

**МУНИЦИПАЛЬНОЕ КАЗЕННОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
НОВОМИХАЙЛОВСКАЯ СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА**

Рассмотрена
на заседании
Педагогического совета
от «26» августа 2026 г.
Протокол №1

Утверждаю
Директор

МКОУ Новомихайловская СОШ


Т.Ю.Свиридова

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
технической направленности**

«Юные кодеры»

Возраст обучающихся: от 7 до 11 лет
Срок реализации программы: 1 год.

Автор-составитель программы:
Миненкова Екатерина Александровна
педагог дополнительного образования

Новомихайловка 2026

Содержание

Раздел 1. Пояснительная записка	3
1.1. Направленность, общая характеристика программы	3
1.2. Цель и задачи программы	5
1.3. Содержание программы	Ошибка! Закладка не определена.
1.4. Планируемые результаты	27
Раздел 2. Комплекс организационно–педагогических условий.	29
2.1. Календарный учебный график	29
2.2. Условия реализации программы	29
2.3. Формы аттестации и контроля	33
2.4. Методические материалы	35
2.5. Рабочая программа воспитания	36
СПИСОК ЛИТЕРАТУРЫ	39

1. Пояснительная записка

1.1. Направленность, общая характеристика программы.

Направленность программы

Программа имеет **техническую направленность** и предназначена для формирования у младших школьников первоначальных навыков алгоритмического и логического мышления через последовательное освоение двух визуальных сред программирования – **Scratch** и **Kodu Game Lab**.

Уровень освоения – **ознакомительный**, рассчитан на детей без предварительной подготовки.

Общая характеристика программы

Программа построена на принципе «обучение через игру» и предусматривает знакомство с двумя визуальными средами программирования.

В части Scratch дети осваивают блочный конструктор для создания анимаций, интерактивных историй и игр, знакомятся с базовыми алгоритмическими конструкциями: линейные алгоритмы, ветвления, циклы, переменные.

В части Kodu Game Lab дети работают в трёхмерном мире с физикой, программирование осуществляется через систему правил **WHEN-DO** (КОГДА-ДЕЛАЙ), что позволяет создавать игры разных жанров.

В основе программы – проектная деятельность. Обучение строится по принципу «от простого к сложному», формируя алгоритмический подход как универсальный инструмент решения задач.

Актуальность:

Актуальность программы обусловлена:

Требованиями ФГОС НОО – формирование основ логического и алгоритмического мышления является важнейшим метапредметным результатом;

Ранним вовлечением в цифровые технологии в условиях стремительной информатизации общества;

Доступностью визуальных сред – Scratch и Kodu позволяют преодолеть барьер сложности текстовых языков, сосредоточившись на логике;

Практической значимостью – освоенные навыки создают базу для дальнейшего изучения текстовых языков программирования;

Вариативностью – знакомство с двумя средами даёт представление о разных подходах к созданию алгоритмов.

Отличительные особенности:

- ✓ **Последовательное освоение двух сред:** Scratch (2D, блочное программирование) → Kodu Game Lab (3D, программирование правилами).
- ✓ **Kodu как среда с автономными персонажами** – объекты «живут» в мире с физикой (гравитация, столкновения), что даёт зрелищный результат даже от простой программы.
- ✓ **Опора на игровой опыт детей** – интерфейс Kodu ассоциируется с Minecraft, что повышает мотивацию.
- ✓ **Деятельностный подход** – ученик осваивает способ поиска решения, а не заучивает операции.
- ✓ **Спиральное усложнение** – от простых анимаций к многоуровневым играм с системой счёта и таймером.
- ✓ **Межпредметность** – создание викторин и тестов по математике, русскому языку, окружающему миру.
- ✓ **Парное и групповое программирование** – развитие коммуникативных навыков.

Новизна

Новизна программы заключается в следующем:

- ✓ **Обучение через создание игровых проектов в двух средах** – ученик выступает в роли разработчика: придумывает сюжет, проектирует мир, программирует правила, тестирует и презентует продукт.
- ✓ **«Сквозное» обучение** – навыки из Scratch становятся основой для перехода к Kodu, что создаёт прочную базу для последующего изучения текстовых языков (Python).
- ✓ **Развитие «вычислительного мышления»** как универсальной компетенции – умение анализировать проблему, разбивать на части, находить закономерности, строить алгоритмы.
- ✓ **Парное программирование и групповые проекты** – формирование навыков работы в команде, распределения ролей и ответственности за общий результат.

Ожидаемые результаты освоения программы

В результате освоения программы обучающиеся будут знать:

- ✓ основные понятия алгоритмики (условие, действие, цикл, событие, переменная);
- ✓ интерфейс и основные инструменты сред Scratch и Kodu Game Lab;
- ✓ структуру программы в Scratch (блочные конструкции) и в Kodu Game Lab (правило WHEN–DO);
- ✓ этапы разработки игрового проекта (от идеи до готового продукта);

В результате освоения программы обучающиеся будут уметь:

- ✓ создавать анимации и интерактивные истории в Scratch;
- ✓ создавать трёхмерные ландшафты и игровые миры в Kodu Game Lab;
- ✓ добавлять и настраивать персонажей и объекты в обеих средах;
- ✓ программировать поведение персонажей с использованием изученных алгоритмических конструкций;
- ✓ создавать игры с системой счёта, таймером, несколькими уровнями;
- ✓ тестировать и отлаживать свои проекты;
- ✓ представлять результаты работы;

приобретут навыки:

- ✓ алгоритмического мышления;
- ✓ проектной деятельности;
- ✓ работы в команде;
- ✓ поиска информации и самостоятельного решения проблем;
- ✓ публичной презентации проекта

Объем и сроки освоения программы.

Программа рассчитана на 1 год обучения с сентября по май. Программа реализуется в течение всего учебного года, включая каникулы. Объем программы составляет 72 часа.

Формы обучения — очная.

Особенности организации образовательного процесса.

Занятия групповые. Число детей в группе до 15 человек. Группы учащихся формируются из детей 7-11 лет. В группе могут быть дети одного возраста или разных возрастных категорий. Группы имеют постоянный состав.

Режим занятий.

Занятия проводятся по утвержденному расписанию 1 раз в неделю по 2 часа с учетом здоровьесберегающих технологий, с перерывом для отдыха не менее 15 минут. Один час - 45 минут.

1.2. Цель и задачи программы

Цель программы:

Формирование у младших школьников основ алгоритмического и логического мышления, творческих и проектных компетенций через практическое освоение визуальных сред программирования Scratch и Kodu Game Lab.

Задачи программы: Обучающие:

- ✓ познакомить с интерфейсом и базовыми инструментами сред Scratch и Kodu Game Lab;

- ✓ сформировать первоначальные понятия алгоритмики (алгоритм, условие, цикл, событие, переменная);
- ✓ научить создавать простые анимации, интерактивные истории и игры в Scratch;
- ✓ научить создавать трёхмерные игровые миры в Kodu Game Lab с использованием системы правил WHEN–DO;
- ✓ научить программировать поведение персонажей и объектов в обеих средах;
- ✓ научить создавать игровые проекты с системой счёта, таймером и несколькими уровнями;

Развивающие:

- ✓ развить алгоритмическое и логическое мышление;
- ✓ развить пространственное воображение и творческие способности;
- ✓ развить навыки самостоятельного поиска информации и решения проблем;
- ✓ развить познавательную активность и интерес к изучению новых технологий;
- ✓ развить умение планировать свою деятельность и доводить проект до завершения.

Воспитательные:

- ✓ воспитывать интерес к программированию и IT-творчеству;
- ✓ формировать навыки работы в команде (парное и групповое программирование);
- ✓ развивать коммуникативные умения и культуру общения;
- ✓ воспитывать ответственность за результат совместной работы;
- ✓ формировать навыки публичной презентации и защиты своих проектов;
- ✓ воспитывать информационную культуру и безопасное поведение в цифровой среде.

1.3. Содержание программы

№ п/п	Наименование базовых разделов, тем	Теория	Практика	Всего часов	Форма аттестации
Раздел 1. Введение в программирование					
1.1	Знакомство с курсом. Техника безопасности. Что такое программирование?	1	-	1	Устный опрос
1.2	Понятие алгоритма. Примеры алгоритмов в жизни. Исполнители.	1	-	1	Беседа, решение логических задач
1.3	Знакомство со средами Scratch и Kodu Game Lab. Обзор возможностей.	1	-	1	Наблюдение
1.4	Входное тестирование. Игровые задания на логику и алгоритмику.	-	1	1	Входное тестирование
Итого по разделу 1		3	1	4	
Раздел 2. Программирование в Scratch					
Тема 2.1. Знакомство со Scratch					
2.1.1	Интерфейс Scratch. Спрайты, сцены, блоки.	1	-	1	Устный опрос
2.1.2	Движение и координаты. Блоки «идти», «повернуть», «перейти».	1	-	1	Наблюдение
2.1.3	Внешность и звук. Смена костюмов, воспроизведение звуков.	1	-	1	Наблюдение
2.1.4	Практическая работа: создание простой	-	1	1	Практическая работа

	анимации «Танцующий спрайт».				
Тема 2.2. Основные алгоритмические конструкции					
2.2.1	Линейные алгоритмы. Последовательность команд.	1	-	1	Устный опрос
2.2.2	Циклы. Блок «повторить» и «всегда».	1	-	1	Устный опрос
2.2.3	Условные операторы. Блоки «если», «если-иначе».	1	-	1	Устный опрос
2.2.4	События. Управление с клавиатуры и мыши.	1	-	1	Наблюдение
2.2.5	Переменные. Создание и использование переменных.	1	-	1	Устный опрос
2.2.6	Операторы сравнения и логические операции.	1	-	1	Устный опрос
2.2.7-8	Практическая работа: игра «Поймай мяч» с системой счёта.	-	2	2	Практическая работа
Тема 2.3. Создание проектов в Scratch					
2.3.1	Создание игры-лабиринта. Управление с клавиатуры.	-	2	2	Практическая работа
2.3.2	Создание игры-платформера. Гравитация и прыжки.	-	2	2	Практическая работа
2.3.3	Создание викторины с вопросами и ответами.	-	2	2	Практическая работа
2.3.4	Работа с несколькими спрайтами. Взаимодействие объектов.	-	2	2	Практическая работа
2.3.5	Создание собственного проекта «Моя первая игра».	-	2	2	Практическая работа
Тема 2.4. Защита проектов в Scratch					

2.4.1	Тестирование и отладка проектов.	1	1	2	Наблюдение
2.4.2	Подготовка к защите: описание проекта, инструкция к игре.	-	2	2	Наблюдение
2.4.3	Презентация и защита проектов «Моя первая игра».	-	2	2	Защита проекта
Итого по разделу 2		10	18	28	
Раздел 3. Программирование в Kodu Game Lab					
Тема 3.1. Знакомство с Kodu Game Lab					
3.1.1	Интерфейс Kodu Game Lab. Управление камерой и объектами.	1	-	1	Устный опрос
3.1.2	Создание ландшафта. Инструменты редактирования мира.	1	-	1	Наблюдение
3.1.3	Добавление объектов и персонажей. Библиотека Kodu.	1	-	1	Устный опрос
3.1.4	Практическая работа: создание простого мира с объектами.	-	1	1	Практическая работа
Тема 3.2. Программирование правил WHEN–DO					
3.2.1	Концепция правил. Структура КОГДА–ДЕЛАЙ.	1	-	1	Устный опрос
3.2.2	Программирование движения персонажей.	1	-	1	Наблюдение
3.2.3	Управление с клавиатуры. Назначение клавиш.	1	-	1	Наблюдение
3.2.4	Программирование взаимодействия объектов.	1	-	1	Наблюдение
3.2.5	Система очков и счёт. Условия победы и	1	-	1	Устный опрос

	поражения.				
3.2.6	Использование таймера. Ограничение времени в игре.	1	-	1	Устный опрос
3.2.7	Программирование противников и препятствий.	1	-	1	Наблюдение
3.2.8	Создание уровней. Переход между уровнями.	1	-	1	Наблюдение
3.2.9-10	Практическая работа: игра «Собери все яблоки».	-	2	2	Практическая работа
Тема 3.3. Создание проектов в Kodu					
3.3.1	Создание гонки: трасса, автомобили, финиш.	-	2	2	Практическая работа
3.3.2	Создание платформера: прыжки, платформы, сбор монет.	-	2	2	Практическая работа
3.3.3	Создание игры-бродилки: лабиринт, враги, ключи.	-	2	2	Практическая работа
3.3.4	Создание симулятора: сбор ресурсов, управление NPC.	-	2	2	Практическая работа
3.3.5	Создание собственного проекта «Моя трёхмерная игра».	-	2	2	Практическая работа
Тема 3.4. Защита проектов в Kodu Game Lab					
3.4.1	Тестирование и отладка игровых проектов.	1	1	2	Наблюдение
3.4.2	Создание скриншотов и описание игрового процесса.	-	2	2	Наблюдение
3.4.3	Подготовка презентации своего проекта.	-	2	2	Наблюдение

3.4.4	Защита проектов «Моя трёхмерная игра».	-	2	2	Защита проекта
Итого по разделу 3		12	20	32	
Раздел 4. Итоговый проект					
4.1	Выбор темы итогового проекта (Scratch или Kodu). Постановка задач.	1	-	1	Беседа
4.2	Разработка сценария и проектирование.	1	-	1	Наблюдение
4.3-5	Создание итогового игрового проекта.	-	3	3	Наблюдение
4.6	Тестирование и отладка проекта.	-	1	1	Наблюдение
4.7	Подготовка к защите: презентация, описание, инструкция.	-	1	1	Наблюдение
4.8	Итоговая защита проектов. Награждение.	-	1	1	Итоговая защита проекта
Итого по разделу 4		2	6	8	
ВСЕГО		27	45	72	

Содержание тем и разделов

Раздел 1. Введение в программирование (4 часа)

Тема 1.1. Знакомство с курсом. Техника безопасности (1 час)

Теория:

- Знакомство с программой курса, его целями и задачами. Правила поведения в компьютерном классе. Инструктаж по технике безопасности и санитарно-гигиеническим нормам работы за компьютером. Правильная посадка, гимнастика для глаз и рук. Организация рабочего места.

Практика: не предусмотрена.

Форма контроля: устный опрос.

Тема 1.2. Понятие алгоритма. Примеры алгоритмов в жизни. Исполнители (1 час)

Теория:

- Что такое алгоритм? Примеры алгоритмов в повседневной жизни (режим дня, приготовление бутерброда, переход улицы). Кто такой исполнитель алгоритма? Команды и их последовательность. Понятие «компьютерная программа». Игровые задания на составление простых алгоритмов (словесных и графических).

Практика: не предусмотрена.

Форма контроля: беседа, решение логических задач.

Тема 1.3. Знакомство со средами Scratch и Kodu Game Lab. Обзор возможностей (1 час)

Теория:

- Что такое среда программирования? Обзор двух сред: Scratch (2D, блочное программирование, анимации, игры) и Kodu Game Lab (3D, создание игровых миров, программирование правилами). Демонстрация готовых проектов в обеих средах. Обсуждение того, что можно создать. Сравнение сред (сходства и различия). Выбор первой среды для изучения.

Практика: не предусмотрена.

Форма контроля: наблюдение, беседа.

Тема 1.4. Входное тестирование. Игровые задания на логику и алгоритмику (1 час)

Теория: не предусмотрена.

Практика:

- Выполнение входного тестирования для определения уровня сформированности логического мышления. Решение алгоритмических задач в игровой форме: лабиринты, головоломки, задания на последовательности и поиск закономерностей. Работа в рабочих листах.

Форма контроля: входное тестирование.

Раздел 2. Программирование в Scratch (28 часов)

Тема 2.1. Знакомство со Scratch (4 часа)

Тема 2.1.1. Интерфейс Scratch. Спрайты, сцены, блоки (1 час)

Теория:

- Запуск программы Scratch. Знакомство с интерфейсом: рабочая область, сцена, список спрайтов, список фонов, палитра блоков (категории: движение, внешность, звук, события, управление, сенсоры, переменные). Понятие «спрайт» и «сцена». Как выбирать, добавлять и удалять спрайты. Как менять фон.

Практика:

- Самостоятельное исследование интерфейса: запуск, просмотр меню, добавление и удаление спрайтов.

Форма контроля: устный опрос.

Тема 2.1.2. Движение и координаты. Блоки «идти», «повернуть», «перейти» (1 час)

Теория:

- Система координат в Scratch (ось X и Y). Понятие координат спрайта. Блоки движения: «идти», «повернуть направо/налево», «перейти в точку», «перейти к случайной позиции», «плыть». Углы поворота (направление). Управление направлением спрайта.

Практика:

- Создание простой программы: спрайт движется по сцене с использованием изученных блоков. Изменение координат вручную и с помощью блоков.

Форма контроля: наблюдение.

Тема 2.1.3. Внешность и звук. Смена костюмов, воспроизведение звуков (1 час)

Теория:

- Понятие «костюм» (costume) спрайта. Редактор костюмов. Блоки внешности: «сменить костюм», «следующий костюм», «изменить размер», «показать/скрыть», «сказать». Библиотека звуков: как добавить звук, как

воспроизводить звук (блоки «звук», «воспроизвести звук до конца»).
Создание эффектов движения с помощью смены костюмов.

Практика:

- Создание анимации: спрайт, который меняет костюмы (идёт, танцует) и воспроизводит звук.

Форма контроля: наблюдение.

Тема 2.1.4. Практическая работа: создание простой анимации «Танцующий спрайт» (1 час)

Теория: не предусмотрена.

Практика:

- Создание анимации «Танцующий спрайт»: спрайт с несколькими костюмами, последовательно сменяющимися; воспроизведение музыкального звука; движение по сцене в ритме музыки. Использование блоков движения, внешности и звука.

Форма контроля: практическая работа.

Тема 2.2. Основные алгоритмические конструкции (8 часов)

Тема 2.2.1. Линейные алгоритмы. Последовательность команд (1 час)

Теория:

- Что такое линейный алгоритм? Последовательное выполнение команд. Примеры линейных алгоритмов. Блоки «когда зелёный флаг нажат» (старт). Составление простых программ, где команды выполняются одна за другой. Важность порядка команд.

Практика:

- Создание программы, где спрайт выполняет последовательность действий: идёт вперёд, поворачивает, говорит, меняет размер, воспроизводит звук.

Форма контроля: устный опрос.

Тема 2.2.2. Циклы. Блок «повторить» и «всегда» (1 час)

Теория:

- Что такое цикл? Зачем нужны циклы? Виды циклов в Scratch: «повторить N раз» (цикл с счётчиком), «всегда» (бесконечный цикл). Примеры использования циклов: движение по кругу, постоянная смена костюмов. Экономия кода с помощью циклов.

Практика:

- Создание программ с циклами: спрайт движется по кругу 10 раз; спрайт бесконечно меняет костюмы.

Форма контроля: устный опрос.

Тема 2.2.3. Условные операторы. Блоки «если», «если-иначе» (1 час)

Теория:

- Что такое условие? Блок «если» и «если-иначе». Логические выражения (истина/ложь). Примеры условий в жизни: «если идёт дождь, то взять зонт, иначе не брать». Условия в Scratch: касание спрайта, касание цвета, нажатие клавиши. Как использовать условия для изменения поведения персонажа.

Практика:

- Создание программы: если спрайт касается края сцены, он отскакивает; если касается другого спрайта, меняет цвет.

Форма контроля: устный опрос.

Тема 2.2.4. События. Управление с клавиатуры и мыши (1 час)

Теория:

- Что такое событие? Блоки событий: «когда зелёный флаг нажат», «когда клавиша нажата», «когда спрайт нажат», «когда получено сообщение». Взаимодействие с пользователем через события. Управление спрайтом с клавиатуры (стрелки, пробел, буквы). Управление с помощью мыши.

Практика:

- Создание программы: спрайт двигается по нажатию стрелок на клавиатуре; изменение цвета при нажатии пробела.

Форма контроля: наблюдение.

Тема 2.2.5. Переменные. Создание и использование переменных (1 час)

Теория:

- Что такое переменная? Для чего нужны переменные в программе? Создание переменных в Scratch (сделать переменную). Типы данных (числа, строки). Присваивание значений переменным. Блоки «установить», «изменить», «показать переменную». Примеры использования переменных: счёт очков, таймер.

Практика:

- Создание переменной «очки». Программа: при касании спрайта (мяча) очки увеличиваются на 1.

Форма контроля: устный опрос.

Тема 2.2.6. Операторы сравнения и логические операции (1 час)

Теория:

- Блоки операторов: «больше», «меньше», «равно», «и», «или», «не». Использование операторов в условиях. Сравнение переменных с числами. Сложные условия (например, если очки больше 10 И время не закончилось). Практическое применение в играх.

Практика:

- Создание условий: если очки больше 5, то спрайт меняет цвет; если очки равны 10, то выводится сообщение «Победа!».

Форма контроля: устный опрос.

Тема 2.2.7–8. Практическая работа: игра «Поймай мяч» с системой счёта (2 часа)

Теория: не предусмотрена.

Практика:

- Создание игры «Поймай мяч»: на сцене появляются мячи в случайных местах; игрок управляет корзинкой (спрайтом) с помощью мыши или клавиатуры; при ловле мяча очки увеличиваются; если мяч пропущен, игра заканчивается или очки уменьшаются. Использование переменной для счёта, условий для проверки касания, циклов и событий.

Форма контроля: практическая работа.

Тема 2.3. Создание проектов в Scratch (10 часов)

Тема 2.3.1. Создание игры-лабиринта. Управление с клавиатуры (2 часа)

Теория:

- Проектирование игры-лабиринта: создание фона с лабиринтом (или рисование стен). Размещение спрайта-игрока и цели. Управление спрайтом с клавиатуры (стрелки). Программирование столкновений со стенами (возврат на старт). Условие победы (достижение цели). Добавление системы счёта и таймера.

Практика:

- Создание игры «Лабиринт»: рисование лабиринта, программирование движения игрока с клавиатуры, проверка столкновений со стенами, условие достижения финиша, добавление счёта (время прохождения).

Форма контроля: практическая работа.

Тема 2.3.2. Создание игры-платформера. Гравитация и прыжки (2 часа)

Теория:

- Что такое платформер? Основные элементы: персонаж, платформы, препятствия, гравитация. Реализация гравитации в Scratch: постоянное движение вниз. Программирование прыжков (условие касания земли). Сбор предметов (монет) и увеличение счёта. Условия поражения (падение в пропасть). Многоуровневые игры.

Практика:

- Создание игры-платформера: персонаж, который умеет прыгать, платформы, монеты для сбора, условие победы (собрать все монеты).

Форма контроля: практическая работа.

Тема 2.3.3. Создание викторины с вопросами и ответами (2 часа)

Теория:

- Создание интерактивной викторины. Использование нескольких спрайтов (вопросы, варианты ответов). Программирование выбора ответа (событие «когда спрайт нажат»). Проверка правильности ответа (условия, операторы сравнения). Переменные для подсчёта правильных ответов. Смена вопросов. Визуализация результата.

Практика:

- Создание викторины по теме (математика, русский язык, окружающий мир): 5–10 вопросов с вариантами ответов, подсчёт баллов, вывод результата.

Форма контроля: практическая работа.

Тема 2.3.4. Работа с несколькими спрайтами. Взаимодействие объектов (2 часа)

Теория:

- Взаимодействие нескольких спрайтов в одной игре. Передача сообщений между спрайтами (блоки «отправить сообщение», «когда получу сообщение»). Создание нескольких персонажей с разным поведением. Совместная работа спрайтов: враги, друзья, нейтральные объекты. Создание сложных игровых механик.

Практика:

- Создание игры с несколькими персонажами: игрок, враг (двигается к игроку), бонус (собирается игроком). Программирование взаимодействия:

если игрок касается врага – проигрыш, если касается бонуса – очки увеличиваются.

Форма контроля: практическая работа.

Тема 2.3.5. Создание собственного проекта «Моя первая игра» (2 часа)

Теория:

- Этапы разработки проекта: идея → сценарий → проектирование → программирование → тестирование → презентация. Самостоятельный выбор темы игры (лабиринт, платформер, викторина, стрелялка, бродилка и т.д.). Проектирование игрового процесса. Планирование этапов работы.

Практика:

- Самостоятельная работа над собственным проектом. Создание игры по выбранной теме с использованием всех изученных конструкций: переменные, условия, циклы, события, взаимодействие спрайтов.

Форма контроля: практическая работа (подготовка проекта к защите).

Тема 2.4. Защита проектов в Scratch (6 часов)

Тема 2.4.1. Тестирование и отладка проектов (2 часа)

Теория:

- Что такое тестирование программы? Почему важно находить и исправлять ошибки. Основные типы ошибок: синтаксические, логические. Как тестировать игру: проверка всех возможных действий пользователя. Отладка программ в Scratch: последовательная проверка блоков, проверка условий, устранение багов.

Практика:

- Тестирование и отладка собственных проектов: поиск и исправление ошибок, проверка работы всех механик, улучшение игрового процесса.

Форма контроля: наблюдение.

Тема 2.4.2. Подготовка к защите: описание проекта, инструкция к игре (2 часа)

Теория:

- Как подготовить проект к защите. Структура описания проекта: название, цель, сюжет/правила игры, управление, использованные алгоритмы. Создание инструкции для игрока. Оформление проекта (название, помощь, экран победы/поражения). Подготовка устного выступления.

Практика:

- Составление описания и инструкции к своему проекту. Подготовка устного выступления (2–3 минуты) с объяснением замысла и использованных алгоритмов.

Форма контроля: наблюдение.

Тема 2.4.3. Презентация и защита проектов «Моя первая игра» (2 часа)

Теория: не предусмотрена.

Практика:

- Публичная презентация и защита собственных проектов: демонстрация игры, объяснение сюжета, правил, использованных алгоритмов и конструкций. Ответы на вопросы одноклассников. Обсуждение работ, взаимооценка, рефлексия.

Форма контроля: защита проекта.

Раздел 3. Программирование в Kodu Game Lab (32 часа)**Тема 3.1. Знакомство с Kodu Game Lab (4 часа)****Тема 3.1.1. Интерфейс Kodu Game Lab. Управление камерой и объектами (1 час)****Теория:**

- Запуск Kodu Game Lab. Знакомство с интерфейсом: игровой мир, панель инструментов (Main Menu, Run, Edit, Object Tool, Path Tool, Terrain Tool, Paint Tool). Режимы: режим редактирования и режим игры. Управление камерой: вращение, масштабирование, перемещение по миру (клавиши WASD, мышь). Понятие «объект» в Kodu.

Практика:

- Исследование интерфейса: запуск программы, работа с камерой, добавление объектов на сцену, просмотр библиотеки объектов.

Форма контроля: устный опрос.

Тема 3.1.2. Создание ландшафта. Инструменты редактирования мира (1 час)**Теория:**

- Инструменты Terrain Tool и Paint Tool. Создание ландшафта: поднятие и опускание земли, создание холмов, гор, впадин. Изменение цвета и

текстуры земли (трава, вода, песок, снег, грязь). Создание воды (озера, реки). Практическое применение: создание основы для будущей игры.

Практика:

- Создание ландшафта по замыслу: холмистая местность с озером, горы, долина.

Форма контроля: наблюдение.

Тема 3.1.3. Добавление объектов и персонажей. Библиотека Kodu (1 час)

Теория:

- Библиотека объектов Kodu: список доступных персонажей (Kodu, ракета, автомобиль, мяч, бабочка, рыба, робот и т.д.). Добавление объектов на сцену с помощью Object Tool. Размещение объектов в определённых местах. Масштабирование и поворот объектов. Выбор персонажа и его начальное положение.

Практика:

- Добавление на созданный ландшафт различных объектов и персонажей. Настройка их расположения и масштаба.

Форма контроля: устный опрос.

Тема 3.1.4. Практическая работа: создание простого мира с объектами (1 час)

Теория: не предусмотрена.

Практика:

- Создание полноценного трёхмерного мира: ландшафт с травой, водой, холмами; размещение деревьев, камней, персонажей. Запуск мира в режиме игры для просмотра результата.

Форма контроля: практическая работа.

Тема 3.2. Программирование правил WHEN–DO (10 часов)

Тема 3.2.1. Концепция правил. Структура КОГДА–ДЕЛАЙ (1 час)

Теория:

- Что такое правило в Kodu? Структура правила: КОГДА (условие) → ДЕЛАЙ (действие). Как программирование в Kodu отличается от Scratch (правила вместо блоков). Создание первого правила: «КОГДА нажать стрелку → ДЕЛАЙ двигаться вперёд». Интерфейс создания правил (Page Up/Page Down). Понятие условий и действий.

Практика:

- Создание первого правила для персонажа Kodu: движение по нажатию клавиш стрелок.

Форма контроля: устный опрос.

Тема 3.2.2. Программирование движения персонажей (1 час)

Теория:

- Управление движением персонажа: правила для разных направлений (вперёд, назад, поворот). Настройка скорости движения. Езда на транспорте: управление автомобилем. Разница между управлением Kodu и другими объектами.

Практика:

- Создание правил для управления Kodu с клавиатуры: стрелки вверх/вниз/влево/вправо. Тестирование движения.

Форма контроля: наблюдение.

Тема 3.2.3. Управление с клавиатуры. Назначение клавиш (1 час)

Теория:

- Назначение различных клавиш для управления (пробел, буквы A, S, D, W). Условие «когда клавиша нажата». Комбинированные действия: нажатие клавиши + дополнительное условие. Примеры: пробел для прыжка, пробел для стрельбы.

Практика:

- Создание правил: нажатие пробела заставляет персонажа прыгать или стрелять. Использование разных клавиш для разных действий.

Форма контроля: наблюдение.

Тема 3.2.4. Программирование взаимодействия объектов (1 час)

Теория:

- Условия взаимодействия объектов: «касается» (включает в себя касание любого объекта или конкретного типа), «столкнулся с», «видит». Действия при взаимодействии: «уничтожить», «изменить счёт», «увеличить скорость», «переместить». Создание игровых механик на основе взаимодействия.

Практика:

- Создание правил: когда Kodu касается мяча, мяч исчезает и добавляются очки. Когда Kodu касается врага, игра заканчивается или теряется жизнь.

Форма контроля: наблюдение.

Тема 3.2.5. Система очков и счёт. Условия победы и поражения (1 час)

Теория:

- Создание переменной «очки» (Score). Изменение очков при определённых действиях (добавить 1 очко, изменить на...). Условия победы: когда очки достигли определённого значения. Условия поражения: когда очки стали равны 0, потеряны все жизни, закончилось время. Визуализация очков и сообщений о победе/поражении.

Практика:

- Создание игры со сбором предметов: за каждый собранный объект добавляется очко. Когда собрано все предметы – появляется сообщение «Победа!». Создание сообщений на экране.

Форма контроля: устный опрос.

Тема 3.2.6. Использование таймера. Ограничение времени в игре (1 час)

Теория:

- Что такое таймер? Встроенный таймер в Kodu. Создание ограничения по времени (таймер обратного отсчёта). Условия по таймеру: «когда таймер истёк → игра окончена». Сочетание таймера с системой очков. Настройка отображения таймера на экране.

Практика:

- Создание игры с ограничением времени: нужно собрать все предметы до истечения таймера. Если время вышло – проигрыш.

Форма контроля: устный опрос.

Тема 3.2.7. Программирование противников и препятствий (1 час)

Теория:

- Создание противников с автономным поведением: движение по патрулю (пути), преследование игрока, случайное движение. Использование Path Tool для задания маршрутов движения. Программирование препятствий: стены, ямы, движущиеся объекты. Взаимодействие игрока с противниками и препятствиями.

Практика:

- Создание противника, который движется по маршруту и преследует Kodu. Программирование условия: если противник касается Kodu → потеря жизни.

Форма контроля: наблюдение.

Тема 3.2.8. Создание уровней. Переход между уровнями (1 час)

Теория:

- Что такое уровень в игре? Создание нескольких уровней в одном проекте. Условия перехода на следующий уровень: сбор всех предметов, достижение определённого счёта. Программирование перехода: «перейти на уровень N» (Switch Page). Сброс состояния при переходе. Создание меню выбора уровня.

Практика:

- Создание двух уровней: на первом уровне нужно собрать 5 яблок, после перейти на второй уровень с новыми задачами.

Форма контроля: наблюдение.

Тема 3.2.9–10. Практическая работа: игра «Собери все яблоки» (2 часа)

Теория: не предусмотрена.

Практика:

- Создание игры «Собери все яблоки»: ландшафт, яблоки разбросаны по миру, персонаж Kodu управляется с клавиатуры, сбор яблок увеличивает счёт, таймер ограничивает время, при сборе всех яблок — победа. Использование изученных правил: движение, сбор предметов, счёт, таймер.

Форма контроля: практическая работа.

Тема 3.3. Создание проектов в Kodu (10 часов)

Тема 3.3.1. Создание гонки: трасса, автомобили, финиш (2 часа)

Теория:

- Проектирование гоночной игры: создание трассы (путь), размещение автомобилей, настройка управления. Программирование движения автомобиля. Условие финиша (пересечение линии финиша). Система очков (время прохождения). Добавление противников (автомобили-соперники). Создание круговой гонки.

Практика:

- Создание гоночной игры: трасса с поворотами, управляемый автомобиль, финишная линия, время прохождения.

Форма контроля: практическая работа.

Тема 3.3.2. Создание платформера: прыжки, платформы, сбор монет (2 часа)

Теория:

- Создание платформера в трёхмерном мире: платформы разной высоты, гравитация (настройка), прыжки (условие касания платформы). Сбор монет, размещённых на платформах. Условие победы: собрать все монеты. Добавление врагов на платформах. Движение игрока с клавиатуры.

Практика:

- Создание трёхмерного платформера: прыжки по платформам, сбор монет, избегание врагов.

Форма контроля: практическая работа.

Тема 3.3.3. Создание игры-бродилки: лабиринт, враги, ключи (2 часа)

Теория:

- Создание лабиринта с помощью стен и объектов. Персонаж, который должен найти выход из лабиринта. Ключи для открытия дверей. Враги, патрулирующие лабиринт. Условие победы: найти выход. Добавление счёта за собранные ключи и уничтоженных врагов. Создание системы жизней.

Практика:

- Создание игры-бродилки: лабиринт с ключами, врагами и выходом. Сбор ключей открывает путь к финишу.

Форма контроля: практическая работа.

Тема 3.3.4. Создание симулятора: сбор ресурсов, управление NPC (2 часа)

Теория:

- Создание симулятора (симуляция жизни/экономики): сбор ресурсов (еда, вода, древесина). Управление несколькими персонажами (NPC) с разным поведением. Программирование NPC: одни собирают ресурсы, другие строят, третьи двигаются по патрулю. Игрок управляет одним персонажем и взаимодействует с NPC. Цель: собрать определённое количество ресурсов для победы.

Практика:

- Создание симулятора: игрок собирает яблоки и воду, NPC помогают или мешают. Условие победы – набрать заданное количество ресурсов.

Форма контроля: практическая работа.

Тема 3.3.5. Создание собственного проекта «Моя трёхмерная игра» (2 часа)

Теория:

- Самостоятельный выбор жанра и темы игры: гонка, платформер, бродилка, лабиринт, симулятор. Проектирование игрового мира, правил, интерфейса. Планирование этапов работы: создание ландшафта, добавление объектов, программирование правил, тестирование. Постановка целей и задач для проекта.

Практика:

- Самостоятельная работа над собственным проектом в Kodu Game Lab. Создание игры по выбранной теме с использованием всех изученных конструкций.

Форма контроля: практическая работа (подготовка проекта к защите).

Тема 3.4. Защита проектов в Kodu Game Lab (8 часов)

Тема 3.4.1. Тестирование и отладка игровых проектов (2 часа)

Теория:

- Тестирование трёхмерных игр: проверка всех правил, всех возможных действий игрока, условий победы и поражения. Поиск и исправление ошибок в правилах. Оптимизация игрового процесса. Доработка баланса игры (скорость, сложность, количество врагов).

Практика:

- Тестирование и отладка собственных проектов в Kodu. Исправление ошибок в правилах, улучшение игрового процесса.

Форма контроля: наблюдение.

Тема 3.4.2. Создание скриншотов и описание игрового процесса (2 часа)

Теория:

- Как сделать скриншоты игрового мира (Print Screen). Создание описания игры: название, жанр, сюжет, правила игры, управление, особенности, цель. Оформление презентации проекта (в печатном или цифровом виде). Подготовка визуальных материалов (скриншоты).

Практика:

- Создание скриншотов своего игрового мира. Написание описания проекта по плану.

Форма контроля: наблюдение.

Тема 3.4.3. Подготовка презентации своего проекта (2 часа)

Теория:

- Структура презентации проекта: название, цели, сюжет и правила, управление, используемые алгоритмы и правила, скриншоты, демонстрация процесса игры. Как интересно представить свой проект. Подготовка устного выступления (3–5 минут). Репетиция защиты.

Практика:

- Создание презентации проекта. Подготовка устного выступления. Репетиция защиты.

Форма контроля: наблюдение.

Тема 3.4.4. Защита проектов «Моя трёхмерная игра» (2 часа)

Теория: не предусмотрена.

Практика:

- Публичная презентация и защита собственных проектов в Kodu Game Lab: демонстрация игры, объяснение сюжета, правил, использованных правил WHEN–DO. Ответы на вопросы одноклассников. Обсуждение работ, взаимооценка, рефлексия.

Форма контроля: защита проекта.

Раздел 4. Итоговый проект (8 часов)

Тема 4.1. Выбор темы итогового проекта. Постановка задач (1 час)

Теория:

- Обсуждение итогового проекта: можно выбрать любую среду (Scratch или Kodu Game Lab) и любую тему. Определение целей и задач проекта. Выбор жанра и сюжета. Обсуждение критериев оценки итогового проекта. Постановка сроков выполнения этапов работы.

Практика: не предусмотрена.

Форма контроля: беседа.

Тема 4.2. Разработка сценария и проектирование (1 час)

Теория:

- Проектирование игры: описание игрового мира, персонажей, правил, механик, интерфейса. Разработка сценария (что происходит на каждом этапе игры). Создание эскизов и схем. Планирование использования алгоритмов и правил.

Практика:

- Разработка письменного сценария игры и проектирование всех элементов проекта.

Форма контроля: наблюдение.

Тема 4.3–5. Создание итогового игрового проекта (3 часа)

Теория: не предусмотрена.

Практика:

- Самостоятельная работа над итоговым проектом:
- в Scratch: создание игры с использованием всех изученных конструкций (переменные, условия, циклы, события, взаимодействие спрайтов);
- в Kodu Game Lab: создание трёхмерного мира, программирование правил, настройка ландшафта, добавление объектов.
- Консультации с преподавателем. Помощь в решении сложных задач.

Форма контроля: наблюдение.

1.4. Планируемые результаты

1. Личностные результаты

Обучающиеся **сформируют:**

- познавательный интерес к программированию и IT-творчеству;
- ответственность, самостоятельность, инициативность, умение доводить дело до конца;
- навыки работы в команде, коммуникативные умения, культуру общения;
- информационную культуру и безопасное поведение в цифровой среде;
- уверенность в своих силах, творческую активность, эстетический вкус.

2. Метапредметные результаты

Регулятивные УУД: ставить цели, планировать работу, контролировать и оценивать свои действия, вносить коррективы, осуществлять рефлексию.

Познавательные УУД: анализировать, сравнивать, обобщать, устанавливать причинно-следственные связи, искать информацию, строить рассуждения, создавать модели.

Коммуникативные УУД: работать в паре/группе, распределять роли,

договариваться, аргументировать свою позицию, задавать вопросы, выступать перед аудиторией, защищать проект.

3. Предметные результаты

Обучающиеся будут знать:

- правила ТБ и организации рабочего места;
- основные понятия алгоритмики: **алгоритм, исполнитель, программа, команда, условие, цикл, событие, переменная**;
- интерфейс и инструменты сред **Scratch** и **Kodu Game Lab**;
- структуру программы в Scratch (блоки) и в Kodu (правила КОГДА–ДЕЛАЙ);
- основные алгоритмические конструкции (линейные алгоритмы, циклы, условия);
- этапы разработки игрового проекта.

Обучающиеся будут уметь:

- создавать анимации, интерактивные истории и игры в **Scratch** (лабиринт, платформер, викторина);
- программировать управление с клавиатуры и мыши в Scratch;
- создавать трёхмерные ландшафты, добавлять объекты и персонажей в **Kodu Game Lab**;
- программировать правила WHEN–DO: движение, взаимодействие, счёт, таймер, уровни, победу/поражение;
- создавать игры разных жанров в Kodu (гонки, платформеры, бродилки, симуляторы);
- тестировать и отлаживать свои проекты;
- оформлять и публично защищать проект (описание, презентация, ответы на вопросы).

Обучающиеся приобретут навыки:

- алгоритмического и пространственного мышления;
- проектной деятельности;
- самостоятельного поиска решений;
- работы в команде;
- публичного выступления;
- ИКТ-компетенций.

Раздел 2. Комплекс организационно–педагогических условий.

2.1. Календарный учебный график.

Находится в разработке.

Месяцы	Дни недели	Время	Форма занятий	Кол-во часов	Место	Форма контроля
			групповая	2	2 этаж, кабинет физики	

2.2. Условия реализации программы

2.2.1. Материально-техническое обеспечение

Наименование	Количество (из расчёта на группу 10–12 человек)
Компьютерный класс (рабочие станции)	10–12 ПК или ноутбуков
Компьютер для преподавателя с проектором/интерактивной доской	1 комплект
Доступ в Интернет (проводной или Wi-Fi)	Да
Операционная система	Windows 10/11 (или macOS)
Программное обеспечение: Scratch (офлайн-версия)	На всех ПК
Программное обеспечение: Kodu Game Lab	На всех ПК
Колонки или наушники	На всех ПК
Клавиатура и мышь	На всех ПК
Принтер (для печати дипломов, материалов)	1
Раздаточные материалы, рабочие	По количеству обучающихся

тетради	
Канцелярские принадлежности (для оформления проектов)	По количеству обучающихся

Требования к компьютерному классу:

- Соответствие санитарно-эпидемиологическим правилам и нормативам (СанПиН 2.4.2.2821-10);
- Наличие естественного и искусственного освещения;
- Температурный режим: 18–22°C;
- Регулярное проветривание;
- Наличие аптечки первой помощи, огнетушителя.

2.2.2. Информационное обеспечение

Ресурс	Назначение
https://scratch.mit.edu/	Официальный сайт Scratch, библиотека проектов, сообщество
https://www.kodugamelab.com/	Официальный сайт Kodu Game Lab, загрузка программы, ресурсы
https://code.org/	Образовательные курсы и материалы по программированию
Электронные методические пособия	Для преподавателя
Готовые проекты (Scratch и Kodu)	Для демонстрации и анализа на занятиях
Презентации по темам	Для объяснения нового материала

2.2.3. Кадровое обеспечение

Требование	Описание
Образование	Высшее педагогическое образование (учитель информатики, педагог дополнительного образования) или среднее профессиональное образование по соответствующему профилю
Квалификация	Владение средами Scratch и Kodu Game Lab на уровне, достаточном для

	обучения детей младшего школьного возраста
Профессиональные компетенции	Знание основ педагогики, возрастной психологии, методики преподавания информатики и программирования в начальной школе; владение ИКТ; умение организовывать проектную деятельность
Личностные качества	Коммуникабельность, терпение, доброжелательность, творческий подход, умение мотивировать детей

2.2.4. Методическое обеспечение

Компонент	Содержание
Учебно-методический комплект	Рабочие программы, методические рекомендации, конспекты занятий, раздаточные материалы, технологические карты
Дидактические материалы	Карточки с заданиями, логические игры, головоломки, схемы алгоритмов, инструкции по работе в средах
Демонстрационные материалы	Презентации по темам, видеоролики, готовые проекты для анализа, скриншоты, пошаговые инструкции
Материалы для контроля	Входные тесты, задания для практических работ, критерии оценки проектов, протоколы защиты проектов

2.2.5. Санитарно-гигиенические требования

Требование	Норматив
Продолжительность одного занятия	45 минут (1 академический час)
Непрерывная работа за компьютером	Не более 15–20 минут для младших школьников
Физминутки и гимнастика для	Проводится каждые 15–20 минут

глаз	занятия
Расстояние до монитора	Не менее 50–60 см
Освещённость	Естественное освещение + искусственное (не менее 300 люкс)
Температура воздуха	18–22°С
Влажность воздуха	40–60%
Проветривание	До и после занятий, во время перемены

2.2.6. Организационные условия

Параметр	Условие
Форма обучения	Очная, групповые занятия
Наполняемость группы	15 человек
Режим занятий	2 раза в неделю по 1 академическому часу (или 1 раз в неделю по 2 академических часа с переменной)
Продолжительность занятия	45 минут
Возраст обучающихся	7–11 лет (1–4 классы)
Набор в группы	Свободный (без предварительного отбора)
Форма приёма	Заявление родителей/законных представителей
Наличие программы	Данная программа

2.2.7. Психолого-педагогические условия

- Учёт возрастных и индивидуальных особенностей младших школьников;
- Создание ситуации успеха для каждого обучающегося;
- Поддержка познавательной активности и инициативы;
- Обеспечение эмоционально-комфортной атмосферы на занятиях;
- Использование игровых форм обучения как ведущей деятельности для младших школьников;
- Дифференциация заданий по уровню сложности;
- Поощрение творческого подхода и нестандартных решений;

- Взаимодействие с родителями (открытые занятия, презентации проектов).

2.3. Формы аттестации и контроля

2.3.1. Виды контроля

Вид контроля	Форма	Периодичность	Цель
Входной	Тестирование, логические задания	Начало курса (Тема 1.4)	Определение начального уровня развития логического мышления и знаний
Текущий	Устный опрос, наблюдение, проверка практических работ	Каждое занятие	Проверка усвоения материала на каждом этапе обучения
Промежуточный	Практические работы, мини-проекты	После каждой темы	Оценка сформированности умений и навыков
Итоговый (по разделам)	Защита проекта в Scratch	Тема 2.4.3	Проверка освоения раздела «Программирование в Scratch»
Итоговый (по разделам)	Защита проекта в Kodu	Тема 3.4.4	Проверка освоения раздела «Программирование в Kodu Game Lab»
Итоговый (финальный)	Итоговая защита проекта	Раздел 4	Комплексная проверка освоения всей программы

2.3.2. Формы подведения итогов реализации программы

Форма	Описание	Периодичность
Конкурс игр	Соревнование между обучающимися на лучший	В конце курса

	игровой проект	
Фестиваль цифрового творчества	Презентация проектов родителям, учителям, одноклассникам	В конце курса
Дипломы/грамоты	Награждение за лучшие проекты, активность, достижения	В конце курса

2.3.3. Критерии оценки проектов

Критерий	Максимальный балл	Показатели
Работоспособность	5 баллов	Проект запускается, все механики работают корректно, нет критических ошибок
Сложность	5 баллов	Использование изученных конструкций: условия, циклы, переменные, события
Оригинальность	5 баллов	Уникальность сюжета, творческий подход, самостоятельность выполнения
Оформление	5 баллов	Визуальная привлекательность, наличие инструкции, названия, экрана победы/поражения
Защита	5 баллов	Качество презентации, умение объяснить замысел и алгоритмы, ответы на вопросы
Итого	25 баллов	

Шкала оценивания:

- **22–25 баллов** – высокий уровень («Отлично»)
- **17–21 баллов** – средний уровень («Хорошо»)
- **10–16 баллов** – достаточный уровень («Удовлетворительно»)
- **менее 10 баллов** – низкий уровень («Требуется доработка»)

2.4. Методические материалы

Методы обучения

Методы обучения

1. Словесные методы

- **Объяснение** – изложение нового материала
- **Беседа** – вопросы-ответы, активизация знаний
- **Рассказ** – последовательное изложение темы
- **Инструктаж** – разъяснение порядка выполнения задания

2. Наглядные методы

- **Демонстрация** – показ готовых проектов, презентаций
- **Показ** – демонстрация действий преподавателем на экране
- **Иллюстрация** – плакаты, схемы, таблицы, скриншоты

3. Практические методы

- **Практическая работа** – самостоятельное выполнение заданий
- **Упражнение** – повторение для закрепления навыков
- **Проектная деятельность** – создание законченного игрового продукта
- **Экспериментирование** – самостоятельное исследование среды

4. Игровые методы

- **Учебная игра** – соревновательные задания
- **Игровые ситуации** – создание игрового контекста («ты — разработчик игр»)
- **Логические игры** – головоломки, лабиринты
- **Создание игр** – разработка проектов в Scratch и Kodu

5. Проблемно-поисковые методы

- **Проблемная ситуация** – задача, требующая поиска решения
- **Самостоятельный поиск** – решение без прямой подсказки учителя
- **Исследование** – самостоятельное изучение новых возможностей

6. Методы контроля

- **Устный опрос** – проверка знаний
- **Практическая проверка** – оценка выполнения задания
- **Тестирование** – письменная/компьютерная проверка
- **Защита проекта** – публичная презентация
- **Самоконтроль и взаимоконтроль** – проверка своих работ и работ товарищей

Формы организации занятий

Форма	Описание
Фронтальная	Работа всей группы под руководством преподавателя
Индивидуальная	Самостоятельная работа каждого
Парная	Работа в парах (взаимопомощь, парное программирование)
Групповая	Работа в малых группах (3–4 человека)

Рекомендации по выбору методов

Возраст	Рекомендуемые методы
1–2 класс (7–9 лет)	Игровые, наглядные, пошаговые инструкции, работа по образцу
3–4 класс (9–11 лет)	Проблемно-поисковые, проектная деятельность, самостоятельное исследование

2.5. Рабочая программа воспитания

1. Цель воспитания

Формирование у обучающихся социально значимых качеств: ответственности, самостоятельности, инициативности, коммуникативных способностей, творческой активности, информационной культуры и уважения к интеллектуальному труду.

2. Задачи воспитания

- воспитывать познавательный интерес к программированию и IT-творчеству;
- развивать навыки работы в команде и коммуникативные умения;
- формировать информационную культуру и безопасное поведение в цифровой среде;
- воспитывать ответственность, самостоятельность, умение доводить дело до конца;
- формировать уважение к результатам своего и чужого труда.

3. Направления воспитательной работы

Направление	Содержание
Гражданско-патриотическое	Беседы о безопасном поведении в Интернете, цифровом этикете; создание проектов на темы «Моя Родина», «Достопримечательности России»
Нравственное	Обсуждение честности, уважения к чужому труду (плагиат), взаимопомощи
Трудовое	Участие в оформлении кабинета, подготовке мероприятий, бережное отношение к оборудованию
Эстетическое	Оформление проектов, создание красивых игровых миров, развитие вкуса
Физическое и здоровьесберегающее	Физминутки, гимнастика для глаз, правильная посадка, соблюдение СанПиН

4. Календарный план воспитательной работы

Мероприятие	Срок	Форма
Инструктаж по технике безопасности	Сентябрь	Беседа, инструктаж
Создание поздравительных открыток в Scratch ко Дню учителя	Октябрь	Творческий проект

Неделя безопасности в Интернете	Ноябрь	Беседа, викторина
Новогодняя выставка игровых проектов	Декабрь	Выставка, конкурс
Создание тематических игр ко Дню защитника Отечества	Февраль	Творческий проект
Фестиваль проектов для родителей (открытое занятие)	Март	Презентация проектов
Создание игр на тему космоса ко Дню космонавтики	Апрель	Творческий проект
Итоговый фестиваль цифрового творчества	Май	Защита проектов, награждение

5. Планируемые результаты воспитания

Обучающиеся проявят:

Качество	Форма проявления
Ответственность	Бережное отношение к оборудованию, выполнение правил ТБ, доведение проектов до конца
Самостоятельность	Активность при выполнении заданий, инициатива в выборе тем проектов
Коммуникативные навыки	Умение работать в паре/группе, слушать других, аргументировать своё мнение
Творческая активность	Оригинальность проектов, нестандартные решения
Информационная культура	Безопасное поведение в Интернете, уважение к интеллектуальному труду

6. Формы подведения итогов воспитательной работы

- Выставки проектов;
- Конкурсы и фестивали;
- Презентации для родителей и одноклассников;
- Портфолио достижений;
- Награждение грамотами и дипломами.

2.6 СПИСОК ЛИТЕРАТУРЫ

1. Нормативно-правовые основы программы:

1. Федеральный Закон от 29.12.2012г. №273-ФЗ «Об образовании в Российской Федерации»;
2. Федеральный Закон от 30.04.2021г. №127-ФЗ «О внесении изменений в Федеральный Закон «О физической культуре и спорте в Российской Федерации» и в Федеральный Закон «Об образовании в Российской Федерации»;
3. Стратегия развития воспитания в РФ на период до 2025 года (распоряжение правительства РФ от 29 мая 2015 года №996-р);
4. Концепция развития дополнительного образования детей до 2030 года (утв. Распоряжением Правительства РФ от 31.03.2022 №678;
5. Постановление Главного государственного санитарного врача РФ от 02.06.2026 № 19 «Об утверждении санитарных правил СП 2.4.2.4283-26 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодёжи»;
6. Постановление главного государственного санитарного врача РФ от 28.01.2021 №2 «Об утверждении санитарных правил и норм СанПин 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;
7. Паспорт федерального проекта «Успех каждого ребенка» (утвержден на заседании проектного комитета по национальному проекту «Образование» 07.12.2018 г., протокол № 3).
8. Приказ Министерства просвещения Российской Федерации от 09.11.2018 г. № 196 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
9. Приказ Министерства Просвещения Российской Федерации от 03.09.2019 №467 «Об утверждении целевой модели развития региональных систем дополнительного образования детей»;
10. Приказ министерства образования и науки Российской Федерации и министерства просвещения Российской Федерации от 5.08.2020г. №882/391 «Об организации и осуществлении образовательной деятельности по сетевой форме реализации образовательных программ»;
11. Приказ Министерства просвещения Российской Федерации от 27.07.2022г. №629 «Об утверждении Порядка организации и осуществления образовательной

деятельности по дополнительным общеобразовательным программам»;

12. Приказ Министерства образования и науки Российской Федерации от 23.08.2017 г. № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ».

13. Методические рекомендации по проектированию дополнительных общеобразовательных программ (включая разноуровневые программы) (разработанные Минобрнауки России совместно с ГАОУ.ВО «Московский государственный педагогический университет», ФГАУ «Федеральный институт развития образования», АНО ДПО «Открытое образование», 2015г.)(Письмо Министерства образования и науки РФ от 18.11.2015 №09-3242).

14. Устав МКОУ Новомихайловская СОШ.

2. Основная литература (для педагога)

1. Астахова, К. И. Создаем игры с Kodu Game Lab : учебно-методическое пособие / К. И. Астахова. – Москва : Лаборатория знаний, 2023. – 128 с. – ISBN 978-5-00101-345-7.
2. Винницкий, Ю. А. Scratch и Kodu Game Lab в начальной школе : методическое пособие / Ю. А. Винницкий. – Москва : Бином. Лаборатория знаний, 2019. – 160 с. – ISBN 978-5-9963-4256-5.
3. Горячев, А. В. Информатика. 3 класс : учебник : в 2 ч. / А. В. Горячев, К. И. Астахова, Е. В. Горячева. – Москва : Просвещение, 2025. – Ч. 1. – 112 с. – ISBN 978-5-09-109876-5.
4. Горячев, А. В. Информатика. 4 класс : учебник : в 2 ч. / А. В. Горячев, К. И. Астахова, Е. В. Горячева. – Москва : Просвещение, 2025. – Ч. 1. – 112 с. – ISBN 978-5-09-109876-5.
5. Горячев, А. В. Лаборатория компьютерных игр. Игры. Исследования. Эксперименты. 3–4 классы : методическое пособие / А. В. Горячев, А. В. Каплан, Д. И. Павлов. – Москва : Бином. Лаборатория знаний, 2020. – 96 с. – ISBN 978-5-9963-4257-2.
6. Патаракин, Е. Д. Педагогика цифрового творчества: от Scratch до Kodu : монография / Е. Д. Патаракин. – Москва : НИУ ВШЭ, 2021. – 200 с. – ISBN 978-5-7598-2456-3.
7. Тимофеева, А. А. Формирование алгоритмического мышления у младших школьников : учебно-методическое пособие / А. А. Тимофеева. – Москва : Педагогическое общество России, 2021. – 144 с. – ISBN 978-5-93134-456-7.
8. Щербакова, М. Н. Интерактивные среды программирования в образовательном процессе : учебное пособие / М. Н. Щербакова. – Москва : Юрайт, 2020. – 180 с. – ISBN 978-5-534-12345-6.

3. Дополнительная литература (для педагога)

1. Босова, Л. Л. Информатика. 3–4 классы : методическое пособие / Л. Л. Босова, А. Ю. Босова. – Москва : Бином. Лаборатория знаний, 2021. – 240 с. – ISBN 978-5-9963-4250-3.

2. Колесникова, Н. И. Информатика в начальной школе: современные подходы : учебно-методическое пособие / Н. И. Колесникова. – Москва : Просвещение, 2020. – 128 с. – ISBN 978-5-09-076543-2.
3. Плаксин, М. А. Информатика. 3 класс : учебник : в 2 ч. / М. А. Плаксин, Н. Г. Иванова, О. Л. Русакова. – Москва : Просвещение, 2025. – Ч. 1. – 112 с. – ISBN 978-5-09-109876-5.
4. Плаксин, М. А. Информатика. 4 класс : учебник : в 2 ч. / М. А. Плаксин, Н. Г. Иванова, О. Л. Русакова. – Москва : Просвещение, 2025. – Ч. 1. – 112 с. – ISBN 978-5-09-109876-5.
5. Microsoft Education. Teaching with Kodu Game Lab : методическое руководство / Microsoft Education. – Redmond : Microsoft Corporation, 2018. – 80 с.

4. Литература для обучающихся

1. Астахова, К. И. Создаем игры с Kodu Game Lab : практикум для школьников / К. И. Астахова. – Москва : Лаборатория знаний, 2023. – 112 с. – ISBN 978-5-00101-345-7.
2. Голиков, Д. В. Scratch для школьников. Создаём игры и анимации / Д. В. Голиков. – Санкт-Петербург : Питер, 2021. – 208 с. – ISBN 978-5-4461-1789-3.
3. Демидова, Е. А. Kodu Game Lab: создаём трёхмерные игры / Е. А. Демидова. – Санкт-Петербург : BHV, 2022. – 192 с. – ISBN 978-5-9775-6789-7.
4. Кузьмин, А. А. Создание игр в Scratch. Практикум / А. А. Кузьмин. – Москва : ДМК Пресс, 2021. – 160 с. – ISBN 978-5-97060-876-5.
5. Попов, А. С. Программирование для детей: Scratch и Kodu / А. С. Попов. – Москва : АСТ, 2020. – 224 с. – ISBN 978-5-17-123456-7.
6. Сухов, А. В. Игры и анимации в Scratch для начинающих / А. В. Сухов. – Москва : Эксмо, 2021. – 192 с. – ISBN 978-5-04-123456-8.

5. Электронные ресурсы

1. Образовательная платформа [Code.org](https://code.org/) [Электронный ресурс] : официальный сайт. – Режим доступа: <https://code.org/> (дата обращения: 15.03.2026).
2. Официальный сайт Kodu Game Lab [Электронный ресурс] : официальный сайт. – Режим доступа: <https://www.kodugamelab.com/> (дата обращения: 15.03.2026).
3. Официальный сайт Scratch [Электронный ресурс] : официальный сайт. – Режим доступа: <https://scratch.mit.edu/> (дата обращения: 15.03.2026).
4. Российская электронная школа [Электронный ресурс] : интерактивные уроки по информатике. – Режим доступа: <https://resh.edu.ru/> (дата обращения: 15.03.2026).
5. Учительский портал [Электронный ресурс] : методические материалы для учителей информатики. – Режим доступа: <https://www.uchportal.ru/> (дата обращения: 15.03.2026).
6. Яндекс.Учебник [Электронный ресурс] : образовательная платформа. – Режим доступа: <https://education.yandex.ru/> (дата обращения: 15.03.2026).