


МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
«СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 25  
ПОСЕЛКА НОВОУЛЬЯНОВСКОГО»

СОГЛАСОВАНО

Руководитель Центра

«Точка роста»

 Е.Н.Разуваева

«30» июня 2023г.



УТВЕРЖДАЮ

Директор МБОУ СОШ № 25

поселка Новоульяновского

А.С.Рогозин

Приказ № 161-ОД от «30» июня 2023г



Дополнительная общеобразовательная (общеразвивающая) программа  
технической направленности  
«Программирование на Python»

Возраст обучающихся: 11 – 17 лет  
Срок реализации: 2023-2024 учебный год

Составил: Рогозин Александр Сергеевич

п. Новоульяновский, 2023

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная (общеразвивающая) программа **«Программирование на Python»** имеет **техническую** направленность.

В современную жизнь человека все шире внедряются компьютеры и информационные технологии. Поэтому все большее значение приобретает компьютерная грамотность.

Курс изучения компьютерной грамотности состоит из двух разделов: пользовательского курса и программирования. Раздел «Программирование» в школьном курсе представлен языком программирования Pascal, а многим учащимся хочется познакомиться с другими языками программирования, самим попробовать разработать программы, которые можно использовать на уроках и во внеурочной деятельности. Данная Программа позволяет реализовать эти желания, так как уделяется большое внимание практической работе учащихся на компьютере, самостоятельной разработке ими программ для решения практических задач.

### **Актуальность программы**

С развитием современных информационных технологий сегодня любой учащийся под руководством опытного педагога может с лёгкостью научиться программировать.

Компьютеры и компьютерные системы – неотъемлемая часть жизни нашего общества. Научившись программировать, мы можем быть не только пользователями информационных технологий, но и активными их создателями.

Языки программирования можно сравнить с иностранными языками, овладеть ими может каждый. Учиться программировать очень интересно. Результат программирования очень часто виден сразу. Кроме того, создание компьютерных игр и обучающих программ способствует развитию логики и креативного мышления. Ещё одной значимой стороной обучения программированию является спрос на рынке труда на специалистов данного направления деятельности.

### **Педагогическая целесообразность Программы**

Педагогическая целесообразность программы заключается в привлечении учащихся к занятиям техническим творчеством, что способствует развитию логического мышления, творческих способностей и навыков решения задач программирования. Программирование мотивирует к занятиям в различных научных областях (физики, информатики, алгебры, геометрии и др.), развивает воображение и способствует ранней профориентации подростков. Для достижения поставленных задач занятия проводятся в формате «от простого к сложному». Учащиеся вспоминают свои знания по основам алгоритмизации и программирования на их основе, углубляя их, учатся составлять простые и сложные программы.

**Цель Программы:** обучение учащихся программированию посредством языка Python, развитие инженерного мышления, воспитание конкурентно способной личности.

### **Задачи Программы**

#### **Обучающие:**

- обучить языку программирования Python и созданию программ на его

основе;

- научить создавать прикладное программное обеспечение;
- расширять кругозор обучающихся в области программирования;
- научить дизайнерскому оформлению созданного ПО.

***Развивающие:***

- развивать память и внимание, познавательную и творческую активность;
- развивать творческие способности, эстетическое и эргономическое восприятие объектов труда;
- развивать логическое мышление.

***Воспитательные:***

- прививать интерес к активному творческому самовыражению, культуре труда;
- воспитывать упорство в достижении желаемого результата;
- воспитывать эстетический вкус;
- воспитывать чувство взаимопомощи, доверия, коллективизма.

**Отличительные особенности данной программы**

Основное количество часов отводится практическому написанию программ. Каждый обучающийся реализует индивидуальный проект в результате освоения программы. Продукт, полученный в результате освоения программы, имеет прикладной характер и может быть использован по необходимости.

**Возрастная категория обучающихся по программе** от 11 до 17 лет.

**Срок реализации** программы составляет 1 год. Общее количество часов в год составляет 68 часов.

**Формы и режим занятий** - групповые – для всей группы при изучении общих практических и теоретических вопросов. Наполняемость группы до 10 человек.

В ходе реализации программы применяется дифференцированный, индивидуальный подход к каждому обучающемуся.

Занятия проводятся 2 раза в неделю по 1 часу.

**Ожидаемые результаты и способы определения их результативности**

По итогам реализации программы, учащиеся

**будут знать:**

- принципы программирования на языке Python;
- основы дизайнерского оформления созданных программ.

По итогам реализации программы, учащиеся

**будут уметь:**

- производить чтение и запись программ на языке Python;
- запускать и отлаживать программу.

**Формы определения результативности обучения**

Результаты освоения программы отслеживаются по итогам опросов, выполнения практических заданий.

**Формы аттестации и оценочные материалы**

В процессе реализации программы предусмотрены следующие виды контроля: **входной контроль проводится** с целью определения уровня знаний учащихся

- *промежуточный контроль* проводится регулярно на занятиях с целью определения степени усвоения материала в форме опроса, решения задач и практических заданий;

- *итоговый контроль* – защита проекта.

### **Формы подведения итогов реализации программы**

Результаты обучения по программе выявляются по итогам проведения олимпиад, соревнований по программированию, защиты проекта.

## **ОСНОВНОЕ СОДЕРЖАНИЕ КУРСА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ ПО ИНФОРМАТИКЕ**

История языков программирования. Компиляция и интерпретация. Знакомство с Python и средой программирования IDLE.

Типы данных в программировании. Определение переменной. Ввод данных с клавиатуры. Первая программа на Python.

Строки как последовательности символов. Списки — изменяемые последовательности. Замена элементов в списке.

Логические выражения. Условный оператор. Инструкция if - elif - else. Проверка истинности if - elif - else. Цикл For. Цикл While.

Кортежи. Словари. Множества. Основные задачи обработки массивов. Введение в словари.

Сортировка выбором (поиск минимума и перестановка). Сортировка пузырьковым методом.

Функции в программировании. Параметры и аргументы функций. Локальные и глобальные переменные. Процедуры.

Решение задач посредством языка программирования Python: Алгоритм Евклида (нахождение наибольшего общего делителя); вычисление факториала на языке программирования Python; двоичный (бинарный) поиск элемента в массиве; перевод чисел из десятичной системы счисления в двоичную; решето Эратосфена - алгоритм определения простых чисел; сумма и произведение цифр числа; числа Фибоначчи (вычисление с помощью цикла while и рекурсии); тестирование простоты числа методом перебора делителей

**ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ С УКАЗАНИЕМ  
КОЛИЧЕСТВА ЧАСОВ, ОТВОДИМЫХ НА ОСВОЕНИЕ КАЖДОЙ  
ТЕМЫ КУРСА (68 часов)**

<b>№ раздела, темы</b>	<b>Раздел, тема</b>	<b>ИТОГО</b>
<b>1</b>	<b>Введение. Знакомство с Python</b>	
1.1.	Знакомство с IDLE Python	2
1.2.	Вычисления и переменные	2
1.3.	Первая программа на Python	2
<b>2</b>	<b>Построение программы на языке Python</b>	
2.1.	Строки и списки	2
2.2.	Синтаксис языка Python	2
2.3.	Операторы Python	2
<b>3.</b>	<b>Инструкция if - elif - else. Выбор подходящего варианта. Ветвление</b>	
3.1.	Инструкция if - elif – else	2
3.2.	Проверка истинности if - elif – else	4
<b>4.</b>	<b>Цикл в языке программирования Python</b>	
4.1	Цикл for	8
4.2	Цикл while	8
4.3	Операторы break и continue	2
<b>5.</b>	<b>Кортежи. Словари. Множества</b>	
5.1.	Кортежи	2
5.2.	Словари	2
5.3.	Множества	2
5.4.	Индексы и срезы	2
<b>6</b>	<b>Функции в программировании</b>	
6.3.	Параметры и аргументы функций	4
6.4.	Локальные и глобальные переменные	4
6.5.	Процедуры. Рекурсия	6
<b>7</b>	<b>Разработка и защита проекта</b>	
7.1.	Разработка и программирование собственного проекта	6
7.2.	Подведение итогов. Индивидуальный проект	4
	<b>Итого</b>	<b>68</b>

**Список литературы**

1. Доусен М. Програмуем на Python / М. Доусен - СПб.: Питер, 2016. - 416с.
2. Лутц М. Изучаем Python, 4 издание / М. Лутц - СПб.: Символ- Плюс, 2011. - 1280 с.
3. Любанович Б. Простой Python. Современный стиль программирования / Б. Любанович. - СПб.: Питер, 2016. - 480с.
4. Прохоренок Н.А., Дронов В.А. Python 3 и PyQt 5. Разработка приложений / Н.А. Прохоренок, В.А. Дронов - СПб.: «БХВ- Петербург», 2016. - 832с.
5. Саммерфильд М. Python на практике / М. Саммерфильд, пер. А.А.Слинкин – М.: ДМК-Пресс, 2014. - 338с.

**Цифровые образовательные ресурсы**

1. <https://pythontutor.ru/>
2. <https://www.python.org/>
3. <https://pythonworld.ru/samouchitel-python>
4. <https://pythoner.name/>

## УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ КУРСА

№ урока	Раздел, тема урока	Кол-во часов	Элементы содержания	Практическая часть	Примечание
<b>Раздел. Введение. Знакомство с Python – 3 часа</b>					
1	Знакомство с IDLE Python	2	Язык программирования Python. Знакомство и первая работа в среде разработки IDLE		
2	Вычисления и переменные	2	Обозначение переменных. Арифметические операторы.		
3	Первая программа	2	Создание и проверка первой программы «Hello world»	Практическая работа: «Hello world!»	
<b>Раздел. Построение программы на языке Python – 6 часов</b>					
4	Строки и списки	2	Знакомство со строками списками	Практическая работа. Арифметические операции со строками.	
5	Синтаксис языка Python	2	Работа со строками и списками. Функции и методы строк. Функции и методы списков	Практическая работа. Сообщение пользователю: написание письма с использованием строк	
6	Операторы Python	2	Операторы сравнения. Операторы присваивания. Логические операторы	Практическая работа. Написание программы расчета количества часов в году	
<b>Раздел. Инструкция if – elif – else. Выбор подходящего варианта. Ветвление – 6 часов</b>					
7	Инструкция if – elif – else	2	Инструкция if – elif – else. Использование инструкции if – elif – else		
8 – 9	Проверка истинности if – elif – else	4	Использование инструкции if – elif – else	Практическая работа. Написание программ с использованием инструкции if – elif – else для определения	

				результата	
<b>Раздел. Цикл в языке программирования Python – 18 часов</b>					
10 – 13	Цикл for	8	Цикл for. Требования к записи цикла. Работа цикла. Порядок выполнения программа	Практическая работа: «Решение задач с циклом for»	
14 – 17	Цикл while	8	Цикл while. Требования к записи цикла. Работа цикла. Порядок выполнения программа	Практическая работа: «Решение задач с циклом while»	
18	Операторы break и continue	2	Оператор прерывания цикла – break. Оператор перехода к следующему шагу цикла – continue. Синтаксис записи программы.		
<b>Раздел. Кортежи. Словари. Множества – 8 часов</b>					
19	Кортежи	2	Отличие кортежа от списка. Работа с кортежем. Операции с кортежем.	Практическая работа: «Работа с кортежем - tuple»	
20	Словари	2	Словари. Работа со словарями. Методы словарей	Практическая работа: «Работа со словарем - dict»	
21	Множества	2	Множества. Set и frozenset.	Практическая работа: «Работа со множествами»	
22	Индексы и срезы	2	Взятие элемента по индексу. Срезы		
<b>Раздел. Функции в программировании – 14 часов</b>					
23 – 24	Параметры и аргументы	4	Именные функции. Функция def.	Практическая работа:	

	функций		Синтаксис программы, содержащей функцию	«Применение и написание функции def»	
25 – 26	Локальные и глобальные переменные	4	Аргументы функций. Анонимные функции. Функция lambda. Область видимости		
27 – 29	Процедуры. Рекурсия	6	Понятие рекурсии. Аргументы произвольной длины. Ключевое слово return. Оправданные случаи использования рекурсии	Практическая работа: «Применение рекурсии. Нахождение факториала»	
<b>Раздел. Разработка и защита проекта – 10 часов</b>					
30-32	Разработка и программирование собственного проекта	6		Выбор вида и темы проекта. Составление технического задания. Программирование. Разработка технической документации и презентации проекта.	
33-34	Подведение итогов. Индивидуальный проект	4	Защита индивидуального проекта		