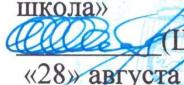


**Муниципальное автономное общеобразовательное учреждение
«Бабкинская средняя школа»
«Точка роста»**

СОГЛАСОВАНО:

Руководитель Центра образования
естественно- научной и технологической
направленностей «Точка роста»
С.А. Попова

«Утверждаю»

Директор МАОУ «Бабкинская средняя
школа»

(Ширяева Е.С.)
«28» августа 2025г.



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
технической направленности
«Программирование»**

Срок реализации: 1 год
Возраст учащихся: 11 - 17 лет

Малыгина Андрея Валерьевича
учителя информатики

пос. Кукуштан, 2025г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Нормативно-правовые документы:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказ Минпросвещения России от 09.11.2018 № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- ФГОС основного общего и среднего общего образования (применительно к возрастным группам);
- СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».

Актуальность.

В условиях цифровой трансформации общества навыки программирования становятся базовыми для успешной профессиональной самореализации. Программа отвечает запросу на раннюю профориентацию в ИТ-сфере и формирует у школьников алгоритмическое мышление, цифровую грамотность и проектную культуру.

Новизна.

Интеграция современных языков и сред программирования (Python, Scratch, визуальные конструкторы) с элементами проектной деятельности и соревновательного программирования.

Отличительные особенности:

- модульность (возможность выбора уровня сложности);
- практико-ориентированный подход (реальные проекты);
- дифференциация по возрасту и уровню подготовки.
- Уровень: стартовый/базовый (в зависимости от модуля).
- Направленность: техническая, информационно-технологическая.

Адресат: учащиеся 5–11 классов, интересующиеся информатикой и программированием (без предварительных навыков или с базовым уровнем).

Наполняемость: 10–15 человек в группе.

Форма обучения: очная, с возможностью использования онлайн-инструментов для домашней работы.

Объём и срок реализации: 34 часа (1 учебный год).

Режим занятий: 1 раз в неделю по 1 академическому часу (45 мин).

1.2. Цель и задачи программы

Цель: формирование у обучающихся базовых компетенций в области программирования, алгоритмического мышления и проектной деятельности.

Задачи:

образовательные:

- познакомить с основными понятиями алгоритмизации и программирования;
- освоить синтаксис и конструкции базового языка программирования (на выбор: Scratch для 5–7 кл., Python для 8–11 кл.);
- научить разрабатывать простые программы и проекты;

развивающие:

- развивать логическое и алгоритмическое мышление;
- формировать навыки анализа, структурирования и решения задач;
- стимулировать креативность и исследовательскую активность;

воспитательные:

- воспитывать целеустремлённость и ответственность за результат;
- формировать культуру командной работы;
- прививать интерес к ИТ-профессиям.

1.3. Планируемые результаты освоения программы

Личностные:

- осознание ценности ИТ-компетенций в современном мире;
- готовность к саморазвитию в сфере программирования.

Метапредметные:

- умение ставить цели и планировать действия;
- навыки работы с информацией (поиск, анализ, представление);
- способность к рефлексии и самооценке.

Предметные:

знают: основные алгоритмические конструкции, синтаксис выбранного языка, этапы разработки программы;

умеют: составлять алгоритмы, писать и отлаживать код, тестировать программы, презентовать проекты;

владеют: навыками работы в среде программирования, базовыми приёмами проектной деятельности.

1.4. Учебно-тематический план

№	Тема	Кол-во часов	Теория	Практика
1	Введение в программирование. Инструменты и среды	2	1	1
2	Алгоритмы и их свойства. Блок-схемы	4	2	2
3	Основы синтаксиса и типы данных	6	2	4
4	Условные операторы и циклы	6	2	4
5	Функции и процедуры	4	1	3
6	Работа с массивами и строками	4	1	3
7	Проектная деятельность: разработка мини-проекта	6	1	5

8	Презентация и защита проектов	2	–	2
	Итого	34	10	24

1.5. Содержание учебно-тематического плана

Введение в программирование. Инструменты и среды

История и перспективы программирования.

Выбор среды разработки (Scratch/Python).

Интерфейс, настройка, первые команды.

Алгоритмы и их свойства. Блок-схемы

Понятие алгоритма, свойства.

Линейные, разветвляющиеся, циклические алгоритмы.

Составление блок-схем.

Основы синтаксиса и типы данных

Переменные, константы.

Типы данных (числа, строки, логические).

Операции и выражения.

Условные операторы и циклы

Конструкция if-else.

Циклы for и while.

Вложенные условия и циклы.

Функции и процедуры

Определение и вызов функций.

Параметры и возвращаемые значения.

Рекурсия (базовый уровень).

Работа с массивами и строками

Одномерные массивы.

Основные операции со строками.

Поиск и обработка данных.

Проектная деятельность: разработка мини-проекта

Выбор темы (игра, калькулятор, чат-бот и т. п.).

Проектирование, кодирование, отладка.

Документирование кода.

Презентация и защита проектов

Подготовка презентации.

Демонстрация работы программы.

Ответы на вопросы, рефлексия.

2. Комплекс организационно-педагогических условий

2.1. Календарный план

Неделя	Тема занятия	Часы
1	Введение. Инструменты	1
2	Алгоритмы. Блок-схемы	1
3	Алгоритмы. Практика	1

Неделя	Тема занятия	Часы
4	Синтаксис. Типы данных	1
5	Синтаксис. Практика	1
6	Условные операторы	1
7	Условные операторы. Практика	1
8	Циклы	1
9	Циклы. Практика	1
10	Функции. Основы	1
11	Функции. Практика	1
12	Массивы. Основы	1
13	Массивы. Практика	1
14	Строки. Основы	1
15	Строки. Практика	1
16–21	Проектная деятельность	6
22	Презентация проектов	1
23	Итоговая рефлексия	1

Входной контроль: анкетирование, тестовое задание на логику.

Текущий контроль: выполнение практических заданий, мини-проекты, викторины.

Промежуточный контроль: защита мини-проектов по темам 3–6.

Итоговый контроль: презентация и защита итогового проекта, тестирование по теории.

2.3. Оценочные материалы

Критерии оценки проекта:

корректность работы программы (0–5 баллов);

оригинальность идеи (0–3 балла);

качество кода (комментарии, структура) (0–3 балла);

презентация (0–2 балла).

Тестовые задания по темам (примеры вопросов в приложении).

Лист наблюдений за активностью учащегося.

2.4. Методическое обеспечение программы

Методические рекомендации для педагога:

алгоритмы разбора типовых задач;

сценарии занятий с раздаточными материалами;

подборка олимпиадных задач для мотивации.

Дидактические материалы:

презентации по темам;

карточки с заданиями;

шаблоны кода для старта проектов.

Цифровые ресурсы:

онлайн-тренажёры (Code.org, Stepik);

видеоуроки по Python/Scratch.

2.5. Условия реализации программы

Материально-техническое обеспечение:

компьютерный класс (1 ПК на учащегося);

проектор, экран;

доступ в Интернет;

лицензии на ПО (Python, Scratch, IDE).