

Муниципальное автономное общеобразовательное учреждение
«Средняя общеобразовательная школа №26» г. Вологда

Принята на заседании
методического педагогического совета
от «28» августа 2025 г.
Протокол №20П

«УТВЕРЖДАЮ»

Директор МАОУ СОШ №26

Лукичева Л.В.

Приказ №

от



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА ТЕХНИЧЕСКОЙ
НАПРАВЛЕННОСТИ
«Основы программирования»**

Направленность: техническая
Возраст учащихся: 13-15 лет
Срок реализации: 1 год
Уровень программы: базовый

Автор-составитель:
Леухина Наталия Станиславовна,
педагог дополнительного образования,
учитель информатики

г. Вологда
2025 г.

ОГЛАВЛЕНИЕ

АННОТАЦИЯ.....	
I. Комплекс основных характеристик дополнительной общеобразовательной общеразвивающей программы:	3
1.1. Пояснительная записка	3
1.2 Цель и задачи программы	5
1.3. Учебный план, содержание программы.....	6
Содержание программы	6
1.4. Планируемые результаты	8
II. Комплекс организационно-педагогических условий:	9
2.1. Календарный учебный график.....	9
2.2 Условия реализации программы	11
2.3. Формы аттестации	12
2.4. Оценочные материалы	13
2.5. Методические материалы.....	14
2.6. Воспитательные компоненты	16
План воспитательной работы	16
Информационные ресурсы и литература Список использованной литературы:	17

1. Комплекс основных характеристик дополнительной общеобразовательной общеразвивающей программы «Основы программирования»

1.1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Направленность программы: техническая.

Программа направлена на формирование базовых знаний и практических навыков программирования, развитие логического и алгоритмического мышления, творческого подхода к решению технических задач.

Уровень сложности: Базовый.

Программа рассчитана на обучающихся с минимальными или отсутствующими навыками программирования.

Актуальность. Программа «Основы программирования» является актуальной и востребованной в современном образовательном пространстве, так как информационные технологии играют всё более значимую роль в жизни общества, экономике и профессиональной деятельности. Формирование базовых навыков программирования у школьников 13-15 лет способствует развитию логического, алгоритмического и критического мышления, что важно для успешного освоения точных наук и будущей профессиональной ориентации. Программа ориентирована на практико-ориентированное обучение: учащиеся не только осваивают теоретические основы, но и применяют знания при решении реальных задач, что повышает мотивацию, самостоятельность и интерес к информационным технологиям.

Новизна программы заключается в инновационном подходе к обучению, сочетании современных интерактивных сред программирования для начинающих, таких как Scratch и Python, с визуальными и текстовыми элементами. Программа внедряет игровые и проектные методы обучения, позволяя учащимся создавать мини-игры, интерактивные истории и простые приложения. Особенностью программы является интеграция практических навыков с элементами алгоритмического и логического мышления, что выделяет её среди традиционных теоретических курсов. Материал подаётся поэтапно: от освоения базовых понятий программирования до самостоятельной реализации небольшого проекта, обеспечивая системность и последовательность обучения.

Отличительные особенности программы включают:

- фокус на практическом освоении программирования через реальные задачи и проекты;
- использование интерактивных и визуальных сред для повышения вовлечённости учащихся;
- развитие логического, алгоритмического и критического мышления через интеграцию практических и теоретических элементов;

- поэтапное обучение с возможностью создания собственных проектов, что формирует навыки самостоятельной работы и проектного мышления.

Таким образом, программа сочетает актуальность, инновационность и уникальные педагогические подходы, обеспечивая эффективное и современное дополнительное образование в области программирования для школьников 8 класса.

Возраст обучающихся: 13-15 лет

Срок реализации программы: 1 учебный год

Продолжительность реализации программы в часах: 34 часа.

Форма обучения: очная.

Режим занятий: 1 раз в неделю, длительность занятия 40 минут.

Количество обучающихся в группе: от 6 до 16 человек.

Дополнительная общеразвивающая программа составлена в соответствии с нормативными документами:

- Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» (с последующими изменениями);

- Указ Президента Российской Федерации от 7 мая 2024 года № 309 «О национальных целях развития Российской Федерации на период до 2030 года и на перспективу до 2036 года»;

- Указ Президента Российской Федерации от 29 мая 2017 года № 240 «Об объявлении в Российской Федерации Десятилетия детства»;

- Распоряжение Правительства Российской Федерации от 23 января 2021 года № 122-р «Об утверждении плана основных мероприятий, проводимых в рамках Десятилетия детства, на период до 2027 года»;

- Указ Президента Российской Федерации от 9 ноября 2022 года № 809 «Об утверждении основ государственной политики в укреплении традиционных российских духовно-нравственных ценностей»;

- Концепция развития дополнительного образования детей до 2030 года (Распоряжение правительства Российской Федерации от 31 марта 2022 г. № 678-р);

- Целевая модель развития региональных систем дополнительного образования детей (приказ Министерства просвещения РФ от 3 сентября 2019 г. № 467, с изменениями);

- Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (Приказ Министерства просвещения РФ от 27 июля 2022 г. № 629);

- Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (Письмо Минобрнауки РФ от 18 ноября 2015 г. № 09-3242);

- Постановление Главного государственного санитарного врача РФ от 28 сентября 2020 г. № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания

и обучения, отдыха и оздоровления детей и молодежи».

1.2 Цель и задачи программы

Цель программы: формирование у обучающихся базовых знаний и навыков программирования, развитие логического и алгоритмического мышления, умения применять полученные знания для решения практических задач.

Задачи программы:

1. Ознакомить учеников с основными понятиями программирования: алгоритм, переменная, типы данных, условные операторы, циклы.

2. Научить применять базовые конструкции программирования на практике.

3. Развивать умение анализировать задачи и создавать алгоритмы их решения.

4. Сформировать навыки работы в средах визуального и текстового программирования.

5. Развивать творческое мышление через проектную и исследовательскую деятельность.

6. Воспитать интерес к информационным технологиям и инженерным дисциплинам.

1.3 Учебный план, содержание программы

№ п/п	Название раздела. Темы	Количество часов			Формы аттестации/контроля
		Всего	Теория	Практика	
1	Введение в программирование	2	1	1	Беседа, устный опрос
2	Алгоритмы и логическое мышление	4	1	3	Решение задач, мини-тест
3	Основы работы в среде программирования Python	6	2	4	Практическое задание проверка программ
4	Переменные и типы данных	3	1	2	Самостоятельная работа, выполнение заданий
5	Условные операторы (ветвления)	4	1	3	Практическая работа, тест
6	Циклы и повторяющиеся действия	4	1	3	Выполнение программы
7	Функции и модули	4	1	3	Практическая работа, защита мини-проекта
8	Проектная деятельность: разработка собственной и игры или программы	6	1	5	Защита проекта, презентация
9	Итоговое занятие. Подведение итогов. Рефлексия.	1	0	1	Итоговое обсуждение, самооценка результатов
	Итого	34	9	25	

Содержание учебного плана

Раздел 1. Введение в программирование (2 часа)

Теоретическая часть(1 час):

- Понятие программирования и его роль в современном мире.
- Обзор языков программирования и их классификация.
- Назначение программ, примеры использования в различных областях.
- Ознакомление со средой программирования Python.

Практическая часть(1 час):

- Знакомство с интерфейсом выбранной среды программирования.
- Выполнение простейших действий: запуск программы, ввод данных, вывод сообщений.

Форма контроля:

Беседа, устный опрос, наблюдение за работой учащихся.

Раздел 2. Алгоритмы и логическое мышление (4 часа)

Теоретическая часть(1 час):

- Понятие алгоритма и исполнителя.
- Свойства алгоритмов: дискретность, определённость, результативность.
- Формы записи алгоритмов: словесная, графическая (блок-схемы), программная.

Практическая часть(3 часа):

- Составление простых алгоритмов для бытовых и учебных задач.
- Создание блок-схем и реализация их в среде программирования.
- Решение задач на последовательность действий и логику.

Форма контроля:

Решение практических задач, мини-тест, обсуждение правильности составленных алгоритмов.

Раздел 3. Основы работы в среде программирования (6 часов)

Теоретическая часть(2 часа):

- Интерфейс среды Python.
- Команды, их структура и назначение.
- Основы синтаксиса: команды вывода, комментарии, выполнение программ.

Практическая часть(4 часа):

- Создание первой программы («Привет, мир!»).
- Работа с простыми командами ввода и вывода.
- Отладка и исправление ошибок.
- Выполнение индивидуальных заданий.

Форма контроля:

Практическая работа, проверка написанных программ, оценка самостоятельности выполнения.

Раздел 4. Переменные и типы данных (3 часа)

Теоретическая часть(1час):

- Понятие переменной.
- Типы данных: числа, строки, логические значения.
- Арифметические операции и выражения.

Практическая часть(2 часа):

- Создание программ с использованием переменных.
- Выполнение арифметических операций.
- Решение практических задач на вычисления.

Форма контроля:

Самостоятельная работа, выполнение заданий в среде программирования.

Раздел 5. Условные операторы (4 часа)

Теоретическая часть(1 час):

- Логические выражения и операторы сравнения.
- Условный оператор if, else, elif (в Python)
- Примеры применения ветвлений в программах.

Практическая часть(3 часа):

- Создание программ с условными переходами.
- Разработка мини-приложений с выбором действий (например, калькулятор, тест).
- Отладка и анализ работы программы при разных условиях.

Форма контроля:

Практическая работа, тест по теме, анализ выполненных программ.

Раздел 6. Циклы и повторяющиеся действия (4 часа)

Теоретическая часть(1 час):

- Понятие цикла, назначение повторяющихся действий.
- Циклы for и while (в Python).
- Вложенные циклы и примеры их применения.

Практическая часть(3 часа):

- Создание программ с использованием циклов.
- Решение задач на повторяющиеся вычисления (таблица умножения, подсчёт чисел).
- Написание программ с вложенными циклами.

Форма контроля:

Проверочная работа, выполнение программных заданий, устное объяснение принципов работы циклов.

Раздел 7. Функции и модули (4 часа)

Теоретическая часть(1 час):

- Понятие функции и процедуры.
- Аргументы и возвращаемые значения.
- Импорт и использование готовых модулей.

Практическая часть(3 часа):

- Создание собственных функций.
- Использование встроенных функций и библиотек.
- Разработка мини-программ с несколькими функциями.

Форма контроля:

Практическая работа, защита мини-проекта (программа с функциями).

Раздел 8. Проектная деятельность: создание собственной программы или игры (6 часов)

Теоретическая часть(1 час):

- Планирование и разработка проекта.
- Этапы проектной работы: идея, алгоритм, реализация, тестирование, презентация.

Практическая часть(5 часов):

- Разработка индивидуального или группового проекта (игра, интерактивная история, калькулятор и т.д.).
- Оформление и тестирование программы.
- Подготовка и проведение презентации проекта.

Форма контроля:

Защита проекта, самооценка, экспертная оценка преподавателя и сверстников.

Раздел 9. Итоговое занятие. Подведение итогов. Рефлексия (1 час)

Практическая часть (1 час): имитация олимпиады (2 задачи, 2 часа); разбор решений; анализ ошибок; рефлексия.

1.4 Планируемые результаты

По завершении освоения программы обучающиеся:

- понимают значение программирования в современном мире, знают его роль в различных сферах деятельности человека;
- знакомы с основными языками и средами программирования, умеют ориентироваться в интерфейсе выбранной среды (Python);
- знают основные понятия программирования: алгоритм, программа, команда, переменная, тип данных, функция, модуль;
- умеют составлять простые алгоритмы и реализовывать их в виде программ, используя различные формы записи (словесную, графическую, программную);
- владеют базовыми конструкциями программирования: последовательность действий, ветвление (условные операторы), повторение (циклы), функции;
- умеют работать с переменными и данными, выполнять вычисления, использовать логические и арифметические операции;

- способны отлаживать и корректировать программы, выявлять и устранять ошибки;
- применяют циклы и условные конструкции для решения практических задач, автоматизации вычислений и обработки информации;
- умеют создавать и использовать функции и модули, что позволяет структурировать код и повышать эффективность программ;
- владеют начальными навыками проектной деятельности — могут планировать, разрабатывать и презентовать собственный проект (программа, мини-игра, интерактивная история и др.);
- проявляют творческий подход и самостоятельность при выполнении практических заданий и проектной работы;
- развивают логическое, алгоритмическое и критическое мышление, умение анализировать задачу и выстраивать последовательность действий;
- осознают собственные достижения, умеют оценивать результаты своей работы и определять направления для дальнейшего развития в сфере информационных технологий.

Таким образом, программа обеспечивает формирование комплексной компетентности учащихся в области программирования, сочетая освоение технических навыков с развитием личностных, когнитивных и социальных качеств, что делает обучение максимально практико-ориентированным и современным.

2. Комплекс организационно-педагогических условий дополнительной общеобразовательной общеразвивающей программы «Основы программирования»

2.1 Календарный учебный график

1. Продолжительность учебного года: с 1 сентября 2025 по 31 мая 2026
2. Количество учебных недель – 34 недели
3. Продолжительность занятий для обучающихся – 40 минут.
4. Входная аттестация в начале изучения программы, промежуточная аттестация проводится в декабре, итоговый контроль – в мае.

№ занятия	Месяц	Тема занятия	Кол-во часов	Форма занятия
1	Сентябрь	Введение в программирование. Зачем нужно программировать	1	Лекция-беседа, презентация
2	Сентябрь	Обзор языков и сред программирования. Знакомство с интерфейсом среды	1	Практикум
3	Сентябрь	Что такое алгоритм. Основные свойства и формы записи	1	Объяснение нового материала, упражнения
4	Сентябрь	Блок-схемы и их использование. Составление алгоритмов	1	Практическая работа

5	Октябрь	Алгоритмы в повседневной жизни. Решение логических задач	1	Игровое занятие, практикум
6	Октябрь	Основы работы в среде программирования. Первая программа «Привет, мир!»	1	Практическое занятие
7	Октябрь	Команды вывода и ввода данных	1	Практикум
8	Октябрь	Ошибки в программе и их исправление	1	Практическая работа
9	Ноябрь	Переменные. Типы данных	1	Объяснение нового материала, практикум
10	Ноябрь	Арифметические операции. Примеры вычислений	1	Практическая работа
11	Ноябрь	Задачи с использованием переменных	1	Самостоятельная работа
12	Декабрь	Условные выражения. Понятие ветвления	1	Лекция-практикум
13	Декабрь	Условный оператор if (если — иначе)	1	Практическое занятие
14	Декабрь	Программы с выбором действий	1	Практическая работа
15	Январь	Решение задач с условными переходами	1	Самостоятельная работа
16	Январь	Понятие цикла. Назначение повторений	1	Объяснение нового материала
17	Январь	Цикл for (повторить N раз)	1	Практическое занятие
18	Январь	Цикл while (повторять пока)	1	Практическая работа
19	Февраль	Задачи на использование циклов	1	Самостоятельная работа
20	Февраль	Комбинирование циклов и условий	1	Практикум
21	Февраль	Функции: назначение и создание	1	Объяснение нового материала
22	Февраль	Вызов и использование функций	1	Практическое занятие
23	Март	Параметры функций и возвращаемые значения	1	Практикум
24	Март	Использование готовых модулей	1	Практическая работа
25	Март	Итоговое задание по функциям	1	Контрольная практическая работа
26	Апрель	Введение в проектную деятельность. Планирование проекта	1	Лекция-практикум
27	Апрель	Разработка идеи и алгоритма проекта	1	Проектная работа
28	Апрель	Реализация проекта (программирование)	1	Практическое занятие
29	Апрель	Отладка программы	1	Практическая работа
30	Май	Тестирование и улучшение проекта	1	Самостоятельная работа
31	Май	Подготовка презентации проекта	1	Проектная работа
32	Май	Защита проектов. Представление результатов	1	Презентация
33	Май	Обсуждение и самооценка результатов обучения	1	Беседа, рефлексия
34	Май	Итоговое занятие. Подведение итогов курса	1	Итоговое занятие, анкетирование

2.2 Условия реализации программы

Материально-техническое обеспечение программы. Для реализации дополнительной общеобразовательной программы «Основы программирования» необходимо наличие соответствующей материально-технической базы.

Характеристика помещения:

- Кабинет информатики или компьютерный класс, соответствующий санитарно-гигиеническим требованиям.
- Хорошее освещение, вентиляция, электропитание и доступ к сети Интернет.

Оборудование и технические средства:

- Персональные компьютеры или ноутбуки (1 на 1–2 учащихся).
- Программное обеспечение: операционная система, среда программирования Python, офисные приложения.
- Проектор или интерактивная доска для демонстрации учебного материала.
- Экран или мультимедийный монитор.
- Принтер, сканер (по возможности).
- Система электробезопасности и техника безопасности при работе с оборудованием.

Оргтехника и мебель:

- Рабочие столы и стулья, соответствующие возрастным требованиям.
- Стол преподавателя с компьютером.
- Доска (меловая, маркерная или интерактивная).
- Шкафы или стеллажи для хранения учебных материалов.

Учебно-методическое и раздаточное обеспечение:

- Учебники, методические пособия, инструкции по программированию.
- Карточки с алгоритмическими и логическими задачами.
- Тетради и рабочие листы для практических заданий.
- Презентационные материалы и электронные образовательные ресурсы.

Безопасность:

- Соблюдение требований охраны труда при работе с компьютером.
- Инструктаж по технике безопасности в начале курса.

Кадровое обеспечение программы. Педагогический состав:

- Педагог должен иметь высшее или среднее профессиональное образование в области информатики, математики, программирования или педагогики.
- Владеть современными образовательными технологиями и методиками преподавания информатики и программирования.

- Иметь навыки работы в средах программирования (Scratch, Python, Visual Studio Code).
- Уметь применять практико-ориентированные и проектные методы обучения.
- Иметь опыт организации проектной деятельности обучающихся.

2.3 Формы аттестации и контроля освоения программы

Для оценки усвоения учебного материала и достижения планируемых результатов программы используются следующие формы контроля:

1. Текущий контроль

Проводится на каждом занятии или после изучения конкретной темы:

- ✓ Устный опрос — проверка понимания теоретического материала, алгоритмов и структуры программ.
- ✓ Практические задания — выполнение отдельных программ, задач на использование переменных, циклов, условий и функций.
- ✓ Решение логических и алгоритмических задач — проверка развития алгоритмического и критического мышления.
- ✓ Самостоятельная работа — выполнение заданий без помощи преподавателя для оценки самостоятельности и усвоения навыков.
- ✓ Наблюдение за деятельностью обучающихся — оценка вовлечённости, аккуратности и правильности выполнения заданий.

2. Промежуточная аттестация

Проводится после изучения определённых модулей или блоков программы:

- ✓ Контрольные работы — выполнение практических заданий на проверку знаний и навыков (например, задачи с циклом, условиями, функциями).
- ✓ Мини-проекты или тестовые проекты — проверка умений создавать законченный фрагмент программы.
- ✓ Тесты (устные или электронные) — проверка усвоения теории и терминологии.

3. Итоговая аттестация

Проводится в конце программы:

- ✓ Проектная работа — разработка и защита собственного проекта (игра, интерактивная история, мини-программа), которая демонстрирует комплексное применение изученных знаний и умений.
- ✓ Презентация проекта — проверка навыков объяснения логики работы программы и аргументации решений.
- ✓ Итоговая практическая работа — создание небольшой программы с использованием всех изученных конструкций (переменные, циклы, условия, функции).

- ✓ Рефлексия и самооценка — оценка собственных достижений и определение направлений для дальнейшего развития.

2.4 Оценочные материалы

Для объективной оценки достижения планируемых результатов программы используются следующие оценочные материалы и инструменты:

1. Практические задания

- Выполнение отдельных программных заданий на освоение переменных, циклов, условных операторов и функций.
- Мини-проекты для закрепления навыков программирования (например, калькулятор, интерактивная история, простая игра).
- Решение алгоритмических и логических задач.

Форма оценки: проверка правильности выполнения заданий, соответствия кода требованиям задачи, корректность работы программы.

2. Тестовые материалы

- Онлайн или бумажные тесты для проверки усвоения теоретических понятий (термины, конструкции программирования, алгоритмы).
- Контрольные вопросы по пройденным темам.

Форма оценки: количество правильно выполненных заданий; качественный анализ ошибок для корректировки работы учащегося.

3. Проектные материалы

- Разработка индивидуального или группового проекта (мини-игра, интерактивная история, небольшое приложение).
- Презентация проекта с объяснением логики работы и алгоритмов.

Форма оценки:

- ✓ полнота и функциональность проекта;
- ✓ правильность и логичность алгоритмов;
- ✓ креативность и оригинальность решения;
- ✓ умение аргументированно представить и защитить проект.

4. Итоговые работы

- Итоговая практическая работа, включающая использование изученных конструкций (переменные, циклы, условия, функции).
- Рефлексивная анкета или самооценка по освоению навыков.

Форма оценки: соответствие работы учебным требованиям, степень самостоятельности и глубина усвоения материала.

Критерии оценивания

- Полнота выполнения задания (все требования выполнены).
- Корректность работы программы (отсутствие ошибок).
- Применение изученных конструкций и методов программирования.
- Самостоятельность и творческий подход к решению задач.

- Умение анализировать и улучшать код.
- Участие в защите проекта и аргументация своих решений.

2.5 Методическое обеспечение дополнительной общеобразовательной программы

Педагогический процесс по программе «Основы программирования» строится на принципах системности, последовательности и практико-ориентированного обучения, что позволяет учащимся постепенно овладевать базовыми знаниями и навыками программирования. Обучение направлено на развитие логического, алгоритмического и критического мышления, а также творческих и проектных компетенций.

Основные цели педагогического процесса:

- формирование базовых навыков программирования;
- развитие алгоритмического и логического мышления;
- формирование навыков самостоятельного поиска решений и проектной деятельности;
- развитие интереса к информационным технологиям и творческому использованию полученных знаний.

Виды занятий:

- 1) Теоретические занятия (лекции-беседы, демонстрации) — введение новых понятий, объяснение алгоритмов и принципов работы программ;
- 2) Практические занятия (практикумы, лабораторные работы) — закрепление изученного материала на практике, выполнение программных заданий;
- 3) Самостоятельная работа — выполнение задач без помощи преподавателя, подготовка к проектной деятельности;
- 4) Проектная деятельность — разработка индивидуальных и групповых проектов, включая создание игр, интерактивных историй и мини-программ;
- 5) Итоговые и рефлексивные занятия — защита проектов, презентации результатов, обсуждение достижений и трудностей.
- 6) Наиболее употребляемые формы и методы занятий:
- 7) Практическая работа и упражнения — выполнение программ, отработка алгоритмов, применение изученных конструкций;
- 8) Игровые методы — алгоритмические игры, образовательные конкурсы, викторины для закрепления знаний;
- 9) Проектный метод — планирование, реализация и защита собственного проекта;
- 10) Метод проблемного обучения — решение нестандартных задач, поиск оптимальных решений;
- 11) Демонстрация и визуализация — показ работы программ и алгоритмов, использование мультимедийных презентаций;

- 12) Обсуждение и рефлексия — анализ собственных действий, оценка результатов, обмен опытом между учащимися.

Для эффективного освоения программы используется широкий спектр методических материалов и продуктов, обеспечивающих разнообразие форм и методов обучения:

Учебно-методические пособия и инструкции:

- пособия по Python для начинающих;
- инструкции по созданию алгоритмов, блок-схем, мини-программ;
- методические рекомендации для практических и проектных заданий;
- пособия по основам логики, алгоритмики и структурированного мышления.

Раздаточные и дидактические материалы:

- карточки с задачами, логическими упражнениями, тестовые задания;
- рабочие листы и тетради для выполнения практических заданий;
- схемы алгоритмов и блок-схем;
- шаблоны для проектов и презентаций.

Электронные ресурсы и программное обеспечение:

- среды программирования Python;
- обучающие мультимедийные презентации и видеоуроки;
- электронные обучающие игры и тренажёры по программированию;
- платформы для создания интерактивных упражнений и тестов.

Методические разработки педагогических мероприятий:

- сценарии игровых занятий и викторин;
- конкурсы программирования и мини-проектов;
- конференции и выставки проектов учащихся;
- инструкции и методические рекомендации для проведения защиты проектов.

Все методические материалы направлены на развитие практических навыков программирования, логического мышления и проектной компетентности обучающихся. Их использование обеспечивает разнообразие форм обучения и поддержку индивидуального темпа освоения программы.

2.6 Блок «Воспитание»

Программа «Основы программирования» учитывает воспитательный компонент и формирует у обучающихся ценностные ориентиры, гражданскую ответственность, патриотизм и навыки социального взаимодействия через проектную, практическую и игровую деятельность.

Цели воспитательной работы:

- развитие ответственности, самостоятельности и дисциплинированности;
- формирование навыков командной работы и сотрудничества;
- воспитание интереса к техническому творчеству и инновационным технологиям;
- формирование уважения к трудовым, культурным и духовным ценностям российского общества;
- формирование гражданской идентичности и безопасного поведения в информационной среде.

Мероприятия по воспитанию в рамках программы

№	Месячный период	Мероприятие	Цель мероприятия	Форма проведения
1	Сентябрь	Вводная беседа «Зачем учиться программированию?»	Формирование мотивации, интереса к техническому творчеству	Лекция-беседа, дискуссия
2	Октябрь	Командная работа над алгоритмическими задачами	Развитие навыков сотрудничества, ответственности	Практическое занятие, групповая работа
3	Ноябрь	Викторина «Знаешь ли ты достижения отечественных IT-технологий?»	Формирование патриотизма, гражданской идентичности	Игровая форма, конкурсы
4	Декабрь	Обсуждение этики и безопасности в интернете	Воспитание этических норм, безопасного поведения	Дискуссия, кейс-обсуждение
5	Январь	Мини-проект «Создай игру вместе с другом»	Развитие коммуникации, совместного творчества	Практическая работа, проектная деятельность
6	Февраль	Конкурс «Лучший алгоритм»	Стимулирование творческого и логического мышления	Соревнование, презентация решений
7	Март	Участие в школьной/муниципальной конференции по проектам	Развитие навыков публичной презентации, гражданской ответственности	Презентация, защита проектов
8	Апрель	Рефлексия «Чему я научился и чему научился у команды?»	Формирование самосознания и оценки своих действий	Обсуждение, коллективная рефлексия
9	Май	Итоговая презентация проектов и награждение лучших	Подведение итогов, стимулирование ответственности и самостоятельности	Презентация, коллективная оценка

Информационные ресурсы и литература

1. Литература, используемая педагогом для разработки программы:

1. Бейкер, К. Программирование для начинающих: от алгоритма к коду. Москва: ДМК Пресс, 2020.
2. Джетти, М. Python. Базовый курс для школьников. Санкт-Петербург: Питер, 2019.
3. МакГрат, Т. Scratch: визуальное программирование для детей и подростков. Москва: БХВ-Петербург, 2018.
4. Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К. Алгоритмы: построение и анализ. Москва: Вильямс, 2021.
5. Митчелл, У. Программирование для детей и подростков. Москва: Бином, 2019.

2. Литература, рекомендуемая для обучающихся и их родителей:

1. Бейтс, С. Изучаем Python через игры и проекты. Москва: Диалектика, 2020.
2. Грин, П. Алгоритмы для детей: логика и мышление через программирование. Санкт-Петербург: Питер, 2019.
3. Ли, Дж. Scratch для школьников: практическое руководство. Москва: БХВ-Петербург, 2021.
4. Райт, Д. Программирование и логика для подростков. Москва: ДМК Пресс, 2018.
5. Фримен, Э., Робсон, Э. Основы программирования для детей и подростков. Санкт-Петербург: Питер, 2021.

3. Электронные ресурсы и информационные платформы:

- Официальный сайт Scratch: <https://scratch.mit.edu>
- Документация Python: <https://www.python.org/doc/>
- Платформа Stepik: <https://stepik.org> — курсы по Python и Scratch
- Образовательный портал «Фоксфорд»: <https://foxford.ru> — интерактивные уроки программирования