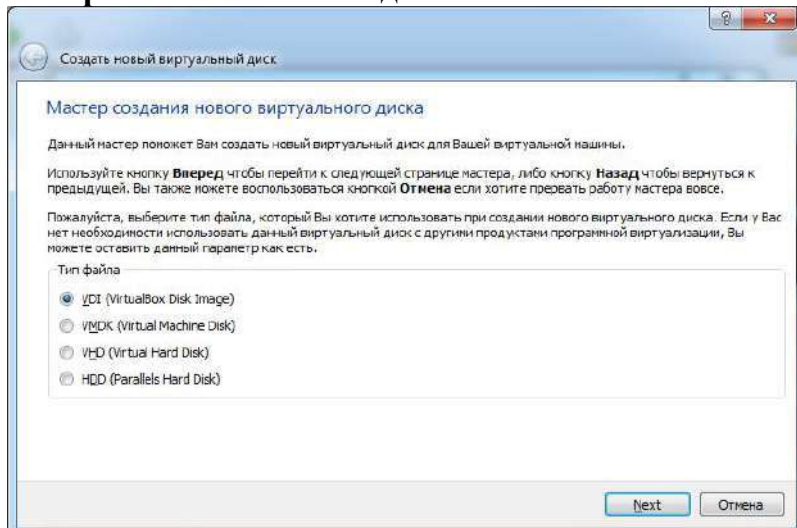
Установить объем памяти 1024 Мб



Создать новый жесткий диск



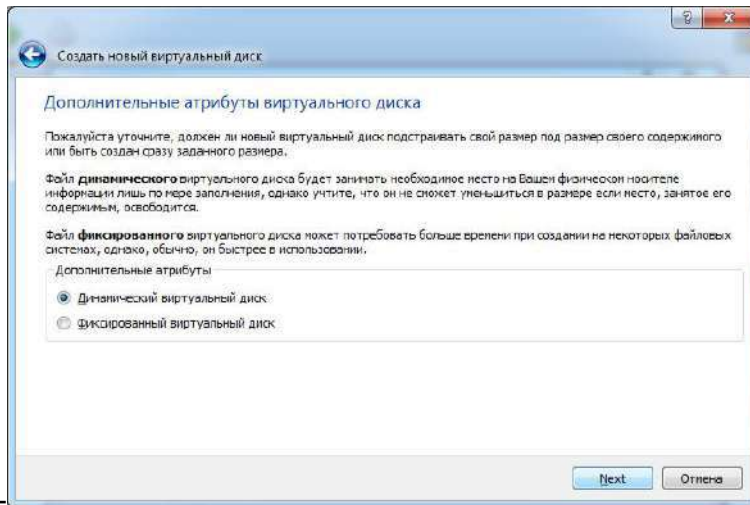
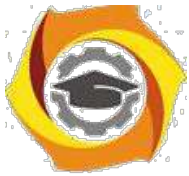
Выбрать тип жесткого диска



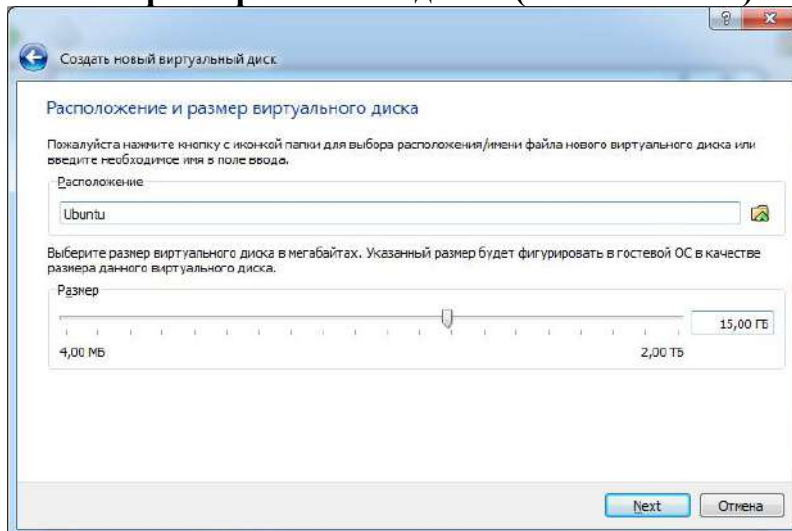


**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

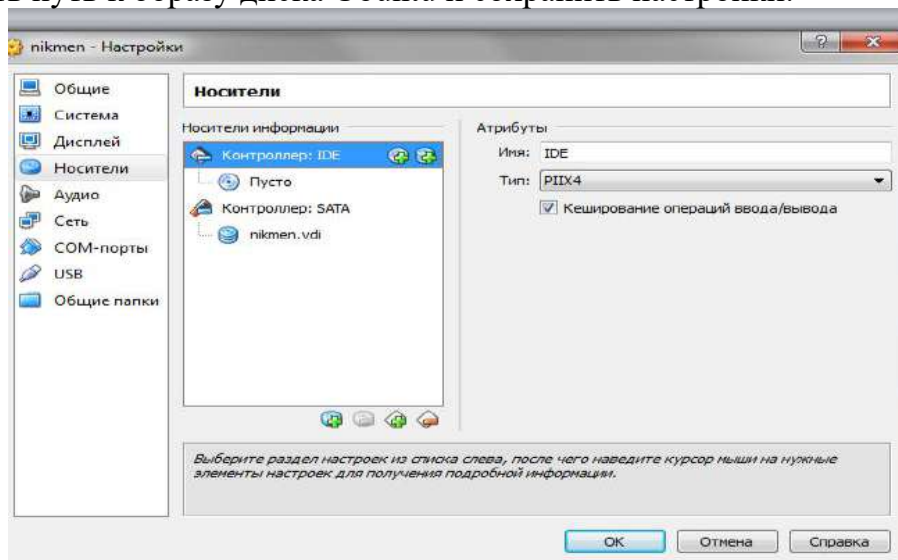
Выбрать атрибуты жесткого диска



Указать размер жесткого диска (не менее 10Гб)

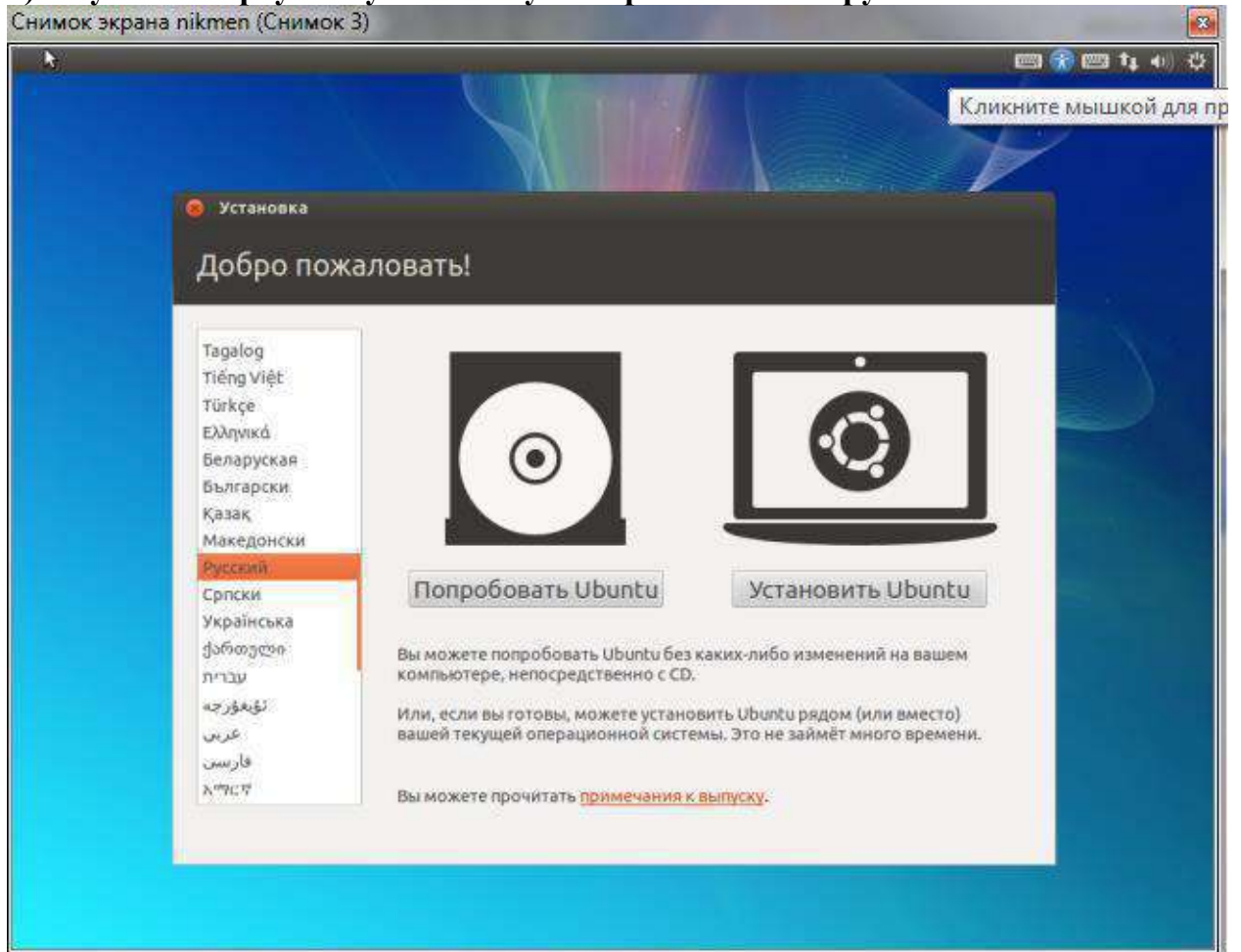


Дождаться установки виртуальной машины. Открыть окно свойств виртуальной машины и перейти в раздел носители. Выбрать IDE контроллер – пусто. Кликнуть по иконке диска в правой панели и выполнить «Выбрать образ загрузочного диска». Указать путь к образу диска Ubuntu и сохранить настройки.





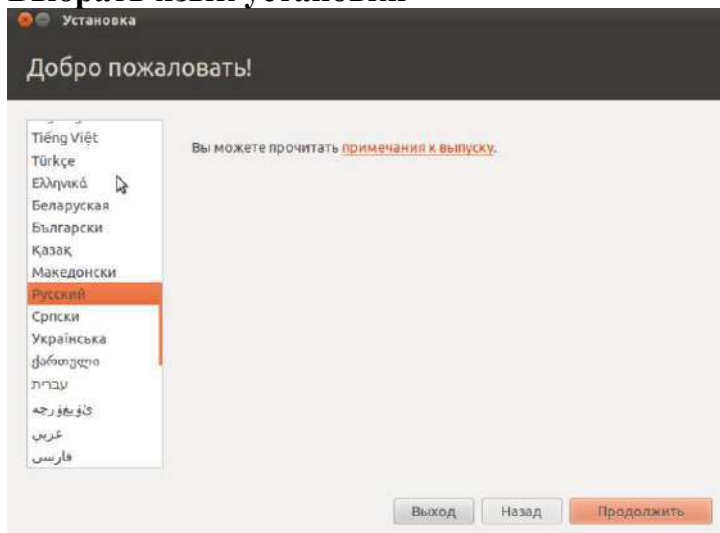
2) Запустить виртуальную машину. Выбрать в списке русский язык.



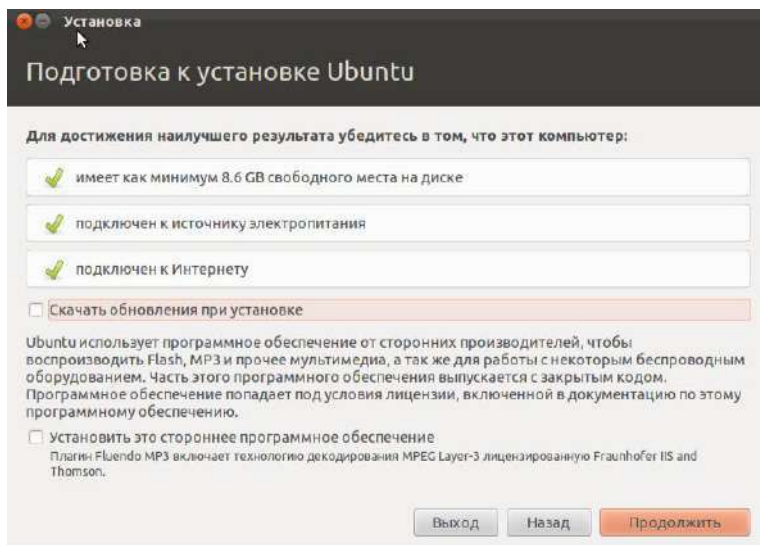
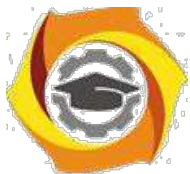
Из предложенных вариантов выбрать «Запустить Ubuntu без установки»
Дождаться загрузки операционной системы. Запустить установка кликнув по
ярлыку «Установить Ubuntu»



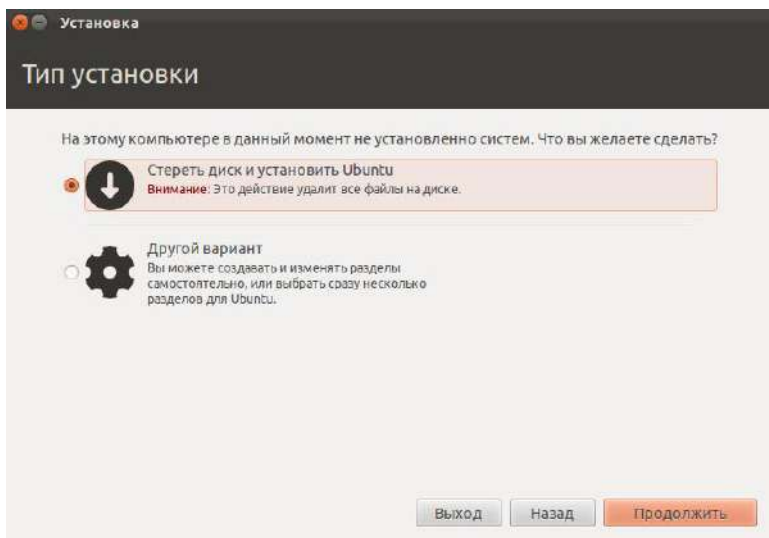
Выбрать язык установки



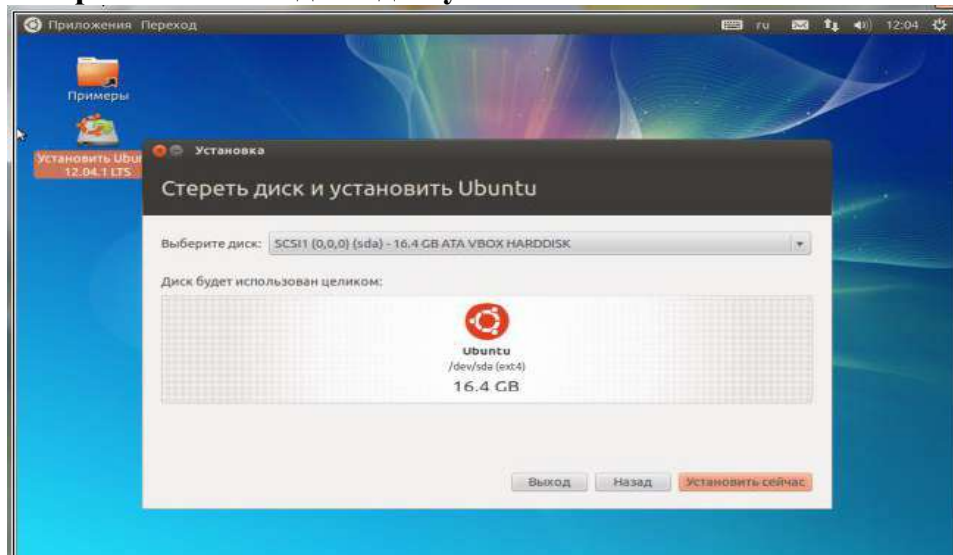
Подтвердить требования к установке



Выбрать тип установки



Выбрать жесткий диск для установки





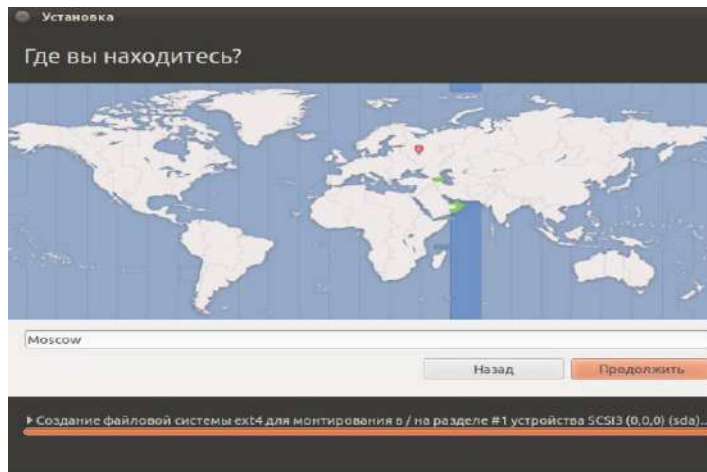
**Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»**

Указать региональные параметры

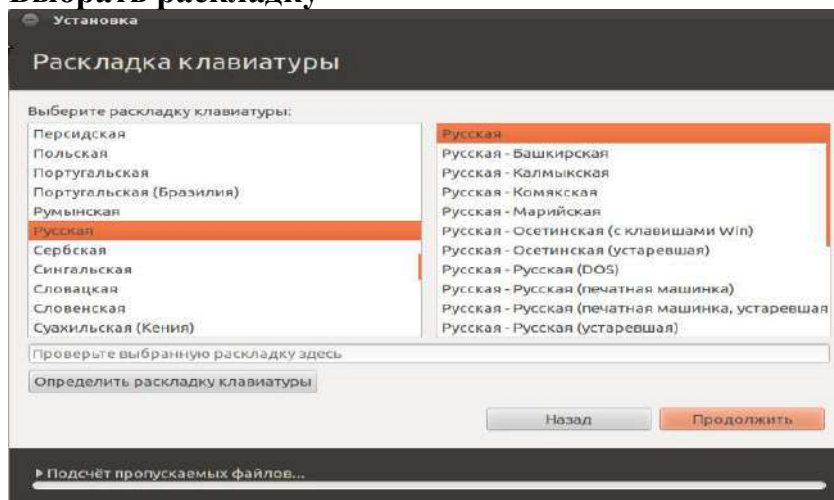


Негосударственное частное образовательное
учреждение
высшего образования
«Технический университет»

13

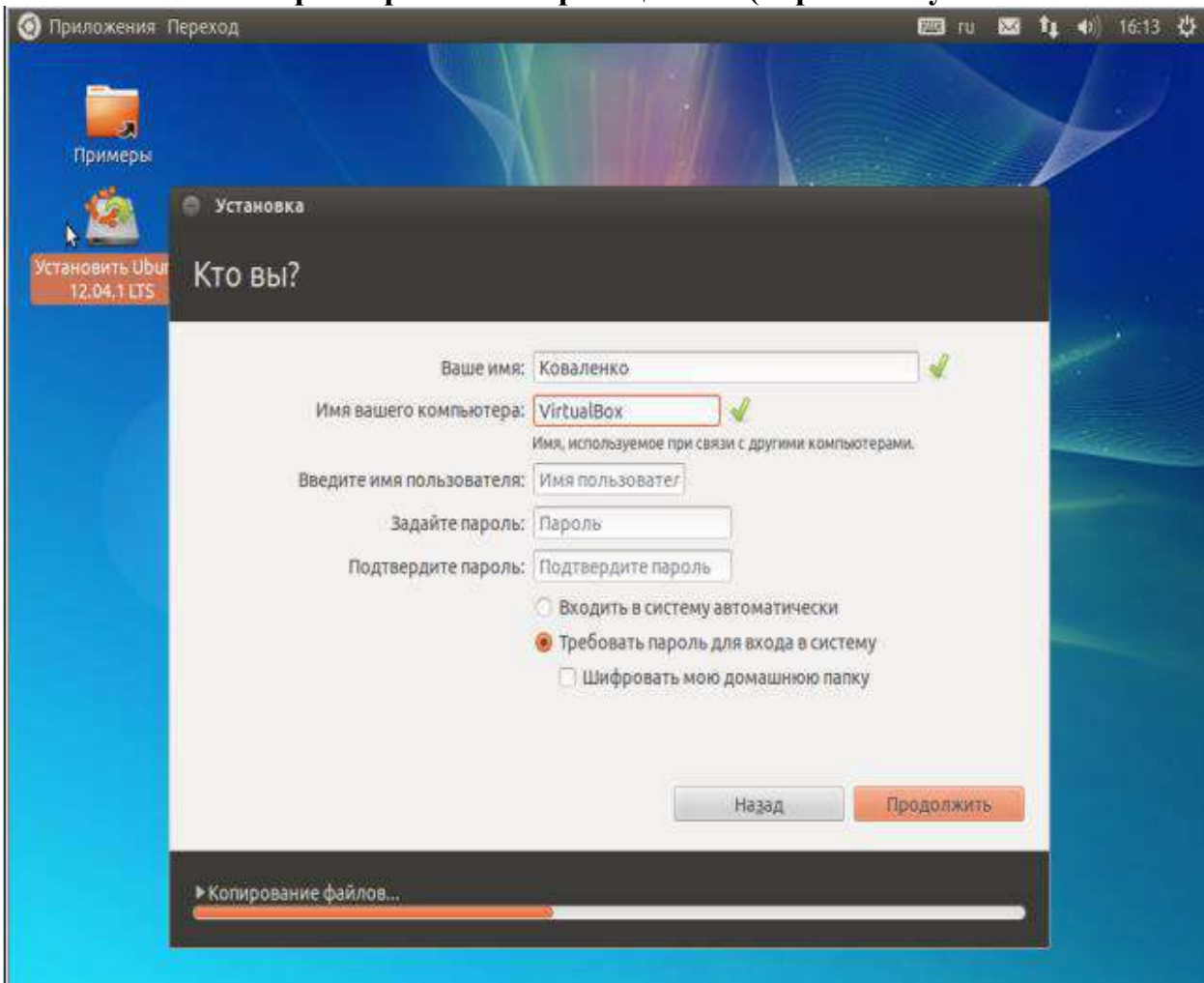


Выбрать раскладку





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.

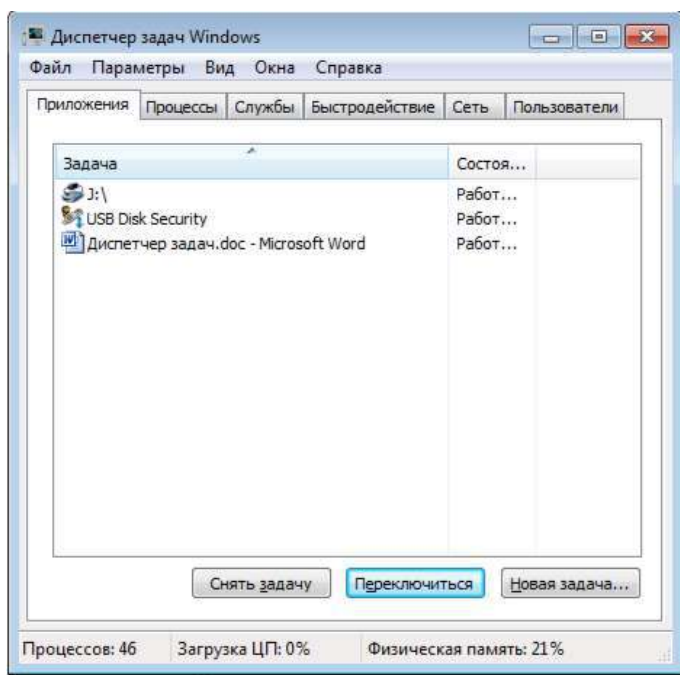


Рисунок 1 – Окно программы Диспетчер задач Windows

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.

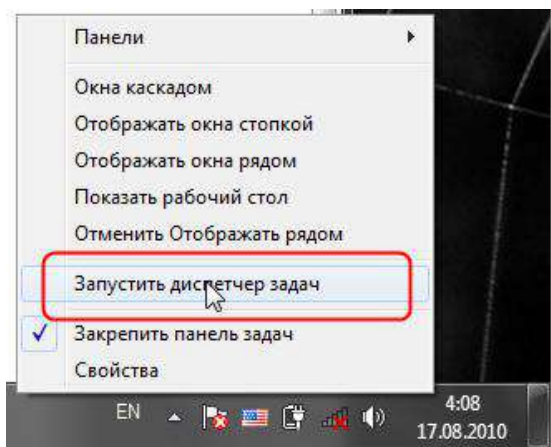


Рисунок 2 – Окно запуска диспетчера задач



**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.

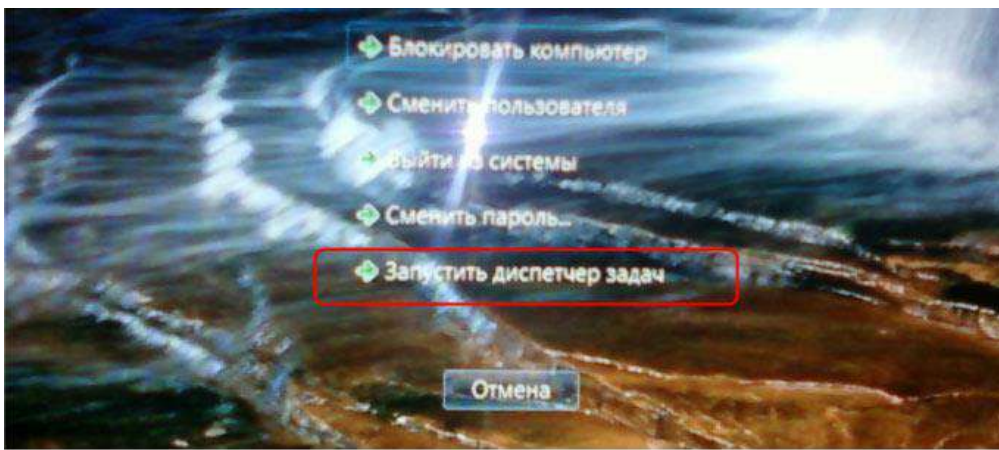
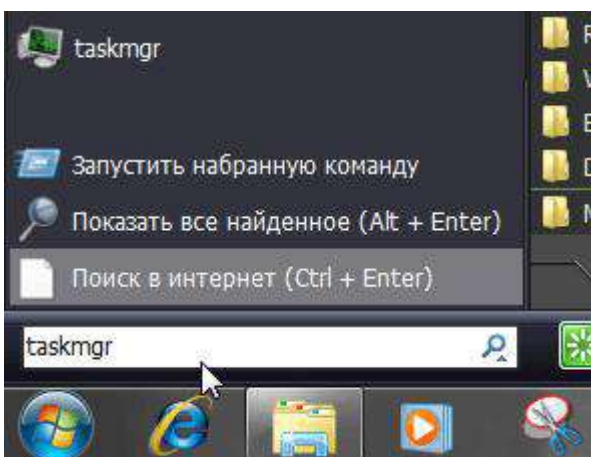


Рисунок 3 – Меню команд

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:

2. Вкладка *Приложения*. Здесь можно найти список запущенных приложений и их состояние.
3. Вкладка *Процессы*. Перечислены все программы и процессы, запущенные в системе, – это основное окно для прекращения работы «зависших» программ или процессов.
4. Вкладка *Службы*. Содержит список программ, которые работают в фоновом режиме.
5. Вкладка *Быстродействие*. Основная вкладка для оценки производительности операционной системы.
6. Вкладка *Сеть*. Отображает объём передаваемых по локальной сети данных.
7. 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».

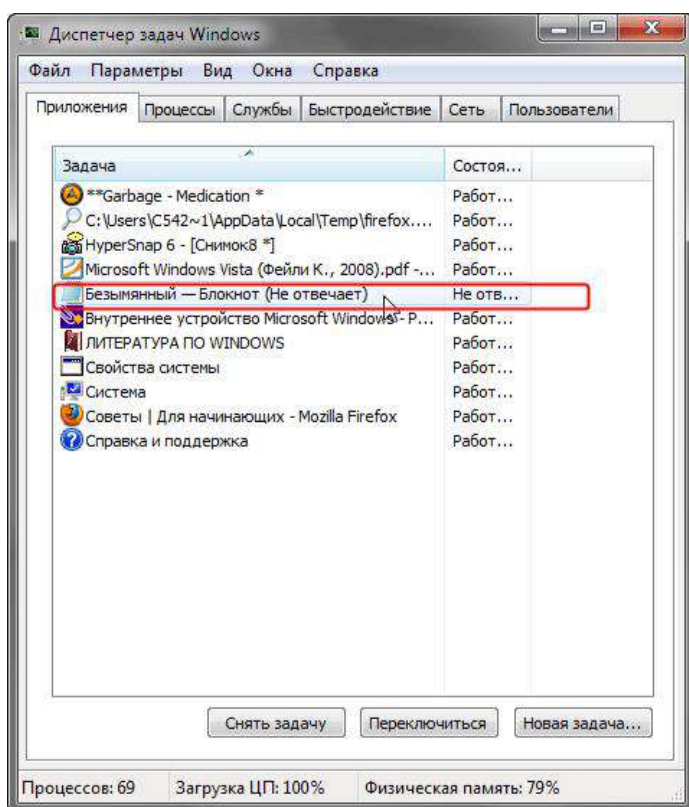


Рисунок 5 – Вкладка «Приложения»

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.

3. 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.

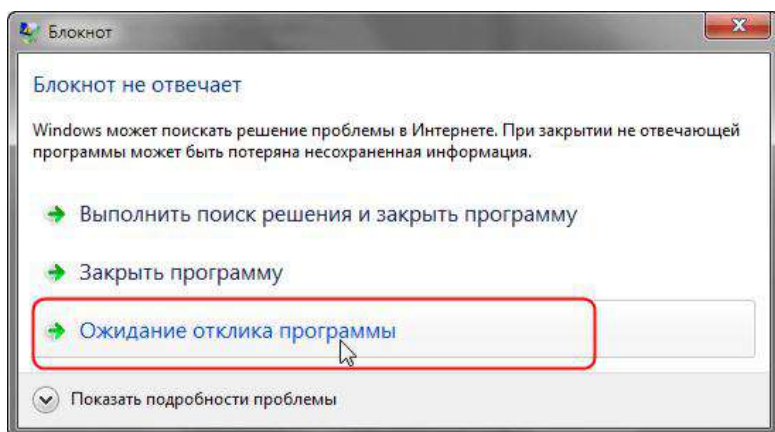


Рисунок 6 – Ожидание отклика программы

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.

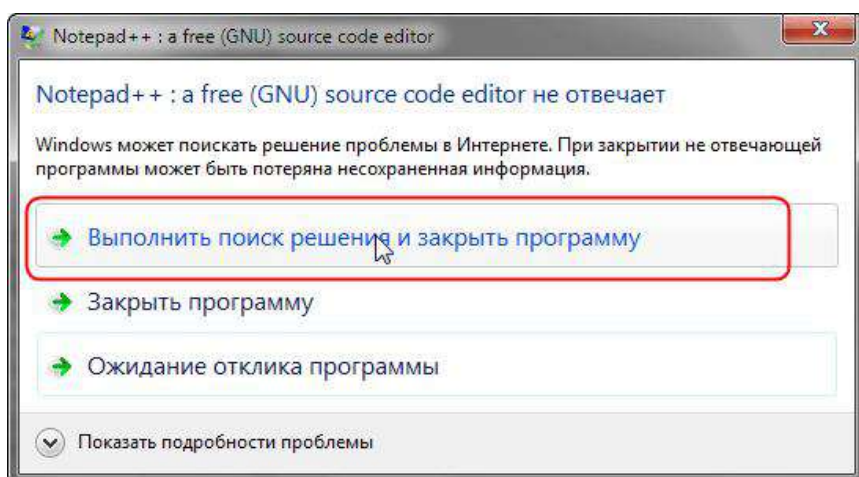




Рисунок 7 – Выполнить поиск решения и закрыть программу

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, а 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.

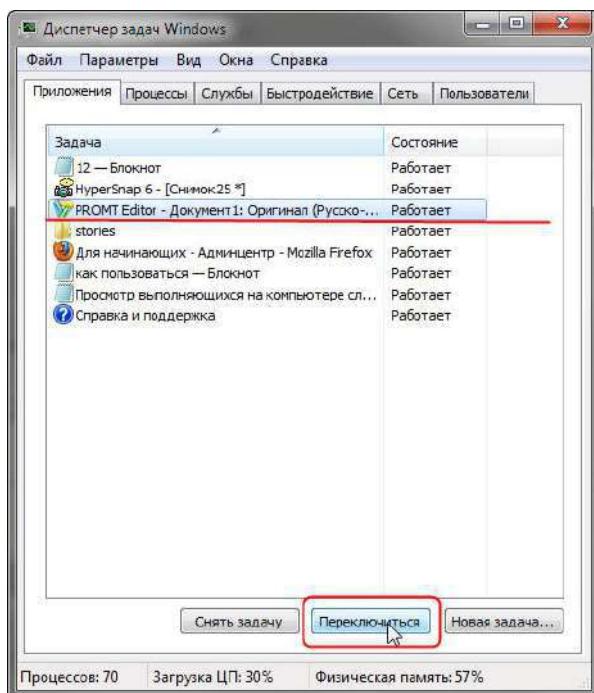


Рисунок 8 – Кнопка переключиться

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.

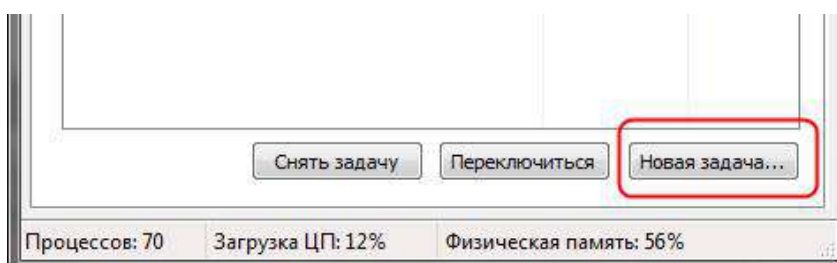




Рисунок 9 – Новая задача

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.

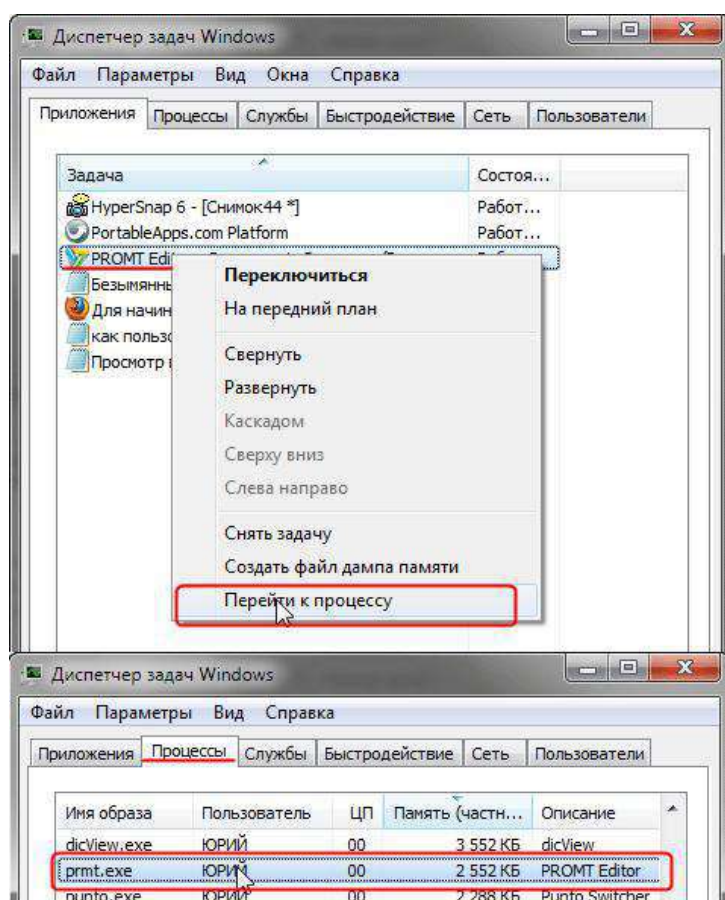


Рисунок 10 – Перейти к процессу

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.

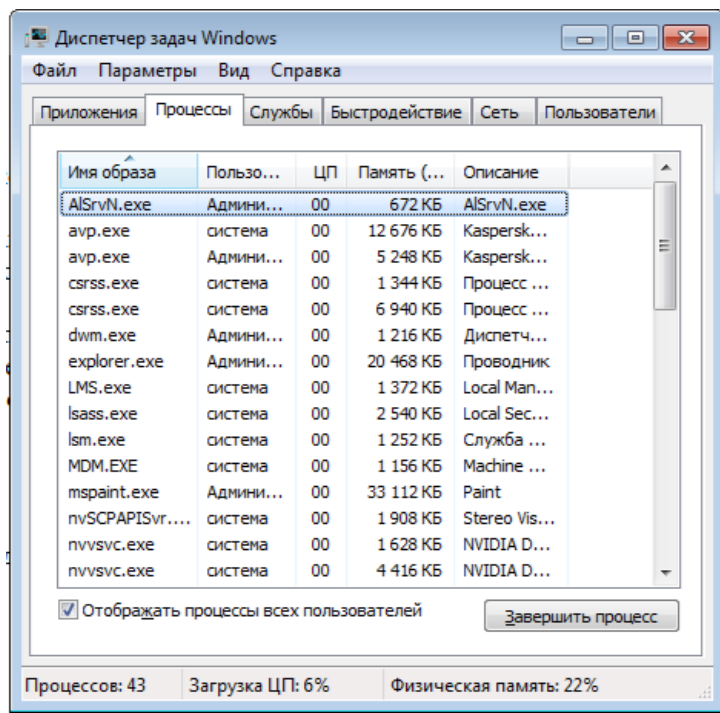


Рисунок 11 – Вкладка «Процессы»

Сколько процессов выполняется на данный момент времени, можно увидеть в нижней части диспетчера.

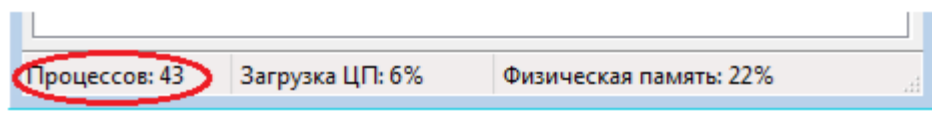


Рисунок 12 – Наличие процессов

- 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:

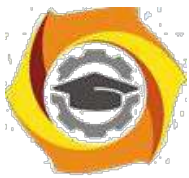
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Служба

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 уничтожало 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.

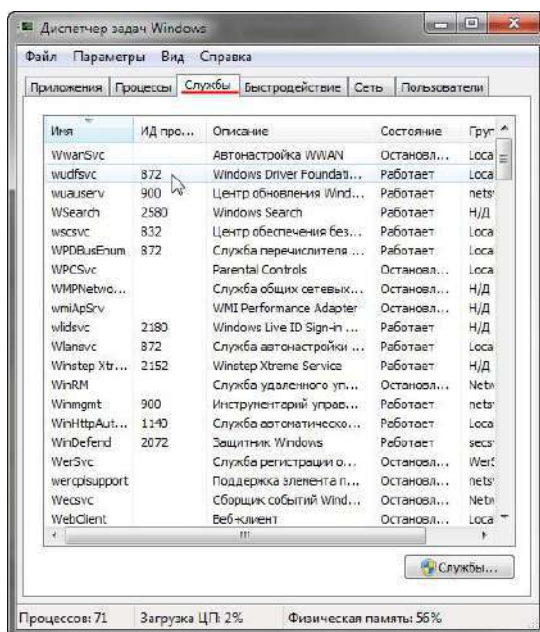


Рисунок 14 – Вкладка «Службы»



**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.

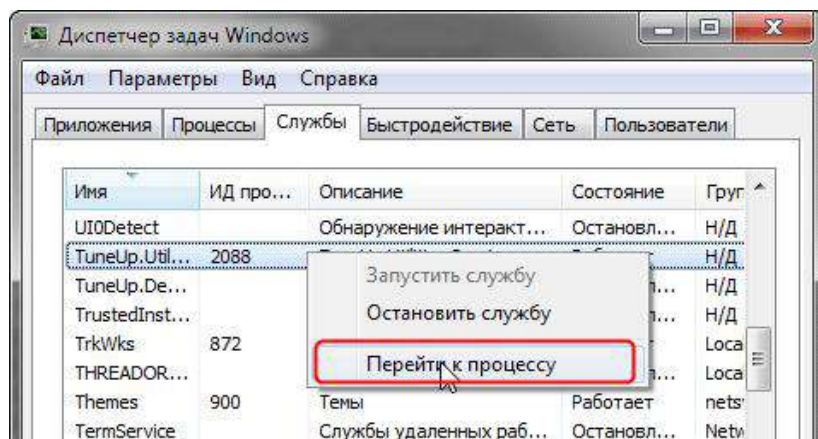


Рисунок 15 – Процесс связанный со службой

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.

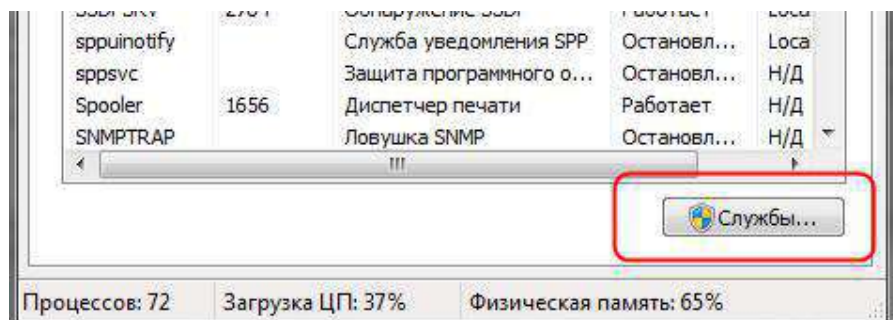


Рисунок 16 – Кнопка Службы

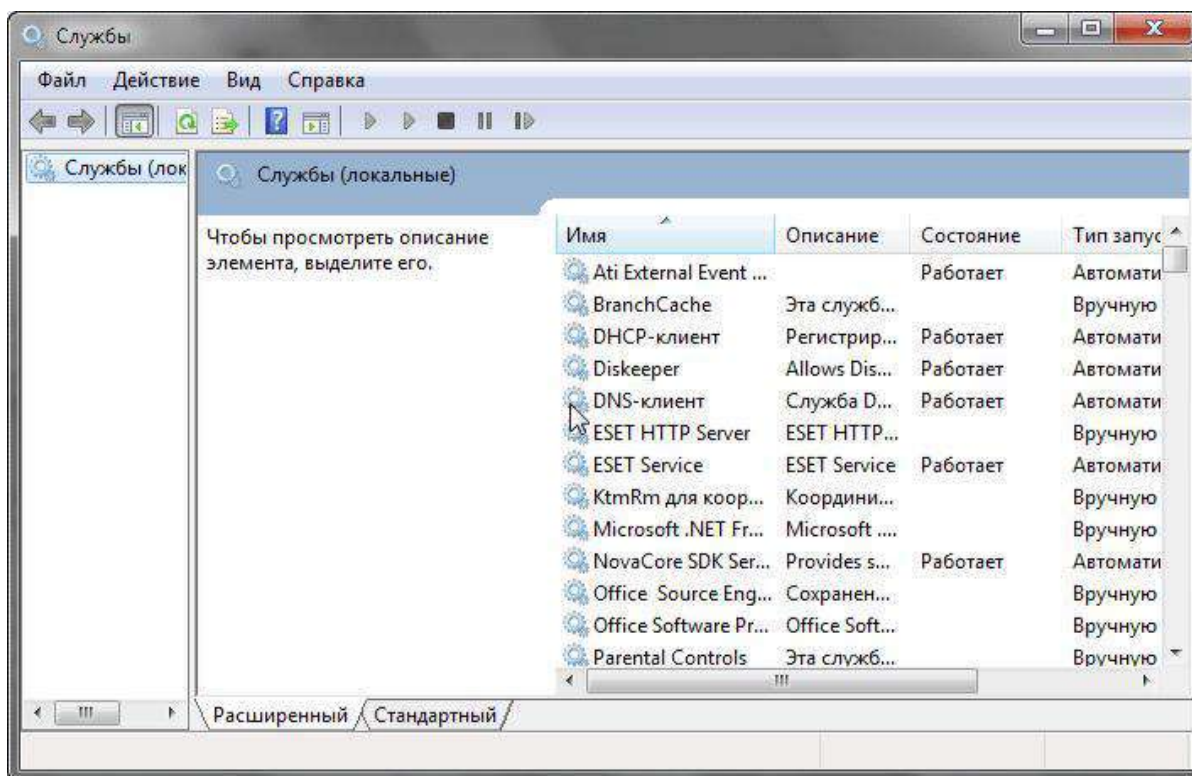


Рисунок 17 – Подробные сведения о
службах Вкладка «Быстродействие»

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.

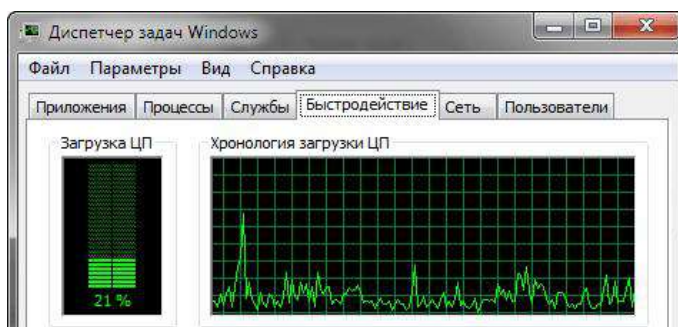


Рисунок 18 – Вкладка «Быстродействие»

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.

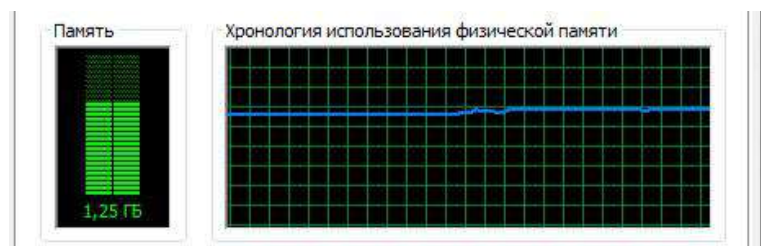


Рисунок 20 – Файл подкачки и хронология использования файла подкачки

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.

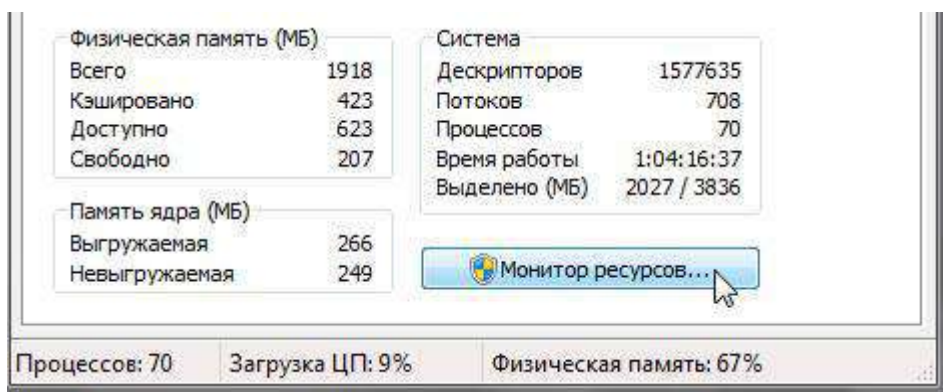


Рисунок 22 – Сведения об использовании памяти и ресурсов

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.

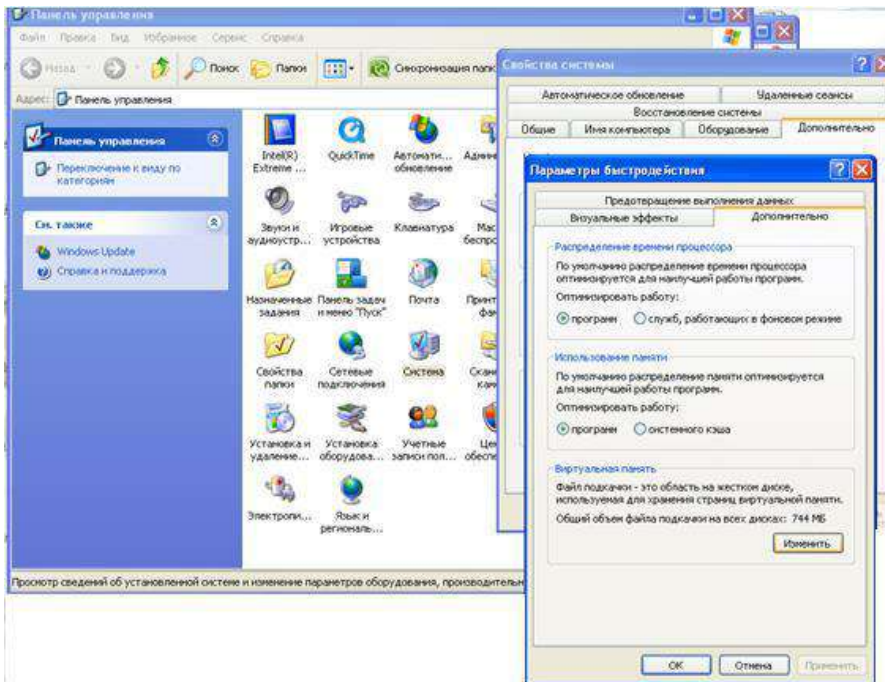
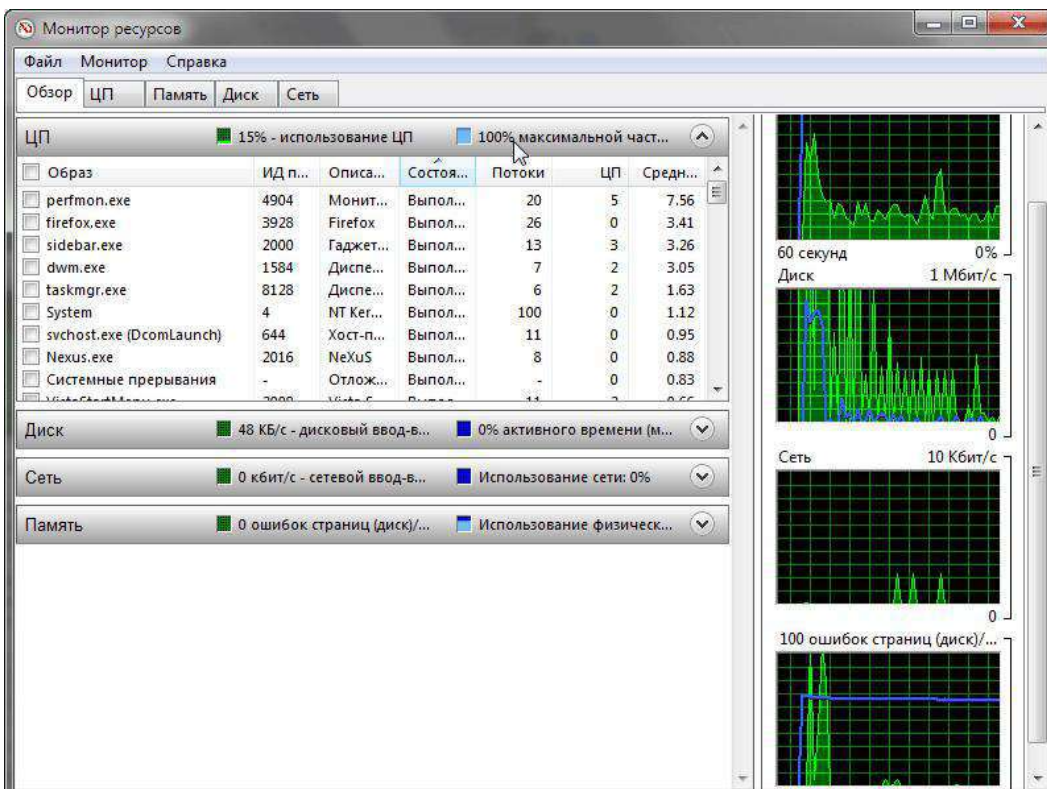


Рисунок 23 – изменение Файла подкачки

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.

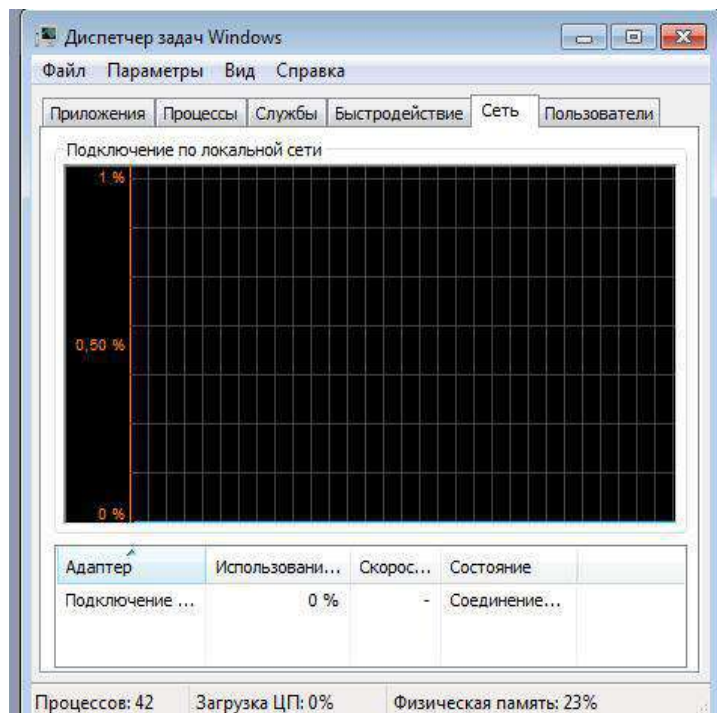
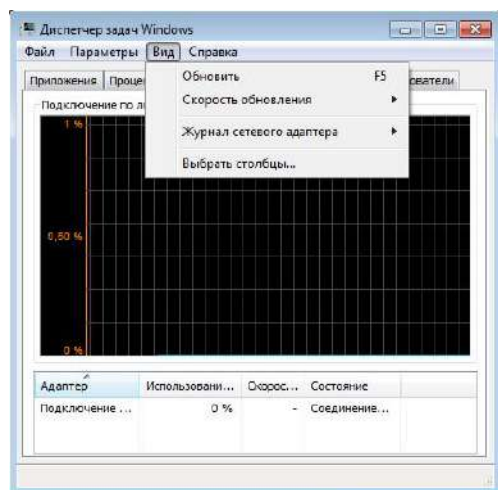


Рисунок 25 – Вкладка «Сеть»

4. 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



4. с помощью команды **Вид** → **Журнал сетевого адаптера** можно включать или выключать отображение дополнительных графиков.

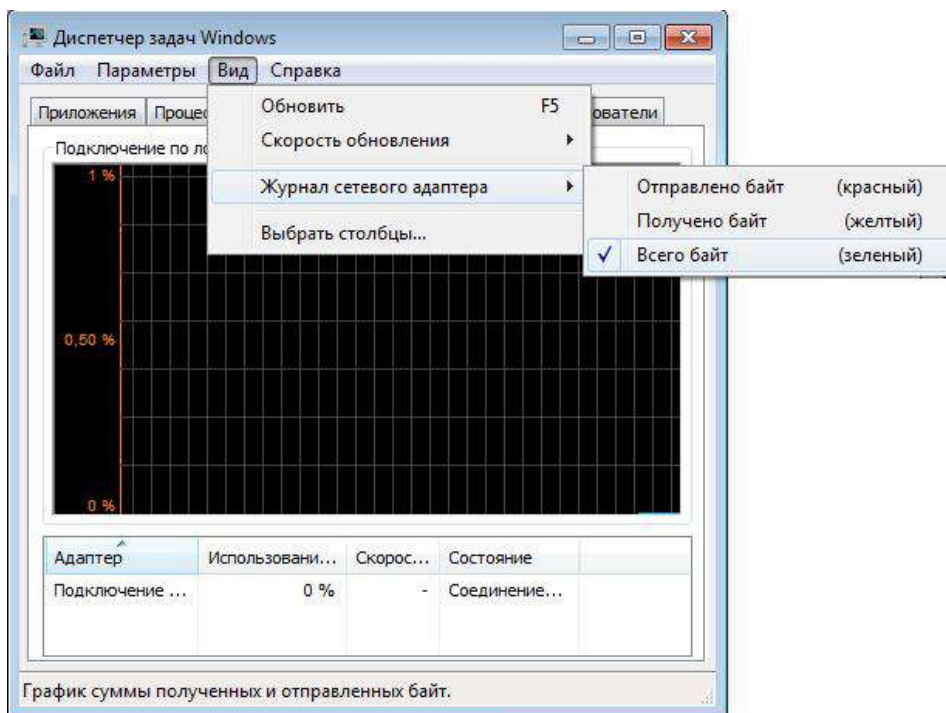


Рисунок 27 – Журнал сетевого адаптера

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.

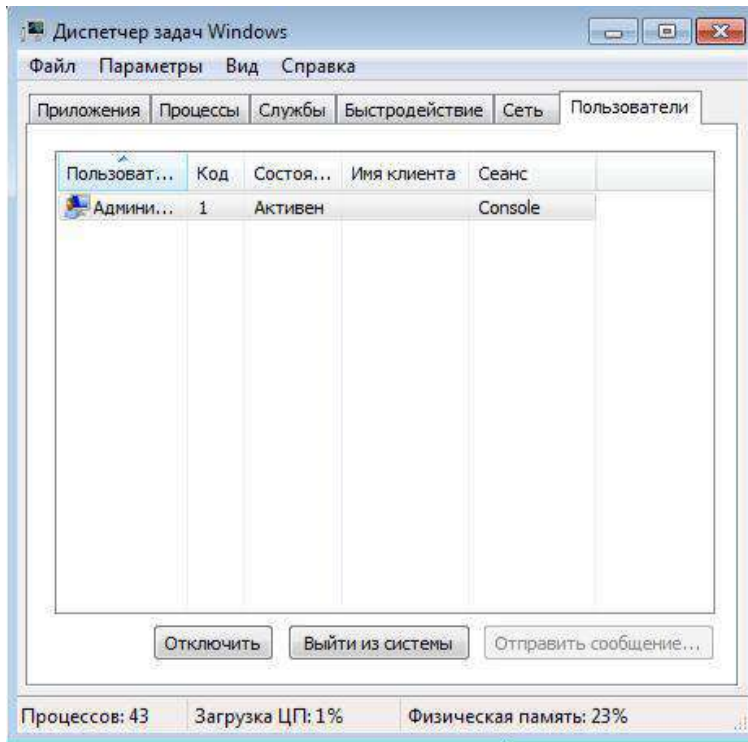


Рисунок 28 – Вкладка «Пользователь»

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.