

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Лапин Вячеслав Александрович
Должность: Директор
Дата подписания: 29.09.2026 13:25:45
Уникальный программный ключ:
df48b51be157e2f6cf8adf83bc04ff59a6aeacac

ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ
УГМК



Негосударственное частное образовательное учреждение высшего образования
«Технический университет УГМК»
(НЧОУ ВО «ТУ УГМК»)



УТВЕРЖДАЮ
Директор НЧОУ ВО «ТУ УГМК»

В.А. Лапин
(инициалы, фамилия)

« 1 » сентября 2026 г.

Дополнительная общеобразовательная общеразвивающая ПРОГРАММА

«Язык программирования Python Start»

(наименование программы)

Направленность: информатика и вычислительная техника

Возраст обучающихся: 12-17 лет

Срок реализации: 8 месяцев

Автор-разработчик:
Антипин Алексей Александрович

Верхняя Пышма, 2026 год

Оглавление

1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ. **Ошибка! Закладка не определена.**

1.1. Пояснительная записка	Ошибка! Закладка не определена.
1.2. Цель и задачи общеразвивающей программы	6
1.3 Содержание общеразвивающей программы	7
1.4 Содержание учебного плана.....	8
1.5 Планируемые результаты	12
1.6 Календарный учебный график	13
2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ	16
2.1 Условия реализации программы	16
2.2 Форма аттестации/контроля и оценочные материалы	16
3. СПИСОК ЛИТЕРАТУРЫ	17

1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

1.1. Пояснительная записка

Направленность общеобразовательной программы:

Современный мир предъявляет новые требования к молодому поколению, вступающему в жизнь, так как будущее сегодняшних детей — это мир, в котором умение читать и писать код становится такой же частью базовой грамотности, как умение работать с текстом или таблицей. Информатизация образования открывает новые возможности для развития методов и организационных форм воспитания и обучения детей, и в этих условиях особенно важно научить ребёнка не запоминать готовые решения, а самостоятельно строить алгоритм и доводить его до работающей программы.

Python — образовательный инструмент, который позволяет учащимся с первого занятия получать видимый результат: программа запускается и делает то, что задумал ученик. Простой синтаксис языка снимает барьер входа, а широта его применения — от расчётов и обработки текста до графики, объектной модели и разработки игр — позволяет выстроить курс так, что каждый следующий модуль опирается на предыдущий и расширяет круг решаемых задач.

Для успешного обучения в школе и жизни в мире информационных технологий важен не столько набор знаний, сколько развитое мышление, умение получать знания и использовать имеющиеся навыки для решения различных учебных задач. Большие возможности при этом открываются при работе с языком программирования, где любая ошибка немедленно проявляется в поведении программы, а её поиск и исправление становятся полноценной учебной задачей.

Направленность дополнительной образовательной программы — техническая. Она заключается в развитии алгоритмического и технического мышления у детей среднего школьного возраста через практическое программирование: от линейных программ и ветвлений до собственных функций, модулей, классов и законченных игровых проектов. Также программа формирует первичные представления о культуре разработки: читаемости кода, тестировании, отладке и совместной работе над проектом.

Развитие навыков программирования является одним из способов формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования логического, алгоритмического и критического мышления.

Актуальность общеобразовательной программы

Python — один из самых востребованных языков программирования в мире: он применяется в веб-разработке, анализе данных, машинном обучении, автоматизации и разработке игр. Именно с него чаще всего начинают знакомство с профессией разработчика, и именно он используется в школьной информатике и на олимпиадах по программированию.

Курс «Программирование на Python» построен так, чтобы на каждом занятии учащийся не просто слушал про конструкции языка, а сразу писал, запускал и отлаживал собственный код. Программа не делит уроки на лекции и практику: объяснение и практическая работа происходят слитно в рамках одного занятия, при этом активная работа за клавиатурой занимает около 70 минут из 90.

Важным моментом подготовки детей к жизни в мире информационных технологий является развитие у них навыков самостоятельного поиска ошибки — умения прочесть сообщение интерпретатора, выдвинуть предположение о причине и проверить его. Обучение программированию через последовательное усложнение собственных проектов — устоявшаяся и результативная методика в дополнительном образовании.

Актуальность программы заключается в:

- необходимости расширения кругозора школьника в области современных информационных технологий;
- формировании навыков алгоритмического мышления и структурного программирования;
- формировании навыка самостоятельной отладки программы и поиска причины ошибки;
- формировании представлений об объектно-ориентированном подходе как основе современной разработки;
- формировании навыка доведения собственного проекта от замысла до работающего продукта;
- развитии цифровой и функциональной грамотности;
- развитии логического и критического мышления через практическую работу с кодом.

Новизна программы

Новизна программы заключается в методике проведения занятия: каждый урок построен вокруг работающего кода ученика, а не вокруг демонстрации преподавателя. Ученик обязан сначала предсказать результат программы вслух и

только затем запустить её, а преподаватель принципиально не показывает готовое решение проекта — вместо него на доске остаются только критерии готовности.

Каждое занятие включает отладочную лабораторию: ученик получает заведомо неработающий фрагмент кода с реалистичной ошибкой, находит причину по сообщению интерпретатора и исправляет её. Такой формат превращает ошибку из повода для расстройства в штатный рабочий инструмент.

Во второй половине курса акцент смещается с отдельных конструкций языка на архитектуру программы: учащиеся проектируют собственные классы, работают с коллекциями объектов, а в финале проходят полный цикл разработки игры — от прототипа и игровой системы до полировки, тестирования и публичной защиты проекта.

Адресат: учащиеся средних и старших классов школы в возрасте от 12 до 17 лет включительно.

Режим занятий: занятия проходят 1 раз в неделю. Дети занимаются в компьютерном классе с преподавателем, самостоятельно работая за компьютером в среде разработки.

Объём: 2 академических часа (1 академический час-45 минут).

Занятия проводятся в группах не более 15 человек и индивидуально. Язык обучения-русский.

Форма обучения:

- Каждое занятие объединяет объяснение и практику: отдельных лекционных занятий в программе нет — новая тема сразу закрепляется написанием собственного кода.
- Обязательный прогноз результата до запуска программы.
- Игровая, задачная и проектная форма работы.
- Обучение от простого к сложному: от вывода первой строки на экран до законченной игры с несколькими состояниями.
- Поощрение вопросов и свободных высказываний по теме.
- Уважение и внимание к каждому ученику.
- Создание мотивационной среды обучения.
- Создание условий для обсуждения найденных ошибок и способов их исправления вместо одностороннего объяснения темы преподавателем.

Основные формы работы — индивидуальная и групповая. По итогам каждого модуля учащиеся представляют итоговый проект модуля и получают обратную связь от преподавателя и группы.

1.2. Цель и задачи общеразвивающей программы

Цель курса — способствовать формированию у учащихся навыков структурного и объектно-ориентированного программирования на языке Python: от написания линейных программ и алгоритмов с ветвлениями до самостоятельной разработки, отладки и защиты собственного игрового проекта.

Задачи:

Обучающие:

1. Познакомить с базовым синтаксисом языка Python и типами данных.
2. Сформировать навык использования управляющих конструкций: ветвлений, циклов и их сочетаний.
3. Сформировать навык работы со списками как основной структурой хранения данных.
4. Сформировать навык декомпозиции задачи на функции и вынесения кода в собственные модули.
5. Сформировать навык построения алгоритмов для графического исполнителя и работы с системой координат.
6. Сформировать базовые представления об объектно-ориентированном программировании: класс, объект, атрибут, метод.
7. Сформировать навык разработки игрового приложения с использованием библиотеки Pygame.
8. Сформировать навык самостоятельной отладки программы и тестирования результата по заданным критериям.

Развивающие:

1. Сформировать и развивать алгоритмическое мышление при построении и проверке решения.
2. Сформировать навык публичного выступления и презентации собственного проекта.
3. Расширять кругозор, развивать умение формулировать задачу и разбивать её на проверяемые шаги.

4. Совершенствовать диалогическую речь учащихся: умение слушать собеседника, задавать вопросы и отвечать на них.

5. Формировать внимание к деталям и аккуратность при работе с формальными языками.

Воспитательные:

1. Воспитывать у учащихся потребность в сотрудничестве и взаимодействии со сверстниками при работе над групповыми проектами.

2. Формировать умение давать качественную обратную связь и реагировать на неё.

3. Формировать информационную культуру и уважение к чужому труду: понимание ценности читаемого кода и корректного заимствования решений.

1.3. Содержание общеразвивающей программы

Учебно-тематический план

№ п/п	Наименование раздела/темы	Всего	Теория	Практика	Форма аттестации (контроля)
Модуль 1. Основы языка					
1	Введение в Python. Функции ввода-вывода	2	1	1	Практическая работа: написание программы
2	Переменные. Численные типы данных	2	1	1	Практическая работа: написание программы
3	Строки	2	1	1	Практическая работа: написание программы
4	Вложенные конструкции	2	1	1	Практическая работа: написание программы
Модуль 2. Управляющие конструкции и списки					
5	Условный оператор	2	1	1	Практическая работа: написание программы
6	Вложенный условный оператор	2	1	1	Практическая работа: написание программы
7	Циклы. Часть 1: while	2	1	1	Практическая работа: написание программы
8	Циклы. Часть 2: for и range	2	1	1	Практическая работа: написание программы
9	Вложенные управляющие конструкции	2	1	1	Практическая работа: написание программы

№ п/п	Наименование раздела/темы	Всего	Теория	Практика	Форма аттестации (контроля)
10	Списки	2	1	1	Практическая работа: написание программы
Модуль 3. Функции и модули					
11	Функции. Часть 1: объявление и параметры	2	1	1	Практическая работа: написание программы
12	Функции. Часть 2: возврат значения	2	1	1	Практическая работа: написание программы
13	Модули random и time	2	1	1	Практическая работа: написание программы
14	Создание модулей	2	1	1	Практическая работа: написание модуля
Модуль 4. Turtle. Математика для разработчика					
15	Turtle: линейные алгоритмы	2	1	1	Практическая работа: построение изображения
16	Turtle: циклы	2	1	1	Практическая работа: построение изображения
17	Turtle: условный оператор	2	1	1	Практическая работа: построение изображения
18	Проект «Приборная панель станции»	2	1	1	Презентация проекта
Модуль 5. Объектно-ориентированное программирование					
19	Классы и объекты	2	1	1	Практическая работа: написание программы
20	Атрибуты и конструктор __init__	2	1	1	Практическая работа: написание программы
21	Методы объекта	2	1	1	Практическая работа: написание программы
22	Список объектов	2	1	1	Практическая работа: написание программы
23	Взаимодействие объектов	2	1	1	Практическая работа: написание программы
24	Проект «Питомник роботов»	2	1	1	Презентация проекта
Модуль 6. Разработка игр на Pygame					
25	Основы Pygame: окно и игровой цикл	2	1	1	Практическая работа: разработка приложения
26	Объекты в игре: класс и коллекция	2	1	1	Практическая работа: разработка приложения
27	Игра «Энергоблок». Часть 1: реактор и утка	2	1	1	Практическая работа: разработка игры

№ п/п	Наименование раздела/темы	Всего	Теория	Практика	Форма аттестации (контроля)
28	Игра «Энергоблок». Часть 2: улучшения	2	1	1	Презентация проекта
29	Игра «Космический мусорщик». Часть 1: сборщик и обломок	2	1	1	Практическая работа: разработка игры
30	Игра «Космический мусорщик». Часть 2: поток обломков и счёт	2	1	1	Практическая работа: разработка игры
31	Игра «Космический мусорщик». Часть 3: жизни и состояния	2	1	1	Презентация проекта
Модуль 7. Серия ИТ-марафонов					
32	Воркшоп: навыки разработчика	2	1	1	Практическая работа: разбор и тестирование кода
33	ИТ-марафон. Часть 1: прототип	2	1	1	Практическая работа: разработка проекта
34	ИТ-марафон. Часть 2: игровая система	2	1	1	Практическая работа: разработка проекта
35	ИТ-марафон. Часть 3: полировка и релиз	2	1	1	Практическая работа: разработка проекта
Модуль 8. Выпускной					
36	Выпускной: демонстрация и код-клиника	2	1	1	Презентация и защита проекта

В течение курса ученики создадут не менее 30 индивидуальных мини-проектов, объединяющих знания и умения соответствующего занятия, и не менее 4 крупных проектов по итогам модулей.

1.4. Содержание учебного плана

Каждое занятие построено вокруг работающего кода ученика: объяснение новой темы сразу закрепляется написанием, запуском и отладкой собственной программы — отдельных лекционных занятий программа не предусматривает. Активная работа за клавиатурой занимает около 70 минут из 90.

Модуль 1. Основы языка

1.1. Введение в Python. Функции ввода-вывода

Язык программирования, алгоритм, программа, синтаксис. Ученики знакомятся со средой разработки, выводят данные функцией `print()` и выполняют вычисления с арифметическими операторами, включая целочисленное деление и остаток.

1.2. Переменные. Численные типы данных

Переменная, присваивание, типы `int` и `float`, преобразование типов. Ученики сохраняют данные в переменных, получают их с клавиатуры функцией `input()` и собирают расчётную программу, преобразуя ввод в число.

1.3. Строки

Строка, конкатенация, f-строка, длина строки, методы обработки текста. Ученики собирают осмысленный текстовый вывод из данных пользователя и приводят введённый текст к нужному виду.

1.4. Вложенные конструкции

Порядок вычисления выражений, вложенный вызов функции, промежуточные переменные. Ученики читают вложенные выражения изнутри наружу и осознанно выбирают между короткой записью и читаемой.

Модуль 2. Управляющие конструкции и списки

2.1. Условный оператор

Логическое условие, операторы сравнения, конструкция `if/elif/else`, отступы, граничные значения. Ученики программируют выбор одного из нескольких вариантов и проверяют программу на границах диапазонов.

2.2. Вложенный условный оператор

Вложенная ветвь, логические операторы `and`, `or`, `not`, дерево решений. Ученики строят двухуровневые проверки и объединяют условия, предварительно нарисовав дерево вариантов.

2.3. Циклы. Часть 1: while

Цикл с условием, счётчик, накопитель, бесконечный цикл. Ученики программируют повторение действий до наступления события и учатся находить причину зависшей программы.

2.4. Циклы. Часть 2: for и range

Цикл по последовательности, функция `range()`, шаг диапазона. Ученики применяют цикл для известного числа повторов и рассчитывают количество итераций до запуска программы.

2.5. Вложенные управляющие конструкции

Условие внутри цикла, вложенный цикл, операторы `break` и `continue`. Ученики обрабатывают двумерные структуры и оценивают количество выполнений внутреннего блока.

2.6. Списки

Список, индекс, длина, перебор, добавление и удаление элементов. Ученики хранят набор значений в одной переменной и вычисляют по нему сводные показатели.

Модуль 3. Функции и модули

3.1. Функции. Часть 1: объявление и параметры

Оператор `def`, параметр и аргумент, вызов функции, локальные переменные. Ученики заменяют повторяющиеся участки кода одним вызовом собственной функции.

3.2. Функции. Часть 2: возврат значения

Оператор `return`, результат функции, ранний выход, композиция функций. Ученики собирают сложные вычисления из простых проверяемых функций.

3.3. Модули `random` и `time`

Стандартная библиотека, оператор `import`, генерация случайных чисел, работа со временем. Ученики создают программу, поведение которой меняется от запуска к запуску, и измеряют время реакции пользователя.

3.4. Создание модулей

Собственный модуль, импорт функций, пространство имён, конструкция `__name__`. Ученики разделяют программу на несколько файлов и оформляют собственную библиотеку функций.

Модуль 4. Turtle. Математика для разработчика

4.1. Turtle: линейные алгоритмы

Графический исполнитель, движение и повороты, перо, система координат. Ученики строят простые фигуры последовательностью команд и переходят между частями рисунка без следа.

4.2. Turtle: циклы

Повторяющиеся элементы фигуры, правило поворота $360/n$, вложенные циклы в графике. Ученики сокращают графический алгоритм циклом и выводят угол правильного многоугольника формулой.

4.3. Turtle: условный оператор

Условие в графическом цикле, чередование, текущие координаты исполнителя, координатные четверти. Ученики программируют рисунок, вид которого зависит от данных и положения на экране.

4.4. Проект «Приборная панель станции»

Ученики собирают законченную графическую сцену из собственных функций: корпус панели, ряд круглых шкал и столбчатые индикаторы, цвет

которых определяется превышением порогового значения, — и презентуют результат группе.

Модуль 5. Объектно-ориентированное программирование

5.1. Классы и объекты

Класс как описание, объект как экземпляр, атрибут, независимость объектов. Ученики создают первый собственный класс и несколько независимых объектов на его основе.

5.2. Атрибуты и конструктор `__init__`

Метод `__init__`, параметр `self`, значения по умолчанию. Ученики задают все атрибуты объекта сразу при создании и сравнивают такой способ с ручным заполнением.

5.3. Методы объекта

Метод класса, изменение состояния через `self`, возврат значения из метода. Ученики наделяют объект собственными действиями и защищают его состояние проверками.

5.4. Список объектов

Коллекция объектов, перебор с обращением к атрибутам, фильтрация, безопасное удаление. Ученики управляют множеством однотипных объектов как единым целым.

5.5. Взаимодействие объектов

Объект как параметр метода, различение `self` и другого объекта, расстояние между объектами. Ученики программируют влияние одного объекта на состояние другого.

5.6. Проект «Питомник роботов»

Ученики самостоятельно проектируют класс под собственную задачу, создают коллекцию объектов, реализуют взаимодействие между ними и представляют группе итоговую сводку по коллекции.

Модуль 6. Разработка игр на Pygame

6.1. Основы Pygame: окно и игровой цикл

Библиотека Pygame, окно приложения, игровой цикл, обработка событий, ограничение частоты кадров. Ученики создают первое интерактивное приложение с объектом, управляемым с клавиатуры.

6.2. Объекты в игре: класс и коллекция

Игровой объект как класс, метод обновления состояния, коллекция объектов на экране. Ученики применяют объектный подход к анимации множества элементов.

6.3. Игра «Энергоблок». Часть 1: реактор и утечка

Обработка событий мыши, активная зона, вывод текста на экран, действия по таймеру. Ученики собирают прототип игры, в которой ресурс не только накапливается, но и постоянно убывает.

6.4. Игра «Энергоблок». Часть 2: улучшения

Система улучшений, стоимость и проверка средств, рост цены, балансировка. Ученики реализуют два разных по эффекту улучшения и представляют готовую игру группе.

6.5. Игра «Космический мусорщик». Часть 1: сборщик и обломок

Движение объектов по разным осям, удержание в границах окна, повторное появление объекта. Ученики собирают основу второй игры модуля.

6.6. Игра «Космический мусорщик». Часть 2: поток обломков и счёт

Коллекция падающих объектов, виды груза, проверка столкновений, подсчёт очков. Ученики превращают одиночный объект в поток и программируют осмысленный выбор игрока.

6.7. Игра «Космический мусорщик». Часть 3: жизни и состояния

Система жизней, состояния игры, функция полного перезапуска, рост сложности. Ученики доводят игру до законченного вида и представляют её группе.

Модуль 7. Серия ИТ-марафонов

7.1. Воркшоп: навыки разработчика

Уточнение задачи, декомпозиция, минимальный воспроизводимый пример, проверки `assert`, код-ревью. Ученики проходят полный цикл разработки маленького модуля и проводят разбор чужого кода по контрольным вопросам.

7.2. ИТ-марафон. Часть 1: прототип

Игровой замысел, вертикальный срез, критерии готовности, структура проекта. Ученики проектируют собственную игру и собирают её первую полностью работающую версию.

7.3. ИТ-марафон. Часть 2: игровая система

Препятствия, ресурс, безопасная генерация объектов, условие поражения. Ученики развивают прототип до повторяемой игровой системы с осмысленной сложностью.

7.4. ИТ-марафон. Часть 3: полировка и релиз

Экраны начала и конца, регрессионное тестирование, приоритизация дефектов, подготовка демонстрации. Ученики доводят проект до состояния, готового к показу постороннему человеку.

Модуль 8. Выпускной

8.1. Выпускной: демонстрация и код-клиника

Ученики представляют финальную версию собственного проекта, объясняют ключевое техническое решение и найденный дефект, участвуют в разборе кода группы, принимают обратную связь по критериям и формулируют план дальнейшего развития проекта.

1.5. Планируемые результаты

Планируемый результат	Способ достижения	Критерий достижения образовательного результата
Личностные результаты обучения		
Освоение социальной роли обучающегося и формирование личностного смысла учения.	Демонстрация связи между умением построить алгоритм и способностью получить работающую программу.	За отведённое время ученик пытается не только выполнить базовое задание, но и улучшить программу за счёт дополнительных проверок и более понятного кода.
Развитие навыков сотрудничества со взрослыми и сверстниками, умение находить выход из спорных ситуаций.	Работа в парах при поиске ошибки и групповые проекты как способ показать преимущества совместной разработки.	Ученик не боится просить помощи и сам пытается помогать одноклассникам и преподавателю.
Овладение начальными навыками адаптации в динамично изменяющемся мире информационных технологий.	Обсуждение областей применения Python и знакомство со сторонними библиотеками в ходе проектной работы.	Ученик самостоятельно интересуется возможностями языка и пробует конструкции за пределами показанных на занятии.
Метапредметные навыки		
Развитие и формирование учебных действий.	Создание условий для обсуждения найденных ошибок, продуктивное взаимодействие со сверстниками при разработке проекта.	Ученик легко общается, не боится просить помощь или оказывать её другим, способен кооперироваться для достижения общей цели.
Умение презентовать свою работу.	Презентация итоговых проектов модулей, обратная связь от преподавателя и других учеников.	Ученик во время презентации своего проекта уверенно удерживает внимание аудитории и объясняет принятые решения.

Планируемый результат	Способ достижения	Критерий достижения образовательного результата
Развитие критического мышления учеников.	Обязательный прогноз результата до запуска программы и последующее сравнение прогноза с фактом.	Ученик умеет объяснить, почему программа работает верно или ошибочно, и предлагает проверку, подтверждающую исправление.
Развитие творческих способностей учеников.	Планирование и реализация собственного игрового проекта по личному замыслу в рамках ИТ-марафона.	Ученик самостоятельно придумывает, обсуждает, планирует и реализует собственный проект.
Предметные навыки		
Усвоение и применение базовых навыков работы с ПК и средой разработки.	Повторение рутинных действий по созданию, сохранению и запуску файлов программы на каждом занятии.	Ученик самостоятельно создаёт, сохраняет и запускает программу, ориентируется в интерфейсе среды разработки.
Сформированность навыка использования базовых конструкций языка.	Выполнение практических заданий на переменные, ветвления, циклы и списки.	Ученик самостоятельно выбирает подходящую конструкцию под задачу и корректно применяет её.
Сформированность навыка декомпозиции задачи.	Выполнение практических заданий по выделению функций и вынесению кода в собственные модули.	Ученик умеет разбить задачу на функции с понятными именами и проверить каждую отдельно.
Сформированность базовых представлений об объектно-ориентированном программировании.	Проектирование собственных классов и работа с коллекциями объектов в модулях 5 и 6.	Ученик может объяснить своими словами, что такое класс, объект, атрибут и метод, и применить их в своей программе.
Сформированность навыка разработки игрового приложения.	Выполнение практических заданий по созданию игрового цикла, обработке событий и управлению состояниями игры.	Ученик умеет самостоятельно собрать интерактивное приложение с игровым циклом, счётом и несколькими состояниями.
Сформированность навыка отладки и тестирования.	Отладочная лаборатория на каждом занятии и проверка проекта по заранее заданным критериям.	Ученик способен по сообщению об ошибке найти её причину, исправить и подтвердить исправление проверкой.

В течение курса ученики создадут не менее 30 индивидуальных мини-проектов и не менее 2 групповых проектов.

1.6. Календарный учебный график

Количество учебных недель по программе – 32 недели

Количество учебных дней – 32 дня

Форма занятий интерактивная, включает в себя объяснение, практическую работу в среде разработки, разбор кода, отладку, демонстрацию результатов работы и рефлекссию.

№ п/п	Неделя	Форма занятия	Всего часов	Название темы	Место проведения	Форма контроля
1	1	интерактивная	2	Введение в Python. Функции ввода-вывода	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
2	2	интерактивная	2	Переменные. Численные типы данных	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
3	3	интерактивная	2	Строки	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
4	4	интерактивная	2	Вложенные конструкции	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
5	5	интерактивная	2	Условный оператор	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
6	6	интерактивная	2	Вложенный условный оператор	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
7	7	интерактивная	2	Циклы. Часть 1: while	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
8	8	интерактивная	2	Циклы. Часть 2: for и range	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
9	9	интерактивная	2	Вложенные управляющие конструкции	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
10	10	интерактивная	2	Списки	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
11	11	интерактивная	2	Функции. Часть 1: объявление и параметры	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
12	12	интерактивная	2	Функции. Часть 2: возврат значения	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
13	13	интерактивная	2	Модули random и time	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
14	14	интерактивная	2	Создание модулей	Компьютерный класс ТУ УГМК	Практическая работа: написание модуля
15	15	интерактивная	2	Turtle: линейные алгоритмы	Компьютерный класс ТУ УГМК	Практическая работа: построение изображения
16	16	интерактивная	2	Turtle: циклы	Компьютерный класс ТУ УГМК	Практическая работа: построение изображения
17	17	интерактивная	2	Turtle: условный оператор	Компьютерный класс ТУ УГМК	Практическая работа: построение изображения
18	18	интерактивная	2	Проект «Приборная панель станции»	Компьютерный класс ТУ УГМК	Презентация проекта
19	19	интерактивная	2	Классы и объекты	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
20	20	интерактивная	2	Атрибуты и конструктор __init__	Компьютерный класс ТУ УГМК	Практическая работа: написание программы

№ п/п	Неделя	Форма занятия	Всего часов	Название темы	Место проведения	Форма контроля
21	21	интерактивная	2	Методы объекта	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
22	22	интерактивная	2	Список объектов	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
23	23	интерактивная	2	Взаимодействие объектов	Компьютерный класс ТУ УГМК	Практическая работа: написание программы
24	24	интерактивная	2	Проект «Питомник роботов»	Компьютерный класс ТУ УГМК	Презентация проекта
25	25	интерактивная	2	Основы Pygame: окно и игровой цикл	Компьютерный класс ТУ УГМК	Практическая работа: разработка приложения
26	26	интерактивная	2	Объекты в игре: класс и коллекция	Компьютерный класс ТУ УГМК	Практическая работа: разработка приложения
27	27	интерактивная	2	Игра «Энергоблок». Часть 1: реактор и утечка	Компьютерный класс ТУ УГМК	Практическая работа: разработка игры
28	28	интерактивная	2	Игра «Энергоблок». Часть 2: улучшения	Компьютерный класс ТУ УГМК	Презентация проекта
29	29	интерактивная	2	Игра «Космический мусорщик». Часть 1: сборщик и обломок	Компьютерный класс ТУ УГМК	Практическая работа: разработка игры
30	30	интерактивная	2	Игра «Космический мусорщик». Часть 2: поток обломков и счёт	Компьютерный класс ТУ УГМК	Практическая работа: разработка игры
31	31	интерактивная	2	Игра «Космический мусорщик». Часть 3: жизни и состояния	Компьютерный класс ТУ УГМК	Презентация проекта
32	32	интерактивная	2	Воркшоп: навыки разработчика	Компьютерный класс ТУ УГМК	Практическая работа: разбор и тестирование кода
33	33	интерактивная	2	ИТ-марафон. Часть 1: прототип	Компьютерный класс ТУ УГМК	Практическая работа: разработка проекта
34	34	интерактивная	2	ИТ-марафон. Часть 2: игровая система	Компьютерный класс ТУ УГМК	Практическая работа: разработка проекта
35	35	интерактивная	2	ИТ-марафон. Часть 3: полировка и релиз	Компьютерный класс ТУ УГМК	Практическая работа: разработка проекта
36	36	интерактивная	2	Выпускной: демонстрация и код-клиника	Компьютерный класс ТУ УГМК	Презентация и защита проекта

2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

2.1. Условия реализации программы

Материально-технические условия реализации

- учебный кабинет (компьютерный класс) с местами для 12 учеников;
- компьютеры для учеников (1 ученик — 1 компьютер), компьютер для преподавателя;
- установленный Python 3 и среда разработки (IDLE, Thonny или Visual Studio Code) на каждом рабочем месте;
- установленная библиотека Pygame и работающий модуль turtle (с поддержкой tkinter) на каждом рабочем месте;
- проектор, подключённый к компьютеру преподавателя, выводящий изображение на экран;
- доступ в Интернет для преподавателя; для занятий учеников постоянный доступ в Интернет не требуется;
- доска или флипчарт, мел или маркер.

Учебно-методическое и информационное обеспечение программы:

- поурочные методические указания к каждому занятию, объединяющие объяснение темы, практические задания и решения для преподавателя;
- карточки ученика с краткой памяткой, условиями задач и критериями проекта;
- подготовленные фрагменты кода с ошибками для отладочной лаборатории;
- таблицы типичных затруднений с вопросами для ученика и таблицы быстрой оценки результата;
- примеры готовых проектов предыдущих потоков для демонстрации ученикам;
- памятка по организации рабочего места и безопасной работе за компьютером.

Кадровое обеспечение: Педагогическая деятельность осуществляется лицами, имеющими среднее профессиональное или высшее образование (в том числе по направлениям, соответствующим направлению программы) и отвечающим квалификационным требованиям, указанным в квалификационных справочниках, и (или) профессиональных стандартах.

Преподаватель придерживается принятого в программе порядка помощи ученику: вопрос — указание на участок кода — поиск парных символов и данных — временное упрощение — короткий образец только в последнюю очередь. Готовое решение проекта ученикам не демонстрируется.

2.2. Форма аттестации/контроля и оценочные материалы

Текущий контроль сформированности результатов освоения программы осуществляется с помощью нескольких инструментов на нескольких уровнях:

- на каждом занятии: демонстрация работающей программы с проверкой преподавателем, взаимоконтроль учеников в парах, самоконтроль по карточке ученика;
- на каждом занятии: выходной билет — короткое задание или вопрос по ключевому понятию темы;
- в конце каждого модуля: презентация итогового проекта модуля и его обсуждение с преподавателем и группой.

Для контроля сформированности результатов освоения программы используются следующие механики:

- проверка программы на соответствие заранее объявленным критериям готовности проекта;
- проверка результата на нескольких сценариях, включая обычный, граничный и повторный;
- устный опрос по терминологии и принципам работы конструкций языка;
- проверка умения найти и исправить ошибку в предложенном фрагменте кода;
- презентация и защита индивидуальных и групповых проектов;
- итоговая защита курса — демонстрация собственной игры с объяснением ключевого технического решения.

Уровень освоения программы оценивается по трёхуровневой шкале: «требуется поддержка» — ученик работает по образцу с опорой на карточку; «базовый» — ученик самостоятельно решает типовые задачи занятия и находит ошибку по сообщению интерпретатора; «уверенный» — ученик самостоятельно проектирует решение, тестирует его на границах и помогает другим ученикам.

3. СПИСОК ЛИТЕРАТУРЫ

Список литературы

1. Официальная документация языка Python. — Python Software Foundation, <https://docs.python.org/3/>.
2. Официальная документация библиотеки Pygame. — Pygame Community, <https://www.pygame.org/docs/>.
3. Лутц М. Изучаем Python. — Вильямс, 2020.
4. Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — Питер, 2021.
5. Свейгарт Эл. Автоматизация рутинных задач с помощью Python. — Диалектика, 2021.
6. Бриггс Дж. Python для детей. Самоучитель по программированию. — Манн, Иванов и Фербер, 2018.
7. Макграт М. Программирование на Python для начинающих. — Эксмо, 2019.
8. Доусон М. Програмируем на Python. — Питер, 2019.
9. Л.Л. Босова, А.Ю. Босова. Информатика. Учебники для 6–9 классов. — БИНОМ. Лаборатория Знаний.
10. И.Г. Семакин, Л.А. Залогова, С.В. Русаков. Информатика. Учебник для 7 класса. — БИНОМ. Лаборатория Знаний.
11. К.Ю. Поляков, Е.А. Еремин. Информатика. Углублённый уровень. — БИНОМ. Лаборатория Знаний.
12. Материалы образовательной платформы Python Institute. — <https://pythoninstitute.org>.