

Администрация городского округа Сокольский

Центр образования цифрового и гуманитарного профилей «Точка Роста»  
филиала Муниципального бюджетного общеобразовательного учреждения  
Сокольской средней школы «Мурзинская средняя школа»

**ПРИНЯТО:**

на педсовете протокол № 1 от  
30.08.2023 г.

**УТВЕРЖДАЮ:**

Директор МБОУ Сокольской СШ:  
Чернова А. А./



**ПРОГРАММА ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ**

**«Программирование на языке Python»**

**Техническая направленность**

Возраст обучающихся: 12-15 лет.

Срок реализации 2 года.

Составитель:

Бекина И. В.,  
педагог дополнительного образования

Д. Мурзино 2023

## Пояснительная записка

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём зачастую не на пользовательском уровне, а на уровне начинающего программиста.

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне, на это выделяется недостаточное количество часов. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является формальное восприятие учащимися основ современного программирования и неумение применять полученные знания на практике.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Для обучения был выбран язык Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

Научившись программировать на языке Python, учащиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач.

Вместе с тем чистота и ясность его конструкций позволит учащимся потом с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

В основу курса «Основы программирования на языке Python» заложены принципы модульности и практической направленности, что обеспечит вариативность обучения. Содержание учебных модулей направлено на:

- детальное изучение алгоритмизации;
- реализацию межпредметных связей;
- организацию проектной и исследовательской деятельности обучающихся.

Важным аспектом программы является самостоятельная работа над заданиями: обучающиеся учатся решать задачи без помощи педагога. Для этого в содержании курса фигурируют задания, в которых:

- для решения задачи необходимо найти какую-то информацию в сети Интернет;
- может потребоваться устранение ошибки, которую не так просто быстро обнаружить;
- условие сформулировано недостаточно прозрачно и ученику необходимо самостоятельно формализовать его (или задать правильные вопросы преподавателю).

Курс «Основы программирования на языке Python» рассчитан на два года обучения

1 год 72 часа

2 год 72 часа

### **Новизна данной образовательной программы**

Программа курса:

- обеспечивает знакомство с фундаментальными понятиями алгоритмизации и программирования на доступном уровне;
- имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту ученика;
- охватывает как алгоритмическое направление, так и вопросы практического использования полученных знаний при решении задач из различных областей знаний;
- ориентирована на существующий парк вычислительной техники и дополнительные ограничения;
- допускает возможность варьирования в зависимости от уровня подготовки и интеллектуального уровня учащихся (как группового, так и индивидуального);
- предусматривает возможность индивидуальной работы с учащимися.

Практическая значимость курса заключается в том, что он способствует более успешному овладению знаниями и умениями по направлению «Программирование» через развитие самостоятельности обучающихся и оптимизацию средств и методов обучения.

Элементы программы курса могут быть рекомендованы для использования учителями информатики при проведении лабораторно-практических и практических занятий.

**Целью** курса является создание условий для изучения методов программирования на языке Python; рассмотрение различных парадигм программирования, предлагаемых этим языком (процедурная, функциональная, объектно-ориентированная); подготовка к использованию как языка программирования, так и методов программирования на Python в учебной и последующей профессиональной деятельности в различных предметных областях.

Настоящий курс направлен на решение следующих **задач**:

- формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- знакомство с принципами и методами функционального программирования;
- знакомство с принципами и методами объектно-ориентированного программирования;
- приобретение навыков работы в интегрированной среде разработки на языке Python;
- изучение конструкций языка программирования Python;
- знакомство с основными структурами данных и типовыми методами обработки этих структур;
- приобретение навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- приобретение навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач;
- развитие у обучающихся интереса к программированию;
- формирование самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- воспитание упорства в достижении результата;
- расширение кругозора обучающихся в области программирования для решения профессиональных задач (в том числе на основе системного подхода);
- способность грамотно строить коммуникацию, исходя из целей и ситуации.

По окончании курса ученик приобретает следующие компетенции:

- знание основ современных языков программирования;
- умение объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;
- умение искать и обрабатывать ошибки в коде;
- умение разбивать решение задачи на подзадачи; • способность писать грамотный, красивый код;
- способность анализировать как свой, так и чужой код;
- способность работать с информацией: находить, оценивать и использовать информацию из различных источников.

## **Общая характеристика курса «Основы программирования на языке Python»**

### **Основные разделы программы**

#### **Модуль 1. Введение в программирование**

Знакомство с виртуальной средой взаимодействия: регистрация, организация личного кабинета, поиск и выкладывание материалов. Знакомство с системой автоматизированной проверки задач.

Основные понятия программирования: исполнитель, система команд, алгоритм, программа, среда разработки, интерпретатор, код программы и редактор кода.

Ввод-вывод в программе, условный оператор, оператор цикла с предусловием. Простейшие программы с использованием условного оператора if, оператора циклов while и операторов ввода-вывода. Технология разработки программы.

На этом этапе обучающиеся разрабатывают первые алгоритмы и программы, а также анализируют, на какие функциональные блоки может быть разбита программа, и определяют работоспособность разработанной программы. В течение модуля ученики решают большое количество задач: от самых простых до сложных.

#### **Модуль 2. Базовые конструкции языка Python**

Понятие о языке высокого уровня Python. Структура программы, переменные и константы, работа с числовыми переменными, арифметические операторы в Python. Основные управляющие конструкции алгоритмов с ветвлением в Python.

Устройство циклов for. Основные управляющие конструкции циклического алгоритма в Python. Простейшие циклы и циклы с переменными.

Работа со списками, строками, множествами и кортежами в Python.

Понятие итератора.

Понятие подпрограммы, процедуры, функции. Функции в Python.

Решение задач.

#### **Модуль 3. Решение прикладных задач в Python**

Понятие ассоциативного массива. Словари в Python. Решение задач. Модули в Python. Подключение и использование модулей стандартной библиотеки. Модульный принцип компоновки программы. Работа с документацией в стандартной библиотеке. Понятие репозитория различных пакетов Python. Работа с внешними библиотеками Python и утилитой pip. Решение задач.

Самостоятельная работа: работа с конспектом, который описывает способы решения задач, разработка проекта по индивидуальному заданию, составление отчёта о выполнении индивидуальной или совместной работы.

Тематика самостоятельных работ:

- задачи, в которых по заданному алгоритму необходимо написать программу;
- задачи, в которых необходимо составить алгоритм решения и написать по нему программу;
- задачи, для решения которых необходимо найти некоторую информацию в

Интернете.

Выполненные самостоятельные работы загружаются в автоматизированную систему проверки задач Яндекс. Контест для дальнейшей оценки.

Зачётное занятие: выполнение итогового индивидуального задания по предложенной теме.

### **Формы организации учебных занятий**

Основной тип занятий — комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий курса выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Единицей учебного процесса является блок уроков (модуль). Каждый такой блок охватывает отдельную информационную технологию или её часть. Внутри блоков разбивка по времени изучения производится учителем самостоятельно, но с учётом рекомендованного календарно-тематического плана. С учётом регулярного повторения ранее изученных тем темп изучения отдельных разделов блока определяется субъективными и объективными факторами.

Каждая тема курса начинается с постановки задачи — характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. С этой целью учитель проводит демонстрацию презентации или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующих минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися в классе. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом ученики не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый ученик получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес ученика к предмету, активность и самостоятельность учащихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний учеников. Выполнение контрольных заданий способствует активизации учебно-познавательной деятельности и ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса.

### **Методы организации учебного процесса**

При организации занятий по курсу «Основы программирования на языке Python» для достижения поставленных целей и решения поставленных задач используются формы проведения занятий с активными методами обучения:

- занятие в форме проблемно-поисковой деятельности;
- занятие с использованием межпредметных связей;
- занятие в форме мозгового штурма;
- занятие в форме частично-поисковой деятельности.

### **Формы и методы контроля:**

- тестирование;

- устный опрос;
- самостоятельные и контрольные работы;
- участие в проектной деятельности.

#### **Общая характеристика учебного процесса:**

- при изучении курса используются практические и самостоятельные работы;
- курс обучения заканчивается написанием программы для решения одной из задач.

#### **Учебный план**

<b>№ п/п</b>	<b>Модуль</b>	<b>Количество часов</b>
1.	Введение в программирование	22
2.	Базовые конструкции в Python	61
3.	Решение прикладных задач в Python	61
	<b>Итого:</b>	<b>144</b>

#### **Личностные, метапредметные результаты освоения курса «Основы программирования на языке Python»**

##### **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, учебноисследовательской 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.

##### **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.

### **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.

### **Тематический план**

#### **Модуль 1. Введение в программирование**

144 часа

<b>№</b>	<b>Тема</b>	<b>Часы</b>
1.	Знакомство с системой Яндекс.Контест. Понятия кода, интерпретатора, программы. Интегрированные среды, исполнение кода. Простейшие программы с выводом на экран	1
2.	Решение задач по теме «Простейшие программы»	1
3.	Условный оператор. Переменные и арифметика	1
4.	Решение задач по теме «Условный оператор. Переменные и арифметика»	2
5.	Знакомство с циклом while	1
6.	Решение задач по теме «Цикл while»	1
7.	Отладчик	1
8.	Решение задач	2
9.	Знакомство с циклом for	2
10.	Решение задач по теме «Цикл for»	2
11.	Погружение в условия. True, False, Break, Continue	2
12.	Решение задач на тему «Погружение в условия»	2
13.	Резервное время на решение задач	1
14.	Решение задач	1
15.	Знакомство со списками	1
16.	Решение задач по теме «Списки»	1
	<b>Итого:</b>	<b>22</b>

#### **Модуль 2. Базовые конструкции в Python**

№	Тема	Часы
17.	Повторение. Дополнительные возможности цикла for. Кортежи. Сортировки	1
18.	Решение задач на тему «Цикл for»	2
19.	Знакомство со срезами и диапазонами. Равенство и совпадение объектов	2
20.	Решение задач по теме «Срезы и диапазоны»	2
21.	Списочные выражения. Методы split и join	2
22.	Решение задач на тему «Методы split и join»	2
23.	Другие методы списков и строк	2
24.	Решение задач на методы списков и строк	2
25.	Резервное время на решение задач	2
26.	Знакомство с функциями	2
27.	Решение задач на тему «Функции»	2
28.	Области видимости переменных	2
29.	Решение задач на тему «Области видимости переменных»	2
30.	Резервное время на решение задач. Подготовка к контрольной работе	2
31.	Контрольная работа по темам 1-го и 2-го модулей	2
32.	Анализ результатов контрольной работы	2
33.	Введение в компьютерную графику. Модуль TKInter	2
34.	Решение задач на тему «Модуль TKInter»	2
35.	Словари и множества	2
36.	Решение задач на тему «Словари множества»	2
37.	Обзор стандартной библиотеки Python	2
38.	Решение задач на тему «Стандартная библиотека Python»	2
39.	Дополнительные библиотеки Python	2
40.	Решение задач на тему «Дополнительные библиотеки Python»	2
41.	Резервное время на решение задач	2
42.	Введение в ООП	2
43.	Решение зада на тему «Введение в ООП»	2
44.	Резервное время на решение задач	2
45.	Итоговая контрольная работа по темам модулей 1,2 и 3	2
46.	Разбор контрольной работы	1
47.	Решение задач	3
	<b>Итого:</b>	<b>61</b>

### Модуль 3. Решение прикладных задач в Python

№	Тема	Часы
48.	Повторение материала 1-го и 2-го модулей	4
49.	Функции. Детальное изучение. Функциональная парадигма программирования	4
50.	Решение задач на тему «Функции и функциональная парадигма программирования»	4
51.	Словари и множества	4
52.	Решение задач на тему «Словари множества»	4
53.	Обзор стандартной библиотеки Python	4



54.	Решение задач на тему «Стандартная библиотека Python»	4
55.	Дополнительные библиотеки Python	4
56.	Решение задач на тему «Дополнительные библиотеки Python»	3
57.	Резервное время на решение задач	4
58.	Введение в ООП	4
59.	Решение зада на тему «Введение в ООП»	3
60.	Резервное время на решение задач	4
61.	Итоговая контрольная работа по темам модулей 1,2 и 3	4
62.	Разбор контрольной работы	3
63.	Решение задач	4
		<b>61</b>
	<b>Итого:</b>	<b>144</b>

### Календарно-тематическое планирование

144 академических часа в год

№ п/п	Название разделов, тем	Количество часов			Дата
		Всего	Теория	Практика	
<b>1</b>	<b>Модуль 1. Введение в программирование</b>	<b>22</b>	<b>8</b>	<b>14</b>	
1.1	Знакомство с системой Яндекс.Контест. Понятия кода, интерпретатора, программы.	1	1		
	Интегрированные среды, исполнение кода. Простейшие программы с выводом на экран				
1.2	Решение задач по теме «Простейшие программы»	1		1	
1.3	Условный оператор. Переменные и арифметика	1	1		
1.4	Решение задач по теме «Условный оператор. Переменные и арифметика»	2		2	
1.5	Знакомство с циклом while	1	1		
1.6	Решение задач по теме «Цикл while»	1		1	
1.7	Отладчик	1	1		
1.8	Решение задач	2		2	
1.9	Знакомство с циклом for	2	1	1	
1.10	Решение задач по теме «Цикл for»	2		2	
1.11	Погружение в условия. True, False, Break, Continue	2	1	1	
1.12	Решение задач на тему «Погружение в условия»	2		2	
1.13	Резервное время на решение задач	1	1		
1.14	Решение задач	1		1	
1.15	Знакомство со списками	1	1		
1.16	Решение задач по теме «Списки»	1		1	
<b>2</b>	<b>Базовые конструкции в Python</b>	<b>61</b>	<b>9</b>	<b>52</b>	
2.1	Повторение. Дополнительные возможности цикла for. Кортежи. Сортировки		1		
2.2	Решение задач на тему «Цикл for»				
2.3	Знакомство со срезами и диапазонами. Равенство и совпадение объектов		1		
2.4	Решение задач по теме «Срезы и диапазоны»				
2.5	Списочные выражения. Методы split и join		1		
2.6	Решение задач на тему «Методы split и join»				

2.7	Другие методы списков и строк		1		
2.8	Решение задач на методы списков и строк				
2.9	Резервное время на решение задач		1		
2.10	Знакомство с функциями		1		
2.11	Решение задач на тему «Функции»				
2.12	Области видимости переменных		1		
2.13	Решение задач на тему «Области видимости переменных»				
2.14	Резервное время на решение задач. Подготовка к контрольной работе		1		
2.15	Контрольная работа по темам 1-го и 2-го модулей				
2.16	Анализ результатов контрольной работы				
2.17	Введение в компьютерную графику. Модуль TKInter		1		
2.18	Решение задач на тему «Модуль TKInter»				
<b>3</b>	<b>Решение прикладных задач в Python</b>	<b>61</b>	<b>13</b>	<b>48</b>	
3.1	Повторение материала 1-го и 2-го модулей				
3.2	Функции. Детальное изучение. Функциональная парадигма программирования		1		
3.3	Решение задач на тему «Функции и функциональная парадигма программирования»		1		
3.4	Словари и множества		1		
3.5	Решение задач на тему «Словари множества»		1		
3.6	Обзор стандартной библиотеки Python		1		
3.7	Решение задач на тему «Стандартная библиотека Python»				
3.8	Дополнительные библиотеки Python		1		
3.9	Решение задач на тему «Дополнительные библиотеки Python»		1		
3.10	Резервное время на решение задач		1		
3.11	Введение в ООП		2		
3.12	Решение зада на тему «Введение в ООП»		1		
3.13	Резервное время на решение задач		2		
3.14	Итоговая контрольная работа по темам модулей 1,2 и 3				
3.15	Разбор контрольной работы				
3.16	Решение задач				
	<b>Итого:</b>	<b>144</b>			

### Список литературы

1. Улли Соммер «Программирование микроконтроллерных плат Arduino/Freduino».
2. Саймон Монк Мейкерство. Arduino и Raspberry Pi. Управление движением, светом и звуком.
3. Джереми Блум «Изучаем Arduino. Инструменты и методы технического волшебства».
4. Лидия Белиовская: Узнайте, как программировать на LabVIEW.
5. Джон Бейктал «Конструируем роботов на Arduino. Первые шаги».
6. Брайан Керниган, Деннис Ритчи «Язык программирования C».
7. Стенли Б. Липпман, Жози Лажойе, Барбара Э. Му «Язык программирования C++. Базовый курс».

8. Томас Кормен, Чарльз Лейзерсон, Рональд Ривест и Клиффорд Штайн «Алгоритмы: построение и анализ».
9. Герберт Шилдт С# 4.0. Полное руководство.
10. Издатель: Амперка Конспект хакера.
11. Ч. Платт Электроника для начинающих (2-е издание).
12. Мэтт Ричардсон Заводим BeagleBone.
13. Роберт К. Мартин Чистый код. Создание, анализ и рефакторинг. Библиотека программиста.
14. Кэти Сьерра и Берт Бейтс «Изучаем Java».
15. Дональд Кнут «Искусство программирования» т1.