

*Приложение №11
к учебному плану дополнительного образования,
утвержденному приказом директора
МАОУ СОШ №72 от 20.08.2025 №59*

**Программа дополнительного образования
«Основы программирования
на языке Python»
(8 класс)**

*Автор-составитель: Игнатьева Анастасия Владимировна,
учитель информатики*

1. Пояснительная записка

Программа предназначена для организации внеурочной деятельности учащихся 8-го класса и направлена на формирование устойчивых навыков алгоритмического мышления и практического программирования на языке Python. Курс создаёт базу для дальнейшего углублённого изучения информатики и подготовки к участию в олимпиадах и конкурсах по программированию.

Общая характеристика программы:

- уровень: базовый;
- направленность: техническая (информационно-технологическая);
- форма организации: кружок/факультатив;
- возраст обучающихся: 14–15 лет (8 класс);
- режим занятий: 1 академический час в неделю.
- общий объём: 34 часа в учебный год.

Нормативно-правовая основа:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- ФГОС основного общего образования (приказ Минобрнауки России от 17.12.2010 № 1897, с изменениями);
- Примерная основная образовательная программа основного общего образования (протокол ФУМО № 1/22 от 18.03.2022);
- Примерная программа воспитания (протокол ФУМО № 3/22 от 23.06.2022);
- локальные акты образовательной организации (учебный план, положение о внеурочной деятельности).

Актуальность программы обусловлена стремительным развитием цифровых технологий и возрастающей потребностью в квалифицированных ИТ-специалистах. Изучение Python как одного из самых востребованных и доступных языков программирования позволяет:

- сформировать у школьников системное представление о принципах программирования;
- развить логическое и алгоритмическое мышление;
- приобрести практические навыки решения задач с помощью кода;
- подготовить базу для участия в олимпиадах, хакатонах и проектной деятельности.

Отличительные особенности программы:

- акцент на практическую отработку навыков (не менее 60 % времени — практикумы и проекты);
- постепенное усложнение материала от базовых конструкций до мини-проектов;

- интеграция с школьным курсом информатики (повторение и углубление тем);
- использование современных сред;
- проектная деятельность как итоговая форма аттестации.

2. Содержание курса

Цель программы: сформировать у обучающихся устойчивые навыки программирования на языке Python, развить алгоритмическое мышление и подготовить к самостоятельному решению прикладных задач.

Задачи:

- **образовательные:**
 - познакомить с синтаксисом и базовыми конструкциями Python;
 - сформировать навыки написания, отладки и тестирования кода;
 - научить использовать встроенные функции и стандартные библиотеки;
 - освоить основы работы с данными (числа, строки, списки, словари);
 - развить умение составлять алгоритмы для решения типовых задач.
- **развивающие:**
 - развить логическое и абстрактное мышление;
 - сформировать способность анализировать задачу и разбивать её на подзадачи;
 - повысить уровень цифровой грамотности и ИКТ-компетенций;
 - стимулировать познавательную активность и интерес к программированию.
- **воспитательные:**
 - воспитать аккуратность и ответственность при работе с кодом;
 - сформировать культуру командной работы и взаимопомощи;
 - развить целеустремлённость и настойчивость в решении сложных задач;
 - привить этические нормы программирования (авторское право, открытый код).

Планируемые результаты

Предметные:

- знают синтаксис Python, типы данных, управляющие конструкции;
- умеют писать, отлаживать и тестировать программы средней сложности;
- владеют навыками работы со встроенными функциями и модулями;
- могут обрабатывать строки, числа, списки, словари и файлы;
- понимают принципы алгоритмизации и структурирования кода;
- способны разрабатывать мини-проекты (калькулятор, игра, анализатор данных).

Личностные:

- проявляют устойчивый интерес к программированию и IT-сфере;
- осознают ценность системного подхода и логического мышления;
- демонстрируют готовность к самообразованию и поиску решений;
- ответственно относятся к результатам своей работы;

- уважают интеллектуальную собственность и этические нормы разработки.

Метапредметные:

- умеют планировать деятельность, ставить цели и оценивать результаты;
- применяют анализ, синтез, сравнение и обобщение при решении задач;
- эффективно работают с информацией (поиск, отбор, структурирование);
- используют ИКТ как инструмент познания и творчества;
- взаимодействуют в команде, распределяют роли и презентуют результаты.

3. Учебный план

№	Тема	Количество часов	Форма аттестации
1	Введение в программирование. Среда разработки	1	Входная диагностика, опрос
2	Базовые конструкции Python	5	Практическая работа, тест
3	Управляющие конструкции (условия, циклы)	5	Практическая работа, мини-проект
4	Структуры данных: списки, строки	5	Практическая работа, кейс
5	Функции и модули	5	Практическая работа, защита функций
6	Работа с файлами и исключениями	4	Практическая работа, задача
7	Основы объектно-ориентированного программирования	4	Практическая работа, схема класса
8	Итоговый проект «Приложение на Python»	5	Защита проекта, презентация
Итого		34	

4. Содержание учебного плана

1. Введение в программирование. Среда разработки (1 ч)

- Теоретическая часть: история и сфера применения Python; установка и настройка среды (IDLE/VS Code); структура программы; комментарии; ввод-вывод данных.
- Практическая часть: первая программа «Hello, World!»; работа с консолью; форматирование вывода; решение простых задач на вывод текста.

2. Базовые конструкции Python (5 ч)

- Теоретическая часть: типы данных (int, float, str, bool); переменные и присваивание; арифметические и логические операции; приоритет операций.
- Практическая часть: вычисление выражений; преобразование типов; решение задач на арифметику и логику; отладка ошибок типов.

3. Управляющие конструкции (условия, циклы) (5 ч)

- Теоретическая часть: условный оператор if-elif-else; циклы for и while; операторы break и continue; вложенные конструкции.
- Практическая часть: задачи на ветвление (максимум из трёх чисел, проверка возраста); циклы (сумма чисел, таблица умножения); игры «Угадай число», «Камень-ножницы-бумага».

4. Структуры данных: списки, строки (5 ч)

- Теоретическая часть: строки и их методы (split, join, replace); списки и их свойства; индексация и срезы; методы списков (append, remove, sort).
- Практическая часть: обработка текста (подсчёт слов, замена символов); работа со списками (поиск элемента, сортировка); задачи на анализ данных.

5. Функции и модули (5 ч)

- Теоретическая часть: объявление и вызов функций; параметры и аргументы; область видимости; встроенные модули (math, random); создание собственных модулей.
- Практическая часть: написание функций для вычислений; рефакторинг кода; использование модулей для генерации случайных чисел; создание библиотеки утилит.

6. Работа с файлами и исключениями (4 ч)

- Теоретическая часть: открытие и закрытие файлов; режимы чтения/записи; обработка исключений (try-except); логирование ошибок.
- Практическая часть: чтение и запись текстовых файлов; анализ данных из файла; обработка ошибок ввода-вывода; создание логов программы.

7. Основы объектно-ориентированного программирования (4 ч)

- Теоретическая часть: классы и объекты; атрибуты и методы; конструктор **init**; инкапсуляция.
- Практическая часть: создание простых классов (Точка, Круг); наследование (Родитель-Ребёнок); проектирование мини-системы (библиотека, магазин).

8. Итоговый проект «Приложение на Python» (5 ч)

- Теоретическая часть: этапы проектной деятельности; выбор темы; планирование архитектуры; тестирование и отладка.
- Практическая часть: разработка приложения (калькулятор, дневник, игра, анализатор текста); документирование кода; подготовка презентации; публичная защита.

5. Планируемые результаты (итоговые)

По окончании программы учащийся:

- **знает:** синтаксис Python, типы данных, управляющие конструкции, принципы ООП;
- **умеет:** писать, отлаживать и тестировать программы; работать с файлами и модулями; проектировать алгоритмы;
- **владеет:** навыками структурирования кода, обработки ошибок, командной разработки;
- **может:** решать прикладные задачи (анализ данных, автоматизация, игры); презентовать и защищать проект.

Формы оценивания:

- текущие практические работы (оценка по критериям: корректность, читаемость, эффективность);
- мини-проекты и кейсы (защита, рецензирование);
- итоговый проект (презентация, код, документация);
- рефлексивные анкеты и самооценка;
- портфолио работ (репозиторий на GitHub/GitLab).

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП

Сертификат 256233904371995990837526139856067300059550830084

Владелец Тетерин Альберт Евгеньевич

Действителен с 05.11.2025 по 05.11.2026