

РАБОЧАЯ ПРОГРАММА
Курса внеурочной деятельности
Программирование на языке Scratch

Класс *6 класс*

Количество часов *34 часа*

1. Планируемые результаты освоения курса внеурочной деятельности

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 «Программирование в среде 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;
- 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.

Предметными результатами изучения курса «Программирование в среде Scratch» в 6-м классе являются формирование следующих **знаний и умений**:

УЧАЩИЕСЯ	
ДОЛЖНЫ ЗНАТЬ	ДОЛЖНЫ УМЕТЬ
Что такое Scratch и его назначение. Основные базовые алгоритмические конструкции. Исполнитель и его система команд. Самодостаточные и открытые скрипты. Принцип создания анимации и движения объектов.	Размещать объекты на сцене. Поворачивать их и масштабировать.
Сцена. Текущие координаты объекта. Спрайт. Интерфейс программы Scratch.	Вставлять стандартный фон из библиотечного модуля среды. Рисовать фон в графическом редакторе. Добавлять фон из файла. Создавать

	спрайты с помощью графического редактора средыScratch. Загружать на сцену спрайты из стандартной коллекции Scratch. Вставлять спрайты из файлов. Центрировать костюм. Масштабировать спрайт. Удалять спрайты.
Команды из ящиков движения, внешности, звука, рисования, контроля, сенсоров, операторов и переменных. События в проектахScratch	Создавать программы для движения спрайтов по сцене, для рисования различных фигур, имитации естественного движения героев в различных направлениях. Озвучивать как полностью проект, так и отдельные события внутри проекта. Создавать программы - с изменением последовательного выполнения скриптов при наличии условий, с использованием циклов с фиксированным числом повторений, с предусловием и постусловием. Использовать в программах операции сравнения данных, арифметические и логические действия над данными, сравнение данных из нескольких списков, глобальные и локальные переменные. Обрабатывать данные с выводом на экран конечного результата
Принцип взаимодействия спрайтов через обмен сообщениями. Последовательные и параллельные потоки в программах Scratch	Создавать Scratch-истории с взаимодействием нескольких исполнителей и неподвижных объектов, а так же с одновременной и попеременной работой нескольких исполнителей.
Виды компьютерных игр. Этапы создания компьютерных игр. Интерфейс игры. Адрес сообществаScratch в Интернете. Авторские права.	Поэтапно создавать компьютерную игру. Создавать программу для перемещения объектапо игровой карте в одном направлении и в пространстве из нескольких связанных между собой комнат. Разрабатывать интерфейс для Scratch проекта. Регистрироваться на сайте сообществаScratch. Просматривать проекты сообщества и публиковать собственные проекты.

2. Содержание курса внеурочной деятельности

I. Интерфейс программы Scratch

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 средеScratch. 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.

II. Начало работы в среде Scratch

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.

Практика. Создание фона сцены на выбранную учащимся тему.

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.

Практика. Создание фона сцены и прорисовка основных спрайтов для Scratch-истории.

III. Основные скрипты программы Scratch.

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* *на 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*.

Практика. Создание программ для передвижения спрайтов по сцене. Создание программ для рисования различных фигур.

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.

Практика. Создание программы для управления внешним видом объекта. Создание Scratch-историй с имитацией хождения и движения объектов.

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*.

Практика. Создание программ с элементами управления объектом. Озвучивание Scratch-историй.

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.

Практика. Создание программ с изменением последовательного выполнения скриптов при наличии условий.

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.

Практика. Создание программ с использованием циклов с фиксированным числом повторений. Создание программ с использованием циклов с предусловием и постусловием.

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.

Практика. Создание программ с использованием операций сравнения данных. Создание программ с использованием арифметических данных и логических операций.

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.

Практика. Разработка сценария Scratch-историй с несколькими событиями. Создание проектов с использованием глобальных и локальных переменных.

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*.

Практика. Создание программ-тестов по принципу сравнения данных из нескольких списков.

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...иждать*. 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*.

Практика. Создание проектов с использованием значений сенсоров и команды *спросить*. Создание программ для обработки данных пользователя с выводом на экран конечного результата.

IV. Работа с несколькими объектами. Синхронизация их работы.

13. Последовательность и параллельность выполнения скриптов.

Теория. Последовательные и параллельные потоки в программах Scratch. Одновременная и попеременная работа нескольких исполнителей.

Практика. Создание Scratch-историй с одновременной и попеременной работой нескольких исполнителей.

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.

Практика. Создание Scratch-историй с взаимодействием нескольких исполнителей и неподвижных объектов. Создание Scratch-историй с взаимодействием нескольких исполнителей.

V. Использование программы Scratch для создания мини-игр.

15. Виды компьютерных игр. Алгоритмическая разработка листинга программы.

Теория. Компьютерные игры – вред или польза. Виды компьютерных игр. Этапы разработки игр программистами.

Практика. Алгоритмическая разработка проекта, запись на естественном языке событий и точек взаимодействия героев будущей игры.

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

Теория. Логика создания персонажей для игры. Перевод алгоритма, написанного на естественном языке, в коды Scratch.

Практика. Разработка и создание основных спрайтов и их костюмов для будущей игры. Разработка скриптов для спрайтов и объектов.

17. Синхронизация работы скриптов для разных спрайтов.

Практика. Доработка основного листинга программы с целью установления связей между спрайтами. Тестирование и отладка программы.

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.

Практика. Создать программу для перемещения объекта по игровой карте и

разработать интерфейс для Scratch-проекта.

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.

Практика. Регистрация на сайте сообщества Scratch. Просмотр проектов сообщества и публикация собственных проектов.

VI. Разработка творческого проекта.

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

3. Тематическое планирование

№ п/п	НАИМЕНОВАНИЕ ТЕМЫ	Т	О	Р	И	А	К	Д	С
1.	Интерфейс программы Scratch		1			0			1
2.	Начало работы в среде Scratch		1			1			2
3.	Основные скрипты программы Scratch		5			13			18
4.	Работа с несколькими объектами. Синхронизация их работы		2			2			4
5.	Использование программы Scratch для создания мини-игр		3			4			7
6.	Разработка творческого проекта		0			2			2
Итого:			12			22			34

Аннотация

Содержание курса «Программирование в среде Scratch» для учащихся 6 классов рассчитано на обучение в объеме 34 учебных часа (1 час в неделю).

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 неискушенный 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.

Изучение Scratch может серьезно помочь школьникам освоить азы алгоритмизации и программирования, а полученные знания пригодятся для дальнейшего и более серьезного изучения программирования. Работа в среде Scratch ведется так же как средство подготовки учащихся к всевозможным конкурсам и выставкам по данной тематике, которые в настоящее время набирают большие обороты.

Изучение данного курса направлено на достижение *следующих целей:*

- ✓ формирование общеучебных умений и навыков на основе средств и методов информатики и ИКТ, в том числе овладение умениями работать с различными видами информации, самостоятельно планировать и осуществлять индивидуальную и коллективную информационную деятельность, представлять и оценивать ее результаты;
- ✓ пропедевтическое изучение понятий основного курса школьной информатики;
- ✓ воспитание ответственного и избирательного отношения к информации; развитие познавательных, интеллектуальных и творческих способностей учащихся,
- ✓ развитие способности к решению творческих задач,
- ✓ развитие самостоятельности.

Ключевым в программе является понятие «проектная научно-познавательная

деятельность школьника», как совместная (с другими субъектами), так и самостоятельная деятельность с использованием методов научного исследования, ведущим мотивом которой является познавательный интерес. Проектная научно-познавательная деятельность не является самоцелью, но рассматривается как среда, в которой наиболее естественным образом раскрывается личностный потенциал школьника.

В этой связи *целями проектной научно-познавательной деятельности школьника* являются:

- ✓ развитие интеллектуальных, познавательных и творческих способностей школьника;
- ✓ развитие метапредметных умений (личностных, познавательных, коммуникативных, регулятивных);
- ✓ развитие способов мыслительной деятельности;
- ✓ формирование целостной картины мира и системного мышления на основе межпредметных связей.

Следует иметь в виду, что возрастные особенности младшего школьника не позволяют в полной мере реализовать проведение полноценных научных исследований. В то же время раннее включение в организованную специальным образом проектную деятельность творческого характера позволяет сформировать у школьника познавательный интерес и исследовательские навыки, которые в старшем возрасте пригодятся им для выполнения научно-познавательных проектов.

Основная задача курса – способствовать формированию у школьников информационной и функциональной компетентности, развитие алгоритмического мышления. Назначение курса – помочь детям узнать основные возможности компьютера и научиться им пользоваться в повседневной жизни. А так же:

- ✓ дать учащимся представление о современном подходе к изучению реального мира, о широком использовании алгоритмов и вычислительной техники в научных исследованиях;
- ✓ сформировать у учащихся умения владеть компьютером как средством решения практических задач;
- ✓ подготовка учеников к активной полноценной жизни и работе в условиях технологически развитого общества;
- ✓ создание условий для внедрения новых информационных технологий в учебно-воспитательный процесс школы;
- ✓ реализовать в наиболее полной мере возрастающий интерес учащихся к углубленному изучению программирования через совершенствование их алгоритмического и логического мышления;
- ✓ формирование знаний о роли информационных процессов в живой природе, технике, обществе;
- ✓ формирование знаний о значении информатики и вычислительной техники в развитии общества и в изменении характера труда человека;
- ✓ формирование знаний об основных принципах работы компьютера, способах передачи информации;
- ✓ формирование знаний об основных этапах информационной технологии решения задач в широком смысле;
- ✓ формирование умений моделирования и применения его в разных предметных областях;
- ✓ формирование умений и навыков самостоятельного использования компьютера в качестве средства для решения практических задач.

Реализация этих задач будет способствовать дальнейшему формированию взгляда школьников на мир, раскрытию роли информатики в формировании естественнонаучной картины мира, развитию мышления, в том числе формированию алгоритмического стиля мышления, подготовке учеников к жизни в информационном обществе.

Требования к подготовке учащихся

Учащиеся должны:

знать:

- ✓ отдельные способы планирования деятельности;
- ✓ составление плана предстоящего проекта в виде рисунка, схемы;
- ✓ составление плана предстоящего проекта в виде таблицы объектов, их свойств и взаимодействий;
- ✓ разбиение задачи на подзадачи;
- ✓ о требованиях к организации компьютерного рабочего места, соблюдать требования безопасности и гигиены в работе со средствами ИКТ;
- ✓ устройства компьютера;
- ✓ примеры информационных носителей.

уметь:

- ✓ приводить простые жизненные примеры передачи, хранения и обработки информации в деятельности человека, в живой природе, обществе, технике;
- ✓ различать программное и аппаратное обеспечение компьютера;
- ✓ запускать программы из меню Пуск;
- ✓ изменять размеры и перемещать окна, реагировать на диалоговые окна;
- ✓ вводить информацию в компьютер с помощью клавиатуры и мыши;
- ✓ применять текстовый редактор для набора, редактирования и форматирования простейших текстов;
- ✓ иметь представление о способах кодирования информации;
- ✓ применять простейший графический редактор для создания и редактирования рисунков;
- ✓ выполнять вычисления с помощью приложения Калькулятор;
- ✓ составлять план проекта, включая:
 - выбор темы;
 - анализ предметной области;
 - разбиение задачи на подзадачи;
- ✓ проанализировать результат и сделать выводы;
- ✓ найти и исправить ошибки;
- ✓ подготовить небольшой отчет о работе;
- ✓ публично выступить с докладом;
- ✓ наметить дальнейшие пути развития проекта.

Scratch позволяет создавать проекты, которые интересны различным возрастным группам:

1. музыкальный проект;
2. анимация;
3. комикс;
4. интерактивная игра;
5. графика;
6. учебная динамичная и интерактивная презентация;
7. учебная модель, демонстрационный эксперимент;
8. обучающая программа;
9. учебный интерактивный тест;
10. социальная реклама (направлена на изменение моделей общественного поведения и привлечения внимания к проблемам экологической культуры, культуры здорового и безопасного образа жизни, милосердия...).

МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА

Оборудование и материалы

Для работы с учебным комплектом необходимо следующее:

- программное обеспечение: Scratch, Paint;
- аппаратное обеспечение: компьютерный класс (10 компьютеров для воспитанников и 1 для педагога), интерактивная система.