

Муниципальное казенное общеобразовательное учреждение
«Средняя общеобразовательная школа № 8» с. Манычское
Центр цифрового и гуманитарного профилей «Точка роста»

СОГЛАСОВАНО:

Руководитель Центра образования
цифрового и гуманитарного профилей
«Точка Роста»

 А.А. Каплунова

УТВЕРЖДАЮ:

Директор МКОУ СОШ №8 с.Манычское

МКОУ А.В. Лазырин

приказ № 109 от 30.08.2024г.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ
ПРОГРАММА
общеразвивающая
технической направленности
«Программирование в Scratch»

Уровень программы: базовый

(ознакомительный, базовый, углубленный)

Возрастная категория: от 10 до 15 лет

Состав группы: 10-15 человек

(количество учащихся)

Срок реализации: 2 год(а)

Автор-составитель:

Багамагазиева Ольга Васильевна

Учитель по профилю «Информатика»

с. Манычское, 2024

НОРМАТИВНО-ПРАВОВЫЕ ОСНОВАНИЯ ДЛЯ ПРОЕКТИРОВАНИЯ ДОПОЛНИТЕЛЬНЫХ ОБЩЕРАЗВИВАЮЩИХ ПРОГРАММ

1. Федеральный Закон от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации» (далее - ФЗ);
2. Федеральный закон Российской Федерации от 14.07. 2022 г. № 295-ФЗ «О внесении изменений в Федеральный закон «Об образовании в Российской Федерации»;
3. Федеральный закон РФ от 24.07.1998 г. № 124-ФЗ «Об основных гарантиях прав ребенка в Российской Федерации» (в редакции 2013 г.);
4. Концепция развития дополнительного образования детей до 2030 года, утвержденной распоряжением Правительства Российской Федерации от 31 марта 2022 г. № 678-р;
5. Стратегия развития воспитания в РФ на период до 2025 года (распоряжение Правительства РФ от 29 мая 2015 г. № 996-р);
6. Постановление Правительства Российской Федерации от 11.10.2023 г. № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
7. Постановление Главного государственного санитарного врача РФ от 28 сентября 2020 г. № 28 "Об утверждении санитарных правил СП 2.4.3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи" (далее - СанПиН);
8. Постановление Главного государственного санитарного врача РФ от 28 января 2021 г. № 2 "Об утверждении санитарных правил и норм [СанПиН 1.2.3685-21 "Гигиенические нормативы и требования к обеспечению безопасности и \(или\) безвредности для человека факторов среды обитания"](#)";
9. Приказ Министерства труда и социальной защиты Российской Федерации от 05.05.2018 г. № 298 "Об утверждении профессионального стандарта "Педагог дополнительного образования детей и взрослых";
10. Приказ Министерства просвещения Российской Федерации от 27 июля 2022 г. N 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (далее - Порядок);
11. Приказ Министерства просвещения Российской Федерации от 03.09.2019 г. № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»;
12. Приказ Министерства науки и высшего образования РФ и Министерства просвещения РФ от 5 августа 2020 г. № 882/391 «Об

деятельности при сетевой форме реализации образовательных программ»;

13. Письмо Минобрнауки России № 09-3242 от 18.11.2015 г. «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»;

14. Письмо Минобрнауки России от 28.08.2015 г. № АК-2563/05 "О методических рекомендациях" (вместе с "Методическими рекомендациями по организации образовательной деятельности с использованием сетевых форм реализации образовательных программ";

15. Письмо Минобрнауки России от 29.03.2016 г. № ВК-641/09 «О направлении методических рекомендаций» (вместе с «Методическими рекомендациями по реализации адаптированных дополнительных общеобразовательных программ, способствующих социально-психологической реабилитации, профессиональному самоопределению детей с ограниченными возможностями здоровья, включая детей-инвалидов, с учетом их особых образовательных потребностей»).

Пояснительная записка

Рабочая программа по внеурочной деятельности «Программирование в Scratch в роботехнике» разработана на основе:

- Федерального государственного стандарта основного общего образования по информатике;
- Базисного учебного плана;
- Методических рекомендаций Реализация образовательных программ по предмету "Информатика" с использованием оборудования центра «Точка роста».

На базе центра «Точка роста» обеспечивается реализация образовательных программ технологической направленности, разработанных в соответствии с требованиями законодательства в сфере образования и с учётом рекомендаций Федерального оператора учебного предмета «Информатика».

Образовательная программа позволяет интегрировать реализуемые подходы, структуру и содержание при организации обучения информатики в 5–9 классах, выстроенном на базе любого из доступных учебно-методических комплексов (УМК).

Использование оборудования «Точка роста» при реализации данной ОП позволяет создать условия:

- для расширения содержания школьного образования по информатике;
- для повышения познавательной активности обучающихся в технической области;
- для развития личности ребёнка в процессе обучения информатики, его способностей, формирования и удовлетворения социально значимых интересов и потребностей;
- для работы с одарёнными школьниками, организации их развития в различных областях образовательной, творческой деятельности.

Актуальность программы

Актуальность программы состоит в том, что мультимедийная среда Scratch позволяет сформировать у учащихся интерес к программированию, отвечает всем современным требованиям объектно-ориентированного программирования. Среда Scratch позволяет сформировать навыки программирования, раскрыть технологию программирования. Изучение языка значительно облегчает последующий переход к изучению других языков программирования. Преимуществом Scratch, среди подобных сред программирования, является наличие версий для различных операционных систем, к тому же программа является свободно распространяемой, что немало важно для образовательных учреждений.

Новизна заключается в том, что Scratch не просто язык программирования, а еще и интерактивная среда, где результаты действий визуализированы, что делает работу с программой понятной, интересной и увлекательной.

Особенность среды Scratch, позволяющая создавать в программе мультфильмы, анимацию и даже простейшие игры, делает образовательную программу «Увлекательное программирование» практически значимой для современного подростка, т.к. дает возможность увидеть практическое назначение алгоритмов и программ, что будет способствовать развитию интереса к профессиям, связанным с программированием

Цели курса:

Главной целью курса имеет развитие познавательных интересов в области информатики и формирование алгоритмического мышления через освоение принципов программирования в объектно-ориентированной среде.

Курс соответствует всем без исключения целям изучения информатики в основной школе, обозначенным во ФГОС:

- формирование информационной и алгоритмической культуры;
- развитие основных навыков и умений использования компьютерных устройств;
- формирование представления об алгоритмах, моделях и их свойствах;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе;
- развитие умений составлять и записывать алгоритм для конкретного исполнителя;
- формирование знаний об алгоритмических конструкциях, логических значениях и операциях;
- знакомство с языками программирования и основными алгоритмическими структурами — линейной, условной и циклической;
- формирование умений формализации и структурирования информации, умения выбирать способ представления данных;
- формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете, умения соблюдать нормы информационной этики и права.

Задачи программы:

- Информационная и медиа грамотность. Работая над проектами в Scratch, дети работают с разными видами информации: текст, изображения, анимация, звук, максимально проявляя свои творческие способности.

- Коммуникативные навыки. Эффективная коммуникация в современном мире требует больше, чем умение читать и писать текст. Работая в Scratch, дети собирают и обрабатывают информацию с различных источников. В результате они становятся более критичными в работе с информацией.
- Критическое и системное мышление. Работая в Scratch, дети учатся критически мыслить и рассуждать. В проектах необходимо согласовывать поведение агентов, их реакции на события.
- Постановка задач и поиск решения. Работа над проектами в Scratch требует умения ставить задачи, определять исходные данные и необходимые результаты, определять шаги для достижения цели.
- Творчество и любознательность. Scratch поощряет творческое мышление, он вовлекает детей в поиск новых решений известных задач и проблем.
- Межличностное взаимодействие и сотрудничество. Scratch позволяет ученикам работать над проектами совместно, ведь спрайты, коды можно легко и свободно экспортировать/импортировать.
- Самоопределение и саморазвитие. Scratch воспитывает в детях настойчивость в достижении целей, создает внутренние мотивы для преодоления проблем, ведь каждый проект в Scratch идет от самого ребенка.
- Ответственность и адаптивность. Создавая проект в Scratch, ребенок должен осознавать, что его увидят миллионы людей, и быть готовым изменить свой проект, учитывая реакцию сообщества.
- Социальная ответственность. Scratch-проекты позволяют поднять социально значимые вопросы, спровоцировать их обсуждение в молодежной среде.

Педагогическая целесообразность программы

Программа «Программирование в среде Scratch.» сочетает в себе различные формы проведения занятий: аудиторные – учебное занятие, практическая работа, создание и защита проекта. Такое сочетание форм позволяет, как качественно сформировать предметные навыки (работа в программе Scratch), так и поддерживать на высоком уровне познавательный интерес обучающихся, готовность к творческой деятельности. Самостоятельное планирование, организация и проведение исследований и обработка снятых материалов развивают навыки исследовательской деятельности и творческие способности обучающихся. Основа курса – проектная научно-познавательная деятельность школьников на занятиях. Именно в этой деятельности наиболее полно раскрывается личностный потенциал школьника. Развиваются ценные качества и умения, необходимые современному человеку: критическое, системное, алгоритмическое и творческое мышление; умение находить решение проблем; умение работать самостоятельно и в команде. Педагогическая целесообразность связана с реализацией следующих возможностей для развития ребенка: создание максимального количества ситуаций успеха;– возможность долговременного влияния на формирование личности– обучающегося, выявление и стимулирование проявлений положительных личностных качеств– ребенка, для постижения самооценности собственной личности; практическая значимость (расширение кругозора, использование приобретаемых– качеств, знаний в повседневной жизни), предоставление обучающемуся широких возможностей для самовыражения–

средствами программирования.

Место курса в учебном плане

Рабочая программа курса по информатике «Программирования «Scratch» рассчитана для внеурочной деятельности обучающихся 10-15 лет сроком на 1 года. Всего 114 часов, по 3 часа в неделю в каждом классе.

Ожидаемые результаты и способы определения их результативности

Обучающиеся, освоившие дополнительную общеобразовательную общеразвивающую программу «Программирование в среде Scratch» достигнут следующих результатов:

Будут знать:

- что такое программирование;
- что такое языки программирования;
- о необходимости составлять программы;
- синтаксис в языках программирования;
- способы создания мультфильмов;
- способы создания игр;
- алгоритм проектной деятельности;
- правила техники безопасности в компьютерном классе.

Будут уметь:

- выбирать и запускать программную среду Scratch;
- работать с основными элементами пользовательского интерфейса программной среды;
- создавать игры;
- создавать мультфильмы;
- использовать меню «быстрых» клавиш, кнопок в окнах диалога, шрифтов;
- сформулировать тематику проекта и выполнить проект.

Способы определения результативности

Для отслеживания результатов обучения по программе используется:

- метод педагогического наблюдения,
- беседа с обучающимися,
- педагогический анализ проводимых отчетных мероприятий.

Формы контроля

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

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

Также в тематическом планировании предполагается две промежуточные контрольные работы.

Требования к планируемым результатам изучения программы

В основном формируются и получают развитие метапредметные результаты, такие как:

- умение самостоятельно планировать пути достижения целей, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата;
- умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

Вместе с тем вносится существенный вклад в развитие личностных результатов, таких как:

- формирование ответственного отношения к учению;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, творческой и других видов деятельности.

В части развития предметных результатов наибольшее влияние изучение курса оказывает на:

- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей с использованием соответствующих программных средств обработки данных;
- формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете, умения соблюдать нормы информационной этики и права.

Регулятивные универсальные учебные действия

Обучающийся научится:

- целеполаганию, включая постановку новых целей, преобразование практической

задачи в познавательную;

- самостоятельно анализировать условия достижения цели на основе учета выделенных учителем ориентиров действия в новом учебном материале;
- планировать пути достижения целей;
- уметь самостоятельно контролировать свое время и управлять им.

Коммуникативные универсальные учебные действия

Обучающийся научится:

- устанавливать и сравнивать разные точки зрения, прежде чем принимать решения и делать выбор;
- аргументировать свою точку зрения, спорить и отстаивать свою позицию не враждебным для оппонентов образом;
- задавать вопросы, необходимые для организации собственной деятельности и сотрудничества с партнером;
- осуществлять взаимный контроль и оказывать в сотрудничестве необходимую взаимопомощь.

Познавательные универсальные учебные действия

Обучающийся научится:

- создавать и преобразовывать модели и схемы для решения задач;
- осуществлять выбор наиболее эффективных способов решения задач в зависимости от конкретных условий;
- давать определение понятиям;
- устанавливать причинно-следственные связи;
- осуществлять логическую операцию установления родовидовых отношений, ограничение понятия;
- обобщать понятия — осуществлять логическую операцию перехода от видовых признаков к родовому понятию, от понятия с меньшим объемом к понятию с большим объемом;
- строить логическое рассуждение, включающее установление причинно-следственных связей.

Полученные навыки работы в Scratch будут полезны в практической деятельности: помогут школьникам освоить азы алгоритмизации и программирования, будут применяться при создании и исследовании компьютерных моделей по школьным дисциплинам, помогут при изучении таких школьных дисциплин, как «Математика», «Музыка», «Изобразительное искусство», а также для более серьезного изучения программирования в старших классах.

Работа с Интернет-сообществом скретчеров позволит освоить навыки информационной деятельности в глобальной сети: размещение своих проектов на сайте, обмен идеями с пользователями интернет-сообщества, овладение культурой общения на форуме.

Формы и методы работы

На занятиях используются как классические для педагогики формы и методы работы, так и нетрадиционные.

Формы проведения занятий:

- урок с использованием игровых технологий; урок-игра;
- урок-исследование;
- творческие практикумы (сбор скриптов с нуля);
- урок-испытание игры;
- урок-презентация проектов;
- урок с использованием тренинговых технологий (работа на редактирование готового скрипта в соответствии с поставленной задачей).

Методы обучения:

- словесные методы (лекция, объяснение);
- демонстративно-наглядные (демонстрация работы в программе, схем, скриптов, таблиц);
- исследовательские методы;
- работа в парах;
- работа в малых группах;
- проектные методы (разработка проекта по спирали творчества, моделирование, планирование деятельности)
- работа с Интернет-сообществом (публикация проектов в Интернет-сообществе скретчеров).

Практическая часть работы – работа в среде программирования со скриптами и проектирование информационных продуктов. Для наилучшего усвоения материала практические задания рекомендуется выполнять каждому за компьютером. При выполнении глобальных проектов рекомендуется объединять школьников в пары. Текущий контроль уровня усвоения материала осуществляется по результатам выполнения рефлексивных упражнений и практических заданий. Итоговый контроль осуществляется по результатам разработки проектов. Формы подведения итогов: презентация проекта, викторина, игра.

Особенности проведения занятий:

- теоретический материал подается небольшими порциями с использованием

игровых ситуаций;

- для закрепления и проверки уровня усвоения знаний применяются рефлексивные интерактивные упражнения;
- практические задания составляются так, чтобы время на их выполнение не превышало 20 минут;
- практические задания могут включать в себя работу с готовым проектом на редактирование скрипта, на дополнение скрипта командами, на сборку скрипта самостоятельно;
- работу по созданию глобальных творческих проектов следует начинать с разъяснения алгоритма разработки проектов, адаптированного под возраст школьников.

Содержание программы

Рассматриваемые вопросы: алгоритм, свойства алгоритмов, способы записи алгоритмов, команды и исполнители, Scratch - возможности и примеры проектов, интерфейс и главное меню Scratch, сцена, объекты (спрайты), свойства объектов, методы и события, программа, команды и блоки, программные единицы: процедуры и скрипты. линейный алгоритм, система координат на сцене Scratch, основные блоки, цикл в природе, циклические алгоритмы, цикл «Повторить n раз», цикл «Всегда», библиотека костюмов и сцен Scratch, анимация формы, компьютерная графика, графические форматы и т. д. Запись звука, форматы звуковых файлов, озвучивание проектов Scratch, сообщество Scratch, регистрация на сайте, публикация проектов Scratch, использование заимствованных кодов и объектов.

Учебно-тематический план 1 год обучения

| №п/ п | Разделы программы И темы учебных занятий | Количество часов | | |
|----------|---|------------------|----------|-------|
| | | теория | практика | всего |
| 1. | Вводное занятие | 1 | 1 | 2 |
| 2. | Знакомство со Scratch. | 2 | 3 | 5 |
| 3. | Знакомство с эффектами | 2 | 2 | 4 |
| 4. | Знакомство с отрицательными числами | 2 | 2 | 4 |
| 5. | Знакомство с пером | 2 | 2 | 4 |
| 6. | Циклы | 3 | 4 | 7 |
| 7. | Условный блок | 2 | 2 | 4 |
| 8. | Знакомство с координатами ХиУ | 2 | 2 | 4 |
| 9. | Творческий блок. Создание мультфильмов и игр. | 30 | 30 | 60 |
| 10. | Подготовка к конкурсам и выставкам | 4 | 4 | 8 |
| 11. | Знакомство с переменными | 2 | 2 | 4 |
| 12. | Знакомство с понятием сенсоры | 2 | 2 | 4 |

| | | | | |
|---------------|--------------------------|-----------|-----------|------------|
| 13. | Итоговый годовой проект. | 2 | 2 | 4 |
| Итого: | | 56 | 58 | 114 |

Календарно-тематическое планирование

| № занятия | Тема | Количество часов | Дата |
|-----------|--|------------------|------|
| 1 | Знакомство со средой Скретч. Понятие спрайта и объекта. Создание и редактирование спрайтов и фонов для сцены. | 3 | |
| 2 | Знакомство со средой Скретч (продолжение). Пользуемся помощью Интернета. Поиск, импорт и редакция спрайтов и фонов из Интернета. | 3 | |
| 3 | Управление спрайтами: команды идти, повернуться на угол, опустить перо, поднять перо, очистить. | 3 | |
| 4 | Координатная плоскость. Точка отсчёта, оси координат, единица измерения расстояния, абсцисса и ордината. | 3 | |
| 5 | Навигация в среде Скретч. Определение координат спрайта. Команда идти в точку с заданными координатами. | 3 | |
| 6 | Создание проекта «Кругосветное путешествие Магеллана». Команда Плыть в точку с заданными координатами | 4 | |
| 7 | Создание проекта «Кругосветное путешествие Магеллана» (продолжение). Режим презентации. | 4 | |
| 8 | Понятие цикла. Команда Повторить . Рисование узоров и орнаментов. | 3 | |
| 9 | Конструкция всегда . Создание проектов «Берегись автомобиля!» и «Гонки по вертикали». Команда если край, оттолкнуться. | 3 | |
| 10 | Ориентация по компасу. Управление курсом движения. Команда повернуть в направлении . Проект «Полёт самолёта». | 4 | |
| 11 | Спрайты меняют костюмы. Анимация. Создание проектов «Осьминог», «Девочка, прыгающая на скакалке» и «Бегущий человек». | 4 | |
| 12 | Создание мультипликационного сюжета «Кот и птичка». | 3 | |
| 13 | Создание мультипликационного сюжета «Кот и птичка» (продолжение). | 3 | |

| | | | |
|----|---|---|--|
| 14 | Соблюдение условий. Сенсоры. Блок если . Управляемый стрелками спрайт. | 3 | |
| 15 | Создание коллекции игр: «Лабиринт», «Кружащийся котёнок». | 3 | |
| 16 | Пополнение коллекции игр: «Опасный лабиринт». | 4 | |
| 17 | Составные условия. Проекты «Хождение по коридору», «Слепой кот», «Тренажёр памяти». | 4 | |
| 18 | Датчик случайных чисел. Проекты «Разноцветный экран», «Хаотичное движение», «Кошки-мышки», «Вырастим цветник». | 4 | |
| 19 | Циклы с условием. Проект «Будильник». | 4 | |
| 20 | Запуск спрайтов с помощью мыши и клавиатуры. Проекты «Переодевалки» и «Дюймовочка». | 3 | |
| 21 | Самоуправление спрайтов. Обмен сигналами. Блоки передать сообщение и когда я получу сообщение . Проекты «Лампа» и | 3 | |
| | «Диалог». | 3 | |
| 22 | Доработка проектов «Магеллан», «Лабиринт». | 4 | |
| 23 | Датчики. Проекты «Котёнок-обжора», «Презентация». | 3 | |
| 24 | Переменные. Их создание. Использование счётчиков. Проект «Голодный кот». | 3 | |
| 25 | Ввод переменных. Проект «Цветы». Доработка проекта «Лабиринт» - запоминание имени лучшего игрока. | 3 | |
| 26 | Ввод переменных с помощью рычажка. Проекты «Цветы» (вариант-2), «Правильные многоугольники». | 3 | |
| 27 | Список как упорядоченный набор однотипной информации. Создание списков. Добавление и удаление элементов. Проекты «Гадание», «Назойливый собеседник». | 3 | |
| 28 | Поиграем со словами. Строковые константы и переменные. Операции со строками. | 3 | |
| 29 | Создание игры «Угадай слово». | 3 | |
| 30 | Создание тестов – с выбором ответа и без. | 3 | |
| 31 | Создание проектов по собственному замыслу. | 3 | |

| | | | |
|----|---|---|--|
| 32 | Регистрация в Скретч-сообществе. Публикация проектов в Сети | 3 | |
| 33 | Итоговый проект | 3 | |
| 34 | Итоговый проект | 3 | |

Перечень учебно-методического и материально-технического обеспечения

Литература:

1. Матяш Н. В. Психология проектной деятельности школьников в условиях технологического образования/ Под ред. В. В. Рубцова. Мозырь: РИФ «Белый ветер», 2000. 285 с.
2. Патаракин Е. Д. Учимся готовить в среде Скретч (Учебно-методическое пособие). М: Интуит.ру, 2008. 61 с.
3. Пахомова Н. Ю. Метод учебного проекта в образовательном учреждении: Пособие для учителей и студентов педагогических вузов. М.: Аркти, 2008. 112 с.
4. Пашковская Ю.В. «Творческие задания в среде программирования Scratch. 5-6 классы. Рабочая тетрадь» - М.: БИНОМ. Лаборатория знаний, 2013.
5. Примерные программы начального общего образования [Электронный ресурс] // Федеральный государственный образовательный стандарт [сайт]. URL: <http://standart.edu.ru/catalog.aspx?CatalogId=531>
6. Хохлова М. В. Проектно-преобразовательная деятельность младших школьников. // Педагогика. 2004. № 5. С. 51–56.
7. Цветкова М.С., Масленикова О.Н. «Практические задания с использованием информационных технологий для 5-6 классов: Практикум» - М.: БИНОМ. Лаборатория знаний, 2007.
8. Цукерман Г. А. Что развивает и чего не развивает учебная деятельность младших школьников? // Вопросы психологии. 1998. № 5. С. 68–81.
9. Скретч [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вернуться домой». URL: <http://letopisi.ru/index.php/Скретч>
10. Школа Scratch [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вернуться домой». URL: http://letopisi.ru/index.php/Школа_Scratch
11. Scratch | Home | imagine, program, share [сайт]. URL: <http://scratch.mit.edu>
12. Scratch | Галерея | Gymnasium №3 [сайт]. URL: <http://scratch.mit.edu/galleries/view/54042>

Электронные образовательные ресурсы:

1. <http://scratch.mit.edu> – официальный сайт Scratch
2. <http://letopisi.ru/index.php/Скретч> - Скретч в Летописи.ру
3. <http://setilab.ru/scratch/category/commun> - Учитесь со Scratch
4. http://socobraz.ru/index.php/Школа_Scratch
5. <http://scratch.sostradanie.org> – Изучаем Scratch

6. <http://odjiri.narod.ru/tutorial.html> – учебник по Scratch
7. <http://younglinux.info> - Цикл из 10 уроков “Введение в Scratch”
8. <http://anngeorg.ru/info/scratch> – Знакомимся с программой Scratch
9. LearningApps.org

Техническое оборудование:

Практические работы проводятся на базе Центра образования цифрового и гуманитарного профилей «Точка роста» с применением материально-технической базы центра.

Компьютерные программы:

- Операционная система Windows
- Браузер Google Chrome
- Среда программирования Scratch 3.0
- Среда КуМИР
- Графический растровый редактор
- Пакет программ Microsoft Office

**Контрольная работа для проверки полученных навыков по темам
«Линейные алгоритмы», «Условные алгоритмы»**

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

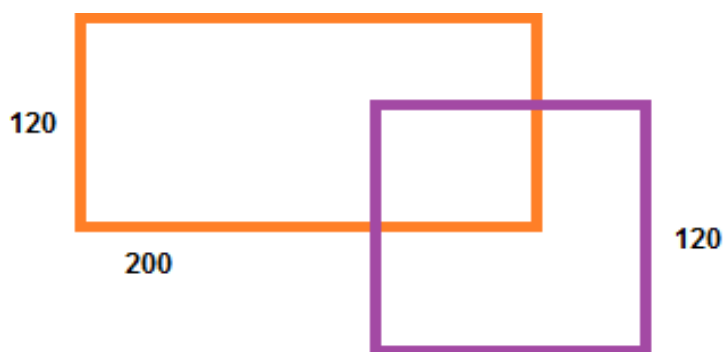


Рис. 94. Иллюстрация к задаче

4. Написать программу в среде Scratch, изображающую следующий рисунок.

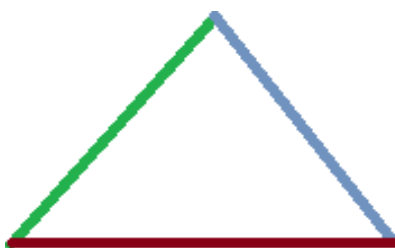


Рис. 95. Иллюстрация к задаче

5. Написать программу в среде Scratch, изображающую символику «Олимпийские кольца» (рис. 96).

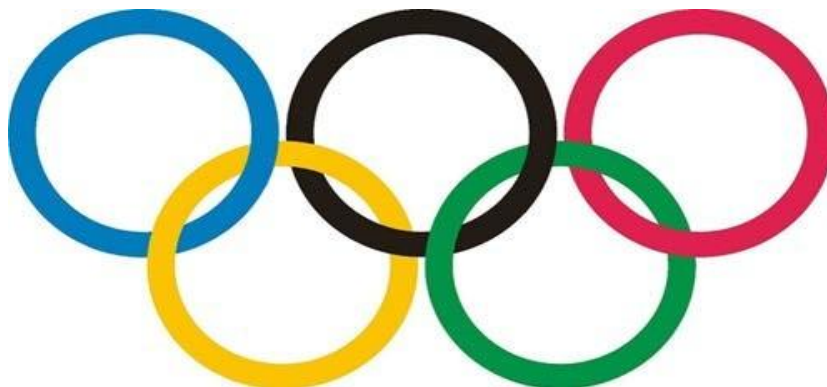


Рис. 96. Иллюстрация к задаче

6. Написать программу в среде Scratch: *Для введённых с клавиатуры чисел x и y вычислить значение выражения $x^2 + y/3$.*

7. Написать программу в среде Scratch: Для введённых с клавиатуры чисел a и b выяснить, делится ли a на b .

8. Написать программу в среде Scratch: Пользователь вводит целое число. Программа должна ответить, чётным или нечётным является это число, делится ли оно на 3 и делится ли оно на 6.

9. Написать программу в среде Scratch: Пользователь вводит порядковый номер пальца руки (начиная с мизинца). Необходимо показать его название на экране.

10. Написать программу в среде Scratch: Пользователь вводит пароль. По данному паролю определите степени доступа: $[0, 1000]$ – доступен модуль A , $[1001, 2500]$ или $[3000, 5000]$ – доступны модули B и C , $[9400, 10000]$ или $[10500, 50000]$ – доступен модуль D . Если значение не попало ни в один из указанных отрезков, то в доступе отказано!

Контрольная работа для проверки полученных навыков по темам «Циклические алгоритмы», «Работа со списками»

1. Написать программу в среде Scratch, изображающую следующий рисунок.

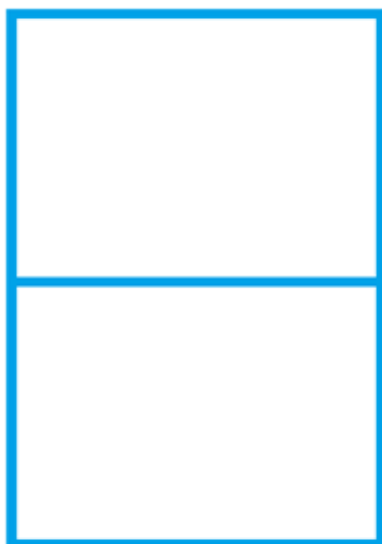


Рис. 97. Иллюстрация к задаче

2. Написать программу в среде Scratch, изображающую следующий рисунок.

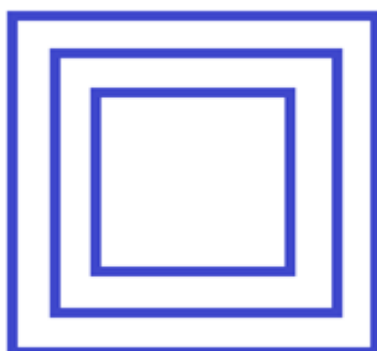


Рис. 98. Иллюстрация к задаче

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

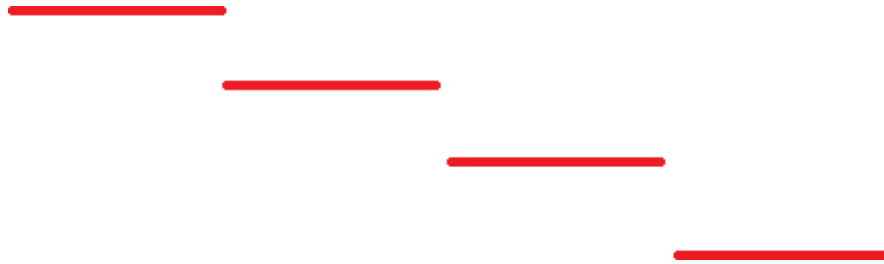


Рис. 99. Иллюстрация к задаче

4. Написать программу в среде Scratch: *Вывести на экран первые десять степеней двойки.*

5. Написать программу в среде Scratch: *Найти наибольший общий делитель двух чисел, введённых пользователем.*

6. Написать программу в среде Scratch: *В списке хранятся данные о температуре в городке N за 12 месяцев. Выведите температуру с марта по сентябрь.*

7. Написать программу в среде Scratch: *В списке хранятся данные о температуре в городке N за 12 месяцев. Выведите максимальную температуру за год.*

8. Написать программу в среде Scratch: *В списке хранится информация о четвертных оценках класса из 20 человек по информатике. Определите, сколько человек получили пятёрки за четверть.*

9. Написать программу в среде Scratch: *В списке хранится информация о четвертных оценках класса из 20 человек по информатике. Определите, какой процент хорошистов в классе.*

10. Написать программу в среде Scratch: *В списке хранится информация о четвертных оценках класса из 20 человек по информатике. Определите средний балл в классе.*

Также предполагается итоговая аттестация в форме разработки и защиты индивидуального проекта.

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: выполнение лабораторной работы 1.

2. Линейные алгоритмы

Рекомендуемое количество часов на данную тему — 2 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:** выполнение лабораторной работы 2.

3. Работа с переменными

Рекомендуемое количество часов на данную тему — 2 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 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:** выполнение лабораторной работы 3.

4. Условные алгоритмы

Рекомендуемое количество часов на данную тему — 2 часа.

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: выполнение лабораторной работы 4.

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: выполнение лабораторной работы 5.

6. 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: выполнение лабораторной работы 6.

Описание лабораторных работ

Лабораторная работа 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 1996 г. С. Пейперт разработал Logo.

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

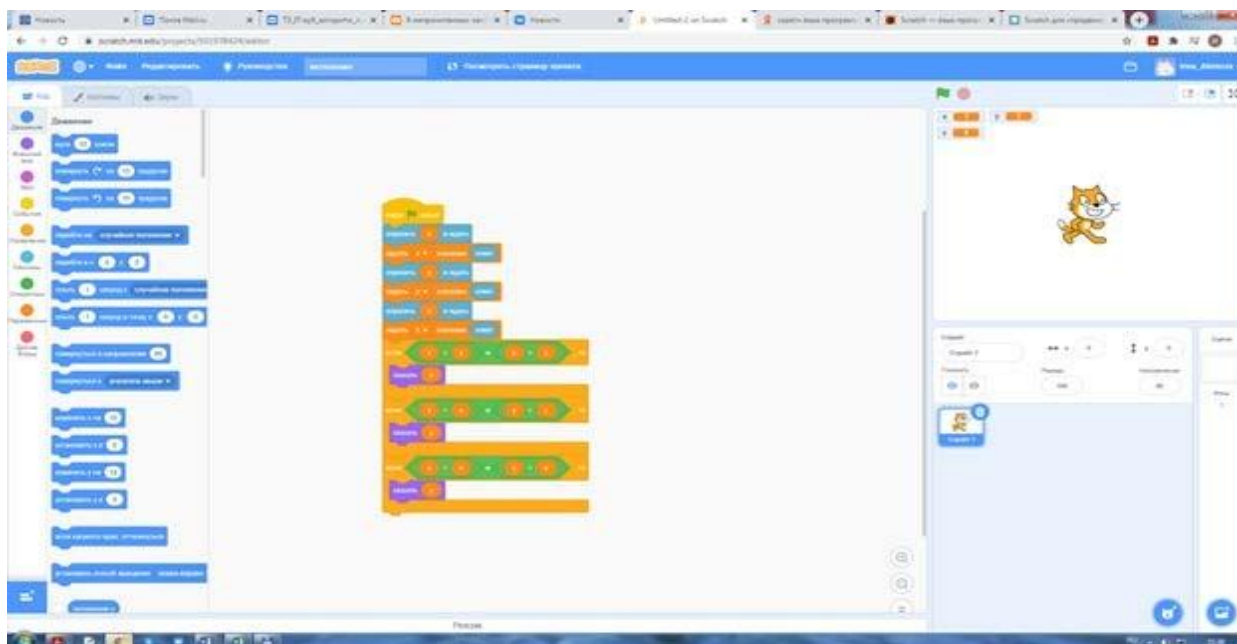


Рис. 100. Вид онлайн-среды Scratch

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 медведя (рис. 101).
3. Выберите фон из коллекции фонов, например, лес (рис. 102).
4. Смените спрайт, удалите фон.



Рис. 101. Вид спрайта Медведь



Рис. 102. Вид фона Лес

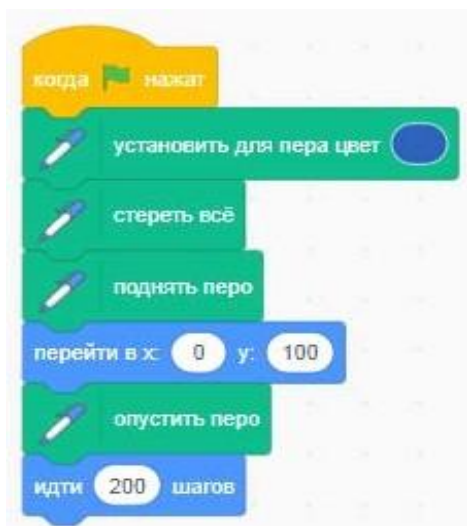


Рис. 103. Вид скрипта

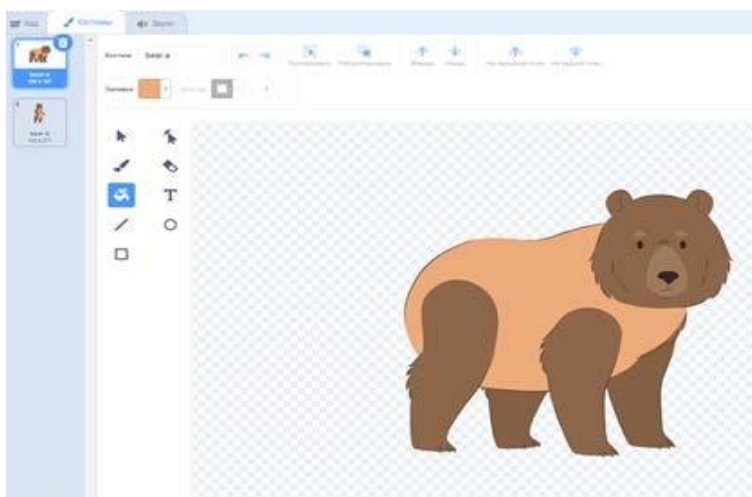


Рис. 104. Изменение спрайта в графическом редакторе

5. Выполните следующий скрипт (рис. 103).
6. Объясните, что выполняет скрипт.
7. Измените спрайт в графическом редакторе Scratch (рис. 104).
8. Выполните следующий скрипт (рис. 105).
9. Объясните, что выполняет скрипт.
10. Выполните следующий скрипт (рис. 106).
11. Объясните, что выполняет скрипт.

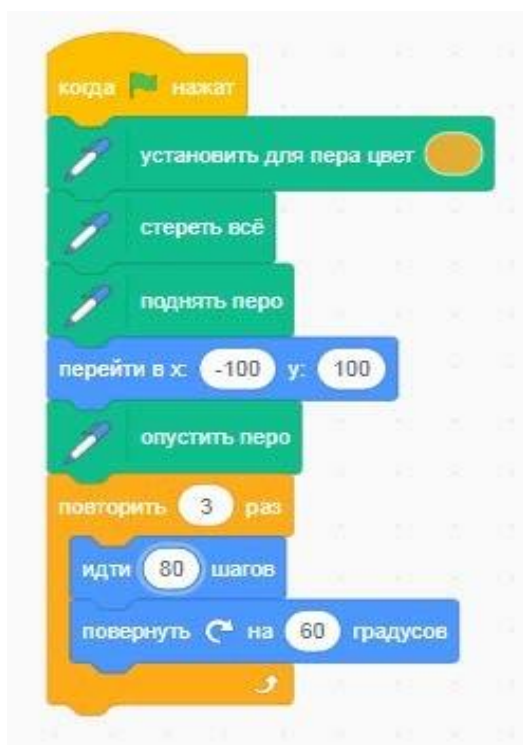


Рис. 105. Вид скрипта

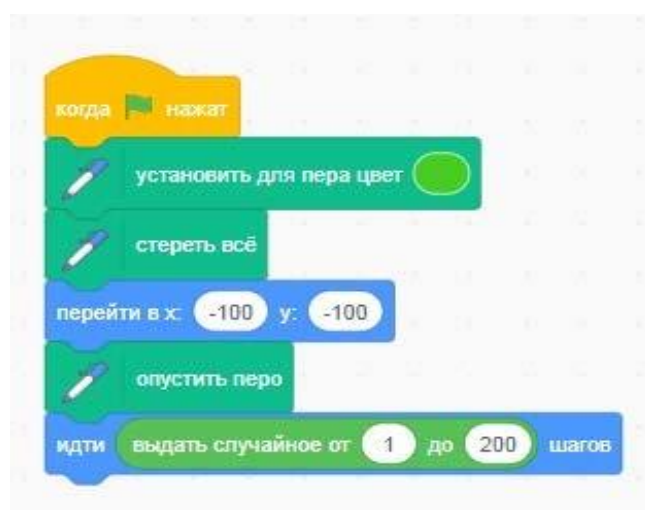


Рис. 106. Вид скрипта

Выводы: 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 2 . 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 107.

Линейные алгоритмы реализуются в любом языке программирования, можно привести множество примеров линейных алгоритмов, например алгоритм вычисления произведения двух целых чисел (рис. 108).



Рис. 107. Блок-схема линейного алгоритма

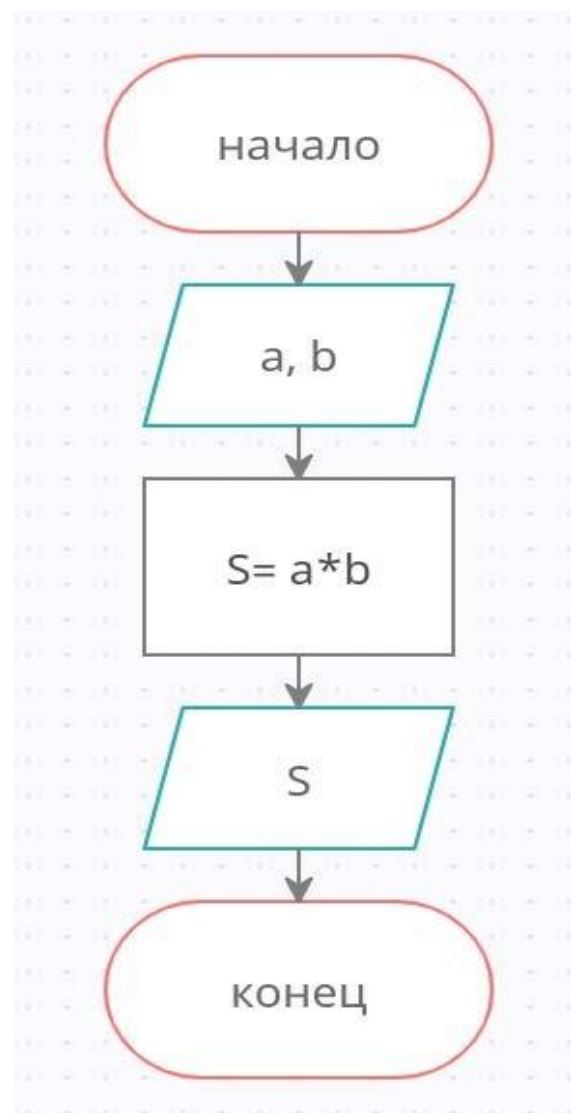


Рис. 108. Пример линейного алгоритма

Практическая часть

Цель работы: ознакомление с построением и выполнением 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 60градусов».
3. Выполните следующую программу (рис. 109). Представляет ли она линейный алго-ритм?

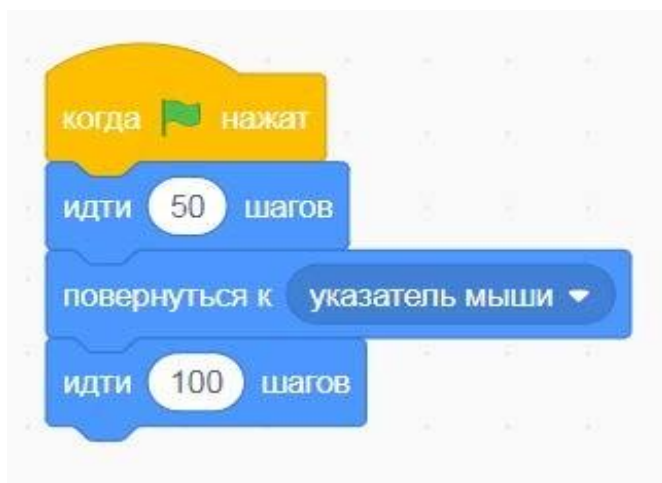


Рис. 109. Вид программы в среде Scratch

4. Пройдите 200 шагов, поверните на 45 градусов по часовой стрелке, пройдите ещё 200 шагов.
5. Поверните против часовой стрелки на 180 градусов, пройдите 200 шагов, поверните на 30 градусов по часовой стрелке, пройдите 100 шагов.
6. Установите зелёный цвет, пройдите 50 шагов, установите зелёный цвет, пройдите 10 шагов.

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

Контрольные вопросы:

1. Дайте определение линейного алгоритма.
2. Приведите пример линейных алгоритмов.
3. Какие блоки вы использовали в лабораторной работе при создании линейных программ?

Лабораторная работа 3 . 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,

подготавливая 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. 110).



Рис. 110. Команда «Создание переменной»

В появившемся диалоговом окне указываем имя переменной.

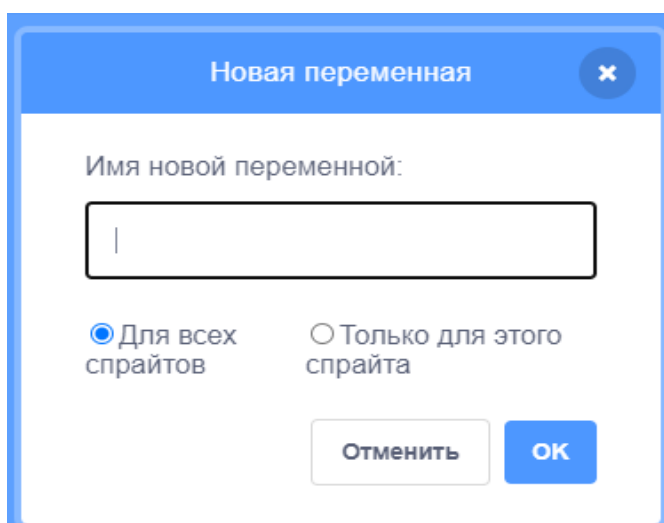


Рис. 111. Диалоговое окно для создания переменной

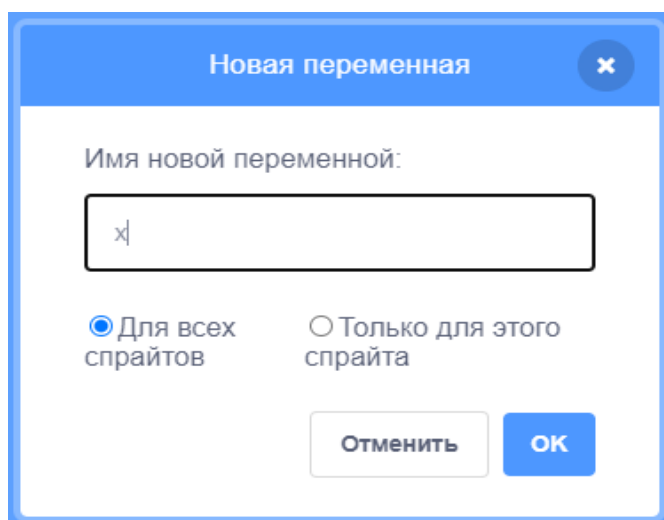


Рис. 112. Диалоговое окно для создания переменной

После задания имени переменной она появляется в проекте (рис. 113).

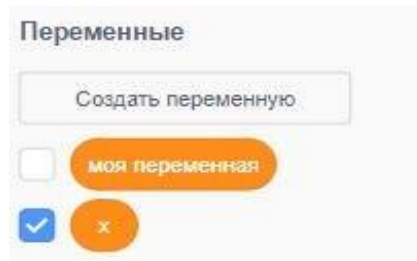


Рис. 113. Вид созданной переменной в разделе «Переменные»

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.

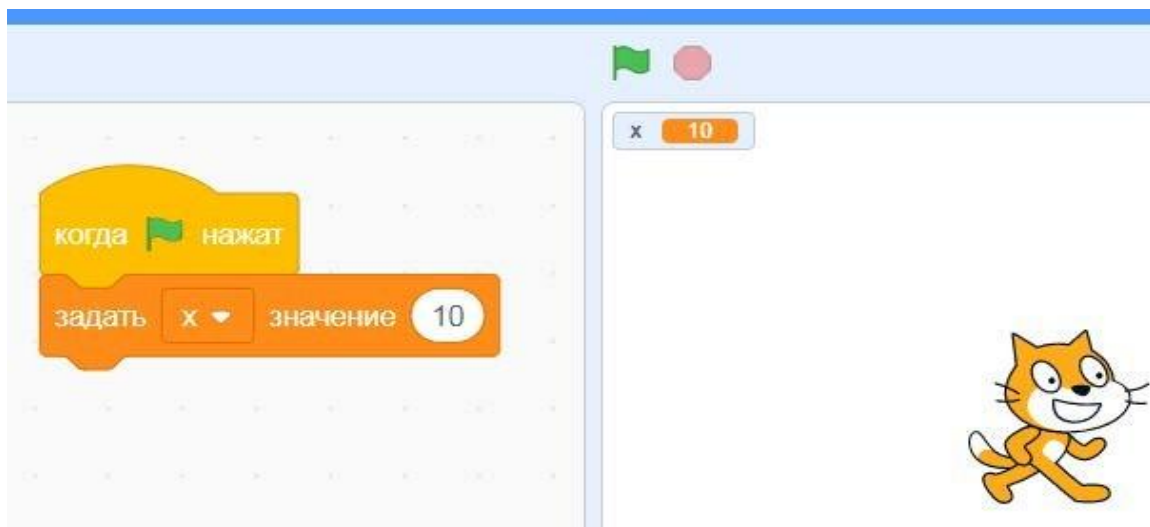


Рис. 114. Пример задания значения переменной

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. 115). По умолчанию переменная и её значение отображается на сцене, но с помощью данных блоков мы можем скрывать или удалять переменную со сцены.

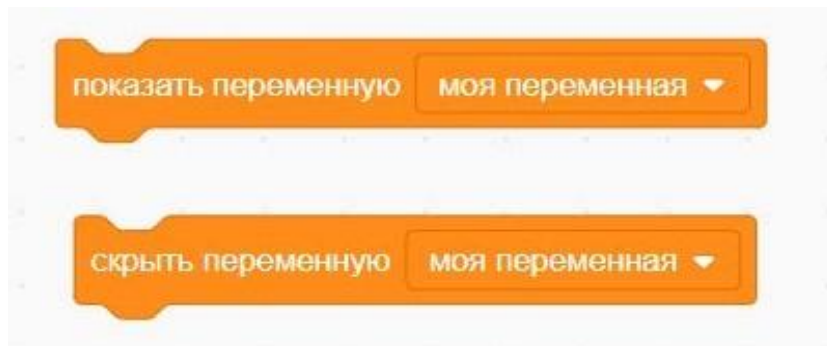


Рис. 115. Вид блоков по работе с переменными

Также можно изменять вид изображения переменной на сцене.

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



Рис. 116. 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 , c , fun . Какие инструменты среды вы использовали для этого?
2. Выполните следующую программу (рис. 117).
3. Выполните следующую программу (рис. 118).



Рис. 117. Вид программы



Рис. 118. Вид программы

4. Выполните в среде Scratch решение следующей задачи: «Ввести значение x . Прибавить к x число 15».
5. Выполните в среде Scratch решение следующей задачи: «Для введённых с клавиатуры чисел c и d вычислить значением выражения $3c - 2d$ ».
6. Выполните в среде Scratch решение следующей задачи: «Написать алгоритм вычисления площади квадрата».

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

Контрольные вопросы:

1. Дайте определение **переменной**.
2. Какие виды переменных можно использовать в среде Scratch?
3. С помощью каких блоков можно задать значение переменной в среде Scratch?

Лабораторная работа 4 . Условные алгоритмы

Теоретическая часть

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

Данный условный алгоритм, который также может носить название разветвляющийся, 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. 119).

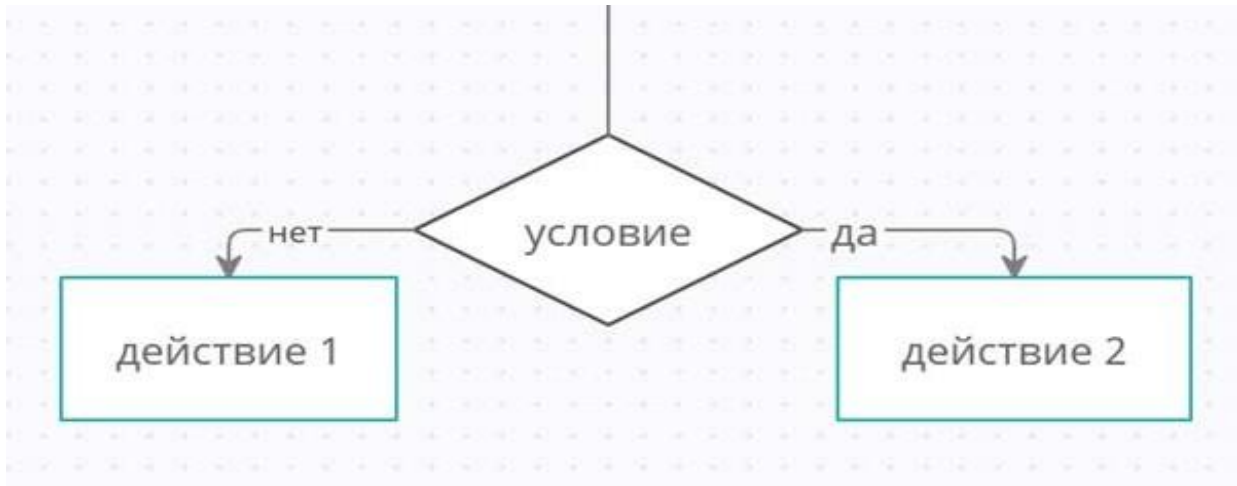


Рис. 119. Полная форма условного алгоритма

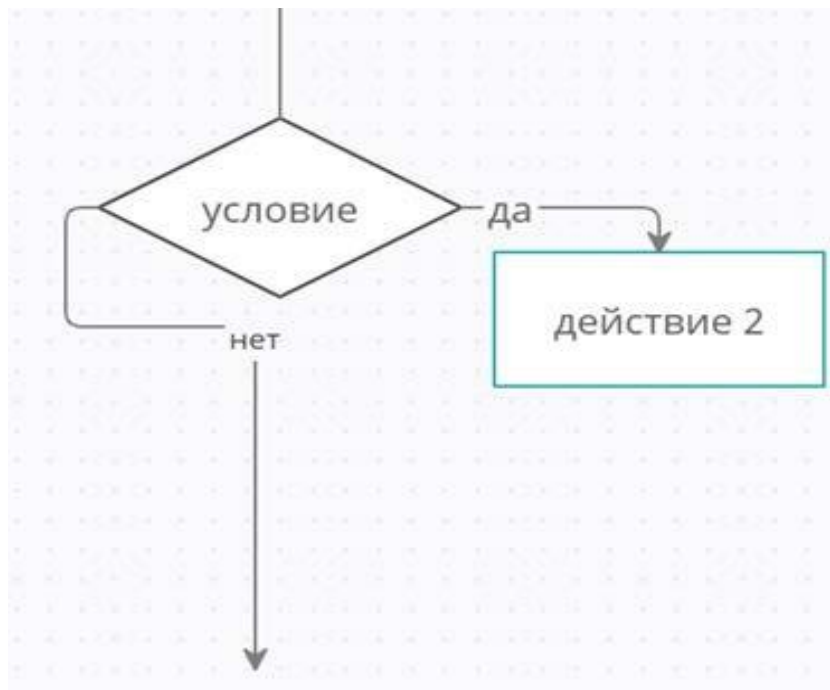


Рис. 120. Сокращённая форма условного алгоритма

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

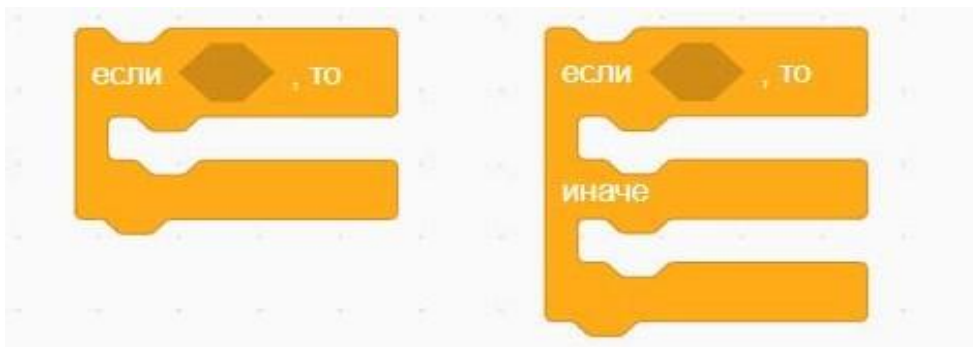


Рис. 121. Блоки для реализации сокращённой и полной формы ветвления

Практическая часть

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

Ход лабораторной работы

Создайте следующий скрипт (рис. 122). Объясните, что выполняет данный скрипт.



Рис. 122. Вид программы

Выполните в среде Scratch решение следующих задач:

1. Если введено число 5, то спрайт говорит пять, иначе спрайт говорит «неверно».
2. Найдите наибольшее из двух чисел, введённых пользователем.
3. Пройдите 100 шагов. Если спрайт коснулся края сцены, то оттолкнуться, если нет — идти ещё 200 шагов.

4. Запросите у пользователя ввод числа. Если введённое пользователем число чёт-ное, то пройти 100 шагов, в противном случае пройдите 300 шагов.

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

Контрольные вопросы:

1. Дайте определение условного алгоритма.
2. Приведите пример условных алгоритмов.
3. Какие блоки используются в среде Scratch для реализации ветвлений?

Лабораторная работа 5 . 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. Далее рассмотрим каждый из них более подробно (рис. 123).

Данный блок используется для организации цикла с параметром, который повторяется заданное число раз. Внутри данного блока обычно располагаются несколько других блоков, которые и образуют тело цикла. Блок-схема работы данного блока представлена ниже (рис. 124):



Рис. 123. Блок для организации цикла

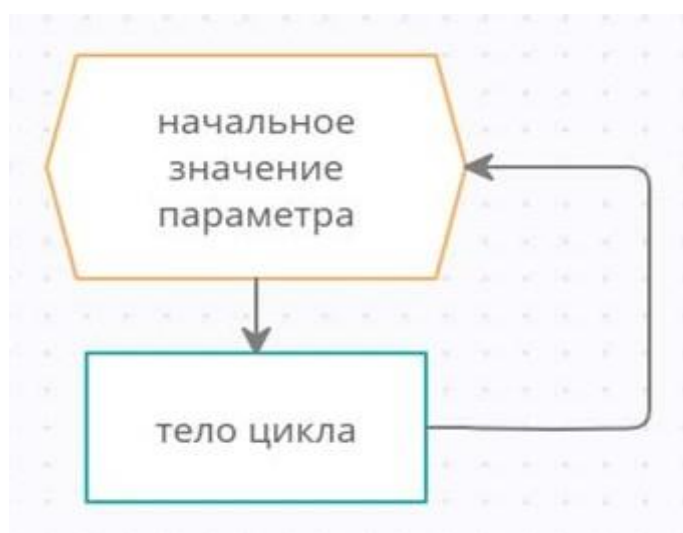
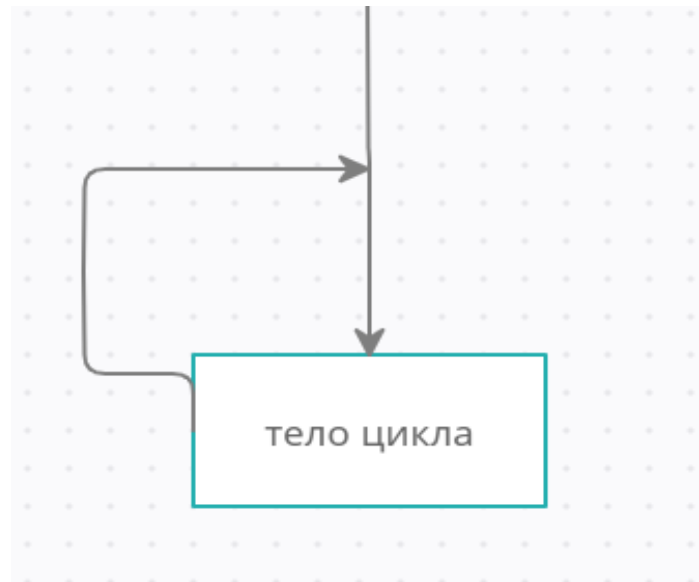




Рис. 125. Блок для организации цикла



Данный блок используется для организации бесконечных повторений. Блок-схема работы данного блока представлена ниже (рис. 126).

Также имеется блок для организации повторения тела цикла (блоков внутри), пока условие не станет истинным. Блок-схема работы данного блока представлена ниже (рис. 127):

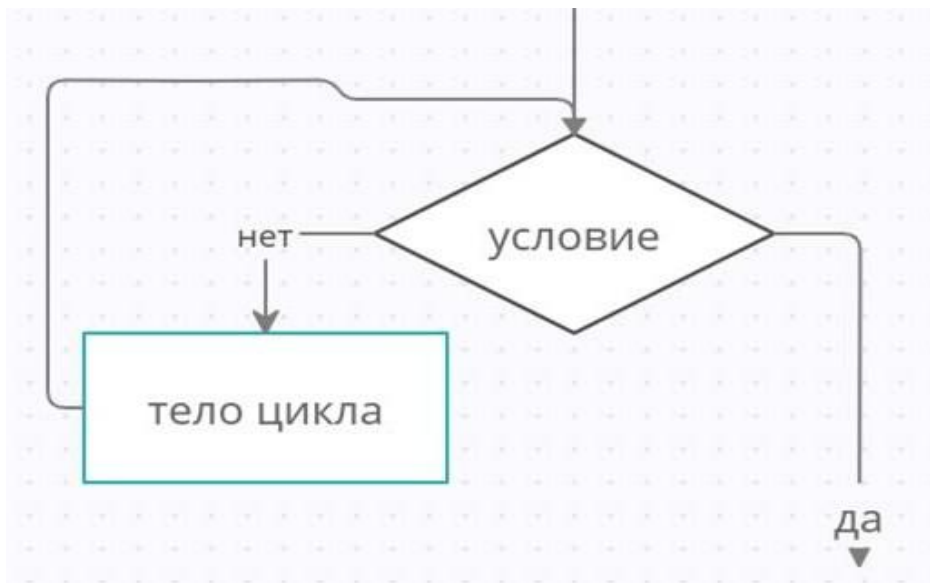


Рис. 127. Блок-схема цикла

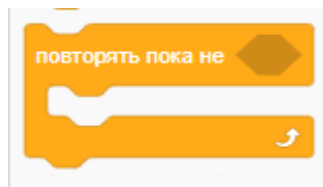


Рис. 128. Блок для организации цикла

Практическая часть

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

Ход лабораторной работы

1. Выполните следующую программу (рис. 129).

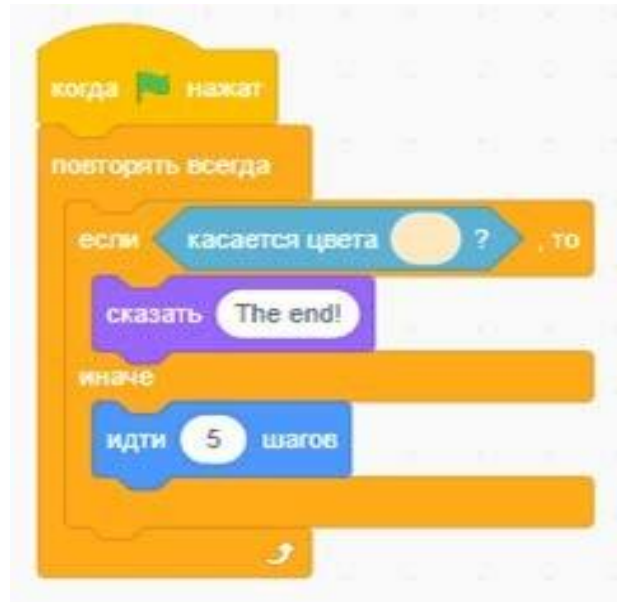


Рис. 129. Вид программы

2. Выполните следующую программу (рис. 130).

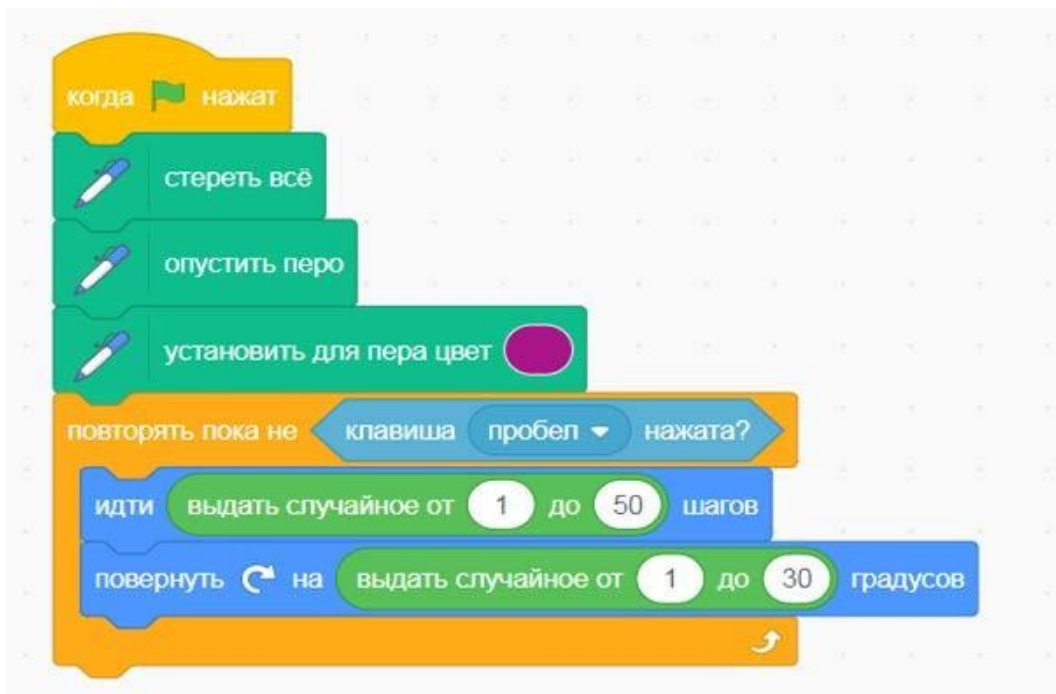


Рис. 130. Вид программы

Выполните в среде Scratch решение следующих задач.

3. Нарисуйте квадрат со стороной 100 (рис. 131).

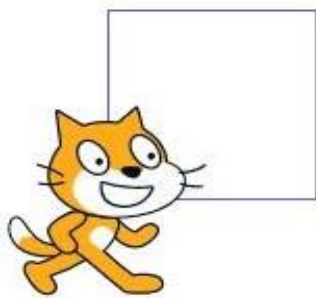


Рис. 131. Пример результат работы программы

4. Нарисуйте квадрат, величина стороны которого вводится пользователем.
5. Изобразите цифру 6 (рис. 132).

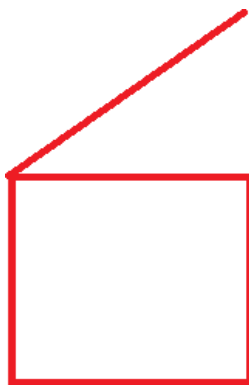


Рис. 132. Пример результат работы программы

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

Контрольные вопросы:

1. Дайте определение **циклического алгоритма**.
2. Приведите примеры циклических алгоритмов.
3. Какие блоки используются для организации циклических алгоритмов в среде Scratch?

Лабораторная работа 6 . Создание подпрограмм

Теоретическая часть

Подпрограммы являются средством реализации структурного программирования. Подпрограммы используются для следующих целей: первая — многократное использование в программе одного и того же фрагмента, а вторая — в случае, когда необходимо выполнить разделение сложной программы на составные части (процедурная декомпозиция). 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 пред-

ставлена всего одна команда — «Создать блок». При выборе данной команды появится диалоговое окно, представленное на рисунке 133.

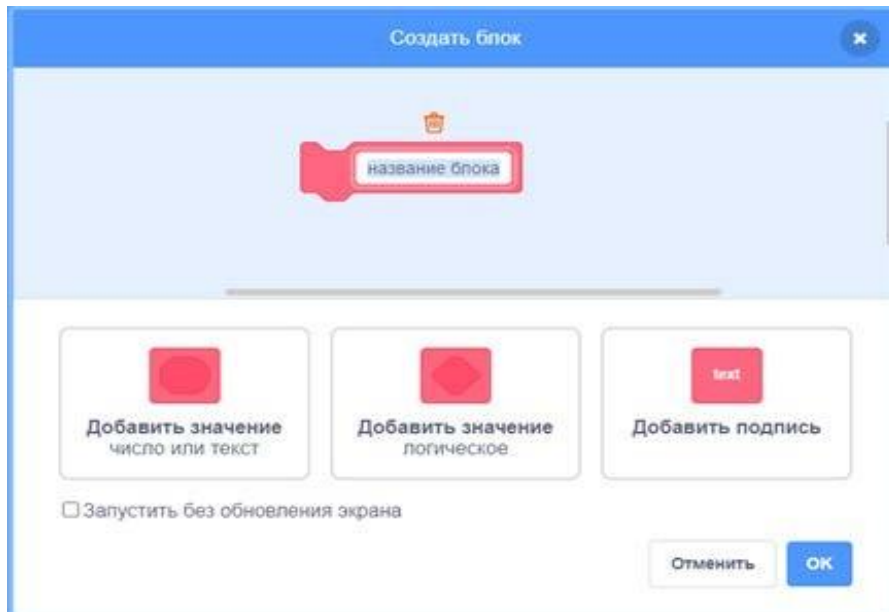


Рис. 133. Вид диалогового окна для создания подпрограммы (нового блока)

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, нажмем ОК, созданный блок появится в списке команд (рис. 134).

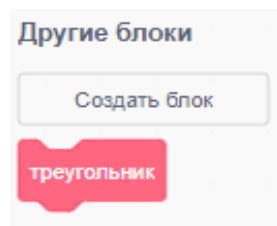


Рис. 134. Вид нового блока

Также в области для создания скриптов появится команда для определения скриптов нового блока (рис. 134).

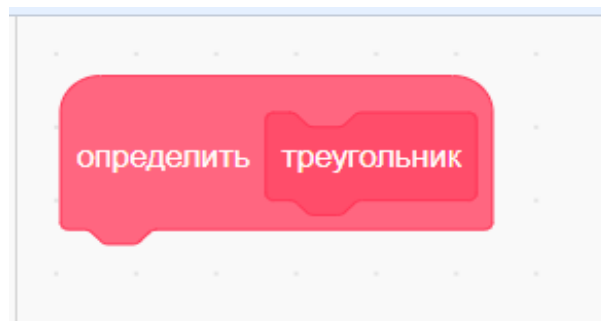


Рис. 135. Вид нового блока в области скриптов

Далее добавляем скрипт для рисования треугольника (рис. 136).

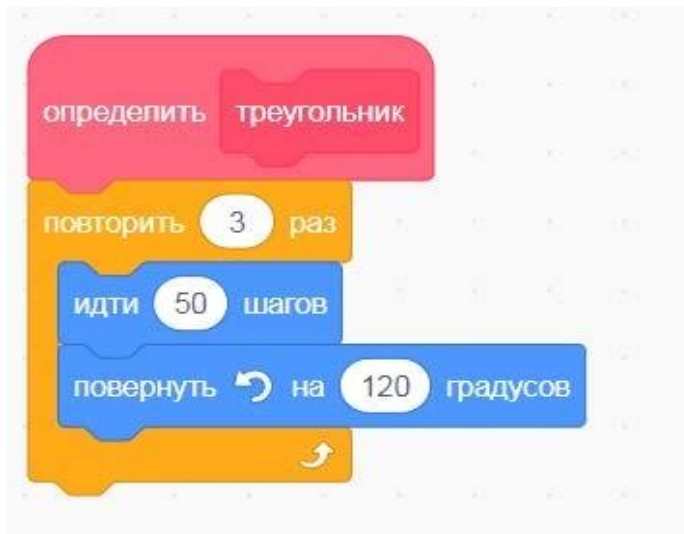


Рис. 136. Вид скрипта

Далее к этому блоку можно обращаться многократно для изображения треугольника. Создадим скрипт, который рисует один треугольник при нажатии клавиши **пробел** (рис. 137).

Рис. 137. Вид программы с новым блоком

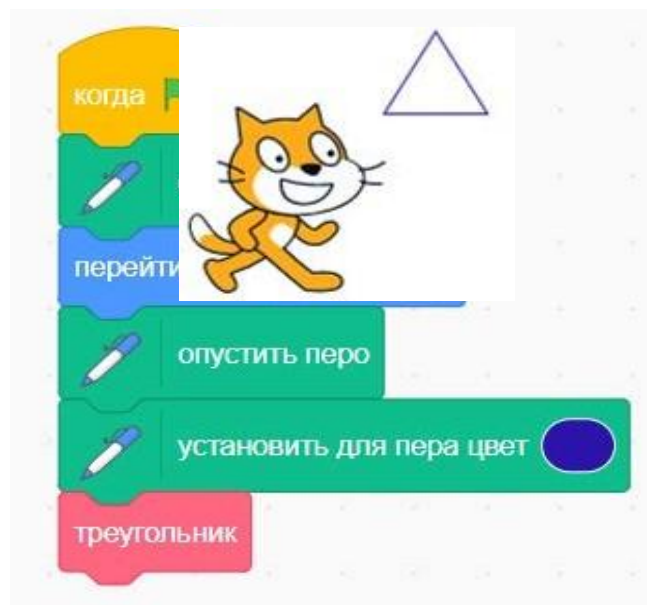


Рис. 138. Результат работы скрипта

Для этого можно объединить их изображение в новую подпрограмму (рис. 139).



Рис. 139. Вид программы с новым блоком

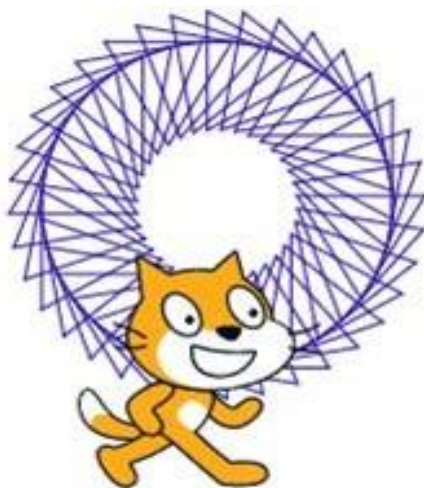


Рис. 140. Результат работы скрипта

Как видно на рисунке, скрипт изобразил несколько треугольников, объединённых в фигуру.

Как было сказано ранее, новый блок может иметь параметры. Параметры служат для передачи данных во внутрь 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. 141). При необходимости можно добавить и второй параметр, также выбрав соответствующую команду «Добавить значение».

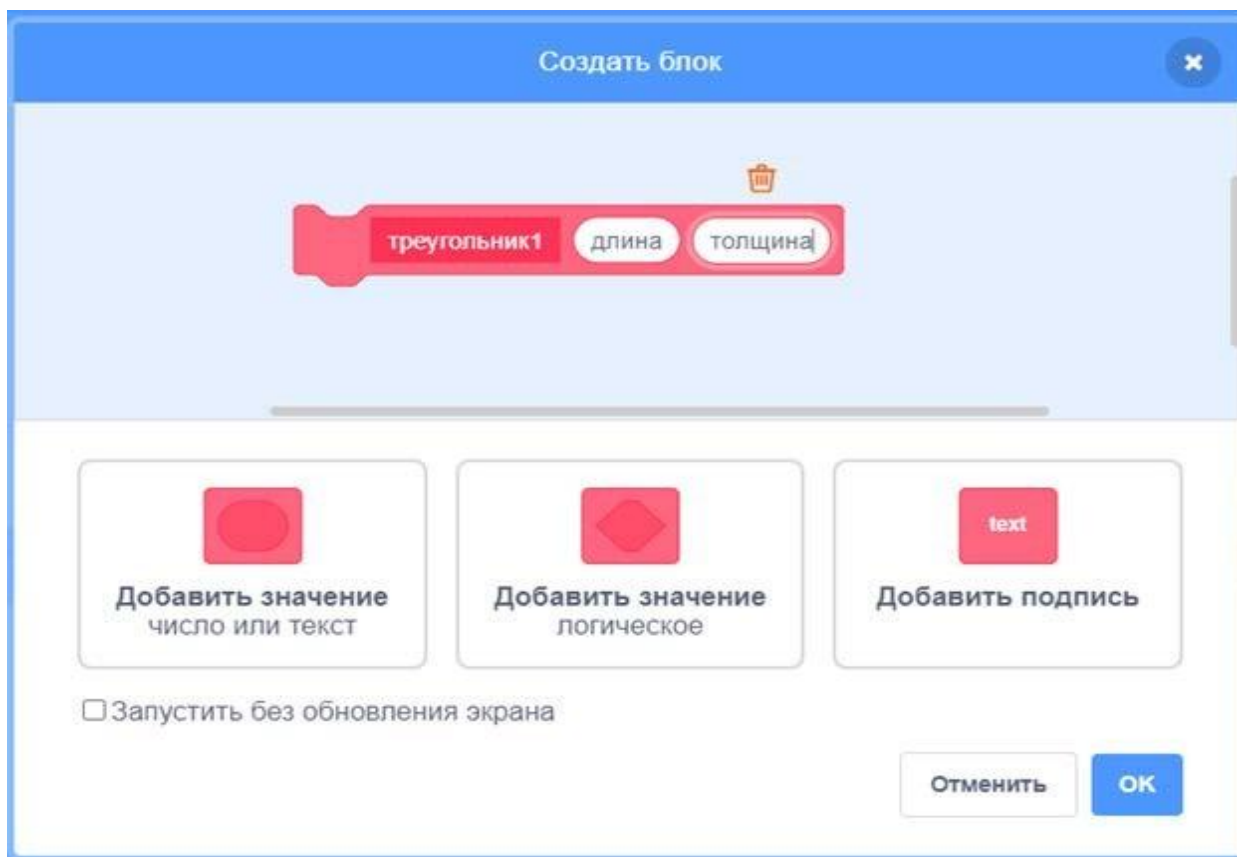


Рис. 141. Вид диалогового окна для создания подпрограммы с параметром
Далее создаём новый скрипт, используя параметры (рис. 142).



Рис. 142. Вид скрипта для создания нового блока

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

Далее создаётся скрипт, также изображающий два треугольника (рис. 143).

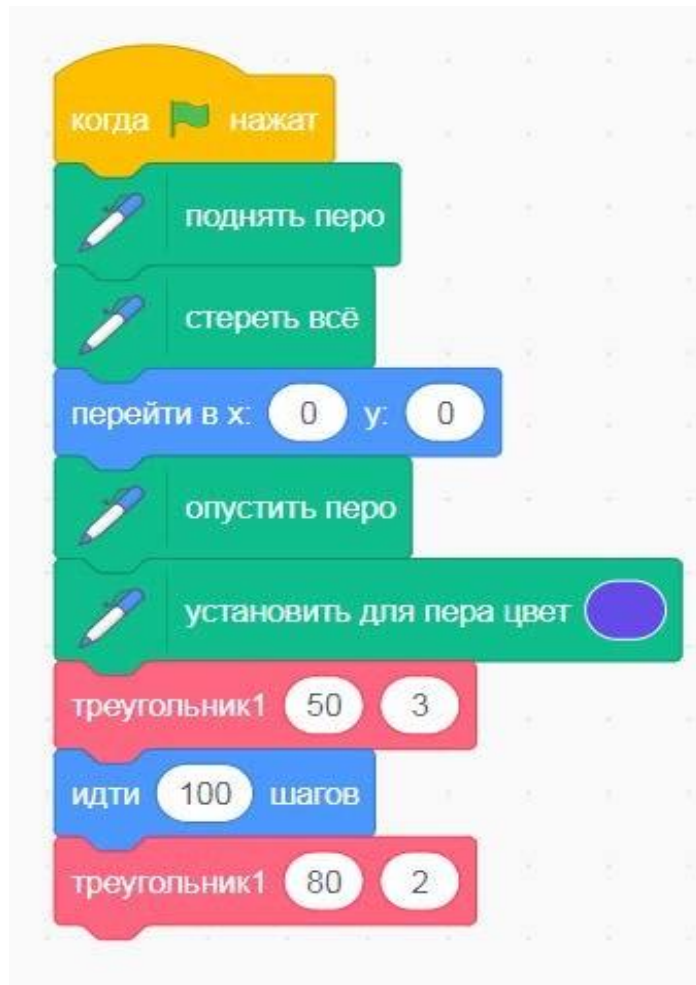


Рис. 143. Вид программы с новым блоком

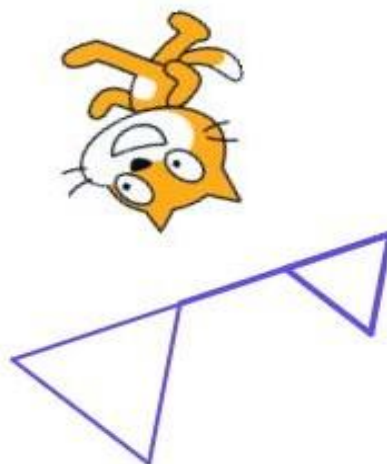


Рис. 144. Результат работы программы

Таким образом, можно создавать несколько новых блоков и 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» (рис. 145).

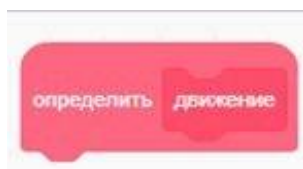


Рис. 145. Создание нового блока

4. Для нового блока реализуйте следующий скрипт, представленный на рисунке 146.

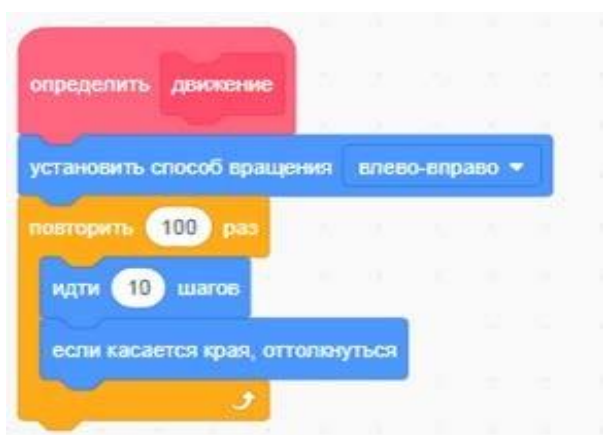


Рис. 146. Вид скрипта для нового блока

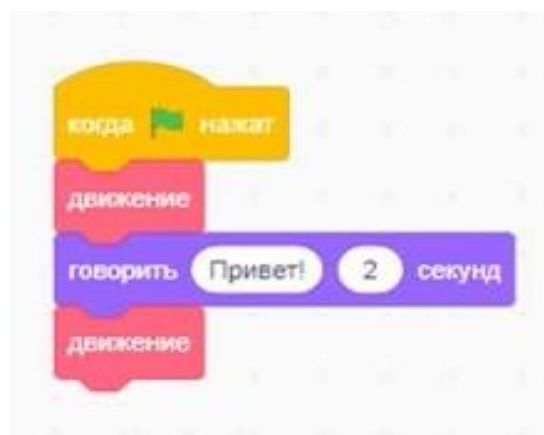


Рис. 147. Вид скрипта с новым блоком

5. Создайте новый скрипт, в котором используется блок «Движение» (рис. 147).

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

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

Контрольные вопросы:

1. Дайте определение **подпрограммы**.
2. Для чего используются новые блоки в среде Scratch?
3. Для чего используются параметры при разработке новых блоков в среде Scratch?

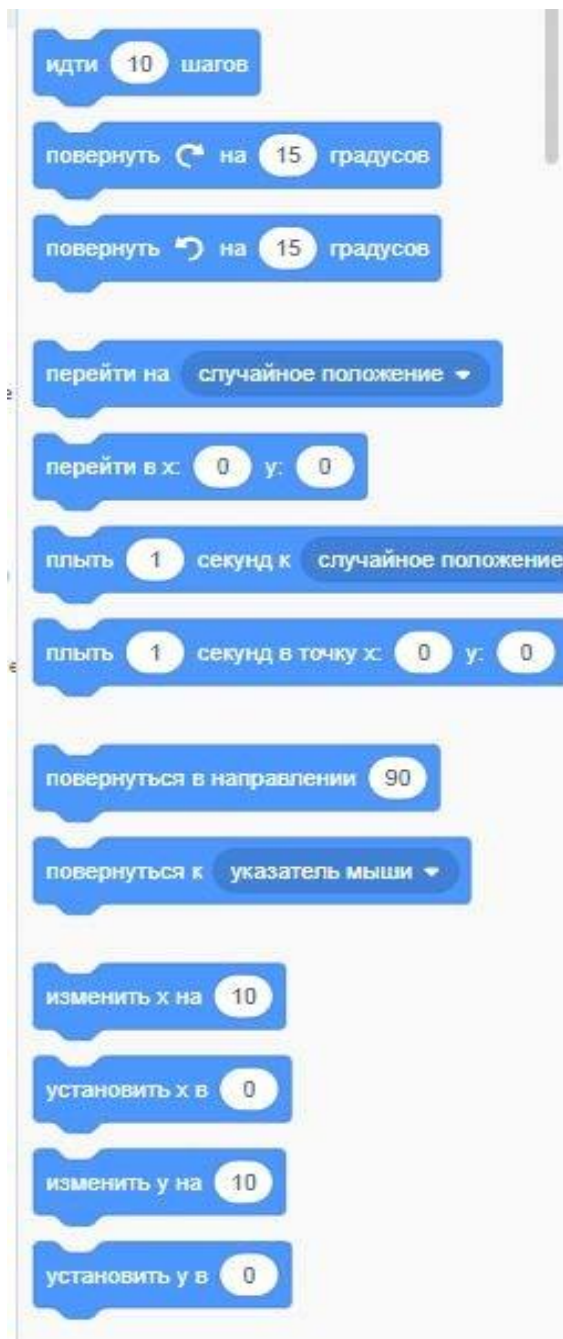
Дидактические материалы

1. 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.

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. 148. Вид фрагмента раздела



«Движение»

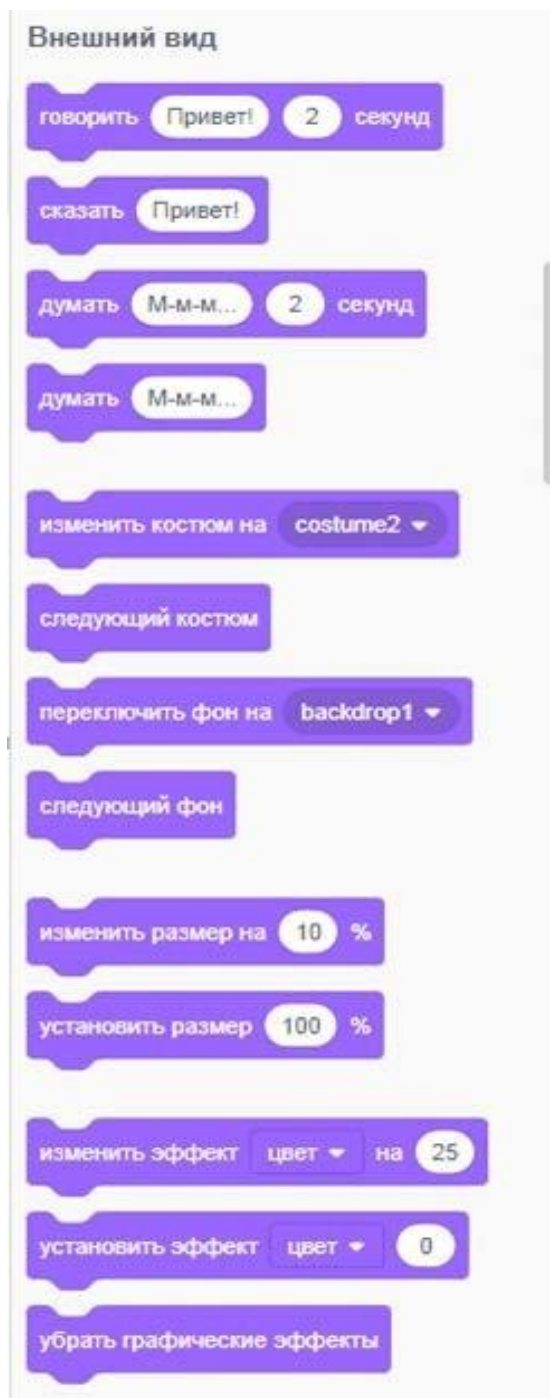


Рис. 149. Вид фрагмента раздела
«Внешний вид»

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 ...».

Раздел **Звук** предназначен для проигрывания звуков. Содержит команды выбора ин-струментов, изменения громкости звуков, темпа и т. д. Команды этого блока могут быть использованы для проигрывания музыкальных фрагментов (рис. 150).

Раздел **События** располагается в начале скрипта. Он позволяет 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. 151).

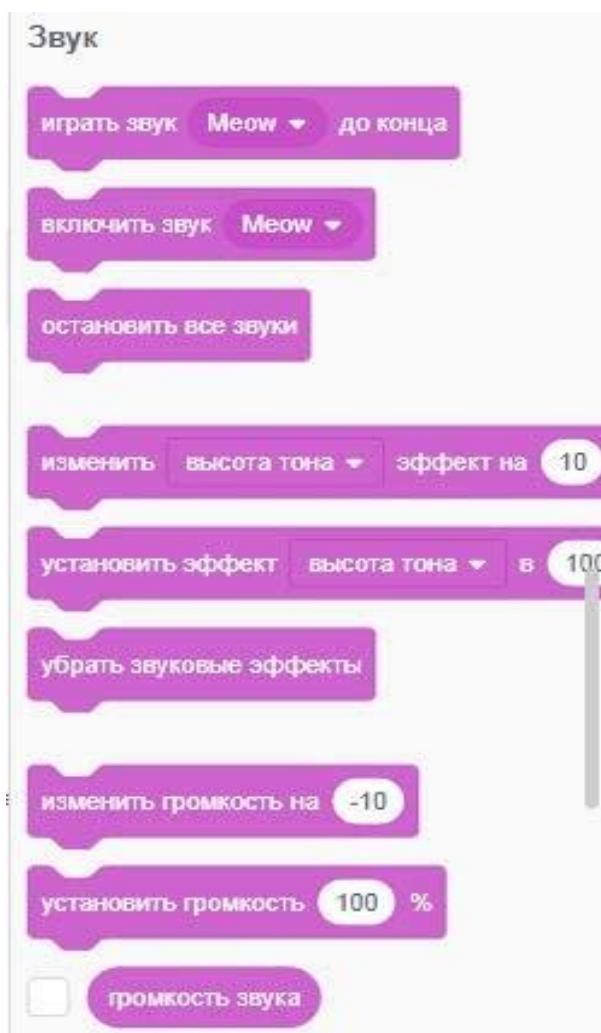


Рис. 150. Вид раздела «Звук»

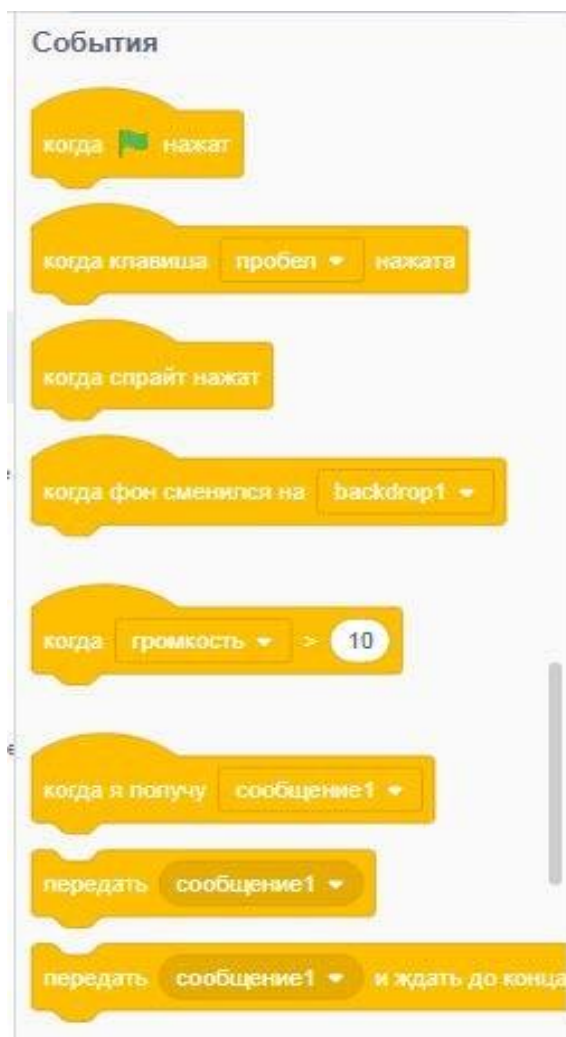


Рис. 151. Вид раздела «События»

Раздел **Управление** позволяет организовывать в программе такие конструкции, как ветвление, цикл различных видов. В данном разделе представлены как раз блоки управления (рис. 152).

Раздел **Переменные** отвечает за использование переменных в программе. Можно создавать, изменять значения как простых переменных, так и списков — фактически массивов (рис. 153).

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. 154).

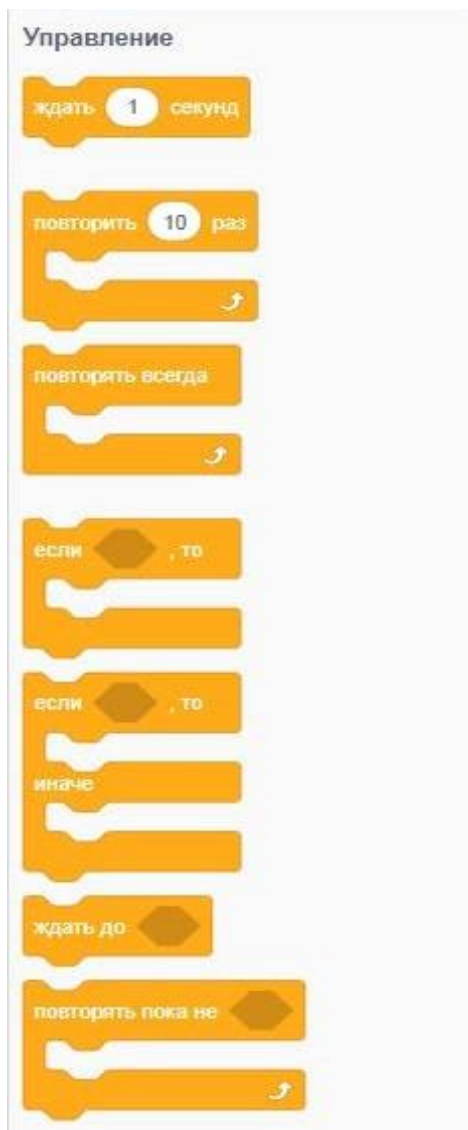


Рис. 152. Вид фрагмента раздела «Управление»

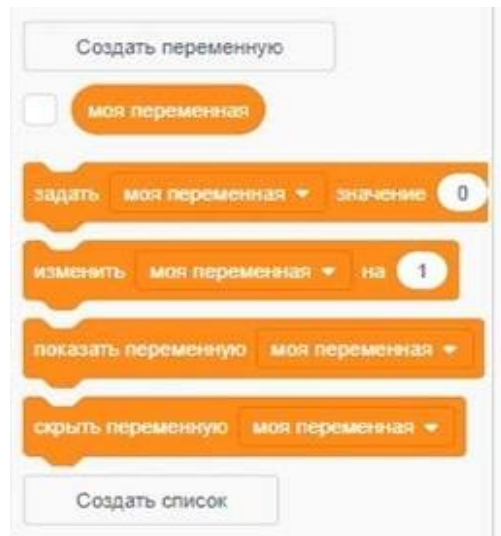


Рис. 153. Вид фрагмента раздела «Переменные»

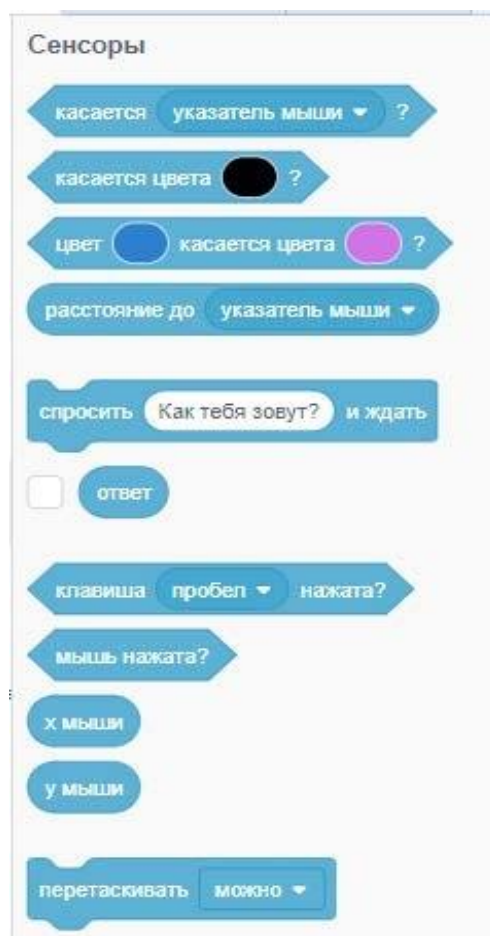


Рис. 154. Вид фрагмента раздела «Сенсоры»



Рис. 155. Вид фрагмента раздела «Операторы»

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. 155).

Самый последний блок **Другие блоки** позволяет реализовывать подпрограммы.