


МКОУ «Красноуральская средняя общеобразовательная школа»

Рассмотрено на заседании  
Методического совета

 Мutowкина Н.В.  
Протокол № 1 от 21.08. 2020

Утверждаю  
Директор МКОУ «Красноуральская  
СОШ»

 Шaлягина Г.В.  
Приказ № 83 от 28.08. 2020



**Рабочая программа  
курса «Программирование на Python»  
10 класс**

Автор-составитель программы:  
учитель информатики  
Мutowкина Н.В.

с.Красный Уралец 2020

## Планируемые результаты учебного курса « Программирование на Python»

### Личностные результаты:

- критическое отношение к информации и избирательность её восприятия;
- осмысление мотивов своих действий при выполнении заданий;
- развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера;
- развитие внимательности, настойчивости, целеустремлённости, умения преодолевать трудности;
- развитие самостоятельности суждений, независимости и нестандартности мышления;
- освоение социальных норм, правил поведения, ролей и форм социальной жизни в группах и сообществах;
- формирование коммуникативной компетентности в общении и сотрудничестве с другими обучающимися.

### Метапредметные результаты:

#### Регулятивные универсальные учебные действия:

- умение принимать и сохранять учебную задачу;
- умение планировать последовательность шагов алгоритма для достижения цели;
- умение ставить цель (создание творческой работы), планировать достижение этой цели;
- умение осуществлять итоговый и пошаговый контроль по результату;
- способность адекватно воспринимать оценку наставника и других обучающихся;
- умение различать способ и результат действия;
- умение вносить коррективы в действия в случае расхождения результата решения задачи на основе её оценки и учёта характера сделанных ошибок;
- умение в сотрудничестве ставить новые учебные задачи;
- способность проявлять познавательную инициативу в учебном сотрудничестве;
- умение осваивать способы решения проблем творческого характера в жизненных ситуациях;
- умение оценивать получающийся творческий продукт и соотносить его с изначальным замыслом, выполнять по необходимости коррекции либо продукта, либо замысла.

#### Познавательные универсальные учебные действия:

- умение осуществлять поиск информации в индивидуальных информационных архивах обучающегося, информационной среде образовательного учреждения, федеральных хранилищах информационных образовательных ресурсов;
- умение использовать средства информационных и коммуникационных технологий для решения коммуникативных, познавательных и творческих задач;
- умение ориентироваться в разнообразии способов решения задач;
- умение осуществлять анализ объектов с выделением существенных и несущественных признаков;
- умение проводить сравнение, классификацию по заданным критериям;
- умение строить логические рассуждения в форме связи простых суждений об объекте;
- умение устанавливать аналогии, причинно-следственные связи;

#### Коммуникативные универсальные учебные действия:

- умение аргументировать свою точку зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов;
- умение выслушивать собеседника и вести диалог;
- способность признавать возможность существования различных точек зрения и права каждого иметь свою;

- умение планировать учебное сотрудничество с наставником и другими обучающимися: определять цели, функции участников, способы взаимодействия;
- умение осуществлять постановку вопросов: инициативное сотрудничество в поиске и сборе информации;
- умение разрешать конфликты: выявление, идентификация проблемы, поиск и оценка альтернативных способов разрешения конфликта, принятие решения и его реализация;
- умение с достаточной полнотой и точностью выражать свои мысли в соответствии с задачами и условиями коммуникации;
- владение монологической и диалогической формами речи.

### **Предметные результаты**

#### **В результате изучения учебного курса Программирование на Python на уровне среднего общего образования**

##### **Выпускник на базовом уровне научится:**

- знать место языка Python среди языков программирования высокого уровня,
- знать особенности структуры программы, представленной на языке Python,
- иметь представление о модулях, входящих в состав среды Python?
- знать возможности и ограничения использования готовых модулей,
- иметь представление о величине, ее характеристиках,
- знать что такое операция, операнд и их характеристики,
- знать принципиальные отличия величин структурированных и не структурированных,
- иметь представление о таких структурах данных, как число, текст, кортеж, список, словарь,
- иметь представление о составе арифметического выражения;
- знать математические функции, входящие в Python,
- иметь представление о логических выражениях и входящих в них операндах, операциях и функциях,
- уметь записывать примеры арифметических и логических выражений всех атрибутов, которые могут в них входить,
- знать основные операторы языка Python, их синтаксис,
- иметь представление о процессе исполнения каждого из операторов,
- уметь разрабатывать программы обработки числовой и символьной информации,
- уметь разрабатывать программы (линейные, разветвляющиеся и с циклами),
- решать основные алгоритмические задачи, представленные в пояснительной записке, в среде Python.

##### **Выпускник на базовом уровне получит возможность:**

- использовать навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; использовать основные управляющие конструкции последовательного программирования и библиотеки прикладных программ; выполнять созданные программы;
- иметь представление о значении полноценных процедур и функций для структурно-ориентированного языка высокого уровня,
- знать правила описания процедур и функций в Python и построение вызова процедуры,
- знать принципиальные отличия между формальными, локальными и глобальными переменными,
- знать область действия описаний в процедурах,

- иметь представление о рекурсии, знать ее реализацию на Python,
- владеть основными приемами формирования процедуры и функции,
- знать, как с помощью Списков определять в программе тип «массив», «матрица»
- знать свойства данных типа «массив», «матрица»
- уметь воспроизводить алгоритмы сортировки массивов и матриц, поиска в упорядоченном массиве, распространять эти алгоритмы на сортировку и поиск в нечисловых массивах
- уметь читать и записывать текстовые файлы в заданном формате.

## Содержание учебного курса

### Тема 1. Знакомство с языком Python (2ч.)

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

### Тема 2. Переменные и выражения (4ч.)

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.

Практическая работа 2.1. Работа со справочной системой

Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.4. Задачи на элементарные действия с числами

### Тема 3. Условные предложения (5ч.)

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"

### Тема 4. Циклы (7ч.)

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"

### Тема 5. Функции (5ч.)

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"

### **Тема 6. Строки - последовательности символов (3ч.)**

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.

### **Тема 7. Сложные типы данных (7ч.)**

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.

### **Тема 8. Стиль программирования и отладка программ. (3ч.)**

Стиль программирования. Отладка программ. Зачет по курсу «Программирование на языке Python».

**Тематическое планирование с указанием количества часов, отводимых на освоение каждой темы.**

<b>№ урока</b>	<b>Тема</b>	<b>Кол-во часов</b>	<b>Электронный ресурс</b>
<b>Тема 1. Знакомство с языком Python</b>		<b>2</b>	
1	Общие сведения о языке Python. Установка Python на компьютер. <i>Практическая работа 1.1. Установка программы Python .</i>	1	
2	Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии. <i>Практическая работа 1.2. Режимы работы с Python</i>	1	
<b>Тема 2. Переменные и выражения</b>		<b>4</b>	
3	Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова. <i>Практическая работа 2.1. Работа со справочной системой</i> <i>Практическая работа 2.2. Переменные</i>	1	
4	Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция. <i>Практическая работа 2.3. Выражения</i>	1	
5	Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных.	1	
6	Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами. <i>Практическая работа 2.4. Задачи на элементарные действия с числами</i>	1	
<b>Тема 3. Условные предложения</b>		<b>5</b>	
7	Логический тип данных. Логические выражения и операторы. <i>Практическая работа 3. 1. Логические выражения</i>	1	
8	Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Сложные условные выражения (логические операции and, or, not). <i>Практическая работа 3.2. Условный оператор</i>	1	
9	Множественное ветвление. <i>Практическая работа 3.3. Множественное ветвление.</i>	1	
10	Реализация ветвления в языке Python. <i>Практическая работа 3.4. "Условные</i>	1	

	<i>операторы“</i>		
11	<b>Самостоятельная работа № 1. Решение задач по теме ” Условные операторы“</b>	1	
<b>Тема 4. Циклы</b>		<b>7</b>	
12	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. <i>Практическая работа 4.1. Числа Фибоначчи</i>	1	
13	Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. <i>Практическая работа 4.2. Решение задачи с циклом for</i>	1	
14	Вложенные циклы. Циклы в циклах. <i>Практическая работа 4.3. Реализация циклических алгоритмов</i>	1	
15	Случайные числа. Функция randrange. Функция random. <i>Практическая работа 4.4. Случайные числа</i>	1	
16	Примеры решения задач с циклом.	1	
17	<i>Практическая работа 4.5. Решение задач с циклом.</i>	1	
18	<b>Самостоятельная работа № 2 ”Составление программ с циклом“</b>	1	
<b>Тема 5. Функции</b>		<b>5</b>	
19	Создание функций. Параметры и аргументы. <i>Практическая работа 5.1. Создание функций</i>	1	
20	Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. <i>Практическая работа 5.2. Локальные переменные</i>	1	
21	Примеры решения задач с использованием функций. <i>Практическая работа 5.3. Решение задач с использованием функций</i>	1	
22	Рекурсивные функции. Вычисление факториала. Числа Фибоначчи. <i>Практическая работа 5.4. Рекурсивные функции</i>	1	
23	<b>Самостоятельная работа № 3 по теме ”Функции”</b>	1	
<b>Тема 6. Строки - последовательности символов</b>		<b>3</b>	
24	Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.	1	



25	Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). <i>Практическая работа 6.1. Строки</i>	1	
26	Примеры решения задач со строками. <i>Практическая работа 6.2. Решение задач со строками.</i>	1	
<b>Тема 7. Сложные типы данных</b>		<b>7</b>	
27	Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков.	1	
28	Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. <i>Практическая работа 7.1. Списки.</i>	1	
29	Списки: примеры решения задач. <i>Практическая работа 7.2. Решение задач со списками.</i>	1	
30	Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python.	1	
31	Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения	1	
32	Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.	1	
33	Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.	1	
<b>Тема 8. Стиль программирования и отладка программ.</b>		<b>2</b>	
34	Стиль программирования. Отладка программ.	1	
35	<i>Зачет по курсу «Программирование на языке Python».</i>	1	