

**Министерство образования и молодежной политики Свердловской области  
Департамент образования администрации города Екатеринбурга  
МАОУ СОШ № 76 с углубленным изучением отдельных предметов**

**СОЛАСОВАНО**

Педагогическим советом  
Протокол № 10  
от «26» июня 2024 г.



**Дополнительная общеобразовательная общеразвивающая программа  
технической направленности  
«Программирование на Python»  
для обучающихся 8 – 11 классов**

**Составитель:  
Порубова Е.В., ВКК**

## 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

**Направленность ДООП – техническая.**

**Актуальность программы** состоит в том, что она позволяет средствами дополнительного образования приобщить учащихся к основам программирования, способствовать расширению уровня знаний и ИТ-компетентностей, способности к самообучению, ранней профориентации.

Программа имеет техническую направленность, подразумевает индивидуальную работу с компьютером в редакторе IDLE. В то же время это совсем не означает, что учащиеся на занятиях не будут работать устно и в группах; наоборот, часто решение какого-либо задания приходит в процессе общения и обсуждения конкретных задач.

**Новизна** Программы заключается в изучении специфики работы профессий сферы компьютерных и информационных технологий (ИТ): программиста, специалиста по тестированию в области ИТ, разработчика Web и мультимедийных приложений и т.д., а также смежных профессий, например, технического писателя, специалиста по информационным ресурсам, техника по защите информации и многие другие.

**Адресат программы:**

Программа адресована учащимся в возрасте от 14 до 16 лет и реализуется с учетом их возрастных, психических и психофизических особенностей. Она содержит элементы машинного курса, а это означает индивидуальный подход к каждому учащемуся и легкое отслеживание динамики развития его навыков.

**Педагогическая целесообразность** состоит в том, что данная Программа позволяет выявить учащихся, проявивших интерес к данному направлению, оказать им помощь в формировании устойчивого интереса к профессиям, связанным с информационными технологиями, освоить технологии дистанционного обучения, способствовать ранней профориентации. Опыт, полученный в процессе обучения по Программе (базовый уровень), станет теоретической и практической основой при выборе будущей профессии.

**Срок реализации программы:** 1 год.

**Уровень программы:** базовый.

**Объем программы:** 75 академических часов.

**Режим занятий:** Продолжительность одного академического часа - 45 мин. Общее количество часов в неделю – 2,5 часа. Занятие проводится 1 раз в неделю по 2,5 часа.

**Учебная группа:** 10 – 15 обучающихся.

**Форма организации образовательного процесса:** очная.

**Формы проведения занятий:**

- лекция;
- самостоятельная работа;
- лабораторные и практические занятия.

**Цель** программы – содействовать развитию творческих способностей учащихся и их интереса в области компьютерного программированию и искусственного интеллекта посредством изучения языка программирования

Python.

**Задачи программы:**

**Образовательные:**

- 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.

**Личностные:**

- повышение общекультурного уровня;
- вооружение правильным методологическим подходом к познавательной и практической деятельности;
- выделение и раскрытие роли информационных технологий и компьютеров в развитии современного общества;
- привитие навыков сознательного и рационального использования компьютера в своей учебной, а затем и профессиональной деятельности;
- воспитание стремления к овладению техникой исследования;
- воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

**Метапредметные:**

- владение умениями организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить;
- планирование – определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработка последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств;
- прогнозирование – предвосхищение результата.

**Содержание программы  
Учебный план**

№ п/п	Тема занятий	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
	<b>Инструктаж по технике безопасности в компьютерном классе</b>	<b>0,5</b>	<b>0,5</b>		Опрос
<b>1.</b>	<b>Раздел 1. Знакомство с языком Python</b>	<b>4,5</b>	<b>2</b>	<b>2,5</b>	Тестирование, решение практических задач
1.1	Занятие 1. Общие сведения о	2	1	1	

	языке Практическая работа: Установка программы Python				
1.2.	Занятие 2. Режимы работы Практическая работа: Режимы работы с Python	2,5	1	1,5	
<b>2.</b>	<b>Раздел 2. Переменные и выражения</b>	<b>10</b>	<b>4</b>	<b>6</b>	Тестирование, решение практических задач
2.1	Занятие 3. Переменные Практическая работа: Работа со справочной системой	2,5	1	1,5	
2.2	Занятие 4. Выражения Практическая работа: Выражения	2,5	1	1,5	
2.3	Занятие 5. Ввод и вывод	2,5	1	1,5	
2.4	Занятие 6. Задачи на элементарные действия с числами Практическая работа 2.5. Задачи на элементарные действия с числами	2,5	1	1,5	
<b>3</b>	<b>Раздел 3. Условные предложения</b>	<b>7,5</b>	<b>3</b>	<b>4,5</b>	
3.1	Занятие 7. Логические выражения и операторы. Практическая работа: Логические выражения	2,5	1	1,5	
3.2	Занятие 8. Условный оператор Практическая работа: «Условный оператор»	2,5	1	1,5	
3.3	Занятие 9. Множественное ветвление Практическая работа: Множественное ветвление Практическая работа: «Условные операторы»	2,5	1	1,5	
<b>4</b>	<b>Раздел 4. Циклы</b>	<b>10</b>	<b>4</b>	<b>6</b>	Тестирование, решение практических задач, творческая работа
4.1.	Занятие 10. Оператор цикла с условием Практическая работа "Числа Фибоначчи"	2,5	1	1,5	
4.2.	Занятие 11. Оператор цикла for Практическая работа Решение задачи с циклом for.	2,5	1	1,5	
4.3.	Занятие 12. Вложенные циклы Практическая работа:	2,5	1	1,5	

	Реализация циклических алгоритмов				
4.4.	Занятие 13. Случайные числа Практическая работа: Случайные числа	2,5	1	1,5	
<b>5</b>	<b>Раздел 5. Функции</b>	<b>10</b>	<b>4</b>	<b>6</b>	
5.1.	Занятие 14. Создание функций Практическая работа Создание функций	2,5	1	1,5	Тестирование, решение практических задач
5.2.	Занятие 15. Локальные переменные Практическая работа Локальные переменные	2,5	1	1,5	
5.3.	Занятие 16. Примеры решения задач с использованием функций Практическая работа Решение задач с использованием функций	2,5	1	1,5	
5.4	Занятие 17. Рекурсивные функции Практическая работа 5.4. Рекурсивные функции	2,5	1	1,5	
<b>6</b>	<b>Раздел 6. Строки - последовательности символов</b>	<b>7,5</b>	<b>3</b>	<b>4,5</b>	
6.1.	Занятие 18. Строки Практическая работа: Строки	2,5	1	1,5	Решение практических задач
6.2.	Занятие 19. Срезы строк	2,5	1	1,5	
6.3	Занятие 20. Примеры решения задач со строками Практическая работа: Решение задач со строками.	2,5	1	1,5	
<b>7</b>	<b>Раздел 7. Сложные типы данных</b>	<b>17,5</b>	<b>7</b>	<b>10,5</b>	
7.1.	Занятие 21. Списки Практическая работа: Списки	2,5	1	1,5	Тестирование, Решение практических задач
7.2	Занятие 22. Срезы списков	2,5	1	1,5	
7.3	Занятие 23. Списки: примеры решения задач Практическая работа 7.2. Решение задач со списками	2,5	1	1,5	
7.4	Занятие 24. Матрицы	2,5	1	1,5	
7.5	Занятие 25. Кортежи	2,5	1	1,5	

7.6	Занятие 26. Введение в словари	2,5	1	1,5	Решение практических задач, презентация проекта, рефлексия.
7.7	Занятие 27. Множества в языке Python	2,5	1	1,5	
<b>8</b>	<b>Раздел 8. Стил программирования и отладка программ</b>	<b>7,5</b>	<b>2</b>	<b>5,5</b>	
8.1	Занятие 28. Стил программирования	2,5	1	1,5	
8.2	Занятие 29. Отладка программ	2,5	1	1,5	
8.3	Занятие 30. Зачет по курсу «Программирование на языке Python»	2,5		2,5	
	<b>ВСЕГО</b>	<b>75</b>	<b>29,5</b>	<b>45,5</b>	

### Содержание учебного плана.

#### Вводное занятие (0,5 часа).

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

#### Раздел 1. Знакомство с языком Python (4,5 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 (10 часов)

Теория: Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Магические функции. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на 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 (7,5 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 (10 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 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 (10 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. Строки - последовательности символов (7,5 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 6.1. Строки Практическая работа 6.2. Решение задач со строками.

*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 7. Сложные типы данных (17,5 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 7.1. Списки.

Практическая работа 7.2. Решение задач со списками. Тест № 7. 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 8. Стиль программирования и отладка программ (7,5 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.

### **Планируемые результаты.**

#### **Предметные результаты:**

– 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 результаты:**

– 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 результаты:**

- 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.

## **2. Комплекс организационно-педагогических условий**

### **Календарный учебный график.**

Продолжительность одного занятия – 40 минут

Перерыв между занятиями составляет 10 минут.

<b>Продолжительность учебного года</b>	<b>Количество учебных недель</b>	<b>Общий объем учебных занятий по программе</b>	<b>Количество занятий в неделю для 1 группы</b>
с 01 сентября до 31 мая	30	75 часов	2,5 часа

Календарный учебный график составляется педагогом дополнительного образования на текущий учебный год на основании учебного плана и оформляется в виде таблицы:

№ п/п	Месяц	Число	Форма занятия	Кол-во часов	Тема занятия	Место проведения	Форма контроля

Календарный учебный график ежегодно утверждается руководителем образовательной организации.

### **Условия реализации программы**

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

Для реализации образовательной программы используются:

- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- специальная и научно-популярная литература для педагога и учащихся;
- канцтовары.

#### ***Информационное обеспечение***

1. персональный компьютер (12 шт.);
2. мультимедийный проектор;
3. видеоматериалы разной тематики по программе;
4. оргтехника;
5. выход в сеть Internet;

#### ***Кадровое обеспечение***

К занятию педагогической деятельностью по ДООП допускаются лица, обучающиеся по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующим направленности дополнительных общеобразовательных программ, и успешно прошедшие промежуточную аттестацию не менее чем за два года обучения. Соответствие образовательной программы высшего образования направленности дополнительной общеобразовательной программы определяется работодателем.

#### **Формы аттестации:**

**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.

Отслеживание личностного развития учащихся осуществляется методом наблюдения, анкетирования. По итогам первого полугодия и по итогам года заполняется «Диагностическая карта», в которой проставляется уровень усвоения программы каждым учащимся объединения.

### **Оценочные материалы**

При реализации данной ДООП используются следующие методики, позволяющие определить достижение обучающимися планируемых результатов: тесты, карточки практикума, методики анализа.

### **Методические материалы**

**Методы обучения:** (словесный, наглядный практический; объяснительно-иллюстративный, репродуктивный, частично-поисковый, исследовательский проблемный; игровой, дискуссионный, проектный и др.) и воспитания (убеждение, поощрение, упражнение, стимулирование, мотивация и др.);

**Формы организации образовательного процесса:** индивидуальная, индивидуально- групповая и групповая.

**Формы организации учебного занятия:** : беседа, защита проектов, игра, конкурс, конференция, круглый стол, лабораторное занятие, лекция, мастер-класс, наблюдение, олимпиада, практическое занятие, презентация, семинар, эксперимент, лабораторный практикум, мысленный эксперимент и др.;

**Образовательные педагогические технологии** технология индивидуализации обучения, технология группового обучения, технология развивающего обучения, технология проблемного обучения, технология дистанционного обучения, технология исследовательской деятельности, технология проектной деятельности, технология игровой деятельности, коммуникативная технология обучения, технология коллективной творческой деятельности, здоровьесберегающая технология и др.

### **Список литературы**

1. Васильев, А. Н. Python на примерах [Текст]:практ. курс /А. Н. Васильев  
- Наука и Техника, 2019 - 432 с.
2. Прохоренок, Н. А. Python 3: самое необходимое [Текст]:практ. курс / Н. А. Прохоренок, В. А. Дронов - БХВ-Петербург, 2019 - 608 с.
3. Гэддис, Т. Начинаем программировать на Python [Текст]: учебник/Т. Гэддис - БХВ-Петербург, 2019 - 768 с.
4. Седжвик, Р. Программирование на языке Python /Р. Седжвик, К. Уэйн, Р. Дондеро - Вильямс, 2017 - 736 с.
5. Харрисон, М. Как устроен Python. [Текст]:практ. курс / М. Харрисон - Питер, 2002 - 272 с.