

Муниципальное бюджетное общеобразовательное учреждение  
средняя общеобразовательная школа  
с углубленным изучением отдельных предметов №6  
357500, г. Пятигорск, ул. Университетская, 6, тел./факс 33-00-16, sch06.5gor@bk.ru



СОГЛАСОВАНО  
Заседанием Педагогического совета  
Протокол № 1 от 21.08.2024



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА**  
**«ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»**  
с использованием оборудования  
«Школьного Кванториума»

Направленность: техническая  
Общий объем программы: 162 часа  
Возраст обучающихся: 15-17 лет  
Срок реализации программы: 1 год  
Уровень: углубленный

Пятигорск, 2024

## Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» разработана в соответствии со следующими нормативными документами:

- Федеральный закон от 29.12.2012 N 273-ФЗ (ред. От 02.07.2021) "Об образовании в Российской Федерации" (с изм. и доп., вступ. в силу с 01.09.2021)
- Приказ Министерства просвещения РФ от 9 ноября 2018 г. N 196 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам"(с изменениями и дополнениями от 30 сентября 2020 года);
- СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» - Постановление Главного государственного санитарного врача от 28.09.2020 №28;
- СП 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» - Постановление Главного государственного санитарного врача от 28.01.2021 №2.

При разработке дополнительной общеобразовательной общеразвивающей программы (далее Программа) учтены требования, отраженные в государственных стандартах Российской Федерации и в федеральных государственных стандартах общего образования.

**Актуальность программы обусловлена** широким распространением информационно-коммуникационных технологий в обществе и необходимостью обеспечивать связанную с этим инфраструктуру специалистами. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём на уровне начинающего программиста.

Изучение основных принципов программирование невозможно без регулярной практики написания программ на каком-либо языке программирования. Для обучения в рамках программы выбран язык Python, который является достаточно эффективным и доступным инструментом достижения задач в области создания программных продуктов.

Синтаксис языка достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на алгоритмических аспектах программирования.

**Отличительная особенность программы, новизна:** Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

**Новизна** программы заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett- Packard, используют этот язык при реализации своих проектов. Крупнейшие интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python.

### **Цель и задачи программы**

**Цель программы:** 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.

#### **Задачи программы:**

- Познакомить с основными алгоритмическими конструкциями и правилами их записи, с основными способами организации данных.
- Научить составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций.
- Научить распознавать необходимость применения той или иной алгоритмической

конструкции при решении задач.

- Научить организовывать данные для эффективной алгоритмической обработки.
- Научить разрабатывать алгоритмы и реализовывать их на языке программирования Python.
- Научить осуществлять отладку и тестирование программы.

### **Сроки и целевая аудитория:**

Программа рассчитана на 1 год обучения (162 часа), возраст 15- 17 лет (10-11 классы.)

Зачисление в группы производится с обязательным условием – написание заявления родителями (законными представителями несовершеннолетних учащихся), подписание согласия на обработку персональных данных. Допуск к занятиям производится только после обязательного проведения и закрепления инструктажа по технике безопасности по соответствующим инструкциям.

**Режим занятий:** занятия проходят с периодичностью 2 раза в неделю по 2 урока (1 урок – 45 минут).

**Основные формы занятий:** групповая и индивидуальная работа.

**Виды занятий:** тематические, комбинированные, игровые занятия, практикум.

### **Планируемые результаты освоения программы**

Предметные:

- 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, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- формирование умения соблюдать нормы информационной этики и права.

Метапредметные:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;

- умение критически оценивать правильность решения учебно- исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владение основами самоконтроля, способность к принятию решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ- компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

#### Личностные:

- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

#### аттестации/контроль

**Входящий контроль** осуществляется при комплектовании группы в начале учебного года. *Цель* – определить исходный уровень знаний учащихся, определить формы и методы работы с учащимися.

*Форма контроля:* тестирование.

**Текущий контроль** осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они активизируют, стимулируют работу учащихся, позволяют более полно проявлять полученные знания, умения, навыки.

**Промежуточный контроль** осуществляется в конце I полугодия учебного года.

Форма контроля: тест, решение практических задач средствами языка программирования.

**Итоговый контроль** осуществляется в конце учебного года. Форма контроля: защита творческого проекта.

## Оценочные материалы

Диагностируются результаты различным образом и на различных этапах деятельности. Для определения уровня знаний и умений

обучающихся проводятся: в начале года - входной контроль, в середине года - промежуточный контроль, в конце года — итоговый контроль. Основной способ оценки – самооценка выполнения программ на языке программирования. Эта оценка присутствует на всех этапах. Одной из форм диагностики результатов является зачет или контрольная работа.

**Входящий контроль:** *Тестирование (Приложение 2)*

**Промежуточный контроль:** *Тестирование, решение задач (Приложение 3)*

**Итоговый контроль:** учащимся предлагается самостоятельно выбрать тему творческого проекта и на основании темы разработать программу, пояснительную записку, презентацию. Содержание программы

**Вводное занятие (2 час).**

Теория: Знакомство с планом работы объединения, инструктаж по ТБ. Практика: Опрос. Тренинг на командообразование.

### Раздел 1. Знакомство с языком Python (6 часов)

**Теория:** Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python

Практическая работа 1.2. Режимы работы с Python

Тест № 1. Знакомство с языком Python

*Учащиеся должны знать / понимать:*

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

*Учащиеся должны уметь:*

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

### Раздел 2. Переменные и выражения (16 часов)

**Теория:** Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задача элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой

Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами

Тест № 2. Выражения и операции.

*Учащиеся должны знать / понимать:*

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;

### Раздел 3. Условные предложения (10 часов)

**Теория:** Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа 3.1. Логические выражения

Практическая работа 3.2. "Условный оператор"

Практическая работа 3.3. Множественное ветвление

Практическая работа 3.4. "Условные операторы"

Самостоятельная работа № 1. Решение задач по теме "Условные операторы".

Зачетная работа № 1. "Составление программ с ветвлением".

Тест № 3. "Условные операторы".

*Учащиеся должны знать / понимать:*

- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- Учащиеся должны уметь:
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

#### **Раздел 4. Циклы (16 часов)**

**Теория:** Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы.

Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 4.1. "Числа Фибоначчи"

Практическая работа 4.2. Решение задачи с циклом for. Практическая работа 4.3. Реализация циклических алгоритмов

Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом. Самостоятельная работа № 2

"Составление программ с циклом"

Тест № 4. Циклы

Творческая работа № 1. "Циклы"

*Учащиеся должны знать / понимать:*

- циклы с условием и их виды;
- правила записи циклов с условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.

*Учащиеся должны уметь:*

- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

#### **Раздел 5. Функции (18 часов)**

**Теория:** Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.

Практическая работа 5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций

Практическая работа 5.4. Рекурсивные функции

Самостоятельная работа № 3 по теме "Функции"

Тест № 5. Функции

*Учащиеся должны знать / понимать:*

- понятие функции;
- способы описания функции;

- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

*Учащиеся должны уметь:*

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

### **Раздел 6 Встроенные модули: random, decimal и fractions. Модуль turtle (12)**

Теория Случайные числа. Псевдослучайные числа Модуль random. Методы shuffle(), choice(), sample(). Модуль string Ещё раз о числовых типах данных. Числа с плавающей точкой float. Модуль decimal. Тип данных Decimal Модуль fractions. Рациональное число. Тип данных Fraction Арифметические операции над Fraction числами. Свойства numerator и denominator. Метод limit\_denominator() Черепашка графика. Модуль turtle Рисование отрезков прямой. Поворот черепашки. Установка углового направления черепашки. Изменение внешнего вида черепашки

Практическая работа 6.1. Решение задач со встроенными модулями

Тест № 6 встроенные модули

### **Раздел 7. Строки - последовательности символов (16 часов)**

Теория: Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 7.1. Строки

Практическая работа 7.2. Решение задач со строками.

Тест № 7 Строки

*Учащиеся должны знать / понимать:*

- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками.

*Учащиеся должны уметь:*

- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке.

### **Раздел 8. Сложные типы данных (20 часов)**

Теория: Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа 8.1. Списки.

Практическая работа 8.2. Решение задач со списками.

Тест № 8. Списки

*Учащиеся должны знать / понимать:*

- сложные типы данных;
- способ описания списка;

- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.

*Учащиеся должны уметь:*

- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.

### **Раздел 9. Стиль программирования и отладка программ (10 часов)**

Стиль программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

*Учащиеся должны знать / понимать:*

- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

*Учащиеся должны уметь:*

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

### **Раздел 10. Работа с датой и временем. Модули date и time (8 часов)**

Тип данных date. Тип данных time Сравнение дат и времени. Функции str() и repr()  
 Форматирование даты и времени Использование локализации. Преобразование строки в дату и время  
 Тип данных timedelta. Атрибуты days, seconds, microseconds. Метод total\_seconds().  
 Сравнение временных интервалов. Операции над временными интервалами timedelta

Практическая работа 10.1. Дата время

### **Раздел 11. Машинное обучение (12 часов)**

Знакомство с процессом создания статистических моделей для нахождения закономерностей на основе подготовленных массивов данных, основными моделями машинного обучения, нейронными сетями, техникой машинного зрения: извлечение признаков, поиск по картинкам, сегментирование, детекция объектов.

### **Раздел 12 Работа над проектом (12 часов)**

Индивидуальное задание. Разработка концепции программы по заданию. Написание и отладка программы. Презентация программного продукта.



## Тематическое планирование

№ п/п	Тема занятий	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
	Инструктаж по технике безопасности в компьютерном классе	2	2		Опрос
1.	Раздел 1. Знакомство с языком Python	16	8	8	Тестирование, решение практических задач
2.	Раздел 2. Переменные и выражения	10	4	6	Тестирование, решение практических задач
3	Раздел 3. Условные предложения	10	4	6	Тестирование, Решение практических задач
4	Раздел 4. Циклы	16	6	10	Тестирование, решение Практических задач,
5	Раздел 5. Функции	18	6	12	Тестирование, решение практических задач
6	Раздел 6 Встроенные модули: random, decimal и fractions. Модуль turtle	12	4	8	Тестирование, решение практических задач
7	Раздел 7. Строки - последовательности символов	16	7	9	Решение практических задач
8	Раздел 8. Сложные типы данных	20	8	12	Тестирование, Решение практических задач
9	Раздел 9. Стиль программирования и отладка программ	10	3	7	Решение практических задач, презентация проекта, рефлексия.
	Раздел 10. Работа с датой и временем. Модули date и time	8	2	6	Тестирование, Решение практических
	Раздел 11 Машинное обучение	12	6	6	Тестирование, Решение практических
	Раздел 12 Работа над проектом	12	2	10	презентация проекта, рефлексия
	<b>ВСЕГО</b>	<b>162</b>	<b>60</b>	<b>102</b>	

## **Материально-технические условия реализации программы:**

### **Материально-техническое обеспечение:**

**Помещение:** кабинет для занятий, соответствующий санитарно-гигиеническим требованиям.

Для реализации образовательной программы используется оборудование, полученное в рамках «Школьный Кванториум».

### **Информационно-методическое обеспечение:**

1. персональный компьютер (**ноутбук**);
2. мультимедийный проектор;
3. видеоматериалы разной тематики по программе;
4. оргтехника;
5. выход в сеть Internet;

### **Аппаратное обеспечение:**

1. Процессор не ниже Core2 Duo;
2. Объем оперативной памяти не ниже 4 ГбDDR3;
3. Дисковое пространство на менее 128 Гб;
4. Монитор диагональю на мене 19”;

### **Программное обеспечение:**

1. Операционная система Windows 7 и выше или Linux;
2. Интерпретатор Python версии 3.8 и выше;
3. IDE JupyterLab, PyCharm, Thonny;
4. Foxit Reader или другой просмотрщик PDF файлов;
5. WinRAR;
6. Пакет офисных программ;
7. Adobe Photoshop или другой растровый графический редактор;
8. Любой браузер для интернет-серфинга.

### Литература для обучающихся:

1. Сайт/справочные материалы [Электронный ресурс] – Режим доступа:

<https://metanit.com/python/>, свободный.

Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.

2. Сайт/Адаптивный тренажер Python [Электронный ресурс] – Режим доступа:

<https://stepik.org/course/431> , свободный.

3. Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu> , свободный.

4. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: [openbookproject.net](http://openbookproject.net) , свободный.

### Литература для педагога

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.

2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.

3. Официальный сайт программы [Электронный ресурс] – Режим доступа:

<https://docs.python.org/>, свободный.

4. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.