

МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ «СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА №5»

РАССМОТРЕНО  
на педагогическом совете  
Протокол № 1  
«27» августа 2021 г.



УТВЕРЖДАЮ  
Директор МАОУ СОШ №5  
И.Г. Попова  
Приказ \* 30 / 4-09  
2021 г.

**РАБОЧАЯ ПРОГРАММА**

**Курса внеурочной деятельности «ОСНОВЫ ПРОГРАММИРОВАНИЯ»**

**для обучающихся 6 классов**

Разработчик:  
Николаева Ольга Евгеньевна  
учитель информатики  
первая квалификационная категория

городской округ Сухой Лог  
2021 год

## I. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Курс внеурочной деятельности «Основы программирования» имеет техническую направленность.

Информатика имеет большое и все возрастающее число междисциплинарных связей, причем как на уровне понятийного аппарата, так и на уровне инструментария. Многие предметные знания и способы деятельности (включая использование средств ИКТ), освоенные обучающимися на базе информатики, находят применение как в рамках образовательного процесса при изучении других предметных областей, так и в иных жизненных ситуациях, становятся значимыми для формирования качеств личности, т. е. ориентированы на формирование метапредметных и личностных результатов. На протяжении всего периода становления школьной информатики в ней накапливался опыт формирования образовательных результатов, которые в настоящее время принято называть современными образовательными результатами.

В настоящее время в России активно строится информационное общество. Одним из критериев признания общества информационным является снижение занятости в сфере производства и увеличение в сфере услуг и информации (поскольку "сырьем" для нефизического труда является именно информация).

Для увеличения количества занятых в информационной сфере людей внеурочная деятельность может предложить корректировку к программе такого фундаментального школьного предмета, как информатика, добавив, например, достаточное количество часов по программированию. После такой подготовки ребенку намного проще адаптироваться к современным требованиям. Более того, некоторые современные информационные гиганты (например, Яндекс) считают, что школьники вполне могут принимать участие в разработках приложений, которые будут полезны многим пользователям. Таким образом, не нужно доказывать необходимость занятий, направленных на обучение программированию, во внеурочной деятельности.

**Отличительные особенности** данной программы от уже существующих программ обучения программированию. В предлагаемой программе учащиеся получают законченный объем знаний определенного уровня по различным направлениям программирования. Затем происходит углубление и расширение знаний по каждому из направлений. Так, например, в направлении "Языки программирования" учащиеся сначала осваивают основные алгоритмические конструкции языков программирования на примере Robot или Scratch, чтобы научиться писать простые программы, а затем учатся работать в команде, составляя одну большую программу из кусочков, написанных разными детьми.

Данная программа «Основы программирования» рассчитана на контингент подростков 11-12 лет.

Исходя из специфики занятий по данной программе, предельная наполняемость учебной группы составляет не более 12 человек

Программа рассчитана 33 учебных часа в год (занятия планируется проводить 1 раз в неделю).

Форма обучения – очная.

Форма реализации: занятия – исследования, занятия – упражнения, творческие работы, игровые проекты.

Режим занятий: 1 раз в неделю по 1 занятию, периодичность занятий – еженедельно. Продолжительность одного занятия 40 минут, включая непосредственно содержательный аспект в соответствии с учебно-тематическим планированием, а также с учетом организационных и заключительных моментов занятия. Время работы за компьютером не более 30 минут.

## II. ЦЕЛИ И ЗАДАЧИ ПРОГРАММЫ

### 2.1. Цель Программы

**Цель программы:** обеспечение достижения планируемых результатов ООП, создание условий для развития проектно-исследовательской компетенции обучающихся, творческие способности в соответствии с интересами и склонностями учащихся.

### 2.2. Задачи Программы

#### **Образовательные задачи:**

- Достижение планируемых результатов ООП в части формирования личностных, метапредметных и предметных результатов обучения;
- ознакомить с основами моделирования и программирования;
- совершенствовать навыки программирования, экспериментирования и умения оценивать современные способы управления;
- обучить соблюдению правил техники безопасности при обращении персональным компьютером,
- научить использовать компьютер как инструмент для игр и приложений;
- сформировать общеучебные умения и навыки на основе средств и методов информатики и ИКТ, в том числе овладение умениями работать с различными видами информации, самостоятельно планировать и осуществлять индивидуальную и коллективную информационную деятельность, представлять и оценивать ее результаты;
- создать условия для пропедевтического (предварительного, вводного, ознакомительного) изучения понятий основного курса школьной информатики, обеспечивающее целенаправленное формирование общеучебных понятий, таких как «объект», «система», «модель», «алгоритм» и др.;
- научить принципам работы с основными прикладными программами и средами программирования.

#### **Развивающие задачи:**

- развивать творческие способности владения компьютером;
- развивать навыки построения моделей и научить основам работы с оборудованием и программным обеспечением;
- обеспечить межпредметную интеграцию знаний и умений, рассматривая прикладные вопросы технической направленности;
- сформировать умение самостоятельно приобретать и применять знания;
- развивать алгоритмическое мышление и творческое воображение.

#### **Воспитательные задачи:**

- воспитать уважение к точным наукам, понимание их жизненной необходимости, стремление к дальнейшему обучению.
- воспитание ответственного и избирательного отношения к информации; развитие познавательных, интеллектуальных и творческих способностей учащихся.

### III. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОГРАММЫ

Программа способствует формированию личностных, регулятивных, познавательных и коммуникативных учебных действий.

#### **Предметные результаты**

##### **Обучающийся научится:**

- знать принципы и структуру Scratch проектов, формы представления и управления информацией в проектах в среде Scratch;
- уметь проектировать, изготовить и разместить в сети или подготовить для иной формы представления Scratch проекты;
- использовать основные алгоритмические конструкции: линейный алгоритм, циклический алгоритм, разветвляющийся алгоритм;
- владеть способами работы с изученными программами;
- применять при создании Scratch проектов основные принципы композиции и колористики;
- осуществлять рефлексивную деятельность, оценивать свои результаты, корректировать дальнейшую деятельность по разработке Scratch проектов;
- владеть специальными знаниями и практическими навыками в области программирования в среде Scratch;
- владеть приемами организации и самоорганизации работы по созданию проектов в среде Scratch;
- иметь положительный опыт коллективного сотрудничества при разработке проектов в среде Scratch;
- иметь опыт коллективной разработки и публичной защиты проектов в среде Scratch.

##### **Обучающийся получит возможность научиться:**

- разрабатывать сценарии новых проектов и игр в среде Scratch;
- разрабатывать и создавать новые спрайты для собственных проектов в среде Scratch;
- владеть навыками структурного программирования для реализации сценариев собственных проектов и игр в среде Scratch;
- разрабатывать процедуры и функции в среде Scratch;
- использовать переменные для передачи данных из одной процедуры в другую;
- создавать собственные проекты Scratch, публиковать их в медиaprостранстве, представлять и защищать их.

##### **Личностные УУД:**

- оценивать свои и чужие поступки, выбирать, как поступить, отвечать за свой выбор
- осознавать важность саморазвития, использовать это знание в жизненных ситуациях
- осознавать целостность мира и многообразие взглядов на него, вырабатывать свои мировоззренческие позиции
- понимать чувства других людей
- идти на взаимные уступки в разных ситуациях
- осознавать свои эмоции, мысли, черты характера, адекватно выражать и контролировать их в общении

##### **Познавательные УУД:**

- находить ответы на вопросы (курса и для решения жизненных задач) в различных источниках информации
- анализировать свои и чужие поступки (действия, чувства; изменения в самом себе)
- обобщать (делать выводы, доказывать), устанавливать причинно-следственные связи (поступков, ситуаций)
- обогатить представления о собственных ценностях, эмоциональном мире человека, их роли в жизни.

### **Регулятивные УУД:**

- определять цель, проблему в учебной деятельности (на занятии)
- выдвигать версии в группе и индивидуально
- определять цель, проблему в жизненно-практической деятельности, оценивать степень и способы достижения цели в жизненных ситуациях, самостоятельно исправлять ошибки
- планировать свою деятельность в учебной и жизненной ситуациях
- прогнозировать последствия собственных и чужих поступков

### **Коммуникативные УУД:**

- излагать свое собственное мнение и позицию с аргументами, фактами
- понимать позицию другого (выраженную в явном и неявном виде: чувства, причины действий, поступков), корректировать свое мнение, достойно признавать его ошибочность
- самостоятельно решать проблемы в общении с взрослыми и друзьями
- преодолевать конфликты, толерантно относиться к другому человеку

### **Метапредметные результаты:**

- умение анализировать;
  - умение работать с текстом;
  - развитие воображения;
- развитие внимания и памяти.

**IV. СОДЕРЖАНИЕ ПРОГРАММЫ**  
**Учебный (тематический) план**

№	Название раздела, темы	Количество часов			Формы аттестации (контроля)
		Теория	Практика	Всего	
1.	Правила техники безопасности. Интерфейс среды программирования Scratch	1		1	Предварительный контроль
2.	Сцена. Редактирование фона	1		1	
3.	Понятие спрайтов		1	1	
4.	Основные скрипты программы Scratch		1	1	
5.	Синий ящик – команды движения		1	1	
6.	Назначение сенсоров		1	1	
7.	Создание программ для передвижения спрайтов по сцене.		1	1	тест
8.	Создание программ для рисования различных фигур	1		1	
9.	Фиолетовый ящик – внешний вид объекта		1	1	
10.	Оживление объекта с помощью добавления костюмов		1	1	
11.	Переупорядочивание костюмов		1	1	
12.	Назначение сенсоров <i>костюм и размер</i>	1		1	
13.	Понятие раскадровки движения		1	1	Промежуточный контроль
14.	Создание программы для управления внешним видом объекта		1	1	
15.	Создание Scratch-историй		1	1	
16.	Лиловый ящик – добавление звуков		1	1	
17.	Озвучивание Scratch-историй		1	1	
18.	Базовая конструкция ветвление		1	1	
19.	Циклы с фиксированным числом повторений	1		1	
20.	Создание программ с использованием циклов		1	1	
21.	Зеленый ящик – операторы		1	1	
22.	Синхронизация работы объектов		1	1	Текущий контроль
23.	Последовательность и параллельность выполнения		1	1	
24.	Последовательные и параллельные потоки	1		1	
25.	Взаимодействие между спрайтами		1	1	
26-27.	Компьютерные игры – вред или польза. Виды компьютерных игр	1	1	2	
28.	Разработка базовых спрайтов для игры		1	1	
29.	Логика создания персонажей для игры		1	1	
30.	Разработка и создание основных спрайтов и их костюмов для будущей игры		1	1	
31.	Синхронизация работы скриптов для разных спрайтов		1	1	
32.	Переход из одной сцены в другую. Создание интерфейса игры		1	1	

33.	Разработка и защита творческого проекта		1	1	Итоговый контроль Зачет
		Всего	7	26	33

### Содержание Программы

№ п/п	Название раздела, кол-во часов	Содержание раздела
1	<b>Интерфейс программы Scratch (1 ч)</b>	<p>Введение. Что такое Scratch. Основные алгоритмические конструкции. Знакомство с интерфейсом программы Scratch. Теория. История создания среды Scratch. Основные базовые алгоритмические конструкции (линейные алгоритмы, с условным оператором, циклического типа с предусловием и постусловием) и их исполнение в среде Scratch. Понятие исполнителя, алгоритма и программы, их назначение, виды и использование. Виды управления исполнителем. Способы записи алгоритма. Основные характеристики исполнителя. Система команд исполнителя. Понятие проект, его структура и реализация в среде Scratch. Основные компоненты проекта Scratch: спрайты и скрипты. Принцип создания анимации и движения объектов. Листинг программы. Сцена. Текущие данные о спрайте. Стил поворота. Закладки. Панель инструментов, Новый спрайт. Координаты мышки. Режим представления. Окно скриптов. Окно блоков. Блоки стека. Блоки заголовков. Блоки ссылок. Самодостаточные и открытые скрипты.</p>
2	<b>Начало работы в среде Scratch (2 ч)</b>	<p>Сцена. Редактирование фона. Добавление фона из файла. Теория. Сцена. Ширина и высота сцены. Текущие координаты объекта. Редактирование текущего фона. Вставка нового фона из файла. Вставка стандартного фона из библиотечного модуля среды. Рисование фона в графическом редакторе. Создание нескольких фонов в одной сцене.</p> <p>Практика. Создание фона сцены на выбранную учащимся тему. Понятие спрайтов. Добавление новых спрайтов. Рисование новых объектов.</p> <p>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.</p> <p>Практика. Создание фона сцены и прорисовка основных спрайтов для Scratch-истории.</p>
3	<b>Основные скрипты программы Scratch (18 ч).</b>	<p>Синий ящик – команды движения. Темно-зеленый ящик – команды рисования.</p> <p>Теория. Команды – <i>идти; повернуться направо (налево); повернуть в направлении; повернуться к; изменить x (y) на; установить x (y) в; если край, оттолкнуться</i>. Принципиаль</p>

ное различие действия команд *идти в* и *плыть в*. Назначение сенсоров *положение x*, *положение y* и *направлении*. Команды – *очистить*, *опустить перо*, *поднять перо*, *установить цвет пера*, *изменить цвет пера на*, *установить цвет пера*, *изменить тень пера*, *установить тень пера*, *изменить размер пера на*, *установить размер пера*, *печать*.

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

Фиолетовый ящик – внешний вид объекта. Оживление объекта с помощью добавления костюмов.

Теория. Костюмы спрайта. Копирование и редактирование костюма спрайта с помощью редактора рисования. Переупорядочивание костюмов. Команды – *перейти к костюму*, *следующий костюм*, *говорить...в течении...секунд*, *сказать*, *думать*, *думать...секунд*, *изменить... эффект на*, *установить эффект...в значение*, *убрать графические эффекты*, *изменить размер на*, *установить размер*, *показаться*, *спрятаться*, *перейти в верхний слой*, *перейти назад на...1 слоев*. Назначение сенсоров *костюм* и *размер*. Понятие раскадровки движения. Изменение костюма спрайта для имитации движения. Практика. Создание программы для управления внешним видом объекта. Создание Scratch-историй с имитацией хождения и движения объектов.

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*, *по- вторять до..* Команды – *когда клавиша...нажата*, *когда щелкнут по*, *ждать...секунд*, *ждать до*, *остановить скрипт*, *остановить все*. Загрузка звуков из стандартной коллекции и из файлов жесткого диска. Запись звука через микрофон. Принципиальная разница работы команд *играть звук* и *играть звук до завершения*. Команды – *остановить все звуки*, *барабану* *играть...тактов*, *оставшиеся...тактов*, *ноту...играть...тактов*, *выбрать инструмент*, *изменить громкость*, *установить громкость*, *изменить темп на*, *установить темп*. Назначение сенсоров *громкость* и *темп*. Практика. Создание программ с элементами управления объектом. Озвучивание Scratch-историй.

Использование в программах условных операторов.

Теория. Базовая конструкция ветвление, назначение, виды (полная и неполная форма). Понятие условия. Изменение порядка выполнения скриптов в зависимости от условия. Разветвление листинга программы. Скрипты условных операторов. Использование неполной формы ветвления в системе Scratch.

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

Функциональность работы циклов. Цикличность выполнения данных пользователя с выводом на экран конечного результата.



4	<b>Работа с несколькими объектами. Синхронизация их работы (4 ч).</b>	<p>13. Последовательность и параллельность выполнения скриптов. Теория. Последовательные и параллельные потоки в программах Scratch. Одновременная и попеременная работа нескольких исполнителей. Практика. Создание Scratch-историй с одновременной и попеременной работой нескольких исполнителей.</p> <p>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 <i>касается</i> и <i>касается цвета</i>. Взаимодействие спрайтов с помощью команд <i>передать</i> и <i>когда я получу</i>. Использование сообщений для создания событий. Практика. Создание Scratch-историй с взаимодействием нескольких исполнителей и неподвижных объектов. Создание Scratch-историй с взаимодействием нескольких исполнителей.</p>
5	<b>Использование программы Scratch для создания мини-игр (7 ч).</b>	<p>15. Виды компьютерных игр. Алгоритмическая разработка листинга программы. Теория. Компьютерные игры – вред или польза. Виды компьютерных игр. Этапы разработки игр программистами. Практика. Алгоритмическая разработка проекта, запись на естественном языке событий и точек взаимодействия героев будущей игры.</p> <p>16. Разработка базовых спрайтов для игры. Формирование базовых скриптов. Теория. Логика создания персонажей для игры. Перевод алгоритма, написанного на естественном языке, в коды Scratch. Практика. Разработка и создание основных спрайтов и их костюмов для будущей игры. Разработка скриптов для спрайтов и объектов.</p> <p>17. Синхронизация работы скриптов для разных спрайтов. Практика. Доработка основного листинга программы с целью установления связей между спрайтами. Тестирование и отладка программы.</p> <p>18. Переход из одной сцены в другую. Создание интерфейса игры. Теория. Односторонний (без возможности вернуться назад) переход из одного пространства в другое. Понятие интерфейса. Элементы интерфейса. Основные принципы дизайна интерфейсов. Обратная связь. Необходимые элементы меню. Практика. Создать программу для перемещения объекта по игровой карте и разработать интерфейс для Scratch-проекта.</p> <p>19. Сообщество Scratch в Интернете. Просмотр и публикация проектов. Теория. Правила работы в сети. Интернет-сообщества. Сообщество Scratch. Регистрация на сайте. Использование заимствованных кодов и объектов. Авторские права. Публикация проектов Scratch. Практика. Регистрация на сайте сообщества Scratch. Просмотр</p>
		проектов сообщества и публикация собственных проектов.

<b>5</b>	<b>Разработка творческого проекта (1 ч)</b>	20. Разработка и защита творческого проекта. Разработка и создание программы с использованием подготовленных материалов. Тестирование и отладка проекта. Зачет.
----------	---	---

### Формы аттестации/контроля и оценочные материалы

Для учащихся предусмотрена безоценочная система изучения курса.

В течение учебного года педагог проводит поэтапную диагностику успешности усвоения программного материала: результат – выполненные работы обучающихся: устные ответы, тестовые работы, упражнения в среде Scratch, программы в среде Scratch, самостоятельные работы, зачет.

Формы, периодичность и порядок проведения текущего контроля успеваемости и промежуточного контроля и итоговой аттестацией учащихся регламентируются учебным (тематическим планом).

### Виды контроля

- **предварительный контроль** (проверка знаний учащихся на начальном этапе освоения Программы). Проводится в начале реализации Программы в виде входного тестирования.
- **текущий контроль** (отслеживание активности обучающихся в выполнении ими практических работ.)
- **итоговый контроль** (заключительная проверка знаний, умений, навыков по итогам реализации Программы). Презентация подготовленных учащимися работ. Зачет

### Формы и содержание итоговой аттестации

Презентация практических работ учащихся.

### Требование к оценке и критерии оценки достижения планируемых результатов

Уровни освоения Программы	Результат
Высокий уровень освоения Программы	Учащиеся демонстрируют высокую заинтересованность в учебной, познавательной и творческой деятельности, составляющей содержание Программы. На итоговом тестировании показывают отличное знание теоретического материала, практическое применение знаний воплощается в качественный продукт
Средний уровень освоения Программы	Учащиеся демонстрируют заметную заинтересованность в учебной, познавательной и творческой деятельности, составляющей содержание Программы. На итоговом тестировании показывают хорошее знание теоретического материала, допуская незначительные ошибки, практическое применение знаний воплощается в качественный продукт
Низкий уровень освоения Программы	Учащиеся демонстрируют низкий уровень заинтересованности в учебной, познавательной и творческой деятельности, составляющей содержание Программы. На итоговом тестировании показывают недостаточное знание теоретического материала, практическая работа не соответствует требованиям

## ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

### Учебно-методическое и организационное обеспечение Программы

- наглядные пособия;
- дидактический и раздаточный материал;
- пособия, таблицы;
- комплекты методической и теоретической литературы в соответствии с направлениями деятельности;
- фото, аудио, видеоматериалы.

### Материально-технические условия реализации Программы.

- учебный кабинет, оснащенный компьютерами и интерактивной доской;
- технические средства обучения: (компьютерное, мультимедийное оборудование, пакет обучающих программ, необходимых при реализации Программы.
- Доступ в Интернет для работы в онлайн среде Scratch.

### Основными видами деятельности являются:

- **информационно-рецептивная**

Информационно-рецептивная деятельность обучающихся предусматривает освоение учебной информации через рассказ педагога, беседу, самостоятельную работу с литературой.

- **репродуктивная**

Репродуктивная деятельность обучающихся направлена на овладение ими умениями и навыками через выполнение образцов изделий, и выполнения работы по заданному технологическому описанию. Эта деятельность способствует развитию усидчивости, аккуратности обучающихся.

- **творческая**

Творческая деятельность предполагает самостоятельную или почти самостоятельную работу обучающихся.

При обучении используются основные методы организации и осуществления учебно-познавательной работы, такие как словесные, наглядные, практические, индуктивные и проблемно-поисковые. Выбор методов (способов) обучения зависит от психофизиологических, возрастных особенностей детей, темы и формы занятий. При этом в процессе обучения все методы реализуются в теснейшей взаимосвязи. Методика проведения занятий предполагает постоянное создание ситуаций успешности, радости от преодоления трудностей в освоении изучаемого материала и при выполнении творческих работ. Этому способствуют совместные обсуждения технологии выполнения заданий, а также поощрение, создание положительной мотивации, актуализация интереса, рекламы своей работ, конкурсы.

Важными условиями творческого самовыражения обучающихся выступают реализуемые в педагогических технологиях идеи свободы выбора. Обучающимся предоставляется право выбора творческих работ и форм их выполнения (индивидуальная, групповая, коллективная), технологий изготовления в рамках изученного содержания.

## Календарно-тематический план программы "Основы программирования"

№ п/п	Месяц	Число	Форма занятия	Тема занятия	Форма контроля
1.	Сентябрь	5	Лекция	Правила техники безопасности. Интерфейс среды программирования Scratch	Предварительный контроль
2.	Сентябрь	12	Лекция	Сцена. Редактирование фона	
3.	Сентябрь	19	Практика	Понятие спрайтов	
4.	Сентябрь	26	Выполнение упражнения	Основные скрипты программы Scratch	
5.	Октябрь	3	Выполнение упражнения	Синий ящик – команды движения	
6.	Октябрь	10	Практика	Назначение сенсоров	
7.	Октябрь	17	Контроль	Создание программ для передвижения спрайтов по сцене.	тест
8.	Октябрь	24	Лекция	Создание программ для рисования различных фигур	
9.	Ноябрь	7	Выполнение упражнения	Фиолетовый ящик – внешний вид объекта	
10.	Ноябрь	14	Выполнение упражнения	Оживление объекта с помощью добавления костюмов	
11.	Ноябрь	21	Выполнение упражнения	Переупорядочивание костюмов	
12.	Ноябрь	28	Лекция	Назначение сенсоров <i>костюм</i> и <i>размер</i>	
13.	Декабрь	5	Контроль	Понятие раскадровки движения	Промежуточный контроль
14.	Декабрь	12	Практика	Создание программы для управления внешним видом объекта	
15.	Декабрь	19	Выполнение упражнения	Создание Scratch-историй	
16.	Декабрь	26	Выполнение упражнения	Лиловый ящик – добавление звуков	
17.	Январь	9	Выполнение упражнения	Озвучивание Scratch-историй	
18.	Январь	16	Выполнение упражнения	Базовая конструкция ветвление	
19.	Январь	23	Лекция	Циклы с фиксированным числом повторов	
20.	Январь	30	Выполнение упражнения	Создание программ с использованием циклов	
21.	Февраль	6	Выполнение упражнения	Зеленый ящик – операторы	
22.	Февраль	13	Контроль	Синхронизация работы объектов	Текущий контроль
23.	Февраль	19	Выполнение упражнения	Последовательность и параллельность выполнения	
24.	Февраль	26	Лекция	Последовательные и параллельные потоки	
25.	Март	20	Практика	Взаимодействие между спрайтами	
26-27.	Март	27, 3	Лекция	Компьютерные игры – вред или польза. Виды компьютерных игр	
28.	Апрель	10	Выполнение упражнения	Разработка базовых спрайтов для игры	

29.	Апрель	17	Выполнение упражнения	Логика создания персонажей для игры	
30.	Апрель	24	Выполнение упражнения	Разработка и создание основных спрайтов и их костюмов для будущей игры	
31.	Май	1	Выполнение упражнения	Синхронизация работы скриптов для разных спрайтов	
32.	Май	8	Контроль	Переход из одной сцены в другую. Создание интерфейса игры	
33-34.	Май	15,22	Разработка проекта	Разработка и защита творческого проекта	Зачет

## Литература

1. Голиков Д.И. «Scratch для юных программистов», «БХВ-Петербург», Санкт-Петербург, 2017.
2. Краля Н. А. Метод учебных проектов как средство активизации учебной деятельности учащихся: Учебно-методическое пособие / Под ред. Ю. П. Дубенского. Омск: Изд-во ОмГУ, 2005. 59 с. Матвеева Н. В. Информатика и ИКТ.: методическое пособие / Н. В. Матвеева, Е. Н. Челак, Н. К. Конопатова, Л. П. Панкратова. М.: БИНОМ. Лаборатория знаний, 2009. 420 с.
3. Патаракин Е. Д. Учимся готовить в среде Скретч (Учебно-методическое пособие). М: Интуит.ру, 2008. 61 с.
4. Скретч [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вернуться домой». URL: <http://letopisi.ru/index.php/Скретч>
5. Скретч онлайн [Электронный ресурс] // Онлайн среда программирования Scratch — URL: <https://scratch.mit.edu/>

## Электронные ресурсы

1. Методические разработки (презентации, видеоуроки, flash-ролики).
2. Сетевые ресурсы Scratch <https://scratch.mit.edu/>.
3. Видеохостинг Youtube (видеоуроки «работа в среде Scratch») [https://www.youtube.com/watch?v=rkwGkko\\_LSU&list=PLxiU3nwEQ4PH7h8vaYenOfxAVZ3qDycWc&index=2](https://www.youtube.com/watch?v=rkwGkko_LSU&list=PLxiU3nwEQ4PH7h8vaYenOfxAVZ3qDycWc&index=2) .