

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Министерство образования Московской области

Управление образования Администрации

Г.о. Балашиха Московской области

ОАНО "ПОЗИЦИЯ"

СОГЛАСОВАНО

Зам. директора по УВР



Луцаева С.А.

Протокол №1
от «26» августа 2024 г.

УТВЕРЖДЕНО

Директор ОАНО "ПОЗИЦИЯ"



Малахов С.А.

Приказ №44
от «26» августа 2024 г.

**ОАНО
"ПОЗИЦИЯ"**

Подписано цифровой

подписью: ОАНО

"ПОЗИЦИЯ"

Дата: 2024.08.27

10:16:33 +03'00'

РАБОЧАЯ ПРОГРАММА

по курсу внеурочной деятельности «Юный программист»

для обучающихся 3 - 4 классов

Г.о. Балашиха, мкр. Железнодорожный 2024 г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.

Дополнительная общеобразовательная программа «Юный программист» (далее программа) имеет техническую направленность, она предназначена для получения учащимися дополнительных знаний в области программирования в игровой, увлекательной форме, используя языки программирования: Scratch, КуМир.

Нормативные правовые акты и государственные программные документы в соответствии с которыми разработана дополнительная общеобразовательная программа:

1. федеральный закон Российской Федерации от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»;
2. приказ Министерства образования и науки Российской Федерации от 29 августа 2013 года № 1008 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
3. письмо Министерства образования и науки Российской Федерации от 18 ноября 2015 года № 09-3242 «Методические рекомендации по проектированию дополнительных общеразвивающих программ»;
4. распоряжение Правительства Российской Федерации от 29 мая 2015 года № 996-р «Стратегия развития воспитания в Российской Федерации на период до 2025 года»;
5. постановление Главного государственного санитарного врача Российской Федерации от 4 июля 2014 года № 41 г. Москва «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».

Тип программы - авторская.

Актуальность программы.

Большинство школьных программ по информатике в основном охватывают пользовательский уровень обучения работы на компьютере. Школьников учат работать с текстовыми документами, элементарной графикой, создавать презентации и так далее. Темам «алгоритмы» и «программирование» отводится небольшое количество часов, изучаются они поздно, в старших классах. Отсутствуют такие необходимые для развития логического мышления предметы, как алгоритмика, логика, которые были, когда учились более старшие поколения. Это замедляет формирование алгоритмического мышления и не способствует развитию интереса учащихся

к программированию. Парадокс ещё заключается в том, что наибольший бал в заданиях Единого Государственного Экзамена и во всех значимых олимпиадах, приносят как раз задания на программирование.

Обучение по данной программе создает благоприятные условия для более раннего «погружения» учащихся в мир логики, и позволяет сформировать у детей стойкий интерес к получению и усовершенствованию знаний в области программирования и IT - технологий, что очень актуально в современном образовательном процессе, также выявить способных к программированию детей и оказать помощь в профессиональном самоопределении.

Новизна программы.

Новизна представленной программы заключается, **во-первых**, в том, что она является первой ступенью непрерывного курса обучения детей программированию, дизайну и сайтостроению со ступени начального до старшего звена в «Школе программирования и дизайна».

В основе организации целостного педагогического процесса – известные принципы преемственности, последовательности и систематичности, но, в данном случае они предполагают такую организацию педагогического процесса, при которой существует взаимосвязь между образовательными программами, при которой последующий курс является более высокой ступенькой ранее проводимой работы, закрепляет и развивает достигнутое, поднимает обучающегося на более высокий уровень развития. Видение перспективы своего творческого развития способствует росту познавательного интереса.

«Школа программирования и дизайна» – это серия дополнительных общеобразовательных программ для детей 8 - 17 лет, основные направления которых - программирование, дизайн и сайтостроение.

Обучение построено по ступенчатому принципу: отучившись по программе «Юный программист» ребёнок может продолжить обучение на более старших ступенях: «Компьютерная графика и дизайн» и «Web – дизайн и программирование». В зависимости от подготовки ребенка возможен переход сразу на какую-либо ступень.

Во-вторых, в сочетании известных педагогических форм, методов, приёмов организации образовательного процесса с инновационными (мультимедийные учебные пособия, флеш-анимации, видеоролики, веб- кейсы и другие интернет – технологии), что стимулирует развитие интереса обучающихся. Созданные проекты размещаются на Международном портале Scratch, одаренные учащиеся работают над индивидуальными проектами, с которыми участвуют в конкурсах по программированию.

Отличительные особенности программы.

Отличительными особенностями программы «Юный программист» являются:

1. Проектный подход.

Использование метода проектов позволяет обеспечить условия для развития у учащихся навыков самостоятельной постановки задач и выбора

оптимального варианта их решения, самостоятельного достижения цели, анализа полученных результатов с точки зрения решения поставленной задачи.

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.

Отличительной особенностью программы является ранний возраст обучающихся программированию. Через разработку проектов учащиеся получают знания, обозначенные в программах более старших классов. Так, например, осваиваются основные алгоритмические конструкции (информатика), действия с десятичными дробями, отрицательными числами, понятие координатной плоскости (математика).

4. Коммуникация.

Огромным достоинством программы является возможность обучению навыкам работы в парах, группе, командах, создание коллективных проектов, чего практически невозможно достичь при изучении традиционных языков Бейсик и Паскаль. Обязательное условие — публичная презентация и защита проектов. Возможность увидеть результаты своего труда в Интернет также стимулирует интерес детей к обучению.

Обучение программированию по программе «Юный программист» осуществляется в свободно распространяемых средах Scratch, КуМир и Python. Для начала обучения детей программированию выбрана среда Scratch, так как данный язык очень прост для ребенка. Программы в Scratch не пишут, а собирают из готовых блоков, то есть снимается проблема работы с текстом программы, не нужно заучивать наизусть названия команд и уметь писать их без ошибок. Основной задачей становится разработка алгоритма и его воплощение в виде компьютерной игры, мультфильма, видеоролика. Освоив язык программирования Scratch ребятам будет легче изучать другие языки.

Вторым языком программирования выбран КуМир. В данной среде при кодировании алгоритма уже требуется изучить синтаксис языка, но все конструкции команды и переменные мы можем задавать простыми русскими словами, что облегчает написание программы для детей. Также плюсами КуМира являются: автоматическое выравнивание, показывающее структуру программы; осуществление постоянного контроля за правильностью, вводимой программы, на полях сообщается о всех обнаруженных ошибках; наглядная визуализация процесса выполнения программы. Среда КуМир обладает расширенным набором Исполнителей (Черепашка, Водолей, Робот, Чертежник) с разнообразными системами команд, несмотря на кажущуюся простоту в ней можно решать вполне достойные задачи. Таким образом, мы продолжаем формировать правильное представление об основных конструкциях языка (цикл, условие, переменная, подпрограмма) и

закрепляем и развиваем навыки, полученные на первом этапе в Scratch. Приобретенные навыки программирования в среде КуМир могут быть в дальнейшем использованы при подготовке как к ОГЭ в 9 классе, так и ЕГЭ в 11 классе.

Педагогическая целесообразность программы заключается в том, что она является целостной и непрерывной в течение всего процесса обучения. Позволяет формировать у обучающихся не только логическое мышление, но и раскрывать в себе творческие возможности, развивать навыки работы с мультимедиа и самореализоваться в современном мире. Не смотря на очевидные преимущества обучения основам программирования в средах Scratch, КуМир в настоящее время они еще мало освоены, очень мало методических разработок и учебников по их освоению .

Цель программы:

создать комфортные условия для формирования и развития у обучающихся алгоритмического мышления в процессе изучения основ программирования в средах Scratch, КуМир .

Задачи программы:

Образовательные:

- ✓ обучение основным базовым алгоритмическим конструкциям;
- ✓ освоение основных этапов решения задачи;
- ✓ обучение навыкам разработки, тестирования и отладки несложных программ;
- ✓ формирование навыков решения олимпиадных задач;
- ✓ знакомство с понятием проекта, его структуры, дизайна и разработки.

Развивающие:

- ✓ развивать познавательный интерес обучающихся;
- ✓ развивать творческое воображение, математическое и образное мышление учащихся;
- ✓ развивать умение работы с компьютерными программами;
- ✓ развивать навыки планирования проекта, умение работать в группе.

Воспитательные:

- ✓ воспитывать интерес к занятиям информатикой;
- ✓ воспитывать культуру общения между учащимися;
- ✓ воспитывать культуру безопасного труда при работе за компьютером;
- ✓ воспитывать культуру работы в глобальной сети.
- ✓ стимулирование интереса и склонности к выбору будущей профессии в сфере информатики и математики, а также в смежных областях.

Сроки реализации программы.

Содержание программы рассчитано на 3 года обучения.

1 год обучения – 34 часа (1 час в неделю в течение учебного года).

2 год обучения – 34 часа (1 час в неделю в течение учебного года).

Характеристика обучающихся.

Дополнительная общеобразовательная программа «Юный программист» предназначена для обучения детей в возрасте 8-10 лет. Учащиеся набираются по желанию. Наполняемость групп: число учащихся в группах - 12 человек.

Формы и режим занятий.

Основными формами организации образовательного процесса являются:

- ✓ лекции, в том числе с использованием наглядных средств обучения;
- ✓ беседы;
- ✓ викторины;
- ✓ дискуссии;
- ✓ практические занятия на компьютере;
- ✓ конкурсы;
- ✓ игры;
- ✓ самостоятельная работа учащихся;
- ✓ соревнования;
- ✓ защита проектов.

Работа с использованием компьютера, с перерывами на теорию, лекции, обсуждение проектов, эвристические беседы, дискуссии, а также перерывами между академическими часами 10 минут.

2. ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ ПО ИТОГАМ ОБУЧЕНИЯ.

По окончании изучения теории учащиеся 1 года обучения должны знать:

- ✓ правила безопасной работы;
- ✓ что такое исполнитель, среда исполнителя, команды исполнителя;
- ✓ что такое алгоритм, скрипт, программа;
- ✓ типы алгоритмов (линейный, разветвляющийся, циклический);
- ✓ интерфейс программы Scratch;
- ✓ логические операции;
- ✓ координаты на плоскости;
- ✓ понятие проект, его структура и реализация в среде Scratch.

По окончании освоения практической части учащиеся 1 года обучения должны уметь:

- ✓ составлять алгоритмы любого типа;
- ✓ оформлять алгоритмы в изучаемой среде программирования;
- ✓ тестировать и отлаживать созданную программу;
- ✓ создавать личные проекты в среде Scratch;
- ✓ использовать сеть для обмена материалами работы.

По окончании изучения теории учащиеся 2 года обучения должны знать:

- ✓ общий подход к решению задач по программированию, разнообразие способов решения задач;
- ✓ основные виды алгоритмов;
- ✓ основы алгоритмического языка КуМир;
- ✓ графические возможности КуМир;
- ✓ понятие «хороший стиль программирования».

По окончании освоения практической части учащиеся 2 года обучения должны уметь:

- ✓ писать простейшие программы на алгоритмическом языке.
- ✓ создавать программы, рисующие несложные картинки.
- ✓ конструировать многомодульные программы.
- ✓ создавать читаемые программы (т.е. использовать «хороший стиль программирования»).

Формы промежуточной и итоговой аттестации.

Промежуточная и итоговая аттестация проводится в следующих формах подведения итогов:

- ✓ опрос;
- ✓ зачёт;
- ✓ тестирование;
- ✓ самостоятельная работа;
- ✓ открытое занятие для родителей;
- ✓ участие в тематических конкурсах, играх, соревнованиях.

Таблицы (КИМ), в которых фиксируются результаты аттестации по полугодиям в приложении.

Способы определения результативности.

С целью диагностики успешности освоения программы, достижений обучающихся и коррекции образовательного процесса применяются следующие методы:

- педагогическое наблюдение за обучающимися при выполнении заданий, в ходе работы над проектом;
- педагогический анализ выполнения творческих работ, результатов тестирования, анкетирования, участия в тематических играх, конкурсах, соревнованиях.
- педагогический мониторинг (диагностика развития личностных качеств обучающихся, диагностика освоения ЗУНов).

3. СОДЕРЖАНИЕ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ДЛЯ ДЕТЕЙ (1 ГОД ОБУЧЕНИЯ).

Тема 1: Введение в мир программирования .

Знакомство. Цель, задачи программы. План работы на учебный год. Режим занятий. Первичный инструктаж по ОТ, ПБ, ГО и ЧС. Первичный инструктаж по темам: «Правила поведения в МАОУДО «ЦДОД», «Охрана жизни и здоровья учащихся на учебных занятиях». Демонстрация примеров проектов, сделанных в среде Scratch обучающимися объединения «Юный программист».

Анкетирование с целью выявления интересов и ожиданий. Входная диагностика. Собеседование.

Тема 2: Думай как компьютер.

Понятие исполнителя, алгоритма. Основные характеристики исполнителя. Способы управления исполнителем. Просмотр видеоролика «Алгоритм». Способы записи алгоритма.

Тренировочные и творческие задания по алгоритмам, описанным различными способами. Игра «Перевозчик». Самостоятельная работа.

Тема 3: Что такое компьютерная программа.

Компьютер как универсальный исполнитель. Понятие программы, назначение, и использование. Как стать программистом. Просмотр видеоролика «Программа, программист, программирование». Виды программ, отличие программы от скрипта.

Тема 4: Знакомство со средой Scratch.

Что такое Scratch. Установка и запуск Scratch. Основные элементы интерфейса программы Scratch. Меню и инструменты. Основные характеристики исполнителя, их особенности в среде исполнителя Scratch. Цветные блоки. Назначение блоков. Скрипты, порядок выполнения скриптов. Установка Scratch. на свой компьютер. Создание, сохранение и открытие проектов в среде Scratch. Настройка среды. Знакомство с проектами сообщества Scratch.

Тема 5: Спрайты. Библиотеки костюмов и сцен.

Спрайты. Информация спрайта. Что могут спрайты. Операции над спрайтами. Костюмы. Выбор костюмов. Создание декораций - фонов. Изменение внешнего вида спрайта и фона. Импорт изображений. Экспорт спрайтов и их использование в проектах.

Творческие задания для детей - создание исполнителей и фонов в проекте: «Круглые рисунки». Создание костюмов. Пользуемся помощью интернета.

Тема 6: Графический редактор Scratch.

Растровая и векторная графика. Особенности работы с растровым и векторным режимами графического редактора Scratch. Общие инструменты рисования. Обозначение центра изображения. Установка прозрачных цветов. Инструменты растрового и векторного режимов рисования. Слои изображения. Группировка фигур.

Рисуем в Scratch в растровом режиме: домик, кувшинку, кораблик, рыбку, робота. Рисуем в векторном режиме: машинку, монстра. Сохраняем спрайты для их использования в проектах.

Тема 7: Учимся управлять спрайтами.

Основные команды движения и события. Коричневые блоки - команды группы «События» - «Когда щёлкнут по флажку» и «Когда клавиша на клавиатуре нажата». Синие блоки - команды группы «Движение».

Знакомство с отрицательными числами. Движение задом наперед.

Создание алгоритма первого проекта в среде Scratch. Анимация «Научим кота бегать», запуская скрипт разными событиями: клик мышкой, нажатие разных клавиш. Задание: «Автомобиль с пятью скоростями». Выполнение дополнительных заданий.

Тема 8: Создание спецэффектов - команды «Внешность»

Фиолетовые блоки - «Внешность» - блоки управления внешностью для спрайтов и сцены. Что такое анимация. Анимирование костюма. Эффекты изображений: учим спрайты исчезать и появляться - телепортация с помощью эффектов, изменять цвет и т. д. Размер и видимость. Диалоги - спрайты умеют думать и говорить. Слои.

Задания: «Осьминог», «Танцующая балерина», «Бегущий человек», «Оживляем динозавра», «Программируем страшилки» - спрайты неожиданно появляются и исчезают, как привидения.

Тема 9: Навигация в среде Scratch.

Навигация . Координатная плоскость. Что такое система координат.

Знакомимся с координатой X. Знакомимся с координатой Y. Перемещение

объектов по горизонтали. Перемещение объектов по вертикали. Что такое угол поворота. Крутой поворот. Вращение спрайтов. Блоки перемещения по координатной плоскости.

Ориентируемся по координатам. Задания: расшифруй пословицы, определи координаты точек, нарисуй рисунок по координатам. Рисование по координатам в среде Scratch. Задания с использованием команд движения: «Движение царапки по периметру сцены», «Движение по диагонали на нажатие клавиш».

Тема 10: Команды группы «перо»

Темно-зеленые блоки - группа «Перо». Спрайты умеют рисовать. Перьевая черепаха. Исследуем печать.

Игры с пером. Рисуем разные фигуры, буквы, лесенки, штрихпунктирные линии, спирали, цветные пятна, узоры из фигур. Задания: «Конфетти», «След от полёта», «Паук плетёт паутину».

Тема 11: Команды управления. Простые циклы

Желтые блоки - команды группы «Управление». Паузы. Простые циклы.

Цикл «Повторить». Запись цикла со счетчиком в виде блок – схемы.

Понятия: «Заголовок цикла», «Тело цикла». Циклы и эффект цвета. Циклы и эффект призрака. Вращение. Бесконечный цикл. Автоматическая печать.

Учим спрайты совершать повторяющиеся действия. Создаём «Вечный двигатель». Усовершенствуем ранее созданные программы в Scratch.

Создаём новые: «Гонки», «Полет самолета», «Разноцветный экран».

Тема 12: Вложенные циклы

Вложенные циклы. Работа с алгоритмами. Советы и секреты.

Задания: «Вращающиеся квадраты», «Танцующий динозавр», «Снежинка», «Цветок», «Разные орнаменты», «Создаем рисунки из многоугольников».

Тема 13: Команды группы - «Звук»

Добавление звуков к спрайтам. Проигрывание звука. Запись звука. Как проигрывать аудиофайлы. Переворачиваем звуки. Игра на барабанах и другие звуки. Создание своей музыки. Полный звукоряд. Устанавливаем темп и такт. Контроль громкости звука. Ноты. Как рисуют ноты.

Задания: «Организуем ансамбль», «Танцы на сцене», «Фейерверки». Создаем песни: «Маленькой ёлочке», «Чижик-пыжик», «Гуси» и другие.

Тема 14: Звуковые события. Распознавание движения

Коричневые блоки - команды группы «События» - «Когда громкость больше параметра», «Когда движение видео больше параметра», «Когда фон меняется».

Задания: «Учим кота реагировать на звук, на изменение фона и на распознавание движения».

Тема 15: Обмен сообщениями

Как осуществить связь между спрайтами в виде беседы. Коричневые блоки - команды группы «События» - «Когда я получу сообщение», «Передать сообщение», «Передать сообщение и ждать». Создаем процедуры (подпрограммы) при помощи передачи сообщений.

Задания: «Рисование квадратов случайных цветов в ответ на передачу сообщения», «Бесконечное движение между фонами», «Лампа», «Презентация». Создаем проекты: «Дюймовочка», или «Путешествие в страну безопасного интернета» или на свою тему.

Тема 16: Блоки Scratch: обзор и типы данных

Командные блоки. Блоки-функции (репортеры). Блоки-триггеры.

Контрольные блоки. Что означает форма блока. Разновидности данных в Scratch. Числовые данные. Строковые данные.

Проект «Кот и птичка в клетке». Создание героев и декораций проекта.

Программирование проекта. Отладка программы.

Тема 17: Математические операторы

Светло-зелёные блоки - группа «Операторы». Арифметические операторы.

Случайные числа. Математические функции.

Решаем задачи, используя операторы. Задания: «Простая арифметика со Скретчем», «Делаем кота большим и маленьким», «Рисующий волшебник», «Барабан», «Расширяющаяся спираль».

Тема 18: Проект «Генератор кубиков»

Проект со случайными числами «Генератор кубиков». Рисование спрайтов и фона проекта. Программирование проекта.

Тема 19: Считывание и распознавание

Голубые блоки - команды группы «Сенсоры». Сенсоры восприятия внутренних и внешних событий: управление с клавиатуры, столкновения спрайтов, касания, сенсор общения с человеком, для управления временем в проектах и т. д. Использование блоков группы «Сенсоры». Создаём игры: «Кошки-мышки»,

«Игра с пончиком».

Тема 20: Истина или ложь

Операторы сравнения. Логические операторы: блоки «и», «или», «не».

Объединение вопросов. Соблюдение условий. Использование логических операторов для проверки областей числовых значений.

Самостоятельная работа: понятия «Истина» и «Ложь». Проекты с использованием блоков «и», «или», «не»: «Сравнение десятичных дробей», «Угадай мои координаты».

Тема 21: Принятие решений и ветвление

Использование блока «если». Структура блока «если». Использование переменных как флагов. Использование блока «если/иначе». Структура блока «если/иначе». Вложенные блоки «если» и «если/иначе».

Работа с алгоритмами с ветвлением. Проекты: «Осторожно лужи», «Управляемый робот», «Тренажер памяти».

Тема 22: Проект «Простой Paint»

Рисуем интерфейс программы для рисования: фон, цветные карандаши и ластик. Создаём скрипты, проверяем работу программы.

Тема 23: Игра «Лабиринт»

Создание игры - «Лабиринт» - три варианта.

Тема 24: Как не зациклится

Сложные условия. Блок «повторять, пока не выполнится условие».. Создание блока «всегда если». Блок «ожидание». Стоп-команды: блок «стоп».

Создание проектов: «Часы», «Стрельба по воробьям».

Тема 25: Команды группы - «Данные». Переменные

Оранжевые блоки - группа команд «Данные». Понятие - переменная.

Создание, переименование и удаление переменной. Использование переменных. Присвоение и изменение значений. Сравнение переменных.

Константа. Датчики.

Создание проектов: «С переменной время», «С переменной счёт», «С переменной прыжок», «Идеальный вес», «Оптические иллюзии».

Тема 26: Строки и списки

Работа со словами. Строковые операторы. Подсчет специальных символов в строке. Создание списков. Команды управления списками. Доступ к элементам списка. Динамические списки.

Игры со списками: «Угадай слово», «Замена букв».

Тема 27: Создание блоков

Создание своего блока. Подпрограмма (процедура). Блоки с параметрами.

Переменные в подпрограммах. Вложенные процедуры.

Создание игр и тестов: «Звёздное небо»

Тема 28: Создание мультфильмов в среде Scratch

Мультфильмы по сказкам «Колобок», «Репка».

Тема 29: Создание комиксов в среде Scratch

Создание комиксов «Охота за приведениями», «Мой город».

Тема 30: Создание интерактивных поздравительных открыток в среде Scratch

Создание открыток «С днём рождения», «С Новым годом».

Тема 31: Создание презентаций в среде Scratch

Презентации: «Мои увлечения», «Моя семья».

Тема 32: Создание компьютерных игр в среде Scratch

Проект «Убегая от преследователя», «Пора в космос».

Тема 33: Создание интерактивных игр

Игра «Бешенные обезьяны», «Интерактивное общение».

Тема 34: Создание интерактивных тестов

Тесты по школьным предметам: «Играем в города по географии», «Зоопарк по биологии».

Тема 35: Создание музыкального клипа в среде Scratch

Создание клипов в среде Scratch: «Говорящий алфавит», «Как незнайка учил ноты».

Тема 36: Создание учащимися разных проектов

Проекты «Театр в Scratch», «Квест в Scratch», «Перевертыши».

Тема 37: Разработка личного проекта

Понятие проект, его структура и реализация в среде Scratch. Этапы разработки проекта (постановка, алгоритмизация, кодирование, тестирование, отладка).

Тема 38: Дизайн проекта.

Создание материалов для проекта.

Тема 39: Программируем проект.

Разработка и создание программы своего проекта.

Тема 40: Тестирование и отладка проекта.

Проверяем и исправляем ошибки.

Тема 41: Защита проекта

4. Учебно-тематический план (1 год обучения).

№	Наименование раздела	Количество часов		
		Теория	Практика	Всего часов
1. Введение в мир программирования				
1.	Введение в мир программирования	1		1
2.	Думай, как компьютер	1		1
3.	Что такое компьютерная программа		1	1
	Всего:	2	1	3
2. Основные приемы программирования и создания проектов в среде Scratch				
4.	Знакомство со средой Scratch		1	1
5.	Спрайты. Библиотеки костюмом и сцен		1	1
6.	Графический редактор Scratch		1	1
7.	Учимся управлять спрайтами		1	1
8.	Создание срецэффектов-команды «Внешность».		1	1
9.	Навигация в среде Scratch		1	1
10.	Команды группы «Перо».		1	1
11.	Команды управления. Простые циклы		1	1
12.	Вложенные циклы.		1	1
13.	Команды группы - «Звук».		1	1
14.	Звуковые события. Распознавание		1	1

	движения.			
15.	Обмен сообщениями.		1	1
16.	Блоки Scratch: обзор и типы данных.		1	1
17.	Математические операторы.		1	1
18.	Проект «Генератор кубиков»		1	1
19.	Считывание и распознавание.		1	1
20.	Истина или ложь.		1	1
21.	Принятие решений и ветвление.		1	1
22.	Проект «Простой Paint».		1	1
23.	Как не зациклится.		1	1
	Всего		20	20
3. Разработка групповых проектов в среде Scratch				
24.	Создание мультфильмов в среде Scratch		1	1
25.	Создание комиксов в среде Scratch		1	1
26.	Создание интерактивных поздравительных открыток в среде Scratch		1	1
27.	Создание презентаций в среде Scratch		1	1
28.	Создание компьютерных игр в среде Scratch		1	1

29.	Создание интерактивных игр		1	1
30.	Создание интерактивных тестов		1	1
31.	Создание музыкального клипа в среде Scratch		1	1
	Итого		8	8
4. Выполнение творческой работы-создание личного проекта в среде Scratch				
32.	Разработка личного проекта		1	1
33.	Разработка личного проекта		1	1
34.	Защита проекта		1	1
	Всего		3	3
	Итого	2	32	34

Учебно-тематический план (2 год обучения).

№	Наименование раздела	Количество часов		
		Теория	Практика	Всего часов
1. Основные приемы программирования и создания проектов в среде КуМир				
1.	Алгоритмы и исполнители	1		1
2.	Знакомство с системой КуМир		1	1
3.	Среда КуМир. Исполнитель Черепашка		1	1
4.	Среда КуМир. Исполнитель Кузнечик		1	1
5.	Среда КуМир. Исполнитель Водолей		1	1
6.	Среда КуМир. Исполнитель Робот		1	1
7.	Вспомогательные алгоритмы		1	1
8.	Проектирование сверху вниз и снизу вверх		1	1
9.	Метод последовательного уточнения		1	1
10.	Цикл пока		1	1
11.	Команда ветвления		1	1
12.	Циклы и условия		1	1
13.	Тестирование алгоритмов		1	1
14.	Исполнитель. Чертежник		1	1
15.	Рисуем с Чертежником		1	1
16.	Математические операции и функции в среде КуМир		1	1
17.	Основные этапы разработки проекта		1	1

	Всего	1	16	17
2. Создание личного проекта				
18	Разработка личного проекта		2	2
19	Дизайн проекта		3	3
20	Программируем проект		10	10
21	Тестирование и отладка проекта		1	1
22	Защита проекта		1	1
	Всего		17	17
	Итого	1	33	34

Методическое обеспечение программы

В программе применяются следующие методы, позволяющие дать детям первоначальные основы программирования:

□□Словесные методы позволяют быстро передавать детям информацию, ставить перед ними учебную задачу, указывать пути ее решения. Это лекции, беседы, дискуссии. Словесные методы и приемы сочетаются с наглядными, игровыми, практическими методами, делая последние более результативными. способствует побуждению у детей интереса к занятию. В качестве

□□Наглядные методы: Наглядность оживляет процесс обучения, наглядности используются презентации, мультимедийные учебные пособия, флеш-анимации, видеоролики, веб-кейсы и другие интернет – технологии, демонстрационный раздаточный материал, карточки.

□□Практические методы: Нельзя научить детей, только показывая и рассказывая, не предлагая самим детям каких - либо действий. Ребенок овладевает опытом только тогда, когда сам участвует в практической деятельности. практической работы учащихся за компьютером. При решении задач используется семинарское занятие

□□Игровые методы: позволяют осуществлять учебные задачи в атмосфере легкости и заинтересованности, активности детей. Используются игровые задания, создание игр на компьютере, метод интерактивной игры, дидактические игры, и упражнения, выполнив которые ребенок легко может усвоить правила поведения, технику безопасности, гимнастику для глаз.

Формы обучения по программе:

□□Фронтальная - подача учебного материала всему коллективу детей, на этих занятиях важен «эффект эмоционального воздействия и сопереживания», что приводит к повышению умственной активности, побуждает ребенка к самовыражению (интегрированные и итоговые занятия, интеллектуальные игры).

□□Индивидуальная - используется при возникновении затруднения, не уменьшая активности детей и содействуя выработке навыков самостоятельной работы. В индивидуальных занятиях нуждаются дети с явно выраженными способностями к той или иной деятельности, дети с доминирующим познавательным интересом.

□□Подгрупповая - предоставляется возможность самостоятельно построить свою деятельность на основе принципа индивидуализации и сознательности и активности, ощутить помощь со стороны друг друга, учесть возможности каждого на конкретном этапе деятельности. Всё это способствует более быстрому выполнению задания.

5. Список используемых источников.

для педагога:

1. Патаракин Е. Д. Руководство для пользователя среды Scratch. Версия 0.2, 2007г.
2. Патаракин Е. Д. Учимся готовить в среде Скретч (Учебно-методическое пособие). М: Интуит.ру, 2008 г.
3. Рындак В. Г., Дженжер В. О., Денисова Л. В. Проектная деятельность школьника в среде программирования Scratch: учебно-методическое пособие — Оренбург, 2009 г.
4. Пашковская Ю. В. Творческие задания в среде Scratch: рабочая тетрадь для 5 - 6 классов/ Ю. В. Пашковская.- М.:БИНОМ. Лаборатория знаний, 2014. - 200 с. :ил.
5. Л. А. Анеликова, О. Б. Гусева Программирование на алгоритмическом языке КуМир. Под редакцией Кушниренко А. Г. - М.: СОЛОН-ПРЕСС, 2013. - 48 с.: ил. - (Элективный курс Профильное обучение).
6. Еремин Е. А. Газета « Информатика». Среда Scratch – первое знакомство. – М.: Первое сентября, 2008 – № 20 (573) – стр. 16–28.
7. У. Сэнд, К. Сэнд Hello World! Занимательное программирование. — СПб.: Питер, 2016. — 400 с.: ил. — (Серия «Вы и ваш ребенок»).
8. Свейгарт, Эл. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих.: Пер. с англ.—М.: ООО -ИД. Вильямс, 2016.—592с.: ил.—Парад. тит. англ.
9. Мэтиз Эрик Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека программиста»).

для ученика:

1. Торгашева Ю. В. Первая книга юного программиста. Учимся писать программы на Scratch. — СПб.: Питер, 2016. — 128 с.: ил. — (Серия «Вы и ваш ребенок»).
2. Программирование для детей/К. Вордерман, Дж. Вудкок, Ш. Макаманус и др. ; пер. с англ. С. Ломакина. - М. :Манн, Иванов и Фербер, 2015. - 224 с. :ил.