

Муниципальное бюджетное общеобразовательное учреждение  
«Средняя общеобразовательная школа № 17 им. И.Л. Козыря пос. Шаумянского»

**Согласовано:**

Руководитель Центра «Точка Роста»  
МБОУ СОШ №17 им. И.Л. Козыря  
пос. Шаумянского

 М.А. Сушкова

«30» 08

**Утверждено:**

Директор МБОУ СОШ №17  
им. И.Л. Козыря пос. Шаумянского

 Г.А. Ривчак

Приказ от «08» 08 № 193



## РАБОЧАЯ ПРОГРАММА

**«Информатика. Программирование на примере графического языка Scratch»  
для 6 класса с использованием оборудования центра «Точка роста»**

*Срок реализации программы - 1 год*

*Группа учащихся - 6 классы*

*Программа рассчитана - на 2 часа в неделю, 68 часов за год.*

Основное назначение курса — изучение алгоритмов и исполнителей, первое знакомство с основными алгоритмическими конструкциями, используемыми в языках программирования; получение позитивного опыта отладки и написания первых завершённых программных продуктов.

Программа курса предполагает знакомство с основными понятиями, используемыми в языках программирования высокого уровня, решение большого количества творческих задач, многие из которых моделируют процессы и явления из таких предметных областей, как информатика, алгебра, геометрия, география, физика, русский язык и др. Многие задания составлены таким образом, чтобы они решались методами учебно-исследовательской и проектной деятельности. Большинство заданий встречаются в разных темах для того, чтобы показать возможности решения одной и той же задачи или проблемы различными средствами, обеспечивающими достижение требуемого результата, что в итоге приведет к способности выбирать оптимальное решение данной задачи или проблемы.

Основные личностные результаты, формируемые в процессе освоения программы курса «Программирование Скрэтч»– это:

- формирование ответственного отношения к учению, способности довести до конца начатое дело на примере завершённых творческих учебных проектов;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстративной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки, благодаря реализованным проектам;
- формирование коммуникативной компетенции в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- развитие эстетического сознания через творческую деятельность на базе иллюстрированной среды программирования.

К основным метапредметным результатам (осваиваемым обучающимися межпредметным понятиям и универсальным учебным действиям, способности их использования как в учебной, так и в познавательной и социальной практике), формируемые в процессе освоения программы курса «Программирование Скрэтч» можно отнести:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата, понимая, что в программировании длинная программа не значит лучшая программа;
- умение оценивать правильность решения учебно-исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;

- владение основами самоконтроля, принятия решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- ИКТ-компетенцию;
- умение сотрудничества и совместной деятельности со сверстниками в процессе проектной и учебно-исследовательской деятельности.

Основные предметные результаты, формируемые в процессе изучения курса «Юный Программирование Scratch» направлены на:

- осознание значения математики и информатики в повседневной жизни человека;
- формирование представлений об основных предметных понятиях — «информация», «алгоритм», «модель» и их свойствах;
- развитие логических способностей и алгоритмического мышления, умений составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- развитие представлений о числах, числовых системах;
- овладение символьным языком алгебры, умение составлять и использовать сложные алгебраические выражения для моделирования учебных проектов, моделировать реальные ситуации на языке алгебры;
- развитие пространственных представлений, навыков геометрических построений и моделирования таких процессов, развитие изобразительных умений с помощью средств ИКТ;
- формирование информационной и алгоритмической культуры, развитие основных навыков использования компьютерных устройств и программ;
- формирование умения соблюдать нормы информационной этики и права.

Представим более детально содержание курса «Юный программист».

### **Раздел 1. Знакомство с программной средой Scratch – 2 ч**

Свободное программное обеспечение. Авторы программной среды Scratch. Параметры для скачивания и установки программной среды на домашний компьютер.

Основные элементы пользовательского интерфейса программной среды Scratch. Внешний вид рабочего окна. Блочная структура систематизации информации. Функциональные блоки. Блоки команд, состояний, программ, запуска, действий и исполнителей. Установка русского языка для Scratch.

Создание и сохранение документа. Понятия спрайта, сцены, скрипта. Очистка экрана.

Основной персонаж как исполнитель программ. Система команд исполнителя (СКИ). Блочная структура программы. Непосредственное управление исполнителем.

Библиотека персонажей. Сцена и разнообразие сцен, исходя из библиотеки данных. Систематизация данных библиотек персонажей и сцен. Иерархия в организации хранения костюмов персонажа и фонов для сцен. Импорт костюма, импорт фона.

*Аналитическая деятельность:*

- выделять аппаратное и программное обеспечение компьютера;
- определять технические устройства для ввода и вывода информации;
- понимать иерархическую организацию библиотеки данных программной среды;
- выделять путь к элементам библиотеки;

- выделять фрагменты изображения для дальнейшей работы с ними;
- планировать работу по созданию сложных изображений путем копирования и масштабирования простых;
- выбирать наиболее подходящий инструмент графического редактора для создания фрагмента изображения;
- различать верхний и нижний цвета изображения;
- придумывать и создавать различные градиенты для заливки замкнутой области;
- планировать создание симметричных изображений.

*Практическая деятельность:*

- выбирать и запускать программную среду Scratch;
- работать с основными элементами пользовательского интерфейса программной среды;
- изменять размер и перемещать окно программы, выбирать необходимый режим окна;
- вводить имя файла с помощью клавиатуры;
- выбирать необходимый файл из нужной папки библиотеки программы;
- создавать, копировать, переименовывать, перемещать, копировать и удалять файлы;
- соблюдать требования техники безопасности при работе в компьютерном классе.

## **Раздел 2. Компьютерная графика – 10 ч**

Компьютерная графика. Векторные и растровые графические редакторы. Встроенный растровый графический редактор. Основные инструменты графического редактора — кисточка, ластик, заливка (цветом или градиентом), рисование линий, прямоугольников, квадратов, эллипсов и окружностей, выбор фрагмента изображения и отражение его по горизонтали или вертикали, использование инструмента печать для копирования выделенной области изображения, работа с текстом. Масштаб фрагмента изображения. Палитра цветов, установка цвета переднего плана и фона, выбор цвета из изображения с помощью инструмента пипетка. Изменение центра костюма. Изменение размера костюма.

Основные возможности изменения внешнего вида исполнителя: 1) использование встроенной библиотеки данных путём импорта её элемента; 2) редактирование выбранного элемента с помощью инструментов встроенного растрового графического редактора; 3) создание собственных изображений в других программах (например, LibreOfficeDraw) и импортирование их в программную среду Scratch.

Знакомство с основными графическими примитивами векторного редактора LibreOfficeDraw. Возможность создания геометрических фигур без внутренней заливки, но с текстовым блоком внутри. Стрелки, их направление.

*Аналитическая деятельность:*

- выделять фрагменты изображения для дальнейшей работы с ними;
- планировать работу по созданию сложных изображений путем копирования и масштабирования простых;
- выбирать наиболее подходящий инструмент графического редактора для создания фрагмента изображения;
- различать верхний и нижний цвета изображения;

- придумывать и создавать различные градиенты для заливки замкнутой области;
- планировать создание симметричных изображений.

*Практическая деятельность:*

- использовать простейшие растровые и векторные редакторы для создания и редактирования изображений;
- изменять центр изображения;
- вносить изменения в изображения из встроенной библиотеки;
- создавать сложные графические объекты путем копирования и модификации простых объектов и их фрагментов,
- использовать возможности работы с цветом.

### **Раздел 3. Алгоритмы и исполнители – 28 ч**

Алгоритм. Понятие алгоритма как формального описания последовательности действий исполнителя, приводящих от исходных данных к конечному результату. Схематическая запись алгоритма. Использование геометрических фигур для схематической записи алгоритма. Создание блок-схем в свободном векторном редакторе LibreOfficeDraw.

#### ***Линейные алгоритмы***

Основные признаки линейного алгоритма. Схематическое описание линейного алгоритма. Геометрические примитивы, используемые для описания линейного алгоритма.

Программное управление исполнителем. Создание программ для перемещения исполнителя по экранному полю. Понятие поворота исполнителя в определенное направление. Прямой угол. Поворот исполнителя на прямой угол по часовой стрелке и против часовой стрелки.

Создание программ для рисования линий. Изменение цвета и толщины рисуемой линии. Особенности пунктирной линии. Написание программы для исполнителя, чтобы он оставлял пунктирную линию при перемещении по экранному полю.

Прямоугольник, квадрат — основные черты. Написание программ для движения исполнителя вдоль сторон квадрата, прямоугольника. Внесение изменений в программу рисования квадрата, если необходимо получить другой размер стороны квадрата.

Прерывание программы.

#### ***Циклические алгоритмы***

Многократное повторение команд как организация цикла. Особенности использования цикла в программе. Упрощение программы путём сокращения количества команд при переходе от линейных алгоритмов к циклическим.

Схематическая запись циклического алгоритма.

Типы циклических алгоритмов. Основные конструкции программной среды, используемые для написания программ исполнителем с применением циклов.

Конечный цикл. Сокращение программы для исполнителя, рисующего линии, квадраты, прямоугольники при использовании цикла. Программа исполнителя для рисования нескольких однотипных геометрических фигур, например, нескольких квадратов из одной вершины, но с различным значением стороны.

Конструкции программной среды спрятаться/показаться. Выполнение программы исполнителем, не показанным на поле выполнения программы.

Написание и отладка программ с применением конструкции цикл в цикле.

Бесконечный цикл. Повторяющаяся смена внешности исполнителя для имитации движения персонажа. Использование бесконечного цикла для создания анимации.

Получение различного эффекта воспроизведения программы при изменении костюма исполнителя Scratch.

#### ***Параллелизм в программной среде***

Использование нескольких исполнителей. Копирование программы одного исполнителя другим. Выполнение одинаковых программ разными исполнителями с использованием различных начальных условий. Параллельное выполнение однотипных действий. Принцип суперкомпьютерных технологий. Таймер для вычисления времени выполнения программы. Уменьшение показаний таймера при использовании параллельных вычислений.

Интерактивность программ. Возможность организации диалога между исполнителями. Операторы для слияния текстовых выражений.

Взаимодействие исполнителей путём касания друг друга или цвета. Использование сенсоров при взаимодействии исполнителей. Задержка выполнения программы.

Работа исполнителей в разных слоях изображения.

#### ***Ветвление в алгоритмах***

Использование ветвления при написании программ. Короткая форма. Полная форма условного оператора. Конструкции ветвления для моделирования ситуации.

Цикл пока. Повторение команд исполнителя при выполнении определенного условия.

#### ***Последовательное выполнение фрагментов программы разными исполнителями***

Типы исполнителей программной среды Scratch. Системы команд исполнителей. Различные системы команд для разных типов исполнителей.

Управление событиями. Передача сообщений исполнителям для выполнения определенной последовательности команд.

Передача управления между различными типами исполнителей.

#### ***Аналитическая деятельность:***

- придумывать задачи для исполнителей программной среды;
- выделять ситуации, для описания которых можно использовать линейный алгоритм, алгоритм с ветвлениями, повторениями;
- определять эффективный способ решения поставленной задачи;
- находить параллельности в выполняемых действиях и программировать их с помощью нескольких исполнителей;
- планировать последовательность событий для заданного проекта.

#### ***Практическая деятельность:***

- составлять и отлаживать программный код;
- использовать конструкции программной среды для создания линейных, разветвленных и циклических алгоритмов;
- организовывать параллельные вычисления;
- организовывать последовательность событий программы, передачу управления от одних исполнителей другим.

#### **Раздел 4. Проектная деятельность и моделирование процессов и систем – 26 ч**

Мультимедийный проект. Описание сюжетных событий. Анимация. Создание эффекта анимации с помощью последовательной смены изображений. Имитационные модели. Интерактивные проекты. Игры.

*Аналитическая деятельность:*

- создавать план появления событий для отражения определенной темы;
- выбирать иллюстративный материал из встроенной библиотеки;
- выбирать метод анимации для конкретной задачи;
- планировать последовательность событий для создания эффекта анимации по выбранному сценарию.

*Практическая деятельность:*

- использовать возможности программной среды Scratch для создания мультимедийных проектов;
- создавать имитационные модели, интерактивные проекты и игры средствами программной среды.

### **Тематическое планирование курса «Программирование СКРЭТЧ».**

№ урока	дата	Тема урока
1		Знакомство со средой Scratch. Внешний вид среды, поля. Анимация.
2		Исполнитель Scratch, цвет и размер пера.
3-4		Основные инструменты встроенного растрового графического редактора.
5-6		Линейный алгоритм. Создание блок-схемы. Основные графические примитивы векторного редактора LibreOfficeDraw.
7-8		Линейный алгоритм. Рисование линий исполнителем Scratch.
9-10		Линейный алгоритм. Исполнитель Scratch рисует квадраты и прямоугольники линейно.
11-12		Конечный цикл. Исполнитель Scratch рисует квадраты, линии.
13-14		Конечный цикл. Исполнитель Scratch рисует несколько линий и фигур. Копирование фрагментов программы.
15-16		Циклический алгоритм. Цикл в цикле. Вложенные и внешние циклы.
17-18		Цикл в цикле. Повторение пунктирной линии с поворотом. Блок-схема цикла.
19-20		Бесконечный цикл. Анимация исполнителя Scratch на основе готовых костюмов.
21-22		Сцена как исполнитель. Создаем модель таймера.
23-24		Бесконечный цикл. Одна программа для исполнителя Scratch, но разные костюмы.
25-26		Одинаковые программы для несколько исполнителей.

№ урока	дата	Тема урока
27-28		Несколько исполнителей. Параллельное выполнение действий для ускорения процесса выполнения программы.
29-30		Разбиение программы на части для параллельного выполнения исполнителями. Таймер. Уменьшение показаний таймера при параллельных вычислениях.
31-32		Два исполнителя со своими программами. Мини-проект «Часы».
33-34		Алгоритмы с ветвлением. Условие ЕСЛИ. Два исполнителя.
35-36		Цикл при условии. Мини-проект «Шарики в лабиринте»
37-38		Цикл при условии. Исполнитель определяет цвета.
39-40		Цикл при условии. Исполнители в разных слоях. Мини-проект «Самолет сквозь облака».
41-42		Перемещение исполнителя из одного слоя в другой. Действия исполнителей в разных слоях. Мини-проект «Дорога».
43-44		Алгоритмы с ветвлением. Условие ЕСЛИ. Взаимодействие исполнителей. Блок-схема с условием.
45-46		Сцена как исполнитель. Последовательное выполнение команд исполнителями.
47-48		Алгоритмы с ветвлением. Программирование клавиш.
49-51		Алгоритмы с ветвлением. Если касается цвета.
52-53		Интерактивность исполнителей. Создание мини-проекта «Лабиринт».
54-56		Игра «Лабиринт». Усложнение.
57		Моделирование ситуации. Мини-проект «Пешеходный переход».
58		Моделирование ситуации. Интерактивность исполнителей. Мини-проект «Водолей».
59		Моделирование. Учебные модели «Рисующий карандаш», «Затухание».
60		Моделирование. Тестовая модель «Комнатные растения».
61		Моделирование. Обучающий проект по маршрутам географических открытий.
62-68		Выполнение проекта
69-70		Защита проектов

В результате освоения курса «Программирование Scratch » школьники получают представление о:

- свободно распространяемых программах;
- функциональном устройстве программной среды Scratch и основных структурных элементах пользовательского интерфейса;
- назначении и использовании основных блоков команд, состояний, программ;
- правилах сохранения документа и необходимости присвоения правильного имени;



- возможности и способах отладки написанной программы;
- сущности понятий «спрайт», «сцена», «скрипт»;
- исполнителях и системах их команд, возможности непосредственного управления исполнителем;
  - наличии заготовок для персонажей и сцен в соответствующих библиотеках, иерархическом устройстве библиотек и возможности импортирования их элементов;
  - возможности использования встроенного растрового редактора, наличии и назначении основных инструментов;
  - использовании других программ (например, LibreOfficeDraw) для создания собственных изображений;
  - алгоритме как формальном описании последовательности действий исполнителя, приводящих от исходных данных к конечному результату;
    - использовании схематического описания алгоритма;
    - программном управлении исполнителем и линейных алгоритмах;
    - написании программ для исполнителей, создающих геометрические фигуры на экране в процессе своего перемещения;
    - необходимости программного прерывания;
    - использовании циклических команд при необходимости повторений однотипных действий;
      - видах циклических алгоритмов и их применении;
      - достижении эффекта перемещения путем использования циклов;
      - возможности распараллеливания однотипных действий за счёт использования нескольких исполнителей;
        - организации интерактивности программ;
        - возможности взаимодействия исполнителей между собой, в различных слоях изображения;
          - видах и формах разветвленных алгоритмов, включая циклы с условием;
          - управлении событиями.
    - использовании метода проектов для моделирования объектов и систем;
    - возможности описания реальных задач средствами программной среды;
    - создании анимационных, игровых, обучающих проектов, а также систем тестирования в программной среде Scratch.

Школьники будут уметь:

- самостоятельно устанавливать программную среду на домашний компьютер;
- изменять некоторые стандартные установки пользовательского интерфейса (например, язык отображения информации);
- использовать различные способы отладки программ, включая пошаговую отладку;
- уверенно использовать инструменты встроенного графического редактора, включая работу с фрагментами изображения и создание градиентов;
  - создавать собственные изображения в других программах (например, LibreOfficeDraw) и импортировать их в программную среду Scratch;
  - использовать графические примитивы векторного редактора LibreOfficeDraw для создания объектов;

- создавать изображения из пунктирных и штрих-пунктирных линий с изменением цвета и толщины линии;
- упрощать программы за счёт использования циклических команд и применять их;
- составлять простые параллельные алгоритмы;
- создавать программы и игры с использованием интерактивных технологий;
- моделировать ситуации с использованием необходимых форм ветвления алгоритма, включая цикл по условию;
- передавать сообщения исполнителям для выполнения последовательности команд (включая разные типы исполнителей).
- планировать и создавать анимации по определенному сюжету;
- создавать игры, используя интерактивные возможности программной среды Scratch;
- планировать и создавать обучающие программы для иллюстрации пройденного материала других предметных областей;
- продумывать и описывать интерактивное взаимодействие для создания простейших тренажеров;
- подходить творчески к построению моделей различных объектов и систем.