


Муниципальное бюджетное общеобразовательное учреждение  
«Абалаковская средняя общеобразовательная школа №1»

<p style="text-align: center;"><b>«Согласовано»</b></p> <p>на заседании педагогического совета</p> <p>Протокол № <u>11</u> от <u>21 июня</u> 2023 г</p>	<p style="text-align: center;"><b>«Утверждаю»</b></p> <p style="text-align: center;"> Абалаковская СОШ №1</p> <p style="text-align: right;">_____ Е.И.Юшкевич</p> <p style="text-align: right;"><u>01-04-236</u> <u>сентябрь</u> 2023 г</p>
---	---

Дополнительная образовательная общеразвивающая программа  
техническое направление  
кружок  
«Программирование»

Возраст обучающихся 14-18 лет  
Срок реализации программы 3 года

Автор-составитель:

Тулупов Ю.В.

педагог дополнительного образования

с. Абалаково

2023-2024 уч. год

Программа «**Программирование**» направлена на формирование компетентности в области **ИКТ технологии** на материале **программирования на языке Python** в следующем формате:

- 3 года занятий по 2 часа в неделю
- индивидуально или группами по 2 человека в возрасте 14-18 лет
- занятия проводятся в компьютерном классе
- по итогу каждый научится создавать программы на языке Python
- мониторинг результатов осуществляется через наблюдение и контрольные задания (проекты), рейтинг на сайте АСМР.ru

### **Пояснительная записка**

Дополнительная общеобразовательная общеразвивающая программа «Программирование» имеет техническую направленность.

**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 и Python, который является достаточно эффективным и доступным инструментом достижения задач в области создания программных продуктов.

Pascal является знакомым для учеников из курса информатики, поэтому они могут продолжить изучать этот язык, а потом перейти на Python. Могут весь курс изучать Python. 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.

#### **Адресат программы:**

Программа адресована детям 8-11 класса

Содержание программы разработано с учетом принятия всех желающих, не имеющих противопоказаний по здоровью.

Дополнительная общеразвивающая программа «Программирование» рассчитана на детей 14–18 лет, проявляющих интерес к информационно-коммуникационным технологиям. Учащиеся объединения являются разными по возрасту и социальному статусу. Объединение комплектуется на основании заявлений законных представителей учащихся (самих учащихся с 14 лет). Группа численностью 10 человек формируется из школьников разного возраста на добровольной основе.

Программа составлена с учётом индивидуальных и возрастных особенностей учащихся. Психолого-педагогические особенности учащихся определяют и методы

индивидуальной работы педагога с каждым из них, темпы прохождения образовательного маршрута

**Объем программы** – 68 часов (2 часа в неделю).

Программа рассчитана на 3 года обучения.

**Форма обучения.** Очная

**Особенности организации образовательного процесса.**

Занятия проводятся со всем составом группы. Группа формируется из обучающихся 14-18-летнего возраста. Состав группы постоянный.

**Формы реализации образовательной программы:** традиционная.

**Режим занятий.**

Продолжительность одного академического часа - 45 мин. Общее количество часов в неделю – 2 часа. Занятие проводится 1 раз в неделю по 2 часа.

**Цель программы:** способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями и навыками программирования, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python или Pascal.

**Задачи программы:**

Образовательные (предметные):

- 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 Pascal и 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.

Рабочая программа внеурочной деятельности «Программирование» разработана на основе:

1. Закон № 273-ФЗ от 29.12.12 г. «Об образовании РФ»;
2. Санитарно-эпидемиологические правила и нормативы Сан-Пин 2.4.2.2821-10 «Санитарно-эпидемиологические требования к условиям и организации обучения в общеобразовательных учреждениях», зарегистрированные в Минюсте России 03 марта 2011 года, регистрационный номер 193;
3. Примерная рабочая программа курса внеурочной деятельности основы программирования на Python (для 7—9 классов образовательных организаций)
4. Авторская программа Босовой Л.Л. и др. «Начала программирование на языке Python.»;
5. Методические рекомендации по формированию учебных планов образовательных учреждений Воронежской области - региональных инновационных площадок по направлению «Введение федеральных государственных образовательных стандартов основного общего образования»;
6. Распоряжение Правительства Российской Федерации от 07 сентября 2010 г. №1570-р « Об утверждении плана действий по модернизации общего образования на 2011-2015 года»;
7. Приказ Министерства образования и науки Российской Федерации от 17 декабря 2010 г. № 1897 (зарегистрирован Минюстом России 1 февраля 2011 г., регистрационный № 19644) «Об утверждении федерального государственного образовательного стандарта основного общего образования»;
8. Разъяснения по отдельным вопросам применения федерального государственного образовательного стандарта основного общего образования ФГОС ООО от

### Календарно-тематический план.

№	Тема.	Дата	Количество часов
1	Общие сведения о языке		1
2	Практическая работа 1.1. Установка программы Python		1
3	Режимы работы		1
4	Практическая работа 1.2. Режимы работы с Python		1
5	Тест № 1. Знакомство с языком Python		1
6	Переменные		1
7	Практическая работа 2.1. Работа со справочной системой		1
8	Практическая работа 2.2. Переменные		1
9	Выражения		1
10	Практическая работа 2.3. Выражения		1
11	Ввод и вывод		1
12	Задачи на элементарные действия с числами		1
13	Практическая работа 2.5. Задачи элементарные действия с числами		1
14	Тест № 2. Выражения и операции.		1
15	Логические выражения и операторы		1
16	Практическая работа 3.1. Логические выражения		1
17	Условный оператор		1
18	Практическая работа 3.2. "Условный оператор"		1
19	Множественное ветвление		1
20	Практическая работа 3.3. Множественное ветвление		1
21	Реализация ветвления в языке Python		1

22	Практическая работа 3.4. "Условные операторы"		1
23	Самостоятельная работа № 1 по теме "Условные операторы".		1
24	Зачетная работа № 1. "Составление программ с ветвлением".		1
25	Тест № 3. "Условные операторы".		1
26	Оператор цикла с условием		1
27	Практическая работа 4.1. "Числа Фибоначчи"		1
28	Оператор цикла for		1
29	Практическая работа 4.2. Решение задачи с циклом for.		1
30	Вложенные циклы		1
31	Практическая работа 4.3. Реализация циклических алгоритмов		1
32	Случайные числа		1
33	Практическая работа 4.4. Случайные числа		1
34	Примеры решения задач с циклом		1
35	Практическая работа 4.5. Решение задач с циклом.		1
36	Самостоятельная работа № 2 "Составление программ с циклом"		1
37	Тест № 4. Циклы		1
38	Творческая работа № 1. "Циклы"		1
39	Создание функций		1
40	Практическая работа 5.1. Создание		1
41	Локальные переменные		1
42	Практическая работа 5.2. Локальные переменные		1
43	Примеры решения задач с использованием функций		1
44	Практическая работа 5.3. Решение задач с использованием функций		1
45	Самостоятельная работа № 3 по теме		1

	"Функции"		
46	Рекурсивные функции		1
47	Практическая работа 5.4. Рекурсивные функции		1
48	Тест № 5. Функции		1
49	Строки		1
50	Срезы строк		1
51	Практическая работа 6.1. Строки		1
52	Примеры решения задач со строками		1
53	Практическая работа 6.2. Решение задач со строками		1
54	Списки		1
55	Срезы списков		1
56	Практическая работа 7.1. Списки		1
57	Списки: примеры решения задач		1
58	Практическая работа 7.2. Решение задач со списками		1
59	Матрицы		1
60	Тест № 7. Списки		1
61	Кортежи		1
62	Введение в словари		1
63	Множества в языке Python		1
64	Стиль программирования и отладка программ		1
65	Стиль программирования		1
66	Отладка программ		1
67	Повторение		1
68			1

## Содержание курса

## Требования к уровню подготовки обучающихся

Знакомство с языком Python Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

**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.

Переменные и выражения. 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. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

**Учащиеся должны знать / понимать:**

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

**Учащиеся должны уметь:**

- пользоваться интерфейсом среды программирования Python;
- использовать команды редактора;
- организовывать ввод и вывод данных;
- записывать арифметические выражения.

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.

**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.

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.

**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;

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.

**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.

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.

**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.

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.

**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.

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.

### **Список литературы**

1. Златопольский Д. М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
2. Лутц Марк. Изучаем Python. – СПб: Симво-плюс, Том 1, 5-е издание, 2020.
3. Примерная рабочая программа курса внеурочной деятельности основы программирования на Python (для 7—9 классов образовательных организаций)
4. Босова, Л. Л. Информатика. 8-9 классы. Начала программирования на языке Python. Дополнительные главы к учебникам / Л. Л. Босова, Н. А. Аквилянов, И. О. Кочергин и др. — М. : БИНОМ. Лаборатория знаний, 2020 —96 с.