

Рабочая программа
курса внеурочной деятельности
«Основы программирования»
для обучающихся 7-9 классов

Пояснительная записка

Классы: 7-9

Всего: 102 часа (1 час в неделю)

Направленность: техническая.

Актуальность: формирование алгоритмического мышления, подготовка к государственной итоговой аттестации (ОГЭ информатика), развитие навыков проектной деятельности.

Язык программирования: Python (рекомендуется как наиболее востребованный, имеющий низкий порог входа и мощные библиотеки). Возможна замена на C# или JavaScript в зависимости от материально-технической базы школы.

Формы занятий: лекция-демонстрация, практическая работа за компьютером, парное программирование, хакатоны, защита проектов.

Учебно-тематический план

№	Модуль / Тема	Кол-во часов
1	Введение в цифровую грамотность и среду разработки	6
1.1	Техника безопасности. Информация и данные. Системы счисления (двоичная, восьмеричная, шестнадцатеричная).	2
1.2	Установка среды разработки (IDE: VS Code, PyCharm Community или Thonny). Первая программа Hello, World!.	2
1.3	Отладка кода. Понятие синтаксической ошибки. Комментирование кода.	2
2	Основы алгоритмизации без компьютера	8
2.1	Алгоритм и его свойства. Способы записи алгоритмов (словесный, блок-схема).	2
2.2	Базовые структуры: следование, ветвление (if-else), цикл (while, for).	4
2.3	Вспомогательные алгоритмы и подпрограммы (функции). Рекурсия на интуитивном уровне.	2
3	Программирование на языке Python: Базовый уровень (7 класс)	18
3.1	Типы данных: целые числа, вещественные числа, строки. Операции над ними.	4
3.2	Ввод-вывод данных (input(), print()). Форматирование строк.	4
3.3	Условные операторы (if, elif, else) и логические выражения.	4
3.4	Циклы (for, while). Операторы управления циклом (break, continue).	4
3.5	Контрольная работа: решение задач из блока «Логика и алгоритмы» ОГЭ.	2
4	Структуры данных среднего уровня (8 класс)	20
4.1	Списки (list): создание, срезы, методы (append, pop, sort).	4
4.2	Строки как неизменяемые последовательности. Методы работы со строками.	4
4.3	Словари (dict) и множества (set). Когда и что использовать.	4
4.4	Функции (def). Локальные и глобальные переменные. Параметры по умолчанию.	4
4.5	Работа с текстовыми файлами (чтение, запись, дозапись). Обработка исключений (try-except).	4
5	Практикум решения прикладных задач (8 класс)	12
5.1	Основы анализа данных: чтение CSV-файлов (библиотека csv), базовая статистика.	4
5.2	Визуализация данных: построение графиков и диаграмм (библиотека matplotlib).	4

5.3	Мини-проект: «Анализ моего цифрового следа» (например, анализ времени в соцсетях или статистики сообщений).	4
6	Объектно-ориентированное программирование (9 класс)	16
6.1	Понятия класса и объекта. Атрибуты и методы. Конструктор <code>__init__</code> .	4
6.2	Инкапсуляция (приватные атрибуты). Наследование.	4
6.3	Практикум: создание иерархии классов (например, игра — персонаж — враг/друг).	4
6.4	Разбор сложных структур данных: стек, очередь, деревья на концептуальном уровне.	4
7	Проектная деятельность и геймдев (9 класс)	16
7.1	Этапы жизненного цикла ПО: от идеи до релиза. Постановка задачи проекта.	2
7.2	Основы разработки игр. Библиотека <code>pygame</code> (или веб-интерфейс на <code>tkinter/HTML+JS</code>). Игровой цикл.	6
7.3	Индивидуальная или групповая разработка финального проекта (чат-бот, телеграм-бот, простая 2D-игра, калькулятор с GUI).	6
7.4	Защита проектов. Рефлексия курса.	2
8	Резерв и промежуточная аттестация	6
	Итого:	102

Содержание модулей (ключевые компетенции)

- **Модули 1–2:** Учащиеся должны уметь читать и составлять блок-схемы, понимать разницу между компилятором и интерпретатором, владеть терминологией (бит, байт, IP-адрес).
- **Модуль 3:** Навык написания линейных и разветвляющихся программ. Умение трассировать код вручную (таблица значений переменных).
- **Модуль 4:** Владение коллекциями данных. Понимание сложности алгоритма на бытовом уровне (почему поиск в словаре быстрее, чем в списке).
- **Модуль 5:** Способность автоматизировать рутинную обработку информации. Критическое мышление при анализе числовых рядов.
- **Модуль 6:** Переход от процедурного мышления к объектному. Умение декомпозировать сложную задачу на взаимодействующие объекты.
- **Модуль 7:** Навык планирования работы, использования систем контроля версий на базовом уровне (GitHub/GitLab для сдачи работ) и командной работы.

Методическое обеспечение

1. **Техническое:** Компьютерный класс с выходом в интернет (от 1 ПК на человека). Для модуля визуализации и ООП рекомендуется установка Anaconda/Python 3.10+.
2. **Информационное:**
 - Российская платформа «Stepik» (курсы по Питону от Computer Science Center).
 - Сборники задач К.Ю. Полякова и Д.М. Ушакова (подготовка к ОГЭ/ЕГЭ).
 - Документация языка Python на русском языке.
 - Репозитории GitHub Classroom для организации выдачи заданий.
3. **Дидактическое:** Папка с раздаточным материалом (шпаргалки по методам списков и строк, таблица истинности, шаблоны блок-схем).

Система оценки результатов

Оценивание проводится безотметочно (зачет/незачет) или переводится в пятибалльную систему согласно локальному акту образовательной организации.

- **Текущий контроль:** выполнение практических заданий в IDE, активность в парном программировании.
- **Промежуточный:** автоматизированные тесты на онлайн-платформах, небольшие проверочные работы (code kata).

- **Итоговый:** защита индивидуального или группового проекта. Оценивается не только работоспособность кода, но и качество репозитория (наличие файла README.md, понятные имена переменных, комментарии).

Планируемые результаты

- **Личностные:** цифровая гигиена, настойчивость в поиске ошибок (дебаггинг), умение доводить проект до конца.
- **Метапредметные:** навыки формализации задачи, декомпозиция, поиск и проверка гипотез через написание кода.
- **Предметные:** уверенное владение синтаксисом Python, знание основных алгоритмов сортировки и поиска, опыт создания минимум одного законченного программного продукта.