

МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«ГИМНАЗИЯ №17» Г. ПЕТУШКИ ПЕТУШИНСКОГО РАЙОНА ВЛАДИМИРСКОЙ
ОБЛАСТИ

Принята на заседании
методического (педагогического) совета
от «___» _____ 20__ г.
Протокол № _____

УТВЕРЖДАЮ:
Директор гимназии
_____ Шмодина С.Ю.
«___» _____ 20__ г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ

ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

Общеинтеллектуальной направленности

«Среда программирования Scratch»

Уровень программы: **базовый**

Возраст учащихся: **11–12 лет**

Срок реализации: **1 год**

Автор:

Хмелева Вера Николаевна,

педагог дополнительного образования

Содержание программы

I. Комплекс основных характеристик программы

1. Пояснительная записка

1.1 Актуальность

1.2 Цель и задачи программы

1.3 Возраст и категории обучающихся

1.4 Уровневость программы

1.5 Формы работы

1.6 Продолжительность реализации

1.7 Планируемые результаты

2. Содержание программы

2.1 Учебный план

2.2 Содержание учебного плана

2.3 Образовательные и учебные форматы

2.4 Формы аттестации и оценочные материалы

II. Комплекс организационно-педагогических условий

1. Материально-техническое обеспечение

2. Учебно-методическое и информационное обеспечение

3. Список литературы

Приложение 1 (календарно-тематическое планирование)

I. Комплекс основных характеристик программы

1. Пояснительная записка

1.1 Актуальность

Сегодня наука и технология развиваются столь стремительно, что образование зачастую не успевает за ними. Одним из самых известных неформальных способов организации внеучебной образовательной деятельности является метод проектов. Самым подходящим инструментом для организации такой деятельности является среда Scratch, которая есть серьезное и современное направление компьютерного дизайна и анимации. Овладев даже минимальным набором операций, самый неискушённый пользователь может создавать законченные проекты. Scratch - это самая новая среда, которая позволяет детям создавать собственные анимированные и интерактивные истории, презентации, модели, игры и другие произведения. Работа в среде Scratch позволяет, с одной стороны, организовать среду для самореализации и самоутверждения учащихся, и, с другой стороны, сформировать у них тягу к творчеству и знаниям и дать подходящие средства её реализации. Быть успешным в такой среде становится проще.

Scratch можно рассматривать как инструмент для творчества, оставив программирование на втором плане. Школьники могут сочинять истории, рисовать и оживлять на экране придуманных ими персонажей, учиться работать с графикой и звуком. Применений возможностям Scratch можно найти множество: в этой среде легко создавать анимированные открытки, мини-игры, мультфильмы. В результате выполнения простых команд может складываться сложная модель, в которой будут взаимодействовать множество объектов, наделенных различными свойствами. Начальный уровень программирования настолько прост и доступен, что Scratch рассматривается в качестве средства обучения не только старших, но и младших школьников.

Когда ученики создают проекты в Scratch, они осваивают множество навыков 21 века, которые будут необходимы для успеха:

- творческое мышление,
- ясное общение,
- системный анализ,
- беглое использование технологий,
- эффективное взаимодействие,
- проектирование,
- умение обучаться и самообучаться,
- самостоятельное принятие решений.

Программа разработана с учетом следующих нормативно-правовых документов

1. Конституция Российской Федерации (принята всенародным голосованием 12.12.1993 с изменениями, одобренными в ходе общероссийского голосования 01.07.2020)
2. Федеральный закон РФ «Об образовании в РФ» (от 29.12.2012 г. №273-ФЗ)
3. Концепция развития дополнительного образования детей (от 14.09.2014 г. №1726-р)
4. Приказ Министерства просвещения Российской Федерации от 09.11.2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
5. Государственная программа Российской Федерации «Развитие образования» (утверждена Постановлением Правительства РФ от 26.12.2017 № 1642 (ред. от 22.02.2021) «Об утверждении государственной программы Российской Федерации «Развитие образования»

6. Стратегия развития воспитания в Российской Федерации на период до 2025 года (утверждена распоряжением Правительства РФ от 29.05.2015 № 996-р «Об утверждении Стратегии развития воспитания в Российской Федерации на период до 2025 года»)
7. Методические рекомендации по созданию и функционированию в общеобразовательных организациях, расположенных в сельской местности и малых городах, центров образования естественно-научной и технологической направленностей («Точка роста») (утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 г. № Р-6)
8. Санитарно-эпидемиологические правила и нормативы СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей» (утв. постановлением Главного государственного санитарного врача РФ от 4 июля 2014 г. № 41).
9. Письмо Министерства образования и науки РФ от 18.10.2015г. №09-3242 «О направлении информации».

1.2 Цель и задачи программы

Целесообразность изучения пропедевтики программирования в игровой, увлекательной форме, используя среду программирования Scratch (5-6 кл) обусловлена следующими факторами:

Во-первых, тем, что в основе Скретч лежит графический язык программирования, который позволяет контролировать действия и взаимодействия между различными типами данных. В среде используется метафора кирпичиков Лего, из которых даже самые маленькие дети могут собрать простейшие конструкции. Но, начав с малого, можно дальше развивать и расширять свое умение строить и программировать.

Во-вторых, существенной ролью изучения программирование и алгоритмизации в развитии мышления, формировании научного мировоззрения школьников именно этой возрастной группы.

В-третьих, в результате изучения данной программы обучающиеся развивают логическое мышление, получают представление об информационных моделях; учатся применять приобретенные знания при решении различных задач и оценивать полученные результаты; овладевают умениями решения учебных задач; развивают интуицию; получают представление об основных информационных процессах в реальных ситуациях.

В-четвертых, занятия по данной программе обеспечат осознание значения информатики в повседневной жизни человека, понимание роли информационных процессов в современном мире.

Цель программы: формирование у школьников базовых представлений о языках программирования, алгоритме, исполнителе, способах записи алгоритма, блок-схемы и программы.

Задачи программы:

Обучающие:

- Обучение основным базовым алгоритмическим конструкциям.
- Освоение основных этапов решения задачи.
- Обучение навыкам разработки, тестирования и отладки несложных программ.
- Обучение навыкам разработки проекта, определения его структуры, дизайна.

Развивающие:

- Развивать познавательный интерес школьников.
- Развивать творческое воображение, математическое и образное мышление учащихся.
- Развивать умение работать с компьютерными программами и дополнительными источниками информации.
- Развивать навыки планирования проекта, умение работать в группе

Воспитывающие:

- Воспитывать интерес к занятиям информатикой.
- Воспитывать культуру общения между учащимися.
- Воспитывать культуру безопасного труда при работе за компьютером.
- Воспитывать культуру работы в глобальной сети.

Программа реализуется на основе следующих принципов:

1. **Обучение в активной познавательной деятельности.** Все темы учащиеся изучают на практике, выполняя различные творческие задания, общаясь в парах и группах друг с другом.
2. **Индивидуальное обучение.** Работа учащихся на компьютере дает возможность организовать их деятельность с индивидуальной скоростью и в индивидуальном объеме.
3. **Принцип природосообразности.** Основной вид деятельности школьников – игра, через нее дети познают окружающий мир, поэтому в занятия включены игровые элементы, способствующие успешному освоению программы.
4. **Преемственность.** Программа построена так, что каждая последующая тема логически связана с предыдущей. Данный принцип учащимся помогает понять важность уже изученного материала и значимость каждого отдельного занятия.
5. **Целостность и непрерывность,** означают что данная ступень является важным звеном единой общешкольной подготовки по информатике и информационным технологиям.
6. **Практико-ориентированность,** обеспечивающая отбор содержания, направленного на решение простейших практических задач планирования деятельности, поиска нужной информации, инструментирования всех видов деятельности на базе общепринятых средств информационной деятельности, реализующих основные пользовательские возможности информационных технологий. При этом исходным является положение о том, что компьютер может многократно усилить возможности человека, но не заменить его.
7. **Принцип дидактической спирали** как важнейший фактор структуризации в методике обучения информатике: вначале общее знакомство с понятием с учетом имеющегося опыта обучаемых, затем его последующее развитие и обогащение, создающее предпосылки для научного обобщения в старших классах.
8. **Принцип развивающего обучения** - обучение ориентировано не только на получение новых знаний в области информатики и информационных технологий, но и на активизацию мыслительных процессов, формирование и развитие у школьников обобщенных способов деятельности, формирование навыков самостоятельной работы.

1.3 Возраст и категории обучающихся

Адресат программы: программа рассчитана на детей 11-12 лет по программе дополнительного образования в рамках работы центра «Точки роста»;

1.4 Уровневость программы

Уровень - базовый

1.5 Формы работы

Формы и методы обучения определены возрастом учащихся: групповая и индивидуальная работа, теоретическая работа чередуется с практической, использование интерактивных форм обучения и др.

При проведении занятий используются компьютеры с установленной программой Scratch, проектор, сканер, принтер, компьютерная сеть с выходом в Интернет.

Формы проведения занятий: беседы, игры, практические занятия, самостоятельная работа, викторины и проекты.

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.

Режим (периодичность и продолжительность занятий):

1 раза в неделю по 1 академическому часу или 1 раз в две недели по 2 академических часа с перерывом между занятиями 10 минут.

1.6 Продолжительность реализации

Продолжительность реализации программы:

1 год обучения 34 часа.

1.7 Планируемые результаты

Результаты обучения.

По окончании обучения учащиеся будут:

знать:

- термины «информация», «сообщение», «данные», «алгоритм», «программа»; понимать различия между употреблением этих терминов в обыденной речи и в информатике;
- основные способы представления и анализа данных;
- алгоритмический язык, основные алгоритмические структуры - линейную, условную и циклическую, логические значения и операции;
- отдельные способы планирования деятельности: составление плана предстоящего проекта в виде рисунка, схемы, таблицы объектов (их свойств и взаимодействий); разбиение задачи на подзадачи;

уметь:

- грамотно применять символику, использовать язык программирования СКРЕТЧ для записи программ;
- использовать язык программирования для описания действий объектов программирования, развитие пространственных представлений и изобразительных умений; овладение навыками беглого использования технологий;
- применять изученные понятия, результаты, методы для решения задач практического характера и задач из смежных дисциплин с использованием при необходимости справочных материалов, компьютера.
- использовать основные и дополнительные компьютерные устройства
- подготовить небольшой отчет о работе, публично выступить с докладом;
- наметить дальнейшие пути развития проекта
- иметь представление об объектно-ориентированном программировании.

Результаты воспитывающей деятельности.

Выполнение программы дает возможность обучающимся достичь следующих результатов развития в личностном направлении:

- умение ясно, точно, грамотно излагать свои мысли в устной и письменной речи, понимать смысл поставленной задачи;
- умение распознавать логически некорректные высказывания, отличать гипотезу от факта;
- развивает креативность мышления, инициативу, находчивость, активность, применение знаний по информатике для решения конкретных жизненных задач;

Результаты развивающей деятельности.

Обучающиеся смогут:

- ставить учебные цели, планировать свои действия в соответствии с поставленной задачей и условиями её решения
- находить в различных источниках информацию, необходимую для решения проблем;
- осуществлять итоговый и пошаговый контроль, сличая результат с эталоном, вносить коррективы в действия в случае расхождения результата решения задачи и ранее поставленной целью.
- понимать и использовать различные средства наглядности (графики, диаграммы, таблицы, схемы и др.);
- применять умение планировать и осуществлять деятельность, направленную на решение задач исследовательского характера;

2. Содержание программы

2.1 Учебный план

№ п/п	Название раздела, темы	Количество часов			Формы аттестации (контроля)
		всего	теория	практика	
1.	Интерфейс программы Scratch (1 ч)				
1.1.	Введение. Что такое Scratch. Основные алгоритмические конструкции. Знакомство с интерфейсом программы Scratch.	1	0	1	собеседование
2.	Начало работы в среде Scratch (2 ч)				Проверка практических умений (контроль за правильностью, полнотой и последовательностью выполнения операций)
2.1.	Сцена. Редактирование фона. Добавление фона из файла.	0,5	0,5	1	
2.2.	Понятие спрайтов. Добавление новых спрайтов. Рисование новых объектов.	0,5	0,5	1	
III	Основные скрипты программы Scratch (18 ч)				Проверка практических умений (контроль за правильностью, полнотой и последовательностью)

					выполнения операций)
3.1	Синий ящик – команды движения. Темно-зеленый ящик – команды рисования.	0,5	1,5	2	
3.2	Фиолетовый ящик – внешний вид объекта. Оживление объекта с помощью добавления костюмов.	0,5	1,5	2	
3.3	Желтый ящик – контроль. Лиловый ящик – добавление звуков.	0,5	1,5	2	
3.4	Использование в программах условных операторов.	0,5	1,5	2	
3.5	Функциональность работы циклов. Цикличность выполнения действий в зависимости от поставленных условий.	0,5	1,5	2	
3.6	Зеленый ящик – операторы. Использование арифметических и логических блоков вместе с блоками управления.	0,5	1,5	2	
3.7	События. Оранжевый ящик – переменные.	0,5	1,5	2	
3.8	Списки.	0,5	1,5	2	
3.9	Голубой ящик – сенсоры. Ввод-вывод данных.	0,5	1,5	2	
IV	Работа с несколькими объектами. Синхронизация их				Проверка практических умений (контроль за правильностью,

	работы (4 ч)				полнотой и последовательностью выполнения операций)
4.1	Последовательность и параллельность выполнения скриптов.	0,5	1,5	2	
4.2	Взаимодействие между спрайтами. Управление через обмен сообщениями.	1	1	2	
V	Использование программы Scratch для создания мини-игр (7 ч)				Проверка практических умений (контроль за правильностью, полнотой и последовательностью выполнения операций)
	Виды компьютерных игр. Алгоритмическая разработка листинга программы.	1	1	2	прогностический, то есть проигрывание всех операций учебного действия до начала его реального выполнения
	Разработка базовых спрайтов для игры. Формирование базовых скриптов.	0,5	1,5	2	пооперационный, то есть контроль за правильностью, полнотой и последовательностью выполнения операций
	Синхронизация работы скриптов для разных спрайтов.	0	1	1	пооперационный, то есть контроль за правильностью, полнотой и последовательностью выполнения операций
	Переход из одной сцены в другую. Создание интерфейса игры.	0,5	0,5	1	пооперационный, то есть контроль за правильностью,

					полнотой и последовательностью выполнения операций
	Сообщество Scratch в Интернете. Просмотр и публикация проектов.	0,5	0,5	1	
VI.	Разработка творческого проекта (2 ч)				
	Разработка и защита творческого проекта	0	2	2	итоговый контроль в формах: практические работы, творческие работы учащихся, создание проекта
	Итого	11	23	34	

2.2 Содержание учебного плана

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. Стил 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; повернуться направо (налево); повернуть в направлении; повернуться 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. Заикливание (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 часа).

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 (2 ч)

20. Разработка и защита творческого проекта. Разработка и создание программы с использованием подготовленных материалов. Тестирование и отладка проекта. Защита проекта (2 часа).

2.3 Образовательные и учебные форматы

2.4 Формы аттестации и оценочные материалы

Формы аттестации

В процессе занятий применяются словесные, наглядные и практические методы, причем две трети времени отводится практической форме работы.

В процессе реализации программы применяются различные средства: таблицы, схемы, фрагменты видеофильмов, компьютерные мультимедийные средства и программы.

Виды контроля:

- *предварительный контроль* (проверка знаний учащихся на начальном этапе освоения Программы). Проводится в начале реализации Программы в виде входного собеседования.

- *текущий контроль* (отслеживание активности обучающихся в выполнении ими творческих работ.)

- *итоговый контроль* (заключительная проверка знаний, умений, навыков по итогам реализации Программы в каждом учебном году). Презентация подготовленных учащимися творческих работ.

Формы и содержание итоговой аттестации:

- тестирование;
- презентация творческой работы (проекта).

Требование к оценке творческой работы

Творческая работа (проект) оценивается положительно при условии, если:

- определена и четко сформулирована цель работы;
- характеризуется оригинальностью идей, исследовательским подходом, подобранным и проанализированным материалом;
- содержание работы изложено логично;
- прослеживается творческий подход к решению проблемы, имеются собственные предложения;
- сделанные выводы свидетельствуют о самостоятельности ее выполнения.

Форма защиты творческой работы (проекта) – очная презентация.

Оценочные материалы

тесты, творческие задания, контрольные работы, вопросы

II. Комплекс организационно-педагогических условий

1. Материально-техническое обеспечение программы

Компьютерное, мультимедийное оборудование, пакет обучающих программ

2. Учебно-методическое и информационное обеспечение

1. УМК «Творческие задания в Scratch»
2. Оф (или онлайн-) тесты комплекта Творческие задания в Scratch
3. Интернет - ресурс: <http://scratch.mit.edu>

3. Список литературы

1. Патаракин Е. Д. Учимся готовить в среде Скретч (Учебно-методическое пособие). М: Интуит.ру, 2008. 61 с.
2. Скретч [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вернуться домой». URL: <http://letopisi.ru/index.php/Скретч>
3. Школа Scratch [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вернуться домой». URL: http://letopisi.ru/index.php/Школа_Scratch
4. Scratch | Home | imagine, program, share [сайт]. URL: <http://scratch.mit.edu>
5. Scratch | Галерея | Gymnasium №3 [сайт]. URL: <http://scratch.mit.edu/galleries/view/54042>.

Календарный учебный график

№ п/п	Месяц	Число	Время проведения занятия	Форма занятия	Количество часов	Тема занятия	Место проведения	Форма контроля
1	сентябрь		15.00-15.45		1	Инструктаж по техники безопасности и правилам работы в компьютерном классе. Введение. Что такое Scratch. Основные алгоритмические конструкции. Знакомство с интерфейсом программы Scratch.	Кабинет информатики	фронтальная
2	сентябрь		15.00-15.45		1	Сцена. Редактирование фона. Добавление фона из файла. Создание фона сцены на выбранную учащимся тему	Кабинет информатики	фронтальная
3	сентябрь		15.00-15.45		1	Понятие спрайтов. Добавление новых спрайтов. Рисование новых объектов.	Кабинет информатики	фронтальная
4	октябрь		15.00-15.45		1	Управление спрайтами. Синий ящик – команды движения. Темно-зеленый ящик – команды рисования	Кабинет информатики	комбинированная
5	октябрь		15.00-15.45		1	Координатная плоскость. Создание программ для передвижения спрайтов по сцене. Создание программ для рисования	Кабинет информатики	комбинированная

						различных фигур		
6	октябрь		15.00-15.45		1	Навигация в среде Scratch. Создание проекта. Задание команд. Режим презентации	Кабинет информатики	комбинированная
7	октябрь		15.00-15.45		1	Цикл. Рисование узоров и орнаментов	Кабинет информатики	комбинированная
8	ноябрь		15.00-15.45		1	Конструкция «Всегда». Ориентация по компасу	Кабинет информатики	комбинированная
9	ноябрь		15.00-15.45		1	Спрайты меняют костюмы. Фиолетовый ящик – внешний вид объекта. Оживление объекта с помощью добавления костюмов	Кабинет информатики	комбинированная
10	ноябрь		15.00-15.45		2	Создание программ с элементами управления объектом. Озвучивание Scratch-историй. Создание мультипликационного сюжета	Кабинет информатики	комбинированная
11	ноябрь		15.00-15.45		1	Голубой ящик – сенсоры. Ввод-вывод данных. Создание проектов с использованием значений сенсоров и команды <i>спросить</i> . Создание программ для обработки данных пользователя с выводом на экран конечного результат	Кабинет информатики	комбинированная
12	декабрь		15.00-15.45		1	Использование в программах условных операторов. Сенсоры. Блок «Если». Управляемый спрайт.	Кабинет информатики	комбинированная

13	декабрь		15.00-15.45		1	Создание программ с изменением последовательного выполнения скриптов при наличии условий. Создание коллекции игр	Кабинет информатики	комбинированная
14	декабрь		15.00-15.45		1	Составные условия Пополнение коллекции игр	Кабинет информатики	комбинированная
15	декабрь		15.00-15.45		1	Функциональность работы циклов. Цикличность выполнения действий в зависимости от поставленных условий	Кабинет информатики	комбинированная
16	январь		15.00-15.45		1	Создание программ с использованием циклов с фиксированным числом повторений. Создание программ с использованием циклов с предусловием и постусловием	Кабинет информатики	комбинированная
17	январь		15.00-15.45		1	Датчик случайных чисел	Кабинет информатики	комбинированная
18	январь		15.00-15.45		1	Зеленый ящик – операторы. Использование арифметических и логических блоков вместе с блоками управления. Самоуправление спрайтов	Кабинет информатики	комбинированная
19	январь		15.00-15.45		1	Создание программ с использованием операций сравнения данных. Создание программ с использованием арифметических данных и логических операций. Доработка проектов с учетом изученного	Кабинет информатики	комбинированная

20	февраль		15.00-15.45		1	Датчики	Кабинет информатики	комбинированная
21	февраль		15.00-15.45		1	События. Оранжевый ящик – переменные.	Кабинет информатики	комбинированная
22	февраль		15.00-15.45		1	Ввод переменных с помощью рычажка. Разработка сценария Scratch-историй с несколькими событиями. Создание проектов с использованием глобальных и локальных переменных	Кабинет информатики	комбинированная
23	февраль		15.00-15.45		1	Списки	Кабинет информатики	комбинированная
24	март		15.00-15.45		1	Создание программ-тестов по принципу сравнения данных из нескольких списков	Кабинет информатики	комбинированная
25	март		15.00-15.45		1	Создание тестов	Кабинет информатики	комбинированная
26	март		15.00-15.45		1	Строковые величины. Создание игры на основе строковых величин	Кабинет информатики	комбинированная
27	март		15.00-15.45		1	Создание Scratch-историй с взаимодействием нескольких исполнителей и неподвижных объектов. Создание Scratch-историй с взаимодействием нескольких исполнителей	Кабинет информатики	комбинированная
28	апрель				1	Виды компьютерных игр.		

						Алгоритмическая разработка листинга программы.		
29	апрель		15.00-15.45		1	Алгоритмическая разработка проекта, запись на естественном языке событий и точек взаимодействия героев будущей игры	Кабинет информатики	комбинированная
30	апрель		15.00-15.45		1	Разработка и создание основных спрайтов и их костюмов для будущей игры. Разработка скриптов для спрайтов и объектов	Кабинет информатики	комбинированная
31	апрель		15.00-15.45		1	Создать программу для перемещения объекта по игровой карте и разработать интерфейс для Scratch-проекта	Кабинет информатики	комбинированная
32	май		15.00-15.45		1	Сообщество Scratch в Интернете. Просмотр и публикация проектов	Кабинет информатики	комбинированная
33	май		15.00-15.45		1	Разработка и защита творческого проекта. Разработка и создание программы с использованием подготовленных материалов. Тестирование и отладка проекта	Кабинет информатики	комбинированная
34	май		15.00-15.45		1	Защита творческих проектов	Кабинет информатики	комбинированная