МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Министерство образования и молодежной политики Свердловской области

г. Екатеринбург

МАОУ СОШ №208 с углубленным изучением отдельных предметов

УТВЕРЖДЕНО

Директор

_____ Чуб Т.С.

Приказ №53/7 от 28.08.2026

РАБОЧАЯ ПРОГРАММА

курса внеурочной деятельности

«Основы программирования на Python»

для обучающихся 7-8 классов

Екатеринбург, 2026

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Курс внеурочной деятельности «Основы программирования на Python» направлен на формирование у обучающихся 7-8 классов фундаментальных навыков алгоритмического мышления и основ программирования на современном языке высокого уровня Python. Курс построен по принципу "от простого к сложному" и охватывает базовые конструкции языка программирования, необходимые для решения широкого круга задач.

Оба года обучения начинаются с нулевого уровня, что позволяет включить в курс обучающихся без предварительной подготовки в программировании. В 7 классе изучаются базовые конструкции языка, в 8 классе курс включает в себя базовые конструкции языка, функции, списки и работу с файлами.

Курс носит практико-ориентированный характер: теоретический материал закрепляется решением задач различного уровня сложности, что способствует развитию навыков формализации задач, построения алгоритмов и их программной реализации.

ЦЕЛИ ИЗУЧЕНИЯ КУРСА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ

Целями изучения курса являются:

- формирование у обучающихся основ алгоритмического и логического мышления;
- развитие навыков формализованного описания задач и построения алгоритмов их решения;
- освоение базовых конструкций языка программирования Python;
- формирование умений разрабатывать, отлаживать и тестировать программы различной сложности;
- развитие компетенций в области обработки числовых, строковых и структурированных данных;
- воспитание ответственного отношения к результатам программной деятельности;
- подготовка обучающихся к участию в олимпиадах и конкурсах по программированию;

- формирование устойчивого интереса к дальнейшему изучению информационных технологий и программирования.

МЕСТО КУРСА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ В ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЕ

Курс внеурочной деятельности «Основы программирования на Python» является курсом по выбору обучающихся. Курс рассчитан на 2 года обучения (7-8 классы).

На изучение курса отводится **68 часов**: в 7 классе – 34 часа (1 час в неделю), в 8 классе – 34 часа (1 час в неделю).

Курс дополняет и расширяет знания, полученные на уроках информатики, и создаёт основу для углублённого изучения программирования в старших классах.

ФОРМЫ ПРОВЕДЕНИЯ ЗАНЯТИЙ КУРСА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ

Для реализации программы используются следующие формы проведения занятий:

- **лекции** (теоретический материал с демонстрацией);
- **практические работы** (решение задач на компьютере в среде программирования);
- **семинары** (разбор решений задач, обсуждение алгоритмов);
- **индивидуальные консультации** (помощь в отладке программ);
- **контрольные работы** (проверка усвоения материала);
- **проектная деятельность** (создание небольших программных проектов);
- **соревнования и конкурсы** (внутренние мини-олимпиады по программированию).

СОДЕРЖАНИЕ КУРСА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ]

7 КЛАСС

Типы данных

Знакомство с языком программирования Python. Установка и настройка среды разработки. Синтаксис языка. Переменные и присваивание. Основные типы данных: целые числа (int), вещественные числа (float), строки (str), логический тип (bool). Преобразование типов данных. Функции int(), float(), str(), bool(). Ввод данных с клавиатуры: функция input(). Вывод данных на экран: функция print(). Параметры функции print(): sep, end. Форматирование вывода. Простейшие линейные программы.

Целочисленная арифметика

Арифметические операции: сложение +, вычитание -, умножение *, деление /, целочисленное деление //, остаток от деления %, возведение в степень **. Приоритет операций. Операции присваивания с арифметикой: +=, -=, *=, /=, //=, %=, **=. Стандартные математические функции. Модуль math: функции abs(), min(), max(), round(). Решение задач на целочисленную арифметику: выделение цифр числа, проверка делимости, работа с разрядами числа.

Условная инструкция

Логический тип данных bool. Операции сравнения: ==, !=, <, >, <=, >=. Логические операции: and (логическое "И"), or (логическое "ИЛИ"), not (логическое "НЕ"). Условная инструкция if. Полная и неполная формы условной инструкции: if-else. Каскадная условная инструкция if-elif-else. Вложенные условные инструкции. Множественное присваивание. Решение задач на ветвления: определение четверти координатной плоскости, проверка свойств чисел, классификация данных.

Цикл for

Понятие цикла. Цикл с параметром for. Функция range(): range(stop), range(start, stop), range(start, stop, step). Перебор значений. Вложенные циклы. Цикл for для перебора символов строки. Цикл for для перебора чисел. Решение задач с использованием цикла for: вычисление сумм и произведений, обработка последовательностей, вывод таблиц и фигур.

Действительные (вещественные) числа

Представление вещественных чисел в компьютере. Особенности арифметики с плавающей точкой. Операции с вещественными числами. Округление: функция round(). Форматирование вывода вещественных чисел. Модуль math: функции sqrt(), ceil(), floor(), trunc(). Решение задач на работу с вещественными числами: вычисление расстояний, площадей, работа с геометрическими фигурами.

Строки

Строка как последовательность символов. Индексация символов в строке (прямая и обратная). Операции со строками: конкатенация (сложение) +, умножение на число *. Функция len() для определения длины строки. Срезы (splices): извлечение символа s[i], подстроки s[i:j], s[i:], s[:j], s[::step]. Методы строк: find(), rfind(), count(), replace(), split(), join(), upper(), lower(), strip(), startswith(), endswith(). Проверка вхождения подстроки: оператор in. Решение задач на обработку строк: подсчёт символов, замена подстрок, палиндромы, шифрование.

Цикл while

Цикл с предусловием while. Отличие цикла while от цикла for. Управление циклом: операторы break (прерывание цикла) и continue (переход к следующей итерации). Бесконечные циклы и способы их завершения. Обработка последовательностей неизвестной длины (обработка потока данных до признака конца ввода). Решение задач с использованием цикла while: алгоритм Евклида, проверка числа на простоту, обработка последовательностей.

8 КЛАСС

Всё то же, что в 7 классе и следующее содержание.

Функция, рекурсия

Определение функций с помощью ключевого слова def. Параметры функции: обязательные и необязательные, именованные параметры. Возвращаемое значение: инструкция return. Функции, не возвращающие значения. Локальные и глобальные переменные. Область видимости переменных. Вызов функций из других функций. Рекурсивные функции: определение, базовый случай, рекурсивный шаг. Примеры рекурсивных

функций: факториал, числа Фибоначчи, сумма цифр числа. Достоинства и недостатки рекурсии. Глубина рекурсии и ограничения. Решение задач с использованием функций и рекурсии.

Списки

Список (list) как изменяемая упорядоченная коллекция элементов. Создание списков. Индексация элементов списка (прямая и обратная). Срезы списков. Функция len() для списков. Операции со списками: конкатенация +, повторение *, проверка вхождения in. Методы списков: append(), extend(), insert(), remove(), pop(), clear(), index(), count(), sort(), reverse(), copy(). Вложенные списки. Перебор элементов списка: цикл for с перебором элементов, цикл for с перебором индексов, функция enumerate(). Ввод списка с клавиатуры. Вывод списка. Обработка списков: поиск элемента, подсчёт элементов, сумма элементов, нахождение минимума и максимума. Решение задач на обработку одномерных массивов (списков).

Списки - просто

Метод split(): разбиение строки на список подстрок. Необязательный параметр split(). Метод join(): объединение списка строк в одну строку. Обработка последовательностей чисел, введённых в одной строке. Задачи на обработку списков чисел: сортировка, фильтрация, преобразование. Генераторы списков (list comprehensions) - базовое знакомство. Решение комплексных задач на обработку списков.

Файлы

Понятие файлового ввода-вывода. Открытие файла: функция open(). Режимы открытия файла: чтение 'r', запись 'w', добавление 'a'. Файловые объекты. Чтение из файла: методы read(), readline(), readlines(). Построчное чтение файла с помощью цикла for. Запись в файл: метод write(). Закрытие файла: метод close(). Менеджер контекста with для автоматического закрытия файлов. Обработка данных из файлов: чтение чисел, обработка текстовых данных, запись результатов. Решение задач на работу с файлами: обработка данных, статистика, фильтрация.

ПЛАНИРУЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ

ЛИЧНОСТНЫЕ РЕЗУЛЬТАТЫ

В результате изучения курса внеурочной деятельности «Основы программирования на Python» у обучающихся будут сформированы следующие личностные результаты:

1. **патриотическое воспитание:**
    - ценностное отношение к достижениям отечественной школы программирования и информационных технологий;
    - понимание роли российских программистов в мировом IT-сообществе;
2. **духовно-нравственное воспитание:**
    - ориентация на моральные и этические нормы при создании программных продуктов;
    - ответственное отношение к авторским правам и интеллектуальной собственности;
    - неприятие плагиата и списывания кода;
3. **гражданское воспитание:**
    - готовность к сотрудничеству и совместной работе над программными проектами;
    - умение конструктивно обсуждать решения задач и код программ;
4. **ценности научного познания:**
    - интерес к изучению основ программирования и алгоритмизации;
    - готовность к самообразованию и самостоятельному решению задач;
    - стремление к оптимизации и улучшению своих программных решений;
    - формирование алгоритмического стиля мышления;
5. **трудовое воспитание:**
    - интерес к профессиям, связанным с программированием и разработкой программного обеспечения;
    - осознание практической значимости навыков программирования;
    - понимание необходимости постоянной актуализации знаний в IT-сфере;
6. **адаптация к изменяющимся условиям:**
    - готовность к освоению новых технологий и языков программирования;
    - умение быстро адаптироваться к новым средам разработки.

МЕТАПРЕДМЕТНЫЕ РЕЗУЛЬТАТЫ

Познавательные универсальные учебные действия:

- **Базовые логические действия:**
  - умение анализировать условие задачи и выделять ключевые требования;
  - способность разбивать сложную задачу на подзадачи;
  - умение сравнивать различные алгоритмические подходы к решению одной задачи;
  - построение логических рассуждений при отладке программ;
  - умение устанавливать причинно-следственные связи между кодом и результатом его выполнения;
- **Базовые исследовательские действия:**
  - формулирование гипотез о поведении программы при различных входных данных;
  - проверка гипотез экспериментальным путём (тестирование программы);
  - анализ эффективности различных алгоритмических решений;
- **Работа с информацией:**
  - умение работать с документацией языка программирования;
  - поиск и использование справочной информации при решении задач;
  - систематизация знаний о конструкциях языка программирования;
  - умение читать и понимать чужой код;

Коммуникативные универсальные учебные действия:

- **Общение:**
  - умение объяснять логику работы своей программы;
  - способность аргументировать выбор того или иного алгоритмического решения;
  - умение представлять результаты своей работы (демонстрация программы);
- **Совместная деятельность (сотрудничество):**
  - умение работать в паре или группе над общим программным проектом;
  - распределение задач между участниками команды;
  - совместная отладка и тестирование программ;
  - конструктивная критика и принятие замечаний по коду;

Регулятивные универсальные учебные действия:

- **Самоорганизация:**

    - умение планировать процесс решения задачи (от анализа условия до тестирования программы);
    - самостоятельный выбор подхода к решению задачи;
    - составление плана разработки программы;
- **Самоконтроль (рефлексия):**
    - умение находить и исправлять ошибки в своих программах (отладка);
    - тестирование программы на различных наборах входных данных;
    - оценка корректности и эффективности своего решения;
    - анализ причин ошибок и поиск способов их предотвращения;
    - рефлексия собственной деятельности по решению задач;
- **Эмоциональный интеллект:**
    - умение справляться с трудностями при решении сложных задач;
    - терпение и настойчивость при отладке программ;

ПРЕДМЕТНЫЕ РЕЗУЛЬТАТЫ

7 КЛАСС

К концу обучения в 7 классе у обучающегося будут сформированы следующие умения:

- создавать, запускать и отлаживать простейшие программы на языке Python;
- использовать переменные различных типов данных (int, float, str, bool);
- выполнять преобразование типов данных;
- организовывать ввод данных с клавиатуры и вывод результатов на экран;
- применять арифметические операции, включая целочисленное деление и остаток от деления;
- использовать условные инструкции (if, if-else, if-elif-else) для организации ветвлений;
- строить сложные логические выражения с использованием операций and, or, not;
- использовать цикл for с функцией range() для организации повторяющихся действий;
- перебирать символы строки с помощью цикла for;
- работать с вещественными числами и выполнять их округление;
- выполнять основные операции со строками (конкатенация, умножение, индексация);

- использовать срезы для извлечения подстрок;
- применять основные методы строк (find, count, replace, split и др.);
- использовать цикл while для организации циклов с условием;
- применять операторы break и continue для управления выполнением цикла;
- решать задачи на обработку числовых последовательностей и строк;
- оформлять код программы в соответствии с правилами хорошего стиля (PEP 8 - базовые принципы).

8 КЛАСС

К концу обучения в 8 классе у обучающегося будут сформированы следующие умения:

- уверенно применять все конструкции, изученные в 7 классе, для решения более сложных задач;
- определять и использовать функции с различными типами параметров (обязательные, необязательные, именованные);
- организовывать возврат значений из функций с помощью инструкции return;
- понимать область видимости переменных (локальные и глобальные);
- разрабатывать рекурсивные функции для решения задач (факториал, числа Фибоначчи и др.);
- анализировать преимущества и недостатки рекурсивных и итеративных решений;
- создавать и обрабатывать списки (одномерные массивы);
- использовать индексацию и срезы для доступа к элементам списка;
- применять методы списков для модификации и анализа их содержимого;
- перебирать элементы списка различными способами;
- использовать методы split() и join() для преобразования строк в списки и обратно;
- решать задачи на обработку числовых и строковых последовательностей с использованием списков;
- открывать файлы для чтения и записи с помощью функции open();
- использовать различные режимы открытия файлов;
- читать данные из файла различными способами (read, readline, readlines, построчно);
- записывать данные в файл;
- использовать менеджер контекста with для работы с файлами;

- решать задачи на обработку данных, хранящихся в файлах;
- комбинировать различные изученные конструкции для решения комплексных задач.

ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

7 КЛАСС

№ п/п	Наименование тем	Количество часов	Основное содержание
1	Типы данных	5	Основы программирования на Python. Установка среды разработки. Синтаксис Python. Переменные. Типы данных: int, float, str, bool. Присваивание. Функции input() и print(). Преобразование типов. Функции int(), float(), str(), bool()
2	Целочисленная арифметика	5	Операции +, -, *, /, **, приоритет операций. Операции // и %. Выделение цифр числа, подсчёт суммы цифр. Модуль math, функции abs(), min(), max(), round()
3	Условная инструкция	6	Тип bool. Операции ==, !=, <, >, <=, >=. Операции and, or, not. Таблицы истинности. Неполная форма if. Блоки кода. Полная форма условной инструкции. Каскадные условные инструкции
4	Цикл for	5	Цикл for, варианты range(). Перебор чисел, вычисление сумм и произведений. Цикл for для перебора символов. Цикл в цикле, вывод фигур и таблиц
5	Действительные (вещественные) числа	3	Арифметика с плавающей точкой, особенности. Функция round(), форматированный вывод. Решение геометрических задач
6	Строки	6	Строка как последовательность, индексация. Конкатенация, умножение, len(). Извлечение подстрок, шаги в срезах find(), count(), replace(), upper(), lower(). Разбиение и объединение строк
7	Цикл while	4	Цикл с предусловием, отличие от for. Управление выполнением цикла. Цикл while для неизвестного количества данных

ОБЩЕЕ КОЛИЧЕСТВО ЧАСОВ ПО ПРОГРАММЕ	34	

8 КЛАСС

№ п/п	Наименование тем	Количество часов	Основное содержание
1	Типы данных	3	Основы программирования на Python. Установка среды разработки. Синтаксис Python. Переменные. Типы данных: int, float, str, bool. Присваивание. Функции input() и print(). Преобразование типов. Функции int(), float(), str(), bool()
2	Целочисленная арифметика	3	Операции +, -, *, /, **, приоритет операций. Операции // и %. Выделение цифр числа, подсчёт суммы цифр. Модуль math, функции abs(), min(), max(), round()
3	Условная инструкция	3	Тип bool. Операции ==, !=, <, >, <=, >=. Операции and, or, not. Таблицы истинности. Неполная форма if. Блоки кода. Полная форма условной инструкции. Каскадные условные инструкции
4	Цикл for	3	Цикл for, варианты range(). Перебор чисел, вычисление сумм и произведений. Цикл for для перебора символов. Цикл в цикле, вывод фигур и таблиц
5	Действительные (вещественные) числа	2	Арифметика с плавающей точкой, особенности. Функция round(), форматированный вывод. Решение геометрических задач
6	Строки	3	Строка как последовательность, индексация. Конкатенация, умножение, len(). Извлечение подстрок, шаги в срезах find(), count(), replace(), upper(), lower(). Разбиение и объединение строк
7	Цикл while	2	Цикл с предусловием, отличие от for. Управление выполнением цикла. Цикл while для неизвестного количества данных

8	Функция, рекурсия	5	Ключевое слово def, параметры, аргументы. Обязательные и необязательные параметры, return. Область видимости переменных. Рекурсия, базовый случай, рекурсивный шаг
9	Списки	7	Списки, создание, индексация, len(). Срезы, методы append, pop, sort и др. Цикл for по элементам и индексам, enumerate(). Преобразование строк в списки и обратно. Сортировка, фильтрация, преобразование
10	Файлы	3	Функция open(), режимы, методы чтения. Запись в файл, with, закрытие файлов
ОБЩЕЕ КОЛИЧЕСТВО ЧАСОВ ПО ПРОГРАММЕ		34	

ПОУРОЧНОЕ ПЛАНИРОВАНИЕ
7 КЛАСС

№ п/п	Тема урока	Количество часов		
		Всего	Контрольные работы	Практические работы
1	Знакомство с Python. Первая программа "Hello, World!"	1		1
2	Переменные и типы данных (int, float, str, bool)	1		1
3	Ввод данных (input) и вывод данных (print)	1		1
4	Преобразование типов данных (int(), float(), str())	1		1
5	Практикум: решение задач на ввод-вывод	1		1
6	Арифметические операции (+, -, *, /, **)	1		1
7	Целочисленное деление (//) и остаток от деления (%)	1		1
8	Работа с цифрами числа (выделение, подсчёт суммы)	1		1
9	Стандартные математические функции (abs, min, max, round)	1		1
10	Практикум: задачи на целочисленную арифметику	1		1
11	Логический тип bool. Операции сравнения	1		1
12	Логические операции (and, or, not)	1		1
13	Условная инструкция if (неполная форма)	1		1
14	Инструкция if-else (полная форма)	1		1
15	Каскадная инструкция if-elif-else	1		1
16	Контрольная работа №1 по теме "Условная инструкция"	1	1	
17	Цикл for. Функция range()	1		1
18	Цикл for: вычисление сумм и произведений	1		1
19	Перебор символов строки с помощью цикла for	1		1
20	Вложенные циклы. Вывод фигур и таблиц	1		1

21	Практикум: задачи на цикл for	1		1
22	Операции с вещественными числами	1		1
23	Округление и форматирование вещественных чисел	1		1
24	Практикум: геометрические задачи с float	1		1
25	Строки: основы и индексация	1		1
26	Операции со строками (конкатенация, умножение)	1		1
27	Срезы строк (извлечение подстрок)	1		1
28	Методы строк (find, count, replace)	1		1
29	Методы split() и join()	1		1
30	Контрольная работа №2 по теме "Строки"	1	1	
31	Цикл while (цикл с предусловием)	1		1
32	Операторы break и continue	1		1
33	Обработка потока данных с помощью цикла while	1		1
34	Итоговая контрольная работа за курс 7 класса	1	1	
ОБЩЕЕ КОЛИЧЕСТВО ЧАСОВ ПО ПРОГРАММЕ		34	3	31

8 КЛАСС

№ п/п	Тема урока	Количество часов		
		Всего	Контрольные работы	Практические работы
1	Типы данных, преобразование типов	1		1
2	Продвинутое форматирование. F-строки	1		1
3	Практикум: комплексные задачи на ввод-вывод	1		1
4	Целочисленная арифметика	1		1
5	Олимпиадные задачи на арифметику (НОД, НОК)	1		1
6	Практикум: задачи на целочисленную арифметику	1		1
7	Условные инструкции (if, if-else, if-elif-else)	1		1
8	Сложные условия и тернарный оператор	1		1
9	Практикум: задачи на условные инструкции	1		1
10	Цикл for и функция range()	1		1
11	Вложенные циклы и их оптимизация	1		1
12	Практикум: задачи на цикл for	1		1
13	Вещественные числа и их обработка	1		1
14	Практикум: геометрические задачи с float	1		1
15	Строки, индексация и срезы	1		1
16	Методы строк и обработка текста	1		1
17	Практикум: комплексные задачи на строки	1		1
18	Цикл while, break и continue	1		1
19	Практикум: обработка потоков данных	1		1
20	Определение функций (def). Параметры и аргументы	1		1
21	Возврат значений (return). Именованные параметры	1		1

22	Локальные и глобальные переменные	1		1
23	Рекурсивные функции	1		1
24	Контрольная работа №1 по теме "Функции и рекурсия"	1	1	
25	Создание и индексация списков	1		1
26	Операции и методы списков	1		1
27	Перебор элементов списка. Функция enumerate()	1		1
28	Методы split() и join() для работы со списками	1		1
29	Обработка списков чисел (сортировка, фильтрация)	1		1
30	Контрольная работа №2 по теме "Списки"	1	1	
31	Открытие и чтение файлов. Режимы открытия	1		1
32	Запись в файлы. Менеджер контекста with	1		1
33	Практикум: обработка данных из файлов	1		1
34	Итоговая контрольная работа за курс 8 класса	1	1	0
ОБЩЕЕ КОЛИЧЕСТВО ЧАСОВ ПО ПРОГРАММЕ		0	0	0