

Муниципальное бюджетное общеобразовательное учреждение  
«Максатихинская средняя общеобразовательная школа №1»

**Дополнительная общеобразовательная общеразвивающая программа научно-  
технической направленности  
«Увлекательное путешествие в Scratch»**

Возраст обучающихся: 7 - 13 лет  
Нормативный срок освоения программы: 2 года

Педагог  
Шибает Сергей Андреевич

Рассмотрено на заседании  
педагогического совета  
протокол № \_\_\_\_\_  
от «    » августа 2023 г.

2023-2024 учебный год.

## Пояснительная записка

Программа дополнительного образования детей технической направленности «Увлекательное программирование. Scratch» составлена на основе авторской программы «Творческие задания в среде программирования Scratch», которая входит в сборник «Информатика. Математика. Программы внеурочной деятельности для начальной и основной школы: 3-6 классы» / М.С. Цветкова, О.Б.Богомолова. - М.: БИНОМ. Лаборатория знаний, 2013. - 128 с.: ил. Программа модифицирована.

### **Нормативно-правовая база Настоящая Программа**

разработана в соответствии с:

1. Федеральным Законом «Об образовании в Российской Федерации» от 29 декабря 2012 года №273-ФЗ;
2. Учебным планом МБОУ «Максатихинская СОШ №1»
3. Основной образовательной программы основного общего образования МБОУ «Максатихинская СОШ №1»
4. Положением об организации внеурочной деятельности МБОУ «Максатихинская СОШ №1».
5. Приказом Министерства Образования и науки от 09.11.2018 № 196 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

**Направленность дополнительной общеобразовательной общеразвивающей программы:**  
техническая

**Уровень программы:** базовый **Актуальность программы:**

Мы живем в век информатизации общества. Информационные технологии проникают в нашу жизнь с разных сторон. Одно из самых удивительных и увлекательных занятий настоящего времени - программирование.

Обучение основам программирования школьников должно осуществляться на специальном языке программирования, который будет понятен детям, будет легок для освоения и соответствовать современным направлениям в программировании.

Основные особенности Scratch:

1. Блочное программирование
2. Манипуляции данными
3. Совместная работа и обмен.

Одной из главных концепций языка Scratch, является развитие собственных задумок с первой идеи до конечного программного продукта. Для этого в Scratch имеются все необходимые средства:

- стандартные для языков процедурного типа: следование, ветвление, циклы, переменные, типы данных (целые и вещественные числа, строки, логические, списки — динамические массивы), псевдослучайные числа;
- объектно-ориентированные: объекты (их поля и методы), передача сообщений и обработка событий;
- интерактивные: обработка взаимодействия объектов между собой, с пользователем, а также событий вне компьютера (при помощи подключаемого сенсорного блока);
- параллельное выполнение: запуск методов объектов в параллельных потоках с возможностью координации и синхронизации;
- создание простого интерфейса пользователя.

**Отличительные особенности:** данная программа обучения основана на преимуществах дополнительного образования и призвана дать необходимые знания и умения в области изучения компьютерных технологий для школьников, а также выявить способных, талантливых детей и развить их способности, дает большие возможности для творческого развития детей, предусматривая индивидуальный подход к ребенку. Благодаря изучению данного курса обучающиеся познакомятся с основами программирования и алгоритмического мышления, что позволит применять полученные знания на предметах информатики и математики.

Отличительной особенностью программы является ее частичная реализация с применением электронного обучения и дистанционных образовательных технологий: СДО ДОМ 365 (разработка цикла занятий); видеоконтент канала YouTube«Уроки по Scratch» и др.

### **Новизна программы**

Актуальность программы состоит в том, что мультимедийная среда Scratch позволяет сформировать у детей стойкий интерес к программированию, отвечает всем современным требованиям объектно-ориентированного программирования.

Изучение языка значительно облегчает последующий переход к изучению других языков программирования. Преимуществом Scratch, среди подобных сред программирования, также является наличие версий для различных операционных систем: для Windows, MacOS, GNU/Linux. В настоящее время имеет смысл рассматривать программы с открытым кодом, что позволяет сформировать у учащихся более широкое представление о возможностях работы с цифровой техникой.

Скретч — свободно распространяемая программа, скачать которую можно, к примеру, с сайта: <http://info.scratch.mit.edu/Scratch1.4> Downloaded.

Аспект новизны заключается в том, что Scratch — это не просто язык программирования, а еще и интерактивная среда, где результаты действий визуализированы, что делает работу с программой понятной, интересной и увлекательной.

Особенность среды Scratch, позволяющая создавать в программе мультфильмы, анимацию и даже простейшие игры, делает образовательную программу «Увлекательное программирование» практически значимой для современного школьника, т.к. дает возможность увидеть практическое назначение алгоритмов и программ, что будет способствовать развитию интереса к профессиям, связанным с программированием.

В зависимости от поставленных задач на занятиях используются различные методы обучения. Все задания расположены по возрастанию степени их сложности. Доступная детям практическая деятельность помогает избежать умственного переутомления.

**Педагогическая целесообразность** данной образовательной программы состоит в том, что, изучая программирование в среде Scratch, у обучающихся формируется не только логическое мышление, но и навыки работы с мультимедиа; создаются условия для активного, поискового учения, предоставляются широкие возможности для разнообразного моделирования.

**Адресат программы:** возраст детей, участвующих в реализации дополнительной общеобразовательной общеразвивающей программы «Увлекательное программирование Scratch» 10-12 лет с базовым уровнем владения ИКТ, наполняемость групп 10-12 человек.

**Практическая значимость:** программа дополнительного образования «Увлекательное программирование Scratch» является практически значимой для современного школьника, т.к. дает возможность увидеть практическое назначение алгоритмов и программ, через создание развивающих игр, видеороликов и других программных продуктов, что будет способствовать развитию интереса к профессиям, связанным с программированием.

**Преимственность программы.** Программа позволяет развить творческие способности учащихся, а также закладывает пропедевтику наиболее значимых тем курса информатики и позволяет успешно готовиться к участию в олимпиадах по математике и информатике, изучается в тесном взаимодействии с данными предметами.

## Особенности реализации образовательного процесса.

- формы организации деятельности обучающихся:
  - ✓ групповые,
  - ✓ индивидуальные;
- методы обучения
  - ✓ словесные методы (лекция, объяснение);
  - ✓ демонстративно-наглядные (демонстрация работы в программе, схем, скриптов, таблиц);
  - ✓ проектные методы (разработка проекта по спирали творчества, моделирование, планирование деятельности)
  - ✓ работа с Интернет - сообществом (публикация проектов в Интернет - сообществе скретчеров).
  - ✓ объяснительно - иллюстративные (методы обучения, при использовании которых, дети воспринимают и усваивают готовую информацию);
  - ✓ частично-поисковые методы обучения (участие детей в
  - ✓ коллективном поиске, решение поставленной задачи совместно с педагогом). исследовательские методы обучения (овладение детьми методами научного познания, самостоятельной творческой работы).

### **Технологии обучения и их обоснование:**

*Технологии проектной деятельности* (Дж. Дьюи), согласно которой, итогом деятельности учащихся является творческий продукт.

Здоровье сберегающие технологии - это система работы образовательного пространства по сохранению и развитию здоровья всех участников - взрослых и детей. Огромное значение в работе детских объединений имеет обучение здоровому образу жизни, активной жизни, активной жизненной позиции, приема саморегуляции и закаливания, психорегулирующей тренировке и двигательным навыкам на основе учета индивидуальных законов развития.

Технологии лично-ориентированного обучения В центре внимания - личность подростка, который должен реализовать свои возможности. Содержание, методы и приемы лично-ориентированных технологий обучения направлены, прежде всего, на то, чтобы раскрыть и использовать субъективный опыт каждого воспитанника, помочь становлению личности путем организации познавательной деятельности.

Технология развивающего обучения является дополнением лично-ориентированного обучения. Его цель - создание условий для развития психологических

особенностей: способностей, интересов, личностных качеств и отношений между людьми; при котором учитываются и используются закономерности развития, уровень и особенности индивидуума.

Технология сотрудничества - это технология совместной развивающей деятельности взрослых и детей, скрепленной взаимопониманием, проникновением в духовный мир друг друга, совместным анализом хода и результата этой деятельности.

#### Игровые технологии

В основу положена игра как вид деятельности. Эта технология используется для обеспечения наибольшей активности обучающихся и достижения высокого уровня освоения содержания программы.

**Формы и методы обучения** определены возрастом учащихся. При проведении занятий используются компьютеры с установленной программой Scratch, проектор, сканер, принтер, компьютерная сеть с выходом в Интернет. Теоретическая работа чередуется с практической, а также используются интерактивные формы обучения.

**Формы проведения занятий:** очная, очно-заочная. При реализации программы (частично) применяется электронное обучение и дистанционные образовательные технологии.

Использование метода проектов позволяет обеспечить условия для развития у ребят навыков самостоятельной постановки задач и выбора оптимального варианта их решения, самостоятельного достижения цели, анализа полученных результатов с точки зрения решения поставленной задачи.

Программой предусмотрены **методы обучения:** объяснительно-иллюстративные, частично-поисковые (вариативные задания), творческие, практические.

**Формы подведения итогов реализации программы** внеурочной деятельности: учебно-исследовательская конференция, защита проектов.

Режим занятий. Занятия проводятся 1 раз в неделю по одному академическому часу. Продолжительность занятия 45 минут. Срок освоения программы - 2 года. Количество часов - 72.

При электронном обучении с применением дистанционных технологий продолжительность непрерывной непосредственно образовательной деятельности составляет не более 30 минут. Во время онлайн-занятия проводится динамическая пауза, гимнастика для глаз.

**Цель:** сформировать у учащихся базовые представления о языках программирования, алгоритме, исполнителе, способах записи алгоритма; организовать проектную научно-познавательную деятельность творческого характера; сформировать у школьника познавательный интерес к учёбе и исследовательские навыки.

Занятия направлены на решение следующих задач.

**Задачи программы:**

**Обучающие:**

- ✓ овладеть навыками составления алгоритмов;
- ✓ овладеть понятиями «объект», «событие», «управление», «обработка событий»;
- ✓ изучить функциональность работы основных алгоритмических конструкций;
- ✓ сформировать представление о профессии «программист»;
- ✓ сформировать навыки разработки, тестирования и отладки несложных программ;
- ✓ сформировать ИТ-компетенции;
- ✓ познакомить с понятием проекта и алгоритмом его разработки;
- ✓ сформировать навыки разработки проектов: интерактивных историй, квестов, интерактивных игр, обучающих программ, мультфильмов, моделей и интерактивных презентаций.

**Развивающие:**

- ✓ способствовать развитию критического, системного, алгоритмического и творческого мышления;
- ✓ развивать внимание, память, наблюдательность; познавательный интерес;
- ✓ развивать умение работать с компьютерными программами и дополнительными источниками информации;
- ✓ развивать навыки планирования проекта, умение работать в группе.

**Воспитательные:**

- ✓ формировать положительное отношение к информатике и ИКТ;
- ✓ развивать самостоятельность и формировать умение работать в паре, малой группе, коллективе;
- ✓ формировать умение демонстрировать результаты своей работы.

**Учебно-тематический план**

№ /п	Наименование темы	Кол-во часов				Формы аттестации/ контроля
		Теория	Практика	Всего	ДО	
<b>I. Интерфейс программы Scratch(1 ч)</b>						
1	Введение. Что такое Scratch. Основные алгоритмические конструкции. Знакомство с интерфейсом программы Scratch.	1	0	1		исследование, практические задачи
<b>II. Начало работы в среде Scratch (4 ч)</b>						
2	Сцена. Редактирование фона. Добавление фона из файла.	1	1	2		исследование, практические задачи
3	Понятие спрайтов. Добавление новых спрайтов. Рисование новых объектов.	1	1	2		исследование, практические задачи

<b>III. Основные скрипты программы Scratch ( 18 ч )</b>						
4-5	Синий ящик - команды движения. Темно-зеленый ящик - команды рисования.	1	1	2		исследование, практические задачи
6-7	Фиолетовый ящик - внешний вид объекта. Оживление объекта с помощью добавления костюмов.	1	1	2		исследование, практические задачи
8-9	Желтый ящик - контроль. Лиловый ящик - добавление звуков.	1	1	2		исследование, практические задачи
10	Использование в программах условных операторов.	1	1	2		исследование, практические задачи
12	Функциональность работы циклов.	1	1	2		исследование, практические задачи
13	Цикличность выполнения действий в зависимости от поставленных условий.					
14	Зеленый ящик - операторы.	1	1	2		исследование, практические задачи
15	Использование арифметических и логических блоков вместе с блоками управления.					
16	События. Оранжевый ящик - переменные.	1	1	2		исследование, практические задачи
18	Списки.	1	1	2		исследование, практические задачи
19						
20	Голубой ящик - сенсоры. Ввод- вывод данных.	1	1	2		исследование, практические задачи
21						
<b>IV. Работа с несколькими объектами. Синхронизация их работы ( 4 ч )</b>						
22	Последовательность и параллельность выполнения скриптов.	1	1	2		исследование, практические задачи
23						
24	Взаимодействие между спрайтами. Управление через обмен сообщениями.	1	1	2		исследование, практические задачи
25						
<b>V. Использование программы Scratch для создания мини-игр ( 10 ч )</b>						
26	Виды компьютерных игр. Алгоритмическая разработка листинга программы.	1	1	2		Создание проектов
27						
28	Разработка базовых спрайтов для игры. Формирование базовых скриптов.	1	1	2		Создание проектов
29						
30	Синхронизация работы скриптов для разных спрайтов.	0	1	1		Создание проектов
31	Переход из одной сцены в другую. Создание интерфейса игры.	1	1	2		Создание проектов
32	Сообщество Scratchв Интернете. Просмотр и публикация проектов.	1	1	2		Создание проектов
<b>VI. Разработка творческого проекта ( 2 ч )</b>						
33-	Разработка и защита творческого	0	2	2		Защита



34	проекта					собственного проекта
<b>Итого:</b>		<b>18</b>	<b>20</b>	<b>38</b>		

# СОДЕРЖАНИЕ ПРОГРАММЫ

## I. Интерфейс программы Scratch (1 ч).

### 1. Введение. Что такое Scratch. Основные алгоритмические конструкции.

#### Знакомство с интерфейсом программы Scratch.

**Теория.** История создания среды Scratch. Основные базовые алгоритмические конструкции (линейные алгоритмы, с условным оператором, циклического типа с предусловием и постусловием) и их исполнение в среде Scratch. Понятие исполнителя, алгоритма и программы, их назначение, виды и использование. Виды управления исполнителем. Способы записи алгоритма. Основные характеристики исполнителя. Система команд исполнителя. Понятие проект, его структура и реализация в среде Scratch. Основные компоненты проекта Scratch: спрайты и скрипты. Принцип создания анимации и движения объектов. Листинг программы. Сцена. Текущие данные о спрайте. Стилль поворота. Закладки. Панель инструментов, Новый спрайт. Координаты мышки. Режим представления. Окно скриптов. Окно блоков. Блоки стека. Блоки заголовков. Блоки ссылок. Самодостаточные и открытые скрипты (1 час).

## II. Начало работы в среде Scratch(2 ч).

### 2. Сцена. Редактирование фона. Добавление фона из файла.

**Теория.** Сцена. Ширина и высота сцены. Текущие координаты объекта. Редактирование текущего фона. Вставка нового фона из файла. Вставка стандартного фона из библиотечного модуля среды. Рисование фона в графическом редакторе. Создание нескольких фонов в одной сцене (0,5 часа).

**Практика.** Создание фона сцены на выбранную учащимся тему (0,5 часа).

### 3. Понятие спрайтов. Добавление новых спрайтов. Рисование новых объектов.

**Теория.** Стандартный объект. Спрайты. Список спрайтов. Редактор рисования для создания новых спрайтов. Инструменты рисования (кисточка, линия, текст, эллипс, ) и редактирования объекта (ластик, заливка, поворот, выбор, печать, пипетка). Центрирование костюма. Масштабирование спрайта. Загрузка на сцену спрайтов из стандартной коллекции среды Scratch. Вставка спрайтов из файлов форматов JPG, BMP, PNG, GIF. Выбор случайного спрайта. Удаление спрайтов (0,5 часа).

**Практика.** Создание фона сцены и прорисовка основных спрайтов для Scratch-истории. (0,5 часа).

## III. Основные скрипты программы Scratch (18 ч).

### 4. Синий ящик - команды движения. Темно-зеленый ящик - команды рисования.

**Теория.** Команды *идти*; *вернуться направо (налево)*; *вернуть в направлении*; *вернуться к*; *изменить x (y) на*; *установить x (y) в*; *если край, оттолкнуться*. Принципиальное различие действия команд *идти в* и *плыть в*. Назначение сенсоров *положение x*, *положение y* и *направлении*. Команды *очистить*, *опустить перо*, *поднять перо*, *установить цвет пера*, *изменить цвет пера на*, *установить цвет пера*, *изменить тень пера*, *установить тень пера*, *изменить размер пера на*, *установить размер пера*, *печать* (0,5 часа).

**Практика.** Создание программ для передвижения спрайтов по сцене. Создание программ для рисования различных фигур (1,5 часа).

## **5. Фиолетовый ящик - внешний вид объекта. Оживление объекта с помощью добавления костюмов.**

**Теория.** Костюмы спрайта. Копирование и редактирование костюма спрайта с помощью редактора рисования. Переупорядочивание костюмов. Команды - *перейти к костюму, следующий костюм, говорить...в течении...секунд, сказать, думать, думать...секунд, изменить...эффект на, установить эффект.в значение, убрать графические эффекты, изменить размер на, установить размер, показаться, спрятаться, перейти в верхний слой, перейти назад на...1 слоев.* Назначение сенсоров *костюм* и *размер*. Понятие раскадровки движения. Изменение костюма спрайта для имитации движения (0,5 часа).

**Практика.** Создание программы для управления внешним видом объекта. Создание Scratch-историй с имитацией хождения и движения объектов (1,5 часа).

## **6. Желтый ящик - контроль. Лиловый ящик - добавление звуков.**

**Теория.** Кнопка с зеленым флажком и ее назначение. Управление последовательностью выполнения скриптов. Понятие управляющих сообщений. Команды - *передать, передать и ждать, когда я получу.* Скрипты для создания условных конструкций программы - *если, если.или.* Скрипты для управления циклами - *всегда, повторить, всегда, если, повторять до..* Команды - *когда клавиша.нажата, когда щелкнут по, ждать.секунд, ждать до, остановить скрипт, остановить все.* Загрузка звуков из стандартной коллекции и из файлов жесткого диска. Запись звука через микрофон. Принципиальная разница работы команд *играть звук* и *играть звук до завершения.* Команды - *остановить все звуки, барабану играть.тактов, оставшиеся.тактов, ноту...играть...тактов, выбрать инструмент, изменить громкость, установить громкость, изменить темп на, установить темп.* Назначение сенсоров *громкость* и *темп* (0,5 часа).

**Практика.** Создание программ с элементами управления объектом. Озвучивание Scratch-историй (1,5 часа).

## **7. Использование в программах условных операторов.**

**Теория.** Базовая конструкция ветвление, назначение, виды (полная и неполная форма). Понятие условия. Изменение порядка выполнения скриптов в зависимости от условия. Разветвление листинга программы. Скрипты условных операторов. Использование неполной формы ветвления в системе Scratch(0,5 часа).

**Практика.** Создание программ с изменением последовательного выполнения скриптов при наличии условий (1,5 часа).

## **8. Функциональность работы циклов. Цикличность выполнения действий в зависимости от поставленных условий.**

**Теория.** Циклы с фиксированным числом повторений. Заголовок цикла. Тело цикла. Циклы с условным оператором. Заголовок цикла. Тело цикла. Предусловие и постусловие. Зацикливание (0,5 часа).

**Практика.** Создание программ с использованием циклов с фиксированным числом повторений. Создание программ с использованием циклов с предусловием и постусловием (1,5 часа).

## **9. Зеленый ящик - операторы. Использование арифметических и логических блоков вместе с блоками управления.**

**Теория.** Числа. Строинги. Логические величины. Логические выражения. Арифметические операции. Логические операции. Операции сравнения. Команды для работы со строингами - *слить, буква.в, длинна строки.* Команда *выдать случайное от.до.* Использование

арифметических и логических блоков в листинге программы. Просмотр полученного результата (0,5 часа).

**Практика.** Создание программ с использованием операций сравнения данных. Создание программ с использованием арифметических данных и логических операций (1,5 часа).

#### **10. События. Оранжевый ящик - переменные.**

**Теория.** События в проектах Scratch. Понятие переменных и необходимость их использования в листинге программы. Глобальные и локальные переменные. Имя переменной и правила его формирования. Команды для переменных - *поставить...в, изменить...на, показать переменную, спрятать переменную*. Удаление переменных. Создание счетчиков с помощью переменных (0,5 часа).

**Практика.** Разработка сценария Scratch-историй с несколькими событиями. Создание проектов с использованием глобальных и локальных переменных (1,5 часа).

#### **11. Списки.**

**Теория.** Создание списков и необходимость их использования в проектах Scratch. Добавление в список данных. Удаление данных из списка. Удаление списка. Команды работы со списками - *добавить...к, удалить... из, поставить ...в...из, заменить элемент...в...на, элемент.из, длина списка* (0,5 часа).

**Практика.** Создание программ-тестов по принципу сравнения данных из нескольких списков (1,5 часа).

#### **12. Голубой ящик - сенсоры. Ввод-вывод данных.**

**Теория.** Понятие сенсора. Правила применения и область действия команд *касается, касается цвета и цвет, касается*.

Функционал

команды *спросить.и ждать*. Сенсоры *мышка по x, мышка по y, мышка нажата?, клавиша.нажата?, расстояние до, перезапустить таймер*. Сенсоры, значение которых можно выводить на экран - *ответ, таймер, громкость, громко?,...значение сенсора и сенсор...* Необходимость ввода данных для их обработки в программе. Ввод данных с помощью команды *спросить*. Вывод конечного результата обработки с помощью команд *говорить и сказать* (0,5 часа).

**Практика.** Создание проектов с использованием значений сенсоров и команды *спросить*. Создание программ для обработки данных пользователя с выводом на экран конечного результата (1,5 часа).

### **IV. Работа с несколькими объектами. Синхронизация их работы (4 ч).**

#### **13. Последовательность и параллельность выполнения скриптов.**

**Теория.** Последовательные и параллельные потоки в программах Scratch. Одновременная и попеременная работа нескольких исполнителей (0,5 часа).

**Практика.** Создание Scratch-историй с одновременной и попеременной работой нескольких исполнителей (1,5 часа).

#### **14. Взаимодействие между спрайтами. Управление через обмен сообщениями.**

**Теория.** Решение проблемы появления новых исполнителей только после того, как старые исполнители выполнили свои действия. Взаимодействие спрайтов с неподвижными объектами с помощью команд *касается* и *касается цвета*. Взаимодействие спрайтов с помощью команд *передать* и *когда я получу*. Использование сообщений для создания событий (1 час).

**Практика.** Создание Scratch-историй с взаимодействием нескольких исполнителей и неподвижных объектов. Создание Scratch-историй с взаимодействием нескольких исполнителей (1 час).

## **V. Использование программы Scratch для создания мини-игр (7 ч).**

**15. Виды компьютерных игр. Алгоритмическая разработка листинга программы.**

**Теория.** Компьютерные игры - вред или польза. Виды компьютерных игр. Этапы разработки игр программистами (1 час).

**Практика.** Алгоритмическая разработка проекта, запись на естественном языке событий и точек взаимодействия героев будущей игры (1 час).

**16. Разработка базовых спрайтов для игры. Формирование базовых скриптов.**

**Теория.** Логика создания персонажей для игры. Перевод алгоритма, написанного на естественном языке, в коды Scratch(0,5 часа).

**Практика.** Разработка и создание основных спрайтов и их костюмов для будущей игры. Разработка скриптов для спрайтов и объектов (1,5 часа).

**17. Синхронизация работы скриптов для разных спрайтов.**

**Практика.** Доработка основного листинга программы с целью установления связей между спрайтами. Тестирование и отладка программы (1 час).

**18. Переход из одной сцены в другую. Создание интерфейса игры.**

**Теория.** Односторонний (без возможности вернуться назад) переход из одного пространства в другое. Понятие интерфейса. Элементы интерфейса. Основные принципы дизайна интерфейсов. Обратная связь. Необходимые элементы меню (0,5 час).

**Практика.** Создать программу для перемещения объекта по игровой карте и разработать интерфейс для Scratch-проекта (0,5 часа).

**19. Сообщество Scratch в Интернете. Просмотр и публикация проектов.**

**Теория.** Правила работы в сети. Интернет-сообщества. Сообщество Scratch. Регистрация на сайте. Использование заимствованных кодов и объектов. Авторские права. Публикация проектов Scratch. (0,5 часа).

**Практика.** Регистрация на сайте сообщества Scratch. Просмотр проектов сообщества и публикация собственных проектов (0,5 часа).

## **VI. Разработка творческого проекта (4 ч)**

**20. Разработка и защита творческого проекта.** Разработка и создание программы с использованием подготовленных материалов. Тестирование и отладка проекта. Защита проекта (4 часа).

## Планируемые результаты

### ***Личностные результаты***

К личностным результатам освоения информационных и коммуникационных технологий как инструмента в учёбе и повседневной жизни можно отнести:

- критическое отношение к информации и избирательность её восприятия;
- уважение к информации о частной жизни и информационным результатам других людей;
- осмысление мотивов своих действий при выполнении заданий с жизненными ситуациями;
- начало профессионального самоопределения, ознакомление с миром профессий, связанных с информационными и коммуникационными технологиями.

***Метапредметными результатами*** изучения курса «Увлекательное программирование Scratch» являются формирование следующих универсальных учебных действий:

### **Регулятивные УУД:**

- планирование последовательности шагов алгоритма для достижения цели;
- поиск ошибок в плане действий и внесение в него изменений.

### **Познавательные УУД:**

- моделирование - преобразование объекта из чувственной формы в модель, где выделены существенные характеристики объекта (пространственно-графическая или знаково-символическая);
- анализ объектов с целью выделения признаков (существенных, несущественных);
- синтез - составление целого из частей, в том числе самостоятельное достраивание с восполнением недостающих компонентов;
- выбор оснований и критериев для сравнения, сериации, классификации объектов;
- подведение под понятие;
- установление причинно-следственных связей;
- владение ИКТ-компетенциями;
- построение логической цепи рассуждений.

### **Коммуникативные УУД:**

- аргументирование своей точки зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов;
- выслушивание собеседника и ведение диалога;
- признание возможности существования различных точек зрения и права каждого иметь свою.

### ***Предметные результаты .***

*Обучающийся:*

- Овладеет этапами проектной деятельности.
- Научится использованию различных методов создания, отладки и корректировки проектов в среде Scratch.
- Научится использованию инструментов встроенного графического редактора (создание и сохранение изображений и спрайтов).
- Получит знания основных видов и задач творческих олимпиад по креативному программированию.

Результативность освоения программы отслеживается на практических занятиях, на которых выполняются определенные задания и после каждого изученного раздела заполняется диагностическая карта успеваемости.

## Условия реализации программы

### Материально-технические условия и информационное обеспечение для реализации модулей программы

Учебная лекционная аудитория, компьютерный класс (15 - 20 рабочих мест). Компьютеры объединены в локальную сеть и имеют выход в Интернет, установлена операционная система Windows, Web-browser, текстовый процессор Word, Scratchv1.4.

Необходим высокоскоростной канал для подключения к сети Интернет.

**Требования к кадровому обеспечению учебного процесса** При проведении лабораторно-практических занятий в компьютерном классе совместно с преподавателем работает инженер-программист, обеспечивающий работоспособность рабочих станций и программного обеспечения.

### Требования к учебно-методическому обеспечению учебного процесса

Необходимые теоретические сведения и задания проектов к темам курса готовятся в виде раздаточного материала

## Методическое обеспечение программы

Раздел	Формы занятий	Приёмы и методы	Основные умения	Техническое оснащение	Диагностические материалы к контролю
<b>Интерфейс программы Scratch</b>	Учебно-практическая, творческие занятия	Объяснение, упражнение, практическая работа	Познакомиться со средой программирования Scratch. Уметь создавать и редактировать спрайты и фоны для сцен. Выполнять и сохранять проекты.	Компьютер	
<b>Начало работы в среде Scratch</b>	Учебно-практическая, творческие занятия	Объяснение, упражнение, практическая работа	Знать навигацию в среде Scratch, систему координат, команды управления, средства	Компьютер	

			рисования группы перо. Уметь составлять проекты.		
<b>Основные скрипты программы Scratch</b>	Учебно-практическая, творческие занятия	Объяснение, упражнение, практическая работа	Уметь составлять алгоритмы и проекты с циклами.	Компьютер	
<b>Работа с несколькими объектами. Синхронизация их работы</b>	Учебно-практическая, творческие занятия	Объяснение, упражнение, практическая работа	Знать этапы создания проектов. Уметь применять анимацию в проектах.	Компьютер	
<b>Использование программы Scratch для создания мини-игр</b>	Учебно-практическая, творческие занятия	Объяснение, упражнение, практическая работа	Уметь составлять алгоритмы и использовать основные структуры среды.	Компьютер	
<b>Разработка творческого проекта</b>	Учебно-практическая, творческие занятия	Объяснение, упражнение, практическая работа	Уметь создавать проекты и представлять их.	Компьютер	



## Оценка качества освоения образовательной программы

Оценка качества освоения образовательной программы проводится по разработке и реализации проектов различных уровней сложности в соответствии с учебным планом и ожидаемыми результатами.

### Примерные задания к проектным работам обучающихся:

1. Создать игру «Шашки» Классические правила шашек. Можно бить назад. Если делаешь ход, когда мог бить, компьютер спросит у второго игрока: "Разрешить сделать ход или заставить бить?"
2. Построй свой город CITYWORLD. Симулятор города. Тут можно строить дороги, здания, сажать деревья и многое другое.
3. Тетрис: стройка. Классическая стройка. Все варианты фигур, можно их поворачивать. Набираешь очки и бьешь рекорды.
4. Авиа-симулятор 3D
5. Создать игру «2048» Популярная игра для мобильных телефонов. Складывай одинаковые числа и набери 2048 очков, чтобы выиграть.
6. Платформер (Scratchnapped). Игра, похожая на "Марио". Главный герой должен собирать монетки, уворачиваться от врагов и проходить уровень за уровнем.
7. Создание часов — с часовой, минутной и секундной стрелками.
8. Создание мультфильма «Как я хожу в школу» — с рисованием маршрута из дома в школу по карте того населённого пункта, в котором ты живёшь.
9. Презентация «Моя семья».
10. Создание теста с выбором ответа из 3 и более вариантов.
11. Рисование графика  $y = ax + b$  в зависимости от параметров  $a$  и  $b$ .

Педагогическое наблюдение, выполнение практических заданий педагога, анализ на каждом занятии педагогом и обучающимися качества выполнения работ и приобретённых навыков общения, выполнение тестовых заданий, презентация проектов.

В данном курсе предусмотрена итоговая аттестация.

### Формы фиксации результатов

- Вопросы для самопроверки;
- В занимательной форме учащиеся знакомятся с тремя анимационными историями победителей ScratchОлимпиады 2021 года. Ребята пробуют себя в роли членов жюри, авторов сценария, программистов, художников и звукорежиссеров.

Бланки тестовых заданий по темам программы, выполненные в Google- forms.

Формы аттестации: выполнение промежуточных групповых и индивидуальных проектов.

Защита итогового проекта проходит в форме представления обучающимся индивидуального проекта по своему выбору, ответов на вопросы преподавателя. Обсуждения с учащимися достоинств и недостатков проекта, а также проверить уровень сложности любого проекта с помощью специального сайта <http://www.drscratch.org/>. Dr. Scratch анализирует ваши проекты, сообщает количество очков, которое набрал проект и

выдает сертификат качества.

Критерии оценивания итогового проекта: самостоятельность выполнения,

- законченность работы,
- соответствие выбранной тематике,
- оригинальность и качество решения
- проект уникален, и продемонстрировано творческое мышление участников
- проект хорошо продуман и имеет сюжет / концепцию
- сложность
- трудоемкость, многообразие используемых функций
- авторы продемонстрировали свою- компетентность, сумели четко и ясно объяснить, как их проект работает.

### **Методические материалы**

Методы организации учебно-познавательной деятельности, позволяющие повысить эффективность обучения по курсу:

- Объяснительно - иллюстративный (беседа, объяснение, инструктаж, демонстрация, работа с пошаговыми технологическими карточками и др.);
- Метод проблемного изложения (учитель представляет проблему, предлагает ее решение при активном обсуждении и участии обучающихся в решении);
- Эвристический (метод творческого моделирования деятельности).
- Метод проектов.

*Используемые педагогические технологии:* коллективно-творческая деятельность, педагогика сотрудничества, здоровьесберегающая технология, информационно коммуникативная технология, технология проблемного обучения, технология деятельностного подхода, технология проектной деятельности, игровые технологии, дистанционные образовательные технологии.

*Дистанционные образовательные технологии:*

СДО «ДОМ 365»

<https://moodle.ciur.ru/course/view.php?id=4911#section-1>); YouTube«Уроки по Scratch»  
(<https://clck.ru/326iUJ>).

## Учебно-методическое обеспечение программы

### Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы.

#### Литература и материалы для учителя:

1. Сборник «Информатика. Математика. Программы внеурочной деятельности для начальной и основной школы: 3 - 6 классы» / М.С. Цветкова, О.Б.Богомолова. - М.: БИНОМ. Лаборатория знаний, 2013. - 128 с.: ил.
2. Ю.В. Пашковская «Творческие задания в среде Scratch» . - М.: БИНОМ. Лаборатория знаний, 2014. - 200 с.: ил.
3. [Вордерман Кэрол, Вудкок Джон, Макаманус Шон](#) . Переводчик: [Ломакин Станислав](#) Программирование для детей [Манн, Иванов и Фербер](#), 2015 г.

#### Интернет-ресурсы

1. <http://scratch.mit.edu>- официальный сайт Scratch
2. <http://letopisi.ru/index.php/Скретч> - Скретч в Летописи.ру
3. <http://setilab.ru/scratch/category/commun>- Учитесь со Scratch

## Календарный план воспитательной работы

### Характеристика объединения «Увлекательное программирование Scratch»

Деятельность объединения «Увлекательное программирование Scratch» имеет техническую направленность.

Количество обучающихся объединения составляет 10 человек.

Обучающиеся имеют возрастную категорию детей от 10 до 12 лет.

Формы работы - фронтальная, групповая, парная, индивидуальная.

**Цель воспитания:** создание условий для формирования у детей творческого потенциала с использованием возможностей среды программирования Scratch.

#### **Задачи воспитания:**

- ✓ формировать положительное отношение к информатике и ИКТ;
- ✓ развивать самостоятельность и формировать умение работать в паре, малой группе, коллективе;
- ✓ формировать умение демонстрировать результаты своей работы;
- ✓ развивать творческое воображение, фантазию, активность.

**Результат воспитания:** заинтересованность и увлеченность обучающихся в участии в различных мероприятиях, акциях, конкурсах разных уровней.

#### **Работа с одарёнными обучающимися кружка «Увлекательное программирование Scratch»:**

- ✓ ведение банка данных одарённых детей (*таблица 1*);
- ✓ разработка индивидуального образовательного маршрута;
- ✓ диагностика одарённости;
- ✓ мониторинг побед (*таблица 2*)

