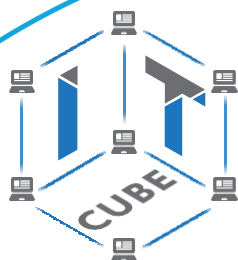




МИНИСТЕРСТВО  
ПРОСВЕЩЕНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ

ОБРАЗОВАНИЕ

НАЦИОНАЛЬНЫЕ  
ПРОЕКТЫ  
РОССИИ



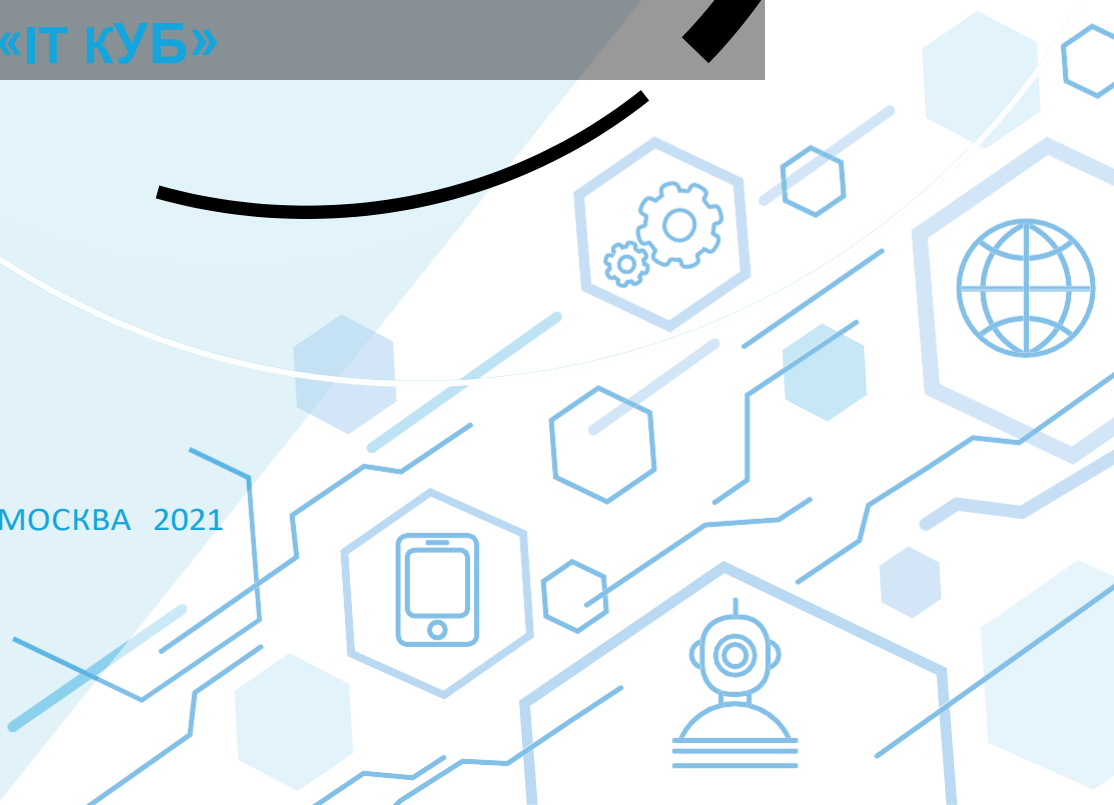
СЕТЬ ЦЕНТРОВ ЦИФРОВОГО  
ОБРАЗОВАНИЯ ДЕТЕЙ «ИТ-КУБ»

РЕАЛИЗАЦИЯ  
ДОПОЛНИТЕЛЬНОЙ  
ОБЩЕОБРАЗОВАТЕЛЬНОЙ  
ПРОГРАММЫ  
ПО ТЕМАТИЧЕСКОМУ  
НАПРАВЛЕНИЮ

## «ОСНОВЫ АЛГОРИТМИКИ И ЛОГИКИ

С ИСПОЛЬЗОВАНИЕМ  
ОБОРУДОВАНИЯ ЦЕНТРА  
ЦИФРОВОГО ОБРАЗОВАНИЯ  
ДЕТЕЙ «ИТ КУБ»

МОСКВА 2021



С. Г. Григорьев

М. А. Родионов

И. В. Акимова

**Реализация дополнительной общеобразовательной программы  
по тематическому направлению «Основы алгоритмики и логики»  
с использованием оборудования  
центра цифрового образования детей «IT-куб»**

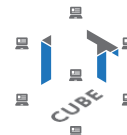
*Методическое пособие*

под ред. С. Г. Григорьева

Москва, 2021

## Содержание

|                                                                                                                                |            |
|--------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>Пояснительная записка .....</b>                                                                                             | <b>3</b>   |
| Нормативная база.....                                                                                                          | 4          |
| Основные понятия и термины .....                                                                                               | 5          |
| Структурирование материалов .....                                                                                              | 5          |
| <b>Описание материально-технической базы центра цифрового образования<br/>детей «ИТ-куб».....</b>                              | <b>7</b>   |
| <b>Примерная рабочая программа для организации работы по тематическому<br/>направлению «Основы алгоритмики и логики» .....</b> | <b>9</b>   |
| Планируемые результаты освоения программы обучающимися, с описанием<br>процедур итоговой и промежуточной аттестации.....       | 9          |
| Тематическое планирование .....                                                                                                | 10         |
| <b>Содержание и форма организации учебных занятий .....</b>                                                                    | <b>13</b>  |
| Планы учебных занятий .....                                                                                                    | 13         |
| Дидактические материалы .....                                                                                                  | 15         |
| Описание лабораторных работ .....                                                                                              | 38         |
| Примеры конспектов уроков.....                                                                                                 | 72         |
| Форма аттестации, примеры контрольно-оценочных материалов .....                                                                | 104        |
| <b>Материалы для организации и проведения учебно-исследовательской<br/>и проектной деятельности школьников.....</b>            | <b>107</b> |
| <b>Перечень доступных источников информации .....</b>                                                                          | <b>118</b> |



## Пояснительная записка

Основы алгоритмизации и программирования являются важной составляющей курса информатики средней школы. В Федеральном государственном образовательном стандарте основного общего образования (ФГОС ООО) указано, что одной из целей изучения курса информатики является развитие у учащихся основ алгоритмического мышления. Под способностью алгоритмически мыслить понимается умение решать задачи различного происхождения, требующие составления плана действий для достижения желаемого результата. Для того чтобы записать алгоритм решения задачи, необходим какой-то формальный язык, например блок-схемы. В примерной программе по информатике предполагается рассмотрение основных алгоритмических конструкций: ветвление, цикл, вспомогательный алгоритм. Также стоит отметить, что основы алгоритмизации в дальнейшем выступают базой для обучения программированию.

Целью программы «Основы алгоритмики и логики» является развитие алгоритмического мышления учащихся, творческих способностей, аналитических и логических компетенций, а также пропедевтика будущего изучения программирования на одном из современных языков.

Для достижения поставленной цели планируется достижение личностных, метапредметных и предметных результатов.

### **Личностные результаты:**

- формирование профессионального самоопределения;
- формирование уважительного отношения к интеллектуальному труду;
- формирование смыслообразования.

### **Познавательные УУД:**

- развитие алгоритмического и логического мышления;
- развитие умений постановки задачи, выделения основных объектов, математические модели задачи;
- развитие умения поиска необходимой учебной информации;
- формирование представления об этапах решения задачи;
- формирование алгоритмического подхода к решению задач;
- формирование ключевых компетенций проектной и исследовательской деятельности;
- формирование мотивации к изучению программирования.

### **Регулятивные УУД:**

- формирование умения целеполагания;
- формирование умения прогнозировать свои действия и действия других участников группы;
- формирование умения самоконтроля и самокоррекции.

### **Коммуникативные УУД:**

- формирование умения работать индивидуально и в группе для решения поставленной задачи;
- формирование трудолюбия, упорства, желания добиваться поставленной цели;
- формирование информационной культуры.

### **Предметные результаты:**

- формирование умения построения различных видов алгоритмов (линейных, разветвляющихся, циклических) для решения поставленных задач;
- формирование умения использовать инструменты среды Scratch для решения поставленных задач;
- формирование умения построения различных алгоритмов в среде Scratch для решения поставленных задач;
- формирование навыков работы со структурой алгоритма.

Программа рассчитана на учащихся в возрасте от 7 до 14 лет, не требует предварительных знаний и входного тестирования.

Занятия проводятся в группах до 12 человек, продолжительность занятия 45 минут, общая продолжительность программы 36 часов.

### Нормативная база

1. Конституция Российской Федерации (принята всенародным голосованием 12.12.1993 с изменениями, одобренными в ходе общероссийского голосования 01.07.2020). — URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_28399/](http://www.consultant.ru/document/cons_doc_LAW_28399/) (дата обращения: 10.03.2021).

2. Федеральный закон от 29.12.2012 № 273-ФЗ (ред. от 31.07.2020) «Об образовании в Российской Федерации» (с изм. и доп., вступ. в силу с 01.09.2020). — URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_140174](http://www.consultant.ru/document/cons_doc_LAW_140174) (дата обращения: 28.09.2020).

3. Паспорт национального проекта «Образование» (утверждён президиумом Совета при Президенте РФ по стратегическому развитию и национальным проектам, протокол от 24.12.2018 № 16). — URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_319308/](http://www.consultant.ru/document/cons_doc_LAW_319308/) (дата обращения: 10.03.2021).

4. Государственная программа Российской Федерации «Развитие образования» (утверждена постановлением Правительства РФ от 26.12.2017 № 1642 (ред. от 22.02.2021) «Об утверждении государственной программы Российской Федерации «Развитие образования»). — URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_286474](http://www.consultant.ru/document/cons_doc_LAW_286474) (дата обращения: 10.03.2021).

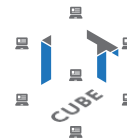
5. Стратегия развития воспитания в Российской Федерации на период до 2025 года (утверждена распоряжением Правительства РФ от 29.05.2015 №996-р «Об утверждении Стратегии развития воспитания в Российской Федерации на период до 2025 года»). — URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_180402/](http://www.consultant.ru/document/cons_doc_LAW_180402/) — (дата обращения: 10.03.2021).

6. Профессиональный стандарт «Педагог (педагогическая деятельность в дошкольном, начальном общем, основном общем, среднем общем образовании), (воспитатель, учитель)» (ред. от 16.06.2019) (приказ Министерства труда и социальной защиты РФ от 18 октября 2013 г. № 544н, с изменениями, внесёнными приказом Министерства труда и соцзащиты РФ от 25 декабря 2014 г. № 1115н и от 5 августа 2016 г. № 422н). — URL: <http://профстандартпедагога.рф> (дата обращения: 10.03.2021).

7. Профессиональный стандарт «Педагог дополнительного образования детей и взрослых» (приказ Министерства труда и социальной защиты РФ от 5 мая 2018 г. № 298н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых»). — URL: [https://profstandart.rosmintrud.ru/obshchiy-informatsionnyy-blok/natsionalnyy-reestr-professionalnykh-standartov/reestr-professionalnykh-standartov/index.php?ELEMENT\\_ID=48583](https://profstandart.rosmintrud.ru/obshchiy-informatsionnyy-blok/natsionalnyy-reestr-professionalnykh-standartov/reestr-professionalnykh-standartov/index.php?ELEMENT_ID=48583) (дата обращения: 10.03.2021).

8. Федеральный государственный образовательный стандарт основного общего образования (утверждён приказом Министерства образования и науки Российской Федерации от 17 декабря 2010 г. № 1897) (ред. 21.12.2020). — URL: <https://fgos.ru> (дата обращения: 10.03.2021).

9. Федеральный государственный образовательный стандарт среднего общего образования (утверждён приказом Министерства образования и науки Российской Федерации от 17 мая 2012 г. № 413) (ред.11.12.2020).— URL: <https://fgos.ru> (дата обращения: 10.03.2021)



10. Федеральный государственный образовательный стандарт начального общего образования (Утверждён приказом Минобрнауки России от 26.11.2010 № 1241, от 22.09.2011 № 2357) — URL: <https://fgos.ru> (дата обращения: 10.03.2021).

11. Методические рекомендации по созданию и функционированию детских технопарков «Кванториум» на базе общеобразовательных организаций (утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 г. № Р-4). — URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_374695/](http://www.consultant.ru/document/cons_doc_LAW_374695/) (дата обращения: 10.03.2021).

12. Методические рекомендации по созданию и функционированию центров цифрового образования «ИТ-Куб» (утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 г. № Р-5). — URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_374572/](http://www.consultant.ru/document/cons_doc_LAW_374572/) (дата обращения: 10.03.2021).

13. Методические рекомендации по созданию и функционированию в общеобразовательных организациях, расположенных в сельской местности и малых городах, центров образования естественно-научной и технологической направленностей («Точка роста»). — (Утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 г. № Р-6). — URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_374694/](http://www.consultant.ru/document/cons_doc_LAW_374694/) (дата обращения: 10.03.2021).

## Основные понятия и термины

**Алгоритм** — конечное точное предписание действий, которые необходимо выполнить для решения поставленной задачи.

**Исполнитель алгоритма** — это некоторый объект (техническое устройство, робот, автомат), способный выполнять определённый набор команд алгоритма.

**Среда Scratch** — визуальный язык программирования, позволяющий создавать интерактивные мультимедийные проекты.

**Линейный алгоритм** — это алгоритм, в котором команды последовательно выполняются однократно одна за другой.

**Условный алгоритм** — это алгоритм, порядок выполнения команд которого зависит от истинности или ложности некоторого условия.

**Циклический алгоритм** — это алгоритм, предусматривающий многократное повторение группы команд, называемых телом цикла.

**Переменная** — это область памяти компьютера, которая имеет название и хранит внутри себя какие-либо данные.

**Список** — в среде Scratch, это сложная переменная, предназначенная для хранения нескольких значений.

**Спрайт** — один из основных компонентов среды Scratch, для которого пишется программа.

**Скрипт** — программа в среде Scratch, которая состоит из блоков-операторов.

## Структурирование материалов

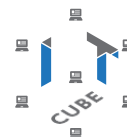
Содержание обучения может быть представлено следующими разделами:

- Знакомство со средой Scratch.
- Линейные алгоритмы.
- Работа с переменными.

- Условные алгоритмы.
- Циклические алгоритмы.
- Работа со списками.
- Создание подпрограмм.
- Индивидуальное задание.

Для каждого раздела подготовлены лабораторные работы с необходимым теоретическим материалом, заданиями и указанием к их выполнению. Также имеются дидактические материалы общей направленности, которые можно использовать при подготовке преподавателей и учащихся к занятиям, при выполнении лабораторных работ.

В конце пособия представлены примеры конспектов уроков на различные темы.



## Описание материально-технической базы центра цифрового образования детей «ИТ-Куб»

Для организации работы центра «ИТ-Куб» в распоряжении «Об утверждении методических рекомендаций по созданию и функционированию центров цифрового образования «ИТ-Куб» от 12.02.2021 рекомендуется следующее оборудование лаборатории:

- ноутбук — рабочее место преподавателя;
- рабочее место обучающегося;
- жёсткая, неотключаемая клавиатура: наличие;
- русская раскладка клавиатуры: наличие;
- диагональ экрана: не менее 15,6 дюймов;
- разрешение экрана: не менее 1920 × 1080 пикселей;
- количество ядер процессора: не менее 4;
- количество потоков: не менее 8;
- базовая тактовая частота процессора: не менее 1 ГГц;
- максимальная тактовая частота процессора: не менее 2,5 ГГц;
- кэш-память процессора: не менее 6 Мбайт;
- объём установленной оперативной памяти: не менее 8 Гбайт;
- объём поддерживаемой оперативной памяти (для возможности расширения): не менее 24 Гбайт;
- объём накопителя SSD: не менее 240 Гбайт;
- время автономной работы от батареи: не менее 6 часов;
- вес ноутбука с установленным аккумулятором: не более 1,8 кг;
- внешний интерфейс USB стандарта не ниже 3.0: не менее трёх свободных;
- внешний интерфейс LAN (использование переходников не предусмотрено): наличие;
- наличие модулей и интерфейсов (использование переходников не предусмотрено): VGA, HDMI;
- беспроводная связь Wi-Fi: наличие с поддержкой стандарта IEEE 802.11n или современнее;
- веб-камера: наличие;
- манипулятор «мышь»: наличие;
- предустановленная операционная система с графическим пользовательским интерфейсом, обеспечивающая работу распространённых образовательных и общесистемных приложений: наличие), МФУ, веб-камера, интерактивный моноблочный дисплей, диагональ экрана: не менее 65 дюймов, разрешение экрана: не менее 3840 × 2160 пикселей, оборудованный напольной стойкой.

В центре «ИТ-Куб» действует несколько лабораторий, в том числе лаборатория для осуществления направления «Основы алгоритмики и логики».



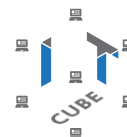
**Рис. 1.** Лаборатория ИТ-Куб

Лаборатории оборудованы ноутбуками Asus, процессор Intel (R) Core™ i5-8256UCPU, 1,60GHz, ОЗУ 600 Гб. Лаборатория оснащена также интерактивной доской, моноблочным интерактивным устройством, маркерной доской, МФУ.

На данном оборудовании могут выполняться лабораторные работы по курсу «Основы алгоритмики и логики», проводятся открытые занятия, защита проектов и т. д. С использованием презентационного оборудования преподаватели проводят объяснение нового материала, приводят примеры работы программ и т. д.



**Рис. 2.** Использование оборудования ИТ-куб при организации занятий по курсу «Основы алгоритмики и логики»



## Примерная рабочая программа для организации работы по тематическому направлению «Основы алгоритмики и логики»

### Планируемые результаты освоения программы обучающимися, с описанием процедур итоговой и промежуточной аттестации

Целью программы «Основы алгоритмики и логики» является развитие алгоритмического мышления учащихся творческих способностей, аналитических и логических компетенций.

Для достижения поставленной цели планируется достижение личностных, метапредметных и предметных результатов.

#### **Личностные результаты:**

- формирование профессионального самоопределения;
- формирование уважительного отношения к интеллектуальному труду;
- формирование смыслообразования.

#### **Познавательные УУД:**

- развитие алгоритмического и логического мышления;
- развитие умений постановки задачи, выделения основных объектов, математические модели задачи;
- развитие умения поиска необходимой учебной информации;
- формирование представления об этапах решения задачи;
- формирование алгоритмического подхода к решению задач;
- формирование ключевых компетенций проектной и исследовательской деятельности;
- формирование мотивации к изучению программирования.

#### **Регулятивные УУД:**

- формирование умения целеполагания;
- формирование умения прогнозировать свои действия и действия других участников группы;
- формирование умения самоконтроля и самокоррекции.

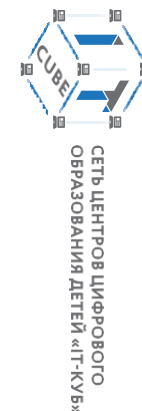
#### **Коммуникативные УУД:**

- формирование умения работать индивидуально и в группе для решения поставленной задачи;
- формирование трудолюбия, упорства, желания добиваться поставленной цели;
- формирование информационной культуры.

#### **Предметные результаты:**

- формирование умения построения различных видов алгоритмов (линейных, разветвляющихся, циклических) для решения поставленных задач;
- формирование умения использовать инструменты среды Scratch для решения поставленных задач;
- формирование умения построения различных алгоритмов в среде Scratch для решения поставленных задач;
- формирование навыков работы со структурой алгоритма.

## Тематическое планирование



| № п/п | Тема                                | Содержание                                                                                                                                                                                      | Целевая установка урока                                                                                  | Кол-во часов | Основные виды деятельности обучающихся на уроке/внеурочном занятии                                     | Использование оборудования               |
|-------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|--------------|--------------------------------------------------------------------------------------------------------|------------------------------------------|
| 1     | <i>Знакомство со средой Scratch</i> | Изучение основных элементов интерфейса среды Scratch, приёмы работы со спрайтами, приёмы работы с фоном, составление простых скриптов из различных блоков                                       | Ознакомление со средой Scratch, изучение основных инструментов среды                                     | 2            | Наблюдение за работой учителя, самостоятельная работа со средой Scratch, ответы на контрольные вопросы | Компьютер, проектор, интерактивная доска |
| 2     | <i>Линейные алгоритмы</i>           | Основные приёмы составления линейных алгоритмов в среде Scratch, решение задач на составление линейных алгоритмов                                                                               | Ознакомление с построением и выполнением линейных алгоритмов, работа с основными блоками в среде Scratch | 4            | Наблюдение за работой учителя, самостоятельная работа со средой Scratch, ответы на контрольные вопросы | Компьютер, проектор, интерактивная доска |
| 3     | <i>Работа с переменными</i>         | Основные приёмы добавления переменных в среде Scratch, использование основных блоков для работы с переменными, основные приёмы составления программ с использованием переменных в среде Scratch | Ознакомление с основами работы с переменными в среде Scratch                                             | 4            | Наблюдение за работой учителя, самостоятельная работа со средой Scratch, ответы на контрольные вопросы | Компьютер, проектор, интерактивная доска |
| 4     | <i>Условные алгоритмы</i>           | Ознакомление с понятием «условный алгоритм», основные приёмы составления условных алгоритмов в среде Scratch, использование основных блоков для составления условных алгоритмов в среде Scratch | Ознакомление с основами работы с условными алгоритмами в среде Scratch                                   | 6            | Наблюдение за работой учителя, самостоятельная работа со средой Scratch, ответы на контрольные вопросы | Компьютер, проектор, интерактивная доска |

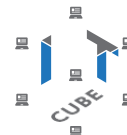


Продолжение

| №<br>п/п | Тема                  | Содержание                                                                                                                                                                                               | Целевая установка урока                                                            | Кол-во<br>часов | Основные виды деятельности обучающихся на уроке/внеурочном занятии                                     | Использование оборудования               |
|----------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-----------------|--------------------------------------------------------------------------------------------------------|------------------------------------------|
| 5        | Контрольная работа    | Решение задач                                                                                                                                                                                            | Проверка полученных навыков по темам «Линейные алгоритмы», «Условные алгоритмы»    | 2               | Самостоятельное выполнение контрольных заданий                                                         | Компьютер, проектор, интерактивная доска |
| 6        | Циклические алгоритмы | Ознакомление с понятием «циклический алгоритм», основные приёмы составления циклических алгоритмов в среде Scratch, использование основных блоков для составления циклических алгоритмов в среде Scratch | Ознакомление с основами работы с циклическими алгоритмами в среде Scratch          | 4               | Наблюдение за работой учителя, самостоятельная работа со средой Scratch, ответы на контрольные вопросы | Компьютер, проектор, интерактивная доска |
| 7        | Работа со списками    | Ознакомление с понятием «список» в среде Scratch, создание списка, работа с блоками по обработке списков, основные приёмы составления программ по работе со списками в среде Scratch                     | Ознакомление с основами работы со списками в среде Scratch                         | 4               | Наблюдение за работой учителя, самостоятельная работа со средой Scratch, ответы на контрольные вопросы | Компьютер, проектор, интерактивная доска |
| 8        | Создание подпрограмм  | Ознакомление с возможностью создания подпрограмм в среде Scratch. Раздел «Другие блоки», создание блока, параметры блока                                                                                 | Ознакомление с основами работы по созданию блоков-подпрограмм в среде Scratch      | 2               | Наблюдение за работой учителя, самостоятельная работа со средой Scratch, ответы на контрольные вопросы | Компьютер, проектор, интерактивная доска |
| 9        | Контрольная работа    | Решение задач                                                                                                                                                                                            | Проверка полученных навыков по темам «Циклические алгоритмы», «Работа со списками» | 2               | Самостоятельное выполнение контрольных заданий                                                         | Компьютер, проектор, интерактивная доска |

Продолжение

| №<br>п/п | Тема                   | Содержание                                                            | Целевая установка урока          | Кол-во<br>часов | Основные виды деятельности обучающихся на уроке/внеурочном занятии  | Использование оборудования               |
|----------|------------------------|-----------------------------------------------------------------------|----------------------------------|-----------------|---------------------------------------------------------------------|------------------------------------------|
| 10       | Индивидуальное задание | Разработка индивидуального или группового проекта в среде Scratch     | Создание проекта в среде Scratch | 4               | Самостоятельная индивидуальная или групповая проектная деятельность | Компьютер, проектор, интерактивная доска |
| 11       | Итоги                  | Защита индивидуальных или групповых проектов, подведение итогов курса | Защита проекта                   | 2               | Самостоятельная индивидуальная или групповая проектная деятельность | Компьютер, проектор, интерактивная доска |
|          | Итого                  |                                                                       |                                  | 36              |                                                                     |                                          |



## Содержание и форма организации учебных занятий

### Планы учебных занятий

#### 1. Знакомство со средой Scratch

Рекомендуемое количество часов на данную тему — 2 часа.

Планируемые результаты:

Предметные: получение навыков по работе в среде Scratch, освоение основных инструментов среды.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Оборудование и материалы:

Компьютер, презентационное оборудование.

Распределение лабораторных работ:

1-е занятие — выполнение лабораторной работы 1.1.

2-е занятие — выполнение лабораторной работы 1.2.

#### 2. Линейные алгоритмы

Рекомендуемое количество часов на данную тему — 4 часа.

Планируемые результаты: получение навыков по работе с линейными алгоритмами в среде Scratch, освоение основных инструментов среды.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Оборудование и материалы:

Компьютер, презентационное оборудование.

Распределение лабораторных работ:

1-е, 2-е занятие — выполнение лабораторной работы 2.1.

3-е, 4-е занятие — выполнение лабораторной работы 2.2.

#### 3. Работа с переменными

Рекомендуемое количество часов на данную тему — 4 часа.

Планируемые результаты: получение навыков по работе с переменными в среде Scratch, освоение основных инструментов среды.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицатель-

ные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

**Личностные:** готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

**Оборудование и материалы:**

Компьютер, презентационное оборудование.

**Распределение лабораторных работ:**

1-е, 2-е занятие — выполнение лабораторной работы 3.1.

3-е, 4-е занятие — выполнение лабораторной работы 3.2.

#### **4. Условные алгоритмы**

Рекомендуемое количество часов на данную тему — 6 часов.

**Планируемые результаты:** получение навыков по работе с условными алгоритмами в среде Scratch, освоение основных инструментов среды.

**Метапредметные:** способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

**Личностные:** готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

**Оборудование и материалы:**

Компьютер, презентационное оборудование.

**Распределение лабораторных работ:**

1-е, 2-е, 3-е занятие — выполнение лабораторной работы 4.1.

4-е, 5-е, 6-е занятие — выполнение лабораторной работы 4.2.

#### **5. Циклические алгоритмы**

Рекомендуемое количество часов на данную тему — 4 часа.

**Планируемые результаты:** получение навыков по работе с циклическими алгоритмами в среде Scratch, освоение основных инструментов среды.

**Метапредметные:** способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

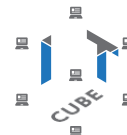
**Личностные:** готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

**Оборудование и материалы:**

Компьютер, презентационное оборудование.

**Распределение лабораторных работ:**

1-е занятие — выполнение лабораторной работы 5.1.



2-е, 3-е занятие — выполнение лабораторной работы 5.2.

4-е занятие — выполнение лабораторной работы 5.3.

#### **6. Работа со списками**

Рекомендуемое количество часов на данную тему — 4 часа.

Планируемые результаты: получение навыков по работе со списками в среде Scratch, освоение основных инструментов среды.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Оборудование и материалы:

Компьютер, презентационное оборудование.

Распределение лабораторных работ:

1-е, 2-е занятие — выполнение лабораторной работы 6.1.

3-е, 4-е занятие — выполнение лабораторной работы 6.2.

#### **7. Создание подпрограмм**

Рекомендуемое количество часов на данную тему — 2 часа.

Планируемые результаты: получение навыков по работе со списками в среде Scratch, освоение основных инструментов среды.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Оборудование и материалы:

Компьютер, презентационное оборудование.

Распределение лабораторных работ:

1-е, 2-е занятие — выполнение лабораторной работы 7.

## **Дидактические материалы**

### **1. Описание среды Scratch**

Scratch был создан на языке Squeak, который представляет собой одну из разновидностей Smalltalk. Главным идеологом Scratch является ученик Пейперта Мич Резник из MIT Media Lab (Массачусетский технологический институт), именно в нём в 1968 г. С. Пейперт разработал Logo. Текущей версией является Scratch 3.0, которая выпущена в январе 2019 г. В 2008 г. Scratch был портирован для микроконтроллерного модуля Arduino.

В настоящее время доступна онлайн-версия <https://scratch.mit.edu/>. Интерфейс онлайн-среды представлен ниже на рисунке 3.

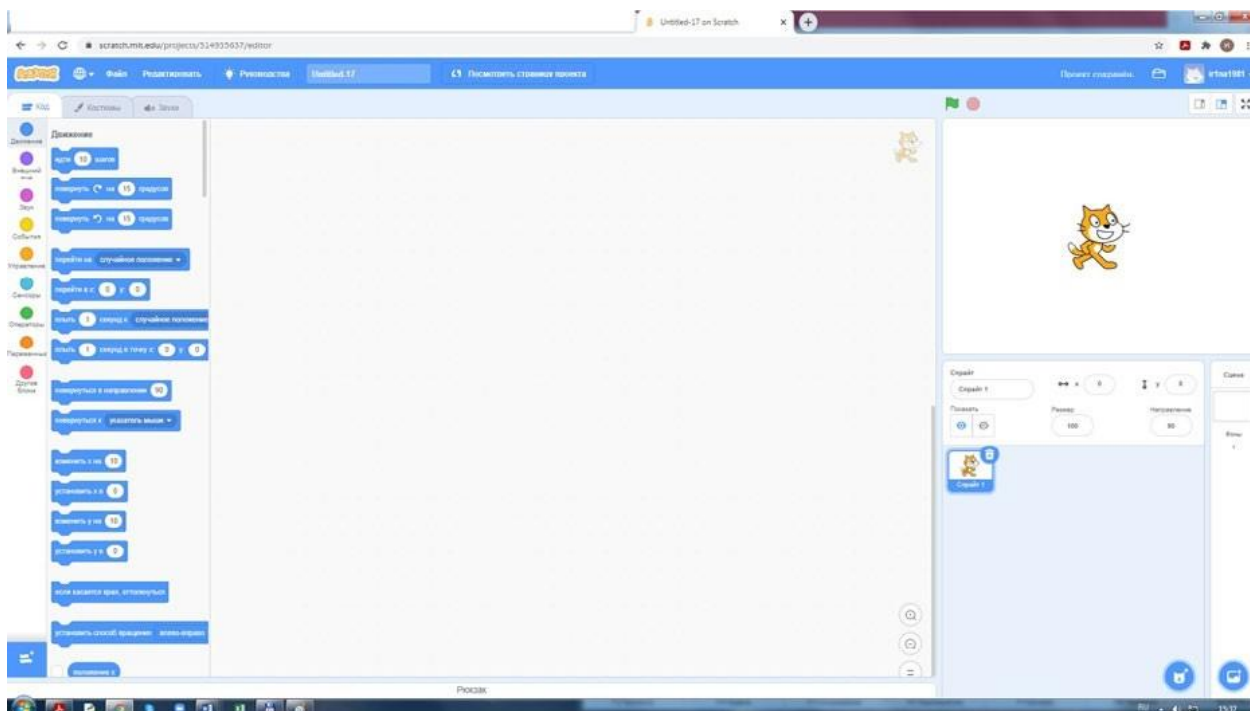


Рис. 3. Среда Scratch в режиме онлайн

Среда является полностью русифицированной. Всего в онлайн-версии доступно порядка 50 языков. Редактор Scratch состоит из нескольких крупных частей: сцена (обычно представлена справа сверху), список спрайтов (обычно справа снизу), палитра блоков и область скриптов (в левой части экрана).

## Справочник

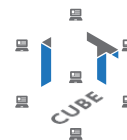
**Спрайты** — это объекты, которыми можно управлять с помощью блоков.

В их роли могут выступать игровые персонажи и герои мультфильмов. Также в левой части представлены закладки «Костюмы» и «Звуки».

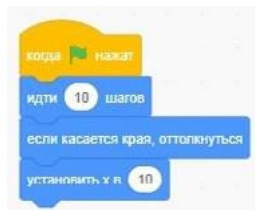
При запуске редактора в проекте по умолчанию всегда есть один персонаж — рыжий Кот, он же является символом Scratch (рис. 4).



Рис. 4. Вид спрайта рыжий Кот

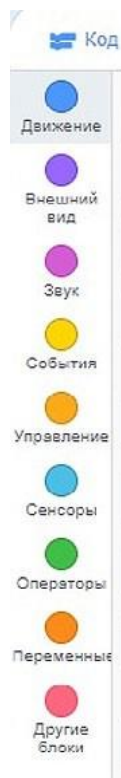


Scratch имеет собственный редактор текста программы, построенный на интересной идее конструкторов Lego: все операторы языка и другие его элементы представлены блоками, которые могут соединяться один с другим, образуя скрипт (фрагмент кода на рисунке 5).

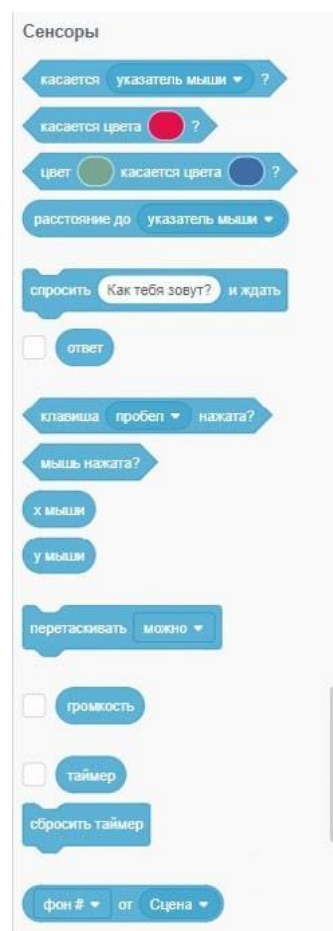


**Рис. 5.** Фрагмент программы в среде Scratch

В левой части среды в закладке Коды содержится девять кнопок, или блоков, которые имеют названия: движение, внешний вид, звук, события, управление, сенсоры, операторы, переменные, другие блоки (рис. 6). При включении одной кнопки все остальные выключаются. Включённая кнопка вся окрашивается в соответствующий ей цвет. При этом в нижней ячейке первого столбца появляются команды, связанные с включённой кнопкой. У всех кнопок разные привязанные к ним команды (рис. 6–7).



**Рис. 6.** Вид закладки Коды



**Рис. 7. Команды раздела Сенсоры**

Также имеется возможность сохранять свои программы на стационарный компьютер, а также загружать проекты со своего компьютера. Для этого необходимо обратиться к команде Файл ☉ Сохранить на свой компьютер, Файл ☉ Загрузить с компьютера (рис. 8)

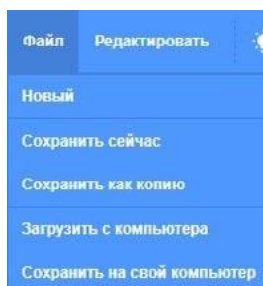


Рис. 8. Команды сохранения

Далее предлагается рассмотреть некоторые элементы интерфейса более подробно.

### Справочник

Одним из основных элементов интерфейса является **Сцена**, которая представляет собой то место, где спрайты выполняют движение, взаимодействуют друг с другом, рисуют и т. д.

Стандартный размер сцены составляет: ширина 480 шагов, а высота 360 шагов.

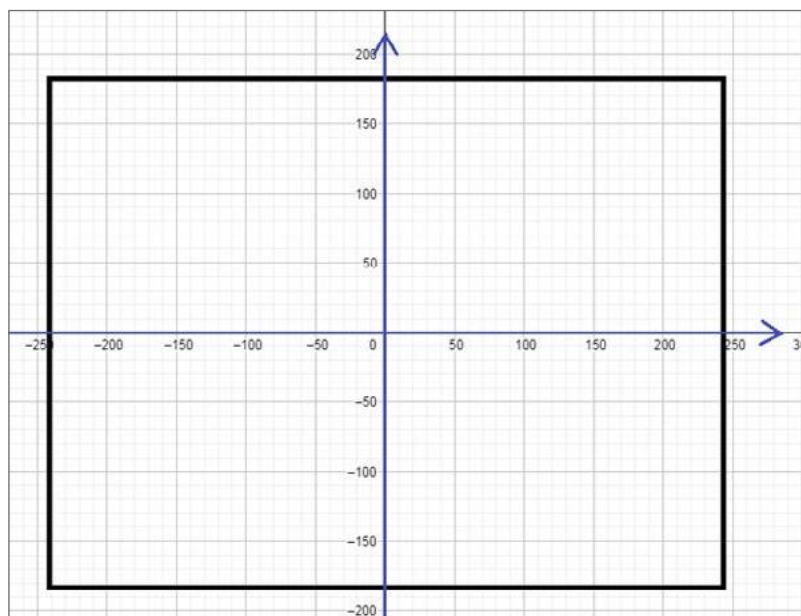
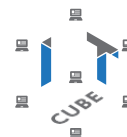


Рис. 9. Вид сцены



Центр сцены — точка  $(0,0)$  в системе координат. Текущее положение спрайта в координатах отображается в каждый момент времени в специальной панели под сценой (рис. 10).



Рис. 10. Координаты спрайта

Есть возможность развернуть сцену на весь экран, для этого используется режим презентации (рис. 11). Зелёный флажок и значок Стоп позволяют пользователю запустить или остановить программу (рис. 11).



Рис. 11. Команды управления над сценой

Список спрайтов также представляет собой важный элемент интерфейса. Он расположен внизу под сценой. Как было сказано ранее, основным спрайтом является рыжий Кот.

Может быть использовано достаточное множество персонажей, а также можно создать свой. Для выбора нового спрайта используются команды, представленные на рисунке 12.

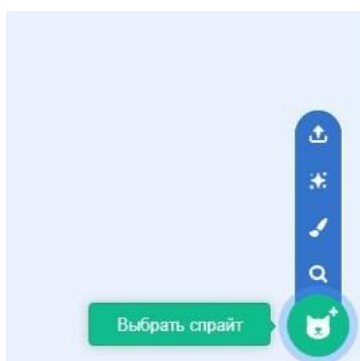
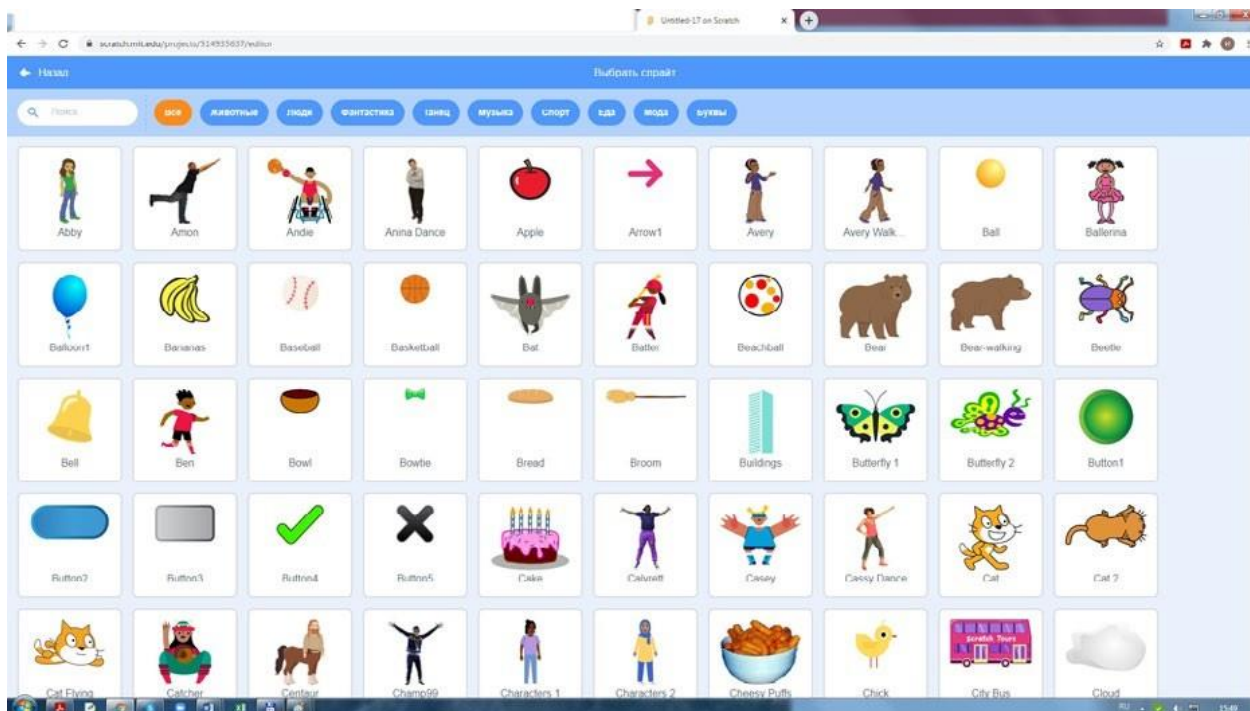


Рис. 12. Команды управления спрайтом

Новый спрайт может быть выбран несколькими способами:

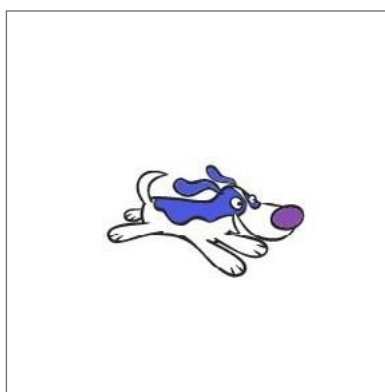
- выбрать из библиотеки спрайтов;
- создать в собственном редакторе;
- загрузить из файла;
- выбрать случайным образом.

Вид библиотеки спрайтов представлен на рисунке 13.

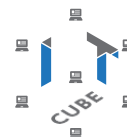


**Рис. 13.** Вид библиотеки спрайтов

Например, можем выбрать спрайт Собака вместо рыжего Кота (рис. 14).



**Рис. 14.** Вид спрайта Собака



У каждого спрайта в проекте свои скрипты, костюмы и звуки. Есть возможность выбрать любой спрайт, чтобы посмотреть на его атрибуты. Для этого необходимо либо кликнуть по его иконке в списке, либо дважды кликнуть по самому спрайту на Сцене. Если щёлкнуть по спрайту правой клавишей мыши, то можно вызвать команды дублирования, экспорта и удаления спрайта (рис. 15).

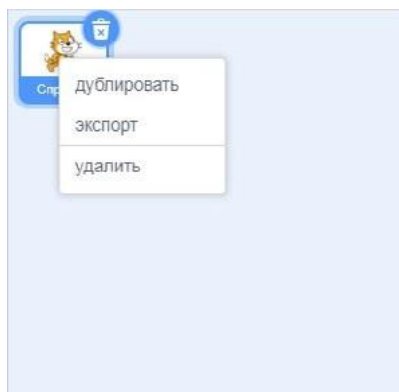


Рис. 15. Команды управления спрайтом

Также можно менять и редактировать фон на сцене, например из коллекции фонов (рис. 16—17). При создании нового проекта фон по умолчанию белый. Есть также возможность редактировать фон, создавать новый фон в редакторе. Для этого необходимо воспользоваться теми же командами, что и при редактировании спрайта (рис. 18).

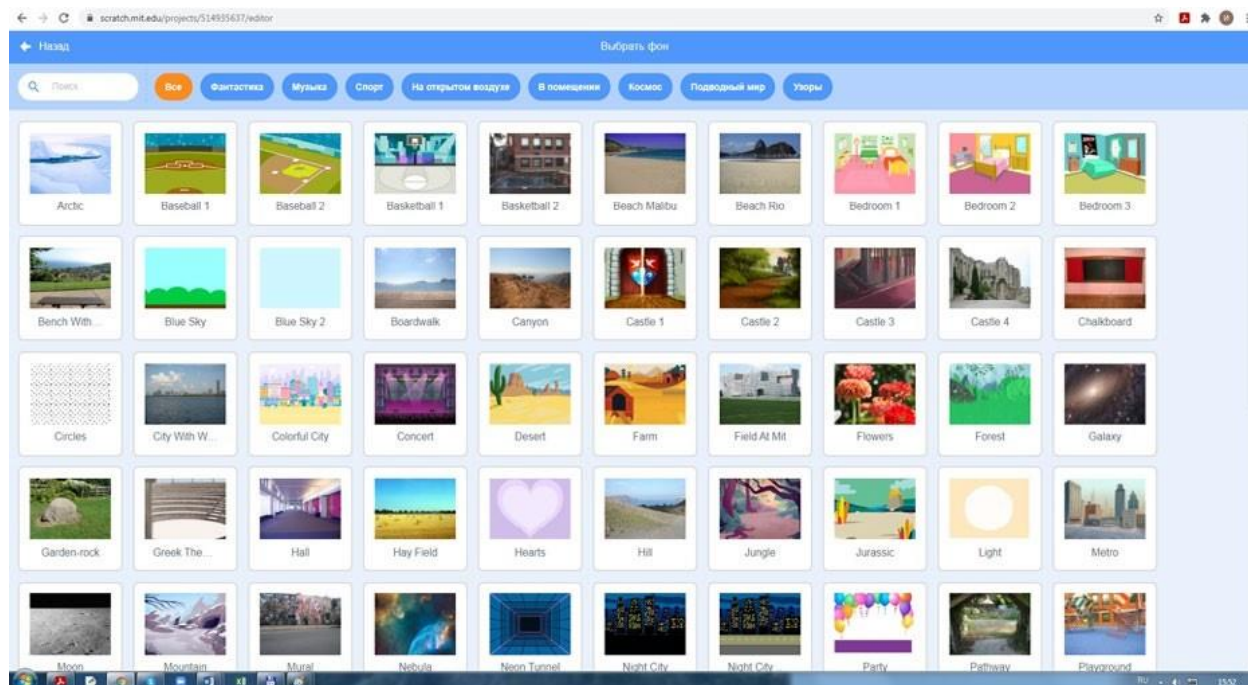


Рис. 16. Вид коллекции фонов



Рис. 17. Вид применённого фона



Рис. 18. Команды редактирования фона

## 2. Виды команд-блоков

Как было сказано ранее, программа на языке Scratch представляет собой набор блоков. Блоки имеют различное функциональное назначение. Ниже на рисунке представлены четыре типа блоков (рис. 19):



Рис. 19. Типология блоков

Опишем каждый вид блоков.

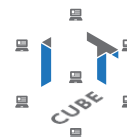
### Справочник

**Командные блоки** содержат обычно конкретную команду. У него имеются небольшие выступы, позволяющие соединять их между собой.

**Контрольные блоки** можно использовать для организации и проверки нескольких действий (например, циклов), которые могут быть представлены командными блоками. Такие блоки можно также объединять между собой, а также заключать другие блоки вовнутрь контрольных блоков.

**Блоки функции** не имеют выемок для объединения с другими блоками, они используются для вставки в другие блоки. По форме такого вида блоков можно определить тип данных, которые они содержат, например, блок с закруглёнными краями содержит числовую информацию.

**Блоки триггеры** имеют выемки только снизу, так как они размещаются поверх группы блоков. Обычно нацелены на ожидание какого-либо события, или действия



пользователя, например клика мышью. После этого события запускается группа нижерасположенных блоков.

### 3. Описание разделов с командами

Можно дать краткую характеристику каждому разделу. Надо отметить, что практически все команды среды Scratch имеют описательное название, поэтому предполагают интуитивную реализацию.

#### Справочник

Раздел **Движение** содержит группу команд, отвечающих за перемещение спрайта по сцене.

Например, команда «идти x шагов» предназначена для перемещения спрайта на x шагов. Среди самых популярных также можно отметить команды: «повернуть на ... градусов», «перейти в x:... y:...», «установить x в...», «установить y в ...» и т. д.

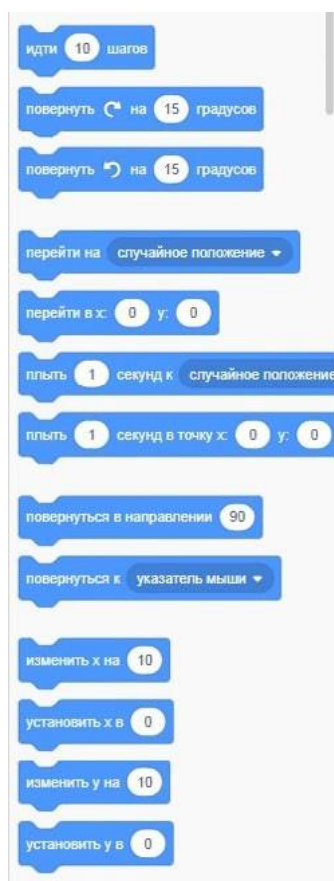


Рис. 20. Вид фрагмента раздела «Движение»

#### Справочник

Раздел **Внешний вид** предназначен для изменения внешнего вида спрайта.

Например, изменение костюма спрайта, смена фона, видимость и невидимость спрайта и т. д. Команды данного блока могут использоваться для вывода информации пользователю, например блок «сказать...».

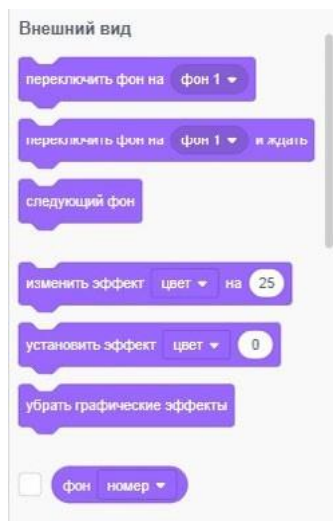


Рис. 2. Вид фрагмента раздела «Внешний вид»

### Справочник

Раздел **Звук** предназначен для проигрывания звуков. Содержит команды выбора инструментов, изменения громкости звуков, темпа и т. д.

Команды этого блока могут быть использованы для проигрывания музыкальных фрагментов.

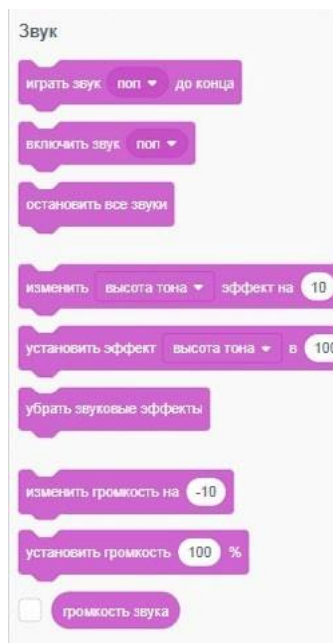


Рис. 22. Вид раздела «Звук»

### Справочник

Раздел **События** располагается в начале скрипта. Его блоки позволяют скрипту реагировать на различные события, такие как нажатие клавиш на клавиатуре, щелчок мышью и т. д. В этом блоке сосредоточены в основном блоки-триггеры.

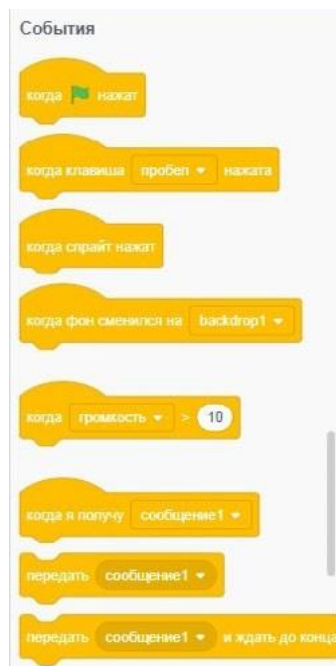
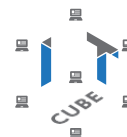


Рис. 23. Вид раздела «События»

### Справочник

Раздел **Управление** позволяет организовывать в программе такие конструкции, как ветвление, циклы различных видов. В данном разделе представлены как раз блоки управления.



Рис. 24. Вид фрагмента раздела «Управление»

## Справочник

Раздел **Переменные** отвечает за использование переменных в программе. Можно создавать, изменять значения как простых переменных, так и списков — фактически массивов.

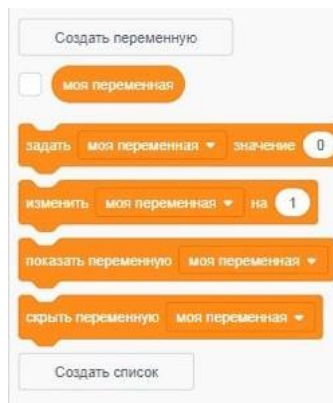


Рис. 25. Вид фрагмента раздела «Переменные»

## Справочник

Раздел **Сенсоры** содержит команды, позволяющие организовать ввод информации в компьютер.

Например, работа с таймером, ввод информации пользователем и т. д. Также в состав данного раздела входят блоки-функции, которые возвращают, например, координату x спрайта.

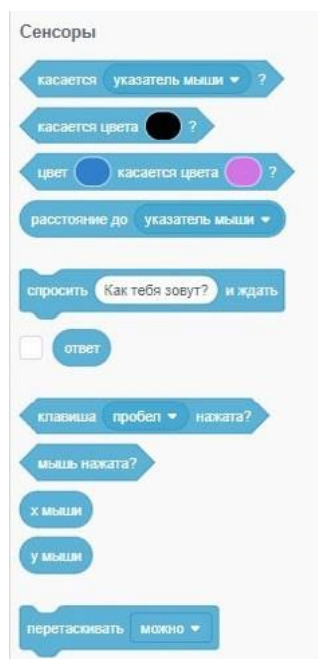
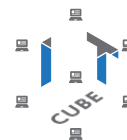


Рис. 26. Вид фрагмента раздела «Сенсоры»



## Справочник

Раздел **Операторы** содержит команды, которые используются совместно с сенсорами, блоками условий и с переменными. Самый нижний зелёный блок содержит в себе множество математических функций, таких как равно, больше, меньше, алгебраические функции и т. д. В составе данного раздела также множество блоков функций.

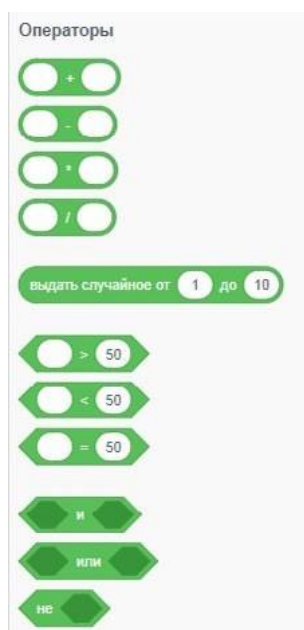


Рис. 27. Вид фрагмента раздела «Операторы»

Нижний блок **Другие блоки** позволяет реализовывать подпрограммы.

#### 4. Описание графического редактора

В состав среды Scratch входит графический редактор, который можно использовать для создания и редактирования персонажей, их костюмов, фонов. Данный графический редактор работает с векторной графикой, и многие его функции совпадают с традиционными функциями графических векторных редакторов.

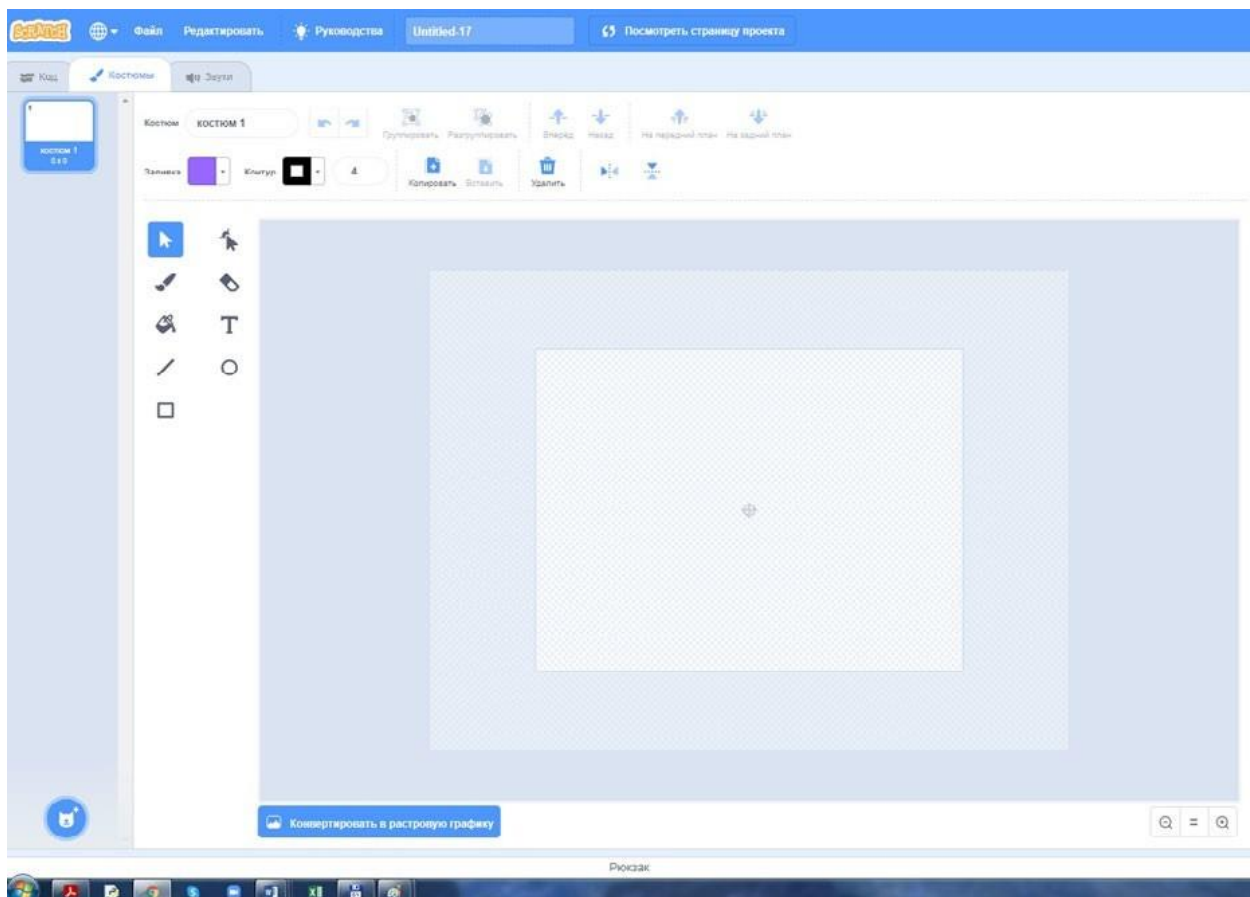


Рис. 28. Вид графического редактора



Рис. 29. Редактирование спрайта

Опишем основные инструменты. На рисунке 30 представлены основные инструменты рисования: кисть, заливка, линия, прямоугольник, изменение формы, ластик, текст, круг.

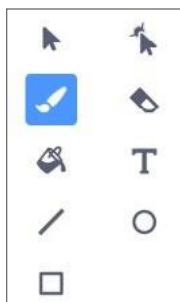
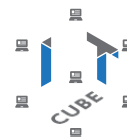


Рис. 30. Основные инструменты для рисования



Работа с данными инструментами также совпадает с работой с подобными инструментами в остальных графических векторных редакторах.

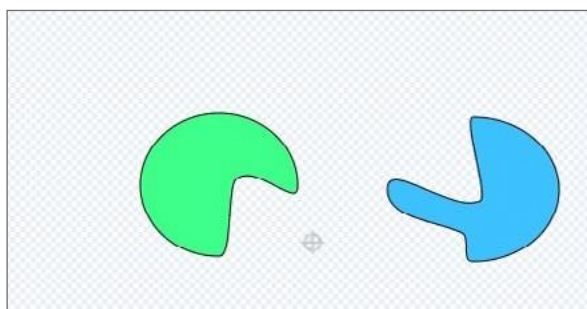


Рис. 31. Рисование овала и изменение его формы

Часто бывает полезно обозначать центр изображения, который представлен на экране.

Можно изменять заливку и контур фигуры с помощью соответствующих инструментов (рис. 32).

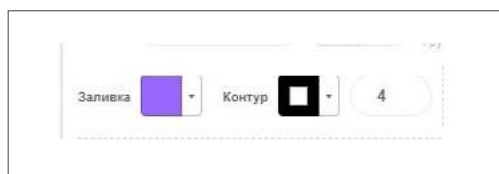


Рис. 32. Инструменты для заливки и контура

Также можно группировать и разгруппировывать фигуры, перемещать фигуры на передний план и т. д. с помощью соответствующих инструментов.

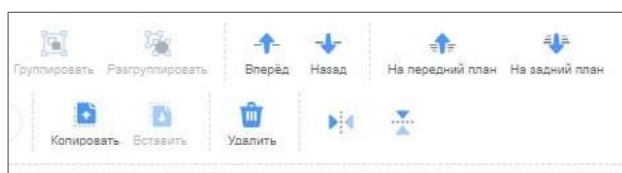


Рис. 33. Инструменты для группировки и перемещения элементов рисунка

Встроенный графический редактор может работать в растровом режиме. Переключение между режимами представлено кнопкой внизу экрана.

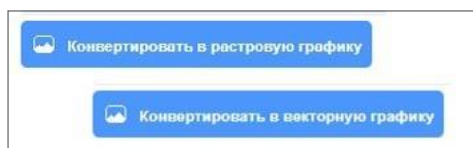


Рис. 34. Команды переключения между графическими режимами

Команды растрового режима практически аналогичны командам векторного режима (рис. 35).



Рис. 35. Вид команд растрового режима графического редактора

## 5. Работа с пером

В среде Scratch у каждого спрайта имеется возможность рисовать во время перемещения. Для этого используется так называемое перо, которое может быть поднято или опущено. Для того чтобы добавить работу с пером в среде Scratch, необходимо обратиться к её расширениям. Для этого необходимо выбрать команду, которая расположена внизу экрана (рис. 36).



Рис. 36. Команда вызова расширений среды Scratch

В разделе расширений можно найти несколько возможностей для среды Scratch, например: перевод текста, работа с роботами и т. д. (рис. 37).

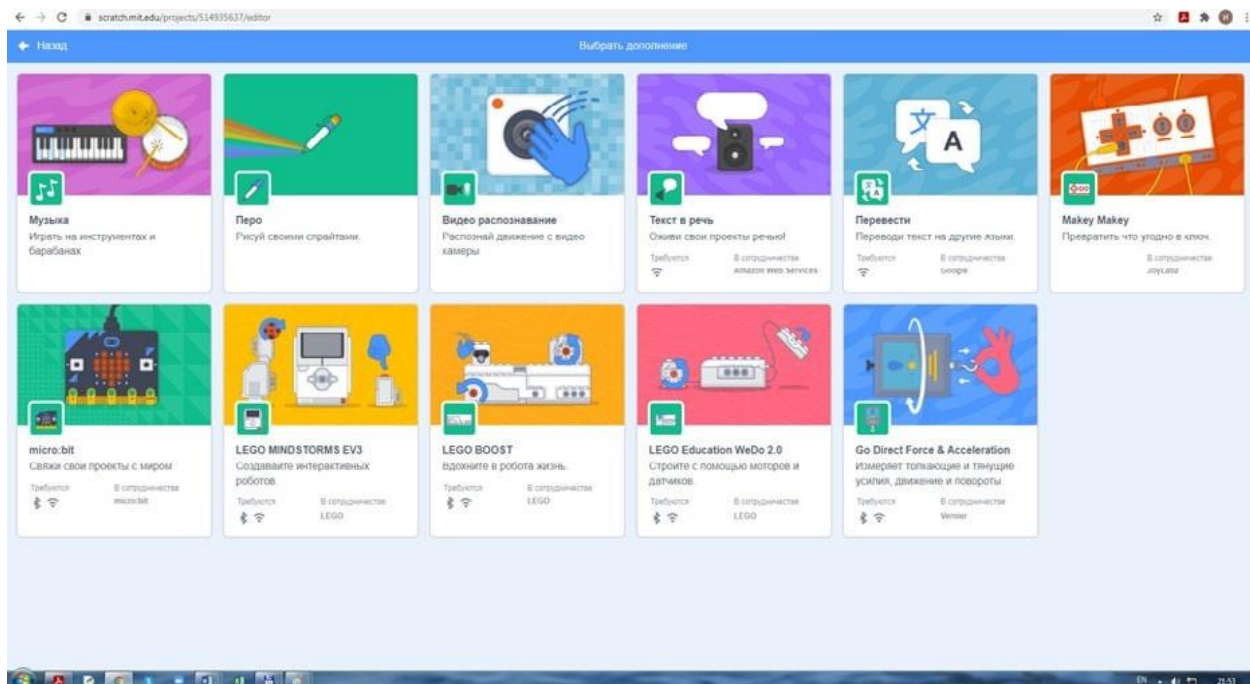
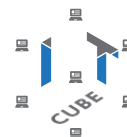


Рис. 37. Вид расширений среды Scratch

После выбора расширения «Перо» в разделе блоков появятся новые. Опишем более подробно команды данного раздела.



Блок «стереть всё» используется для очищения сцены (рис. 38). Стоит отметить, что изображения, выполненные пером спрайта, не являются частью фона.



Рис. 38. Вид блока для очищения сцены

Блок «Печать» используется для отпечатывания изображения спрайта на сцене. Такая копия спрайта называется штампом. Штамп не может перемещаться и не может иметь скриптов.



Рис. 39. Вид блока для создания штампа

Блок «опустить перо» позволяет рисовать при перемещении. После выполнения этой команды, перо будет оставлять след при движении спрайтов.



Рис. 40. Вид блока для рисования на сцене

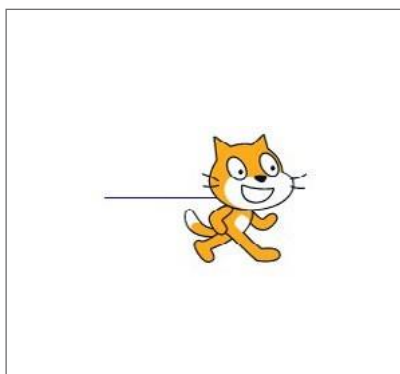


Рис. 41. Спрайт с оставленным изображением

Блок «поднять перо» поднимает вверх перо спрайта, после чего оно не будет рисовать при перемещении.



Рис. 42. Вид блока для поднятия пера

Можно также изменять размеры пера при рисовании. Для этого используются блоки «изменить размер пера» и «установить размер пера» (рис. 43–44).

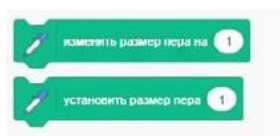


Рис. 43. Вид блоков для изменения размера пера

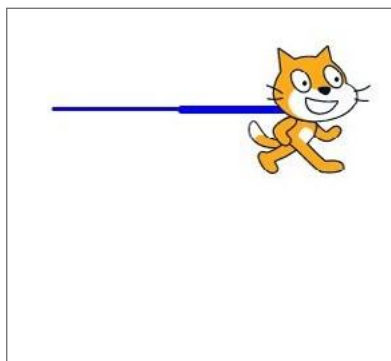


Рис. 44. Рисование с изменённым размером пера

Имеется возможность изменять цвет пера для рисования. Для этого используются блоки «установить для пера цвет...», «изменить цвет пера» и «установить цвет пера» (рис. 45). Кроме того, можно изменить конкретные атрибуты цвета пера, выбрав их из выпадающего списка.

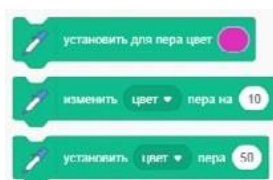


Рис. 45. Вид блоков для изменения цвета пера

## 6. Создание музыки

В расширениях среды Scratch также заложена возможность работы с музыкальными блоками. При выборе расширения «Музыка» в разделе блоков появляются соответствующие команды (рис. 46). В разделе собраны команды для проигрывания звуков музыкальных инструментов, например барабана.

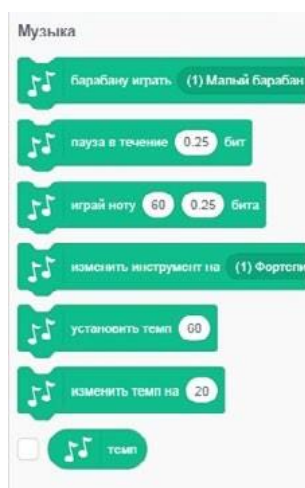
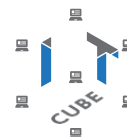


Рис. 46. Вид блоков раздела «Музыка»

Блок «Барабану играть...» предназначен для проигрывания выбранного звука определенное количество тактов. «Барабан» представляет собой перевод слова «drum», но в данном блоке имеются и другие музыкальные инструменты. В данном блоке содержится



несколько инструментов (рис. 47). Также для изменения музыкального инструмента используется блок «изменить инструмент на...» (рис. 48).

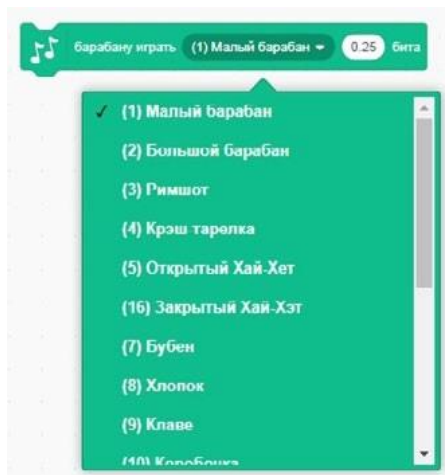


Рис. 47. Выбор инструмента в блоке «Барабану играть»

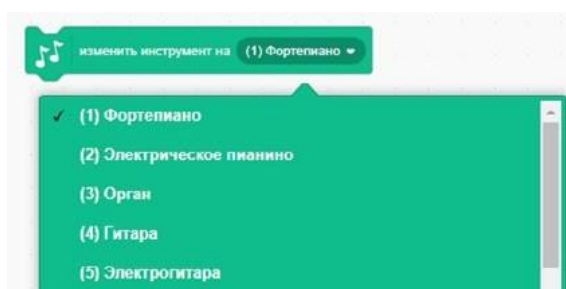


Рис. 48. Выбор инструмента в блоке «Изменить инструмент»

Для выбора длины такта или его изменения используются блоки «Изменить темп на...», «Установить темп...» (рис. 49). Темпом является скорость (тактов в минуту) проигрывания нот и инструмента.

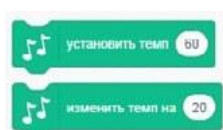


Рис. 49. Вид блоков для изменения такта

Блок «Играть ноту...» используется для проигрывания музыкальной ноты указанное число тактов. Пользователь может выбрать какую-то ноту играть в выпадающем окне (рис. 50).



Рис. 50. Вид блока для проигрывания ноты

Блок «Пауза в течение...» используется для организации паузы на указанное число тактов (рис. 51).



Рис. 51. Вид блока для организации паузы

Также в музыкальном разделе имеется переменная «Темп», которая содержит темп спрайта в тактах в минуту (рис. 52).



Рис. 52. Вид блока переменная «Темп»

## 7. Описание основных арифметических операторов и функций

В среде Scratch представлены основные арифметические операции: сложение (+), вычитание (−), умножение (\*) и деление (/). Данные команды представлены в разделе «Операторы», описанном ранее.



Рис. 53. Вид блоков операторов

Также представлен блок «mod», в котором находится остаток от деления, блок «округлить» выполняет округление числа, блок «модуль» для получения модуля числа (рис. 54).



Рис. 54. Вид блоков

Стоит отметить, что в данном блоке «модуль», есть возможность выбора другой математической функции. Так, в состав среды входят такие математические функции, как вычисление корня, тригонометрические функции, экспонента, вычисление натурального логарифма и т. д.

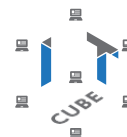


Рис. 55. Выбор в блоке математических функций

Также имеется возможность использования генератора случайных чисел, который заложен в блоке «выдать случайное от ... до ...». Данная команда возвращает случайное число в указанном диапазоне, например в диапазоне от 1 до 10.



Рис. 56. Блок генератора случайных чисел

Данные блоки обычно используются для вставки в другие блоки, которые позволяют вставку чисел.

Приведём пример скрипта, который выводит на экран случайное число от 1 до 100 (рис. 57).

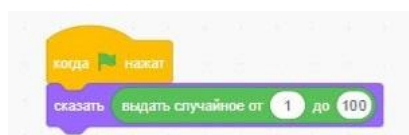


Рис. 57. Вид программы



Рис. 58. Результат работы скрипта

## 8. Работа со строками

Работа со строками сосредоточена в разделе «Операторы», и предоставляет возможность использования в программе нескольких основных действий со строками. Опишем эти действия более подробно.

Блок «Объединить (...) (...)» предоставляет возможность склеивать строки, т. е. выполнять их конкатенацию.

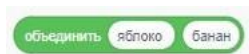


Рис. 59. Вид блока склейки строк

Блок «Буква ... в ...» используется для возвращения буквы из указанной позиции в строке.



Рис. 60. Вид блока поиска буквы

Блок «Длина ...» возвращает длину указанной строки.

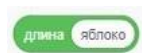


Рис. 61. Вид блока поиска длины

Блок «(...) содержит ...» возвращает истину (TRUE), если указанная строка содержит подстроку (второй параметр). Возвращает ложь (FALSE) в противном случае.



Рис. 62. Вид блока проверки вхождения

Приведём пример скрипта, обрабатывающего строковую информацию. Следующий скрипт проговаривает слово по буквам (рис. 63). Для работы блока необходимо создать переменную *s*, которая будет содержать номер текущей буквы в слове.



Рис. 63. Вид программы



Рис. 64. Результат работы программы

### 9. Работа со звуком

Ранее было сказано, что спрайты также могут произносить различные звуки, которые могут добавить определённый интерес в программу: например, когда персонаж рад или расстроен. Для работы со звуком используется специальная закладка — «Звуки», которая расположена в верхней части экрана. Для создания звуков имеется несколько возможностей (по аналогии работы со спрайтами).

У пользователя есть возможность загружать их из библиотеки звуков (рис. 65), записывать звуки с микрофона, загружать звуки с внешнего носителя. Scratch читает только файлы в форматах MP3 и WAV.

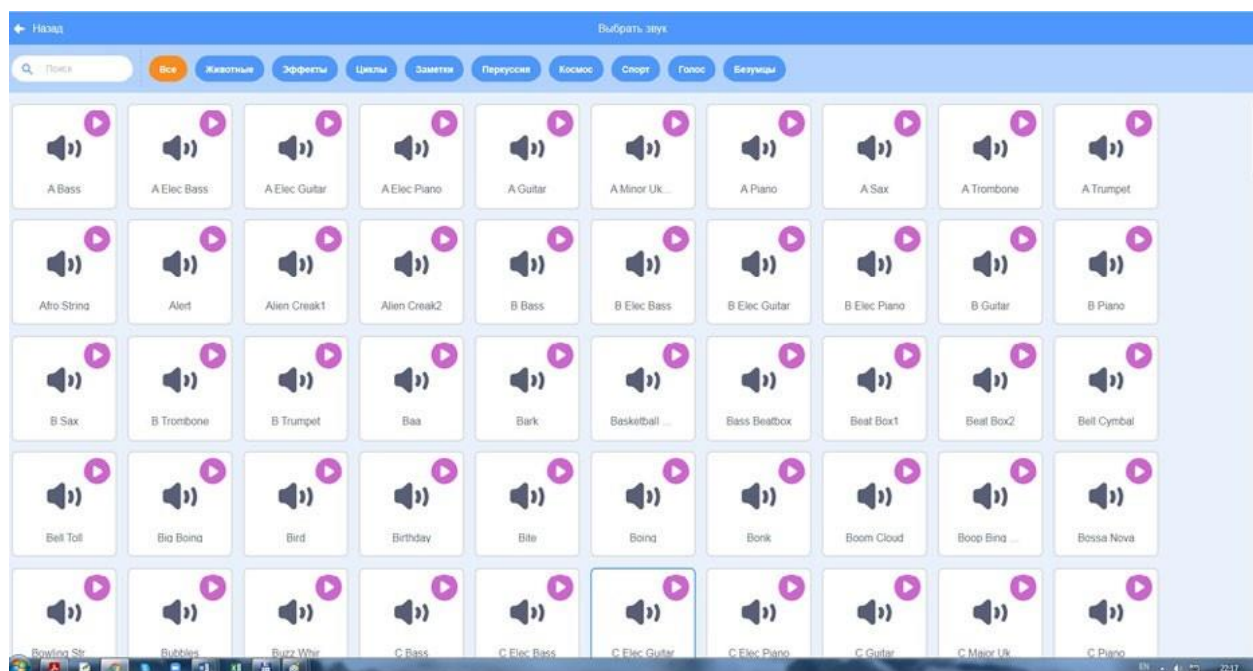
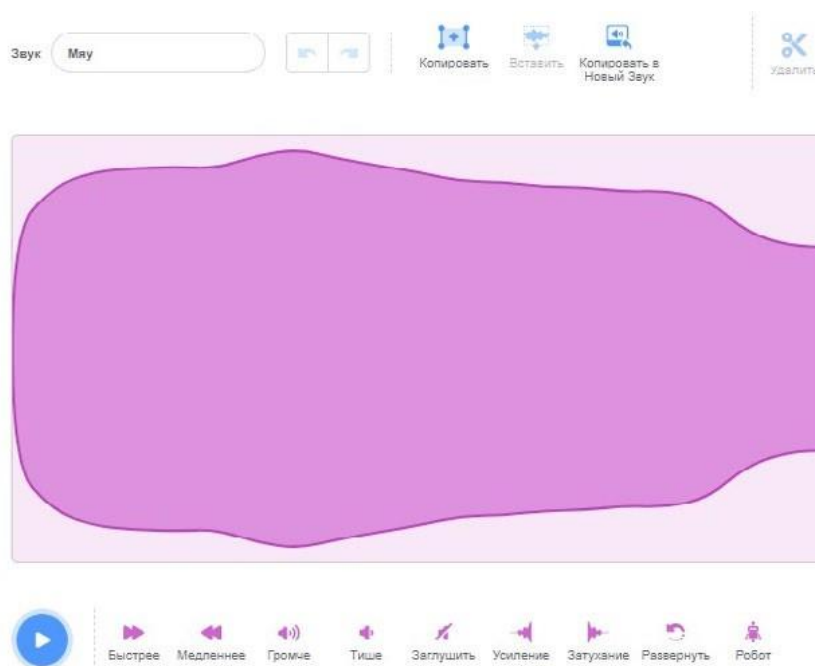


Рис. 65. Библиотека звуков

Также в среде имеются инструменты для редактирования звуковой информации (рис. 66).



**Рис. 66.** Редактирование звукового файла

## Описание лабораторных работ

### Лабораторная работа 1.1. «Знакомство со средой Scratch»

#### Теоретическая часть

Как известно, программы пишутся на языке программирования. Программа же представляет собой набор инструкций, которые компьютер должен выполнить. Scratch представляет собой визуально-ориентированный язык программирования для детей.

Scratch был создан на языке Squeak, который представляет собой одну из разновидностей Smalltalk. Главным идеологом Scratch является ученик Пейперта Мич Резник из MIT Media Lab (Массачусетский технологический институт), именно в нём в 1968 г. С. Пейперт разработал Logo.

В настоящее время доступна онлайн версия <https://scratch.mit.edu/>.

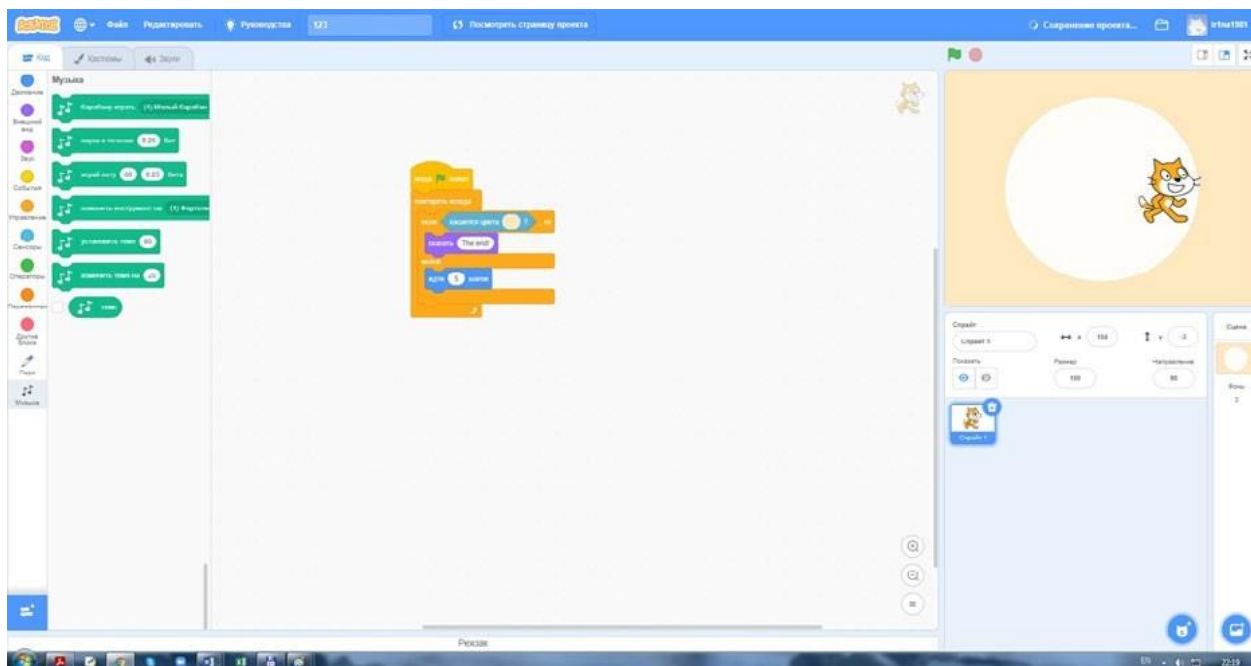
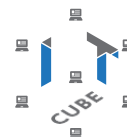


Рис. 67. Вид онлайн-среды Scratch

Scratch предназначен для визуального программирования. Программа на языке Scratch состоит из блоков, описывающих действия, которые выполняет спрайт. Роль спрайта по умолчанию выполняет рыжий Кот, но есть возможность смены спрайта.

Подробное описание среды представлено в дидактических материалах.

### *Практическая часть*

Цель работы: ознакомление со средой Scratch, изучение основных инструментов среды.

#### **Ход лабораторной работы**

1. Откройте среду Scratch.
2. Выберите спрайт для работы, например Цыплёнок (рис. 68).

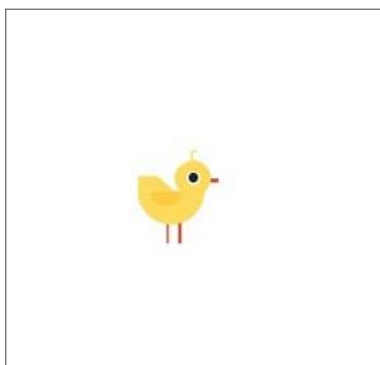


Рис. 68. Вид спрайта Цыплёнок

3. Выберите фон из коллекции фонов, например, Голубое небо (рис. 69).

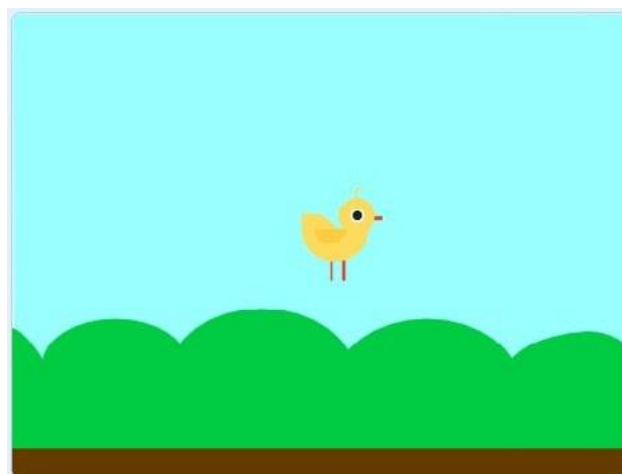


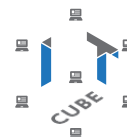
Рис. 69. Вид фона Голубое небо

4. Смените спрайт, удалите фон.
5. Выполните следующий скрипт.



Рис. 70. Вид скрипта

6. Объясните, что выполняет скрипт.



7. Измените спрайт в графическом редакторе Scratch.

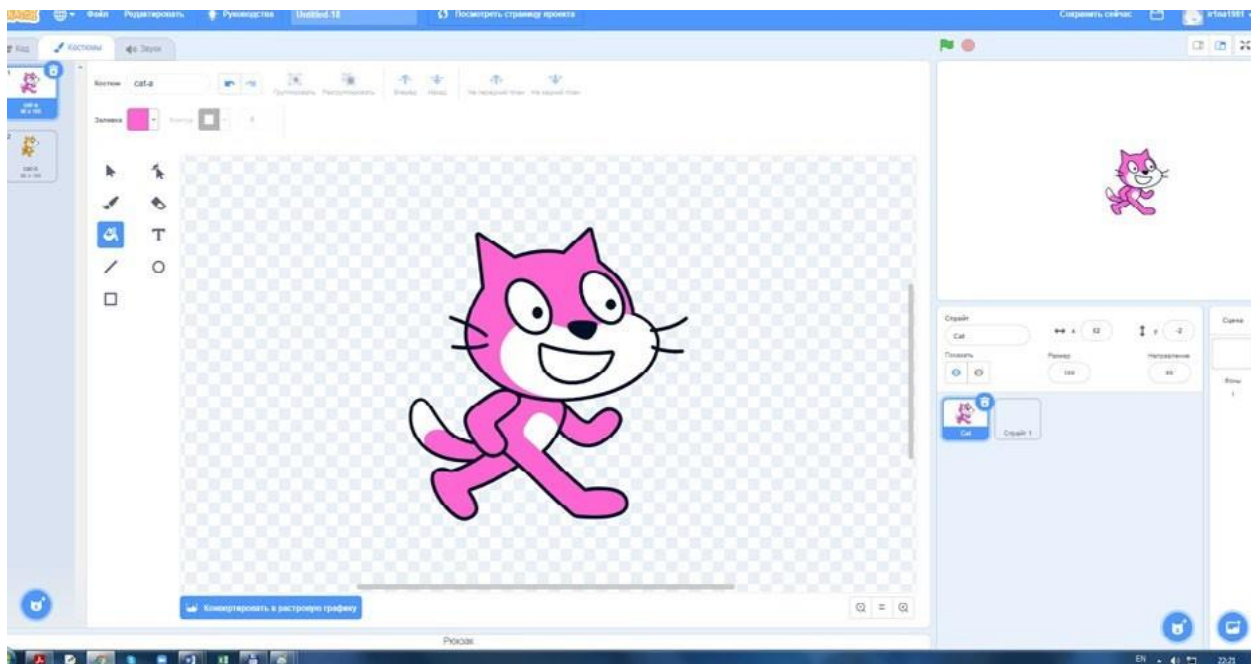


Рис. 71. Изменение спрайта в графическом редакторе

8. Выполните следующий скрипт (рис. 72).

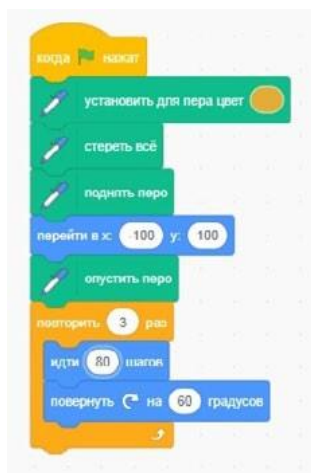


Рис. 72. Вид скрипта

9. Объясните, что выполняет скрипт.

10. Выполните следующий скрипт (рис. 73).

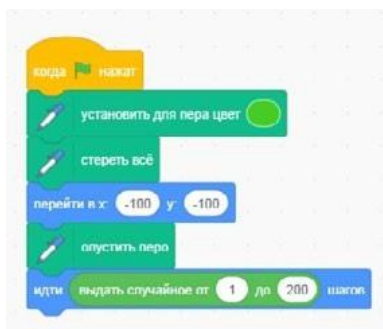


Рис. 73. Вид скрипта

11. Объясните, что выполняет скрипт.

Вывод: в процессе выполнения лабораторной работы вы получили представление о работе в среде Scratch.

Контрольные вопросы:

- 1) Как изменить вид спрайта в среде Scratch?
- 2) Можно ли редактировать фон в среде Scratch?
- 3) Какой вид графики можно создавать с помощью графического редактора в среде Scratch?

## Лабораторная работа 1.2. «Знакомство со средой Scratch»

### Теоретическая часть

В данной лабораторной работе используются дидактические материалы, описывающие работу в графическом редакторе среды Scratch.

### Практическая часть

Цель работы: ознакомление со средой Scratch, ознакомление с графическим редактором, изучение блоков по работе с костюмами спрайтов, изучение блоков по работе с фонами.

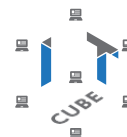
#### Ход лабораторной работы

Выполните в среде Scratch решение следующих задач.

1. Выберите из коллекции персонажей спрайт Шар (рис. 74).



Рис. 74. Вид спрайта Шар



2. Зайдите в закладку «Костюмы» и ознакомьтесь с костюмами данного спрайта (рис. 75).

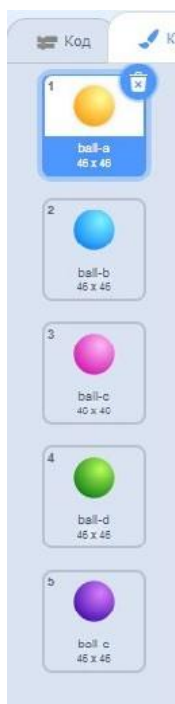


Рис. 75. Вид костюмов спрайта в графическом редакторе

3. Создайте следующий скрипт (рис. 76).

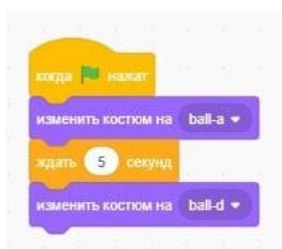


Рис. 76. Вид программы

4. Объясните, что выполняет данный скрипт.

5. Создайте следующие скрипты (рис. 77).

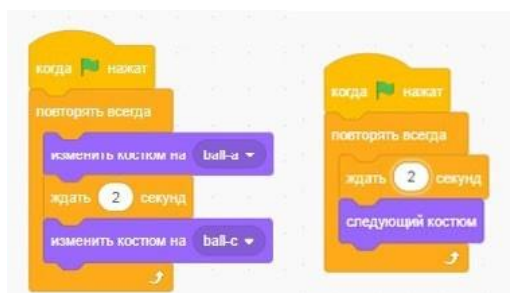


Рис. 77. Вид программы

6. Объясните, что выполняет данный скрипт.

7. Создайте свой собственный спрайт в графическом редакторе в виде облака (рис. 78).

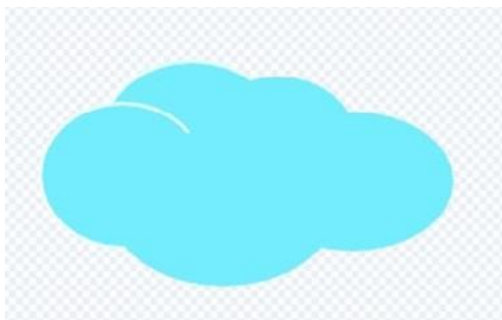


Рис. 78. Примерный вид спрайта

8. Создайте второй костюм для спрайта Облако, немного изменив его (рис. 79). Для создания второго костюма можно продублировать костюм с помощью специальной команды.



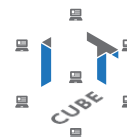
Рис. 79. Примерный вид спрайта

Примерный вид скрипта представлен ниже (рис. 80).



Рис. 80. Вид программы

9. Напишите скрипт, изменяющий костюмы спрайта. Объясните, какие блоки вы использовали в своей программе.



10. Создайте скрипт для своего спрайта, используя блоки «изменить размер на ...», «изменить эффект на ...». Примерный вид спрайта представлен ниже (рис. 81).

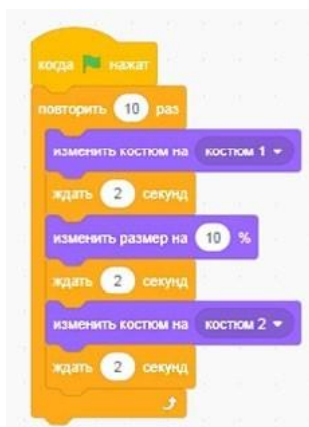


Рис. 81. Вид программы

11. Создайте новый проект, загрузите фоны из коллекции фонов (рис. 82).



Рис. 82. Вид фонов

12. Создайте следующий спрайт (рис. 83). Объясните, что выполняет данный скрипт.



Рис. 83. Вид программы

13. Измените фон в графическом редакторе.

14. Создайте скрипт с использованием блоков «Переключить фон на...».

15. Создайте скрипт, который создаёт изменяющуюся в размере и цвете надпись «Я изучаю Scratch».

Вывод: в ходе выполнения лабораторной работы вы получили представление о смене костюмов спрайтов, о действиях в графическом редакторе среды Scratch.

Контрольные вопросы:

1) Какие блоки вы использовали при выполнении лабораторной работы?

2) Какие инструменты вы использовали при создании изображений в графическом редакторе?

## Лабораторная работа 2.1. «Линейные алгоритмы»

### Теоретическая часть

Линейный алгоритм — это вид алгоритма, который образует команды, выполняемые однократно, одна за другой, в той последовательности, в которой они были изначально описаны. Линейная структура одна из самых простых. Примерная блок-схема линейного алгоритма представлена на рисунке 84.



Рис. 84. Блок-схема линейного алгоритма

Линейные алгоритмы реализуются в любом языке программирования, можно привести множество примеров линейных алгоритмов, например алгоритм вычисления суммы двух целых чисел.

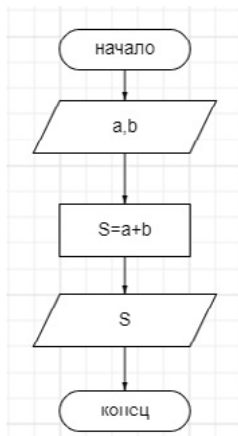


Рис. 85. Пример линейного алгоритма

### Практическая часть

Цель работы: ознакомление, построение и выполнение линейных алгоритмов, работа с основными блоками в среде Scratch.

#### Ход лабораторной работы

Выполните в среде Scratch решение следующих задач.

1. Составьте программу «Пройти 10 шагов».
2. Составьте программу «Пройти 100 шагов, повернуться по часовой стрелке на 60 градусов».
3. Выполните следующую программу. Представляет ли она линейный алгоритм?

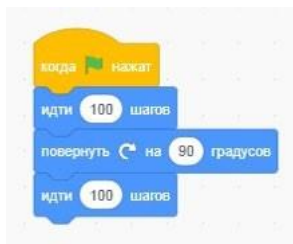
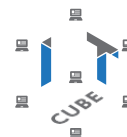


Рис. 86. Вид программы в среде Scratch

4. Составьте программу «Идти 200 шагов, повернуть на 90 градусов по часовой стрелке, пройти ещё 100 шагов».
5. Составьте программу «Повернуть против часовой стрелки на 45 градусов, пройти 100 шагов, повернуть на 90 градусов по часовой стрелке, пройти 100 шагов».
6. Составьте программу «Установить красный цвет, идти 50 шагов, установить жёлтый цвет, идти 10 шагов».

Вывод: в ходе выполнения лабораторной работы вы получили представление о создании линейных алгоритмов в среде Scratch.

Контрольные вопросы:

- 1) Какой алгоритм называется линейным?
- 2) Какие можно привести примеры линейных алгоритмов?
- 3) Какие блоки вы использовали в лабораторной работе при создании линейных программ?

## Лабораторная работа 2.2. «Линейные алгоритмы»

### Теоретическая часть

Используется теоретический материал из лабораторной работы 2.1.

### Практическая часть

Цель работы: ознакомление, построение и выполнение линейных алгоритмов, работа с основными блоками в среде Scratch.

#### Ход лабораторной работы

1. Рассмотрите работу блоков «Плыть ... секунд в точку», «Перейти в случайное положение», «Перейти в x: ... y: ...».
2. Создайте следующий скрипт (рис. 87). Объясните, что выполняет данный скрипт.

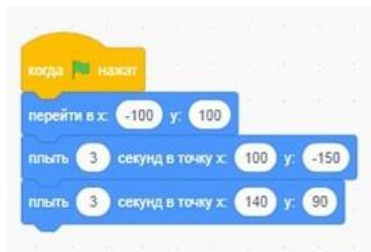
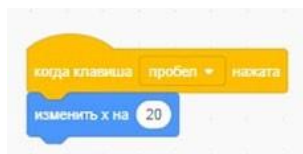


Рис. 87. Вид программы

3. Создайте следующий скрипт (рис. 88). Объясните, что выполняет данный скрипт.



**Рис. 88.** Вид программы

Выполните в среде Scratch решение следующих задач.

4. С помощью клавиши «стрелка вниз» (несколько нажатий) переместите спрайт в нижнюю часть сцены.

5. Измените фон сцены на изображение школы (рис. 89). С помощью клавиши «стрелка вверх» переместите спрайт к входу в школу.



**Рис. 89.** Вид сцены

Вывод: в процессе выполнения лабораторной работы вы получили представление о создании линейных алгоритмов в среде Scratch.

Контрольные вопросы:

- 1) Какой алгоритм называется линейным?
- 2) Какие инструменты среды Scratch вы использовали при создании скриптов?
- 3) Для чего используется блок «Перейти на случайное положение»?

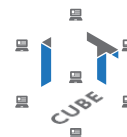
### Лабораторная работа 3.1. «Работа с переменными»

#### Теоретическая часть

Говоря о работе с различными алгоритмами, конечно же, нельзя пропустить работу с числами. Выполняя несложные линейные программы, которые обрабатывают числовые данные, например введенные пользователем, можно создать хороший задел на будущее, подготавливаясь к составлению сложных алгоритмов как в среде Scratch, так и к написанию непосредственно программ в среде программирования.

Переменная представляет собой область памяти компьютера, которая имеет название и хранит внутри себя какие-либо данные.

Переменная в среде Scratch бывает простая (хранит одно значение), также может представлять собой список для хранения больше одного значения.



Для создания таких программ пользователю потребуется работа с переменными. Среда Scratch предоставляет нам возможность вводить в проект как простые переменные, так и массивы. Для этого необходимо обратиться к разделу «Переменные». Для создания переменной необходимо выбрать команду «Создать переменную» (рис. 90).

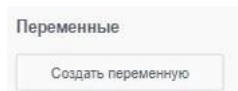


Рис. 90. Команда «Создание переменной»

В появившемся диалоговом окне указываем имя переменной.

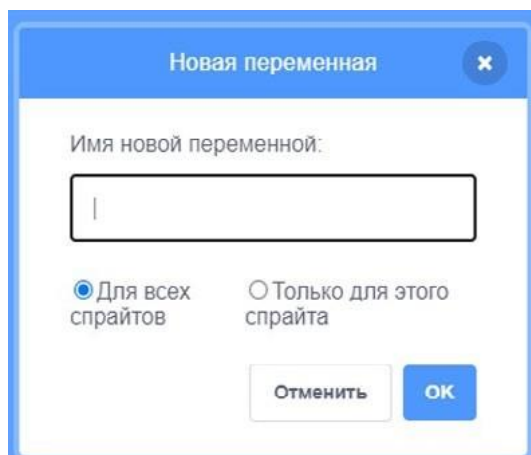


Рис. 91. Диалоговое окно для создания переменной

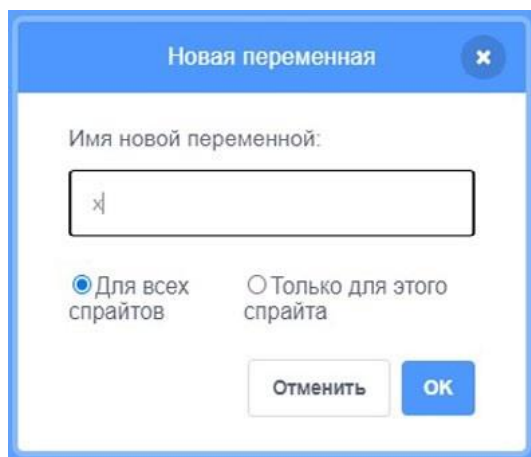


Рис. 92. Диалоговое окно для создания переменной

После задания имени переменной она появляется в проекте (рис. 93).

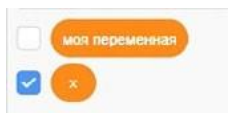


Рис. 93. Вид созданной переменной в разделе «Переменные»

Затем, после создания переменной, можно использовать различные команды для работы с ней, например задавать значение переменной, показывать значение переменной на сцене и т. д. Класс задач, решаемых с использованием переменных, очень широк.

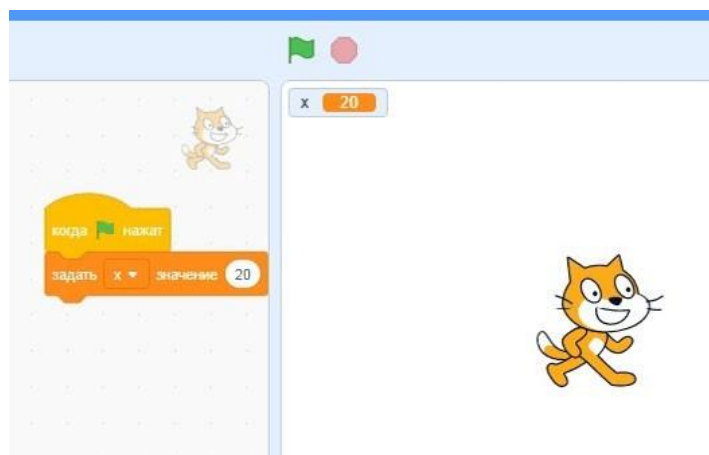


Рис. 94. Пример задания значения переменной

Опишем работу с основными блоками из раздела «Переменные».

Блок «Показать переменную ...» используется для того, чтобы показывать значение переменной на сцене. Блок «Скрыть переменную...» используется, чтобы перестать показывать значение переменной на сцене (рис. 95). По умолчанию переменная и её значение отображаются на сцене, но с помощью данных блоков можно скрывать или удалять переменную со сцены.

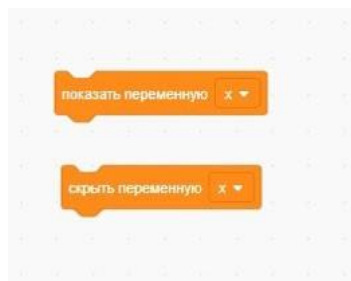


Рис. 95. Вид блоков по работе с переменными

Также можно изменять вид изображения переменной на сцене (рис. 96).

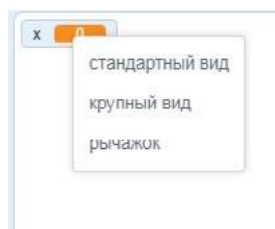
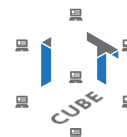


Рис. 96. Вид команд для изменения вида переменных



По умолчанию переменная показывается в левом верхнем углу сцены, но можно перемещать её в любое место с помощью мыши.

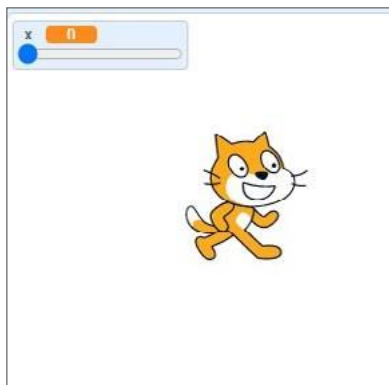


Рис. 97. Изменённый вид переменной на сцене

### Практическая часть

Цель работы: ознакомление с основами работы с переменными в среде Scratch.

#### Ход лабораторной работы

1. Создайте в среде Scratch следующие переменные:  $x$ ,  $z$ ,  $sum$ . Какие инструменты среды вы использовали для этого?
2. Выполните следующую программу (рис. 98).



Рис. 98. Вид программы

3. Выполните следующую программу (рис. 99).



Рис. 99. Вид программы

4. Выполните в среде Scratch решение следующей задачи: «Ввести значение  $x$ . Прибавить к  $x$  число 3».

5. Выполните в среде Scratch решение следующей задачи: «Для введенных с клавиатуры чисел  $x$  и  $y$  вычислить значение выражения  $7x - y$ ».

6. Выполните в среде Scratch решение следующей задачи: «Написать алгоритм вычисления площади прямоугольника».

Вывод: в процессе выполнения лабораторной работы вы получили представление о работе с переменными в среде Scratch.

Контрольные вопросы:

- 1) Что называется переменной?
- 2) Какие виды переменных можно использовать в среде Scratch?
- 3) С помощью каких блоков можно задать значение переменной в среде Scratch?

### Лабораторная работа 3.2. «Работа с переменными»

#### Теоретическая часть

Используется теоретический материал из лабораторной работы 3.1.

#### Практическая часть

Цель работы: ознакомление с основами работы с переменными в среде Scratch.

#### Ход лабораторной работы

Выполните в среде Scratch решение следующих задач.

1. Для введенных с клавиатуры чисел  $x$  и  $y$  вычислите значение выражения  $x^3 - 5y$ .
2. Для введенных с клавиатуры чисел  $a$ ,  $b$ ,  $c$  вычислите значение выражения  $\sqrt{2a} + \frac{b}{c} - 1$ .
3. Для введенных с клавиатуры чисел  $x$  и  $y$  вычислите значение выражения  $x^6 + 3y$ .
4. Для введенных с клавиатуры чисел  $x$  и  $y$  вычислите значение выражения  $4y^2 + 2xy + x^2$ .
5. Для введенных с клавиатуры чисел  $x$  и  $y$  вычислите значение выражения  $6x + \frac{1}{3}y + 13,76$ .

Вывод: в ходе выполнения лабораторной работы вы получили представление о работе с переменными в среде Scratch.

Контрольные вопросы:

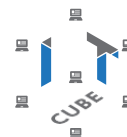
- 1) Как создать переменную в среде Scratch?
- 2) Какие основные блоки вы использовали для решения задач в лабораторной работе?
- 3) Какие виды алгоритмов вы создавали в лабораторной работе?

### Лабораторная работа 4.1. «Условные алгоритмы»

#### Теоретическая часть

Условный алгоритм — это алгоритм, порядок выполнения действий в котором зависит от выполнения (или не выполнения) какого-либо условия.

Условный, или, как ещё его называют, разветвляющийся алгоритм, обеспечивает выбор одного из двух альтернативных путей в зависимости от истинности условия.



В этом случае логика принятия решения может быть записана следующим образом:  
ЕСЛИ <условие> ТО <действия 1> ИНАЧЕ <действия 2> ВСЁ

Ветвление может быть реализовано в полной или в сокращённой форме (рис. 100–101).



Рис. 100. Полная форма условного алгоритма



Рис. 101. Сокращенная форма условного алгоритма

Условные алгоритмы реализуются практически в любом языке программирования. В среде Scratch для реализации разветвляющихся алгоритмов можно использовать два вида блоков, которые представлены в разделе «Управление».



Рис. 102. Блок для реализации сокращённой формы



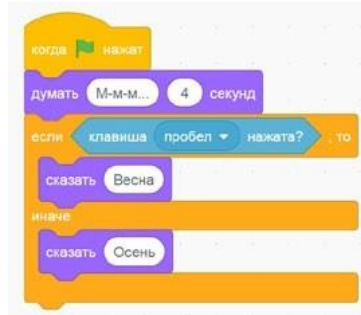
Рис. 103. Блок для реализации полной формы

### Практическая часть

Цель работы: ознакомление с основами работы с условными алгоритмами в среде Scratch.

#### Ход лабораторной работы

Создайте следующий скрипт (рис. 104). Объясните, что выполняет данный скрипт.



**Рис. 104.** Вид программы

Выполните в среде Scratch решение следующих задач.

1. Если введено число 1, то спрайт говорит «один», иначе спрайт говорит «не знаю».
2. Найдите наибольшее из двух чисел, введённых пользователем.
3. Проверьте, является ли введённое пользователем число чётным.
4. Если введённое пользователем число неотрицательно, то вычтите из него 50, в противном случае прибавьте к нему 100.

Вывод: в процессе выполнения лабораторной работы вы получили представление о создании условных алгоритмов в среде Scratch.

Контрольные вопросы:

- 1) Что называется условным алгоритмом?
- 2) Какие можно привести примеры условных алгоритмов?
- 3) Какие блоки используются в среде Scratch для реализации ветвлений?

## Лабораторная работа 4.2. «Условные алгоритмы»

### Теоретическая часть

Используется теоретический материал из лабораторной работы 4.1.

### Практическая часть

Цель работы: ознакомление с основами работы с условными алгоритмами в среде Scratch.

#### Ход лабораторной работы

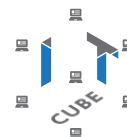
Выполните в среде Scratch решение следующих задач.

1. Определите, является ли введённое пользователем число двузначным.
2. Найдите наибольшее из трёх чисел, введённых пользователем.
3. Напишите программу решения линейного уравнения.
4. Спрайт загадывает число от 0 до 100. Пользователь должен угадать, больше ли число 30.

Вывод: в процессе выполнения лабораторной работы вы получили представление о создании условных алгоритмов в среде Scratch.

Контрольные вопросы:

- 1) Что называется условным алгоритмом?
- 2) Какие блоки вы использовали при решении задач в лабораторной работе?
- 3) Использовали ли вы переменные при решении задач?



## Лабораторная работа 5.1. «Циклические алгоритмы»

### Теоретическая часть

Операторы цикла используются для организации многократно повторяющихся вычислений. Любой цикл состоит из тела цикла, т. е. тех операторов, которые выполняются несколько раз, начальных установок, модификации параметра цикла и проверки условия продолжения выполнения цикла.

Один проход цикла называется итерацией. Проверка условия выполняется на каждой итерации либо до тела цикла (тогда говорят о цикле с предусловием), либо после тела цикла (цикл с постусловием). Разница между ними состоит в том, что тело цикла с постусловием всегда выполняется хотя бы один раз, после чего проверяется, надо ли его выполнять ещё раз. Проверка необходимости выполнения цикла с предусловием делается до тела цикла, поэтому возможно, что он не выполнится ни разу.

Для реализации циклических алгоритмов в среде Scratch используются несколько блоков.



Рис. 105. Блок для организации цикла

Данный блок используется для организации так называемого цикла с параметром, количество повторений которого заранее известно. Внутри данного блока обычно располагаются несколько других блоков, которые и образуют тело цикла. Блок-схема работы данного блока представлена на рисунке 106:



Рис. 106. Блок-схема цикла



Рис. 107. Блок для организации цикла

Данный блок используется для организации бесконечных повторений. Блок-схема работы данного блока представлена ниже (рис. 108):



Рис. 108. Блок-схема цикла

Также имеется блок для организации повторения тела цикла (блоков внутри), пока условие не станет истинным. Блок-схема работы данного блока представлена на рисунке 109:



Рис. 109. Блок-схема цикла

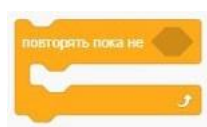


Рис. 110. Блок для организации цикла

### Практическая часть

Цель работы: ознакомление с основами работы с циклическими алгоритмами в среде Scratch.

#### Ход лабораторной работы

1. Выполните следующую программу.

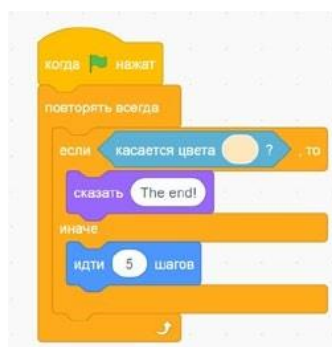
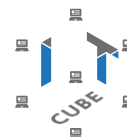


Рис. 111. Вид программы



2. Выполните следующую программу.

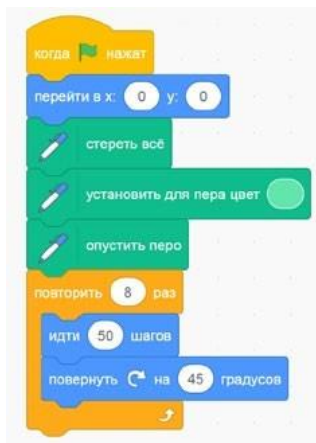


Рис. 112. Вид программы

Выполните в среде Scratch решение следующих задач.

3. Нарисуйте квадрат со стороной 100 (рис. 113).

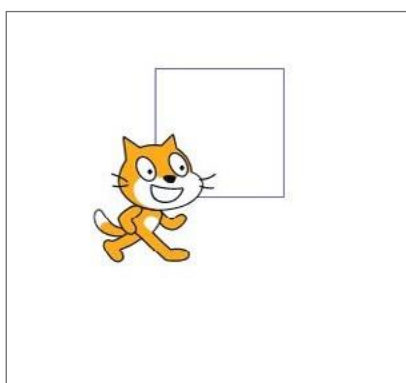


Рис. 113. Пример результата работы программы

4. Нарисуйте квадрат, величина стороны которого вводится пользователем.

5. Изобразите цифру 5.

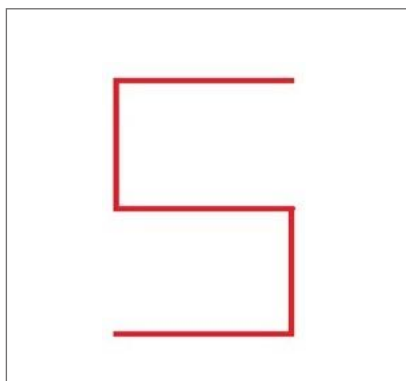
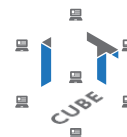


Рис. 114. Пример результата работы программы

Вывод: в процессе выполнения лабораторной работы вы получили представление о

создании циклических алгоритмов в среде Scratch.

*Рис. 111.* Вид программы



Контрольные вопросы:

- 1) Что называется циклическим алгоритмом?
- 2) Какие можно привести примеры циклических алгоритмов?
- 3) Какие блоки используются для организации циклических алгоритмов в среде Scratch?

## Лабораторная работа 5.2: «Циклические алгоритмы»

### Теоретическая часть

Используется теоретический материал из лабораторной работы 5.1.

### Практическая часть

Цель работы: ознакомление с основами работы с циклическими алгоритмами в среде Scratch.

#### Ход лабораторной работы

1. Выполните следующую программу. Объясните, что выполняет данный скрипт.

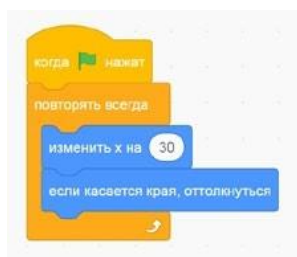


Рис. 115. Вид программы

2. Выполните следующую программу. Объясните, что выполняет данный скрипт.

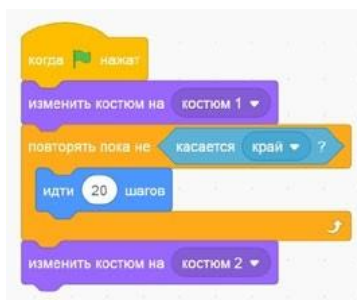
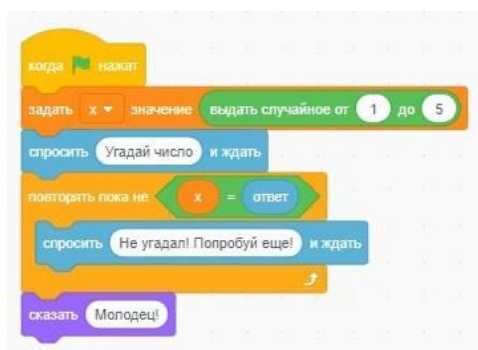


Рис. 116. Вид программы

3. Выполните следующую программу. Объясните, что выполняет данный скрипт.



Выполните в среде Scratch решение следующих задач.

4. Напишите скрипт, который двигается до края сцены и затем изменяет свой размер в 2 раза.

5. Найдите сумму пяти чисел, введенных пользователем.

Вывод: в процессе выполнения лабораторной работы вы получили представление о создании циклических алгоритмов в среде Scratch.

Контрольные вопросы:

- 1) Какие алгоритмы называются циклическими?
- 2) Какие блоки вы использовали при решении задач в лабораторной работе?
- 3) Как задать переменной случайное значение?

### Лабораторная работа 5.3. «Циклические алгоритмы»

#### Теоретическая часть

Используется теоретический материал из лабораторной работы 5.1.

#### Практическая часть

Цель работы: ознакомление с основами работы с циклическими алгоритмами в среде Scratch.

#### Ход лабораторной работы

Разработайте проект «Танцующий человек».

Примерный ход выполнения проекта.

1. Из коллекции спрайтов выберите спрайт Бен (рис. 118). Этот спрайт имеет несколько костюмов.

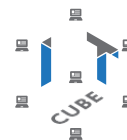


Рис. 118. Вид спрайта Бен

2. Добавьте из коллекции фонов подходящий фон (рис. 119).



Рис. 111. Вид программы



*Рис. 119.* Вид изменённого фона

3. Создайте следующий скрипт, который реализует изменение костюмов спрайта заданное число раз, например 20.

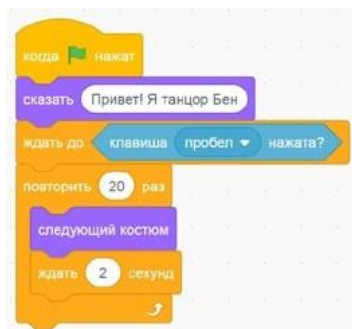


Рис. 120. Вид программы

4. Измените скрипт таким образом, чтобы Бен спрашивал у пользователя, сколько раз ему выполнить танец.

5. Создайте аналогичный проект с изменением спрайта заданное число раз.

Вывод: в процессе выполнения лабораторной работы вы получили представление о создании циклических алгоритмов в среде Scratch.

Контрольные вопросы:

- 1) Какие алгоритмы называются циклическими?
- 2) Какие блоки вы использовали при решении задач в лабораторной работе?
- 3) Как задать переменной значение, введённое с клавиатуры?

### Лабораторная работа 6.1. «Работа со списками»

#### Теоретическая часть

Списки в среде Scratch используются в том случае, если необходимо хранить в программе несколько значений. Список представляет собой пронумерованные строки с данными. Список имеет имя, для обращения к элементам списка используются числовые индексы. Первый элемент списка в среде Scratch имеет номер 1. Пустой список — это список, в котором ещё нет ни одного элемента. Списки в среде Scratch аналогичны массивам в языках программирования Pascal, C++ и т. д.

Для создания списка в среде Scratch необходимо обратиться к разделу «Переменные», выбрать команду «Создать список» (рис. 121).

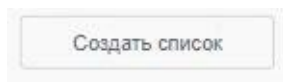
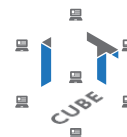


Рис. 121. Блок создания списка



Далее появляется диалоговое окно, аналогичное диалоговому окну для создания переменной (рис. 122).

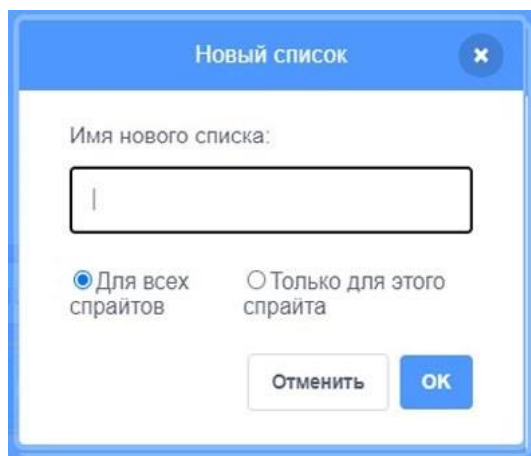


Рис. 122. Диалоговое окно для задания имени списка

Созданный список также может отображаться на сцене (рис. 123).

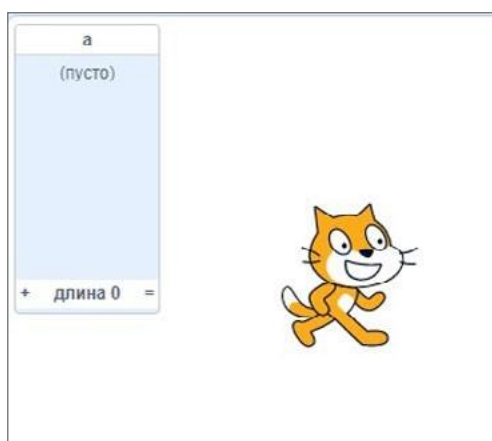


Рис. 123. Вид пустого списка

Далее опишем заполнение списка. Можно добавить элемент в список на сцене, нажимая знак «+» внизу списка.

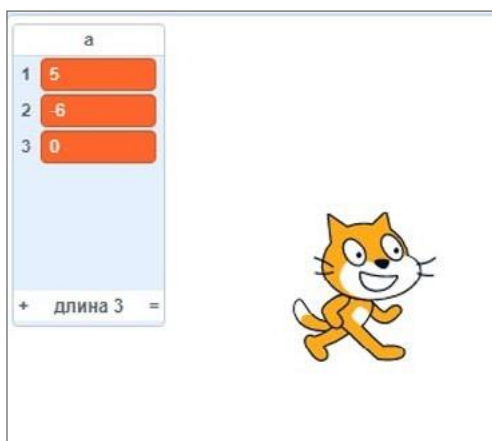
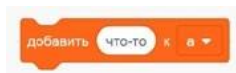


Рис. 124. Вид заполненного списка

Как можно обратить внимание, на сцене также отображается количество элементов в списке: «длина 3». Элементы списка также можно удалить, для этого необходимо нажать крестик рядом с тем элементом списка, который вы хотите удалить.

Также для добавления элемента в конец из списка используется блок «Добавить ... к списку» (рис. 125).



**Рис. 125.** Блок для добавления элемента к списку

Также имеется специальный блок, который вставляет значение в конкретную позицию (рис. 126).



**Рис. 126.** Блок для добавления элемента к списку

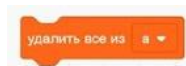
Для удаления элемента из списка также может использоваться специальный блок «Удалить ... из списка» (рис. 127).



**Рис. 127.** Блок для удаления элемента из списка

В данном блоке указывается номер удаляемого элемента в качестве параметра команды. Удаление уменьшает длину списка.

Также можно удалить все элементы из списка специальным блоком (рис. 128).



**Рис. 128.** Блок удаления всех элементов из списка

В результате список становится пустым, т. е. длиной 0.

Также имеется блок для замены конкретного элемента в списке, для чего в блоке указывается номер элемента и новое значение (рис. 129).



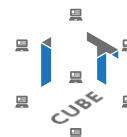
**Рис. 129.** Блок для замены элемента в списке

Опишем ещё несколько блоков для работы со списками в среде Scratch.

Блок для получения элемента из списка по его номеру представлен на рисунке 130.



**Рис. 130.** Блок для получения элемента из списка



Блок для получения длины списка представлен на рисунке 131.



Рис. 131. Блок для получения длины списка

Блок для проверки принадлежности элемента списку представлен на рисунке 132.



Рис. 132. Блок для проверки принадлежности элемента списку

Также можно выполнить поиск номера вхождения конкретного элемента в списке с помощью специального блока (рис. 133).

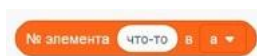


Рис. 133. Блок для поиска номера элемента в списке

В среде имеется возможность загружать значения списка из файла, а также сохранять значения списка в файл. Для этого используются команды «Экспорт» и «Импорт» (рис. 134).

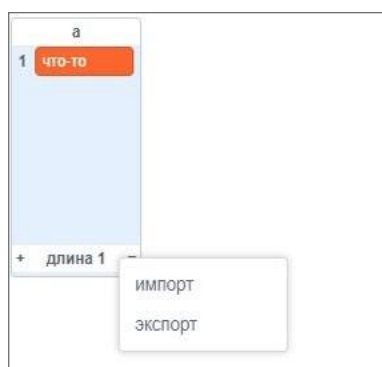


Рис. 134. Команды для работы с файлами

Считается, что списки хранятся в обычном текстовом файле по одному элементу в каждой строке. Например, загрузим в список числовые данные, хранящиеся в текстовом файле data1.txt.

Для этого подготовим файл (рис. 135), задав числовые значения по одному в столбик.

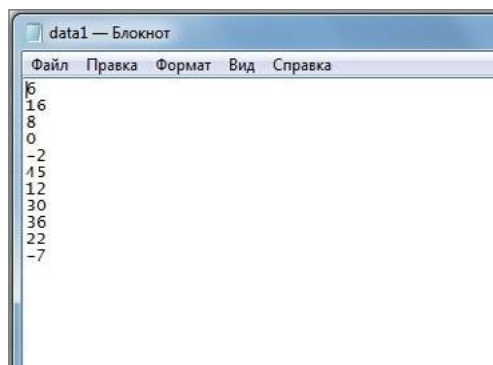


Рис. 135. Вид файла с данными

Затем выберем команду «Импорт» и в открывшемся диалоговом окне выберем соответствующий файл.

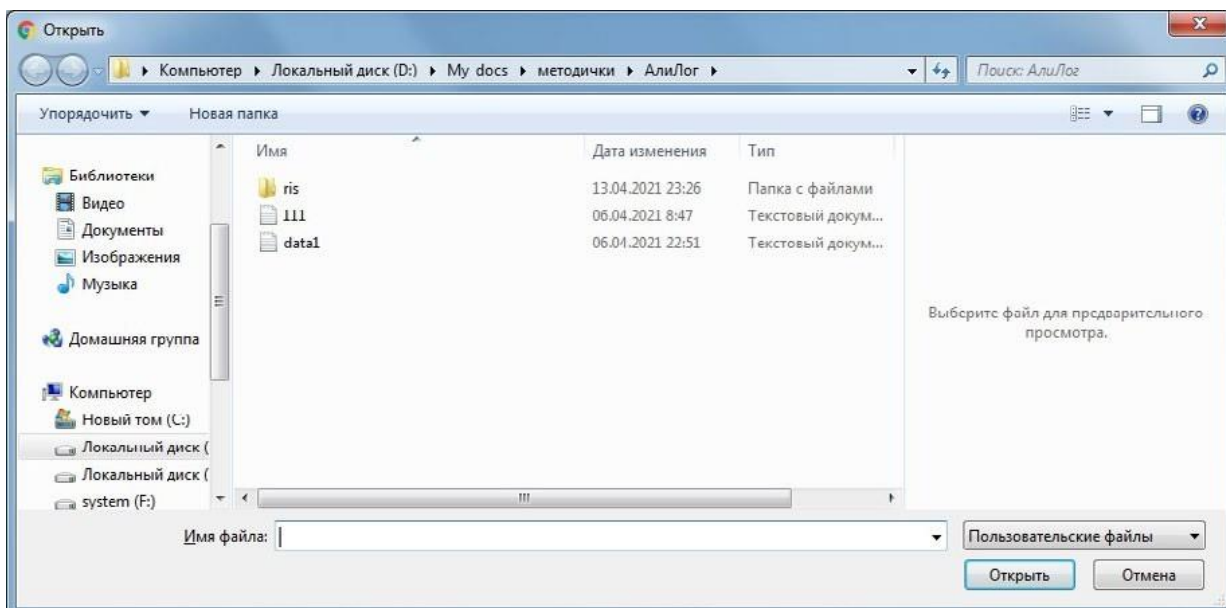


Рис. 136. Диалоговое окно для выбора файла

В результате данные из файла загрузятся в список (рис. 137) и с ними можно в дальнейшем работать.



Рис. 137. Вид списка с загруженными значениями

Аналогичным образом осуществляется сохранение списка в файл с помощью команды «Экспорт» (рис. 138). В результате создаётся текстовый файл, содержащий элементы списка.

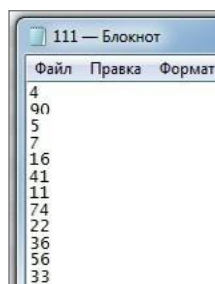
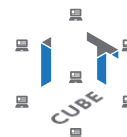


Рис. 138. Вид файла с сохранёнными данными



Именно так он и сам сохраняет списки в файле.

### Практическая часть

Цель работы: ознакомление с основами работы со списками в среде Scratch.

#### Ход лабораторной работы

1. Создайте список из названий месяцев, как показано на рисунке 139.

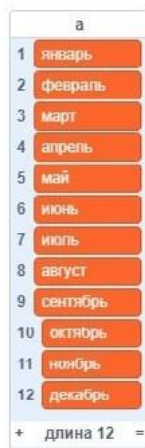


Рис. 139. Вид списка

2. Напишите программу в среде Scratch, удаляющую из списка третий элемент.
3. Создайте список, представленный на рисунке 140.



Рис. 140. Вид списка

4. Напишите программу в среде Scratch, которая проверяет, есть ли в списке элемент, равный 5.

5. Напишите программу в среде Scratch, которая проверяет, находится ли в списке элемент, введенный пользователем.

Вывод: в ходе выполнения лабораторной работы вы получили представление о создании и работе со списками в среде Scratch.

Контрольные вопросы:

- 1) Что собой представляют списки в среде Scratch?
- 2) Какие команды используются в среде Scratch для создания списков?
- 3) Какие основные блоки используются в среде Scratch для работы со списками?

## Лабораторная работа 6.2. «Работа со списками»

### Теоретическая часть

Используется теоретический материал из лабораторной работы 6.1.

### Практическая часть

Цель работы: ознакомление с основами работы со списками в среде Scratch.

#### Ход лабораторной работы

1. Создайте файл, в котором хранятся 10 целых чисел.

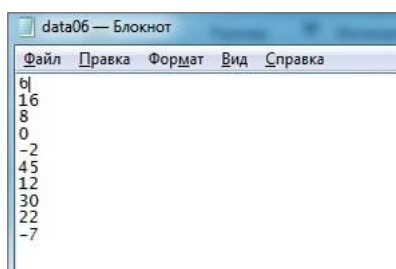


Рис. 141. Примерный вид файла с данными

2. Создайте список d в среде Scratch в разделе переменные.
3. Импортируйте данные из файла в список. Для этого выберите команду «Импорт».



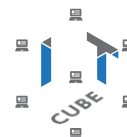
Рис. 142. Вид импортированного списка

4. Выведите на экран только нечётные элементы списка.
5. Найдите максимальный элемент списка.
6. Подсчитайте сумму чётных элементов списка.

Вывод: в процессе выполнения лабораторной работы вы получили представление о создании и работе со списками в среде Scratch.

Контрольные вопросы:

- 1) Что называется списком?
- 2) Какие блоки по работе со списками вы использовали при решении задач в лабораторной работе?
- 3) Какие блоки вы использовали для решения задач в лабораторной работе?



## Лабораторная работа 7. «Создание подпрограмм»

### Теоретическая часть

Подпрограммы являются средством реализации структурного программирования. Подпрограммы используются для следующих целей: многократное использование в программе одного и того же фрагмента, когда происходит разделение сложной программы на составные части — процедурной декомпозиции. Также подпрограммы являются средством реализации восходящего и нисходящего проектирования.

В среде Scratch также имеются возможности создания и использования подпрограмм. Для этого необходимо обратиться к разделу «Другие блоки». В данном разделе представлена всего одна команда — «Создать блок». При выборе данной команды появится диалоговое окно, представленное на рисунке 143.

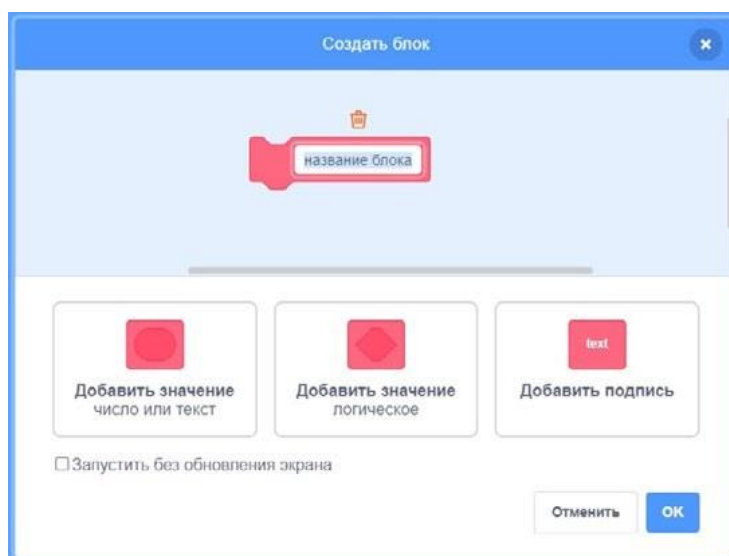


Рис. 143. Вид диалогового окна для создания подпрограммы (нового блока)

В этом окне необходимо ввести название нового блока, а также указать его параметры.

Блок можно создать и без параметров. Например, назовём его «треугольник». После того как вводится название нового блока, нажмём ОК. Созданный блок появится в списке команд (рис. 144).

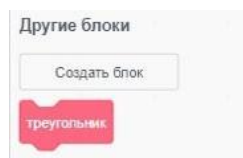


Рис. 144. Вид нового блока

Также в области для создания скриптов появится команда для определения скриптов нового блока (рис. 145).

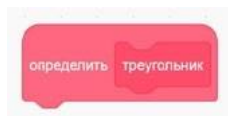


Рис. 145. Вид нового блока в области скриптов

Далее добавляем скрипт для рисования треугольника (рис. 146).

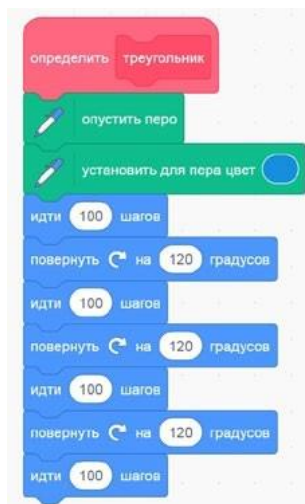


Рис. 146. Вид скрипта

Далее к этому блоку можно обращаться многократно для изображения треугольника. Создадим скрипт, который рисует один треугольник при нажатии клавиши пробел (рис. 147).

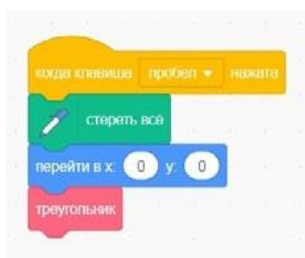


Рис. 147. Вид программы с новым блоком

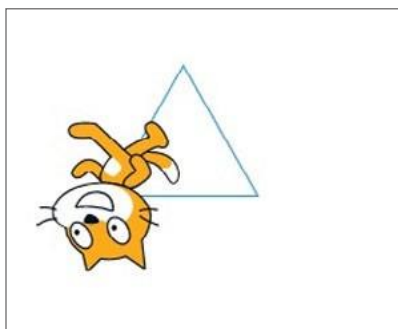


Рис. 148. Результат работы скрипта

Далее создадим блок для изображения нескольких треугольников (рис. 149).

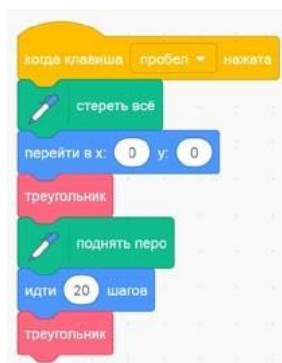


Рис. 149. Вид программы с новым блоком

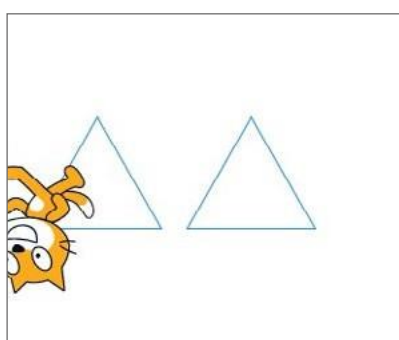


Рис. 150. Результат работы скрипта

Как видно на рисунке, скрипт изобразил два треугольника.

Как было сказано ранее, новый блок может иметь параметры. Параметры служат для передачи данных вовнутрь подпрограммы. Разработаем новый блок «Треугольник1», у которого в качестве параметра будет размер треугольника. Также обратимся к команде «Создать блок». Но теперь в диалоговом окне необходимо также указать параметр — «Добавить значение число или текст» (рис. 151). При необходимости можно добавить и второй параметр, также выбрав соответствующую команду «Добавить значение».



Рис. 151. Вид диалогового окна для создания подпрограммы с параметром



**Рис. 152.** Вид диалогового окна для создания подпрограммы с несколькими параметрами

Новый блок также отображается в соответствующем разделе (рис. 153).



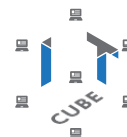
**Рис. 153.** Вид двух новых блоков

Далее создаём новый скрипт, используя параметры (рис. 154).



**Рис. 154.** Вид скрипта для создания нового блока

Как можно увидеть, в блоке включаются новые параметры, их можно просто перетаскивать мышью в необходимые пазы.



Далее создаётся скрипт, изображающий два треугольника (рис. 155).

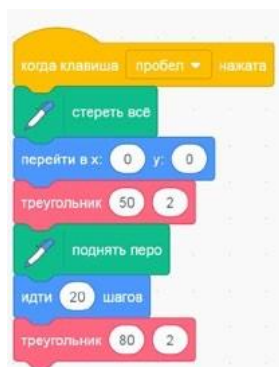


Рис. 155. Вид программы с новым блоком

Таким образом, можно создавать несколько новых блоков, использовать их в программе в среде Scratch.

### Практическая часть

Цель работы: ознакомление с основами создания блоков-подпрограмм в среде Scratch.

#### Ход лабораторной работы

1. Создайте в среде Scratch новый блок, который реализует движение по сцене до края.
2. Откройте раздел «Другие блоки» и выберите команду «Создать блок».
3. Создайте блок без параметров и назовите его «Движение» как на рисунке 156.

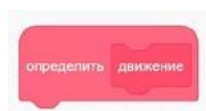


Рис. 156. Создание нового блока

4. Для нового блока реализуйте следующий скрипт, представленный на рисунке.

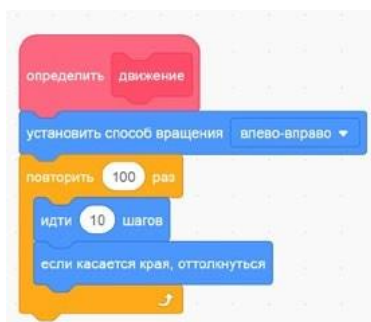
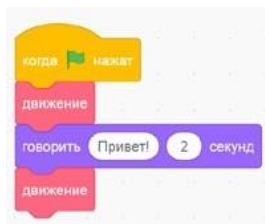


Рис. 157. Вид скрипта для нового блока

5. Создайте новый скрипт, в котором используется блок «Движение» (рис. 158).



**Рис. 158.** Вид скрипта с новым блоком

6. Создайте новый блок для изображения квадрата. Блок должен содержать параметр — размер стороны квадрата.

Вывод: в процессе выполнения лабораторной работы вы получили представление о создании новых блоков в среде Scratch.

Контрольные вопросы:

- 1) Что называется подпрограммой?
- 2) Для чего используются новые блоки в среде Scratch?
- 3) Для чего используются параметры при разработке новых блоков в среде Scratch?

## Примеры конспектов уроков

### Тема урока «Знакомство со средой Scratch»

Тип урока: комбинированный.

Цель урока: ознакомление со средой Scratch, изучение основных инструментов среды.

#### Планируемые результаты:

Предметные: получение навыков по работе в среде Scratch, освоение основных инструментов среды.

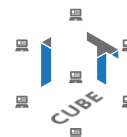
Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Время реализации: 1 академический час.

Оборудование и материалы:

Компьютер, проектор.



### Ход урока

I. Этап постановки цели и задач урока, мотивации к учебной деятельности — 5 мин.

Деятельность учителя: предлагает учащимся начать знакомство с основами алгоритмизации. Каждый из нас ежедневно сталкивается с множеством задач, например, в каком порядке стоит решать задачу по математике. Сложности решения данных задач могут существенно отличаться: от нескольких действий до сложных размышлений, возможно требующих работы с информационными источниками, консультаций. Многие задачи решаются с помощью последовательности шагов. Приведите примеры таких задач, опишите последовательность действий, которые вы выполняете для их решения.

Деятельность учеников: приводят примеры задач, с которыми они сталкивались сами, предлагают последовательности действий для их решения.

Деятельность учителя: сообщает, что последователь действий, необходимых для решения задачи, называется алгоритмом решения задачи. Начинается знакомство с основами алгоритмизации. В этом поможет среда Scratch.

II. Этап актуализации знаний и пробного учебного действия — 8 мин.

Деятельность учителя: предлагает учащимся начать работу с онлайн-версией редактора Scratch, расположенного по адресу <https://scratch.mit.edu/projects/504430540/editor>. Предлагает рассмотреть основные элементы интерфейса редактора (рис. 159).

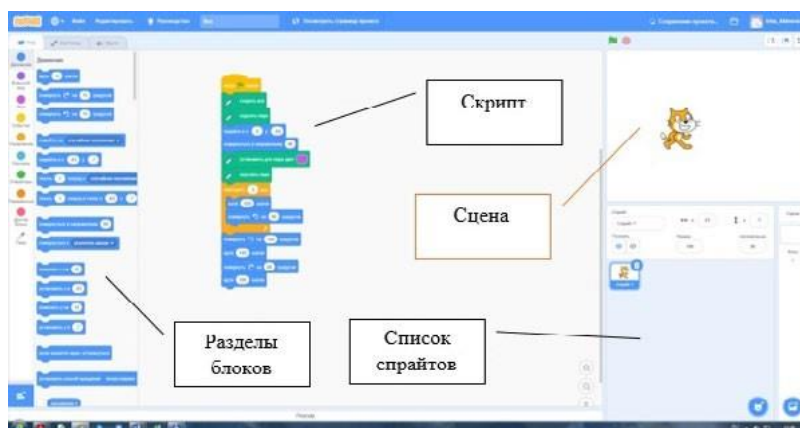


Рис. 159. Интерфейс среды Scratch

Деятельность учеников: загружают онлайн-версию редактора Scratch, изучают основные элементы интерфейса.

Деятельность учителя: кратко описывает основные элементы интерфейса. Спрайт представляет собой персонаж, для которого пишется программа-скрипт. Сцена представляет собой место, холст, по которому двигается персонаж-спрайт. Скрипт — это набор блоков, которые выполняет персонаж. Скрипт можно составить, перетаскивая блоки из различных разделов с помощью мыши. Блоки имеют различные функции, будем постепенно знакомиться с ними в течение нашего курса.

Дидактические и методические материалы: раздел «Описание среды Scratch» дидактических материалов пособия.

III. Этап изучения нового материала — 16 мин.

Деятельность учителя: предлагает учащимся выполнить несколько действий в среде Scratch.

1. Зайдите в закладку «События» (рис. 160).



Рис. 160. Вид среды Scratch

2. Перетащите мышью блок «Когда  нажат» (рис. 161).



Рис. 161. Вид среды Scratch

3. Откройте закладку «Движение» и перетащите блок «Идти ... шагов». Подсоедините новый блок к первому (рис. 162).

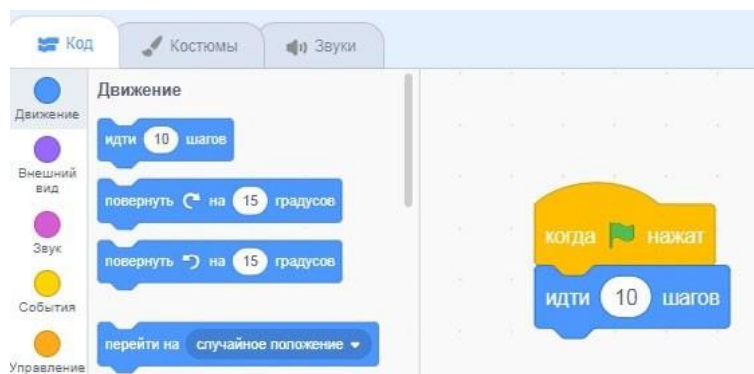
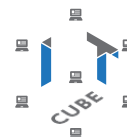


Рис. 162. Вид среды Scratch



4. Измените значение в блоке «Идти ... шагов» на 200 (рис. 163). В итоге получили первый скрипт в среде Scratch.



Рис. 163. Вид скрипта в среде Scratch

5. Запустите скрипт, нажав на зелёный флажок  над сценой.

Как изменилось положение спрайта?

Деятельность учеников: выполняют шаги, предложенные учителем.

Деятельность учителя: предлагает учащимся составить следующий скрипт в среде Scratch (рис. 164).

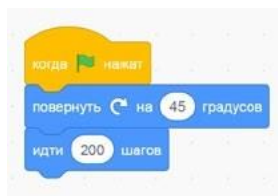


Рис. 164. Вид скрипта в среде Scratch

Деятельность учеников: выполняют скрипт, предложенный учителем.

Деятельность учителя: предлагает учащимся составить следующий скрипт в среде Scratch (рис. 165). Как изменилась сцена? Выполните скрипт несколько раз.

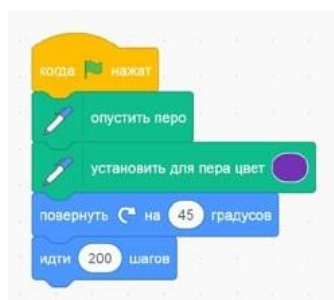


Рис. 165. Вид скрипта в среде Scratch

Деятельность учеников: выполняют скрипт, предложенный учителем, отвечают на вопросы учителя.

Деятельность учителя: предлагает учащимся составить следующий скрипт в среде Scratch (рис. 166). Чем данный скрипт отличается от предыдущего? Как различаются результаты их работы?

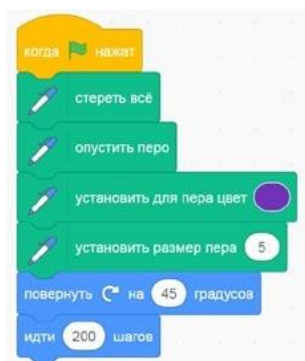


Рис. 166. Вид скрипта в среде Scratch

Деятельность учеников: выполняют скрипт, предложенный учителем, отвечают на вопросы учителя.

Деятельность учителя: предлагает учащимся составить следующий скрипт в среде Scratch (рис. 167). Чем данный скрипт отличается от предыдущего? Как различаются результаты их работы?

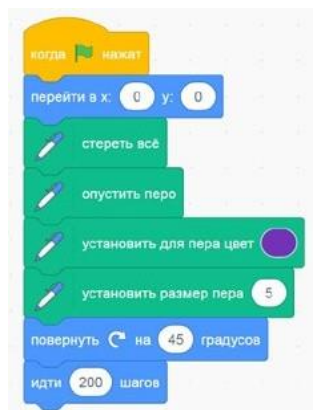


Рис. 167. Вид скрипта в среде Scratch

Деятельность учеников: выполняют скрипт, предложенный учителем, отвечают на вопросы учителя.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

IV. Этап проверки понимания и первичного закрепления — 8 мин.

Деятельность учителя: предлагает учащимся выполнить следующие задания.

1. Выполните скрипт, опишите результаты работы (рис. 168).

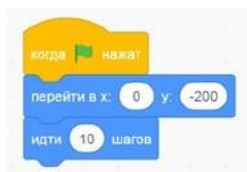
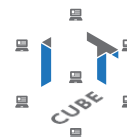


Рис. 168. Вид скрипта в среде Scratch



2. Зайдите в библиотеку спрайтов. Для этого обратитесь к инструменту, представленному на рисунке 169.



Рис. 169. Инструменты для редактирования и выбора спрайта

Смените вид спрайта на Жука (рис. 170–171).



Рис. 170. Вид спрайта Жук

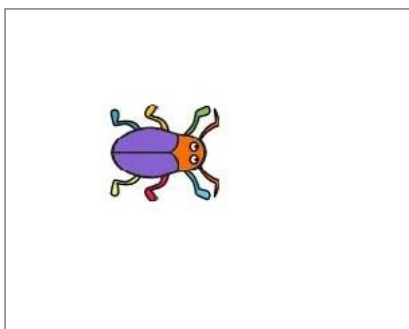


Рис. 171. Вид нового спрайта на сцене

3. Зайдите в графический редактор среды Scratch и измените вид спрайта.

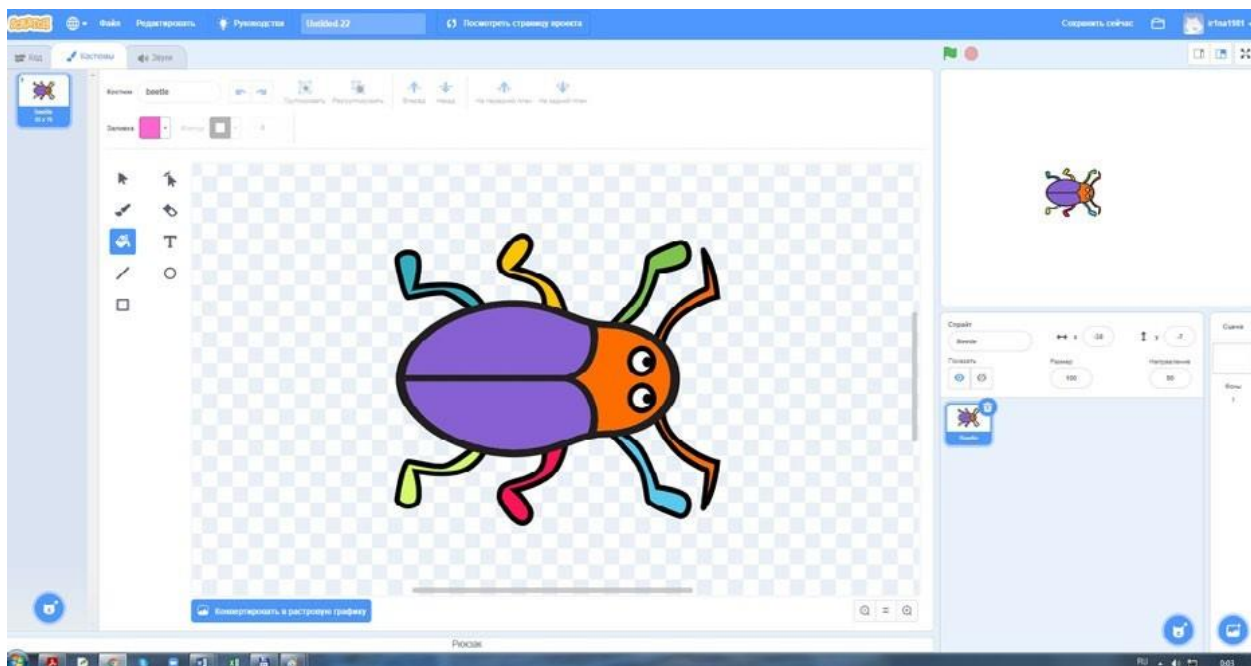


Рис. 172. Изменение спрайта в графическом редакторе

Деятельность учеников: самостоятельно выполняют задания учителя.

Дидактические и методические материалы: раздел «Описание графического редактора» дидактических материалов пособия.

V. Этап контроля усвоения и коррекции ошибок — 4 мин.

Деятельность учителя: предлагает учащимся ответить на следующие вопросы:

- 1) Что собой представляет алгоритм? Приведите примеры алгоритмов.
- 2) Какие основные элементы интерфейса среды Scratch вы можете назвать?
- 3) Как изменить спрайт в среде Scratch?

Деятельность учеников: отвечают на вопросы учителя.

VI. Информация о домашнем задании, инструктаж по его выполнению — 2 мин.

Деятельность учителя: предлагает учащимся дома выполнить следующие задания в среде Scratch:

- 1) Создать свой собственный спрайт в графическом редакторе среды Scratch.
- 2) Создать свой собственный фон для сцены в среде Scratch.

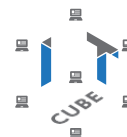
VII. Этап рефлексии деятельности на уроке — 2 мин.

Деятельность учителя: предлагает учащимся оценить свою деятельность на уроке. Для этого можно воспользоваться различными интерактивными ресурсами, например <https://www.mentimeter.com/app>.

Учащимся предлагается оценить свою деятельность на уроке, выбрав в смартфоне один из вариантов ответов (рис. 173):



Рис. 173. Интерактивный ресурс для рефлексии деятельности учащихся на занятии



### Тема урока «Условные алгоритмы»

Тип урока: комбинированный.

Цель урока: ознакомление с построением и выполнением условных алгоритмов в среде Scratch.

#### Планируемые результаты:

Предметные: получение навыков по построению и выполнению условных алгоритмов в среде Scratch, работа с основными инструментами среды.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Время реализации: 1 академический час.

Оборудование и материалы:

Компьютер, проектор.

#### Ход урока

I. Этап постановки цели и задач урока, мотивации к учебной деятельности — 5 мин.

Деятельность учителя: сообщает учащимся, что линейная последовательность действий в реальной жизни встречается крайне редко. Чаще приходится принимать решение в зависимости от сложившейся ситуации. Можно попросить учащихся привести несколько своих примеров, когда необходимо делать выбор. Например, если завтра урок по геометрии, то необходимо взять учебник по геометрии, если же урока нет, то его можно оставить дома. Можете ли вы предложить примеры таких задач, где необходимо выбрать действие в зависимости от выполнения некоторого условия?

Деятельность учеников: приводят примеры из своей жизнедеятельности.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

II. Этап актуализации знаний и пробного учебного действия — 8 мин.

Деятельность учителя: предлагает учащимся вспомнить, как выполняется построение линейного алгоритма в среде Scratch.

Деятельность учеников: описывают свои действия при построении линейной программы в среде Scratch.

Деятельность учителя: предлагает учащимся ответить на вопросы. Какие блоки вы использовали для построения линейных программ? В каких разделах эти блоки находились?

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

III. Этап изучения нового материала — 16 мин.

Деятельность учителя: предлагает учащимся решить следующую задачу.

Задача 1. Найти наибольшее из двух чисел.

Учитель вместе с учащимися строит в среде решение задачи. Сначала необходимо задать в программе две переменные, предлагается назвать их  $x$  и  $y$  (рис. 174–175).

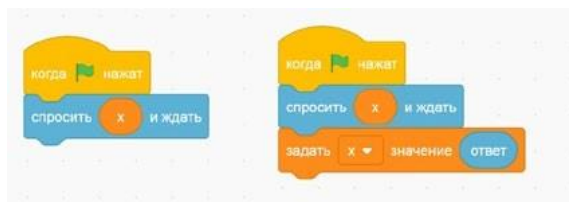


Рис. 174. Разработка алгоритма решения задачи



Рис. 175. Разработка алгоритма решения задачи

Затем предлагается обратиться к команде Scratch, реализующей ветвление (рис.176).

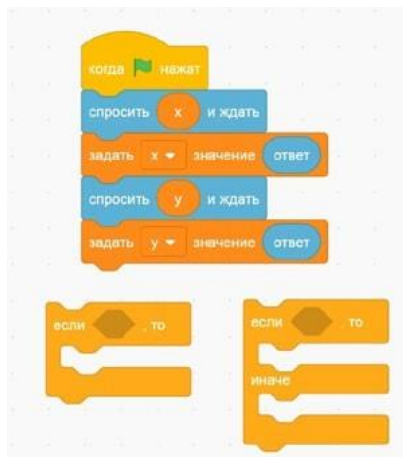
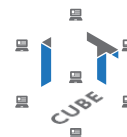


Рис. 176. Разработка алгоритма решения задачи

Учитель обращает внимание, что в среде имеются два вида ветвления, с реализацией блока ИНАЧЕ и без него. Предлагает учащимся выбрать, какая команда нужна для решения данной задачи.

Учащиеся могут предложить использовать полную форму — с блоком ИНАЧЕ.



Тогда решение примет следующий вид (рис.177).



Рис. 177. Разработка алгоритма решения задачи

Предлагается выполнить данный алгоритм для различных значений (рис. 178–179).

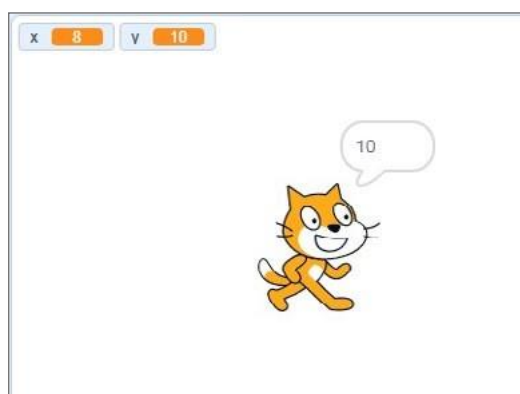


Рис. 178. Результат работы алгоритма



Рис. 179. Результат работы алгоритма

Учитель предлагает также рассмотреть ситуацию, когда введенные числа равны. Будет ли влиять на ответ постановка строгого знака «больше» в алгоритме?

Учащиеся тестируют данный случай (рис. 180).



**Рис. 180.** Результат работы алгоритма

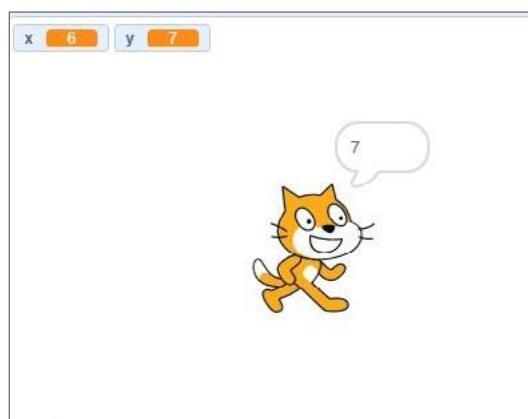
Делают вывод, что программа также в этом случае выдаст верный ответ: работает блок ИНАЧЕ. И поскольку значения переменных равны, то выведется на экран значение переменной  $y$ .

Если же учащиеся предлагали вариант — сокращенная форма оператора ветвления, то им учитель предлагает проверить работу такой программы (рис. 181).



**Рис. 181.** Программа с неполной формой

Результат работы программы представлен ниже (рис. 182—183).



**Рис. 182.** Результат работы программы

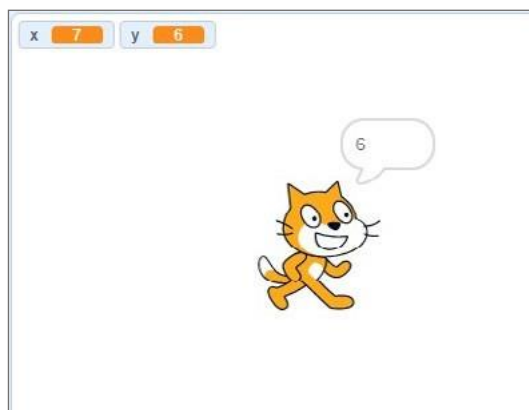
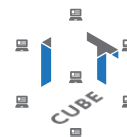


Рис. 183. Результат работы программы

Учащиеся должны сделать вывод, что данная программа работает некорректно.

Далее учитель предлагает решать следующую задачу.

Задача 2. Найти наибольшее из трёх чисел.

Учитель предлагает учащимся подумать, какие варианты можно предложить.

На основе решения предыдущей задачи учащиеся могут высказать несколько предположений о поиске алгоритма решения задачи 2.

Если решение задачи изначально вызывает затруднения, то учитель может задавать наводящие вопросы, чтобы ученики, отвечая на них, смогли высказать свои гипотезы.

Учитель: Сколько переменных необходимо сравнить?

Учащиеся: Три. Значит, добавляем в программу ещё одну переменную.

Предлагается добавить в программу ещё одну переменную (рис. 184).



Рис. 184. Разработка алгоритма решения задачи

Далее учитель спрашивает, можно ли применить тот алгоритм, который использовался в первой задаче.

Учащиеся предлагают свои варианты построения программы решения данной задачи. В качестве примера можно показать алгоритм, представленный на рисунке 185.



Рис. 185. Разработка алгоритма решения задачи

Учащиеся тестируют составленную программу для нескольких значений переменных (рис.186).

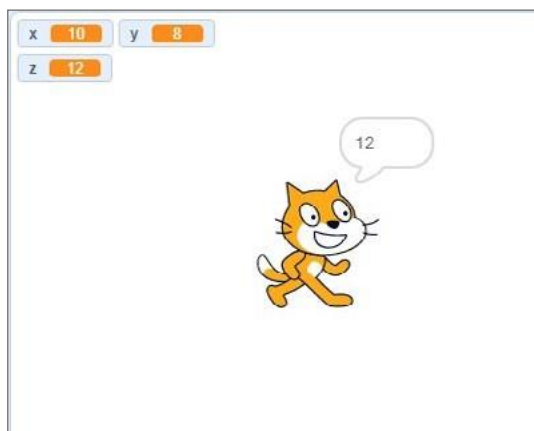
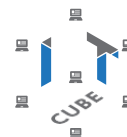


Рис. 186. Результат работы программы

Учитель спрашивает, можно ли не использовать вложенное условие; объединить два условия в одно.



Учащиеся предлагают ещё несколько вариантов решения. Один из них представлен на рисунке 187.



Рис. 187. Разработка алгоритма решения задачи

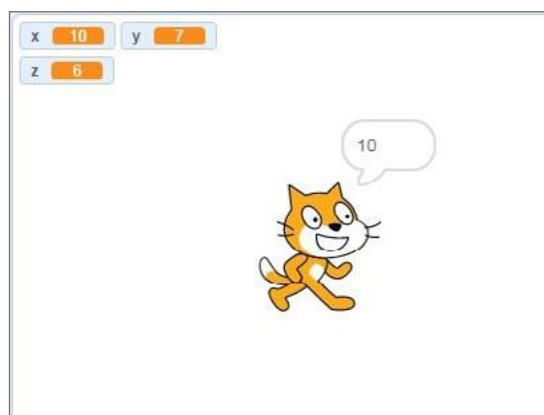


Рис. 188. Результат работы программы

IV. Этап проверки понимания и первичного закрепления — 8 мин.

Деятельность учителя: предлагает учащимся решить следующую задачу.

Даны три числа. Вывести полусумму наибольшего и наименьшего из них.

Деятельность учеников: решают задачу, предложенную учителем, тестируют её при различных введённых значениях.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

V. Этап контроля усвоения и коррекции ошибок — 4 мин.

Деятельность учителя: предлагает ответить учащимся на следующие вопросы:

- 1) Какой алгоритм является условным?
- 2) Какие можно привести примеры условных алгоритмов?
- 3) Какие блоки вы использовали для разработки условных программ?

Деятельность учеников: отвечают на вопросы учителя.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

VI. Информация о домашнем задании, инструктаж по его выполнению — 2 мин.

Деятельность учителя: сообщает учащимся домашнее задание. Выполните решение следующей задачи:

Написать программу, которая запрашивает у пользователя день недели, и выводит сообщение — рабочий день или выходной.



Рис. 189. Примерный вид ответа на задачу

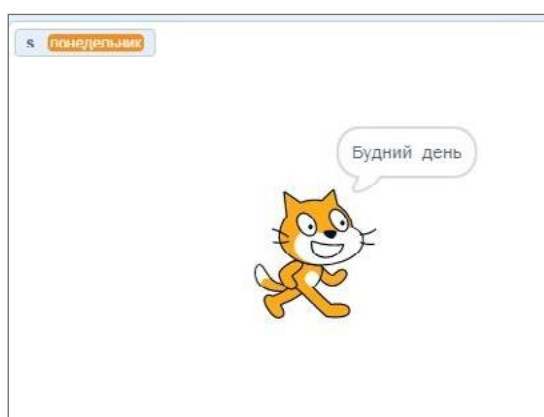


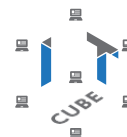
Рис. 190. Примерный вид ответа на задачу

VII. Этап рефлексии деятельности на уроке — 2 мин.

Деятельность учителя: предлагает учащимся оценить свою деятельность на уроке с помощью одного из интерактивных ресурсов.

Для оценки своей деятельности учащимся предлагается продолжить следующие предложения:

- 1) Сегодня на уроке я узнал...
- 2) Сегодня я научился в среде Scratch...
- 3) Мне хотелось бы в дальнейшем работать в среде Scratch, так как ...



## Тема урока «Циклические алгоритмы»

Тип урока: комбинированный.

Цель урока: ознакомление с построением и выполнением линейных алгоритмов в среде Scratch.

### Планируемые результаты:

Предметные: получение навыков по построению и выполнению циклических алгоритмов в среде Scratch, работа с основными инструментами среды.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности; осознанный выбор будущей профессии на основе понимания её ценностного содержания и возможностей реализации собственных жизненных планов.

Время реализации: 1 академический час.

Оборудование и материалы:

Компьютер, проектор.

### Ход урока

I. Этап постановки цели и задач урока, мотивации к учебной деятельности — 5 мин.

Деятельность учителя: сообщает учащимся, что циклические алгоритмы представляют собой третий вид алгоритмов. Разные виды циклов встречаются очень часто при составлении алгоритмов различной степени сложности. Это связано с тем, что в реальной жизни нас окружают задачи, решение которых требует многократного выполнения одной и той же группы действий. Например, когда вы хотите построить график функции по точкам, то многократно необходимо выполнить следующие действия: вычислить значение функции по значению аргумента, построить точку с получившимися координатами на координатной плоскости. Можете ли вы привести примеры подобных повторений действий, которые необходимо выполнить для решения задачи?

Деятельность учеников: приводят примеры циклических алгоритмов.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

II. Этап актуализации знаний и пробного учебного действия — 8 мин.

Деятельность учителя: спрашивает у учащихся, какие блоки они использовали для построения линейных и разветвляющихся программ в среде Scratch. Какие блоки используются для рисования?

Деятельность учеников: отвечают на вопросы учителя.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

III. Этап изучения нового материала — 16 мин.

Деятельность учителя: ставит перед учениками следующую задачу.

Задача 1. Нарисовать квадрат.

Учащиеся могут предложить следующее решение (рис. 191).

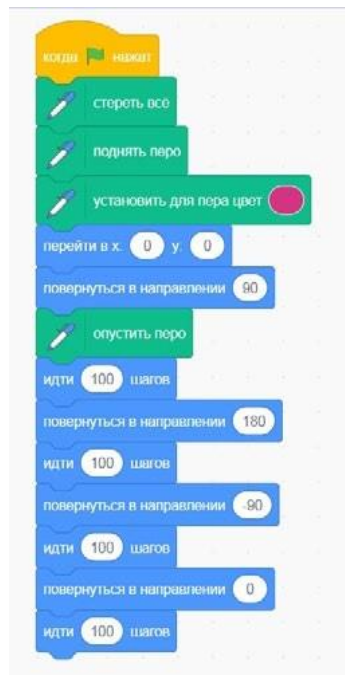


Рис. 191. Разработка алгоритма решения задачи

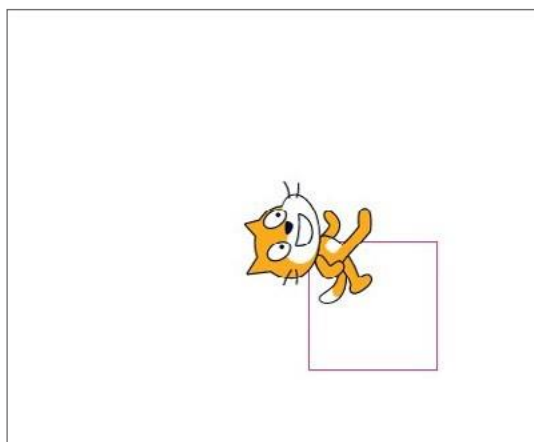


Рис. 192. результат работы программы

Учитель спрашивает, всё ли нас устраивает в решении задачи; нет ли чего лишнего.

Учащиеся отмечают, что они несколько раз выполняли одно и то же действие.

Учитель ставит проблему: можно ли ту же задачу решить короче? Какое действие вы повторяете многократно?

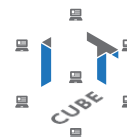
Учащиеся выделяют следующую группу действий: пройти 100 шагов, повернуть в определённом направлении.

Учитель сообщает, что такая алгоритмическая конструкция, когда многократно выполняется одно и то же действие, называется цикл.

Предлагается данную задачу решить с помощью цикла.

Учитель: Сколько раз нам надо выполнить одну и ту же последовательность действий?

Ученики: Четыре раза.



Учитель: Выбираем конструкцию — повторить.

Какое именно действие?

Ученики: Пройти 100 шагов.

Учитель: Ставим их в цикл.

Еще какую?

Ученики: Повернуть на 90 градусов.

Учитель: Получаем следующий алгоритм (рис. 193).



Рис. 193. Алгоритм решения задачи

Результат работы представлен на рисунке 194.

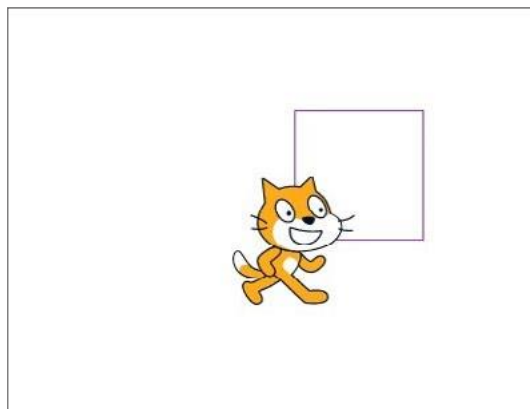


Рис. 194. Результат работы программы

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

IV. Этап проверки понимания и первичного закрепления — 8 мин.

Деятельность учителя: предлагает учащимся для решения следующую задачу.

Задача 2. Изобразите цифру 5.

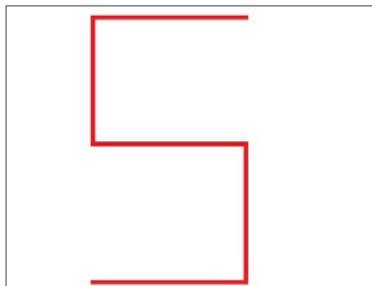


Рис. 195. Постановка задачи

Деятельность учеников: выполняют задание учителя.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

V. Этап контроля усвоения и коррекции ошибок — 4 мин.

Деятельность учителя: предлагает учащимся ответить на следующие вопросы:

- 1) Какой алгоритм называется циклическим?
- 2) Какие можно привести примеры циклических алгоритмов?
- 3) Какие блоки в среде Scratch используются для реализации циклических алгоритмов?

Деятельность учеников: отвечают на вопросы учителя.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

VI. Информация о домашнем задании, инструктаж по его выполнению — 2 мин.

Деятельность учителя: предлагает учащимся дома выполнить решение следующей задачи: «На тренировке спортсмен ежедневно пробегает некоторую дистанцию, с каждым днём увеличивая её на 10%. Составить программу, определяющую по расстоянию, преодоленному спортсменом в первый день тренировки, длину дистанции на  $k$ -ый день». При решении данной задачи используйте соответствующий фон и спрайт.

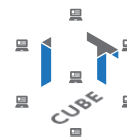
Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

В качестве решения домашнего задания можно предложить следующий проект в среде Scratch.

В качестве спрайта можно выбрать готового персонажа, внешне похожего на спортсмена (рис. 196).



Рис. 196. Вид персонажа



Также можно выбрать подходящий фон для сцены.

Ниже представлен примерный вариант программы в среде Scratch для решения данной задачи.



Рис. 197. Вид программы в среде Scratch

В данной программе использован блок для реализации цикла с параметром. Для решения нами были добавлены две переменные:  $n$  — для обозначения начальной дистанции,  $k$  — для задания количества дней.

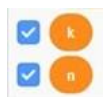


Рис. 198. Вид переменных

Значения этих переменных предполагается ввести с клавиатуры. Также в начале программы реализован небольшой диалог для объяснения пользователю, какие данные необходимо ввести.

Ниже представлен вид работы данного скрипта.



Рис. 199. Вид сцены

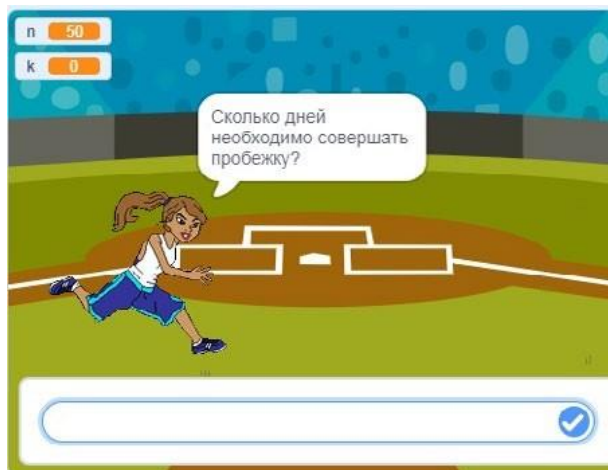


Рис. 200. Вид сцены



Рис. 201. Вид сцены

VII. Этап рефлексии деятельности на уроке — 2 мин.

Деятельность учителя: предлагает учащимся провести рефлексия, ответив на следующие вопросы:

- 1) Достигли ли вы цели, поставленной на уроке?
- 2) Какие новые знания вы получили?
- 3) Какие новые навыки работы в среде Scratch вы получили?

### Тема урока «Работа со списками»

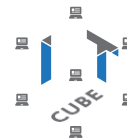
Тип урока: комбинированный.

Цель урока: ознакомление с созданием списка и его использованием в среде Scratch.

### Планируемые результаты:

Предметные: получение навыков по построению и выполнению программ по обработке списков в среде Scratch, работа с основными инструментами среды.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный



план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности; осознанный выбор будущей профессии на основе понимания её ценностного содержания и возможностей реализации собственных жизненных планов.

Время реализации: 1 академический час.

Оборудование и материалы:

Компьютер, проектор.

### *Ход урока*

I. Этап постановки цели и задач урока, мотивации к учебной деятельности — 5 мин.

Деятельность учителя: предлагает учащимся ознакомиться с темой урока — «Работа со списками». Он задаёт вопрос: что можно понимать под списком?

Деятельность учеников: делают свои предположения.

Учитель далее спрашивает: часто ли встречаются списки в жизни?

Ученики отвечают, что да — список номеров в телефонной книге, список учеников в классе и т. д.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

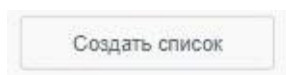
II. Этап актуализации знаний и пробного учебного действия — 8 мин.

Деятельность учителя: предлагает учащимся вспомнить, какими инструментами среды Scratch они пользовались при работе с переменными.

Деятельность учеников: отвечают на вопросы учителя.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

Учитель обращает внимание учеников, что также в разделе «Переменные» можно найти инструмент «Создать список» (рис. 202).



**Рис. 202.** Вид инструмента «Создать список»

III. Этап изучения нового материала — 16 мин.

Деятельность учителя: описывает основные инструменты среды Scratch по работе со списками, которые описаны в лабораторной работе 6.

Далее переходит к решению задачи. Предлагает учащимся выполнить следующую задачу.

Задача 1. За неделю были собраны данные о среднесуточной температуре. Представить их в виде списка в среде Scratch.

Деятельность учеников: описывают данный список, называют его  $t$  (рис. 203).

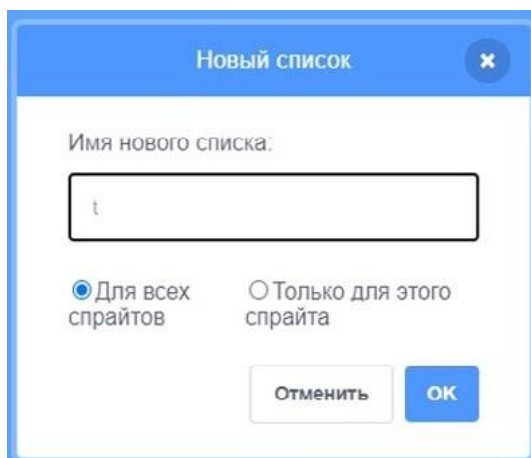


Рис. 203. Создание списка  $t$

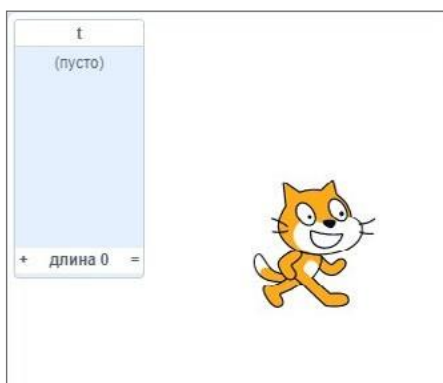


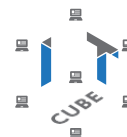
Рис. 204. Вид пустого списка на экране

Далее заполняют список данными вручную (рис. 205).



Рис. 205. Вид заполненного списка

Далее учитель предлагает подсчитать количество дней, когда температура была выше 15 градусов.



Что для этого потребуется? Описать переменную-счётчик (рис. 206), которую будем увеличивать при выполнении условия элемент\_списка >15.



Рис. 206. Описание новой переменной

Учитель спрашивает, какие блоки для этого предполагается использовать.

Деятельность учеников: отвечают на вопросы учителя, описывают переменную.

Во время решения задачи учащиеся сталкиваются с проблемой, что необходима ещё одна переменная — для обращения к номеру элемента списка. Её также необходимо увеличивать в теле цикла.

Вид программы представлен ниже (рис. 207).

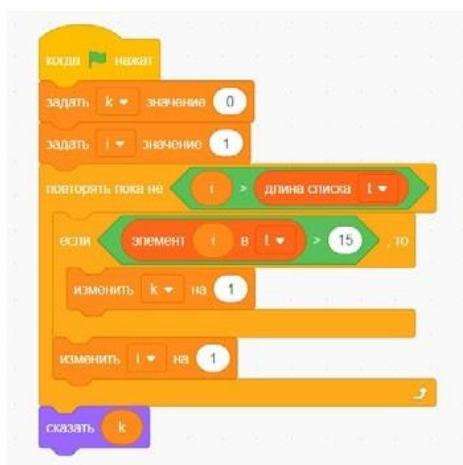


Рис. 207. Вид программы

Результат работы программы представлен на рисунке 208.

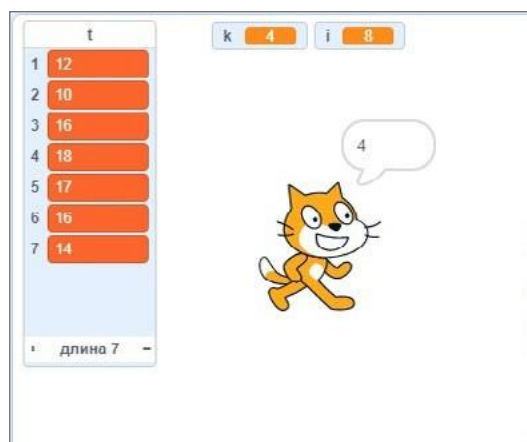


Рис. 208. Результат работы программы

IV. Этап проверки понимания и первичного закрепления — 8 мин.

Деятельность учителя: предлагает учащимся для решения следующую задачу.

Задача 2. Для данных из задачи 1 вывести на экран номера дней недели, когда температура была выше 15 градусов.

Деятельность учеников: выполняют задание учителя.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

V. Этап контроля усвоения и коррекции ошибок — 4 мин.

Деятельность учителя: предлагает учащимся ответить на следующие вопросы:

- 1) Что называется списком в среде Scratch?
- 2) Какие можно привести примеры данных, которые можно хранить в виде списка?
- 3) Какие основные инструменты вы использовали при работе со списками во время урока?

Деятельность учеников: отвечают на вопросы учителя.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

VI. Информация о домашнем задании, инструктаж по его выполнению — 2 мин.

Деятельность учителя: предлагает учащимся выполнить дома решение следующей задачи: «Для данных из задачи 1 определить среднюю температуру за неделю».

VII. Этап рефлексии деятельности на уроке — 2 мин.

Деятельность учителя: предлагает учащимся провести рефлексию, ответив на следующие вопросы:

- 1) Достигли ли вы цели, поставленные на уроке?
- 2) Какие новые знания вы получили?
- 3) Какие новые навыки работы в среде Scratch вы получили?

### Тема урока «Создание подпрограмм»

Тип урока: комбинированный.

#### Планируемые результаты:

Предметные: получение навыков по построению и выполнению подпрограмм в среде Scratch, работа с основными инструментами среды.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

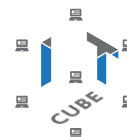
Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности; осознанный выбор будущей профессии на основе понимания её ценностного содержания и возможностей реализации собственных жизненных планов.

Время реализации: 1 академический час.

Оборудование и материалы:

Компьютер, проектор.

Цель урока: ознакомление с основами работы по созданию блоков-подпрограмм в среде Scratch.



### Ход урока

I. Этап постановки цели и задач урока, мотивации к учебной деятельности — 5 мин.

Деятельность учителя: предлагает учащимся ответить на следующий вопрос: «Часто ли при работе в среде Scratch приходится многократно выполнять одну и ту же группу действий?»

Деятельность учеников: отвечают утвердительно.

Учитель предлагает привести примеры таких программ. Учащиеся могут привести пример с изображением олимпийской символики. Учитель предлагает реализовывать подобные задачи с помощью блоков-подпрограмм, которые объединяют в себе группу многократно повторяемых блоков.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

II. Этап актуализации знаний и пробного учебного действия — 8 мин.

Деятельность учителя: предлагает учащимся вспомнить, какими инструментами среды Scratch они пользовались при необходимости строить изображения на сцене, как изменять цвет пера, его толщину. Для этого предлагается ответить на небольшой тест, вопросы которого представлены ниже.

Тест

1. Чему равна ширина сцены?

А) 320 точек

Б) 480 точек

В) 260 точек

Г) может меняться

2. Есть ли в Scratch графический редактор?

А) да

Б) нет

3. Для того чтобы добавить инструмент «Перо» необходимо:

А) выбрать данный блок в разделе «Рисование»

Б) добавить раздел «Перо» из расширений среды Scratch

В) загрузить данный блок из файла

Г) данного инструмента нет в среде Scratch

4. Сколько параметров можно изменить в команде «Изменить ... на»?

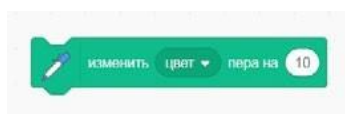


Рис. 209. Иллюстрация к вопросу 4 теста

А) 1

Б) 2

В) 3

Г) 4

Деятельность учеников: отвечают на вопросы учителя.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

III. Этап изучения нового материала — 16 мин.

Деятельность учителя: предлагает учащимся рассмотреть несколько изображений цветов.



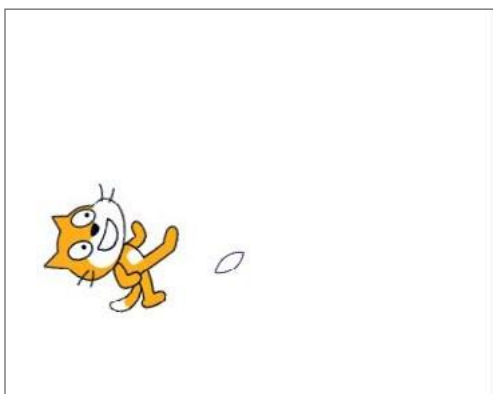
**Рис. 210.** Изображение цветов

Можно обратить внимание, что в изображении цветов часто встречаются повторяющиеся фрагменты, например лепестки.

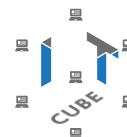
Поэтому предлагается построить программу по изображению цветка. Сначала разработаем спрайт, который строит один лепесток цветка. Пример такого скрипта представлен на рисунке 211.



**Рис. 211.** Вид программы для изображения лепестка



**Рис. 212.** Результат работы программы



Учащиеся могут предложить свои варианты изображения лепестка, изменив цвет, размер и т. д.

Если необходимо изобразить цветок из нескольких лепестков, то данный скрипт придётся повторять многократно. Поэтому целесообразно данный скрипт оформить в виде подпрограммы.

Далее переходим в раздел «Другие блоки», создаём новый блок «Leaf», у нового блока нет параметров. Чтобы сделать подпрограмму более общей, предлагается убрать у блока работу с пером.

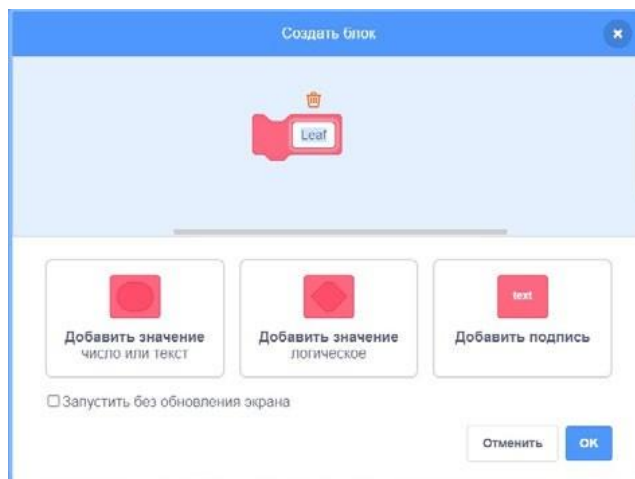


Рис. 213. Создание нового блока



Рис. 214. Вид нового блока

Теперь можно использовать эту подпрограмму как отправную точку. С её помощью можно рисовать более сложный цветок, состоящий, например, из пяти лепестков. Скрипт будет иметь следующий вид (рис. 215).

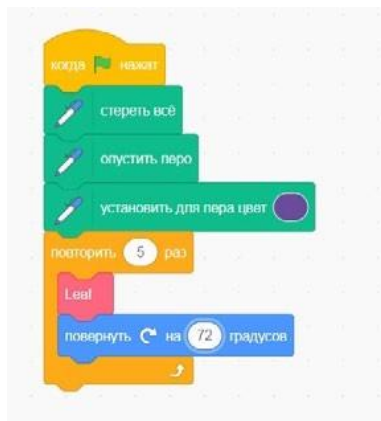


Рис. 215. Вид программы для изображения цветка

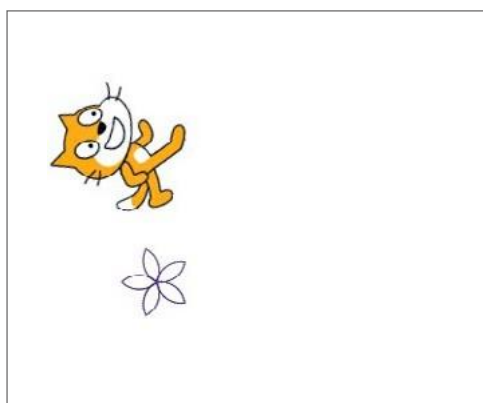
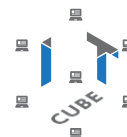


Рис. 216. Результат работы программы

Учащиеся могут изменять в подпрограмме размер лепестка, цвет цветка и получать новые изображения (рис. 217).



Рис. 217. Вид программы для изображения цветка



Для того чтобы изобразить несколько цветков, можно предложить создать ещё один новый блок «Leafs», в составе которого будет предыдущий блок «Leaf».



Рис. 218. Вид новых блоков для изображения лепестков цветка

Используем новую подпрограмму «Leafs» для изображения (рис. 219).



Рис. 219. Вид программы для изображения цветка

Далее можно использовать данные блоки «Leaf» и «Leafs» для создания более сложных узоров с лепестками. Например, изобразить два цветка разных цветов (рис. 220).

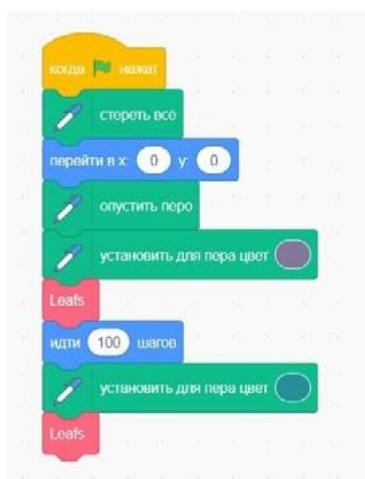


Рис. 220. Вид программы

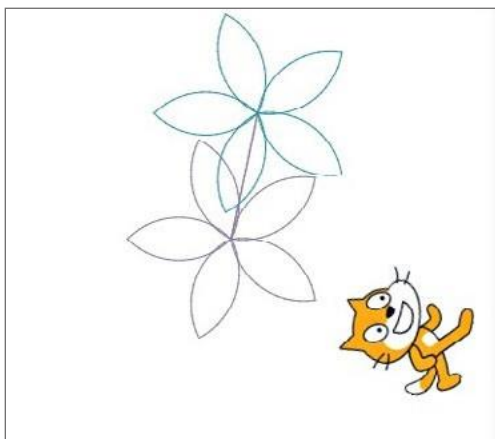


Рис. 221. Результат работы программы

Также можно использовать и другие подпрограммы для изображений элементов цветка, например его стебля (рис. 222). Размер стебля учащиеся могут изменять сами, тогда у каждого будут изначально создаваться индивидуальные изображения.

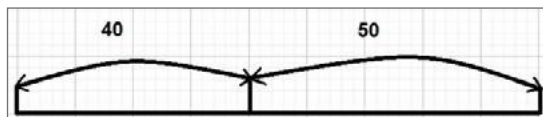


Рис. 222. Примерные размеры лепестка

В результате новый блок может иметь следующий вид (рис. 223).

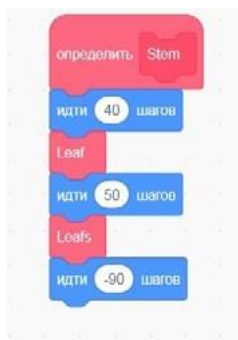


Рис. 223. Вид нового блока

Рассмотрим работу скрипта с новой подпрограммой (рис. 224).



Рис. 224. Вид программы

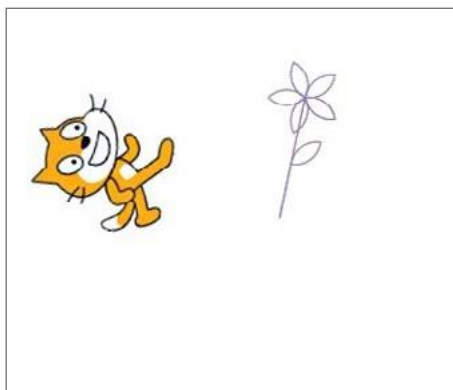


Рис. 225. Результат работы программы

IV. Этап проверки понимания и первичного закрепления — 8 мин.

Деятельность учителя: предлагает учащимся написать скрипт, который создаёт сложные цветочные узоры, используя созданные подпрограммы «Leaf», «Leafs», «Stem». Примером такого узора может служить рисунок 226.

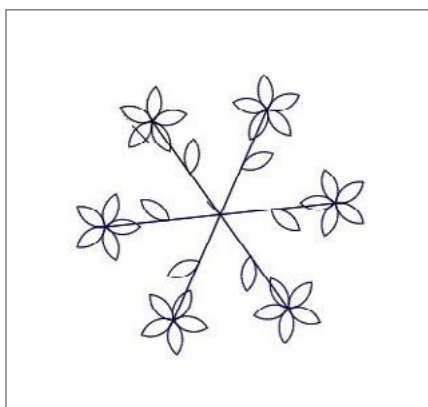


Рис. 226. Примерный вид узора

V. Этап контроля усвоения и коррекции ошибок — 4 мин.

Деятельность учителя: предлагает учащимся ответить на следующие вопросы:

- 1) Для чего используются подпрограммы в среде Scratch?
- 2) Как создать новый блок в среде Scratch?
- 3) Какие основные инструменты вы использовали при работе с подпрограммами во время урока?

Деятельность учеников: отвечают на вопросы учителя.

Дидактические и методические материалы: раздел «Описание разделов с командами» дидактических материалов пособия.

VI. Информация о домашнем задании, инструктаж по его выполнению — 2 мин.

Деятельность учителя: предлагает учащимся дома выполнить решение следующей задачи: разработайте подпрограмму для изображения снежинки. Пример снежинки представлен на рисунке 227.

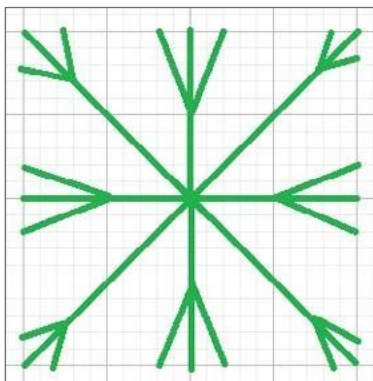


Рис. 227. Вид рисунка для домашнего задания

VII. Этап рефлексии деятельности на уроке — 2 мин.

Деятельность учителя: предлагает учащимся провести рефлексия, ответив на следующие вопросы:

- 1) Достигли ли вы цели, поставленной на уроке?
- 2) Какие новые знания вы получили?
- 3) Какие новые навыки работы в среде Scratch вы получили?

### Форма аттестации, примеры контрольно-оценочных материалов

Во время проведения курса предполагается текущий, промежуточный и итоговый контроль.

Текущий контроль осуществляется регулярно во время проведения каждого лабораторного занятия. Он заключается в ответе учащихся на контрольные вопросы, демонстрации полученных скриптов в среде Scratch, фронтальных опросов учителем.

В тематическом планировании предполагается проведение двух промежуточных контрольных работ.

#### Контрольная работа по темам «Линейные алгоритмы», «Условные алгоритмы»

1. Написать в среде Scratch следующую программу: *Пройти 200 шагов, повернуть на 90 градусов по часовой стрелке, пройти ещё 100 шагов.*
2. Написать в среде Scratch следующую программу: *Пройти 100 шагов, повернуть против часовой стрелки на 90 градусов, пройти 50 шагов.*
3. Написать в среде Scratch программу, изображающую следующую фигуру (рис. 228).

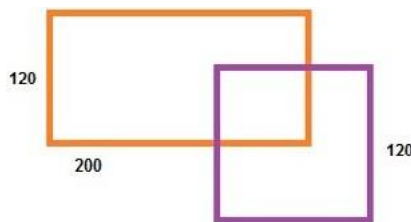
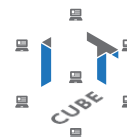


Рис. 228. Иллюстрация к задаче 3



4. Написать в среде Scratch программу, изображающую следующую фигуру (рис. 229).

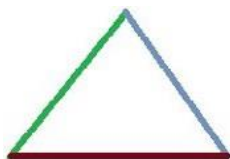


Рис. 229. Иллюстрация к задаче 4

5. Написать в среде Scratch программу, изображающую символику «Олимпийские кольца» (рис. 230).



Рис. 230. Иллюстрация к задаче 5

6. Написать в среде Scratch следующую программу: Для введенных с клавиатуры чисел  $x$  и  $y$  вычислить значение выражения  $x^2 + y^2$ .
7. Написать в среде Scratch следующую программу: Для введенных с клавиатуры чисел  $a$  и  $b$  выяснить, делится ли  $a$  на  $b$ .
8. Написать в среде Scratch следующую программу: Пользователь вводит целое число. Программа должна ответить, четным или нечетным является это число, делится ли оно на 3; делится ли оно на 6.
9. Написать в среде Scratch следующую программу: Пользователь вводит порядковый номер пальца руки (начиная с мизинца). Необходимо показать его название на экране.
10. Написать в среде Scratch следующую программу: Пользователь вводит пароль. По данному паролю определите степени доступа:  $[0, 1000]$  – доступен модуль А,  $[1001, 2500]$  или  $[3000, 5000]$  – доступны модули В и С,  $[9400, 10000]$  или  $[10500, 50000]$  – доступен модуль D. Если значение не попало ни в один из указанных отрезков, то в доступе отказано!

### Контрольная работа по темам «Циклические алгоритмы», «Работа со списками»

1. Написать в среде Scratch программу, изображающую следующую фигуру (рис. 231).

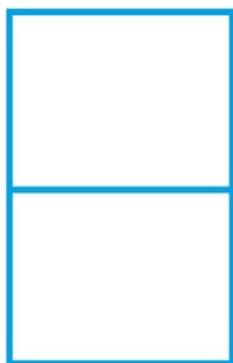


Рис. 231. Иллюстрация к задаче 1

2. Написать в среде Scratch программу, изображающую следующую фигуру (рис. 232).

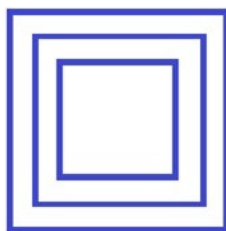


Рис. 232. Иллюстрация к задаче 2

3. Написать в среде Scratch программу, изображающую рисунок 233.

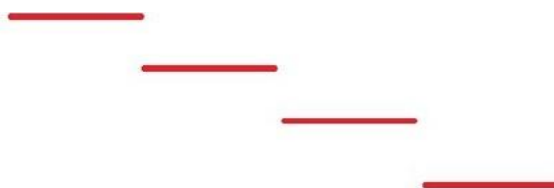
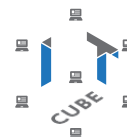


Рис. 233. Иллюстрация к задаче 3

4. Написать в среде Scratch программу: *Вывести на экран первые десять степеней двойки.*
5. Написать в среде Scratch программу: *Найти наибольший общий делитель двух чисел, введенных пользователем.*
6. Написать в среде Scratch программу: *В списке хранятся данные о температуре в городке N за 12 месяцев. Выведите температуру с марта по сентябрь.*
7. Написать в среде Scratch программу: *В списке хранятся данные о температуре в городке N за 12 месяцев. Выведите максимальную температуру за год.*
8. Написать в среде Scratch программу: *В списке хранится информация об оценках за четверть по информатике класса из 20 человек. Определите, сколько человек получили «пятёрки» за четверть.*
9. Написать в среде Scratch программу: *В списке хранится информация об оценках за четверть по информатике класса из 20 человек. Определите, какой процент хорошистов в классе.*
10. Написать в среде Scratch программу: *В списке хранится информация об оценках за четверть по информатике класса из 20 человек. Определите средний балл в данном классе.*

Также предполагается итоговая аттестация в форме разработки и защиты проекта.



## Материалы для организации и проведения учебно-исследовательской и проектной деятельности школьников

Создание в среде Scratch проекта, который предполагает использование анимации, музыкального сопровождения, интерактива, требует осознанной исследовательской деятельности с начальным чётким целеполаганием, построением плана разработки проекта, сверки результатов и т. д.

Если же речь идёт о создании сложного проекта, в котором возможно будут использоваться несколько спрайтов, смены сцен, то проект может быть разбит на подзадачи, под-проекты. Каждую подзадачу будут выполнять различные группы участников проекта.

Такая работа изначально требует от учащихся совместной разработки плана проекта. План проекта может состоять из следующих этапов, представленных ниже.

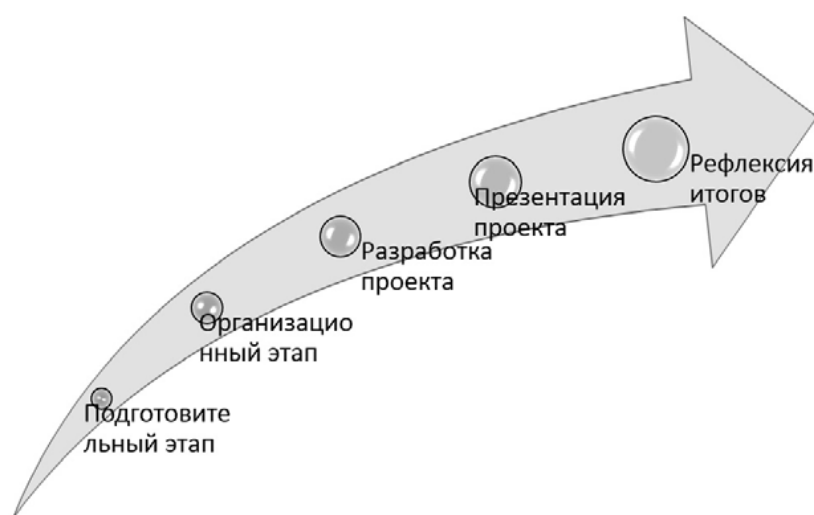


Рис. 234. Этапы работы над проектом

На подготовительном этапе ставятся цель и задачи проекта, составляется план достижения цели, определяются объекты, которые будут исследоваться в проекте, их взаимосвязи. При необходимости проект разбивается на подпроекты, тогда определяется последовательность их выполнения.

На организационном этапе распределяются обязанности участников проекта, намечаются сроки выполнения проекта.

В помощь участникам проекта можно предложить заполнить следующий учётный лист.

### Исследовательский проект

Тема проекта:

Творческое название (при наличии):

Основополагающий вопрос:

Авторы:

- 1.
- 2.
- 3.
- ...

*Предметная область:*

*Краткая аннотация:*

*Проблемные вопросы учебной темы:*

*Темы исследования учащихся:*

*Этапы выполнения проекта:*

На этапе разработки участниками создаётся наполнение проекта, представление спрайтов, разрабатываются скрипты. Происходит отладка действий основных персонажей.

На этапе презентации участники представляют проект на обсуждение.

Этап рефлексии отводится под обсуждение итогов проекта, оценки своих действий, формулирование выводов.

Для оценивания проекта могут быть разработаны специальные оценочные листы.

Ниже представлен пример оценочного листа:

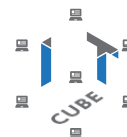
Таблица

**Лист оценивания проекта**

| Критерий оценивания                                                                     | 1-я группа | 2-я группа | ... |
|-----------------------------------------------------------------------------------------|------------|------------|-----|
| Актуальность темы                                                                       |            |            |     |
| Соответствие содержания проекта заявленной теме                                         |            |            |     |
| Техническая сложность                                                                   |            |            |     |
| Оригинальность                                                                          |            |            |     |
| Дизайн                                                                                  |            |            |     |
| Наличие соответствующего музыкального сопровождения с указанием в титрах авторов музыки |            |            |     |
| Уровень проработанности проекта                                                         |            |            |     |
| Возможность применения проекта в школе                                                  |            |            |     |
| Итоговое количество баллов                                                              |            |            |     |

Ниже приведены возможные темы исследовательских проектов учащихся:

1. Игра на различных музыкальных инструментах: имитация игры мелодий на различных музыкальных инструментах.
2. Игра «Приключения героя в стране математики»: изучение обыкновенных дробей, арифметических действий над обыкновенными дробями.
3. Игра «Приключение в стране геометрии»: изучение и рисование различных геометрических фигур.
4. Игра с элементами сказки «Репка», «Колобок».
5. Весёлый тест по информатике.
6. Моделирование физических процессов.
7. Разработка различных игр.
8. Работа с системами счисления.



Ниже приведён пример проекта на тему «Тест по математике».

В качестве спрайта можно создать персонаж Гарри Поттер, нарисовав его в графическом редакторе (рис. 235).



Рис. 235. Создание спрайта Гарри Поттер

Также подбирается сцена по сюжету (рис. 236).

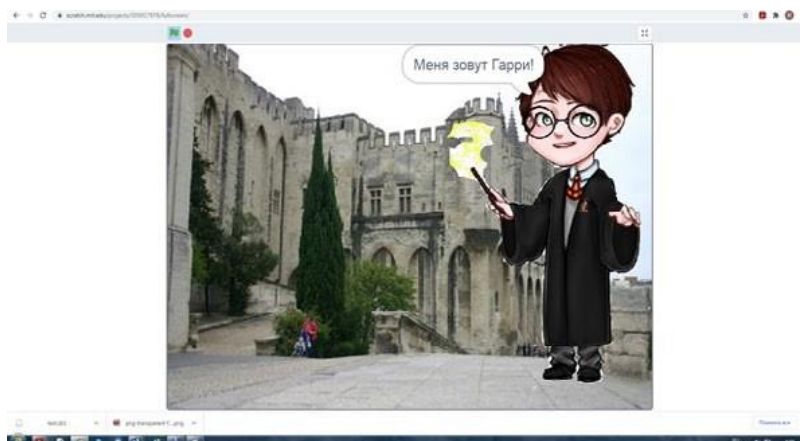


Рис. 236. Создание сцены

Для создания теста можно использовать переменную  $s$  в качестве счётчика правильных ответов. Также можно создать переменную  $x$ , для запоминания ответа тестируемого.

Примерный фрагмент скрипта игры представлен на рисунке 237.

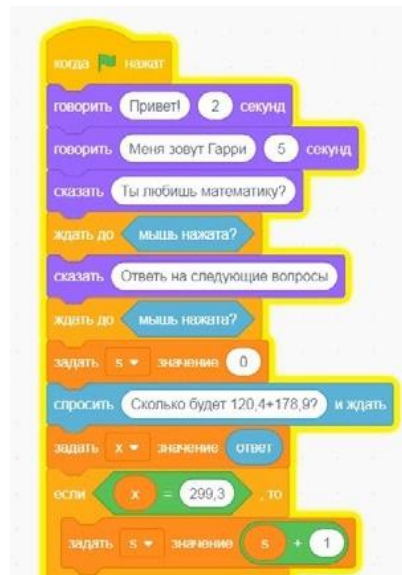


Рис. 237. Фрагмент программы в среде Scratch

Ниже приведён фрагмент ответов на тест (рис. 238).



Рис. 238. Фрагмент ответов на тест

Физические задачи дают богатую почву для разработки различного вида проектов. Приведём пример проекта «Полёт ядра». Данный проект имеет межпредметную связь с физикой. Для его выполнения необходимо изучить физический процесс: надо определить траекторию полёта тела, брошенного под определённым углом к горизонту. Задача состоит в следующем.

Рассмотрим полёт ядра, которым выстрелила пушка, направленная под углом  $q$  к земле, и ядро вылетает со скоростью  $v_0$ .

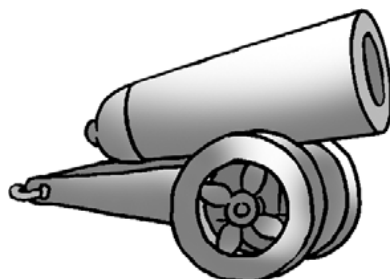


Рис. 239. Иллюстрация к задаче

Необходимо проанализировать траекторию полёта. Для этого разложим вектор скорости  $v_0$  на горизонтальную и вертикальную составляющие.

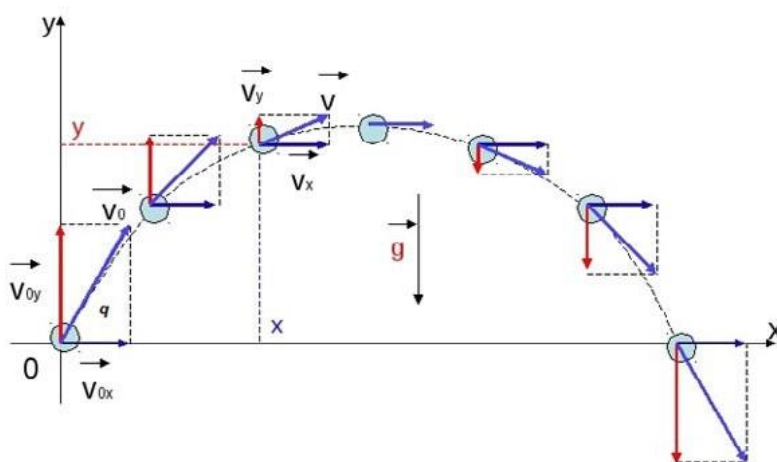


Рис. 240. Иллюстрация к задаче

Горизонтальный компонент остаётся неизменным, а на вертикальный компонент действует сила притяжения. Когда движения, связанные с этими двумя компонентами, объединяются, они образуют путь в виде параболы.

Далее составим уравнение движения ядра без учёта сопротивления воздуха. Как видно из рисунка, начало координат определено в точке, от которой начинается полёт ядра. Напишем уравнение для изменения координаты  $x$ :

$$x(t) = v_{0x}t.$$

Уравнение для координаты  $y$  имеет вид:

$$y(t) = v_{0y}t - \frac{gt^2}{2}$$

$$\text{где } v_{0x} = v_0 \cos(q), v_{0y} = v_0 \sin(q), \quad (1)$$

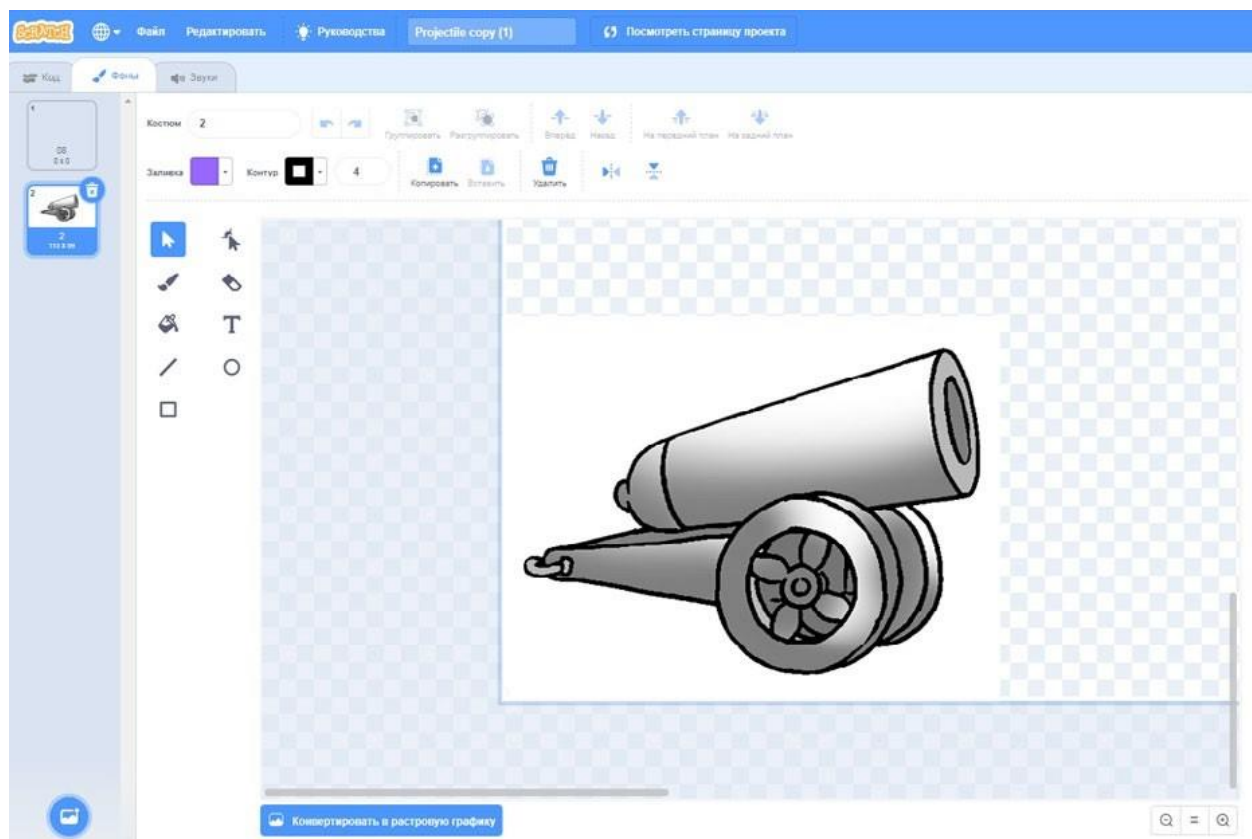
а  $g = 9,8 \text{ м/с}^2$  — ускорение свободного падения.

Используя эти уравнения, можно вычислить общее время полета, максимальную высоту и расстояние, на которое ядро перемещается по горизонтали. Соответствующие уравнения примут вид:

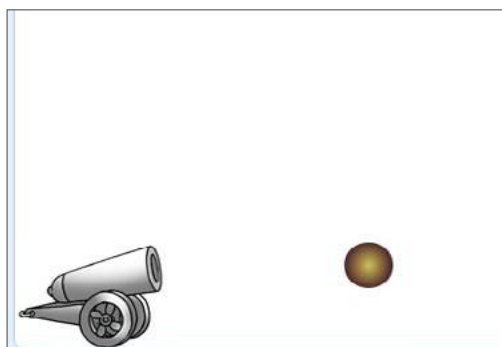
$$h \approx \frac{(v_0 \sin q)^2}{2g}, d \approx \frac{v_0^2 \sin 2q}{g}, t \approx \frac{2v_0 \sin q}{g}. \quad (2)$$

Теперь полученные формулы необходимо реализовать в виде программы в среде Scratch.

На сцену добавим фон с иллюстрацией стреляющей пушки (рис. 241).



**Рис. 241.** Создание фона в графическом редакторе



**Рис. 242.** Вид сцены с созданным фоном

В проекте будет использовано несколько спрайтов: ядро, кнопка запуска. Создадим в редакторе спрайт Ядро, для него создаётся основной код в программе.

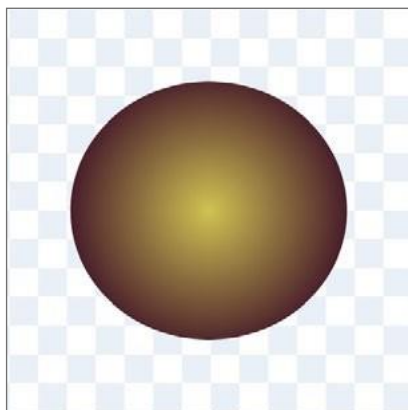


Рис. 243. Вид спрайта Ядро

Разработаем скрипт для реализации полёта ядра. Нам потребуется несколько переменных, которые будут использоваться в формулах (2):

- $q$  — начальный угол;
  - $d$  — расстояние, которое пролетело ядро по горизонтали;
  - $dy$  — местоположение спрайта по вертикали;
  - скорость — скорость ядра;
  - $t$  — время полёта;
  - $temp$  — приращение;
  - $vx$ ,  $vy$  — составляющие скорости;
  - $xpos$ ,  $ypos$  — текущее положение спрайта.
- Опишем эти переменные в соответствующем разделе (рис. 244).

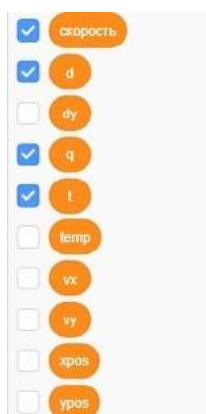


Рис. 244. Вид описанных переменных

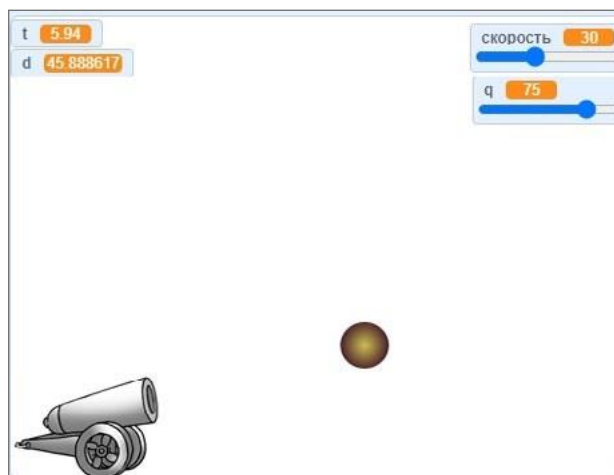


Рис. 245. Вид сцены с переменными

Как видно на рисунке, двум переменным, отвечающим за значение угла и скорости, задан вид «Рычажок».

Для этого введём новый блок — «Инициализация», в котором будут задаваться начальные значения для подготовки выстрела (рис. 246).

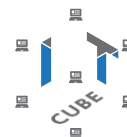


Рис. 246. Вид блока «Инициализация»

В данном скрипте используются формулы (1) для задания вертикальной и горизонтальной составляющих скорости. В начале блока задаётся начальное положение ядра, которое определяется таким образом, чтобы имитировать вылет из дула пушки. В зависимости от изображения пушки они могут быть изменены. Цвет пера также может быть изменён.



Рис. 247. Задание начального положения спрайта Ядро



Далее перейдём к программированию основного скрипта (рис. 248).



Рис. 248. Вид программы

Разберём данный скрипт более подробно. В его основе — бесконечный цикл, который заканчивается, когда спрайт Ядро достигнет земли, т. е. переменная  $dy$  (местоположение спрайта по вертикали) достигает отрицательного значения. Можно обратить внимание, что по вертикали спрайт Ядро возвращается в начальное положение. В противном случае вычисляется положение по горизонтали  $d$  (по формуле (2)). Далее рассчитываются значения переменных  $xpos$  и  $ypos$ . Значения в формулах получаются следующим образом: по вертикали расстояние составляет 300 точек, от  $-120$  до  $180$ , а по горизонтали — соответственно от  $-130$  до  $240$ , т. е. 370 точек. Эти величины должны соответствовать 100 м. Поэтому ведётся пересчёт (масштабирование) значений  $xpos$  и  $ypos$ .

Затем с помощью блока «Перейти в ...» спрайт перемещается в новую позицию. Приращение времени  $t$  выбрано 0,02. При увеличении значения  $t$  спрайт ускоряется.

Результат работы программы при начальных значениях скорости 60 и угла  $q$  в 60° представлен ниже на рисунке 249. Как можно видеть, ядро улетело практически на 80 м.

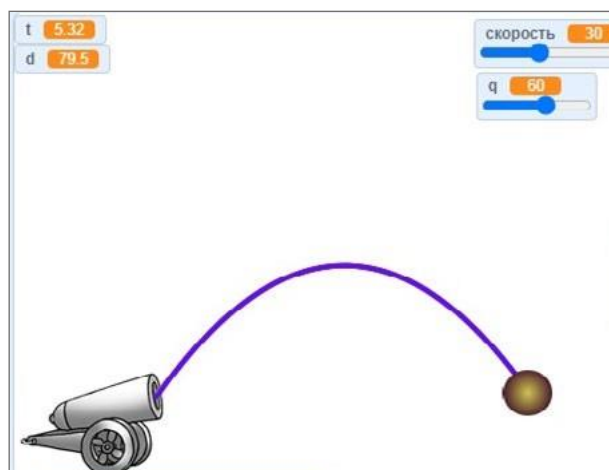


Рис. 249. Результат работы программы

Изменим начальные значения скорости и угла на 30 и 75°. Вид траектории полета ядра представлен ниже.

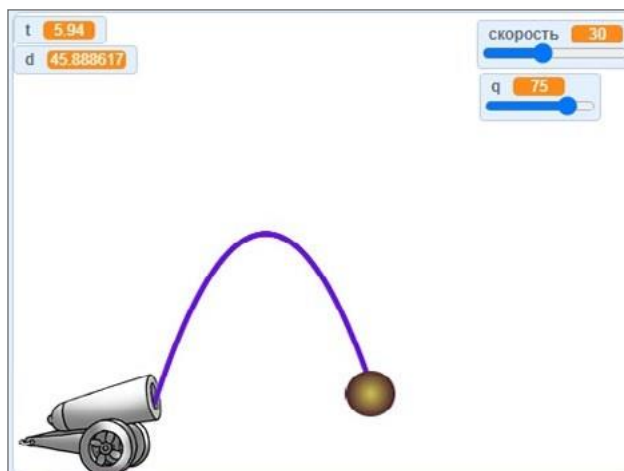


Рис. 250. Результат работы программы

В первой программе движение ядра начинается при нажатии на зелёный флажок. Условим проект, добавим спрайт кнопка Огонь.

Сначала нарисуем эту кнопку в графическом редакторе (рис. 251).



Рис. 251. Вид спрайта Кнопка Огонь

Поместим новый спрайт на сцене (рис. 252).

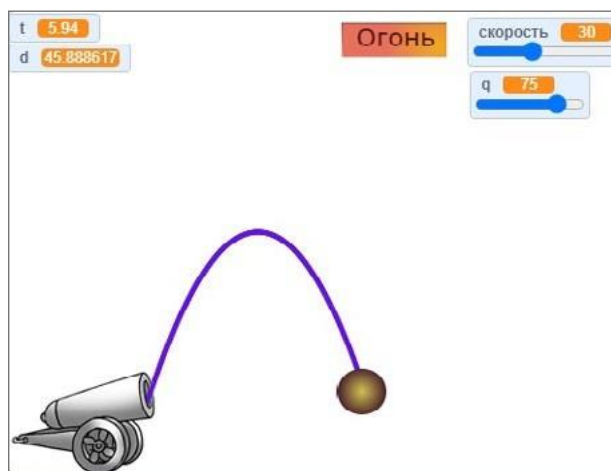


Рис. 252. Вид сцены с новым спрайтом

Далее напишем скрипт для нового спрайта Кнопка Огонь. Он представлен на рисунке 253. Для чего необходимо изменение положения спрайта? Это сделано для имитации нажатия клавиши — небольшое движение вниз с возвратом на прежнюю позицию.

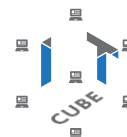


Рис. 253. Вид программы для нового спрайта

Для передачи управления спрайту Ядро используем блок «передать...», в котором зададим сообщение «огонь». Затем необходимо изменить немного скрипт для спрайта Ядро. Приведём на рисунке изменённое начало скрипта.



Рис. 254. Вид начала скрипта для спрайта Ядро

В начале скрипта изменили блок «Когда флажок нажат» на «Когда я получу...». В результате ядро «вылетает» из пушки по нажатию кнопки «Огонь».

Проект может иметь продолжение: например, задать поворот дула пушки и т. д.

## Перечень доступных источников информации

1. Босова Л. Л. Информатика. 8 класс : учебник. / Босова Л. Л. — М.: БИНОМ. Лаборатория знаний, 2016. — 176 с.
2. Винницкий Ю. А. Scratch и Arduino для юных программистов и конструкторов. / Винницкий Ю. А. — СПб.: БХВ-Петербург, 2018. — 176 с.
3. Голиков Д. В. Scratch для юных программистов. / Голиков Д. В. — СПб.: БХВ-Петербург, 2017. — 192 с.
4. Лаборатория юного линуксоида. Введение в Scratch. — <http://younglinux.info/scratch>
5. Маржи М. Scratch для детей. Самоучитель по программированию. / Маржи М. — пер. с англ. М. Гескиной и С. Таскаевой. — М.: Манн, Иванов и Фербер, 2017. — 288 с.
6. Пашковская Ю. В. Творческие задания в среде Scratch: рабочая тетрадь для 5—6 классов. / Пашковская Ю. В. — М., 2018. — 195 с.
7. Первин Ю. А. Методика раннего обучения информатике. / Первин Ю. А. — М.: БИНОМ. Лаборатория знаний, 2008. — 228 с.
8. Поляков К. Ю. Информатика. 7 класс (в 2 частях) : учебник. Ч. 1 / Поляков К. Ю., Еремин Е. А. — М.: БИНОМ. Лаборатория знаний, 2019. — 160 с.
9. Рындак В. Г., Дженжер В. О., Денисова Л. В. Проектная деятельность школьника в среде программирования Scratch: учебно-методическое пособие. / Рындак В. Г., Дженжер В. О., Денисова Л. В. — Оренбург: Оренб. гос. ин-т. менеджмента, 2009. — 116 с.
10. Свейгарт Эл. Программирование для детей. Делай игры и учи язык Scratch! / Свейгарт Эл. — М.: Эксмо, 2017. — 304 с.
11. Семакин, И. Г. Информатика и ИКТ: учебник для 9 класса. / Семакин, И. Г., Залогова, Л. А. и др. М: БИНОМ. Лаборатория знаний, 2014. — 171 с.
12. Торгашева Ю. В. Первая книга юного программиста. Учимся писать программы на Scratch. / Торгашева Ю. В. — СПб.: Питер, 2016. — 128 с.
13. Уфимцева П. Е. Обучение программированию младших школьников в системе дополнительного образования с использованием среды разработки Scratch / Уфимцева П. Е., Рожина И. В. // Наука и перспективы. — 2018. — № 1. — С. 29—35.
14. <https://scratch.mit.edu/> Сообщество Scratch.



Григорьев Сергей Георгиевич  
Родионов Михаил Алексеевич  
Акимова Ирина Викторовна

**Реализация дополнительной общеобразовательной программы  
по тематическому направлению «Основы алгоритмики и логики»  
с использованием оборудования центра цифрового образования  
детей «IT-Куб»**

*Методическое пособие*

Под ред. Григорьева С. Г.

Центр Естественно-научного и математического образования  
Руководитель Центра *З. Г. Гапонюк*  
Ответственный за выпуск *М. В. Кузнецова*  
Редактор *М. В. Кузнецова*  
Художественное оформление  
Компьютерная вёрстка и техническое редактирование *Э. В. Алексеев*  
Корректор *Н. А. Смирнова*