

Управление образования администрации Кемеровского городского округа  
Муниципальное бюджетное образовательное учреждение  
дополнительного образования  
«Центр развития творчества детей и юношества Кировского района»

Принята на заседании  
методического совета

от «04» 08 2023г.  
Протокол № 12

Утверждаю:  
Директор МБОУДО «ЦРТДиЮ  
Кировского района»  
/Т.В. Вяткина/  
«04» 08 2023г.



**Дополнительная общеобразовательная общеразвивающая программа**  
**«Программирование на Python»**  
**(стартовый, базовый, продвинутый уровень)**  
Возраст учащихся: 12 – 16 лет  
Срок реализации: 3 года

**Разработчик:**

Ашмянец Людмила Сергеевна,  
педагог дополнительного образования

## Оглавление

<b>Пояснительная записка</b>	<b>3</b>
<b>Цель программы стартового уровня обучения</b>	<b>6</b>
<b>Учебный план программы стартового уровня обучения</b>	<b>7</b>
<b>Содержание учебного плана программы стартового уровня обучения</b>	<b>10</b>
<b>Цель программы базового уровня обучения</b>	<b>13</b>
<b>Учебный план программы базового уровня обучения</b>	<b>14</b>
<b>Содержание учебного плана программы базового уровня обучения</b>	<b>16</b>
<b>Цель программы продвинутого уровня обучения</b>	<b>19</b>
<b>Учебный план программы продвинутого уровня обучения</b>	<b>21</b>
<b>Содержание учебного плана программы продвинутого уровня обучения</b>	<b>26</b>
<b>Результаты освоения учебного курса</b>	<b>32</b>
<b>Комплекс организационно-педагогических условий</b>	<b>34</b>
<b>Условия реализации программы</b>	<b>44</b>
<b>Формы контроля и подведения итогов реализации программы</b>	<b>44</b>
<b>Список литературы</b>	<b>46</b>

## **Пояснительная записка**

В школьном курсе информатики программирование нередко представлено лишь на элементарном уровне, на это выделяется недостаточное количество часов. Следствием этого является формальное восприятие учащимися основ современного программирования и неумение применять полученные знания на практике. При этом через программирование ребенок развивает вычислительное мышление, которое помогает детям развивать навыки решения задач, креативное мышление, умение учиться и навыки совместной работы.

**Программа «Программирование на Python» относится к технической направленности** и основывается на положениях основных законодательных, нормативных и рекомендательных актах Российской Федерации.

Программа основывается на положениях основных законодательных, нормативных и рекомендательных актах Российской Федерации.

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».

2. Указ Президента РФ от 07.05.2012 № 599 «О мерах по реализации государственной политики в области образования и науки».

3. Концепции развития дополнительного образования детей до 2030 года (утверждена распоряжением Правительства РФ от 31.03.2022 № 678-р).

4. Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (утвержден Приказом Минпросвещения России от 27.07.2022 № 629)

5. Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (Приложение к письму Департамента государственной политики в сфере воспитания детей и молодежи Министерства образования и науки РФ от 18.11.2015 г. № 09-3242).

6. Стратегия развития информационного общества в Российской Федерации, утв. Указом Президента РФ от 09.05.2017 № 203 «О Стратегии развития информационного общества в Российской Федерации на 2017-2030 годы».

7. Постановление от 28.09.2020 г. № 28 об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».

8. Приказ Министерства просвещения Российской Федерации от 03.09.2019 г. № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»;

9. Устав МБОУДО «ЦРТДиЮ Кировского района»; «Положением о дистанционном обучении учащихся МБОУДО «ЦРТДиЮ Кировского района».

Стоит отметить, что нормативная база, а именно Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» с изменениями на 06.03.2019 г. согласно статье 16. Реализация образовательных программ с применением электронного обучения и дистанционных образовательных технологий позволяет организациям, осуществляющим образовательную деятельность, вправе применять электронное обучение, дистанционные образовательные технологии при реализации образовательных программ в порядке, установленном Правительством Российской Федерации. (в ред. Федеральных законов от 26.07.2019 N 232-ФЗ, от 26.05.2021 N 144-ФЗ).

**Актуальность программы.** Одним из популярнейших языков программирования был и остается Python, который отличается логичностью, краткостью, прозрачностью синтаксиса, но то же время имеет большое количество библиотек и модулей, которые позволяют решать практически любые задачи от программирования математических вычислений до создания игр и чат-ботов для мобильных приложений. Программа «Программирование на Python» научит подростков основным приемам написания программ на современном языке программирования. Актуальность программы обусловлена интересом учащихся к техническому творчеству в области написания программ и предполагает широкую практическую и самостоятельную деятельность детей, позволяет профессионально ориентировать подростков уже в среднем звене школы.

Обучение программированию позволяет обучающимся применять «вычислительные» подходы в разных контекстах и дисциплинах. Если ребенок научился раскладывать большую задачу на маленькие фрагменты, находить сходства в разных элементах, выявлять и устранять незначимые детали, выстраивать фрагменты в единый алгоритм для получения результата, он сможет решить задачи в любой дисциплине.

**Отличительная особенность программы.**

К концу обучения, учащиеся будут знать:

- методы представления и алгоритмов обработки данных, дискретизации, о
- программной реализации алгоритмов;

- основные конструкции языка программирования Python;
- математические и компьютерные модели, их использование;
- различные виды программного обеспечения и задачи, решаемых с его помощью.

К концу обучения учащиеся должны уметь:

- составлять несложные программы;
- работать с основными видами программных систем и интернет-сервисов;
- определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы.

**Адресатом программы** являются учащиеся 12-16 лет, проявляющим интерес к информационно-коммуникационным технологиям.

**Объем программы** – общее количество учебных часов, запланированных на весь период обучения (3 года), необходимых для освоения программы составляет 468 часов.

**Срок освоения программы.** Программа рассчитана на три периода обучения (стартовый, базовый, продвинутой уровни).

**Формы организации образовательного процесса:** групповые и индивидуальные занятия и другие виды деятельности в программе определяются содержанием программы. Образовательная деятельность учащихся включает в себя следующие виды занятий: лекции, практические и семинарские занятия, мастер – классы, консультаций и другие виды учебных занятий.

**Виды занятий:** традиционные теоретические и практические занятия.

**Методы и формы контроля:** собеседование, анкетирование; контрольный опрос, тестирование; контрольные упражнения; зачёт, зачётная игра; контрольное задание.

Эффективность процесса обеспечивается посредством таких методов, как метод импровизации, метод моделирования, метод проектов, метод погружения в цифровую среду.

**Режим занятий.** Программой предполагается следующий режим работы: стартовый уровень обучения 108 часов – 1 занятие в неделю по 3 часа, базовый уровень обучения 144 часа – 2 занятия в неделю по 2 часа и продвинутой уровень обучения - 216 часов – 2 занятия в неделю по 3 часа. После 30 – 45 минут теоретических занятий организуется перерыв длительностью не менее 10 мин.

Наполняемость групп составляет: до 12 человек.

## **Цель и задачи программы**

**ЦЕЛЬ ПРОГРАММЫ СТАРТОВОГО УРОВНЯ ОБУЧЕНИЯ:** освоение процесса разработки игр с помощью библиотеки PyGame, знакомство учащихся с различными профессиями в игровой индустрии и получение необходимых навыков для создания игр.

Для достижения цели при изучении дисциплины решаются следующие задачи:

### **Предметные:**

- сформировать у обучающихся навык разработки игр с помощью библиотеки PyGame;
- сформировать у обучающихся навык построения логических схем с использованием скриптов;
- познакомить обучающихся с основами языка программирования Python;
- дать обучающимся представление о профессиях гейм-дизайнера, левел-дизайнера и программиста.

### **Метапредметные:**

- сформировать у обучающихся интерес к увлечению программированием и раскрытию своих способностей в сфере IT-технологий;
- научить ребенка работать в режиме командной разработки игрового продукта;
- научить обучающегося создавать свой проект и презентовать его;
- сформировать у обучающихся функциональную грамотность;
- расширить у обучающихся кругозор, развить память, внимание, творческое воображение, абстрактно-логическое и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания.

### **Личностные:**

- формировать у обучающихся информационную культуру;
- развивать у обучающихся учебно-познавательные компетенции в процессе тренировки навыков, решения задач, касающихся написания программ различными методами;
- развивать у обучающихся компетенции самоорганизации в процессе выработки и тренировки наиболее эффективной стратегии выполнения тестовых заданий.

### Учебный план программы стартового уровня обучения

№ п/п	Название раздела, темы	Количество часов			Формы контроля
		Всего	Теория	Практика	
	<b>Инструктаж по технике безопасности в компьютерном классе</b>	<b>0,5</b>	<b>0,5</b>	<b>-</b>	Опрос
<b>1.</b>	<b>Раздел 1. Знакомство с библиотекой PyGame</b>	<b>6</b>	<b>2</b>	<b>4</b>	Тестирование, решение практических задач
1.1	Что такое Pygame?	3	1	2	
1.2	Каркас приложения, FPS	3	1	2	
<b>2.</b>	<b>Раздел 2. Базовые моменты</b>	<b>12</b>	<b>4</b>	<b>8</b>	Тестирование, решение практических задач
2.1	Установка Pygame	3	0,3	2,3	
2.2	Рисование графических примитивов	3	1	2	
2.3	Добавление объектов на экран	3	1	2	
2.4	Анимации и звук Практическая работа 2.5. Анимации и звук	3	1,3	1,3	
<b>3.</b>	<b>Раздел 3. События</b>	<b>9</b>	<b>3</b>	<b>6</b>	Тестирование, Решение практических задач
3.1	Как обрабатывать события от клавиатуры Практическая работа: как обрабатывать события от клавиатуры	3	1	2	
3.2	Как обрабатывать события от мыши Практическая работа: как обрабатывать события от мыши	3	1	2	
3.3	Событие pygame.QUIT.	3	1	2	
<b>4.</b>	<b>Раздел 4. Свойства и методы</b>	<b>12</b>	<b>3,3</b>	<b>8,3</b>	Тестирование, решение практических задач, творческая работа
4.1	Создание поверхностей (Surface) и их анимация. Метод blit	3	1	2	
4.2	Передвижение и прыжок.	3	0,3	2,3	
4.3	Класс Rect.	3	1	2	
4.4	Враги в игре	3	1	2	
<b>5.</b>	<b>Раздел 5. Модули</b>	<b>12</b>	<b>4</b>	<b>8</b>	Тестирование,

5.1	Как рисовать текст различными шрифтами	3	1	2	решение практических задач
5.2	Экран проигрыша	3	1	2	
5.3	Как работать с изображениями. Модули image и transform	3	1	2	
5.4	Добавляем звук в игровой процесс. mixer и music	3	1	2	
<b>6.</b>	<b>Раздел 6. Контроль</b>	<b>9</b>	<b>3</b>	<b>6</b>	Решение практических задач
6.1	Что такое спрайты и как с ними работать	3	1	2	
6.2	Как делать контроль столкновений	3	1	2	
6.3	Стрельба по врагам	3	1	2	
<b>7.</b>	<b>Раздел 7. Логика</b>	<b>21</b>	<b>-</b>	<b>21</b>	Тестирование, Решение практических задач
7.1	Улучшение логики столкновений	3	-	3	
7.2	Простой счётчик очков	3	-	3	
7.3	Улучшение счётчика очков	3	-	3	
7.4	Музыка, звуковые эффекты	3	-	3	
7.5	Счётчик жизней	3	-	3	
7.6	Пополнение жизней	3	-	3	
7.7	Кнопки	3	-	3	
<b>8.</b>	<b>Раздел 7. Контрольные работы</b>	<b>18</b>	<b>3</b>	<b>15</b>	Тестирование, Решение практических задач
8.1	Циклы событий	3	0,3	2,3	
8.2	Размещение предметов на экране	3	0,3	2,3	
8.3	Воспроизведение звуковых эффектов и музыки	3	0,3	2,3	
8.4	Обработка пользовательского ввода	3	0,3	2,3	
8.5	Скорость игры	3	0,3	2,3	
8.6	Добавление изображений	3	0,3	2,3	
<b>9.</b>	<b>Раздел 8. Заключительная часть</b>	<b>9</b>	<b>2</b>	<b>7</b>	Решение практических задач, презентация проекта, рефлексия
9.1	Таблица рекордов	3	1	2	
9.2	Создание APK файла	3	1	2	
9.3	Зачет по курсу «Обучение библиотеки PyGame». Презентация игр.	3	-	3	



	<b>ИТОГО</b>	<b>108</b>	<b>24,3</b>	<b>83,3</b>	
--	--------------	------------	-------------	-------------	--

**Содержание учебного плана программы стартового уровня обучения**

## **Раздел 1. Знакомство с библиотекой PyGame.**

### **Тема 1.1 Что такое Pygame?**

*Теория:* Знакомство с библиотекой PyGame.

*Практика:* Создание своего первого игрового окна.

### **Тема 1.2 Каркас приложения, FPS.**

*Теория:* Понятие игрового движка.

*Практика:* Создание первых объектов.

## **Раздел 2. Базовые моменты.**

### **Тема 2.1 Установка Pygame.**

*Практика:* Правильная загрузка библиотеки в среду программирования.

### **Тема 2.2 Рисование графических примитивов.**

*Теория:* Объект блок и его свойства. Графические примитивы.

*Практика:* Добавление объектов на экран. Прохождение тестирования.

### **Тема 2.3 Добавление объектов на экран.**

*Теория:* Свойства персонажей.

*Практика:* Создание собственного персонажа.

### **Тема 2.4 Анимации и звук.**

*Теория:* Загрузка и настройка мультимедиа. Создание анимации.

*Практика:* Анимирование собственного персонажа. Добавление звука.

## **Раздел 3. События.**

### **Тема 3.1 Как обрабатывать события от клавиатуры.**

*Теория:* Обработка события от клавиатуры. Свойства событий.

*Практика:* Тестирование, опрос.

### **Тема 3.2 Как обрабатывать события от мыши.**

*Теория:* Обработка события от мыши.

*Практика:* Тестирование, опрос.

### **Тема 3.3 Событие pygame. QUIT.**

*Теория:* Событие pygame.QUIT.

*Практика:* Тестирование, опрос.

## **Раздел 4. Свойства и методы.**

### **Тема 4.1 Создание поверхностей (Surface) и их анимация. Метод blit.**

*Теория:* Принципы моделирования объектов.

*Практика:* Создание поверхностей (Surface) и их анимация.

### **Тема 4.2 Передвижение и прыжок.**

*Теория:* Методы.

*Практика:* Работа с анимацией.

### **Тема 4.3 Класс Rect.**

*Теория:* Классы. Свойство классов. Класс Rect.

*Практика:* Работа с анимацией. Тестирование.

#### **Тема 4.4 Враги в игре.**

*Теория:* Классы. Методы.

*Практика:* Работа с анимацией. Создание персонажей.

### **Раздел 5. Модули.**

#### **Тема 5.1 Как рисовать текст различными шрифтами.**

*Теория:* Текст в игре. Стили.

*Практика:* Добавление текста.

#### **Тема 5.2 Экран проигрыша.**

*Теория:* Изучение окон «GameOver».

*Практика:* Создание экрана проигрыша.

#### **Тема 5.3 Как работать с изображениями.**

*Теория:* Модули image и transform.

*Практика:* Работа с изображениями. Тестирование и опрос.

#### **Тема 5.4 Добавляем звук в игровой процесс.**

*Теория:* Модули mixer и music.

*Практика:* Добавление звука в игровой процесс.

### **Раздел 6. Контроль.**

#### **Тема 6.1 Что такое спрайт и как с ними работать.**

*Теория:* Знакомство со спрайтами. Характеристики и свойства.

*Практика:* Работа со спрайтами.

#### **Тема 6.2 Как делать контроль столкновений.**

*Теория:* Методы collidepoint, colliderect.

*Практика:* Работа со спрайтами «Враги».

#### **Тема 6.3 Стрельба по врагам.**

*Теория:* Изучение новых функций.

*Практика:* Работа со спрайтами «Враги».

### **Раздел 7. Логика.**

#### **Тема 7.1 Улучшение логики столкновений.**

*Практика:* Добавление новых возможностей.

#### **Тема 7.2 Простой счетчик очков.**

*Практика:* Создание простого счетчика очков.

#### **Тема 7.3 Улучшение счетчика очков.**

*Практика:* Работа с счетчиком очков.

#### **Тема 7.4 Музыка, звуковые эффекты.**

*Практика:* Добавление звуковых эффектов в игровой процесс.

#### **Тема 7.5 Счетчик жизней.**

*Практика:* Создание счетчика жизней.

#### **Тема 7.6 Пополнение жизней.**

*Теория:* Работа с счетчиком жизней.

*Практика:* Работа со спрайтами «Враги».

**Тема 7.7 Кнопки.**

*Практика:* Улучшение работы других объектов.

**Раздел 8. Контрольные работы.**

**Тема 8.1 Циклы событий.**

*Практика:* Тестирование. Создание игры с учетом темы.

**Тема 8.1 Размещение предметов на экране.**

*Практика:* Тестирование. Размещение предметов в созданной игре для решения контрольных работ.

**Тема 8.2 Воспроизведение звуковых эффектов и музыки.**

*Практика:* Тестирование. Добавление звука в созданную игру для решения контрольных работ.

**Тема 8.3 Обработка пользовательского ввода.**

*Практика:* Тестирование. Контрольная работа «Обработка ввода с клавиатуры» и «Обработка ввода с помощью мыши»

**Тема 8.4 Скорость игры.**

*Практика:* Тестирование. Работа над игрой.

**Тема 8.4 Добавление изображений.**

*Практика:* Тестирование. Работа с изображениями в созданной игре для решения контрольных работ.

**Раздел 9. Заключительная часть.**

**Тема 9.1 Таблица рекордов.**

*Теория:* Последний этап в игре «Таблица рекордов».

*Практика:* Создание окна.

**Тема 9.2 Создание APK файла.**

*Теория:* Знакомство с APK файлом. Установка «buildozer». Настройки файла.

*Практика:* Создание APK. Загрузка игры Google Play Market.

**Тема 9.3 Зачет по курсу.**

*Практика:* Разделение на группы и получение вводных для проекта. Завершение индивидуального проекта. Подготовка презентации проекта. Выступление. Рефлексия.

## **Цель и задачи программы**

**ЦЕЛЬ ПРОГРАММЫ БАЗОВОГО УРОВНЯ ОБУЧЕНИЯ:** способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Для достижения цели при изучении дисциплины решаются следующие задачи:

### **Предметные:**

- познакомить обучающихся с основными видами программных систем и интернет-сервисов;
- познакомить обучающихся с особенностями синтаксиса языка программирования Python;
- познакомить обучающихся с принципами объектно-ориентированного программирования;
- сформировать у обучающихся навык правильного оформления кода;
- обучить обучающихся приемам составления простых программ на языке Python;
- сформировать и развить у обучающихся навыки алгоритмического и логического мышления.

### **Метапредметные:**

- сформировать и развить логическое, алгоритмическое, критическое, креативное и пространственное мышление обучающихся;
- сформировать у обучающихся навык публичного выступления и презентации;
- сформировать у обучающихся навык планирования деятельности;
- сформировать и расширить у обучающихся словарный запас на английском языке.

### **Личностные:**

- воспитать у обучающихся потребность в сотрудничестве, взаимодействии со сверстниками, умение подчинять свои интересы определенным правилам, развить эмоциональный интеллект;
- сформировать у обучающихся умение давать качественную обратную связь и реагировать на нее;
- формировать у обучающихся информационную культуру.

### Учебный план программы базового уровня обучения

№ п/п	Название раздела, темы	Количество часов			Формы контроля
		Всего	Теория	Практика	
	<b>Инструктаж по технике безопасности в компьютерном классе</b>	<b>0,5</b>	<b>0,5</b>	<b>-</b>	Опрос
<b>1.</b>	<b>Раздел 1. Введение в Python</b>	<b>10</b>	<b>4</b>	<b>6</b>	Тестирование, решение практических задач
1.1	Занятие 1. Язык Python. Среда программирования. Особенности ввода-вывода	2	1	1	
1.2	Занятие 2. Типы данных, операции. Оператор присваивания.	2	1	1	
1.3	Занятие 3. Числа. Стандартные операции.	6	2	4	
<b>2.</b>	<b>Раздел 2. Алгоритмические конструкции</b>	<b>20</b>	<b>10</b>	<b>10</b>	Тестирование, решение практических задач
2.1	Занятие 4. Условный оператор	4	2	2	
2.2	Занятие 5. Цикл while. Теория чисел	8	4	4	
2.3	Занятие 6. Цикл for	8	4	4	
<b>3.</b>	<b>Раздел 3. Строки</b>	<b>18</b>	<b>12</b>	<b>8</b>	Тестирование, Решение практических задач
3.1	Занятие 7. Литералы строк	4	3	2	
3.2	Занятие 8. Срезы строк	4	3	2	
3.3	Занятие 9. Методы строк	10	6	4	
<b>4.</b>	<b>Раздел 4. Функции</b>	<b>26</b>	<b>14</b>	<b>12</b>	Тестирование, решение практических задач, творческая работа
4.1	Занятие 10. Парадигма и преимущества структурного программирования	4	2	2	
4.2	Занятие 11. Граф и стек вызовов функции. Области видимости переменных	4	2	2	
4.3	Занятие 12. Прямая рекурсия	10	6	4	
4.4	Занятие 13. Косвенная рекурсия	8	4	4	
<b>5.</b>	<b>Раздел 5. Списки и кортежи</b>	<b>22</b>	<b>9</b>	<b>13</b>	Тестирование,
5.1	Занятие 14. Списки и кортежи в Python. Сходства и различия	6	2	4	

5.2	Занятие 15. Операции со списками	10	4	6	решение практических задач
5.3	Занятие 16. Срезы списков	2	1	1	
5.4	Занятие 17. Матрицы. Операции над матрицами.	4	2	2	
<b>6.</b>	<b>Раздел 6. Словари и множества</b>	<b>8</b>	<b>4</b>	<b>4</b>	Решение практических задач
6.1	Занятие 18. Словари	4	2	2	
6.2	Занятие 19. Множества	4	2	2	
<b>7.</b>	<b>Раздел 7. Объектно-ориентированное программирование</b>	<b>32</b>	<b>1</b>	<b>31</b>	Тестирование, Решение практических задач
7.1	Занятие 20. Классы в Python	2	1	1	
7.2	Занятие 21. Разработка собственного класса	10	-	10	
7.3	Занятие 22. Разработка и программирование собственного проекта	20	-	20	
<b>8.</b>	<b>Раздел 8. Заключительное занятие</b>	<b>8</b>	<b>-</b>	<b>8</b>	Тестирование, Решение практических задач
8.1	Подведение итогов. Индивидуальный проект	8	-	8	
	<b>ИТОГО</b>	<b>144</b>	<b>54</b>	<b>90</b>	

## Содержание учебного плана программы базового уровня обучения

### Раздел 1. Введение в Python.

#### Тема 1.1 Язык Python. Среда программирования. Особенности ввода-вывода.

*Теория:* Язык программирования Python. Достоинства и недостатки.

*Практика:* Установка языка программирования Python и среды программирования.

#### Тема 1.2 Типы данных, операции. Оператор присваивания.

*Теория:* Ввод и вывод числовой информации.

*Практика:* Тренировочное задание на ввод и вывод числовой информации.

#### Тема 1.3 Числа. Стандартные операции.

*Теория:* Стандартные операции с целыми и действительными числами.

*Практика:* Решение простых задач в интерактивном режиме.

### Раздел 2. Алгоритмические инструкции.

#### Тема 2.1 Условный оператор.

*Теория:* Условная и циклическая инструкция. Каскадность и вложенность алгоритмических инструкций.

*Практика:* Решение задач на анализ чисел потока и целочисленной арифметики.

#### Тема 2.2 Цикл while. Теория чисел.

*Теория:* Теория чисел. Фильтрация потока чисел.

*Практика:* Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности.

#### Тема 2.3 Цикл for.

*Теория:* Синтаксис for: break и continue.

*Практика:* Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности.

### Раздел 3. Строки.

#### Тема 3.1 Литералы строк.

*Теория:* Понятие «литералы строк».

*Практика:* Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.

#### Тема 3.2 Срезы строк.

*Теория:* Форматирование строки.

*Практика:* Преобразование строки. Решение упражнений.

#### Тема 3.3 Методы строк.

*Теория:* Методы работы со строкой.

*Практика:* Применение методов строки. Решение задач.



## **Раздел 4. Функции.**

### **Тема 4.1 Парадигма и преимущества структурного программирования.**

*Теория:* Обзор парадигм программирования. Особенности применения языков программирования.

*Практика:* Выполнение тренировочных упражнений.

### **Тема 4.2 Граф и стек вызовов функции. Области видимости переменных.**

*Теория:* Применение стек и граф вызовов функций.

*Практика:* Выполнение тренировочных упражнений.

### **Тема 4.3 Прямая рекурсия.**

*Теория:* Понятие прямой рекурсии и ее применение.

*Практика:* Выполнение тренировочных упражнений.

### **Тема 4.3 Косвенная рекурсия.**

*Теория:* Понятие косвенной рекурсии и ее применение.

*Практика:* Выполнение тренировочных упражнений.

## **Раздел 5. Списки и кортежи.**

### **Тема 5.1 Списки и кортежи в Python. Сходства и различия.**

*Теория:* Представление списка и кортежа в памяти компьютера, сходства и различия.

*Практика:* Решение задач на ввод-вывод элементов одномерного и многомерного списка.

### **Тема 5.2 Операции со списками.**

*Теория:* Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле).

*Практика:* Решение задач на ввод-вывод элементов одномерного и многомерного списка и кортежа, преобразование, поиск, замену, подсчет.

### **Тема 5.3 Срезы списков.**

*Теория:* Методы работы со списком и кортежем. Методы сортировки списка.

*Практика:* Решение задач повышенной трудности.

### **Тема 5.4 Матрицы. Операции над матрицами.**

*Теория:* Вычисление сложности алгоритма. Многомерные списки.

*Практика:* Решение задач повышенной трудности.

## **Раздел 6. Словари и множества.**

### **Тема 6.1 Словари.**

*Теория:* Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем. Словари со смешанными значениями. Кодирование и декодирование текста.

*Практика:* Решение задач на заполнение, преобразование, поиск, замену,

подсчет, вывод элементов словаря.

### **Тема 6.2 Множества.**

*Теория:* Понятие множества. Создание множеств. Множество, преимущества и недостатки, методы работы с множеством.

*Практика:* Решение задач повышенной трудности.

## **Раздел 7. Объектно-ориентированное программирование.**

### **Тема 7.1 Классы в Python.**

*Теория:* Понятия «класс», «метод INIT», «экземпляр», «наследование», «полиморфизм», «исключения», «виджет», «интерфейс», «событие».

*Практика:* Решение задач повышенной трудности.

### **Тема 7.2 Разработка собственного класса.**

*Теория:* Принципы разработки собственного класса. Обработка и генерация исключений. Виджет, методы виджета. Графическая библиотека tkinter, класс Tk.

*Практика:* Решение задач повышенной трудности.

### **Тема 7.3 Разработка и программирование собственного проекта.**

*Практика:* Выбор вида и темы проекта. Составление технического задания. Программирование.

## **Раздел 8. Подведение итогов.**

### **Тема 8.1 Подведение итогов. Индивидуальный проект.**

*Практика:* Защита индивидуального проекта.

## **Цель и задачи программы**

**ЦЕЛЬ ПРОГРАММЫ ПРОДВИНУТОГО УРОВНЯ ОБУЧЕНИЯ:** формирование у обучающихся информационных компетенций, овладение ими базовыми понятиями теории алгоритмов, умеющего разрабатывать эффективные алгоритмы и реализовывающего их в виде программы, написанной на языке программирования Python.

Для достижения цели при изучении дисциплины решаются следующие задачи:

### **Предметные:**

- сформировать у обучающихся навыки алгоритмического мышления и понимание необходимости формального описания алгоритмов;
- обучить обучающихся стандартными приемами написания программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ на языке Python;
- сформировать у обучающихся навык разработки программ на Python, включая тестирование и отладку программ;
- обучить обучающихся элементарными навыками формализации прикладной задачи.

### **Метапредметные:**

- обучить обучающихся самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- обучить обучающихся самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата, понимая, что в программировании длинная программа – не значит лучшая;
- сформировать у обучающихся умение критически оценивать правильность решения учебно-исследовательской задачи;
- сформировать у обучающихся умение корректировать свои действия, вносить изменения в программу и отлаживать ее в соответствии с изменяющимися условиями;
- обучить обучающихся основами самоконтроля, принятия решения.

### **Личностные:**

- формировать у обучающихся мировоззрения, соответствующего современному уровню развития науки и техники;
- развивать у обучающихся готовность и способность к образованию, в том числе самообразованию;

– формировать у обучающихся навыки сотрудничества со сверстниками, детьми младшего возраста, взрослыми в образовательной, учебно-исследовательской, проектной и других видах деятельности.

### Учебный план программы продвинутого уровня обучения

№ п/п	Название раздела, темы	Количество часов			Формы контроля
		Всего	Теория	Практика	
	<b>Инструктаж по технике безопасности в компьютерном классе</b>	<b>0,5</b>	<b>0,5</b>	<b>-</b>	Опрос
<b>1.</b>	<b>Раздел 1. Знакомство с языком Python</b>	<b>18</b>	<b>5</b>	<b>12</b>	Тестирование, решение практических задач
1.1	Знакомство с внутренней структурой компьютера с точки зрения программ. Понятие алгоритм и программа. Виды языков программирования. Общие сведения о языке Python.	3	1	2	
1.2	Ввод и вывод данных на языке Python. Параметры вывода. Ввод и отладка программ в среде.	3	1	2	
1.3	Переменные в Python, основные понятия.	3	1	2	
1.4	Целочисленная арифметика. Алгоритмы работы с числами.	3	1	2	
1.5	Текущий контроль. Повторение пройденного материала. Создание программы по индивидуальному заданию	6	-	6	
<b>2.</b>	<b>Раздел 2. Понятие переменных и условных операторов</b>	<b>30</b>	<b>6</b>	<b>24</b>	Тестирование, решение практических задач
2.1	Понятие условных операторов и методы их применения на практике.	3	1	2	

	Знакомство с понятием True и False.				
2.2	Логические операции not/and/or	3	1	2	
2.3	Текущий контроль. Решение задач и создание программы для закрепления полученных знаний и навыков	6	-	6	
2.4	Знакомство с понятием «типы числовых данных»	3	1	2	
2.5	Модуль math: определение, подключение и варианты использования.	3	1	2	
2.6	Строковый тип данных. Работа с строками len(), str()	3	1	2	
2.7	Способы использования строковых данных. Оператор in	3	1	2	
2.8	Текущий контроль. Создание программы для закрепления полученных знаний и навыков	6	-	6	
<b>3.</b>	<b>Раздел 3. Циклы for и while и их применение</b>	<b>27</b>	<b>7</b>	<b>20</b>	Тестирование решение практических задач
3.1	Общие понятия о циклах и о функциях. Знакомство с циклом for в Python	3	1	2	
3.2	Функция range() и параметры.	3	1	2	
3.3	Наиболее частые сценарии использования цикла	3	1	2	
3.4	Знакомство с циклом while в Python	3	1	2	
3.5	Практическое использование цикла while для закрепления знаний.	3	-	3	
3.6	Использование операторов continue, break и else в циклах	3	1	2	
3.7	Анализ кода написанного	3	-	3	

	обучающимися и выявление типичных ошибок и способов их обнаружения.				
3.8	Оптимизация и читабельность кода. Разбор ситуаций.	3	1	2	
3.9	Вложенные циклы. Где применяются и для чего.	3	1	2	
<b>4.</b>	<b>Раздел 4. Строковый тип данных</b>	<b>18</b>	<b>4</b>	<b>14</b>	Тестирование, решение практических задач
4.1	Индексация.	3	1	2	
4.2	Срезы.	3	1	2	
4.3	Методы Строк. Обзор.	3	1	2	
4.4	Строки в памяти компьютера	3	1	2	
4.5	Текущий контроль. Написание программы по творческим заданиям	6	-	6	
<b>5.</b>	<b>Раздел 5. Сложные типы данных</b>	<b>33</b>	<b>9</b>	<b>24</b>	Тестирование, Решение практических задач
5.1	Промежуточный мониторинг по 1-4 разделам	3	1	2	
5.2	Введение в списки. Основы работы со списками.	3	1	2	
5.3	Методы списков.	3	1	2	
5.4	Вывод элементов списка	3	1	2	
5.5	Методы строк split, join	3	1	2	
5.6	Списочные выражения	3	1	2	
5.7	Сортировка списков	3	1	2	
5.8	Кортежи	3	1	2	
5.9	Введение в словари	3	1	2	
5.10	Написание программы по творческим заданиям	6		6	
<b>6.</b>	<b>Раздел 6. Функции и модули</b>	<b>15</b>	<b>5</b>	<b>10</b>	Тестирование,
6.1	Функции без параметров и с параметрами	3	1	2	

6.2	Локальные и глобальные переменные. Область видимости.	3	1	2	Решение практических задач
6.3	Функции с возвратом значений	3	1	2	
6.4	Модули. Модуль random	3	1	2	
6.5	Текущий контроль. Демонстрация программы учащегося, разбор ошибок и возможности улучшения	3	1	2	
<b>7.</b>	<b>Раздел 7. Работа с файлами</b>	<b>12</b>	<b>2</b>	<b>10</b>	
7.1	Типы файлов с точки зрения программиста. Запись информации в файл, чтение из файла	3	1	2	
7.2	Работа с текстовыми файлами: запись, чтение изменение. Решение задач.	3	1	2	
7.3	Создание программы для закрепления полученных знаний и навыков	6		6	
<b>8.</b>	<b>Раздел 8. Введение в Объектно – ориентированное программирование</b>	<b>30</b>	<b>6</b>	<b>24</b>	Тестирование, Решение практических задач
8.1	Концепция ООП. Объекты и классы.	3	1	2	
8.2	Принципы ООП: Рекурсия	3	1	2	
8.3	Программирование объектной модели. Решение задач.	3	1	2	
8.4	Принципы ООП: инкапсуляция.	3	1	2	
8.5	Принципы ООП: наследование и полиморфизм	3	1	2	



8.6	Объектно - ориентированное программирование. Решение задач	3	1	2	
8.7	Текущий контроль. Создание программы для закрепления полученных знаний и навыков	12	-	12	
<b>9.</b>	<b>Раздел 9. Итоговый проект. Проектная работа в группах</b>	<b>32</b>	<b>1</b>	<b>31</b>	Решение практических задач, презентация проекта, рефлексия
9.1	Разделение на группы и получение вводных для проекта	3	1	2	
9.2	Работа в группах	21	-	21	
9.3	Итоговый мониторинг	3	-	2	
9.4	Презентация результатов работы	6	-	6	
	<b>ИТОГО</b>	<b>216</b>	<b>47</b>	<b>169</b>	

## Содержание учебного плана программы продвинутого уровня обучения

### Раздел 1. Знакомство с языком Python.

**Тема 1.1 Знакомство с внутренней структурой компьютера с точки зрения программ. Понятие алгоритм и программа. Виды языков программирования. Общие сведения о языке Python.**

*Теория:* Понятие алгоритм и программа. Общие сведения о языке Python.

*Практика:* Практическая работа, установка на компьютер интерпретатор Python. Знакомство с IDE.

**Тема 1.2 Ввод и вывод данных на языке Python. Параметры вывода. Ввод и отладка программ в среде.**

*Теория:* Ввод и вывод данных на языке Python. Параметры вывода. Ввод и отладка программ в среде.

*Практика:* Решение задач.

**Тема 1.3 Переменные в Python, основные понятия.**

*Теория:* Переменные, основные понятия и отличия от других языков.

*Практика:* Решение задач.

**Тема 1.4 Целочисленная арифметика. Алгоритмы работы с числами.**

*Теория:* Целочисленная арифметика, операторы. Алгоритмы работы с числами.

*Практика:* Решение задач.

**Тема 1.5 Текущий контроль. Повторение пройденного материала. Создание программы по индивидуальному заданию.**

*Теория:* Первая программа.

*Практика:* Создание программы по индивидуальному заданию. Оценка усвоения материала. Тест.

### Раздел 2. Понятие переменных и условных операторов.

**Тема 2.1 Понятие условных операторов и методы их применения на практике. Знакомство с понятием True и False.**

*Теория:* Понятие условных операторов и методы их применения на практике. Знакомство с понятием True и False.

*Практика:* Решение задач с условными операторами.

**Тема 2.2 Логические операции not/and/or.**

*Теория:* Логические операции not/and/or.

*Практика:* Решение задач и создание программы для закрепления полученных знаний и навыков.

**Тема 2.3 Текущий контроль. Решение задач и создание программы для закрепления полученных знаний и навыков.**

*Практика:* Создание программы по индивидуальному заданию. Оценка

усвоения материала. Тест.

#### **Тема 2.4 Знакомство с понятием «типы числовых данных».**

*Теория:* Знакомство с понятием «типы числовых данных».

*Практика:* Решение задач.

#### **Тема 2.5 Модуль math: определение, подключение и варианты использования.**

*Теория:* Модуль math: определение, подключение и варианты использования.

*Практика:* Решение задач.

#### **Тема 2.6 Строковый тип данных. Работа с строками len(), str().**

*Теория:* Строковый тип данных. Работа с строками len(), str().

*Практика:* Решение задач.

#### **Тема 2.7 Способы использования строковых данных. Оператор in.**

*Теория:* Способы использования строковых данных. Оператор in.

*Практика:* Решение задач.

#### **Тема 2.8 Текущий контроль. Создание программы для закрепления полученных знаний и навыков.**

*Практика:* Создание программы по индивидуальному заданию с целью закрепления знаний по разделу. Тестирование.

### **Раздел 3. Циклы for и while и их применение.**

#### **Тема 3.1 Общие понятия о циклах и о функциях. Знакомство с циклом for в Python.**

*Теория:* Общие понятия о циклах и о функциях. Знакомство с циклом for в Python.

*Практика:* Решение задач.

#### **Тема 3.2 Функция range() и параметры.**

*Теория:* Функция range() и параметры.

*Практика:* Решение задач.

#### **Тема 3.3 Наиболее частые сценарии использования цикла.**

*Теория:* Наиболее частые сценарии использования цикла.

*Практика:* Анализ программ написанных другими, выявление типичных ошибок и способов их обнаружения.

#### **Тема 3.4 Знакомство с циклом while в Python.**

*Теория:* Знакомство с циклом while в Python.

*Практика:* Практическое использование цикла while в решении конкретных задач.

#### **Тема 3.5 Практическое использование цикла while для закрепления знаний.**

*Практика:* Практическое использование цикла while для закрепления

знаний.

### **Тема 3.6 Использование операторов continue, break и else в циклах.**

*Теория:* Использование операторов continue, break и else в циклах.

*Практика:* Решение задач с условными операторами.

### **Тема 3.7 Анализ кода написанного обучающимися и выявление типичных ошибок и способов их обнаружения.**

*Практика:* Анализ кода написанного обучающимися и выявление типичных ошибок и способов их обнаружения.

### **Тема 3.8 Оптимизация и читабельность кода. Разбор ситуаций.**

*Теория:* Правильный рефакторинг. Оптимизация и читабельность кода.

*Практика:* Разбор и устранение ошибок.

### **Тема 3.9 Вложенные циклы. Где применяются и для чего.**

*Теория:* Вложенные циклы. Использование инструкции break во вложенных циклах.

*Практика:* Решение задач с циклами.

## **Раздел 4. Строковый тип данных.**

### **Тема 4.1 Индексация.**

*Теория:* Индексация, работа со строками как с массивами.

*Практика:* Написание программ по заданиям.

### **Тема 4.2 Срезы.**

*Теория:* Срезы, выделение определенных данных из строк.

*Практика:* Написание программ по заданиям.

### **Тема 4.3 Методы Строк. Обзор.**

*Теория:* Методы строк, Обзор.

*Практика:* Написание программ по заданиям.

### **Тема 4.4 Строки в памяти компьютера.**

*Теория:* Строки в памяти компьютера, таблица символов Unicode.

*Практика:* Тестирование.

### **Тема 4.5 Текущий контроль. Написание программы по творческим заданиям.**

*Практика:* Написание программы по творческим заданиям, для закрепления материала.

## **Раздел 5. Сложные типы данных.**

### **Тема 5.1 Промежуточный мониторинг по 1 -4 разделам.**

*Практика:* Контрольные работы по 1 – 4 разделам.

### **Тема 5.2 Введение в списки. Основы работы со списками.**

*Теория:* Введение в списки. Основы работы со списками.

*Практика:* Решение задач.

### **Тема 5.3 Методы списков.**

*Теория:* Методы списков. Обзор.

*Практика:* Решение задач.

#### **Тема 5.4 Вывод элементов списка.**

*Теория:* Вывод элементов списка.

*Практика:* Решение задач.

#### **Тема 5.5 Методы строк split, join.**

*Теория:* Методы строк split, join и их применение.

*Практика:* Решение задач.

#### **Тема 5.6 Списочные выражения.**

*Теория:* Списочные выражения и их применение.

*Практика:* Решение задач.

#### **Тема 5.7 Сортировка списков.**

*Теория:* Сортировка списков.

*Практика:* Решение задач.

#### **Тема 5.8 Кортежи.**

*Теория:* Кортежи. Свойства и их применение.

*Практика:* Решение задач.

#### **Тема 5.9 Введение в словари.**

*Теория:* Введение в словари.

*Практика:* Решение задач.

#### **Тема 5.10 Написание программы по творческим заданиям.**

*Практика:* Написание программы по индивидуальному заданию с целью применения полученных по теме знаний. Тестирование.

### **Раздел 6. Функции и модули.**

#### **Тема 6.1 Функции без параметров и с параметрами.**

*Теория:* Функции с параметрами и без параметров.

*Практика:* Решение задач.

#### **Тема 6.2 Локальные и глобальные переменные. Область видимости.**

*Теория:* Локальные и глобальные переменные. Область видимости.

*Практика:* Решение задач.

#### **Тема 6.3 Функции с возвратом значений.**

*Теория:* Возврат значений из функции. Оператор return.

*Практика:* Решение задач.

#### **Тема 6.4 Модули. Модуль random.**

*Теория:* Модули. Модуль random.

*Практика:* Решение задач.

#### **Тема 6.5 Текущий контроль. Демонстрация программы учащегося, разбор ошибок и возможности улучшения.**

*Практика:* Написание программы по индивидуальному заданию с целью

применения полученных по теме знаний. Тестирование.

## **Раздел 7. Работа с файлами.**

### **Тема 7.1 Типы файлов с точки зрения программиста. Запись информации в файл, чтение из файла.**

*Теория:* Типы файлов. Программирование файлового ввода/вывода.

*Практика:* Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

### **Тема 7.2 Работа с текстовыми файлами: запись, чтение изменение.**

*Теория:* Текстовые и двоичные файлы. Этапы обработки файла. Программирование обработки файлов. Типовые задачи по изменению файлов.

*Практика:* Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

### **Тема 7.3 Создание программы для закрепления полученных знаний и навыков.**

*Практика:* Написание программы по индивидуальному заданию с целью применения полученных по теме знаний. Тестирование.

## **Раздел 8. Введение в Объектно – ориентированное программирование.**

### **Тема 8.1 Концепция ООП. Объекты и классы.**

*Теория:* Концепция ООП. Объекты и классы.

*Практика:* Самостоятельная работа. Рефлексия.

### **Тема 8.2 Принципы ООП: Рекурсия.**

*Теория:* Последовательное конструирование алгоритма. Объектная модель решения задачи. Объектно-ориентированный анализ (ООА). Этапы ООА.

*Практика:* Решение задач.

### **Тема 8.3 Программирование объектной модели.**

*Теория:* Программирование объектной модели.

*Практика:* Решение задач.

### **Тема 8.4 Принципы ООП: инкапсуляция.**

*Теория:* Абстракция. Поля и методы класса. Программирование объектов и классов.

*Практика:* Решение задач.

### **Тема 8.5 Принципы ООП: наследование и полиморфизм.**

*Теория:* Принципы ООП: наследование и полиморфизм.

*Практика:* Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

### **Тема 8.6 Объектно - ориентированное программирование.**

*Теория:* Концепции, принципы и примеры реализации ООП.

*Практика:* Решение задач.

**Тема 8.7 Текущий контроль. Создание программы для закрепления полученных знаний и навыков.**

*Практика:* Представление проектов с использованием метода ООП. Тестирование.

**Раздел 9. Итоговый проект. Проектная работа в группах.**

**Тема 9.1 Разделение на группы и получение вводных для проекта.**

*Теория:* Разделение на группы и получение вводных для проекта.

*Практика:* Обсуждение тем.

**Тема 9.2 Работа в группах.**

*Практика:* Работа в группах по проектам, консультация и практическая помощь.

**Тема 9.3 Итоговый мониторинг.**

*Практика:* Самостоятельная работа. Рефлексия.

**Тема 9.4 Презентация результатов работы.**

*Практика:* Презентация результатов работы.

## **Результаты освоения учебного курса**

### **Личностные результаты**

Учащиеся знают, как применять правила поведения в компьютерном классе и этические нормы работы с информацией коллективного пользования и личной информацией обучающегося. Учащиеся смогли выделить нравственный аспект поведения при работе с любой информацией и при использовании компьютерной техники коллективного пользования. Учащиеся научились самостоятельно соблюдать правила работы с файлами, правила поведения в компьютерном классе, цель которых – сохранение школьного имущества и здоровья одноклассников

### **Метапредметные результаты образовательной деятельности**

- сформировано умение ставить учебные цели;
- сформировано умение использовать внешний план для решения поставленной задачи;
- сформировано умение планировать свои действия в соответствии с поставленной задачей и условиями ее реализации;
- сформировано – умение работать с различными источниками информации, извлекать и анализировать необходимую информацию из открытых источников;
- сформировано умение выполнять проекты в соответствии с техническим заданием;
- ознакомились с правилами индивидуального и коллективного безопасного поведения при работе с компьютерной техникой и оборудованием.

### **Предметные результаты образовательной деятельности по программе стартового уровня**

- ознакомились с основными предметными понятиями программирования, компьютерный наук и их свойств;
- ознакомились с базовым синтаксисом и инструментарием языка программирования Python;
- сформировано умение применять объектно-ориентированную парадигму в программировании;
- сформирован навык разработки игр и программ на основе изучения языка программирование Python.

### **Предметные результаты образовательной деятельности по программе базового уровня**

- сформирован навык кодировать и декодировать сообщения по заданным правилам;



- сформировано умение использовать переменные различных типов при написании программ на Python;
- сформировано умение использовать оператор присваивания при написании программ на Python;
- сформировано умение поиска ошибок в программном коде на Python;
- сформирован навык анализа блок-схем и программ на Python.

**Предметные результаты образовательной деятельности по программе продвинутого уровня**

- сформировано умение определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных.
- сформировано представление об основных предметных понятиях и их свойствах;
- развиты логические способности и алгоритмическое мышление, умение составлять и записывать алгоритм для конкретного исполнителя;
- сформирован навык разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- сформировано умение использовать основные управляющие конструкции объектно – ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- сформировано умение соблюдать нормы информационной этики и права.

**Раздел 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-  
ПЕДАГОГИЧЕСКИХ УСЛОВИЙ  
КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК  
Стартовый уровень**

№	месяц	неделя	Форма занятия	Кол-во часов	Разделы и темы	Форма контроля
1	Сентябрь	1 неделя	Комбинированное занятие	3	Что такое Pygame?	коллективная рефлексия, тестирование
2	Сентябрь	2 неделя	Комбинированное занятие	3	Каркас приложения, FPS.	опрос
3	Сентябрь	3 неделя	Обучающее практическое занятие	3	Установка Pygame.	практическое задание
4	Сентябрь	4 неделя	Практическое занятие	3	Рисование графических примитивов.	практическое задание
5	Октябрь	1 неделя	Комбинированное занятие	3	Добавление объектов на экран.	опрос, тестирование
6	Октябрь	2 неделя	Комбинированное занятие	3	Анимации и звук.	практическое задание
7	Октябрь	3 неделя	Комбинированное занятие	3	Как обрабатывать события от клавиатуры Практическая работа: как обрабатывать события от клавиатуры	тестирование, опрос
8	Октябрь	4 неделя	Комбинированное занятие	3	Как обрабатывать события от мыши Практическая работа: как обрабатывать события от мыши	опрос
9	Ноябрь	1 неделя	Комбинированное занятие	3	Событие pygame.QUIT.	тестирование, опрос
10	Ноябрь	2 неделя	Комбинированное занятие	3	Создание поверхностей (Surface) и их анимация. Метод blit	практическое задание

11	Ноябрь	3 неделя	Обучающее практическое занятие	3	Передвижение и прыжок.	практическое задание
12	Ноябрь	4 неделя	Обучающее практическое занятие	3	Класс Rect.	практическое задание
13	Декабрь	1 неделя	Обучающее практическое занятие	3	Враги в игре	практическое задание
14	Декабрь	2 неделя	Комбинированное занятие	3	Как рисовать текст различными шрифтами	практическое задание
15	Декабрь	3 неделя	Обучающее практическое занятие	3	Экран проигрыша	практическое задание
16	Декабрь	4 неделя	Комбинированное занятие	3	Как работать с изображениями. Модули image и transform	тестирование, опрос
17	Январь	1 неделя	Комбинированное занятие	3	Добавляем звук в игровой процесс. mixer и music	коллективная рефлексия, дискуссия, тестирование
18	Январь	2 неделя	Практическое занятие	3	Что такое спрайты и как с ними работать	практическое задание
19	Январь	3 неделя	Практическое занятие	3	Как делать контроль столкновений	практическое задание
20	Январь	4 неделя	Обучающее практическое занятие	3	Стрельба по врагам.	практическое задание
21	Февраль	1 неделя	Практическое занятие	3	Улучшение логики столкновений	практическое задание
22	Февраль	2 неделя	Практическое занятие	3	Простой счётчик очков	практическое задание
23	Февраль	3 неделя	Обучающее практическое занятие	3	Улучшение счётчика очков	практическое задание
24	Февраль	4 неделя	Практическое занятие	3	Музыка, звуковые эффекты	практическое задание
25	Март	1 неделя	Практическое занятие	3	Счётчик жизней	практическое задание
26	Март	2 неделя	Практическое занятие	3	Пополнение жизней	практическое задание

27	Март	3 неделя	Комбинированное занятие	3	Кнопки	практическое задание
28	Март	4 неделя	Комбинированное занятие	3	Циклы событий	тестирование, опрос
29	Апрель	1 неделя	Комбинированное занятие	3	Размещение предметов на экране	коллективная рефлексия, дискуссия, тестирование
30	Апрель	2 неделя	Практическое занятие	3	Воспроизведение звуковых эффектов и музыки	практическое задание
31	Апрель	3 неделя	Практическое занятие	3	Обработка пользовательского ввода	практическое задание
32	Апрель	4 неделя	Практическое занятие	3	Скорость игры	практическое задание
33	Май	1 неделя	Практическое занятие	3	Добавление изображений	практическое задание
34	Май	2 неделя	Практическое занятие	3	Таблица рекордов	практическое задание
35	Май	3 неделя	Практическое занятие	3	Создание APK файла	практическое задание
36	Май	4 неделя	Практическое занятие	3	Зачет по курсу «Обучение библиотеки PyGame». Презентация игр.	практическое задание
<b>ИТОГО:</b>				<b>108</b>		

### Базовый уровень

№	месяц	неделя	Форма занятия	Кол-во часов	Разделы и темы	Форма контроля
1	Сентябрь	1 неделя	Комбинированное занятие	2	Язык Python. Среда программирования . Особенности ввода-вывода	коллективная рефлексия, тестирование
2	Сентябрь	1 неделя	Комбинированное занятие	2	Типы данных, операции. Оператор присваивания.	опрос

3	Сентябрь	2 неделя 3 неделя	Обучающее практическое занятие	6	Числа. Стандартные операции.	практическое задание
4	Сентябрь	3 неделя 4 неделя	Практическое занятие	4	Условный оператор	практическое задание
5	Сентябрь Октябрь	4 неделя 1 неделя 2 неделя	Комбинированное занятие	8	Цикл while. Теория чисел.	опрос, тестирование
6	Октябрь	2 неделя 3 неделя 4 неделя	Комбинированное занятие	8	Цикл for	практическое задание
7	Октябрь Ноябрь	4 неделя 1 неделя	Комбинированное занятие	4	Литералы строк	тестирование, опрос
8	Ноябрь	1 неделя 2 неделя	Комбинированное занятие	4	Срезы строк	опрос
9	Ноябрь	2 неделя 3 неделя 4 неделя	Комбинированное занятие	10	Методы строк	тестирование, опрос
10	Декабрь	1 неделя	Комбинированное занятие	4	Парадигма и преимущества структурного программирования	практическое задание
11	Декабрь	2 неделя	Обучающее практическое занятие	4	Граф и стек вызовов функции. Области видимости переменных	практическое задание
12	Декабрь Январь	3 неделя 4 неделя 1 неделя	Обучающее практическое занятие	10	Прямая рекурсия	практическое задание
13	Январь	1 неделя 2 неделя 3 неделя	Обучающее практическое занятие	8	Косвенная рекурсия	практическое задание
14	Январь	3 неделя 4 неделя	Комбинированное занятие	6	Списки и кортежи в Python. Сходства и различия	практическое задание
15	Февраль	1 неделя 2 неделя 3 неделя	Обучающее практическое занятие	10	Операции со списками	практическое задание
16	Февраль	3 неделя	Комбинированное занятие	2	Срезы списков	тестирование, опрос
17	Февраль	4 неделя	Комбинированное занятие	4	Матрицы. Операции над матрицами.	коллективная рефлексия, дискуссия,

						тестирование
18	Март	1 неделя	Практическое занятие	4	Словари	практическое задание
19	Март	2 неделя	Практическое занятие	4	Множества	практическое задание
20	Март	3 неделя	Обучающее практическое занятие	2	Классы в Python	практическое задание
21	Март Апрель	3 неделя 4 неделя 1 неделя	Практическое занятие	10	Разработка собственного класса	практическое задание
22	Апрель Май	2 неделя 3 неделя 4 неделя 1 неделя 2 неделя	Практическое занятие	20	Разработка и программирование собственного проекта	практическое задание
23	Май	3 неделя 4 неделя	Обучающее практическое занятие	8	Подведение итогов. Индивидуальный проект	практическое задание
<b>ИТОГО:</b>				<b>144</b>		

### Продвинутый уровень

№	месяц	неделя	Форма занятия	Кол-во часов	Разделы и темы	Форма контроля
1	Сентябрь	1 неделя	Комбинированное занятие	3	Знакомство с внутренней структурой компьютера с точки зрения программ. Понятие алгоритм и программа. Виды языков программирования . Общие сведения о языке Python.	коллективная рефлексия, тестирование
2	Сентябрь	1 неделя	Комбинированное занятие	3	Ввод и вывод данных на языке Python. Параметры вывода. Ввод и отладка программ в среде.	опрос

3	Сентябрь	2 неделя	Обучающее практическое занятие	3	Переменные в Python, основные понятия.	практическое задание
4	Сентябрь	2 неделя	Практическое занятие	3	Целочисленная арифметика. Алгоритмы работы с числами.	практическое задание
5	Сентябрь	3 неделя	Комбинированное занятие	6	Текущий контроль. Повторение пройденного материала. Создание программы по индивидуальному заданию	опрос, тестирование
6	Сентябрь	4 неделя	Комбинированное занятие	3	Понятие условных операторов и методы их применения на практике. Знакомство с понятием True и False.	практическое задание
7	Сентябрь	4 неделя	Комбинированное занятие	3	Логические операции not/and/or	тестирование, опрос
8	Октябрь	1 неделя	Комбинированное занятие	6	Текущий контроль. Решение задач и создание программы для закрепления полученных знаний и навыков	опрос
9	Октябрь	2 неделя	Комбинированное занятие	3	Знакомство с понятием «типы числовых данных»	тестирование, опрос
10	Октябрь	2 неделя	Комбинированное занятие	3	Модуль math: определение, подключение и варианты использования.	практическое задание
11	Октябрь	3 неделя	Обучающее практическое занятие	3	Строковый тип данных. Работа с строками len(),	практическое задание

					str()	
12	Октябрь	3 неделя	Обучающее практическое занятие	3	Способы использования строковых данных. Оператор in	практическое задание
13	Октябрь	4 неделя	Обучающее практическое занятие	6	Текущий контроль. Создание программы для закрепления полученных знаний и навыков	практическое задание
14	Ноябрь	1 неделя	Комбинированное занятие	3	Общие понятия о циклах и о функциях. Знакомство с циклом for в Python	практическое задание
15	Ноябрь	1 неделя	Обучающее практическое занятие	3	Функция range() и параметры.	практическое задание
16	Ноябрь	2 неделя	Комбинированное занятие	3	Наиболее частые сценарии использования цикла	тестирование, опрос
17	Ноябрь	2 неделя	Комбинированное занятие	3	Знакомство с циклом while в Python	коллективная рефлексия, дискуссия, тестирование
18	Ноябрь	3 неделя	Практическое занятие	3	Практическое использование цикла while для закрепления знаний.	практическое задание
19	Ноябрь	3 неделя	Практическое занятие	3	Использование операторов continue, break и else в циклах	практическое задание
20	Ноябрь	4 неделя	Обучающее практическое занятие	3	Анализ кода написанного обучающимися и выявление типичных ошибок и способов их обнаружения.	практическое задание



21	Ноябрь	4 неделя	Практическое занятие	3	Оптимизация и читабельность кода. Разбор ситуаций.	практическое задание
22	Декабрь	1 неделя	Практическое занятие	3	Вложенные циклы. Где применяются и для чего.	практическое задание
23	Декабрь	1 неделя	Обучающее практическое занятие	3	Индексация.	практическое задание
24	Декабрь	2 неделя	Комбинированное занятие	3	Срезы.	практическое задание
25	Декабрь	2 неделя	Комбинированное занятие	3	Методы Строк. Обзор.	практическое задание
26	Декабрь	3 неделя	Комбинированное занятие	3	Строки в памяти компьютера	практическое задание
27	Декабрь	3 неделя 4 неделя	Практическое занятие	6	Текущий контроль. Написание программы по творческим Заданиям	практическое задание
28	Декабрь	4 неделя	Практическое занятие	3	Промежуточный мониторинг по 1-4 разделам	практическое задание
29	Январь	1 неделя	Комбинированное занятие	3	Введение в списки. Основы работы со списками.	практическое задание
30	Январь	1 неделя	Комбинированное занятие	3	Методы списков.	практическое задание
31	Январь	2 неделя	Комбинированное занятие	3	Вывод элементов списка	практическое задание
32	Январь	2 неделя	Комбинированное занятие	3	Методы строк split, join	практическое задание
34	Январь	3 неделя	Комбинированное занятие	3	Списочные выражения	практическое задание
35	Январь	3 неделя	Комбинированное занятие	3	Сортировка списков	практическое задание
36	Январь	4 неделя	Комбинированное занятие	3	Кортежи	практическое задание
37	Январь	4 неделя	Комбинированное занятие	3	Введение в словари	практическое задание
38	Февраль	1 неделя	Практическое	6	Написание	практическое

			занятие		программы по творческим Заданиям	задание
39	Февраль	2 неделя	Комбинированное занятие	3	Функции без параметров и с параметрами	практическое задание
40	Февраль	2 неделя	Комбинированное занятие	3	Функции с возвратом значений	практическое задание
41	Февраль	3 неделя	Комбинированное занятие	3	Модули. Модуль random	практическое задание
42	Февраль	3 неделя	Практическое занятие	3	Текущий контроль. Демонстрация программы учащегося, разбор ошибок и возможности улучшения	практическое задание
43	Февраль	4 неделя	Комбинированное занятие	3	Типы файлов с точки зрения программиста. Запись информации в файл, чтение из файла	практическое задание
44	Февраль	4 неделя	Комбинированное занятие	3	Работа с текстовыми файлами: запись, чтение изменение. Решение задач.	практическое задание
45	Март	1 неделя	Практическое занятие	6	Создание программы для закрепления полученных знаний и навыков	практическое задание
46	Март	2 неделя	Комбинированное занятие	3	Концепция ООП. Объекты и классы.	практическое задание
47	Март	2 неделя	Комбинированное занятие	3	Принципы ООП: Рекурсия	практическое задание
48	Март	3 неделя	Комбинированное занятие	3	Программирование объектной модели. Решение задач.	практическое задание
49	Март	3 неделя	Комбинированное занятие	3	Принципы ООП: инкапсуляция.	практическое задание

50	Март	4 неделя	Комбинированное занятие	3	Принципы ООП: наследование и полиморфизм	практическое задание
51	Март	4 неделя	Комбинированное занятие	3	Объектно - ориентированное программирование . Решение задач	практическое задание
52	Апрель	1 неделя 2 неделя	Практическое занятие	12	Текущий контроль. Создание программы для закрепления полученных знаний и навыков	практическое задание
53	Апрель	3 неделя	Практическое занятие	3	Разделение на группы и получение вводных для проекта	практическое задание
54	Апрель Май	3 неделя 4 неделя 1 неделя 2 неделя	Практическое занятие	21	Работа в группах	практическое задание
55	Май	3 неделя	Практическое занятие	3	Итоговый мониторинг	практическое задание
56	Май	3 неделя 4 неделя	Практическое занятие	6	Презентация результатов работы	практическое задание
<b>ИТОГО:</b>				<b>216</b>		

## **Условия реализации программы**

### **Материально-техническое обеспечение**

Условия реализации программы: учебный кабинет для педагога, оснащенный оборудованием (стандарт).

– компьютер или ноутбук, на которых установлено соответствующее программное обеспечение: (подключение к единой Wi-Fi сети с доступом в интернет);

– презентационное оборудование.

Для обучающегося стабильный выход в сеть Интернет.

### **Кадровое обеспечение**

Программа реализуется педагогами дополнительного образования, имеющими опыт работы в учреждении дополнительного образования технической направленности, образование – не ниже средне-профессионального, профильное или педагогическое.

### **Формы контроля и подведения итогов реализации программы**

На занятиях используются: входной и текущий контроль, итоговый контроль.

Входной контроль осуществляется через наблюдение за деятельностью учащихся, предполагает собеседование с учащимися, в ходе которого определяется наличие у них минимального необходимого уровня входных компетенций: уверенный пользователь ПК.

Текущий контроль осуществляется посредством наблюдения за деятельностью учащихся на каждом занятии и фиксации их умений во время работы над практическими заданиями/работами по разделам, тестами. Отмечается активность участия учащихся в мероприятиях, степень самостоятельности при работе над практическими заданиями, самостоятельный поиск и разработка интересных тем для доклада (или мини-проекта) по направлениям программы. Промежуточный и итоговый контроль предполагает разработку и реализацию проекта, выступление с демонстрацией результатов работы с применением дистанционных технологий.

## МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДОПОЛНИТЕЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Педагогические технологии, методы, приемы и формы организации образовательного процесса

При реализации программы используются следующие педагогические технологии:

– технологии развивающего обучения, направленные на общее целостное развитие личности, на основе активно-деятельного способа обучения, учитывающие закономерности развития и особенности индивидуума;

– технологии личностно-ориентированного обучения, направленные на развитие индивидуальных познавательных способностей каждого ребенка, максимальное выявление, раскрытие и использование его опыта;

– технологии дифференцированного обучения, обеспечивающие обучение каждого обучающегося на уровне его возможностей и способностей;

– технологии сотрудничества, реализующие демократизм, равенство, партнерство в отношениях педагога и обучающегося, совместно вырабатывают цели, содержание, дают оценки, находясь в состоянии сотрудничества, сотворчества.

– проектные технологии – достижение цели через детальную разработку проблемы, которая должна завершиться реальным, осязаемым практическим результатом, оформленным тем или иным образом;

– компьютерные технологии, формирующие умение работать с информацией, исследовательские умения, коммуникативные способности. В практике выступают различные комбинации этих технологий, их элементов.

Основным методом организации учебной деятельности по программе является метод кейсов.

Кейс – описание проблемной ситуации понятной и близкой обучающимся, решение которой требует всестороннего изучения, поиска дополнительной информации и моделирования ситуации или объекта, с выбором наиболее подходящего.

Преимущества метода кейсов.

– Практическая направленность. Кейс-метод позволяет применить теоретические знания к решению практических задач.

– Интерактивный формат.

Кейс-метод обеспечивает более эффективное усвоение материала за счет высокой эмоциональной вовлеченности и активного участия обучаемых. Участники погружаются в ситуацию с головой.

– Конкретные навыки. Кейс-метод позволяет совершенствовать «гибкие навыки» (soft skills), которым не учат в университете, но которые оказываются крайне необходимы в реальном рабочем процессе.

В ходе работы над кейсом целесообразно использовать следующие методы, приемы, средства и формы организации, внесенные в таблицу.

№	Формы организации	Методы и приемы	Дидактический материал	Формы контроля
1	Лекция с разбором решения практического задания	устное изложение с визуальным рядом, позволяющий в доступной форме донести до обучающихся материал;	Видео, приложения, шаблоны файлов, веб-доски и веб-плакаты.	Проверка синхронного выполнения материала лекции.
2	Обсуждение, рефлексия рассмотренных тем	устный опрос в ходе демонстрации видеоряда	Видео-презентация	рефлексивный самоанализ, контроль и самооценка обучающихся
3	Практическое задание, сходное с разбиравшимся на лекции;	репродуктивный практический метод; частично-поисковый	Видео, шаблоны исходных файлов, вебплакаты	Просмотр хода выполнения; обсуждение итогов
4	Проект	исследовательский метод практический метод частично-поисковый	Веб-доски и вебдокументы, видео, инструкции по работе над проектом, шаблоны файлов	Защита проекта
5	Соревнование	практический	Веб-доски и	Подведение

		метод	вебдокументы	итогов
6	Исследование	исследовательски й метод	Презентация, видео, описание хода исследования и т.д.	Конференция

## Список литературы

1. Гуриков С.Р. Основы алгоритмизации и программирования на Python / С.Р. Гуриков. – М.: «Форум», 2018 – 343 с. Текст непосредственный.
2. Поляков К.Ю. Программирование. Python.C++: учебное пособие / К.Ю. Поляков. М.: Бином. Лаборатория знаний, 2019. В 4-х т. Текст непосредственный.
3. Кормен Томас Алгоритмы. Построение и анализ. 5 издание/Томас Кормен, Чарльз Лейзерсон, Рональд Ривест и Клиффорд Штайн. – М.: Вильямс, 2018. – 1324 с. Текст непосредственный.
4. Никулин С.К., Полтавец Г.А., Полтавец Т.Г. Содержание научно-технического творчества учащихся и методы обучения. М.: Изд. МАИ. 2022. Текст непосредственный.
5. Полтавец Г.А., Никулин С.К., Ловецкий Г.И., Полтавец Т.Г. Системный подход к научно-техническому творчеству учащихся (проблемы организации и управления). УМП. М.: Издательство МАИ. 2020. Текст непосредственный.
6. Прохоренок Н. «Python 3 и PyQt 5. Разработка приложений» / Н. Прохоренок, В. Дронов. СПб: «БХВ-Петербург», 2019. – 832 с. Текст непосредственный.
7. Свейгар.Эл. Учим Python, делая крутые игры М: Эксмо, 2018. – 416 с. Текст непосредственный.
8. Советов Б. Я., Цехановский В.В. – ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ 7 изд., пер. и доп. Учебник – М.: Издательство Юрайт – 2019 – 327с. Текст электронный.  
<https://znanium.com/bookread2.php?book=1031314>
9. Дети и технологии / Т.А. Аймалетдинов; Аналитический центр НАФИ. – М.: НАФИ, 2018. -72с. Текст электронный.  
<https://znanium.com/bookread2.php?book=1031314>
10. Ли Воган. «Непрактичный» Python занимательные проекты для тех, кто хочет поумнеть. – СПб.: БХВ-Петербург, 2021. – 457 с. Текст непосредственный.