

# ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ

## Государственное бюджетное общеобразовательное учреждение «Школа №171»

Принято на заседание  
педагогического совета  
Протокол № 1  
От 30.08. 2017 г.

«УТВЕРЖДАЮ»  
Директор ГБОУ Школа №171  
  
Л.П. Карпенко  
«01» сентября 2017 г.  
Приказ от 01.09.2017 № 8/1



## ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

### «Основы программирования»

*Направленность: естественнонаучная*

*Уровень программы: ознакомительный*

*Возраст учащихся: 10-12 лет*

*Срок реализации: 9 месяцев (36 часов)*

*Автор-составитель:*

*Авчуринская Наталья Ивановна,  
учитель информатики*

Москва

## **Пояснительная записка**

Подготовка школьников к изучению программирования во многих странах мира начинается с различных программных сред, одна из которых – программная среда SCRATCH, разработанная MIT (<http://scratch.mit.edu/>)

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.).

Курс «Основы программирования на Scratch» посвящён знакомству учащихся 10-12 лет с основными алгоритмическими конструкциями, развитию логических способностей школьников на базе среды программирования SCRATCH.

### **1.1. Цели реализации курса:**

Образовательная: получить представление об алгоритмах и исполнителях, основных алгоритмических конструкциях языков программирования; формировать компьютерную грамотность;

Развивающая: обеспечить высокую мотивацию к проектной деятельности и дальнейшему изучению языков программирования;

Воспитательная: вооружение учащихся правильным методологическим подходом к познавательной и практической деятельности; воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

### **1.2. Задачи курса.**

Для реализации поставленных целей обучающийся научится:

- самостоятельно устанавливать программную среду на домашний компьютер;
- использовать различные способы отладки программ, включая пошаговую;
- уверенно использовать инструменты встроенного графического редактора, в т. ч. работать с фрагментами изображений и создавать градиенты;
- использовать графические примитивы векторного редактора LibreOffice Draw для создания изображений и импортировать их в программную среду Scratch;

- использовать команды организации цикла для оптимизации программ исполнителей;
- составлять программы для выполнения параллельных алгоритмов;
- использовать интерактивные возможности среды Scratch для создания программ и игр;
- применять различные формы ветвления алгоритмов при моделировании ситуаций, включая цикл по условию;
- планировать и создавать анимацию заданного сюжета;
- моделировать и создавать обучающие программы, иллюстрирующие пройденный материал, изучаемый по программам других предметов;
- моделировать интерактивное взаимодействие с исполнителями для создания простейших тренажеров;
- использовать творческий подход к построению моделей различных объектов и систем.

## **2. Общая характеристика курса.**

Элективный курс «Основы программирования со SCRATCH» рассчитан на 34 часа и посвящён знакомству с основными понятиями, используемыми в языках программирования высокого уровня, решению 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.

Данный курс рассчитан на обучающихся 5–6-х классов.

## **3. Основные разделы курса.**

Раздел 1. Знакомство с программной средой Scratch.

Раздел 2. Компьютерная графика.

Раздел 3. Алгоритмы и исполнители.

Раздел 4. Проектная деятельность и моделирование процессов и систем.

### **3.1. Перечень форм организации учебной деятельности обучающихся.**

Программа может быть использована как дополнительный модуль к основной программе по информатике при использовании часов учебного плана из предметной области технология.

Курс рекомендуется реализовывать как в урочной, так и во внеурочной формах деятельности обучающихся. Программа курса имеет инженерно-

техническую направленность. Многие задания составлены таким образом, чтобы они решались методами учебно-исследовательской и проектной деятельности.

Большинство заданий встречаются в различных темах для того, чтобы показать возможности решения одной и той же задачи или проблемы различными средствами, обеспечивающими достижение требуемого результата, что в итоге должно привести к выработке способности оптимизировать решение поставленной задачи или проблемы.

#### **4. Личностные, метапредметные и предметные результаты освоения учебного курса «Основы программирования со SCRATCH»**

##### **4.1. Личностные и метапредметные результаты**

###### **4.1.1. Личностные результаты:**

- 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 иллюстрированной среде программирования, мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности 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;
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

###### **4.1.2. Метапредметные результаты:**

- 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 критически 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;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

#### **4.2. Предметные результаты:**

- осознание значения математики и информатики в повседневной жизни человека;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развитие логических способностей и алгоритмического мышления, умения 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.

## **5. Содержание учебного курса «Основы программирования со SCRATCH»**

### **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 «спрайт», «сцена», «скрипт». Очистка экрана. Основной персонаж как исполнитель программ. Система команд исполнителя (СКИ). Блочная структура программы. Непосредственное управление исполнителем.

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

#### *Аналитическая деятельность:*

- определять технические устройства для ввода и вывода информации;
- понимать иерархическую организацию библиотек исполнителей программной среды;
- выделять путь к элементам библиотеки;

#### *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;
- создавать, копировать, переименовывать, перемещать, копировать и удалять файлы;
- соблюдать требования техники безопасности при работе в компьютерном классе.

## **Раздел 2. Компьютерная графика – 5 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 изображения с помощью инструмента «Пипетка». Изменение центра костюма. Изменение размера костюма.

Основные способы изменения внешнего вида исполнителя:

- 1) использование встроенной библиотеки исполнителей путём импорта её элемента;
- 2) редактирование выбранного элемента с помощью инструментов встроенного растрового графического редактора;
- 3) создание собственных изображений в других программах (например, LibreOffice Draw) и импортирование их в программную среду Scratch.

Знакомство с основными графическими примитивами векторного редактора LibreOffice Draw. Создание 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 путём копирования и масштабирования простых;
- различать верхний и нижний цвета изображения;

- придумывать и создавать различные градиенты для заливки замкнутой области;

- планировать создание симметричных объектов с использованием возможности отражения фрагмента изображения.

#### *Практическая деятельность:*

- использовать простейшие растровые и векторные редакторы для создания и редактирования изображений;

- изменять центр изображения костюма исполнителя;

- редактировать изображения встроенной библиотеки исполнителей;

- создавать сложные графические объекты путем копирования и модификации простых объектов и их фрагментов;

- использовать возможности работы с цветом.

### **Раздел 3. Алгоритмы и исполнители – 14 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 LibreOffice Draw.

#### *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 «поворот исполнителя в определённое направление». Угол поворота  $90^\circ$ . Поворот исполнителя на  $90^\circ$  по часовой или против часовой стрелки.

Создание программ для рисования линий. Изменение цвета и толщины рисуемой линии. Особенности пунктирной линии. Программа 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.



Редактирование программы рисования квадрата для получения квадрата другой с другим размером стороны.

Прерывание программы.

### *Циклические алгоритмы*

Организация циклического алгоритма при необходимости многократного повторения команд. Особенности использования цикла в программе. Оптимизация 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 исполнителям с применением циклов.

Конечный цикл. Оптимизация 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 в 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 различных эффектов воспроизведения программы за счёт изменения костюмов исполнителей Scratch.

### *Параллелизм в программной среде*

Использование нескольких исполнителей. Копирование программ исполнителей. Выполнение одинаковых программ разными исполнителями при использовании различных начальных условий. Параллельное выполнение однотипных действий. Принцип суперкомпьютерных технологий. Таймер для вычисления времени выполнения программы.

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

Взаимодействие исполнителей. Сенсоры касания объектов или цвета. Использование сенсоров при взаимодействии исполнителей. Задержка выполнения программы.

Слои изображения. Выполнение команд исполнителями 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, разветвлённых и циклических алгоритмов;

- организовывать параллельные вычисления;
- моделировать последовательность событий программы, передачу управления от одних исполнителей другим.

#### **Раздел 4. Проектная деятельность и моделирование процессов и систем – 12 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.

Курс является практикоориентированным, поэтому каждое занятие содержит как теоретический, так и практический материал.

#### **5. Тематическое планирование**

Номер занятия	Тема урока
1	Знакомство со средой Scratch. Внешний вид среды, поля. Анимация
2	Исполнитель Scratch, цвет и размер пера
3	Основные инструменты встроенного растрового графического редактора
4	Линейный алгоритм. Создание блок-схемы. Основные графические примитивы векторного редактора LibreOffice Draw
5	Линейный алгоритм. Рисование линий исполнителем Scratch
6	Линейный алгоритм. Исполнитель Scratch рисует квадраты и

Номер занятия	Тема урока
	прямоугольники линейно
7	Конечный цикл. Исполнитель Scratch рисует квадраты, линии. Блок-схема цикла
8	Конечный цикл. Исполнитель Scratch рисует несколько линий и фигур. Копирование фрагментов программы
9	Циклический алгоритм. Цикл в цикле. Вложенные и внешние циклы
10	Цикл в цикле. Повторение пунктирной линии с поворотом
11	Бесконечный цикл. Анимация исполнителя Scratch на основе готовых костюмов
12	Сцена как исполнитель. Создаем модель таймера
13	Бесконечный цикл. Одна программа для исполнителя Scratch, но разные костюмы
14	Одинаковые программы для несколько исполнителей
15	Несколько исполнителей. Параллельное выполнение действий исполнителями
16	Разбиение программы на части для параллельного выполнения исполнителями. Таймер. Звук
17	Два исполнителя со своими программами. Мини-проект «Часы»
18	Алгоритмы с ветвлением. Условие ЕСЛИ. Два исполнителя. Блок-схема алгоритма с условием
19	Цикл при условии. Мини-проект «Шарики в лабиринте»
20	Цикл при условии. Исполнитель определяет цвета
21	Цикл при условии. Исполнители в разных слоях. Мини-проект «Самолет сквозь облака»
22	Перемещение исполнителя из одного слоя в другой. Действия исполнителей в разных слоях. Мини-проект «Дорога»
23	Алгоритмы с ветвлением. Условие ЕСЛИ. Взаимодействие исполнителей
24	Сцена как исполнитель. Последовательное выполнение команд исполнителями
25	Алгоритмы с ветвлением. Программирование клавиш
26	Алгоритмы с ветвлением. ЕСЛИ касается цвета
27	Интерактивность исполнителей. Создание мини-проекта «Лабиринт»
28	Игра «Лабиринт». Усложнение
29	Моделирование ситуации. Мини-проект «Пешеходный переход»
30	Моделирование ситуации. Интерактивность исполнителей. Мини-проект «Водолей»
31	Моделирование. Учебные модели «Рисующий карандаш»,

Номер занятия	Тема урока
	«Затухание»
32	Моделирование. Модель теста «Комнатные растения»
33	Моделирование. Обучающий проект по маршрутам географических открытий
34	Обобщение пройденного за курс

## 6. Ожидаемые результаты

По окончании курса учащиеся будут уметь:

- самостоятельно устанавливать программную среду на домашний компьютер;
- использовать различные способы отладки программ, включая пошаговую;
- работать во встроенном графическом редакторе;
- использовать графические примитивы векторного редактора LibreOffice Draw;
- использовать команды организации цикла для оптимизации программ исполнителей;
- составлять программы для выполнения параллельных алгоритмов;
- использовать интерактивные возможности среды Scratch для создания программ и игр;
- применять различные формы ветвления алгоритмов при моделировании ситуаций, включая цикл по условию;
- планировать и создавать анимацию заданного сюжета;
- моделировать и создавать обучающие программы, иллюстрирующие пройденный материал, изучаемый по программам других предметов;
- моделировать интерактивное взаимодействие с исполнителями для создания простейших тренажеров;
- использовать творческий подход к построению моделей различных объектов и систем.

## 7. Учебно-методическое и материально-техническое обеспечение образовательного процесса

### 7.1. Учебно-методическое обеспечение

1. Патаракин Е.Д. Учимся готовить в среде Скретч: учебно-методическое пособие) — И.: Интуит.ру, 2007. – 61 с.: ил.

2. Босова Л.Л., Методика применения интерактивных сред для обучения младших школьников программированию / Л.Л. Босова, Т.Е. Сорокина // Информатика и образование. – №7 (256). – 2014.
3. Сорокина Т.Е. Пропедевтика программирования в курсе информатики 5–6 классов на базе среды SCRATCH: мат. конф./ XII открытая Всерос. конф. (15–16 мая 2014 г.). – Казань: Казанский (Приволжский) федеральный университет, 2014. – 369 с. – С. 313–315.

## **7.2. Компьютерная техника и интерактивное оборудование**

Помещение для занятий должно быть оборудовано персональными компьютерами для каждого обучающегося. Для полноценного изучения мультимедийных возможностей среды программирования необходимо оснащение компьютеров оборудованием для воспроизведения звука.

Наличие проектора и колонок позволит повысить наглядность представления обучающих материалов. При использовании интерактивной доски появляются дополнительные методы организации образовательного процесса.