

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Дополнительная общеобразовательная общеразвивающая программа «Креативное программирование в Scratch»» (далее – Программа) имеет технологическую направленность.

* Программа составлена на основе;
* методических рекомендаций утвержденных распоряжением Министерства просвещения РФ от 12.01.2021 №Р-6 «Об утверждении методических рекомендаций по созданию и функционированию в общеобразовательных организациях, расположенных в сельской местности и малых городах, центров образования естественно-научной и технологической направленностей».
* программы учебного курса «Проекты на основе ИКТ» (Цветкова М.С., Богомолова О.Б. «Информатика. Математика. Программы внеурочной деятельности для начальной и основной школы: 3-6 классы» - М.: Бином, 2015.).Методических рекомендаций. Рабочая программа учебного предмета: Методическое пособие.-Смоленск: ГАУДПО СОИРО, 2019

Программа реализуется с использованием оборудования Центра образования естественно-научной и технологической направленностей "Точка роста".

Актуальность

Мультимедийная среда Scratch позволяет сформировать у детей стойкий интерес к программированию, отвечает всем современным требованиям объектно-ориентированного программирования. Среда Scratch позволяет сформировать навыки программирования, раскрыть технологию программирования.

Название «Scratch» в переводе с английского имеет несколько значений. Это и царапина, которую оставляет Котенок - символ программы, и каракули, символизирующие первый, еще неуклюжий самостоятельный опыт, и линия старта. Со Scratch удобно стартовать. Сами разработчики характеризуют программу так: «Scratch предлагает низкий пол (легко начинать), высокий потолок (возможность создавать сложные проекты) и широкие стены (поддержка большого многообразия проектов)».

Подобно тому, как дети только-только начинающие говорить, учатся складывать из отдельных слов фразы, и Scratch обучает из отдельных кирпичиков-команд собирать целые программы.

В Scratch можно сочинять истории, рисовать и оживлять на экране придуманные персонажи, создавать презентации, игры, в том числе и интерактивные, исследовать параметрические зависимости.

Любой персонаж в среде Scratch может выполнять параллельно несколько действий – двигаться, поворачиваться, изменять цвет, форму и.т.д.; благодаря чему юные скретчисты учатся осмысливать любое сложное действие как совокупность простых. В результате они не только осваивают базовые концепции программирования (циклы, ветвления, логические операторы, случайные числа, переменные, массивы), которые пригодятся им при изучении более сложных языков, но и знакомятся с полным циклом решения задач, начиная с этапа описания идеи и заканчивая тестированием и отладкой программы.

Scratch легко перекидывает мостик между программированием и другими школьными науками. Так возникают межпредметные проекты. Они помогут сделать наглядными понятия отрицательных чисел и координат, уравнения плоских фигур, изучаемых на уроках математики. В них оживут исторические события и географические карты. А тесты по любым предметам сделают процесс обучения веселым и азартным.

Scratch – свободно распространяемая программа. Она одинаково хорошо устанавливается и в Windows, и в Ubuntu, и в Macintosh.

Scratch создали американцы Митч Резник и Алан Кей. На русский язык программа переведена доцентом Нижегородского университета Евгением Патаракиным.

*Некоторые отличительные особенности программы:*

1. *Проектный подход.* В процессе обучения происходит воспитание культуры проектной деятельности, раскрываются и осваиваются основные шаги по разработке и созданию проекта.

2. *Межпредметность.* В программе прослеживается тесная взаимосвязь с математикой, физикой, географией, русским языком, музыкой и другими предметами школьного цикла. Знания, полученные на других предметах, логичным образом могут быть использованы при разработке проектов.

3. *Пропедевтика.* Через разработку проектов учащиеся получают знания, обозначенные в программах старших классов. Так, например, осваиваются основные алгоритмические конструкции (информатика), понятие координатной плоскости (математика) и т.п.

4. *Вариативность.* Учащиеся с достаточной степенью свободы и самостоятельности могут выбирать темы проектов.

5. *Коммуникация.* Программой предусмотрена работа в командах, парах, использование возможностей сетевого сообщества для взаимодействия. Обязательное условие - публичная презентация и защита проектов.

Программа построена таким образом, чтобы помочь учащимся заинтересоваться программированием вообще и найти ответы на вопросы, с которыми им приходится сталкиваться в повседневной жизни при работе с большим объемом информации; при решении практических и жизненных задач. Программа строится на использовании среды Scratch при обучении детей, что позволяет создавать собственные программы для решения конкретной задачи. Это является **отличительной особенностью** данной программы.

**Новизна программы** заключается в том, что Scratch 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 по программированию практически значимой для современного учащегося, т.к. дает возможность увидеть практическое назначение алгоритмов и программ, что будет способствовать развитию интереса к профессиям, связанным с программированием.

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

**Цель программы** - воспитание творческой личности, обогащенной общетехническими знаниями и умениями, развитие индивидуальных творческих способностей, интереса к науке и технике.

**Задачи программы**:

**Обучающие:**

* овладеть навыками составления алгоритмов;
* изучить функциональность работы основных алгоритмических конструкций;
* сформировать представление о профессии «программист»;
* сформировать навыки разработки программ;
* 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.

**Принципы обучения**, реализуемые программой:

* сознательности;
* наглядности;
* доступности;
* связи теории с практикой;
* творческой активности.

Важным условием развития творческого и познавательного интереса учащегося является индивидуальный подход к нему в процессе обучения.

**Организация образовательного процесса**

**Срок реализации** Программы «Креативное программирование в Scratch» - **1 год**.

Рекомендуемый **возраст детей**: **11-13 лет.**

**Наполняемость группы от 6 человек.**

На программу ***1 года обучения*** отводится **72 часа.**

**Режим занятий**:- 1 раз в неделю по 2 часа.

В данной программе используется индивидуальная, групповая и фронтальная формы работы.

Содержание практических занятий ориентировано не только на овладение учащимися навыками программирования, но и на подготовку их как грамотных пользователей ПК; формированию навыков участия в дистанционных конкурсах и олимпиадах, умений успешно использовать навыки сетевого взаимодействия.

**Планируемые результаты освоения программы**

В результате изучения курса получат дальнейшее развитие личностные, регулятивные, коммуникативные и познавательные универсальные учебные действия, учебная (общая и предметная) и общепользовательская ИКТ-компетентность обучающихся.

В основном формируются и получат развитие

**метапредметные результаты**, такие как:

* умение самостоятельно планировать пути достижения целей, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;
* умения соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата;
* умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;
* умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
* владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
* умение организовывать учебное сотрудничество совместную деятельность с учителем и сверстниками;
* формирование и развитие компетентности в области использования информационно-коммуникационных технологий (далее ИКТ-компетентности).

Вместе с тем вносится существенный вклад в развитие

**личностных результатов**, таких как:

* формирование ответственного отношения к учению;
* формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, творческой и других видов деятельности.

В части развития

**предметных результатов** наибольшее влияние изучение курса оказывает:

* на формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей с использованием соответствующих программных средств обработки данных;
* формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете, умение соблюдать нормы информационной этики и права.

**Регулятивные универсальные учебные действия**

Обучающийся научится:

* целеполаганию, включая постановку новых целей, преобразование практической задачи в познавательную;
* самостоятельно анализировать условия достижения цели на основе учета выделенных учителем ориентиров действия в новом учебном материале;
* планировать пути достижения целей;
* уметь самостоятельно контролировать свое время и управлять им.

**Коммуникативные универсальные учебные действия**

Обучающийся научится:

* устанавливать и сравнивать разные точки зрения, прежде чем принимать решения и делать выбор;
* аргументировать свою точку зрения, спорить и отстаивать свою позицию не враждебным для оппонентов образом;
* задавать вопросы, необходимые для организации собственной деятельности и сотрудничества с партнером;
* осуществлять взаимный контроль и оказывать в сотрудничестве необходимую взаимопомощь.

**Познавательные универсальные учебные действия**

Обучающийся научится:

* создавать и преобразовывать модели и схемы для решения задачи;
* осуществлять выбор наиболее эффективных способов решения задач в зависимости от конкретных условий.

**Проверка результативности**

Знания, умения, навыки, полученные на занятиях, необходимо подвергать педагогическому контролю, с целью выявления качества усвоенных детьми знаний в рамках программы обучения.

Формами педагогического контроля могут быть: итоговые занятия один раз в полугодие, контрольные задания, тематические выставки, устный опрос, тестирование, которые способствуют поддержанию интереса к работе, направляют учащихся к достижению более высоких вершин творчества.

**Аттестация учащихся:**

Оценка теоретических знаний и практических умений и навыков учащихся по теории и практике проходит по трем уровням: **высокий, средний, низкий.**

**Высокий уровень –** учащиеся должны знать правила техники безопасности при работе, грамотно излагать программный материал, знать основные блоки команд, уметь выполнять алгоритмы, описанные с использованием конструкций ветвления и повторения и уметь самостоятельно создавать и выполнять программы для решения алгоритмических задач в программе Scratch.

**Средний уровень –** учащиеся должны знать основные блоки команд, уметь выполнять алгоритмы, описанные с использованием конструкций ветвления и повторения, грамотно и по существу излагать программный материал, не допуская существенных неточностей в ответе.

**Низкий уровень –** учащиеся не знают значительной части материала, допускают существенные ошибки, с большими затруднениями выполняют практические задания.

При обработке результатов учитываются **критерии** для выставления уровней:

**Высокий уровень** *–* выполнение 100% - 70% заданий;

**Средний уровень** *–* выполнение от 50% до 70% заданий;

**Низкий уровень** *-* выполнение менее 50% заданий.

**УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **НАИМЕНОВАНИЕ ТЕМЫ** | **ТЕОРИЯ** | **ПРАКТ.** | **ВСЕГО** |
| **I. Интерфейс программы Scratch (3 ч)** | | | | |
| 1 | Введение. Техника безопасности. Что такое Scratch. Основные алгоритмические конструкции. Знакомство с интерфейсом программы Scratch. | 2 | 1 | 3 |
| **II. Начало работы в среде Scratch (4 ч)** | | | | |
| 2 | Сцена. Редактирование фона. Добавление фона из файла. | 1 | 1 | 2 |
| 3 | Понятие спрайтов. Добавление новых спрайтов. Рисование новых объектов. | 1 | 1 | 2 |
| **III. Основные скрипты программы Scratch (33 ч)** | | | | |
| 4 | Синий ящик – команды движения. Темно-зеленый ящик – команды рисования. | 1 | 2 | 3 |
| 5 | Фиолетовый ящик – внешний вид объекта. Оживление объекта с помощью добавления костюмов. | 1 | 3 | 4 |
| 6 | Желтый ящик – контроль. Лиловый ящик – добавление звуков. | 1 | 3 | 4 |
| 7 | Использование в программах условных операторов. | 1 | 3 | 4 |
| 8 | Функциональность работы циклов. Цикличность выполнения действий в зависимости от поставленных условий. | 1 | 3 | 4 |
| 9 | Зеленый ящик – операторы. Использование арифметических и логических блоков вместе с блоками управления. | 1 | 3 | 4 |
| 10 | События. Оранжевый ящик – переменные. | 1 | 3 | 4 |
| 11 | Списки. | 1 | 2 | 3 |
| 12 | Голубой ящик – сенсоры. Ввод-вывод данных. | 1 | 2 | 3 |
| **IV. Работа с несколькими объектами. Синхронизация их работы (6 ч)** | | | | |
| 13 | Последовательность и параллельность выполнения скриптов. | 1 | 2 | 3 |
| 14 | Взаимодействие между спрайтами. Управление через обмен сообщениями. | 1 | 2 | 3 |
| **V. Использование программы Scratch для создания мини-игр (20 ч)** | | | | |
| 15 | Виды компьютерных игр. Алгоритмическая разработка листинга программы. | 2 | 2 | 4 |
| 16 | Разработка базовых спрайтов для игры. Формирование базовых скриптов. | 1 | 3 | 4 |
| 17 | Синхронизация работы скриптов для разных спрайтов. | 0 | 3 | 3 |
| 18 | Переход из одной сцены в другую. Создание интерфейса игры. | 2 | 4 | 6 |
| 19 | Сообщество Scratch в Интернете. Просмотр и публикация проектов. | 1 | 2 | 3 |
| **VI. Разработка итогового творческого проекта (6 ч)** | | | | |
| 20 | Разработка и защита творческого проекта | 0 | 6 | 6 |
| **Итого:** | | **21** | **51** | **72** |

**Содержание программы**

**I. Интерфейс программы Scratch (3 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 средеScratch. Основные компоненты проекта Scratch: спрайты и скрипты. Принцип создания анимации и движения объектов. Листинг программы. Сцена. Текущие данные о спрайте. Стиль поворота. Закладки. Панель инструментов, Новый спрайт. Координаты мышки. Режим представления. Окно скриптов. Окно блоков.Блоки стека. Блоки заголовков. Блоки ссылок. Самодостаточные и открытые скрипты (3 час).

**II. Начало работы в среде Scratch (4 ч).**

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

**Теория.** Сцена. Широта 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 (1 часа).

**Практика.**Создание фона сцены на выбранную учащимся тему (1 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 (1 часа).

**Практика.**Создание фона сцены и прорисовка основных спрайтов для Scratch-истории. (1 часа).

**III. Основные скрипты программы Scratch (33 ч).**

**4.Синий ящик – команды движения. Темно-зеленый ящик – команды рисования.**

**Теория.** Команды – *идти*; *повернуться направо(налево)*; *повернуть в направлении*; *повернуться к; изменить х (у) на; установить х (у) в; если край, оттолкнуться.* Принципиальное различие действия команд *идти в* и *плыть в*. Назначение сенсоров *положение х, положение у* 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* (1 часа).

**Практика.**Создание программ для передвижения спрайтов по сцене. Создание программ для рисования различных фигур (2 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(1 часа).

**Практика.**Создание программы для управления внешним видом объекта. Создание Scratch-историй с имитацией хождения и движения объектов (3 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*  (1 часа).

**Практика.** Создание программ с элементами управления объектом. Озвучивание Scratch-историй (3 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 (1 часа).

**Практика.** Создание программ с изменением последовательного выполнения скриптов при наличии условий (3 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 (1 часа).

**Практика.** Создание программ с использованием циклов с фиксированным числом повторений. Создание программ с использованием циклов с предусловием и постусловием (3 часа).

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

**Теория.** Числа. Стринги. Логические величины. Логические выражения. Арифметические операции. Логические операции. Операции сравнения. Команды для работы со стрингами – *слить, буква…в, длинна строки.* Команда *выдать случайное от…до.* Использование арифметических и логических блоков в листинге программы. Просмотр полученного результата (1 часа).

**Практика.** Создание программ с использованием операций сравнения данных. Создание программ с использованием арифметических данных и логических операций (3 часа).

**10. События. Оранжевый ящик – переменные.**

**Теория.** События в проектахScratch. 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 (1часа).

**Практика.** Разработка сценария Scratch-историй с несколькими событиями. Создание проектов с использование глобальных и локальных переменных (3 часа).

**11. Списки.**

**Теория.** Создание списков и необходимость их использования в проектахScratch. 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* (1 часа).

**Практика.** Создание программ-тестов по принципу сравнения данных из нескольких списков (2 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…иждать.* Сенсоры *мышка по х, мышка по у, 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* (1 часа).

**Практика.** Создание проектов с использованием значений сенсоров и команды *спросить.* Создание программ для обработки данных пользователя с выводом на экран конечного результата (2 часа).

**IV. Работа с несколькими объектами. Синхронизация их работы (6 ч).**

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

**Теория.** Последовательные и параллельные потоки в программах Scratch. Одновременная и попеременная работа нескольких исполнителей (1 часа).

**Практика.** Создание Scratch-историй с одновременной и попеременной работой нескольких исполнителей (2 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 (2 час).

**V. Использование программы Scratch для создания мини-игр (20 ч).**

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

**Теория.** Компьютерные игры – вред или польза. Виды компьютерных игр. Этапы разработки игр программистами (2 час).

**Практика.** Алгоритмическая разработка проекта. Запись на естественном языке событий и точек взаимодействия героев будущей игры (2 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 (1 часа).

**Практика.** Разработка и создание основных спрайтов и их костюмов для будущей игры. Разработка скриптов для спрайтов и объектов (3 часа).

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

**Практика.**Доработка основного листинга программы с целью установления связей между спрайтами. Тестирование и отладка программы (3 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 (2 час).

**Практика.** Создать программу для перемещения объектапо игровой карте и разработать интерфейс для Scratch-проекта (4 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 проектовScratch. (1 час).

**Практика.** Регистрация на сайте сообществаScratch. Просмотр проектов сообщества и публикация собственных проектов (2 часа).

**VI. Разработка творческого проекта (6 ч)**

**20.Разработка и защита творческого проекта.** Разработка и создание программы с использованием подготовленных материалов. Тестирование и отладка проекта. Защита проекта **(6 часов).**

ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

**Методическое обеспечение реализации программы**

Методика обучения по Программе основана на методических рекомендациях, утвержденных распоряжением Министерства просвещения РФ от 12.01.2021 №Р-6 «Об утверждении методических рекомендаций по созданию и функционированию в общеобразовательных организациях, расположенных в сельской местности и малых городах, центров образования естественно-научной и технологической направленностей» и состоит из сочетания словесного изложения теоретического материала с показом иллюстрирующего материала и приемов решения практических и расчетных задач. Обучающиеся закрепляют полученные знания путем самостоятельного решения задач, выполнения практических работ. Для развития творческого мышления и навыков аналитической деятельности проводятся тематические дискуссии, мозговые штурмы, интеллектуальные игры.

Формы занятий

* лекция;
* семинар;
* практическая работа;
* самостоятельная работа;
* защита проектов.

Материально-технические условия реализации Программы

Программа реализуется с использованием оборудования Центра образования естественно-научной и технологической направленностей "Точка роста".

**Список используемой литературы**

1. Базисный учебный план общеобразовательных учреждений Российской Федерации. М.: Просвещение, 2008. 25 с. (Стандарты второго поколения).
2. Герасимова Т. Б. Организация проектной деятельности в школе. // Преподавание истории в школе. 2007. № 5. С. 17–21.
3. Краля Н. А. Метод учебных проектов как средство активизации учебной деятельности учащихся: Учеб-но-методическое пособие / Под ред. Ю. П. Дубенского. Омск: Изд-во ОмГУ, 2005. 59 с.
4. Матвеева Н. В. Информатика и ИКТ. 3 класс: методическое пособие / Н. В. Матвеева, Е. Н. Челак, Н. К. Конопатова, Л. П. Панкратова. М.: БИНОМ. Лаборатория знаний, 2009. 420 с.
5. Патаракин Е. Д. Учимся готовить в среде Скретч (Учебно-методическое пособие). М: Интуит.ру, 2008. 61 с.
6. Скретч [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вернуться домой». URL: http://letopisi.ru/index.php/Скретч
7. Хохлова М. В. Проектно-преобразовательная деятельность младших школьников. // Педагогика. 2004. № 5. С. 51–56.
8. Школа Scratch [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вернуться домой». URL: http://letopisi.ru/index.php/Школа\_Scratch
9. Scratch | Home | imagine, program, share [cайт]. URL: http://scratch.mit.edu

Scratch | Галерея | Gymnasium №3 [cайт]. URL: http://scratch.mit.edu/galleries/view/54042