

Муниципальное бюджетное общеобразовательное учреждение
«Средняя общеобразовательная школа №3»
муниципального образования-городской округ
город Скопин Рязанской области



Утверждаю
Директор школы
_____ Е.В.Барабаш

Рабочая программа дополнительного образования
« Scratch-программирование»

Направленность: техническая

Возраст обучающихся: 10 – 12 лет

Срок реализации программы: 1 год

Количество детей в группе: от 12 чел

Количество часов в год: 163 часа

Разработчик:

Генералова Е.С., учитель информатики

Оглавление

1. Пояснительная записка.....	3
1.1. Направленность и уровень освоения программы.....	4
1.2. Актуальность, новизна педагогическая целесообразность разработки и реализации программы.....	5
1.3. Соответствия программы действующим нормативно правовым актам и государственным программным документам	7
1.4. Отличительная особенность программы.....	8
1.5. Возрастно – психологические особенности учащихся.....	9
1.6. Принцип формирования учебных групп, количества учащихся. Срок реализации программы количества учебных часов.....	10
1.7. Цели и задачи программы.....	11
2.Формы проверки результатов.....	13
2.1 Учебный план.....	14
2.2. Форма промежуточного и итогового контроля.....	18
2.3. Содержание учебного плана	19
2.4. Планируемые результаты.....	26
3. Условие реализации программы.....	27
3.1 Материально – техническое обеспечения.....	29
3.2 Кадровое обеспечение.....	30
3.3 Учебно-информационное обеспечение	

Пояснительная записка.

Программирование в среде Scratch - это дополнительная общеразвивающая программа, которая реализуется в соответствии с технической направленностью образования.

Это среда, которая позволяет легко и просто, но в тоже время обдуманно, исследовать и проявить свои творческие способности.

Данная программная среда дает принципиальную возможность составлять сложные по своей структуре программы, не заучивая наизусть ключевые слова, и при этом в полной мере проявить свое творчество и понять принципы программирования.

Одной из важнейших особенностей Scratch и Codu (Коду), как языка программирования, является его событийно-ориентированный характер. Это означает, что все объекты взаимодействуют при помощи обмена сообщениями. Такая схема обмена информацией делает Scratch близким к современным объектно-ориентированным языкам и позволяет впоследствии более просто организовать переход к изучению Python.

Программирование в визуальной – ориентированной среде Scratch (Скретч) и Codu (Коду) позволяет развивать творческие способности школьников, их логическое мышление, привлекать к активному использованию информационных технологий.

1.1. Направленность программы

Курс является практико-ориентированным и предполагает групповую и индивидуальную проектную деятельность учащихся. Разделение функций и ролей среди участников проекта может быть основано на следующих принципах:

- по функции или роду деятельности (сценарист, художник, программист и т. п.);
- по частям проекта (каждый участник выполняет одновременно несколько ролей, разрабатывая свою часть общего проекта).

Направленность программы – техническое.

1.2. Актуальность, новизна педагогическая целесообразность разработки и реализации программы

Актуальность данной дополнительной образовательной программы продиктована развитием современного информационного общества. Широким внедрением информационных технологий в образовательные процессы и обычную жизнь каждого человека, а также обусловлена тем, что способствует развитию мотивации к получению новых знаний, возникновению интереса к программированию как к инструменту самовыражения в творчестве, помогает в повышении самооценки, в самоопределении и выявлении профессиональной направленности личности. Учащиеся получают возможность оперативно обмениваться информацией, идеями, планами по интересующим участников совместных проектов вопросам, расширяя, таким образом, свой кругозор, повышая культурный уровень. Новизна программы заключается в комбинировании исследовательской деятельности с изучением основ программирования и создания проекта в программной среде Scratch и Codu. Аспект новизны заключается в том, что Scratch и Codu 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 мультфильмы, анимацию и даже простейшие игры, делает программу практически значимой для современного школьника. Среда программирования Codu превращает учащихся из пользователей игр в их создателей. В частности, в процессе обучения работе с Kodu учащиеся: лучше поймут процесс разработки программ для компьютеров; изучат новые методы разрешения проблем и опробуют их на практике; научатся более эффективно исполнять инструкции, получаемые в интерактивном и автономном режимах; научатся составлять истории в разных форматах с использованием различных средств;

отработают математические методы, используемые в средствах ветвления и начисления баллов. Это дает возможность увидеть практическое назначение алгоритмов и программ, что будет способствовать развитию интереса к профессиям, связанным с программированием.

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

1.3. Соответствия программы действующим нормативно правовым актам и государственным программным документам

Дополнительная общеобразовательная программа «Программирование в визуально – ориентированной среде Scratch (Скретч) и Codu (Коду)» разработана на основе следующих нормативно-правовых документов:

Федеральный Закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ.

Концепция развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 04.09.2014 № 1726-р).

Сан-Пин к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей (утверждено постановлением Главного государственного санитарного врача РФ от 04.07.2014 № 41)

Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (утвержден приказом Министерства образования и науки РФ от 29.08.2013 № 1008).

Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (Приложение к письму Департамента государственной политики в сфере воспитания детей и молодежи Министерства образования и науки РФ от 18.11.2015 № 09-3242).

Формирование алгоритмической культуры является одной из актуальных задач последнего времени.

Развитие логики и алгоритмического мышления при использовании фундаментальных понятий программирования соответствует плану мероприятий по реализации Концепции математического образования в РФ (Приказ Минобрнауки России № 265 от 3 апреля 2014 г.).

1.4. Отличительная особенность программы

Отличительные особенности среды программирования Scratch и Codu это:

- объектная ориентированность;
- поддержка событийно-ориентированного программирования;
- дружественный интерфейс;
- разумное сочетание абстракции и наглядности; организация текстов программ из элементарных блоков;
- наличие средств взаимодействия программ на Scratch и Codu с реальным миром посредством дополнительного устройства;
- встроенная библиотека объектов; встроенный графический редактор;

Среда Scratch является наиболее простой и доступной - ее удобно использовать для обучения школьников, начиная с младших классов, но и ребята более старшего возраста с удовольствием программируют в этой среде. Codu –открытая система, всегда можно взять любой проект и посмотреть, как реализован тот или иной алгоритм. Scratch заставляет творчески думать, учит общению, логическому мышлению и программированию. В Scratch и Codu реализованы основные алгоритмические структуры: ветвления, циклы. Scratch называют “разогревающим” языком. Он рассматривается многими, как преамбула к изучению более сложных, более известных языков, таких как Паскаль, Си и т. д..

1.5. Возрастно – психологические особенности учащихся

Возраст детей. Программа ориентирована на дополнительное образование учащихся среднего школьного возраста (12 – 14 лет). Важным условием для формирования теоретического мышления в среднем школьном возрасте является формирование научных понятий. Теоретическое мышление позволяет ученику решать задачи, ориентируясь не на внешние, наглядные признаки и связи объектов, а на внутренние, существенные свойства и отношения.

Занятия по Scratch программированию развивают логику, повышают системность мышления, а также развивают творческие способности. Все это так же влияет на степень осознанности в принимаемых решениях. Даже, если ребенок не станет программистом, то понимание, как составляются программы обязательно пригодятся в другой деятельности, какую бы профессию ребенок не выбрал в будущем.

1.6. Принцип формирования учебных групп, количества учащихся.

Срок реализации программы количества учебных часов

Занятия по дополнительной образовательной программе проводятся 2 раза в неделю по 2 часа (продолжительность учебного часа – 40 минут). В программе учитываются возрастные особенности детей. Условия набора детей в группу: принимаются все желающие, не имеющие медицинских противопоказаний и соответствующие входным требованиям.

Программа рассчитана на 1 год обучения, 163 часа в год. Обучение по программе осуществляется в очной форме. Форма организации образовательной деятельности обучающихся – индивидуально-групповая, индивидуальная, групповая. Программа предполагает организацию только аудиторных занятий, однако при необходимости возможно проведение и дистанционных занятий. Занятия проводятся в следующих формах: учебное занятие, игра, дискуссия, семинар, лекция, проектная работа, экскурсия, творческая мастерская.

1.7. Цель и задачи программы

Цель программы : познакомить с программированием и его возможностями, помочь сформировать у детей базовые представления о языках программирования, сформировать интерес к программированию, создать устойчивую мотивацию для дальнейшего развития в сфере программирования, сформировать алгоритмическое мышление.

Задачи программы:

Обучающие

- Расширить знания в области информационных технологий.
- Познакомить со специальной терминологией в области компьютерных программ.
- Способствовать формированию и развитию умений и навыков программирования на языке «Scratch». Составлять и записывать алгоритм для конкретного исполнителя при разработке трехмерных игр.
- Овладеть основами компьютерного 3d моделирования в событийной среде Codu, основными современными методами и средствами создания графического отображения проекта ландшафтной архитектуры.
- Овладение технологией проектной деятельности в процессе индивидуального и коллективного творчества.

Развивающие

- Активизировать познавательный процесс.
- Способствовать пробуждению и развитию творческой активности обучающихся.

- Раскрыть индивидуальные способности обучающегося, в том числе эмоционально-образное восприятие окружающего мира, ассоциативное и образное мышление.
- Способствовать развитию специфических навыков: действий с воображаемыми предметами, умению пространственно мыслить.
- способствовать развитию алгоритмического, творческого, логического и критического мышления

Воспитательные:

- формировать информационную культуру учащихся;
- способствовать формированию активной жизненной позиции;
- воспитывать толерантное отношение в группе;
- добиться максимальной самостоятельности детского творчества;
- воспитывать собранность, аккуратность при подготовке к занятию;
- воспитывать умение планировать свою работу;
- сформировать интерес к профессиям, связанным с программированием.

2. Формы проверки результатов

Текущий контроль осуществляется с целью оперативного управления учебным процессом и его коррекции. Текущий контроль уровня усвоения материала осуществляется по результатам выполнения учащимися заданий по каждому разделу курса. Задания составлены с учетом возраста учащихся. При безоценочной системе курса итоговый контроль по разделам курса может быть реализован в форме итогового задания или проекта, которые позволяют определить достижение планируемых результатов.

Функция итогового контроля заключается в определении полноты освоения содержания программы. Итоговый контроль включает:

- выполнение заданий по пройденным темам;
- творческие задания;
- индивидуальные или групповые проектные работы;
- лично или социально значимые практические работы.

Основное требование к указанным выше работам – комплексный характер. При их выполнении ученик должен проявить все знания и умения, приобретенные на курсе.

Для определения достижений и результатов прохождения программы используются:

- педагогическое наблюдение;
- оценка продуктов творческой деятельности учащихся;
- выступление учащегося с сообщением, докладом по теме, определенной учителем или самостоятельно выбранной;
- беседы, опросы

2.1. Учебный план

№	Тема	Количество часов		Формы аттестации (контроля)
		теория	практика	
1	Основы программирования в Скретч (Scratch)	1		беседа
2	Простая анимация в Скретч (Scratch)	1	1	практическая работа
3	Управление персонажем клавиатурой в Скретч (Scratch)	0,5	1	практическая работа
4	Взаимодействие с другими персонажами	0,5	1	практическая работа
5	Звук, реплики и копирование скриптов от одного персонажа к другому	1	1	практическая работа
6	Переменные в Скретч	1	2	практическая работа
7	Поздравительная открытка в Scratch (Скретч или Скратч)	1	1	проект
8	Перестановка переменных местами в Scratch (Скратч)	1	1	практическая работа
9	Использование клонов в Scratch (Скретч/Скратч)	1	1	практическая работа
10	События и кнопки в Scratch.	1		практическая работа
11	Слои и эффект движения в Scratch (Скретч)	1	1	практическая работа
12	Простая пушка в Scratch	1	1	практическая работа
13	Стрельба из пушки с синусами и косинусами в Scratch.	1	2	практическая работа
14	Списки (массивы), рисование пером и пузырьковая сортировка в Scratch.	1	2	практическая работа
15	Ловля покемона	1	1	практическая работа
16	Простая база данных на примере записной книжки	1	2	практическая работа
17	Как обойти слона	1	1	практическая работа
18	Простой графический редактор в Scratch.	1	1	практическая работа

19	Точные и неточные часы	1		практическая работа
20	Использование циклов в Scratch	1	3	практическая работа
21	Разница между растровой и векторной графикой	1	1	практическая работа
22	Доска рекордов	1	1	практическая работа
	Итого:	46		
23	Алгоритм - как фундаментальное понятие в информатике (Kodu Game Lab).	1		практическая работа
24	Исполнители вокруг нас(Kodu Game Lab).	0,5		беседа
25	Формы записи алгоритмов. Блок-схемы алгоритмов (Kodu Game Lab).	1	1	практическая работа
26	Линейные алгоритмы (Kodu Game Lab).	1	1	практическая работа
27	Ветвление в алгоритме (Kodu Game Lab) .	1	1	практическая работа
28	Ветвления в построчной записи алгоритма (Kodu Game Lab)	1	1	практическая работа
29	Циклические алгоритмы (Kodu Game Lab).	1	2	практическая работа
30	Пошаговая запись результатов выполнения алгоритмов (Kodu Game Lab)	1	1	практическая работа
31	Творческий мини-проект: «Алгоритмы» (Kodu Game Lab)	1	2	практическая работа
32	Программирование в среде KoduGameLab.	1	1	практическая работа
33	Общие сведения о программировании (Kodu Game Lab).	1		практическая работа
34	Знакомство с интерфейсом и основными принципами работы с программой KoduGameLab.	1	1	практическая работа
35	Главное меню, панель инструментов (редактирование) (Kodu Game Lab).	1	1	практическая работа
36	Создание персонажей с учетом игровой	1	2	практическая работа

	среды, действия персонажей (Kodu Game Lab) .			
37	Перемещение персонажей в макросреде Kodu с использованием клавиатуры и мыши.	0,5	1	практическая работа
38	Создание ландшафтов (миров), добавление объектов (Kodu Game Lab) .	1	1	практическая работа
39	Режим программирования, основные операторы Kodu.	1	1	практическая работа
40	Создание путей, выбор поведения персонажей. Создание клонов(Kodu Game Lab).	1	1	практическая работа
41	Разработка стратегии и атмосферы игры (Kodu Game Lab).	1	3	практическая работа
42	Страницы, функции, ракурс обзора (Kodu Game Lab).	1	1	практическая работа
43	Таймер, индикатор здоровья. Kodu Game Lab опция «Родитель»	0,5	1	практическая работа
44	Мини-проект на тему: «Разработка и создание собственного мира в KoduGameLab».	1	3	проект
45	Защита проекта	0,5	2	проект
46	Архитектурная графика: классика и современность Практическое применение архитектурной графики	2	3	практическая работа
47	Возможности SketchUp Знакомство с интерфейсом программы	2	2	практическая работа
48	Панели инструментов в SketchUp	2	2,5	практическая работа

49	Логический механизм интерфейса в SketchUp	2	2	практическая работа	
50	Дополнительная информация по элементам в SketchUp	2	2	практическая работа	
51	Основные инструменты в SketchUp	2	2,5	практическая работа	
52	Инструменты рисования в SketchUp	2	3,5	практическая работа	
53	Инструменты изменения в SketchUp	1	2	практическая работа	
54	Инструменты построения в SketchUp	1		практическая работа	
55	Моделирование здания в SketchUp	1,5	4	проект	
56	Моделирование участка в SketchUp	2	5	проект	
57	Импорт и экспорт в SketchUp	1		беседа	2
58	Работа над проектом	2	5	проект	
59	Защита проекта	0,5	3,5	проект	
	Итого:	68			
	Итого всего:	163			

2.2. Форма промежуточного и итогового контроля

- Текущий контроль осуществляется с целью оперативного управления учебным процессом и его коррекции.
- Текущий контроль уровня усвоения материала осуществляется по результатам выполнения учащимися заданий по каждому разделу курса. Задания составлены с учетом возраста учащихся.

При безоценочной системе курса итоговый контроль по разделам курса может быть реализован в форме итогового задания или проекта, которые позволяют определить достижение планируемых результатов. Функция итогового контроля заключается в определении полноты освоения содержания программы.

Итоговый контроль включает:

- выполнение заданий по пройденным темам;
- творческие задания;
- индивидуальные или групповые проектные работы;
- лично или социально значимые практические работы.

Основное требование к указанным выше работам – комплексный характер. При их выполнении ученик должен проявить все знания и умения, приобретенные на курсе.

Для определения достижений и результатов прохождения программы используются:

- педагогическое наблюдение;
- оценка продуктов творческой деятельности учащихся;
- выступление учащегося с сообщением, докладом по теме,
- беседы, опросы.

2.3. Содержание учебного плана

	Этап		
		Теоретическая часть	Практическая часть
1	Основы программирования в Скретч (Scratch)	Знакомство с идеологией Scratch - Правила безопасности работы на ПК и в интернете.	Знакомство с проектами других участников сообщества, первая программа.
2	Простая анимация в Скретч (Scratch)	Знакомство с основными понятиями (сцена, проект, спрайт, скрипт) - Интерфейс программы - Главное меню.	Создание небольших проектов с использованием простейших команд исполнителя.
3	Управление персонажем клавиатурой в Скретч (Scratch)	Тестирование работы клавиш со стрелками, не изменяя исходный проект.	Создание сценария, создание исполнителей, создание анимационного проекта, озвучивание проекта
4	Взаимодействие с другими персонажами	Взаимодействие скриптов использование блоков: передать сообщение; передать сообщение и ждать; когда я получу сообщение.	Составление программы состоящей из четырех скриптов
5	Звук, реплики и копирование скриптов от одного персонажа к другому	Звуковые форматы - Конвертация звука - Озвучивание	Усложнение компьютерных игр, в программах использование сложных условий.
6	Переменные в Скретч	Значение переменной отображение ее на сцене с помощью монитора.	Создание переменной отображение ее на сцене с помощью монитора.
7	Поздравительная открытка в Scratch (Скретч или Скратч)	Создание фона. Прорисовка фон1 , фон2 , фон3 в векторной графике.	Создание поздравительной открытки. Конвертация в векторную графику.
8	Перестановка переменных местами в Scratch (Скратч)	Перестановка значений переменной отображение ее на сцене с помощью монитора.	Создание переменной, перестановка значений переменной отображение ее на сцене с помощью монитора.
9	Использование клонов в Scratch (Скретч/Скрат	Взаимодействие скриптов (клонов) использование блоков: передать сообщение; передать сообщение и ждать;	Создание скриптов- клонов.

	ч)	когда я получу сообщение.	
10	События и кнопки в Scratch.	Событие и функции, поддерживаемые средой Scratch.	Создание небольших проектов с использованием простейших команд.
11	Слои и эффект движения в Scratch (Скретч)	Слои и эффекты движения	Создание мини игры с эффектами движения
12	Простая пушка в Scratch	Игра в жанре шутер. Способы реализации механики стрельбы.	Создание мини игры в жанре шутер. С использованием механики стрельбы.
13	Стрельба из пушки с синусами и косинусами в Scratch.	Игра в жанре шутер. Способы реализации механики стрельбы	Изменение готового проекта, изучение кода.
14	Списки (массивы), рисование пером и пузырьковая сортировка в Scratch.	Перо контроль размера пера, его цвет и тень.	Прорисовка объекта с помощью пера.
15	Ловля покемона	Мини игра (флеш игра) Ловля Покемонов -	Подбор спрайтов, фона и написание плана по созданию флеш игры
16	Простая база данных на примере записной книжки	Знакомство с базой данных и записной книжкой в Scratch.	Создание записной книжки в Scratch.
17	Как обойти слона	Взаимодействие скриптов использование блоков: передать сообщение; движение; событие	Создание спрайтов для флеш игры
18	Простой графический редактор в Scratch.	Редактор растровый (пиксельный) или векторный режим. Картинки в векторной обработке состоят не из отдельных пикселей, а из различных геометрических объектов (точек, линий, прямоугольников, окружностей и т.д.).	Редактирование готовых рисунков в растровой и векторной графике
19	Точные и неточные часы	Виды часов: точные и неточные. Настройка точного времени.	Создание сценария, создание исполнителей, создание анимационного проекта, озвучивание проекта

20	Использование циклов в Scratch	Виды циклов в scratch -3 вида циклов: повторять всегда, повторить... раз, повторять пока не...	Создание спрайта с 3 циклами: повторять всегда, повторить... раз, повторять пока не...
21	Разница между растровой и векторной графикой	Растровая и векторная графика разница между ними	Создание рисунков в растровой и векторной графике
22	Проект "Компьютерная игра" Создание правил игры.	Камера, компьютер, программное обеспечение.	Создание игры. Озвучивание игры
23	Алгоритм - как фундаментальное понятие в информатике (Kodu Game Lab).	Основные определения (алгоритм, программа, команда, система команд исполнителя, исполнитель) - Графическая запись алгоритма - Свойства алгоритмов	Составление и запись алгоритма. Реализация его в Kodu
24	Исполнители вокруг нас(Kodu Game Lab).	Понятие исполнителя. Неформальные и формальные исполнители.	Запись условного алгоритма с помощью блок-схем. Составление алгоритмов.
25	Формы записи алгоритмов. Блок-схемы алгоритмов (Kodu Game Lab).	Влияние последовательности шагов на результат исполнения алгоритма.	Запись последовательности шагов и результата алгоритма.
26	Линейные алгоритмы (Kodu Game Lab).	Знакомство с терминами «исполнитель», «формальный исполнитель», «среда исполнителя», «система команд исполнителя»	Исполнение линейного алгоритма для формального исполнителя с заданной системой команд.
27	Ветвление в алгоритме (Kodu Game Lab) .	Назначение среды KoduGameLab.	Рисование простейших объектов моделирование прямолинейного движение с разными скоростями.
28	Ветвления в	Знакомство с логикой работы	Создание объектов с

	построчной записи алгоритма (Kodu Game Lab)	программы, создание нового мира, создание ландшафтов.	помощью алгоритма.
29	Циклические алгоритмы (Kodu Game Lab).	Арифметические операторы и циклические алгоритмы, поддерживаемые средой Codu.	Создание циклического алгоритма (Kodu Game Lab).
30	Пошаговая запись результатов выполнения алгоритмов (Kodu Game Lab)	Пошаговая запись результата после выполнения алгоритма.	Воспроизведение пошаговой записи алгоритмов.
31	Творческий мини-проект: «Алгоритмы» (Kodu Game Lab)	Знакомство с принципами групповой работы. Освоение всех этапов проектной деятельности.	Создание алгоритма
32	Программирование в среде KoduGameLab.	Освоение принципов создания компьютерных игр.	Знакомство с проектами других участников сообщества, первая программ.
33	Общие сведения о программировании (Kodu Game Lab).	Знакомство с основными элементами пользовательского интерфейса Kodu.	Выполнение задания по наглядному образцу.
34	Знакомство с интерфейсом и основными принципами работы с программой KoduGameLab .	Интерфейс и основные принципы работы с программой KoduGameLab.	Изучение интерфейса Программы Codu
35	Главное меню, панель инструментов (редактирование) (Kodu Game Lab).	Панель управления и инструментарий.	Знакомство с панелью управления и инструментарием.

36	Создание персонажей с учетом игровой среды, действия персонажей (Kodu Game Lab) .	Понимание принципов программирования характеристик и поведения персонажей.	Выполнение упражнений по образцу, создание своего персонажа.
37	Перемещение персонажей в макросреде Kodu с использованием клавиатуры и мыши.	Использованием клавиатуры и мыши для перемещение персонажей в макросреде Kodu с	Выполнение упражнений по образцу, создание своего персонажа.
38	Создание ландшафтов (миров), добавление объектов (Kodu Game Lab) .	Приемы создания ландшафта игрового мира.	Создание небольших проектов с использованием простейших команд исполнителя.
39	Режим программирования, основные операторы Kodu.	Операторы и их функции.	Знакомство с операторами и их функциями.
40	Создание путей, выбор поведения персонажей. Создание клонов(Kodu Game Lab).	Персонажи и их функции.	Выполнение упражнений по образцу, создание своего персонажа.
41	Разработка стратегии и атмосферы игры (Kodu Game Lab).	Алгоритм разработки стратегии и атмосферы игры (Kodu Game Lab).	Выполнение упражнений по образцу, создание атмосферы игры
42	Страницы, функции, ракурс обзора (Kodu Game Lab)	Знакомство со страницами, функциями, ракурса обзора (Kodu Game Lab).	Выполнение упражнений по образцу, создание ракурса обзора игры

	ab).		
43	Таймер, индикатор здоровья. Kodu Game Lab опция «Родитель»	Знакомство с таймером, индикатором здоровья. Kodu Game Lab опция «Родитель»	Выполнение упражнений по образцу, создание индикатора здоровья игры.
44	Мини-проект на тему: «Разработка и создание собственного мира в KoduGameLab».	Знакомство с принципами групповой работы. Освоение всех этапов проектной деятельности.	Практическая работа «Разработка собственного мира»
45	Защита проекта	Презентация созданной игры.	Презентация созданной игры.
46	Архитектурная графика: классика и современность Практическое применение архитектурной графики	Обзор интерфейса SketchUp.	Выполнение упражнений по образцу, создание архитектурной графики.
47	Возможности SketchUp Знакомство с интерфейсом программы	SketchUp возможности интерфейса программы	Выполнение упражнений по образцу, создание архитектурной графики.
48	Панели инструментов в SketchUp	Знакомство с панелью инструментов	Выполнение упражнений по образцу с использованием панели инструментов.
49	Логический механизм интерфейса в SketchUp	Знакомство с интерфейсом SketchUp	Выполнение упражнений по образцу.
50	Дополнительная информация по элементам в SketchUp	Дополнительные элементы и их функции.	Выполнение упражнений по образцу с использованием дополнительных

			элементов.
51	Основные инструменты в SketchUp	Использование основных инструментов	Выполнение упражнений по образцу.
52	Инструменты рисования в SketchUp	Инструменты для прорисовки объектов.	Выполнение упражнений по образцу, прорисовка объектов.
53	Инструменты изменения в SketchUp	Использование инструментов.	Выполнение упражнений по образцу.
54	Инструменты построения в SketchUp	Дополнительные функции инструментов.	Выполнение упражнений по образцу.
55	Моделирование здания в SketchUp	Способы создания моделей зданий.	Выполнение упражнений по образцу, моделирование зданий.
56	Моделирование участка в SketchUp	Способы создания моделей участка.	Выполнение упражнений по образцу, создание моделей участка.
57	Импорт и экспорт в SketchUp	Импорт и экспорт в SketchUp	
58	Работа над проектом	Знакомство с принципами групповой работы. Освоение всех этапов проектной деятельности.	Выполнение упражнений по образцу, создание архитектурной графики.
59	Защита проекта	Презентация «Создание своего игрового мира».	Защита проекта.

Теоретические занятия могут проходить в форме лекции, консультации, демонстрации, презентации, круглого стола или беседы. Практические занятия могут проводиться в форме семинара, мастер-класса, совместной работы, самостоятельной работы, презентации или конкурса. Выбор той или иной формы определяется особенностями учеников.

2.4. Планируемые результаты

Перечень планируемых результатов обучения по программе:

Знать:

- Основные термины и понятия в данной сфере;
- Основные двухмерные форматы переносимой компьютерной графики;
- основные трехмерные форматы компьютерной графики, основные навыки создание проектов.

Уметь:

- спроектировать, изготовить и разместить в сети или подготовить для иной формы представления Scratch проекты;
- Программировать в простом графическом редакторе Scratch (Скретч), составлять и записывать алгоритм для конкретного исполнителя при разработке трехмерных игр;
- создавать графическое отображение идеи проекта в двухмерном и трехмерном формате;
- разрабатывать 3d модель в программе Codu;
- овладеть основными современными методами и средствами создания графического отображения проекта ландшафтной архитектуры.

Владеть:

- способами работы с изученными программами;
- комбинированной техникой создания графического отображения объекта ландшафтной архитектуры, своей идеи и замысла посредством владения комплексом компьютерных программ.

3. Условие реализации программы

Курс “Программирование в среде Scratch ” составляет 163 часа.

Материал курса спланирован так, чтобы быть освоенным за год.

Для мотивации, в качестве демонстрации, а также обучения используется открытый банк работ, накопленный сообществом Scratch и Codu.

Методика преподавания курса предусматривает проведение по каждой новой теме теоретического занятия, выполнение учащимися самостоятельного практического задания на каждом уроке. Содержание задания определяется учителем для каждого ученика индивидуально, с учетом возможностей, интересов и склонностей ребенка. Сложность практической работы ученик выбирает сам. Прохождение курса сопровождается созданием учащимися проектов по предлагаемым темам. Занятия по программе проводятся на основе общих педагогических принципов технологии проектного обучения.

Включает в себя:

- проектирование предполагаемого результата, который достигается в процессе обучения.

Используемые методы:

- объяснительно-иллюстративный;
- проблемный;
- поисковый.

Обучение должно быть доступным (принцип предполагает последовательное усложнение практических заданий - в создании проектов программы);

Принцип систематичности обучения - предполагает такое построение учебного процесса, в ходе которого происходит связывание ранее усвоенного с новым разучиваемым материалом;

Принцип увлекательности (интересности) - успешное осуществление обучения; этот прием делает сам процесс овладения программирования интересным, приносящим чувство радости и удовлетворение.

3.1. Материально – техническое обеспечения

1. Персональный компьютер, нетбук, ноутбук.
2. Операционная система может быть любая.
3. Доступ в интернет со скоростью не менее 1 Мбит/сек.
4. Браузер – любой.
5. Актуальная версия java-plugin в используемом браузере.
6. On-line или Of - line версия Scratch 1.3 или 2.0.
7. Текстовый редактор MS Word 2007 и выше или аналогичный.
8. Растровый графический редактор Paint или аналогичный.
9. Программы, обеспечивающие возможность работы с мультимедийным контентом: воспроизведение видеоизображений, качественный стереозвук в наушниках, речевой ввод с микрофона и др.
10. Программа для просмотра pdf-файлов.
11. Акустические колонки или наушники.
12. Микрофон.

3.2. Кадровое обеспечение программы

Для реализации программы необходимы преподаватели:

- знающие предметную область “Информатика”;
- владеющие методикой преподавания “Информатики и ИКТ” в начальной и средней школе;
- имеющие навыки программирования;
- имеющие навыки программирования на Scratch;
- освоившие материал представляемой программы.

3.3. Учебно-информационное обеспечение

1. Вудкок Дж., Вордерман К. “Программирование для детей”. - М.: МИФ, 2017.
2. Голиков Д. “ Scratch для юных программистов”. - С.П.: БХВ-Петербург, 2017.
3. Голиков Д. “ Scratch для учителей и родителей”. - С.П.: БХВ-Петербург, 2019.
4. Зорина Е.М. «Путешествие в страну Алгоритмию». - М.: ДМК-Пресс, 2016.
М.: Бином, 2016.
5. Маржи М. “ Scratch для детей. Самоучитель по программированию”. - М.: МИФ, 2017.
6. Пашковская Ю. “Творческие задания в среде Scratch. Рабочая тетрадь для 5–6 клас-сов”. - М.: Бином, 2017
7. Ресурсы о программировании на Scratch. URL: <http://scratch4russia.com/store/>. Последняя проверка 13.06.19.
8. Список электронных книг про Scratch URL: <http://scratch4russia.com/альтернативные->
9. Торгашова Ю. “Первая книга юного программиста. Учимся писать программы» - М.: Эра, 2016
10. Школа Scratch [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вернуться домой». URL: <http://letopisi.ru/index.php/cratch>.