

Муниципальное бюджетное учреждение дополнительного образования «Сунтарский центр детского творчества» Н.М.Родионовой муниципального района «Сунтарский улус (район) Республики Саха (Якутия)

---

Принята на заседании  
методического (педагогического)  
совета от «04»06.2021 г. Протокол №  
8

Утверждаю:

Директор МБУ ДО СЦДТ

А.В. Антонова

« 31 » 08

2021 г.



## ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

технической направленности

«Программирование в Scratch»

Срок реализации: 1 год

Возраст детей: 7-12 лет

Составитель: Иванов Руслан Рудольфович  
педагог дополнительного образования

Год разработки 2021 год

## Пояснительная записка

**Направленность** программы - научно-техническая. Обучение по данной программе направлено на приобретение учащимися базовых знаний в области программирования и привлечение их к современным информационным технологиям. Она основана на дополнительных образовательных программах Ротландского сетевого университета

### Обоснование необходимости реализации программы

Программа «Программирование в Scratch» построена таким образом, чтобы помочь учащимся заинтересоваться программированием и найти ответы на вопросы, с которыми им приходится сталкиваться в повседневной жизни при работе с большим объемом информации; при решении практических и жизненных задач. Курс позволяет создавать собственные программы для решения конкретной задачи. Это является отличительной особенностью данной программы.

Актуальность проектной деятельности сегодня осознаётся всеми. ФГОС нового поколения требует использования в образовательном процессе технологий деятельного типа, методы проектно-исследовательской деятельности определены как одно из условий реализации основной образовательной программы образования.

Следует иметь в виду, что возрастные особенности обучающихся среднего возраста не позволяют в полной мере реализовать проведение полноценных научных исследований. Раннее включение в организованную специальным образом проектную деятельность творческого характера позволяет сформировать у школьника познавательный интерес и исследовательские навыки.

Организация научно-познавательной деятельности обучающихся требует использования инструмента (средства) для выполнения как исследовательских, так и творческих проектов. В качестве такого инструмента я вижу среду программирования Scratch.

### Цели программы

Формирование основ знаний о программировании, подготовка учащихся к применению современных технологий как инструмента для решения практических научно-технических задач.

### Результат программы

Профессиональные и предметные:

- знание и понимание принципов программирования;

Универсальные:

- наличие высокого познавательного интереса учащихся,
- умение ориентироваться в информационном пространстве, продуктивно использовать техническую литературу для поиска сложных решений;
- умение ставить вопросы, связанные с темой проекта, выбор наиболее эффективных решений задач в зависимости от конкретных условий;
- наличие критического мышления;

- проявление технического мышления, познавательной деятельности, творческой инициативы, самостоятельности;
- способность творчески решать технические задачи;
- способность правильно организовывать рабочее место и время для достижения поставленных целей.

### **Материально-техническая база**

Компьютер с ОС Windows 7, 8, 10;

Программа Scratch (версия не ниже 3)

## Учебно-тематический план

Базовая часть курса может быть освоена за один или два учебных года — 34 или 68 учебных недель плюс (при наличии желания и возможностей) — дополнительное время на выполнение творческих проектов. Часы для одногодичного варианта обучения проставлены перед скобками, для двухгодичного варианта — в скобках.

Количество занятий	Номер урока в учебнике	Название
4	1	Знакомство со Скретч
6	2	Повороты и направления
6	3	Процедуры, координаты, перо
6	4	Процедуры с параметрами
6	5	События в жизни исполнителей
6	6	Ветвления, клоны и переменные
6	7	Циклы ПОКА, условия, сенсоры, логические операторы
6	8	Графические редакторы Скретч
6	9	Работа со звуком
6	10	Повторение пройденного
10		Творческая работа Готовый проект
<b>Итого: 68</b>		

## Содержание

### Урок 1. Знакомство со Скретч

- Познакомить со средой программирования Скретч.
- Познакомить с элементами языка Скретч в процессе программирования несложного сценария для одного или двух объектов.
- Формировать умение решать и программировать несложные задачи и проекты.

## **Урок 2. Повороты и направления**

- Познакомить с понятиями, задающими положение спрайта (координаты) и его ориентацию (направление), а также с командами, меняющими координаты и направление.
- Формировать умение решать задачи в среде Скретч.
- Формировать навыки работы по техническому заданию (ТЗ).
- Формировать навыки составления ТЗ и алгоритма для реализации проекта в среде Скретч.

## **Урок 3. Процедуры, координаты, перо**

- Структурное программирование
- Объектное программирование
  - Инкапсуляция
  - Наследование. Объект и экземпляр объекта
  - Событийное управление
- Структурное и объектное программирование
- Скретч
- Процедурное мышление (рекомендация учителю)

## **Урок 4. Процедуры с параметрами**

- Процедурное программирование сверху вниз
  - Огородное пугало
    - Задача 1. Огородное пугало
- Универсальность кода
- Процедуры с параметрами
  - Нужен параметр
  - Как работает процедура с параметром
  - Как создать процедуру с параметром
  - Процедуры с несколькими параметрами
    - Задача 2. Прямоугольник
- Комментарии
- Стиль записи кода
- Запустить без обновления экрана
- Что мы узнали

## **Урок 5. События в жизни исполнителей**

- Параллельное и последовательное выполнение скриптов

- Создаём событие
- Прографируем историю
- Жизнь и удивительные приключения событий
  - Для кого сообщение?
  - Передать и ждать
- Смена фона создаёт событие
  - В лес за грибами
- События в жизни и программировании
- Справочник по командам группы «События»
- Что мы узнали

## Урок 6. Ветвления, клоны и переменные

- Ветвления
  - Арканоид
  - Шарик в свободном полёте
  - Ракетка
  - Отражаем шарик от ракетки
  - Падение шарика приводит к остановке игры
- Клоны
  - Как сделать 36 кирпичей и не пасть духом
  - Делаем кирпичи
- Переменные
  - Как определить конец игры
  - Как создать переменную
  - Работаем с переменной
  - Следим за значением переменной в программе
  - Переменные и клоны
- Улучшаем проект
  - Как отражать шарик по законам физики
  - Вывод формулы для отражения шарика
  - Улучшаем интерфейс игры
  - Убыстряем реакцию ракетки
  - Устраняем ляп
- Что мы узнали

## Урок 7. Циклы ПОКА, условия, сенсоры, логические операторы

- Циклы **ПОКА**

- Тележка с песком
- Слои
- Проблемы с шариком
- Цикл **ПОКА НЕ**
- Новые проблемы с шариком
- Цикл **ПОКА = ПОКА НЕ НЕ**
- Переменные локальные и глобальные
- Конструирование условий, датчики
  - Условия
    - Логические датчики
    - Операции сравнения
    - Логические операции
  - Измерительные датчики

### **Урок 8. Графические редакторы Скретч**

- Графика растровая и векторная
- Интерфейсы редакторов: что общего
  - Палитра
  - Цвет и толщина линии
  - Верхнее меню редактора
  - Сохранение в файл
- Растровый редактор
  - Кисть
  - Линия
  - Прямоугольник
  - Овал
  - Текст
  - Заливка
  - Ластик
  - Выбрать и вырезать
  - Волшебная палочка
  - Выбрать и копировать (штамп)
- Векторный редактор
  - Принципы векторного редактирования
  - Селектор
  - Изменение формы
  - Карандаш

- Линия
- Прямоугольник и Овал
- Текст
- Заливка
- Штмп
- Ближе, дальше
- Группировка

## Урок 9. Работа со звуком

- Подключаем звук
  - Выбрать в библиотеке
    - Содержимое библиотеки
  - Записать с микрофона
  - Загрузить из файла
  - Подключить к программе
- Редактируем звук
  - Запомнить фрагмент в буфере обмена
  - Удалить/Вырезать фрагмент
  - Вставить/Заменить фрагмент
  - Вставить/Заменить фрагмент записью с микрофона
  - Применить эффект
- Кодирование мелодий с нот
  - Введение в нотную грамоту
    - Октавы
    - Длительность нот
    - Тон, полутон, знаки альтерации
  - Кодирование нот
  - Команды, исполняющие ноты
  - От теории к практике
    - Задача 1. Гамма
    - Ключ
    - Размер и такт
    - Пауза
    - Длительность ноты (или паузы) в ударах по метроному
    - Кодировем гамму
      - Задача 2. Ёлочка
      - Задача 3. Ёлочка с барабанами



- Полезные ссылки

## Урок 10. Повторение пройденного

- Что же такое Скретч
  - Набор исполнителей
  - Язык программирования
  - Среда программирования
  - Среда разработки
- Сверху вниз
  - Задача 1. Улица
- Параллельное и последовательное
  - Задача 2. Комнаты
- Что дальше

## Конструктор занятий

Методические рекомендации пятой части Азбуки структурно отличаются от методических рекомендаций предыдущих частей с точки зрения планирования занятий. К каждому уроку предлагаются готовые ориентировочные планы (занятий может быть несколько), но педагог может реализовать собственный сценарий любого занятия, используя «конструктор занятий»

Что входит в состав конструктора? Это разнообразные файлы:

- уроки учебника, содержащие
  - вопросы для повторения
  - задачи
  - зачёт(ы)
  - проекты (условия, технические задания или части технических заданий)
- иллюстративные материалы:
  - картинки спрайтов и фонов
  - звуковые файлы
  - видео (выполненных задач и проектов)
  - слайды презентаций
- раздаточные материалы (файлы для распечаток):
  - задачи (условия)
  - домашние задания

Построению сценария занятия помогут:

- методические рекомендации к урокам
- программы решений проектов
- описания решений проектов
- решения зачётов
- электронный журнал.

Расположение файлов — элементов «конструктора» указывается в методических рекомендациях к каждому уроку.

### **Ориентировочные планы**

В методических рекомендациях даются ориентировочные планы занятий по каждому уроку. Они помогут составить планы, максимально соответствующие обучению конкретной (Вашей) группы обучающихся. Предлагаемые планы соответствуют «ожидаемому результату» (рекомендуемому минимуму). Для выполнения большего количества (или всех) проектов учебника потребуется больше времени, больше занятий.

Учебно-тематический план на один учебный год расположен на странице [abct5\Начало\Планы](#)

### **Техническое задание (ТЗ)**

Во втором уроке появится необычный элемент — Техническое задание (ТЗ) — подробное описание свойств создаваемого продукта. ТЗ тренировочных заданий может включать алгоритмы. Им необходимо строго следовать. Другие задания содержат лишь отдельные рекомендации по составлению ТЗ, а некоторые (как правило, под рубрикой «Сам себе режиссёр») предлагают обучающемуся самостоятельно построить ТЗ, ориентируясь на план, приведённый в учебнике на странице проектов урока 2.

Для составления ТЗ предлагается doc-шаблон.

### **Алгоритм**

Составлению программы (на языке, понятном исполнителю — интерпретатору Скретч) предшествует составление алгоритма (на естественном или графическом языке, понятном человеку).

Принципиальный момент Азбуки – формулировка алгоритма для каждой задачи/проекта. Сложность этого этапа состоит в том, что обучающиеся, познакомившись с командами языка программирования, часто с большим трудом формулируют алгоритм решения. Обычно они пытаются озвучить программу.

Чтобы сформулировать алгоритм, нужно тренироваться, нужно обсуждать варианты. Поэтому после первых ознакомительных занятий необходимо целенаправленно, в каждом проекте «гнуть» свою линию – сначала обсуждать алгоритм решения, потом собирать программу.

Алгоритм можно оформить по-разному:

- в виде комментария на программном поле Скретч

- с помощью doc-шаблона, который сопровождает doc-шаблон ТЗ
- в том виде, который удобен учителю.

### Оценивание

Результаты проверки выполненных работ педагог заносит в электронный журнал. Информацию по каждой работе удобно вносить в виде примечаний: педагог может быстро сообщить каждому обучающемуся замечания по любой работе, и обучающиеся могут самостоятельно ознакомиться в журнале со своими оценками и их обоснованием.

Сложность заданий оценивается в «куках». Максимальная сложность — 10 кук, минимальная — 1 кук.

Предлагается 4-х уровневый (0–3 балла) вариант оценки решений. Уровнями решения принимаются:

- Решение без замечаний
- Решение с мелкими погрешностями
- Решение с существенными погрешностями
- Отсутствие решения

Под решением (в общем случае) подразумевается совокупность:

- ТЗ
- алгоритм
- код

$\text{Общий балл} = (\text{ТЗ} + \text{Алгоритм} + \text{Код}) * \text{куки}$
--

В заданиях каждая из трёх указанных частей (ТЗ, алгоритм, код) могут быть сформулированы в разной степени. Оцениваться должна та часть работы, которая должна быть выполнена обучающимися.

Принимается следующая разбалловка:

*ТЗ*

- 3 — без замечаний
- 2 — мелкие погрешности
- 1 — существенные погрешности
- 0 — не составлено или дано в условии

*Алгоритм*

- 3 — без замечаний
- 2 — мелкие погрешности
- 1 — существенные погрешности
- 0 — не составлен или дан в условии

*Код*

- 3 — без замечаний
- 2 — мелкие погрешности
- 1 — существенные погрешности
- 0 — не составлен

Уточним, какие погрешности можно отнести к мелким, а какие к существенным.

### **Примеры погрешностей ТЗ**

*Мелкие погрешности:*

- Погрешности оформления
- Незначительное несоответствие поставленной задаче

*Серьёзные погрешности:*

- значительное несоответствие поставленной задаче

### **Примеры погрешностей алгоритма**

*Мелкие погрешности:*

- Погрешности оформления
- Незначительная недоработка условия задачи

*Серьёзные погрешности:*

- Значительное несоответствие поставленной задаче
- Построен не по принципу нисходящей разработки сверху вниз

### **Примеры погрешностей кода**

*Мелкие погрешности:*

- Код не выровнен в поле программы
- На поле программ присутствует «мусор»
- Код не читается по порядку (слева направо, сверху вниз)

*Серьёзные погрешности:*

- Неговорящие имена процедур
- Код построен не по принципу нисходящей разработки сверху вниз
- Отсутствуют ключевые комментарии или они не поясняют код
- Не соответствует алгоритму
- Есть тесты (наборы входных данных, начальных условий), которые не проходят

Система оценивания может варьироваться в зависимости от урока, для которого выполняются задания. Например, требование разработки алгоритма по принципу нисходящей разработки уместно применять только с 3-го урока.

Система оценивания поясняется обучающимся, чтобы они могли самостоятельно применять её для оценки своих работ и работ других обучающихся.

Систему оценивания педагог может задать самостоятельно. Главное, чтобы обучающиеся понимали её и учились самостоятельно оценивать выполненную работу. В таком случае потребуется редакция заготовки электронного журнала.

## **Особые замечания**

### **Тезаурус**

Это толковый словарик, описание важных терминов (которые встречаются в учебнике, но не только). Он появился именно в этой, пятой части Азбуки.

Вход в тезаурус на странице Начало. В тезаурус можно перейти со страниц учебника по гиперссылкам.

### **Подсказки приложения Скретч**

Система подсказок, входящих в приложение Скретч, в методических рекомендациях пока не использована. Познакомьтесь с этими примерами самостоятельно (Шаг за шагом, Как?, Блоки) и решите, будете ли Вы их использовать и как. Основная проблема — они не переведены на русский язык, и не все обучающиеся с лёгкостью смогут их прочитать.

### **Проблема чтения**

Хотелось бы отметить очень важную проблему, с которой на практике сталкиваются многие учителя. Зачастую школьники не любят читать тексты учебников. Можно ли изменить эту ситуацию? Давайте попробуем. Давайте внесём в занятия интригу.

В домашних заданиях методички будут встречаться задания: «Поиск сокровищ». Какие сокровища можно найти, изучая Скретч, где их искать?

Каждый урок учебника содержит множество различных материалов. Не обязательно разбирать на занятиях их все. Что-то можно опустить. Но обучающиеся не должны знать заранее, что пропущено. Они должны, работая дома с учебником, найти пропущенный фрагмент, разобраться с ним и рассказать на следующем занятии, что они узнали. А затем все вместе группой решить, «тянет» ли найденная информация на «сокровище»?

Например, при объяснении урока 2 педагог может опустить кусочек о цикле (Три важных замечания) и командах, предшествующих циклу (вход в цикл). Безусловно, такой фрагмент – сокровище!

Не может быть, чтобы не нашёлся хотя бы один обучающийся, который не загорелся поиском «сокровищ», но ведь за ним потянутся и другие! Возможно, мы сумеем заинтересовать детей таким чтением и последующим обсуждением важных для программирования вещей.

### **Планируемый результат:**

1. Закрепят комплекс базовых технологий, применяемых при создании проектов.
2. Разовьют эмоциональную сферу, образное мышление, внимание, фантазию, пространственное воображение и творческие способности.
3. Сформируют стремления к получению качественного законченного результата.

### **Литература для педагога:**

1. Программирование на Scratch. Часть 1. Д.В. Голиков и А.Д. Голиков
2. Программирование на Scratch. Часть 2. Д.В. Голиков и А.Д. Голиков
3. Методика обучения программированию на Scratch для учителей и родителей. Знакомство с интерфейсом.

### **Литература для учащихся, родителей:**

1. Программирование на Scratch. Часть 1. Д.В. Голиков и А.Д. Голиков
2. Программирование на Scratch. Часть 2. Д.В. Голиков и А.Д. Голиков
3. Методика обучения программированию на Scratch для учителей и родителей. Знакомство с интерфейсом.

### **Интернет – ресурсы:**

1. <https://scratch.mit.edu/>
2. <https://robotlandia.ru/abc5/>