



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЮУрГГПУ»)

ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ

Пропедевтика изучения программирования в школе

Выпускная квалификационная работа
по направлению 44.03.05 Педагогическое образование (с двумя профилями
подготовки)
Направленность программы бакалавриата

«Информатика. Английский язык»

Выполнила:
Студентка группы ОФ-513/093-5-1
Наумова Надежда Александровна

Проверка на объем заимствований:

89,42 % авторского текста

Работа рекомендована к защите
рекомендована/не рекомендована

« 20 » март 20 17 г.

и.о. зав. кафедрой И, ИТ и МОИ

 Рузаков А.А.

Научный руководитель:
к.п.н., доцент кафедры И, ИТ и МОИ
Носова Людмила Сергеевна



Челябинск

2017



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЮУрГГПУ»)

ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

**КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ**

**ПРОПЕДЕВТИКА ИЗУЧЕНИЯ
ПРОГРАММИРОВАНИЯ В ШКОЛЕ**

**Выпускная квалификационная работа
по направлению 44.03.05 Педагогическое образование
Направленность программы бакалавриата
«Педагогическое образование»
профиль «Информатика. Английский язык»**

Проверка на объем заимствований:
_____ % авторского текста

Работа _____ к защите
рекомендована/не рекомендована

« ___ » _____ 20__ г.
зав. кафедрой _____
(название кафедры)
_____ Фамилия И.О.

Выполнил (а):
Студент (ка) группы ОФ-513/093-5-1
Наумова Надежда Александровна

Научный руководитель:
к.п.н, доцент кафедры ИИТиМОИ

Носова Людмила Сергеевна

Челябинск

2017

Оглавление

ВВЕДЕНИЕ	3
ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ОСНОВА ПРОПЕДЕВТИКИ ИЗУЧЕНИЯ ПРОГРАММИРОВАНИЯ В НАЧАЛЬНОЙ ШКОЛЕ.....	5
1.1. Понятие «пропедевтика».....	5
1.2. Особенности изучения программирования в школе	6
1.3. Особенности пропедевтического курса программирования в начальной школе	9
1.4. Программные среды и языки программирования для младших школьников.....	13
Выводы по Главе 1	25
ГЛАВА 2. РАЗРАБОТКА КУРСА ВНЕКЛАССНЫХ ЗАНЯТИЙ «ПРОГРАММИРОВАНИЕ В НАЧАЛЬНОЙ ШКОЛЕ».....	26
2.1. Анализ нормативных документов.....	26
2.2. Пропедевтический курс «программирование в начальной школе»	31
2.3. Программно-методическая поддержка курса	48
2.4. Апробация результатов исследования в школе	51
Вывод по главе 2	52
ЗАКЛЮЧЕНИЕ	53
Библиографический список	54
Приложение	56

ВВЕДЕНИЕ

В настоящее время в начальной школе действует Федеральный государственный образовательный стандарт второго поколения (ФГОС начального общего образования). Данный документ направлен на формирование личностных, метапредметных и предметных результатов обучения [26].

Тем не менее, стандартом не предусмотрен отдельный предмет «Информатика и ИКТ». Формирование ИКТ компетентностей учащихся организуется на всех предметах. Это требует от учителей новых образовательных методик и организации информационно-образовательной среды.

Проанализировав ФГОС начального общего образования, мы выявили, что целями освоения ИКТ в начальной школе являются следующие:

- овладение основами логического и алгоритмического мышления, пространственного воображения и математической речи, измерения, пересчета, прикидки и оценки, наглядного представления данных и процессов, записи и выполнения алгоритмов;
- приобретение первоначальных представлений о компьютерной грамотности;
- формирование первоначальных представлений о свойствах информации, способах работы с ней, в частности, с использованием компьютера;
- формирование информационной культуры ребенка.

Кроме того, в современном мире требуется все больше и больше квалифицированных разработчиков программного обеспечения, специалистов в области информационных технологий. При этом в школе отдается предпочтение пользовательскому подходу к обучению работы на компьютере, а не формированию алгоритмического мышления. Тем более, в основной школе нет системного подхода к обучению программированию. Объектом изучения становится сам язык программирования, нежели основы

разработки программ, которые затем можно применять в решении практических задач.

Таким образом, мы видим два противоречия:

1. Между требованиями стандарта начальной школы к формированию ИКТ компетенций учащихся и отсутствием в программе отдельного предмета «Информатика и ИКТ» в начальной школе.
2. Между потребностью в высококвалифицированных специалистах в области разработки программного обеспечения и содержанием курса информатики в основной школе.

Решение данных проблем может стать пропедевтика логико-алгоритмического компонента предмета, в том числе программирования, начиная с начальной школы. Пропедевтика является важным звеном в системе непрерывного образования детей.

Цель работы: изучить особенности пропедевтики программирования и разработать пропедевтический курс «Программирование в начальной школе».

Объект исследования: программирование в начальной школе.

Предмет исследования: процесс обучения программированию в программной среде Scratch в начальной школе.

В соответствии с целью работы были поставлены следующие **задачи:**

1. Изучить теоретические основы по проблеме исследования.
2. Разработать пропедевтический курс по программированию в среде Scratch для начальной школы.
3. Разработать программно-методическую поддержку пропедевтического курса в виде электронного пособия.

Гипотеза: если в программу начального общего образования, в качестве внеурочной деятельности, включить вопросы, связанные с пропедевтикой программирования, то это будет способствовать овладению учащимися основами логического и алгоритмического мышления.

ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ОСНОВА ПРОПЕДЕВТИКИ ИЗУЧЕНИЯ ПРОГРАММИРОВАНИЯ В НАЧАЛЬНОЙ ШКОЛЕ

1.1. Понятие «пропедевтика»

Рассмотрим, что означает понятие «пропедевтика»? Данное понятие рассматривается не только в педагогике, но и в философии, медицине, хореографическом искусстве.

Термин пропедевтика (от греч. *propaidéuō*, «подготовительное обучение») означает введение в искусство или в ту или иную науку, другими словами в простейшей форме краткое повествование искусства или науки, предварительный (вводный или подготовительный) курс, который предшествует более подробному изучению предмета. Различные аспекты пропедевтики информатики на разных уровнях образования рассматривались в работах М.П. Лапчик, И.Г. Семакин, Е.К. Хеннер, А.А. Кузнецов, Т.Б. Захарова, А.С. Захаров и др [16].

В период пропедевтического периода учащиеся получают начальные знания, умения, простые навыки культуры поведения. В то же время учитель использует пропедевтический курс для многостороннего исследования воспитанников и далее составляет первые психолого-педагогические характеристики. Эти характеристики в дальнейшем определяют план коррекционной работы с учащимся. Особое внимание учитель обращает на интерес ребенка к той или иной дисциплине, умение работать самостоятельно, особенность произвольного внимания, речи, навыка общения с другими детьми и учителем и т.д. Уроки в этот период организуются в форме игр, прогулок и экскурсий, заучивания стихотворений и песен, художественного повествования и чтения, лепки, рисования, ручного труда.

Так же в этот период школьники постепенно овладевают навыками чтения (без книги, на дидактическом материале), письма (изучение начертания букв) и счета (ознакомление с математическими терминами, цифрами, арифметическими действиями).

В это же время школьники учатся правилам поведения, как обращаться с учебными принадлежностями, как работать в коллективе. Учитель совместно с логопедом выделяют детей с различными дефектами речи, определяют план индивидуальных занятий и индивидуальный подход к каждому ребенку, нацеленный на снижение или даже в некоторых случаях преодоление изъяна. Школьный врач определяет у детей состояние зрения, слуха и далее устанавливает необходимость проведения медикаментозного лечения, или же некоторые рекомендации учителю, в том числе занятия по лечебной гимнастике. Все это определяет необходимые условия для успешного обучения всех учащихся определенного класса и индивидуально для каждого ребенка.

На базе учебных документов и общения с детьми учитель подводит итоги работы, определяет учебно-воспитательную работу и выделяет тех учеников, которые имеют необходимость в индивидуальном подходе на уроке или же при выполнении домашнего задания, а в некоторых случаях дополнительных индивидуальных занятиях.

1.2. Особенности изучения программирования в школе

Анализ ФГОС основного общего образования позволил выдвинуть требования к выпускникам по направлению «Программирование» [27]:

На базовом уровне выпускник должен:

- владеть навыками алгоритмического мышления и осознавать необходимость формального описания алгоритмов;
- уметь понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- знать основные конструкции программирования;
- уметь анализировать алгоритмы с использованием таблиц;

- владеть стандартными приемами написания на алгоритмическом языке программ для решения стандартных задач с использованием основных конструкций программирования и отладки таких программ;
- использовать готовые прикладные компьютерные программы по выбранной специализации.

На углубленном уровне выпускник должен:

- владеть универсальным языком программирования высокого уровня (по выбору), представлением о базовых типах данных и структурах данных;
- уметь использовать основные управляющие конструкции;
- владеть навыками и опытом разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- владеть элементарными навыками формализации прикладной задачи и документирования программ.

Тема «Алгоритмизация и программирование» является одной из самых сложных тем в курсе информатика. В настоящее время существует большое количество проблем, связанных с данным направлением.

Каждый учитель информатики встречается с большим количеством вопросов при построении изучения темы «Алгоритмизация и программирование»: как выстроить изложение материала, какие использовать методические разработки, в какой форме проводить занятия, какие составить практические задания, какой материал использовать учащимся при изучении и другие. Все вышеперечисленные вопросы появляются из-за отсутствия четко и в полном объеме изложенных учебно-методических материалов для изучения представленной темы.

В таблице 1 приведен результат анализа тем, связанных с программированием, по программам, представленных в федеральном перечне учебников с 2014/2015 по 2017/2018 учебные годы обучения: Босова Л.Л., Семакин И.Г., Угринович Н.Д., Поляков К.Ю. и Гейн А.Г.[14, 15, 21-25].

Анализ учебников по теме «Программирование»

Класс	Программа	Параграфы
Основная школа		
6 класс	Босова Л.Л.	§4.19. Графические исполнители (DRAW, LINE, CIRCLE).
9 класс	Босова Л.Л.	§2.4. Запись вспомогательных алгоритмов на языке Паскаль. §2.3.2. Разработка алгоритма методом последовательного уточнения для исполнителя Робот
	Семакин И.Г.	Глава 6. §35. Знакомство с языком Паскаль. §37. Программирование ветвлений на Паскаль.
	Угринович Н.Д.	§4.7. Графические возможности объектно-ориентированного программирования Visual Basic 2005.
Старшая школа		
10 класс	Семакин И.Г., углубленный уровень	§1.7.2. Алгоритмическая машина Тьюринга. §1.7.3. Алгоритмическая машина Поста. §4.3.1. Способы создания сайтов. Понятие о языке HTML.
11 класс	Поляков К.Ю., углубленный уровень	§31. HTML и XHTML. §32. Динамический HTML. §74. Язык VRML.
	Семакин И.Г., углубленный уровень	§2.2.1. Паскаль – язык структурного программирования. §2.4.2. Система программирования Delphi.

10-11 класс	Гейн А.Г.	§33. Знакомьтесь: исполнитель Паркетчик. §52. Лабораторная работа. Программирование на Ассемблере «Малютки».
----------------	-----------	---

Как видно из таблицы, авторы учебников используют для обучения программированию различные языки и среды программирования. Это приводит к тому, что с одной стороны, у учителя есть свобода выбора, с другой стороны, учителя вынуждены осваивать все многообразие языков программирования и связанных с ними сред программирования.

1.3. Особенности пропедевтического курса программирования в начальной школе

Согласно статистике, раздел «Программирование» является самой нелюбимой и самой трудной в освоении как школьниками, которые изучают информатику, так и учителями, которые преподают ее. Социологические исследования в данной тематике гласят, что всего лишь 5% пользователей ПК работают над написанием различных программ, а остальные 95% всего лишь пользуются результатами разработчиков, к тому же у детей нет мотивации и желания попасть в эти несчастные 5%. И даже жизненные истории известных программистов мира, таких как Павел Дуров, Марк Цукерберг, Евгений Касперский, Билл Гейтс, Стив Возняк и др., не вызывают большого интереса к программированию у школьников.

Не исключено, что в классе, возможно, есть несколько «вундеркиндов», которые выполняют задания быстрее всех, однако их желание программировать мотивировано лишь приобретением свободного времени для общения или отдыха, но ни как не написанием более сложной и интересной программы. То же самое и с олимпиадами по информатике. Зачастую очень трудно добровольно определить кандидата, поэтому

участвовать идет тот ученик, на которого по воле случая упал пристальный взгляд учителя.

К тому же неправильный выбор учителем подхода к обучению, приводит к тому, что у учеников резко пропадает интерес (на 3-4 занятии) к изучаемой теме и начинают внушать себе, что им это не дано, и они никогда не смогут понять это. И если бы завтра, допустим, раздел «Программирование» исключили бы из школьного курса, многие, по-видимому, облегченно вздохнули бы.

Безусловно, привлечь детей к теме «Программирования» можно и нужно. Как это сделать?

Во-первых, в выборе методов и форм обучения необходимо отталкиваться от возрастных особенностей школьников.

Как правило, возраст младших школьников охватывает период жизни от 6 до 11 лет. В этом возрасте способность восприятия и наблюдения внешней реальности еще не абсолютна, они воспринимают вещи окружающей среды не точно, подчеркивая при этом случайные особенности и признаки, по какой-либо причине заинтересовавшие их. Непроизвольный характер внимания школьников, заключается в том, что его можно легко отвлечь любым внешним раздражителем, который мешает процессу обучения. К тому же присутствует недоразвитость концентрации внимания, то есть учащиеся не способны удерживать внимание на определенном объекте долгое время. Память младших школьников имеет наглядно-образный характер, то есть они лучше запоминают внешние характеристики изучаемого предмета, чем их логическую сущность. Дети не в полной мере связывают отдельные части составляющего и с трудом представляют целостную структуру явления. В общем, запоминание носит механический характер, основанный на повторном повторении действия или сильном впечатлении от увиденного. В связи с этими особенностями, учителю необходимо правильно конструировать план занятия, разбивая урок на

отдельные этапы, характеризующиеся сменой деятельности (урок за партой, гимнастика, урок за ПК и т.д.).

Во-вторых, важной составляющей детей, обучающихся в начальной школе, являются психологические особенности. Детям, которые недавно ходили в садик, очень тяжело адаптироваться в новой роли ученика. Ведущей деятельностью теперь становится обучение, что представляет для ребенка огромный труд и требует внимания, дисциплины, волевых усилий учащегося. В связи с этим от учителя требуется большая фантазия в выборе форм и методов проведения занятий. Известно, что дети значительно лучше усваивают информацию, если она представлена в виде игры. Необходимо помнить что учащиеся в начальной школе – это новоиспеченные выпускники детских садов, которым очень трудно сидеть 40 минут и слушать нудную лекцию о программировании, алгоритмах и т.д. Наиболее продуктивной урок пройдет под названием, например, путешествие в страну «Алгоритмия» или соревнование по написанию алгоритма, где главным призом станет звездочка на обложку тетради.

Не только дети, но и взрослые, воспринимают графическую информацию, представленную в виде ярких картинок, схем и диаграмм куда лучше, чем сплошной текст. В связи с этим учителю информатики необходимо внимательно подходить к выбору программной среды. Главные условия подходящей программы: наглядность, мультимедийность и простота. В настоящее время существует большое количество различных программных сред и языков программирования, представленные на разный возраст и уровень подготовки учащихся. Главной задачей учителя является правильная подобранная программа и форма проведения занятия.

Еще одной проблемой является отсутствие отдельного предмета «Информатика и ИКТ» в некоторых учебных заведениях. В этом вопросе приходит на помощь внеурочная деятельность. Но, как правило, она располагается после основных уроков, когда дети довольно таки устали и с нетерпением ждут возвращения домой. По этой причине многие родители,

оберегая своих чад от «лишней загрузки» отказываются от внеурочной деятельности по информатике. Но как же формирование логики у детей?

В современной психологии отмечается существенное воздействие освоения основ алгоритмизации на формирование у обучаемых логического, алгоритмического и креативного мышления. Информатика в паре с математикой закладывает в образование основные проявления человеческого интеллекта: способность к действию, размышлению, обучению. Главную роль в курсе информатики играет формирование у обучаемых способности к выполнению действий на базе сформированного у него алгоритмической манеры мышления. Человек, который живет в современном информационном обществе, обязан владеть алгоритмическим мышлением.

Нельзя не отметить, что многие считают, главной задачей преподавателя информатики – это объяснить школьникам, как пользоваться современными информационными технологиями. Безусловно, это важно, но при овладении основ алгоритмизации формируется системно-информационная картина мира, умение выделять объекты, процессы и явления, а так же вырабатывается умение самостоятельно ставить задачи и решать их.

Все выше перечисленные особенности в какой-то степени влияют на планирование, подготовку и ход проведения занятия. Главной задачей учителя информатики начальных классов является особое внимание ко всем составляющим: возрастным, психологические особенности младших школьников, а так же время проведения внеурочной деятельности. Только все в совокупности может привести к продуктивной деятельности школьника во время внеурочной деятельности.

1.4. Программные среды и языки программирования для младших школьников

Учебные языки программирования – это языки программирования и среды программирования, предназначенные для пропедевтики программирования, алгоритмизации и развития логико-алгоритмического (операционного) мышления у детей.

На сегодняшний день существует множество различных языков программирования – начиная от самых простых (тех, на которых обучаются дети младших классов), и заканчивая такими мощными инструментами разработки, как C++, C#, Java, OpenGL и т.д. Однако взрослые языки программирования не подходят детям, которые только начинают изучать программирование, ведь у них еще не сформировано абстрактное мышление. Языки программирования, разработанные специально для младших школьников, учитывают их интеллектуальные и психофизические особенности.

Перечислим некоторые из них:

1. Lightbot – среда программирования, больше напоминающая логическую игру-головоломку для детей в возрасте 5-8 лет.

Пользователю необходимо запрограммировать движение робота на выполнение простой задачи – преодолевая преграды, дойти до цели. Программа формируется путём перетаскивания подходящих команд в ячейки памяти робота. Сложность игры состоит в том, что необходимо минимизировать размер программы, так как память робота фиксирована, и найти наиболее оптимальный маршрут перемещения робота.

Несмотря на простоту, с помощью LightBot можно формировать у ребёнка операционное мышление, а так же сформировать понятие программы, процедуры, циклы и др. Обзор форумов свидетельствует о том, что среда является популярной не только среди детей, но и среди взрослых программистов. К недостаткам данной среды относится подключение к сети Интернет [7].

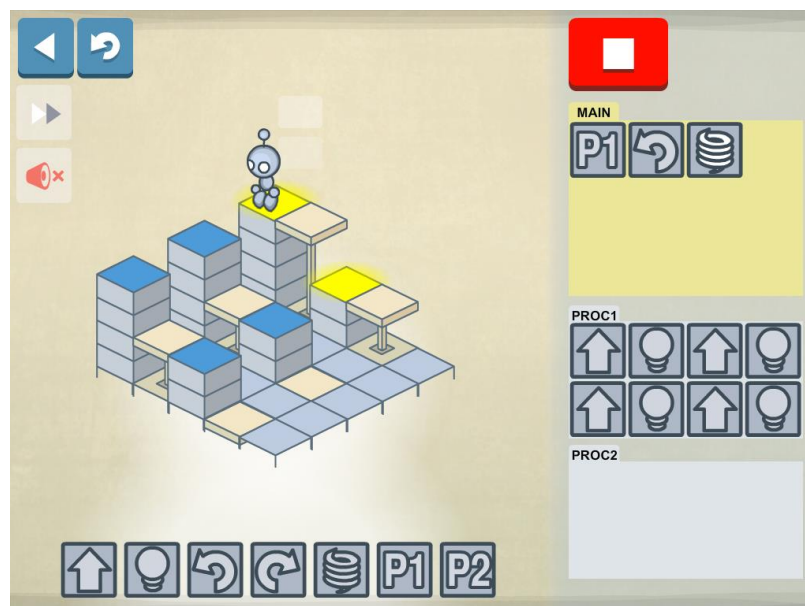


Рис. 1. Lightbot

2. RoboMind – это среда для обучения программированию, для детей 8-12 лет, реализованная с помощью простого скриптового языка. Программный продукт позволяет новичкам не только ознакомиться основам программирования, но и дает возможность разобраться в таких сферах, как искусственный разум и робототехника.

Персонажем, выполняющим действия алгоритма, является робот, который может перемещаться по двухмерной карте, содержащей различные предметы, "осматривать" предметы вокруг, отмечать пройденный путь. Программа также ориентирована на изучение языка программирования Robo для использования основных команд управления движения робота. Для домашнего использования программа распространяется свободно, для других целей доступна платная версия.

Начиная с версии 4.0, появилась возможность экспортировать скрипты RoboMind в некоторых роботов реального мира, например, робот Lego Mindstorms NXT 2.0 поддерживает данную возможность. К недостаткам можно отнести использование встроенного языка Robo, который не используется в других программных средах [10].

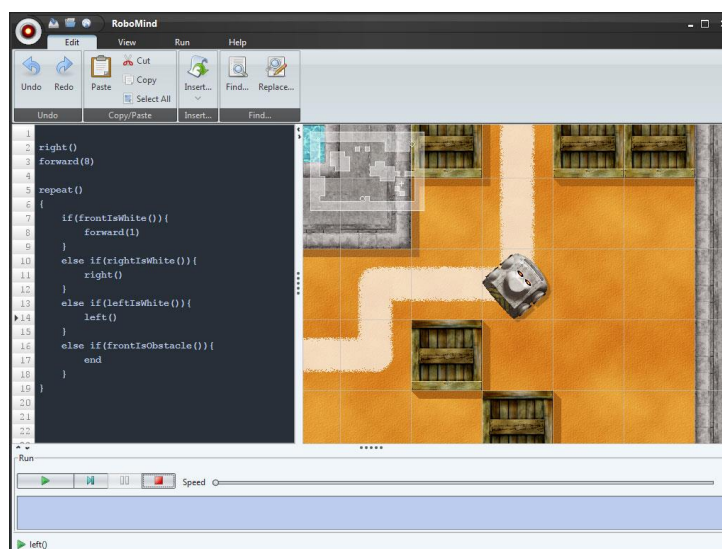


Рис. 2. RoboMind

3. Karel, Karel ++, Karel J. Robot – среды программирования для начинающих в возрасте 10-14 лет, используются для составления программ управления «роботом». В первой версии среды Karel для управления роботом создан свой собственный язык программирования Karel. Однако в следующей версии Karel ++ программисты использовали язык C++, в версии Karel J — язык Java, а в Robot — язык Python. Это позволило разработчикам объединить простой и доступный интерфейс в изучении сложных понятий языков высокого уровня. Работа с ней может быть началом к изучению многих языков программирования. Программа находится в свободном доступе, однако есть небольшой минус – отсутствие русскоязычного интерфейса [6].



Рис. 3. Karel

4. Guido van Robot – это язык программирования, использующий простой синтаксис, чтобы помочь учащимся понять сущность циклов и процедур. Самым большим достоинством данной программной среды является мгновенная визуальная обратная связь. Словом, это интерактивный вводный язык программирования, который отлично подходит для изучения базовых концепций программирования, применимых на любом языке высокого уровня.

Суть GvR в том, что это робот в виде треугольника на экране, который перемещается по миру, содержащий улицы и проспекты, стены и звуки, которые Guido может собирать или устанавливать. Его действия полностью руководствуются программой, написанной пользователем [5].

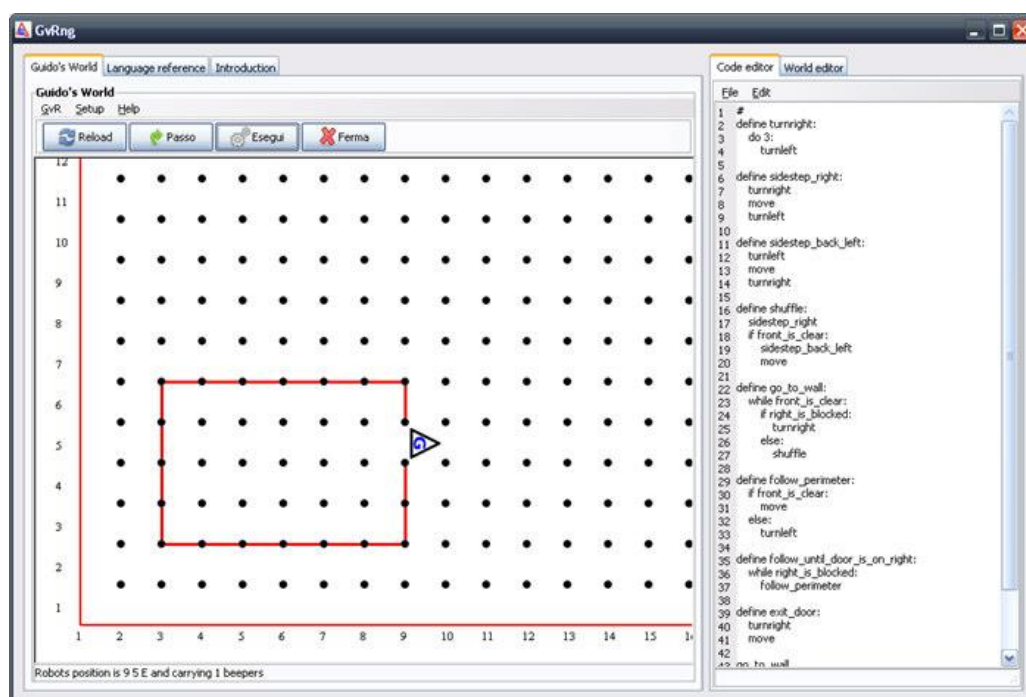


Рис. 4. Guido van Robot

5. Little Wizzard – среда программирования для детей начальной школы, предназначенная для изучения основных понятий программирования. Используя для разработки программ только мышь, дети приобретают возможность не только составлять программы, но и знакомиться с такими понятиями как переменная, выражение, ветвление, условие и логические блоки. Процесс программирования очень прост, так как элементы языка представлены в виде интуитивно-понятных символов [8].

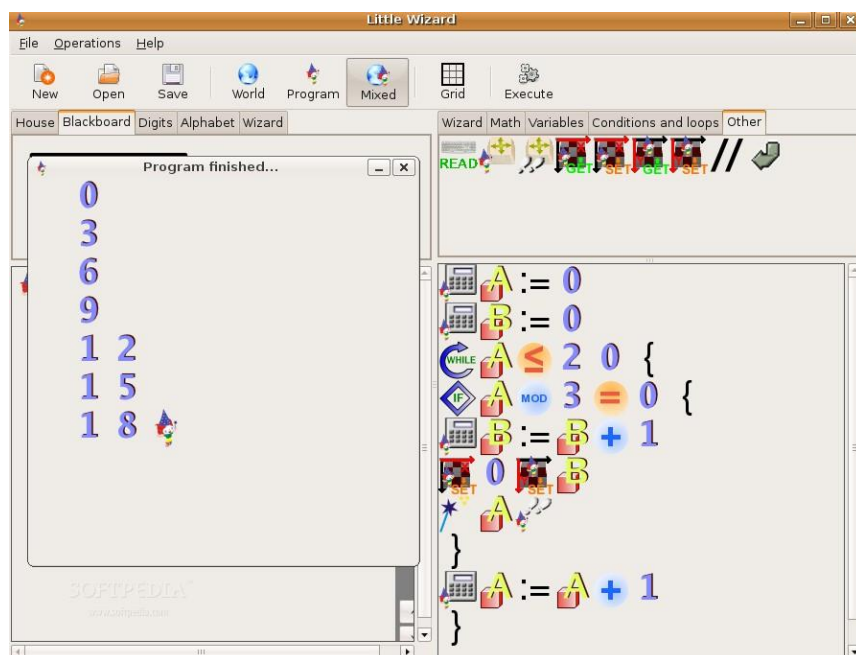


Рис. 5. Little Wizzard

6. Alice – среда программирования, предназначена для детей 8 – 12 лет. Это одна из самых увлекательных программ для наглядного обучения объектно-ориентированному программированию. Alice это своего рода интерактивная среда разработки, в которой, работая, можно в игровой манере создавать, анимировать, изменять, а так же программировать поведение трёхмерного мира, объектов и персонажей в нём, настраивать камеру и т.д.

Управление программой представлено в двух режимах: визуальный режим и режим редактирования кода. В первом режиме можно настраивать сцену, настраивать различные ракурсы камеры, добавлять 3D модели (программа включает довольно большую библиотеку животных, людей, транспортных средств, различных объектов окружающего мира и т.д.) и собирать композицию. Каждая добавленная модель становится объектом к которой в дальнейшем пишется алгоритм действий.

В режиме редактирования кода можно писать алгоритмы, создавать процедуры, а так же изменять классы. Исполнение алгоритмов протекает последовательно, буквально, так же как и в интерпретируемых языках программирования.

Благодаря русскоязычному интерфейсу, освоить данную программную среду не составит большого труда. К тому же в Интернет есть огромное количество готовых проектов и обучающего материала [1].

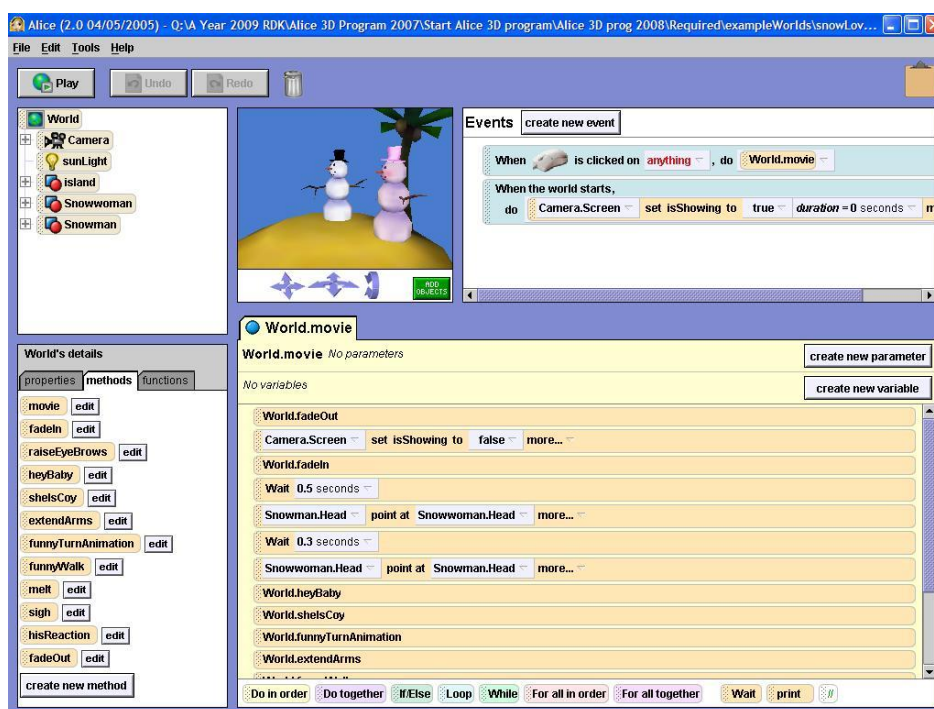


Рис. 6. Alice

7. Greenfoot – среда программирования на Java, которая выступает в роли конструктора игр. Программирование в данной среде рекомендуется детям в возрасте 10 – 14 лет.

В среде можно создавать свой собственный мир, добавлять актеров, изменять расположение элементов на локации, программировать их логику при помощи кнопок и различных визуальных диалоговых окон. Интерфейс прост, удобен и имеет все необходимое для удобной разработки.

Главными недостатками являются: англоязычный интерфейс и Java-зависимость [4].

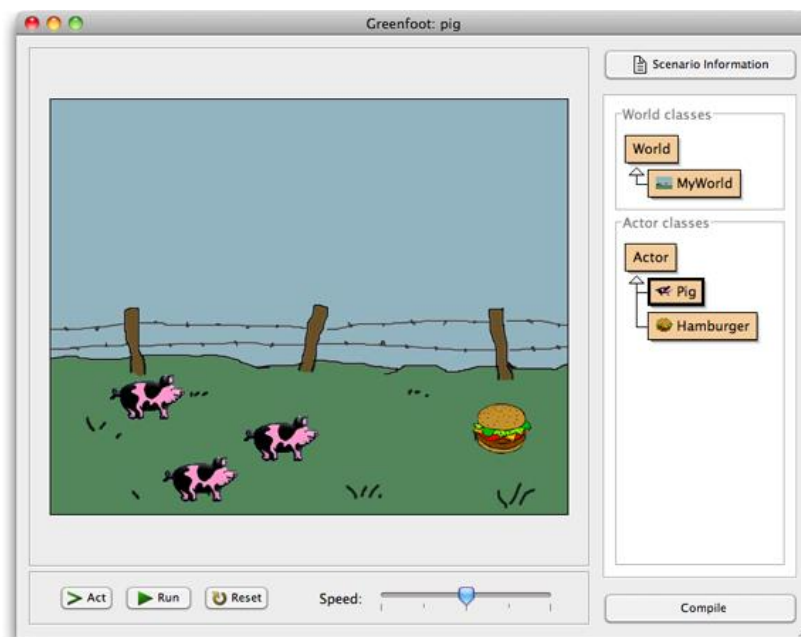


Рис. 7. Greenfoot

8. Peter – средство для визуального программирования; главное достоинство: простота и быстрота конструирования приложений под управлением операционной системы Windows. Написание программы немного напоминает создание головоломки из отдельных предметов: разработка графической презентации с созданием программного алгоритма для управления элементами.

9. Logo – язык программирования, предназначенный для детей дошкольного и младшего школьного возраста. Данный язык используется при работе в системе ЛогоМиры. С черепашкой или с любым другим персонажем, выбранным ребенком, быстрее понять основную логику работы программ, из-за наличия «направления», которое можно изменять. Обучение Черепашки каким-либо действиям приводит ребенка к осмыслению своих действий и мыслей, что создает процесс обучения программированию более эффективным.

Главный недостаток: отсутствие бесплатной версии, что приводит к значительному сокращению учебных заведений, которые могут себе позволить приобрести лицензию на данный продукт [9].

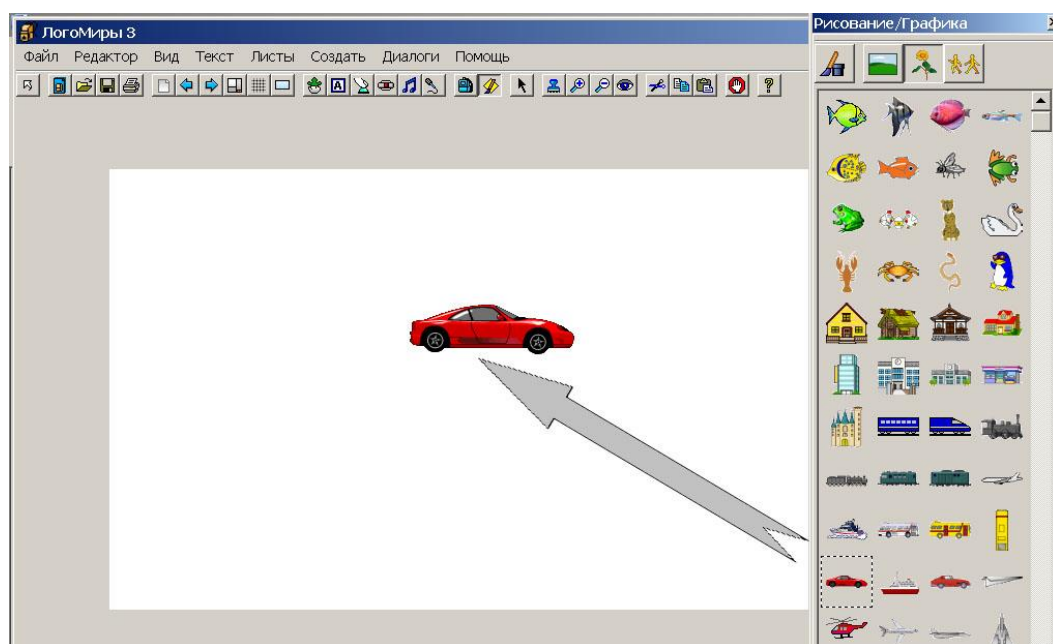


Рис. 8. Logo

10. Etoys – компьютерная среда, для детей в возрасте от 9 до 12 лет. Имеет мощные мультимедийные возможности для реализации программ – сценариев и работает на различных платформах. Основные идеи Etoys – «все окружающее - объекты», программа – это не текст, который компилируется, а затем выполняется, как в других учебно-развивающих средах, а возможность общения с этими объектами и изменение их. Etoys гарантирует простую и одновременно мощную среду исследования с методологией объектной модели, для различных объектов, сформированных учащимися, которые включают двумерные и трехмерные изображения, текст, html текст, звуки, и видео, а так же возможность работать с другими пользователями Etoys в режиме реального времени по локальной сети или по сети Интернет. Программа не только помогает детям изучить понятия программирования, но и используется для моделирования физических явления, решения математических задач и т.д. Etoys многоязычная программная среда, которая пользуется большим спросом в таких странах как США, Юной Америки, Японии, Индии, Корее но к сожалению в России данная программа вообще не популярна [3].

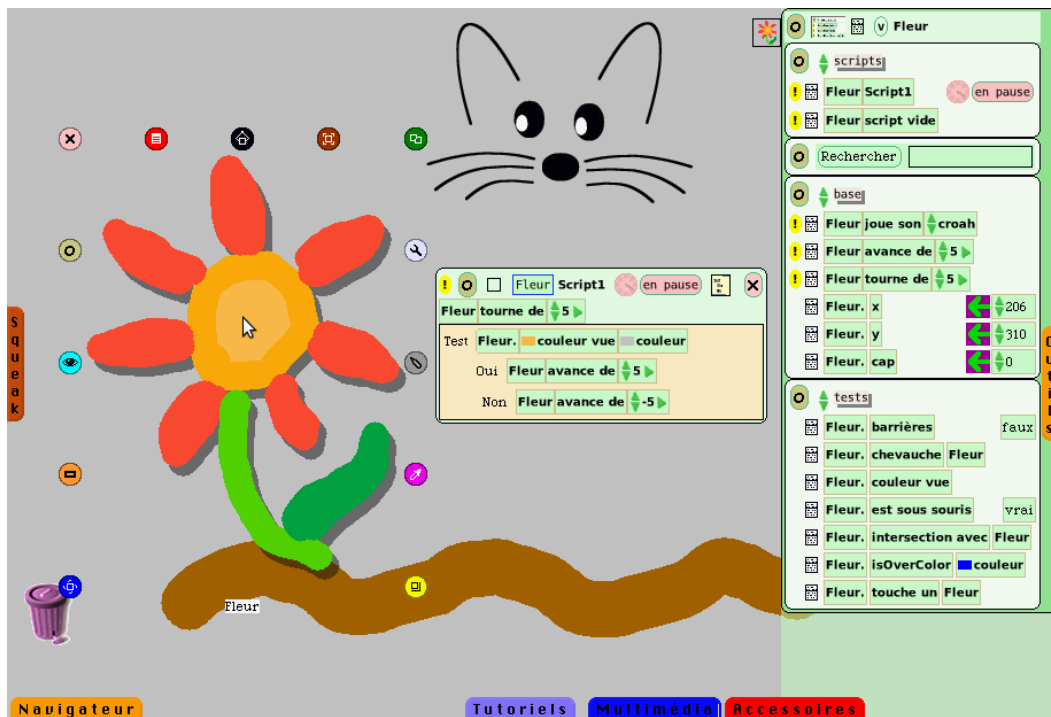


Рис. 9. Etoys

11. Baltie – учебная графическая среда и средство программирования для детей 6 – 16 лет. Она не требует никаких первоначальных знаний, что очень привлекает большое количество людей из разных стран для изучения программирования в учебных заведениях. Программа предполагает программирование спрайта-волшебника при помощи набора иконок из библиотеки. Набор иконок включает: изменение движения спрайта, цифры, логические и математические функции, управление координатами, добавление графики и анимации, управление структуры (циклы, процедуры), пользовательский ввод, назначение переменных, преобразование типов данных и работа со строками, работа с файлами. Последняя версия Baltie поддерживает язык C#. Данная программная среда используется во многих странах, за исключением России [2].

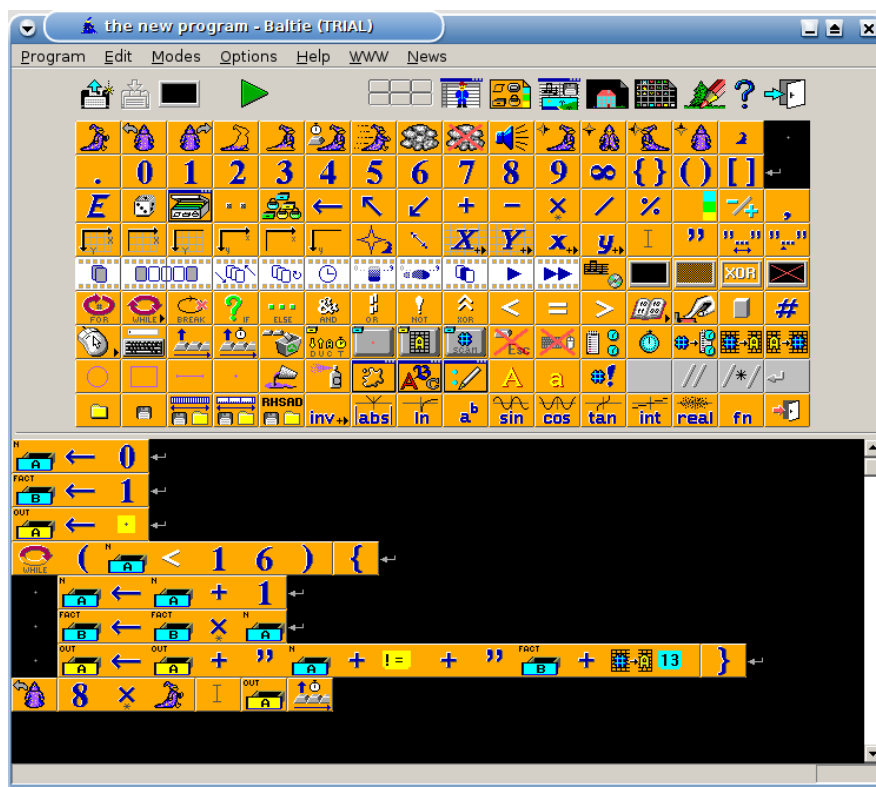


Рис. 10. Baltie

12. Squeak – открытая, инновационная среда для программирования, диалект языка Smalltalk (для детей 9 – 12 лет). Программа имеет широкий диапазон возможностей: начиная от мультимедийных приложений и различных обучающих платформ (Scratch и Etoys) до создания веб-серверов. Код программы выполняется (компилируется) «виртуальной машиной» Squeak, что позволяет без каких-либо затруднений перенести код программы в любую среду операционной системы.

К недостаткам можно отнести сложность установки и наличие платных компонентов [12].

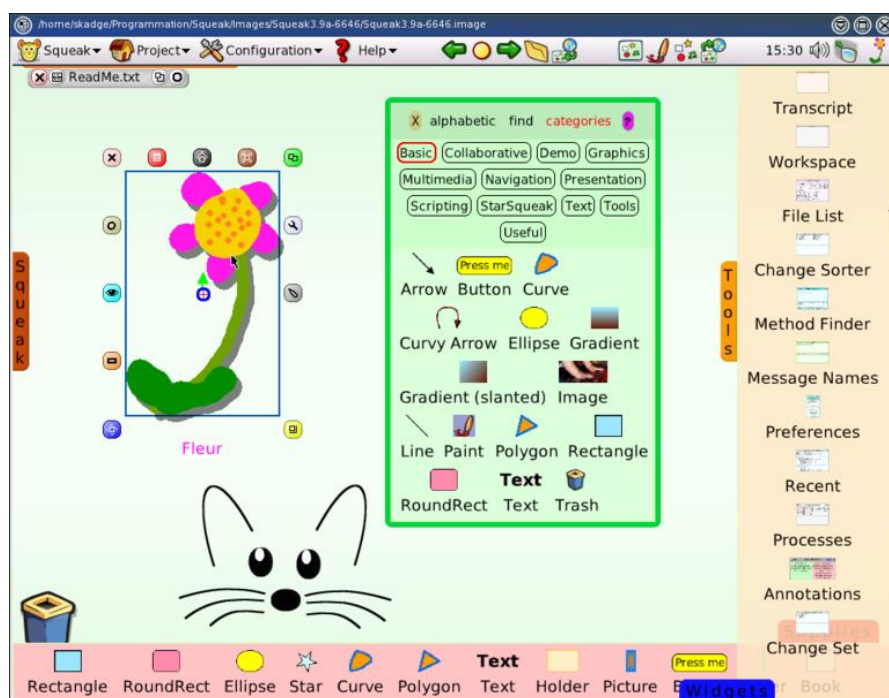


Рис. 11. Squeak

13. Scratch – визуальная объектно-ориентированная среда программирования, разработанная для обучения детей в возрасте 6 – 14 лет. Scratch создан как продолжение языков программирования Squeak и Logo. Он подойдет тем детям, которым уже не интересно выполнять привычные и повторяющиеся алгоритмы. В Scratch можно создавать интерактивные проекты, составляя блоки алгоритмов и присваивая их каждому объекту. Программа включает большую библиотеку животных, героев, различных предметов на любой вкус. Основная цель – познакомить детей с понятиями программирования в ходе создания проектов. Существует большое количество бесплатных методических поддержек, которые помогут ребенку при большом желании освоить данную программу самостоятельно, без участия родителей или учителей. Так же существует международное сообщество пользователей Scratch, что дает возможность не только создавать проекты, но и делиться новыми идеями и разработками по всему миру [11].

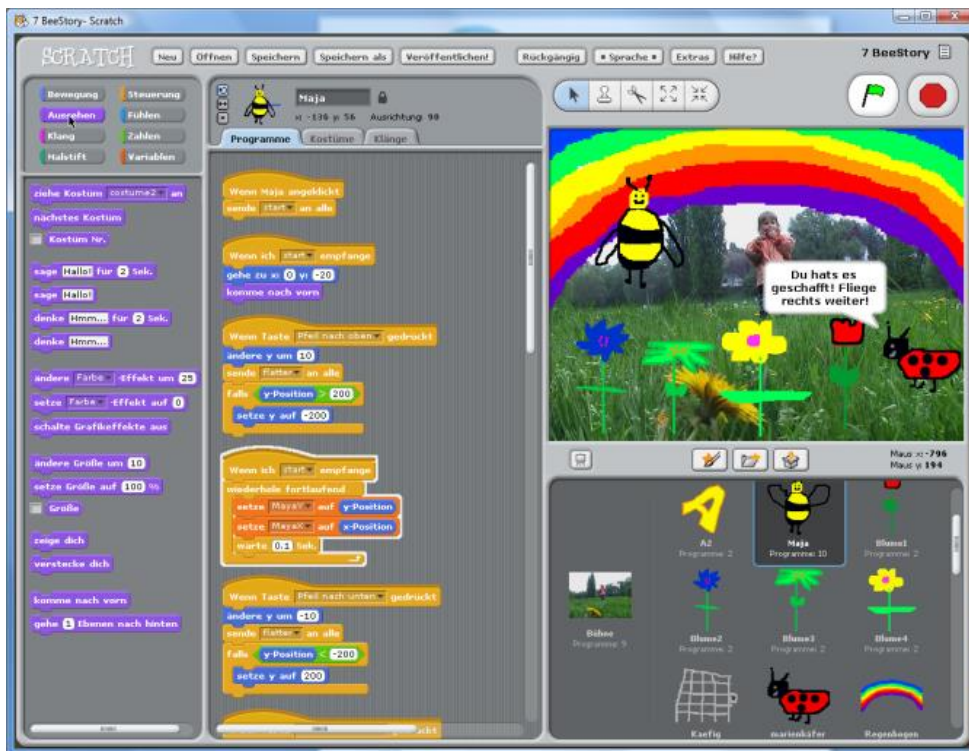


Рис. 12. Scratch

Из анализа программных сред и языков программирования можно сделать вывод, что у каждой программы есть свои достоинства и недостатки. Большой выбор программ предоставляет большие возможности для изучения программирования в игровой форме.

Выводы по Главе 1

В данной главе были рассмотрены основные теоретические сведения пропедевтики изучения программирования в начальной школе, обоснованы возрастные и психологические особенности младших школьников.

В ходе данной главы был проанализирован ряд программных сред и языков программирования для изучения темы «Программирование» в начальной школе: Lightbot, RoboMind, Karel, Karel ++, Karel J. Robot, Guido van Robot, Little Wizzard, Alice, Greenfoot, Peter, Logo, Etoys, Batlie, Squeak, Scratch.

Рассмотренный в первой главе теоретический минимум служит основной составляющей разработки пропедевтического курса «Программирование в начальной школе».

ГЛАВА 2. РАЗРАБОТКА КУРСА ВНЕКЛАССНЫХ ЗАНЯТИЙ «ПРОГРАММИРОВАНИЕ В НАЧАЛЬНОЙ ШКОЛЕ»

2.1. Анализ нормативных документов

Федеральные государственные образовательные стандарты (далее ФГОС) включают в себя комплекс требований, обязательных при реализации образовательными учреждениями, имеющими государственную аккредитацию, основных образовательных программ таких как: начальное общее, основное общее, среднее (полное) общее, начальное профессиональное, среднее профессиональное и высшее профессиональное образование.

Федеральные государственные образовательные стандарты гарантируют:

- 1) целостность образовательного пространства Российской Федерации;
- 2) построение единой содержательной линии основных образовательных программ начального общего, основного общего, среднего (полного) общего, начального профессионального, среднего профессионального и высшего профессионального образования.

Воспитание считается одной из главных составляющих образовательного процесса наравне с обучением. Важная педагогическая задача – воспитание высоконравственного, ответственного, компетентного и инициативного гражданина Российской Федерации. В соответствии с федеральным государственным образовательным стандартом начального общего образования (далее ФГОС НОО) основная образовательная программа реализуется образовательным учреждением, в том числе и через внеурочную деятельность [26].

Внеурочная деятельность – это не только составная часть учебно-воспитательного процесса, но и одна из форм организации свободного времени учеников, нацеленная на своеобразную помощь ребенку в преодолении каких-либо затруднений в новой учебной деятельности и создание учебной мотивации, интереса в данной дисциплине [26].

При осуществлении образовательным учреждением внешкольной деятельности учащихся могут использоваться возможности образовательных организаций дополнительного образования детей. Чтобы процесс обучения был непрерывным, во время каникул необходимо возможности организаций отдыха детей и их оздоровления, это могут быть лагеря, летние школы и т.д. В случае с одаренными детьми или же детьми с ограниченными возможностями, учебные планы можно разрабатывает совместно с самими детьми или же с родителями (законными представителями).

Цель внеурочной деятельности: создание условий для раскрытия и развития своих увлечений и интересов на базе свободного выбора, знакомство с культурными традициями, нравственными ценностями, создание условий для интеллектуального и физического развития детей.

Существуют некоторые принципы организации внеурочной деятельности:

1. Принцип гуманистической направленности. Учитываются в полной мере потребности, интересы и желания детей, при организации вне учебной деятельности, кроме того реализуются условия для формирования у учащихся умений и навыков самопознания и самореализации.

2. Принцип системности. Формируется система внеурочной деятельности школьников, в которой устанавливаются взаимосвязи между:

- всеми соучастниками образовательного процесса: учащимися, педагогами, родителями;
- урочной и внеурочной деятельностью;
- основными компонентами образовательного процесса: целевым, содержательно-деятельностным и оценочно-результативным;
- региональной, муниципальной, школьной, классной, индивидуальной системами обучения и дополнительного образования школьников.

3. Принцип вариативности. Определяется несколько направлений, форм, методов и способов организации внеурочной деятельности, с помощью которых дети имеют полную свободу выбора и добровольное участие в какой-либо деятельности.

4. Принцип креативности. Задачи учителя – во внеурочное время развивать творческие способности детей, создавать благоприятную среду для занятий по индивидуальным и коллективным программам.

5. Принцип успешности и социальной значимости. Стремление учителя направлено на удовлетворение потребности ребенка в достижении успеха и самореализации. Принципиально, чтобы результаты учебной деятельности ребенка были не только лично ценными, но и значимыми для окружающих, к примеру, одноклассников, родителей, учительского коллектива, организации школы и т.д. Время, отведенное на внеурочную деятельность, составляет не более 1350 часов за 4 года обучения [26].

Перечисленные принципы предполагают различные способы организации внеурочной деятельности, которые тесно связаны с основным образованием.

Внеурочная деятельность в ФГОС НОО организуется по пяти направлениям развития личности:

- духовно-нравственное;
- общекультурное;
- общеинтеллектуальное;
- социальное;
- спортивно-оздоровительное.

Для достижения цели определяются следующие задачи:

1. Формирование личности каждого младшего школьника в условиях креативной и творческой среды, посредством самореализации и самоопределения.

2. Формирование инициативности детей.

3. Воспитание патриотизма и уважения к народным ценностям.
4. Популяризация здорового образа жизни.
5. Подведение школьников на самостоятельное выполнение заданий.
6. Привлечение учащихся к организации научного труда.
7. Формировать интерес к научно-исследовательской деятельности.

Содержание внеурочной деятельности

Методисты Д.В. Григорьев и П.В. Степанов считают, что во внеурочной деятельности целесообразно включать следующие виды деятельности:

1. Игровую деятельность.
2. Познавательную деятельность.
3. Досугово-развлекательную деятельность.
4. Проблемно-ценностное общение.
5. Художественное творчество.
6. Социальное творчество.
7. Трудовую деятельность.
8. Спортивно-оздоровительную деятельность.
9. Туристско-краеведческую деятельность.

Ожидаемые результаты внеурочной деятельности.

Наиболее сложная задача – выявление продуктивности внеурочной деятельности. Однако существуют различные методы, которые упрощают анализы результатов (беседа, тестирование, анкетирование, олимпиада, конкурсы, соревнования, анализ портфолио ребенка и др.).

В Федеральном стандарте представлены следующие ожидаемые результаты обучения во внеурочное время:

- создание оптимальных условий для развития и отдыха учащихся;
- творческое самоопределение и самореализация детей
- формирование навыков коллективной и организаторской деятельности;

- психологический комфорт и социальная защищенность каждого ребенка;
- осуществление игрового сюжета в учебной деятельности детей.

Предметная область «Математика и информатика»

В Федеральном государственном образовательном стандарте начального общего образования отражены требования к предметной области «Математика и информатика», обеспечивающие изучения темы «Программирование». Рассмотрим основные положения данной области.

Метапредметные результаты:

- Умение вводить текст с клавиатуры, записывать (фиксировать) измеряемые величины в цифровой форме и анализировать изображения, звуки.
- Готовность выступать с аудио-, видео- и графическим сопровождением.

Предметные результаты:

- овладение основам логического и алгоритмического мышления, пространственного воображения;
- умение действовать в соответствии с алгоритмом и строить простейшие алгоритмы;
- умение представлять, анализировать и интерпретировать данные;
- приобретение первоначальных представлений о компьютерной грамотности.

Таким образом, в результате изучения предметной области «Математика и информатика» учащиеся получают представление об основных понятиях логико-алгоритмического компонента программирования.

2.2. Пропедевтический курс «программирование в начальной школе»

Одним из фундаментальных компонентов в информатике является программирование, особенно для подготовки любого ИТ-специалиста. К тому же первоначальные сведения о программировании включены в образовательные программы гуманитарных специальностей, таких как социология и психология. Это означает, что у разработчиков программ и стандартов образования нет сомнения в необходимости изучения раздела «Программирование» даже гуманитариями, пусть и на начальном уровне. Но чтобы обеспечить высокие результаты изучения программирования в старших классах или в высших учебных заведениях, необходимо заложить прочную основу на первоначальном этапе обучения, а именно в начальной школе. Тем самым это доказывает актуальность пропедевтического курса «программирование в начальной школе».

Предлагаемый курс предназначен для организации внеурочной деятельности в начальной школе, при поддержке программной среды Scratch. Программа составлена с учетом возрастных особенностей учащихся младшего школьного возраста и предусмотрена на работу в учебном компьютерном классе. Занятия проводятся по группам (10 – 15 человек), рассчитанные на 30 минут 1 раз в неделю.

Срок освоения программы: 1 семестр. Объем курса: 17 уроков.

Цель курса: формирование основ логико-алгоритмического мышления и программирования, а так же овладение школьниками навыками работы на компьютере.

Задачи курса:

Образовательные:

- познакомить с техникой безопасности при работе с ПК, а так же при нахождении в компьютерном кабинете;
- сформировать первоначальные знания о разделе «Программирование»;
- научить применять полученные знания при решении задач;

- познакомить с программной средой Scratch;
- получить практические навыки программирования в среде Scratch;
- научить основным навыкам и умениям использования компьютерных устройств;
- научить способам разработки, тестирования и отладки простых программ;

Воспитательные:

- воспитывать умение работать в коллективе, группе, паре, разумно распределяя обязанности;
- воспитывать уважительное отношение к сверстникам и учителям;

Развивающие:

- формировать информационную и алгоритмическую культуры;
- развивать творческие способности учащихся;
- развивать самостоятельность и ответственность;
- развивать логику и причинно-следственные связи;
- развивать память, внимание, способность к анализу и концентрации

Требования к результатам освоения курса:

Сформулированные задачи освоения пропедевтического курса обеспечивает выполнение следующих требований к уровню подготовки учащихся, окончивших данный курс:

Личностные результаты:

- мотивы познания и творчества, ответственность и любопытность, большой познавательный кругозор;
- желание и стремление учеников к саморазвитию и самореализации;
- готовность к развитию творческих способностей, за счет становления логического и алгоритмического мышления;
- интерес к предмету «Информатика и ИКТ» и желание применять полученные умения и навыки, не только на других учебных дисциплинах, но и в повседневной жизни;

- готовность к увеличению «багажа знаний» в области информационно-коммуникационных технологий;
- готовность в принятии решений с некоторой долей ответственности;
- готовность к достижению цели как индивидуально, так и коллективно;
- осознание важности изучения предмета «Информатика и ИКТ» в условиях информатизации общества;
- умение выделять основную мысль, за счет анализа информации и критического оценивания;
- осознание основных гигиенических и технических правил безопасного использования средств ИКТ.

Предметные результаты:

- понимание основных терминов раздела «Программирование», таких как «информация», «алгоритм», «данные», «программа»;
- умение формировать и отличать линейные, разветвляющиеся и циклические алгоритмы;
- умение формально выполнять алгоритмы, которые написаны с использованием условных операторов и циклов;
- умение писать простейшие программы в среде Scratch.

Метапредметные результаты:

- умение организации своей собственной учебной деятельности и рационального распределении времени;
- определение промежуточных целей и задач, направленных на достижение конечной цели;
- прогнозирование конечных результатов;
- самоконтроль и соотнесение полученных результатов с поставленными задачами и целями;
- готовность к коррекции данных при обнаружении ошибок;
- использование методов информационного поиска и выделение важной информации;

- визуализация и представление информации;
- выбор более эффективных методов решения задач;
- готовность к дружественному и продуктивному сотрудничеству в группе или в паре;
- использование средств ИКТ не только в учебной деятельности, но в повседневной жизни.

Способы проверки знаний и умений учащихся:

- проекты;
- итоговая контрольная работа;
- отзывы учителей и родителей;

В таблице 2 представлено учебно-тематическое планирование курса:

Таблица 2

Тематическое планирование курса

№	Тема урока	Количество часов		
		Всего	Теория	Практика
1	Введение. Техника безопасности.	1	1	-
2	Алгоритмы. Линейный алгоритм.	1	1	-
3	Блок – схемы. Условный алгоритм.	1	1	-
4	Решение задач.	1	1	-
5	Блок – схемы. Циклический алгоритм.	1	1	-
6	Решение задач.	1	1	-
7	Знакомство со Scratch.	1	-	1
8	Анимация имени.	1	-	1
9	Исполнитель Scratch. Цвет и размер пера.	1	-	1
10	Основные инструменты графического редактора программной среды Scratch.	1	-	1
11	Рисование исполнителем Scratch линий. Линейный алгоритм.	1	-	1

12	Рисование квадратов и прямоугольников линейно.	1	-	1
13	Цикл. Исполнитель рисует линии и фигуры.	1	-	1
14	Исполнитель Scratch рисует несколько линий и фигур. Конечный цикл.	1	-	1
15	Циклический алгоритм. Проект узор.	1	-	1
16	Рисование пунктирных линий. Проект светофор.	1	-	1
17	Создание мультфильма.	1	-	1
Итого:		17	7	10

Поурочное планирование курса.

Урок 1. Введение. Техника безопасности.

Тип урока: урок усвоения новых знаний.

Цель урока: познакомить учащихся с техникой безопасности в компьютерном классе и при работе с ПК.

Задачи:

Образовательная: систематизировать знания по технике безопасности;

Развивающая: развить эстетические чувства; научить соблюдать дисциплину и бережно отношение к окружающим предметам;

Воспитательная: развить коммуникабельность и сплоченность в коллективе; развить творческие способности.

Основные понятия: безопасность, техника безопасности, ПК.

Методическая рекомендация: Учитель в игровой форме и с помощью технической поддержки рассказывает детям о технике безопасности при работе с ПК и при нахождении в компьютерном кабинете. В конце урока проводится викторина, направленная на проверку усвоенной информации.

Вопросы для контроля:

1. Что категорически запрещается делать в кабинете информатики?
2. Что следует проверить перед работой с ПК?
3. Что делать, если компьютер неисправен?

Урок 2. Алгоритмы. Линейный алгоритм.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условия для формирования у обучающихся представление о понятии «Алгоритм».

Задачи:

Образовательная: познакомить учащихся с понятием «алгоритм», «линейный алгоритм»; научить составлять и выполнять алгоритмы.

Развивающая: развивать логическое мышление учащихся, память, внимание, воображение, познавательную активность, способность быстро воспринимать информацию.

Воспитательная: воспитывать у учащихся самостоятельность, активность, интерес к предмету, правила поведения.

Основные понятия: алгоритм, линейный алгоритм, команда.

Методические рекомендации: Учитель организует урок в виде путешествия, где учащиеся сами ставят цели и задачи урока.

Вопросы для контроля:

1. Что такое алгоритм?
2. Какой алгоритм называется линейным?
3. С какой команды начинается алгоритм?
4. Какой командой заканчивается алгоритм?

Урок 3. Блок – схемы. Условный алгоритм.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условия для формирования у учащихся представление об условных алгоритмах и построении блок-схем.

Задачи:

Образовательная: : дать первоначальное представление об условных алгоритмах, блок-схемах; научить формировать условные алгоритмы и формально исполнять их; научить выделять главную информацию, анализировать содержание алгоритма; научить составлять блок-схемы условного вида.

Развивающая: развивать творческие способности детей; развивать умение прогнозировать результаты.

Воспитательная: формировать стремление в приобретение новых знаний; воспитывать коммуникативные качества учащихся;

Основные понятия: блок-схема, алгоритм, условный алгоритм.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Дайте определение блок-схемы.
2. Какой алгоритм называется условным?
3. С помощью каких фигур строить блок-схема условного алгоритма?
4. Как начинается любой алгоритм?
5. Как заканчивается любой алгоритм?

Урок 4. Решение задач.

Тип урока: урок применения предметных умений.

Цель урока: создать условия для практического применения знаний и умений по теме условные алгоритмы.

Задачи:

Образовательная: научить решать задачи с использованием циклических блок-схем.

Развивающая: развивать способность к самостоятельному решению поставленной задачи; формировать логическое мышление, развивать память;

Воспитательная: способствовать повышению интереса к предмету; воспитывать у учеников коммуникабельность, аккуратность и терпение; объективное оценивание своих достижений и сопоставление их с целями.

Основные понятия: блок-схема, цикл, циклический алгоритм.

Основные понятия: блок-схема, алгоритм, условный алгоритм.

Методические рекомендации: На данном уроке учитель демонстрирует способы решения задач с помощью условных алгоритмов. При необходимости организует не только коллективную работу, но и индивидуальную. В конце урока формирует некоторые вопросы, демонстрирующие результаты освоения темы.

Вопросы для контроля:

1. Дайте определение блок-схемы.
2. Какой алгоритм называется условным?
3. С помощью каких фигур строить блок-схема условного алгоритма?
4. Как начинается любой алгоритм?
5. Как заканчивается любой алгоритм?

Урок 5. Блок-схемы. Циклический алгоритм.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условия для формирования представления о циклических алгоритмах и построении блок-схем.

Задачи:

Образовательная: дать первоначальное представление о циклических алгоритмах, блок-схемах; научить формировать циклические алгоритмы и формально исполнять их; научить выделять главную информацию, анализировать содержание цикла; научить составлять блок-схемы циклического вида.

Развивающая: развивать такие компоненты как память, логическое мышление, анализ; научить школьников формировать свое собственное мнение и аргументируя, высказывать его.

Воспитательная: вырабатывать познавательный интерес к теме курса; развивать коммуникативные качества, такие как, умение слушать, высказывать свое мнение, и обосновывать его.

Основные понятия: блок-схема, цикл, циклический алгоритм, цикл с предусловием, цикл с постусловием.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Что такое цикл?
2. Чем отличается циклический алгоритм от других видов алгоритмов?
3. Какие фигуры используются при формировании циклического алгоритма?
4. Как начинается любой алгоритм?

Урок 6. Решение задач.

Тип урока: урок применения предметных умений.

Цель урока: создать условия для практического применения знаний и умений по теме циклические алгоритмы.

Задачи:

Образовательная: научить решать задачи с использованием циклических блок-схем.

Развивающая: развивать способность к самостоятельному решению поставленной задачи; формировать логическое мышление, развивать память;

Воспитательная: способствовать повышению интереса к предмету; воспитывать у учеников коммуникабельность, аккуратность и терпение; объективное оценивание своих достижений и сопоставление их с целями.

Основные понятия: блок-схема, цикл, циклический алгоритм.

Методические рекомендации: На данном уроке учитель демонстрирует способы решения задач с помощью циклических алгоритмов. При необходимости организует не только коллективную работу, но и индивидуальную. В конце урока формирует некоторые вопросы, демонстрирующие результаты освоения темы.

Вопросы для контроля:

1. Что такое цикл?
2. Чем отличается циклический алгоритм от других видов алгоритмов?
3. Какие фигуры используются при формировании циклического алгоритма?
4. Как начинается любой алгоритм?

Урок 7. Знакомство со Scratch.

Тип урока: урок усвоения новых знаний.

Цель урока: познакомить учащихся с интерфейсом программы Scratch.

Задачи:

Образовательная: дать представление программы Scratch, определение скрипта и спрайта.

Развивающая: развивать у обучающихся интерес к предмету, умения и навыки при работе с программой Scratch.

Воспитательная: воспитывать трудолюбие, дисциплинированность, доброжелательное отношение друг другу и умение преодолевать трудности.

Основные понятия: интерфейс программы, спрайт, скрипт, блок,

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Каким цветом представлен блок «контроль»?
2. Можно ли изменить костюм персонажа и как это сделать?

3. Что такое скрипт?
4. Дайте определение спрайта.
5. Сколько различных командных блоков существует в Scratch?
6. Как можно поменять фон?

Урок 8. Анимация имени.

Тип урока: урок усвоения новых знаний.

Цель урока: научить создавать простейшую анимацию в среде Scratch.

Задачи:

Образовательная: создать условия для отработки навыков и умений работы в среде Scratch.

Развивающая: создать условия для развития внимания, памяти и воображения.

Воспитательная: содействовать развитию культуры речи и эстетического вкуса.

Основные понятия: анимация, исполнитель в Scratch, область сцены, объекты,

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. С помощью какого блока можно добавлять звук?
2. Какие существуют способы добавления новых спрайтов?
3. Как изменить угол наклона объекта?
4. Как изменить цвет объекта?

Урок 9. Исполнитель Scratch. Цвет и размер пера.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условия для формирования у детей представления об исполнителе Scratch.

Задачи:

Образовательная: ознакомить учащихся с возможностями использования пера: его цвет и размер; обеспечить представление о написании алгоритма и выполнении поставленной задачи.

Развивающая: развивать умение анализировать, выделять основное и второстепенное, умение слушать.

Воспитательная: воспитывать уважительное отношение к сверстникам и преподавателям, тактичность, вежливость, коммуникабельность.

Основные понятия: блок контроля, блок управления пером, исполнитель.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Какие характеристики существуют у объекта?
2. С помощью, каких команд начинается и заканчивается алгоритм?
3. С какой целью используется команда «очистить»?
4. В какой цвет окрашен блок «перо»?

Урок 10. Основные инструменты графического редактора программной среды Scratch.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для формирований представлений об инструментах графического редактора, встроенного в программную среду Scratch.

Задачи:

Образовательная: дать первоначальные сведения о графическом редакторе программной среды Scratch, инструментах и возможностях, при создании графических изображений.

Развивающая: совершенствовать учебно-интеллектуальные умения, а именно мотивацию к изучению предмета, внимание при восприятии информации, логическое осмысление предложенного материала, самостоятельное выполнение заданий, навыки самоконтроля.

Воспитательная: воспитывать ответственное отношение к работе и уважение не только к учителям, но и к сверстникам.

Основные понятия: графический редактор, инструменты графического редактора, блок внешности, блок управления пером

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Какие вы знаете инструменты графического редактора?
2. Какой блок окрашен в фиолетовый цвет?
3. Какой блок окрашен в зеленый цвет?

Урок 11. Рисование исполнителем Scratch линий. Линейный алгоритм.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для закрепления знаний о линейном алгоритме и применение знаний и навыков в ходе практического задания.

Задачи:

Образовательная: формировать навыки создания блок–схем линейного алгоритма; приобретать навыки решения конкретной задачи.

Развивающая: развивать учебно-организационные навыки, а именно воспринимать информацию, определять цель и намечать задачи;

Воспитательная: воспитывать упорность, трудолюбие, аккуратность, терпение при выполнении заданий,

Основные понятия: блок-схема, алгоритм, линейный алгоритм, виды элементов блок-схемы алгоритма.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Какие вы знаете инструменты графического редактора?
2. Какой блок окрашен в фиолетовый цвет?
3. Какой блок окрашен в зеленый цвет?

Урок 12. Рисование квадратов и прямоугольников линейно.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для формирования представлений о способах начертания фигур линейно.

Задачи:

Образовательная: формировать пространственное представление; закрепление знаний о линейном алгоритме.

Развивающая: развивать творческие способности детей, креативность в подходе решения вопросов.

Воспитательная: формировать ответственное отношение к учебе, дисциплинированность.

Основные понятия: команда «повернуть в направление», отладка программы.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Какие вы знаете геометрические фигуры?
2. Чем отличается квадрат от прямоугольника?
3. С помощью какой команды начинается выполнение алгоритма?

Урок 13. Цикл. Исполнитель рисует линии и фигуры.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для формирований представлений начертания фигур с помощью циклов.

Задачи:

Образовательная: создать представление о циклическом алгоритме;

Развивающая: совершенствовать умственные свойства и возможности учащихся;

Воспитательная: устранять небольшие недостатки в воспитании младших школьников, например неорганизованность, нетактичность и т.д.

Основные понятия: цикл, циклический алгоритм, блок-схема циклического алгоритма.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Каким способом можно открыть ранее созданный проект?
2. Сколько градусов имеет прямой угол?

Урок 14. Исполнитель Scratch рисует несколько линий и фигур. Конечный цикл.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для формирований представлений начертания фигур с помощью циклов.

Задачи:

Образовательная: совершенствовать и закрепить знания и умения в построении циклических алгоритмов.

Развивающая: развивать волевые качества младших школьников, умения самостоятельно преодолевать трудности.

Воспитательная: способствовать трудовому обучению школьников.

Основные понятия: цикл, конечный цикл, блок-схема конечного цикла, бесконечный циклический алгоритм, имя спрайта, изменение костюма объекта.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Какое отличие программы, которая рисует 3 ступеньки, от той которая рисует 8 ступенек?
2. Что характерного в бесконечном цикле?

Урок 15. Циклический алгоритм.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для закрепления представлений о циклическом алгоритме.

Задачи:

Образовательная: обеспечить в течение урока усвоение основных понятий циклического алгоритма.

Развивающая: формировать способность рассуждать логически, четко и кратко формулировать свои мысли.

Воспитательная: способствовать физическому воспитанию, а именно организовывать физкультминутки в целях снятия напряжения и усталости.

Основные понятия: цикл, блок-схема циклического алгоритма.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Какую команду следует использовать при дальнейшем перемещении по экрану, без следов рисования?
2. Где осуществляется выбор готовых костюмов для спрайта?

Урок 16. Рисование пунктирных линий.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для формирования представлений о циклических алгоритмах.

Задачи:

Образовательная: обеспечить контроль знаний и умений в ходе выполнения заданий.

Развивающая: развивать умение анализировать, выделять основное и второстепенное, умение слушать.

Воспитательная: воспитывать упорность, трудолюбие, аккуратность, терпение при выполнении заданий,

Основные понятия: цикл, пунктирные линии, алгоритм рисования пунктирных линий.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Дайте определение цикла.
2. Чему вы научились на сегодняшнем уроке?

Урок 17. Создание мультфильма.

Тип урока: урок закрепления знаний и умений.

Цель урока: создать условие для закрепления представлений о возможностях программной среды Scratch.

Задачи:

Образовательная: способствовать применению умений и навыков детей в ходе урока.

Развивающая: развивать творческое мышление, креативность, вкус.

Воспитательная: воспитывать ответственное отношение к работе и уважение не только к учителям, но и к сверстникам.

Основные понятия: спрайт, скрипт, алгоритм, команда.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Чему вы сегодня научились?
2. Какие блоки вы использовали для создания мультфильма?

2.3. Программно-методическая поддержка курса

В качестве программно-методической поддержки пропедевтического курса «Программирование в начальной школе» был разработан сайт с помощью WordPress. Учебное пособие располагается по адресу <http://j98232m5.beget.tech>.

На рисунке 13 представлена главная страница программно-методической поддержки курса.

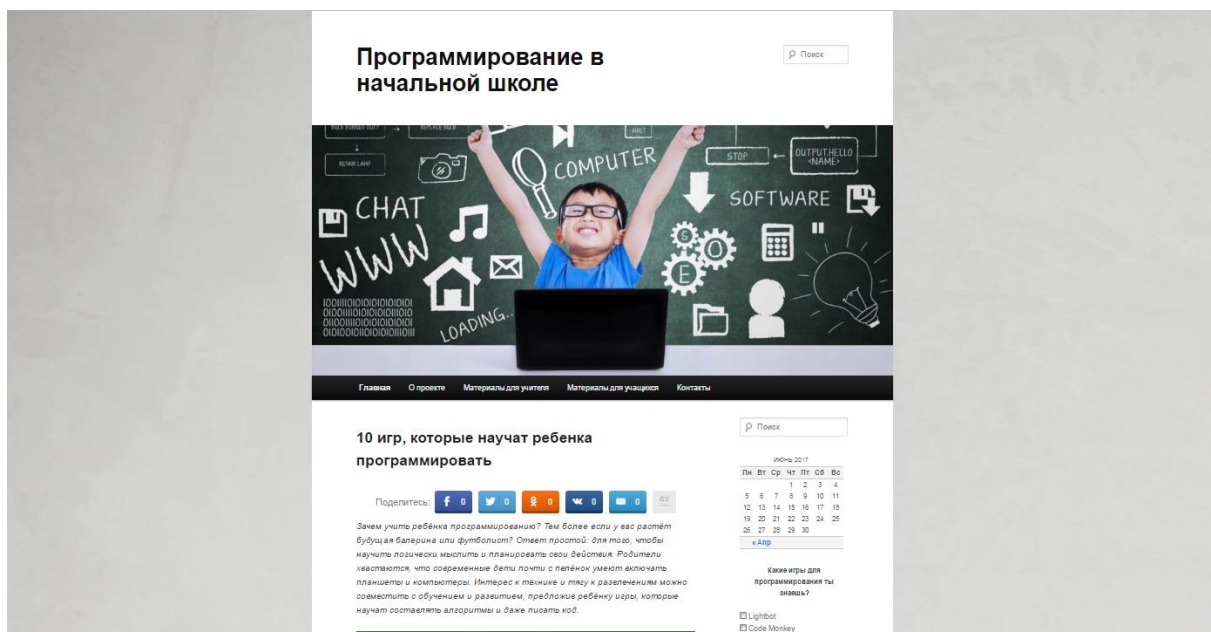


Рис. 13. Главная страница

На главной странице представлена статья, рассказывающая о 10 играх, которые научат ребенка программировать. Так же представлены такие разделы как главная, о проекте, материалы для учителя, материалы для учащихся и контакты.

В разделе «О проекте» (рис. 14) находится информация о содержании данного сайта.

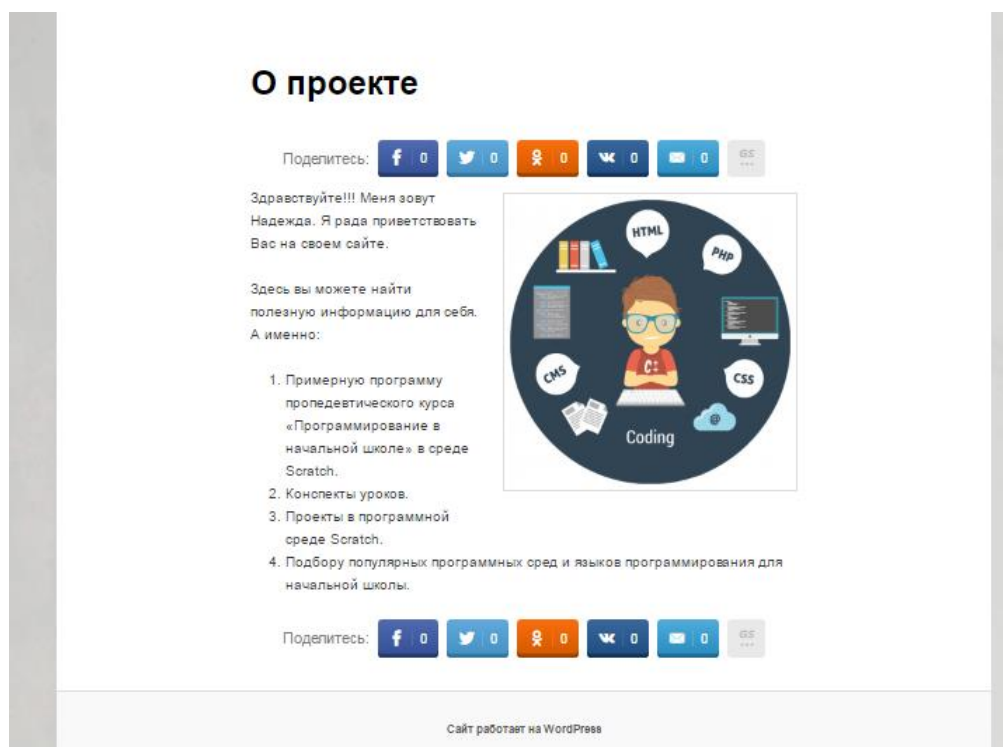


Рис. 14. раздел «О проекте»

В следующем разделе «Материалы для учителя» представлены примерная программа и конспекты уроков в хронологическом порядке (рис.15).

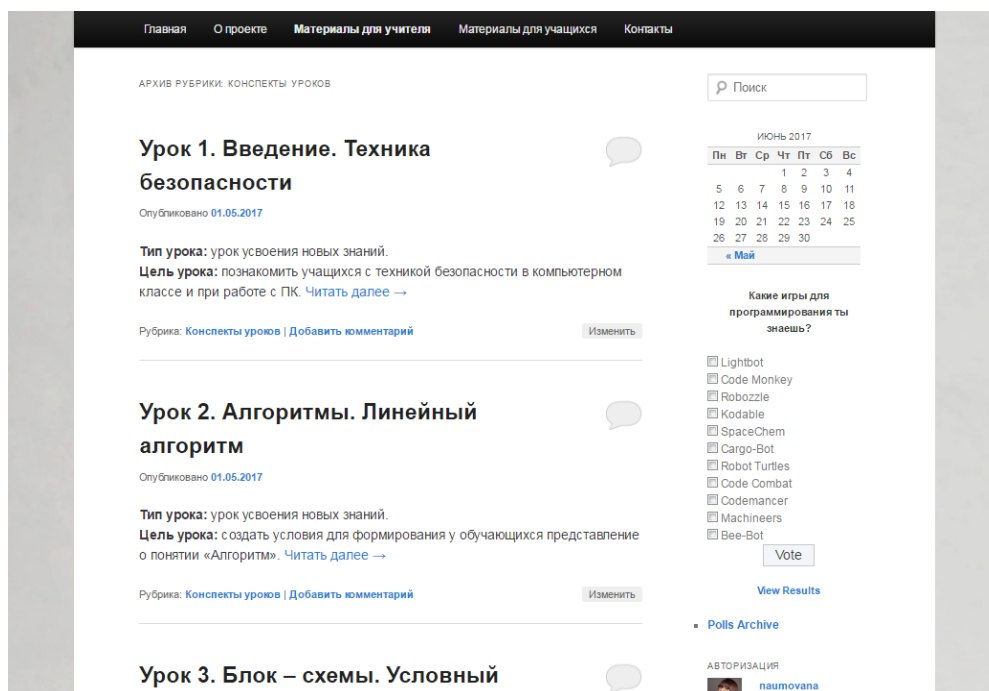


Рис. 15. Конспекты уроков

В разделе «Материалы для учащихся» расположены теоретические (рис. 16) и практические (рис. 17) сведения о программировании в Scratch.

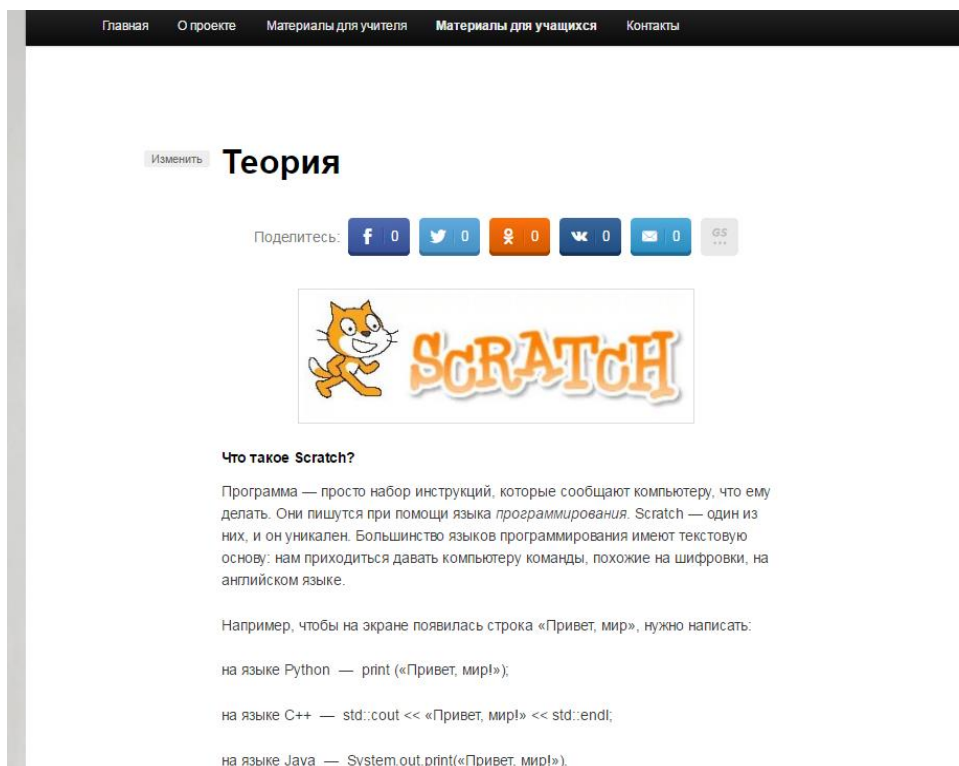


Рис. 16. Теория

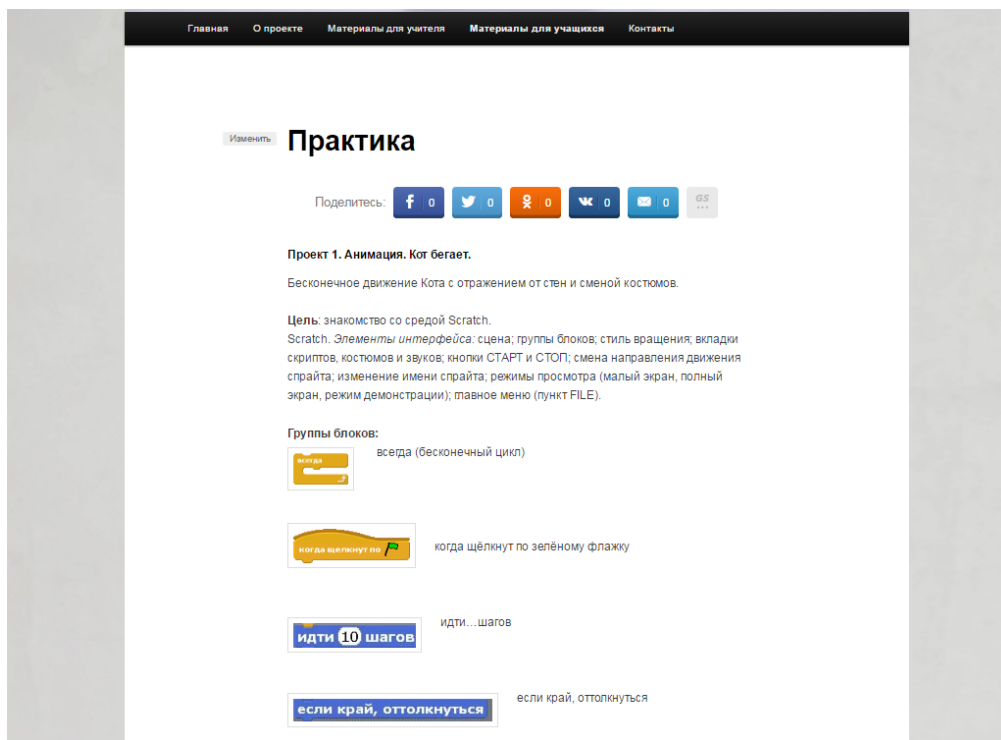


Рис. 17. Практика

В разделе «Контакты» представлена информация для организации связи с разработчиком сайта.

2.4. Апробация результатов исследования в школе

Педагогическая апробация проводилась в рамках научно-исследовательской практики в МБОУ СОШ № 121 г. Челябинска. Курс изучался в 6-х классах. В течении 3-х занятий была рассмотрена тема программирования.

Апробация прошла успешно. Способствовала этому правильная мотивация, цели и задачи изучения темы.

Тема курса оказалась незнакома для учащихся, но они быстро включились в работу. В таком возрасте дети особенно заинтересованы в получении знаний, особенно если информация наглядная и простая.

Вывод по главе 2

Во второй главе были рассмотрены образовательные стандарты, описан пропедевтический курс «Программирование в начальной школе» и программно-методическая поддержка, в виде образовательного портала.

Из анализа ФГОС НОО мы выяснили, что внеурочная деятельность это не только составная часть учебно-воспитательного процесса, но и одна из форм организации свободного времени учеников, нацеленная на своеобразную помощь ребенку в преодолении каких-либо затруднений в новой учебной деятельности и создание учебной мотивации, интереса в данной дисциплине [1].

Аппробация курса проводилась в рамках научно-исследовательской практики в МБОУ СОШ № 121.

Таким образом, во второй главе мы разработали и опробовали пропедевтический курс «Программирование в начальной школе» и программно-методическую поддержку.

ЗАКЛЮЧЕНИЕ

Подводя итоги проделанной работы, можно сделать вывод о том, что цель исследования была достигнута, а именно изучить особенности пропедевтики программирования и разработать пропедевтический курс «Программирование в начальной школе», используя возможности внеурочной деятельности по информатике в начальной школе.

Разработанный курс, рассчитанный на проведения занятий во внеурочное время в начальной школе, с одной стороны позволяет осуществить пропедевтику изучения программирования у школьников, содействовать развитию алгоритмического мышления, формированию логико-алгоритмической компоненты. С другой стороны обеспечить учителю освоение новой программной среды Scratch.

В процессе исследования были решены следующие задачи:

1. Изучены теоретические основы по проблеме исследования.
2. Разработан пропедевтический курс по программированию в среде Scratch для начальной школы.
3. Разработана программно-методическая поддержка пропедевтического курса в виде образовательного портала.

В подтверждении гипотезы можно сказать, что пропедевтический курс программирования ведет к овладению учащимися основам логического и алгоритмического мышления.

Таким образом, все поставленные задачи были выполнены, что говорит о правильности поставленной гипотезы.

Библиографический список

1. Alice [Электронный ресурс] - <http://soft.mydiv.net/win/download-Alice.html>
2. Baltie [Электронный ресурс] – <http://progopedia.ru/language/baltie/>
3. Etoys [Электронный ресурс] – <https://ru.wikipedia.org/wiki/EToys>
4. Greenfoot [Электронный ресурс] - <https://www.greenfoot.org/download>
5. Guido van Robot [Электронный ресурс] - <http://gvr.sourceforge.net>
6. Karel J. Robot [Электронный ресурс] - <http://karel.sourceforge.net>
7. Lightbot [Электронный ресурс] - <http://lightbot.com>
8. Little Wizzard [Электронный ресурс] - <http://littlewizzard.sourceforge.net>
9. Logo [Электронный ресурс] - <https://myrobot.ru/logo/aboutlogo.php>
10. RoboMind [Электронный ресурс] - <https://www.robomindacademy.com/go/robomind/home>
11. Scratch [Электронный ресурс] - <https://scratch.mit.edu>
12. Squeak [Электронный ресурс] – <http://squeak.org>
13. Богомолова Е.В. Программа курса «Теория и методика обучения информатике на начальной ступени» [Текст] // Информатика и образование. 2007. № 1. С. 86–99.
14. Босова Л., Босова А. Информатика. [Текст]: Учебник для 6 класса / Л. Босова, А. Босова – М.:БИНОМ. Лаборатория знаний, 2016. – 224с.
15. Босова Л., Босова А. Информатика. [Текст]: Учебник для 9 класса / Л. Босова, А. Босова – М.:БИНОМ. Лаборатория знаний, 2016. – 208с.
16. Википедия [Электронный ресурс] - <https://ru.wikipedia.org/wiki/Пропедевтика>
17. Ершов А.П., Звенигородский Г.А., Первин Ю.А. Школьная информатика (концепции, состояния, перспективы) [Текст] // Информатика и образование. 1995.
18. Ибрагимов Г.И., Андриянова Т.М. Теория обучения [Текст]: Теория обучения: учебное пособие / Под редю Г.И. Ибрагимова. [Ибрагимова Е.М., Андриянова Т.М.] – М.:Гуманитар. Издю центр ВЛАДОС, 2011. – 383 с.

19. Изучение основ информатики и вычислительной техники. В 2-х частях. [Текст] – М.: Просвещение, 1985, 1986.
20. Лапчик М.П. и др. Методика преподавания информатики [Текст]: Учеб. Пособие для студ. Пед. Вузов / М.П. Лапчик, И.Г. Семакин, Е.К. Хеннер; Под общей ред. М.П. Лапчика. – М.: Издательский центр «Академия», 2001. – 624с.
21. Поляков К., Информатика. [Текст]: 11 класс. Углубленный уровень: учебник в 2 ч. / К.Ю. Поляков, Е.А. Еремин - М. : БИНОМ. Лаборатория знаний, 2016.
22. Семакин И., Информатика. [Текст]: Учебник для 9 класса /И.Г. Семакин, Л.А. Залогова, С.В. Русаков, Л.В. Шестакова. – М.: БИНОМ. Лаборатория знаний, 2016. – 208с.
23. Семакин И., Информатика. [Текст]: 10 класс. Углубленный уровень: учебник в 2 ч. / И.Г. Семакин, Т.Ю. Шеина, Л.В. Шестакова - М. : БИНОМ. Лаборатория знаний, 2016.
24. Семакин И., Информатика. [Текст]: 11 класс. Углубленный уровень: учебник в 2 ч. / И.Г. Семакин, Т.Ю. Шеина, Л.В. Шестакова - М. : БИНОМ. Лаборатория знаний, 2016.
25. Угринович Н. Информатика и ИКТ. [Текст]: Учебник для 9 класса / Н. Угринович. – М.: БИНОМ. Лаборатория знаний, 2012. – 295с.
26. Федеральный государственный образовательный стандарт начального общего образования 2009г [Электронный ресурс] - <http://минобрнауки.рф/документы/543>
27. Федеральный государственный образовательный стандарта основного общего образования 2010г [Электронный ресурс] - <http://минобрнауки.рф/документы/543>
28. Цветкова М.С. Путешествие робота Вопросика в страну Информатики: Курс информатики для младших школьников // Информатика в младших классах. 1999. № 3. С. 3–64. № 4. С. 3–58.

Конспект урока № 2

Тема урока: Путешествие в страну «Алгоритмы».

Тип урока: изучение и первичное закрепление новых знаний в нестандартной форме «урок-путешествие».

Цель урока: создать условие для формирования у обучающихся представление о понятии «Алгоритм».

Задачи:

Образовательная:

- Познакомить учеников с понятиями «алгоритм», «линейный алгоритм».
- Научить составлять и выполнять алгоритмы.

Развивающая:

- Развивать логическое мышление учащихся, память, внимание, воображение, познавательную активность, способность быстро воспринимать информацию.

Воспитательная:

- Воспитывать у учащихся самостоятельность, активность, интерес к предмету, правила поведения.

Этапы урока:

1. Организационный момент – 1 мин.
2. Постановка темы и цели урока – 5 мин.
3. Изучение нового материала – 15 мин.
4. Физкультминутка – 2 мин.
5. Закрепление изученного материала – 5 мин.
6. Итог урока, рефлексия, домашнее задание – 2 мин.

Перечень средств ИКТ, используемых на уроке:

- Интерактивная доска.
- Презентация к уроку.

СТРУКТУРА И ХОД УРОКА

Деятельность учителя	Деятельность учащихся
1. Организационный момент	
<p><i>Приветствие учащихся, создание благоприятной атмосферы.</i></p> <p>Громко прозвенел звонок- Начинается урок Наши ушки на макушке, Глазки хорошо открыты. Слушаем, запоминаем, Ни минутки не теряем.</p> <p>Мы пришли сюда учиться, Не лениться, а трудиться, Работаем старательно Слушаем внимательно.</p> <p>Каждый день, всегда, везде, На занятиях, в игре, Смело, четко говорим И тихонечко сидим.</p> <p>Здравствуйтесь ребята. Сегодня мы отправимся с вами в одну очень интересную страну. А вот в какую, это вам предстоит угадать. Но сначала, давайте наберемся сил и скушаем по конфетке.</p>	<p><i>Приветствуют учителя, демонстрируют готовность к работе.</i></p>
<p><i>Учитель раздает детям по одной конфетке.</i></p>	<p><i>Кушают конфеты.</i></p>

2. Постановка темы и цели урока.	
Итак, сколько действий мы с вами сделали, и какие, чтобы скушать конфетку?	<i>Дети начинают вспоминать последовательность, отвечают на вопрос. Взять конфетку. Развернуть ее. Съесть конфетку. Выбросить фантик в мусорное ведро.</i>
Как можно назвать последовательность действий, которые мы с вами выполнили?	<i>Школьники высказывают свои мнения.</i>
Чтобы отгадать, взгляните на слайд. Из первых букв животных сложите слово. <i>Учитель показывает на слайде картинки животных: А – аист, Л - лиса, Г - гусь , О - олень, Р - рысь, И - индюк, Т - тигр, М - медведь.</i>	Алгоритм.
Верно! Алгоритм. Именно в эту страну нам предстоит отправиться, где сможем узнать, что такое алгоритм и как правильно составить его.	
3. Изучение нового материала.	
Как вы думаете, что такое алгоритм?	<i>Рассуждают, высказывают свои мнения.</i>
<i>Учитель показывает слайд с определением на интерактивной доске.</i> Алгоритм – это набор инструкций,	

<p>описывающий порядок действий, для достижения результата. Каждый из нас ежедневно использует различные алгоритмы: инструкции, правила, рецепты и т. п. Обычно мы это делаем не задумываясь. Например, открывая дверь ключом, никто не размышляет над тем, в какой последовательности выполнять действия.</p>	
<p>Подумайте пожалуйста и приведите пример ваших повседневных действий.</p>	<p><i>Размышляют. Выдвигают версии</i> Почистить зубы, заправить кровать, одеться, налить чай и т.д.</p>
<p>Молодцы! Каждый шаг в алгоритме принято называть командой. В начале принято записывать команду «начало», а в конце команду «конец».</p> <p>Давайте с вами немного потренируемся. У нас есть некоторые продукты: масло, сыр, колбаса и хлеб. Предлагаю сделать бутерброд.</p> <p><i>Учитель дублирует информацию на слайде в виде картинок.</i></p>	
<p>Итак, ребята, с чего начнем?</p>	<p>Отрежем кусок хлеба.</p>
<p>Верно. Но не забываем, что алгоритм начинается всегда с команды «Начало». Следующие шаги?</p>	<ol style="list-style-type: none"> 1. Смажем кусок хлеба маслом. 2. Отрежем ломтик сыра. 3. Положим сыр на хлеб. 4. Отрежем кусок колбасы. 5. Положим колбасу на бутерброд.

<p>Хорошо. Молодцы. Но как необходимо закончить алгоритм?</p>	<p>Командой «Конец».</p>
<p>Верно. Мы составили алгоритм. Все действия в нем следуют друг за другом. Предположите, как называется такой алгоритм?</p>	<p>Линейный алгоритм.</p>
<p>Верно. Линейный алгоритм – действия, которые выполняются друг за другом.</p>	
<p>Знаете, ребята, хозяин страны «Алгоритм» хотел написать инструкции, но все перепутал! Давайте поможем ему разобраться.</p>	<p><i>Дети с интересом соглашаются.</i></p>
<p>Посмотрите на доску. <i>На слайде изображены алгоритмы с ошибками, дополненные картинками.</i> Предлагаю помочь незнакомцу и исправить ошибки.</p>	<p>Смотрят на доску. Анализируют алгоритмы.</p>
<p>УБОРКА КВАРТИРЫ С ПОМОЩЬЮ ПЫЛЕСОСА.</p> <p>Начало.</p> <ol style="list-style-type: none"> 1. Принести пылесос в комнату. 2. Подключить к сети. 3. Нажать ВКЛ на корпусе. 4. Нажать ВЫКЛ. на корпусе. 5. Пропылесосить. 6. Отключить от сети. 7. Поставить на место. 	<p>4 и 5 пункт поменять местами.</p>

<p>Конец.</p>	
<p>ВЫМОЙ ТАРЕЛКУ.</p> <p>Начало.</p> <ol style="list-style-type: none"> 1. Открой кран. 2. Возьми тарелку. 3. Надень шапку. 4. Вымой тарелку. 5. Вытри её. 6. Поставь в шкаф. 7. Закрой кран. <p>Конец.</p>	<p>Лишний пункт №3.</p>
<p>ПОЛЕЙ ЦВЕТОК.</p> <p>Начало.</p> <ol style="list-style-type: none"> 1. Возьми лейку. 2. Открой кран. 3. Наполни лейку водой. 4. Поставь лейку на место. <p>Конец.</p>	<p>Не хватает команды «Полей цветок» после команды «Наполни лейку водой».</p>
<p>4. Физкультминутка</p>	
<p>Повторяйте за мной движения и слова.</p> <p><i>Учитель демонстрирует движения.</i></p> <p>Ветер дует нам в лицо, Закачалось деревцо. Ветерок всё тише, тише. Деревцо всё выше, выше.</p>	

<p>Листочки зашелестели, И ребята тихо сели.</p>	
<p>5. Закрепление изученного материала</p>	
<p>Открываем тетради. Сейчас я буду вам говорить команды, а вы будете их исполнять. Начало. Отступи 5 клеточек с левого и верхнего края, ставим точку. Проведите от точки, не отрывая руки, 3 клеточки вверх, 3 клетки вправо, 2 клетки вниз, 2 клетки вправо, 1 клетку вниз, 5 клеток влево. Конец.</p>	
<p>Что у нас получилось? Запишите алгоритм выполнения этого рисунка. <i>Один ученик работает у доски, остальные в тетради.</i></p> <ol style="list-style-type: none"> 1. Начало. 2. • 3. ↑ 3 4. → 3 5. ↓ 2 6. → 2 7. ↓ 1 8. ← 5 9. Конец. 	<p>Получилась фигура. <i>Открывают тетради и выполняют задание.</i></p>
<p>6. Итог урока, рефлексия, домашнее задание</p>	

Ребята, чему вы учились на этом уроке?	Мы учились правильно составлять алгоритм.
Что такое алгоритм?	Алгоритм – это последовательность действий, приводящих к решению поставленной задачи
Ребята, о каком алгоритме мы сегодня говорили?	Мы говорили о линейном алгоритме.
Какие формы записи алгоритма мы использовали сегодня на уроке?	Мы использовали словесную, условными знаками.
Вспомните, где в нашей повседневной жизни мы встречаемся с алгоритмами?	Убраться в комнате, сделать домашнее задание, помыть руки и т.д.
Молодцы. Мы сегодня хорошо потрудились. Оцените себя, какую оценку вы себе поставите за урок? У: Встаньте, кто поставил себе оценку «5». Встаньте, кто поставил себе оценку «4». А плохие оценки мы сегодня не ставим себе.	<i>Ученики оценивают себя.</i>
Запишите домашнее задание: Девочкам написать алгоритм приготовления блинов. Мальчикам составить алгоритм замены колеса и составить.	<i>Записывают домашнее задание в дневник.</i>

У кого есть вопросы по домашнему заданию? <i>Разъясняет в случае возникновения вопросов.</i> Всем спасибо, урок окончен.	<i>Задают вопросы.</i>
--	------------------------

Конспект урока № 7

Тема урока: Знакомство со Scratch.

Тип урока: урок усвоения новых знаний.

Цель урока: познакомить учащихся с интерфейсом программы Scratch.

Задачи:

Образовательная: дать представление программы Scratch, определение скрипта и спрайта.

Развивающая: развивать у обучающихся интерес к предмету, умения и навыки при работе с программой Scratch.

Воспитательная: воспитывать трудолюбие, дисциплинированность, доброжелательное отношение друг другу и умение преодолевать трудности.

Этапы урока:

1. Организационный момент – 1 мин.
2. Изучение нового материала – 10 мин.
3. Физкультминутка – 2 мин.
4. Закрепление изученного материала – 17 мин.

Перечень средств ИКТ, используемых на уроке:

- Интерактивная доска, компьютеры для каждого ученика, программа Scratch.

СТРУКТУРА И ХОД УРОКА

1. Организационный момент 1 мин.

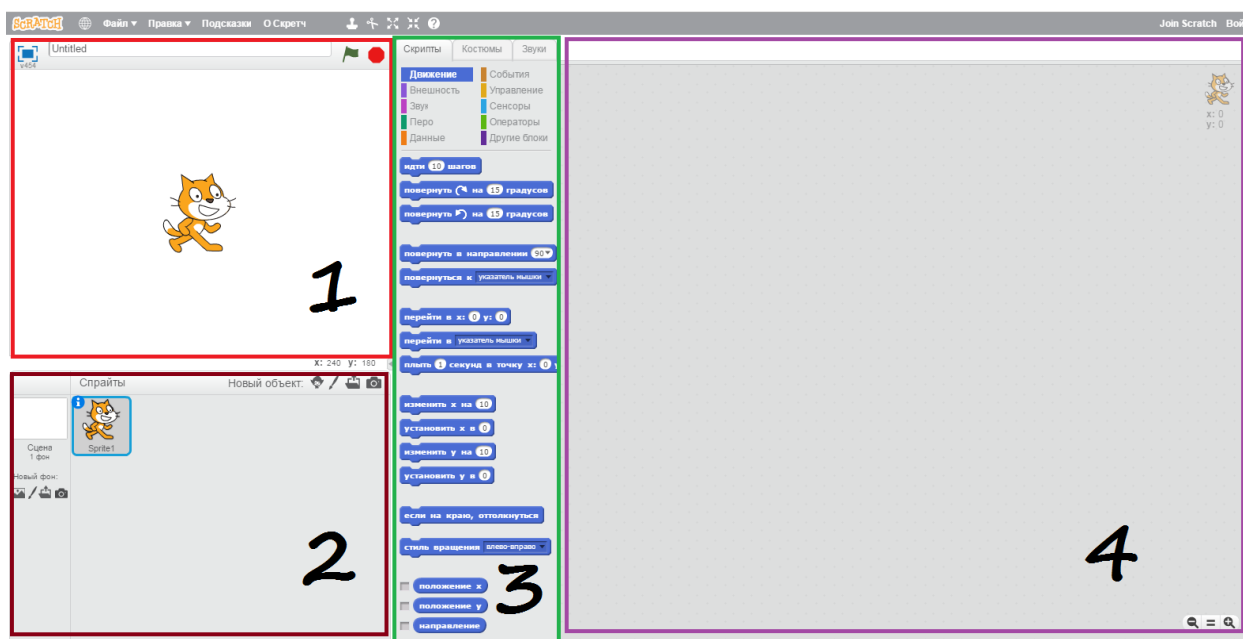
Учитель: *Приветствует класс, отмечает присутствующих.*

Здравствуйте, ребята. Сегодня мы с вами познакомимся с программой Scratch и даже напишем первую программу.

2. Изучение нового материала 10 мин.

Но для начала рассмотрим интерфейс программы.

Так выглядит экран программы, которая состоит из четырех частей.



Область под номером 1 – это сцена, на которой видно как работает программа.

Область под номером 2 – это список спрайтов. Спрайт – это объекты в программе, такие как животные, автомобили, люди, буквы и т.д.

Область под номером 3 – набор блоков, которые задают спрайту различные свойства, например движение, звук, контроль, внешность и т.д.

Область под номером 4 – это рабочая область (область скриптов), куда мы будем складывать блоки из 3 окна. Скрипт – это программа, которая состоит из блоков и описывает какое-либо действие спрайта.

Предлагаю область под номером 3 рассмотреть внимательней.

В Scratch большое количество блоков разного цвета. Мы познакомимся со всеми постепенно в течении курса, но для начала будем использовать синие, фиолетовые, розовые и желтые.

В данном случае мы выбрали раздел движение.

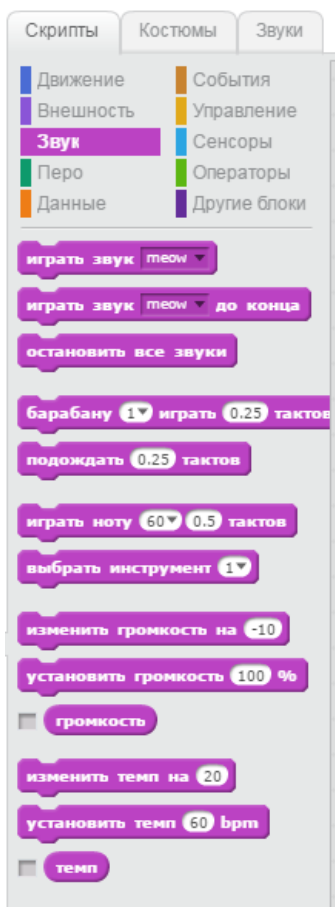
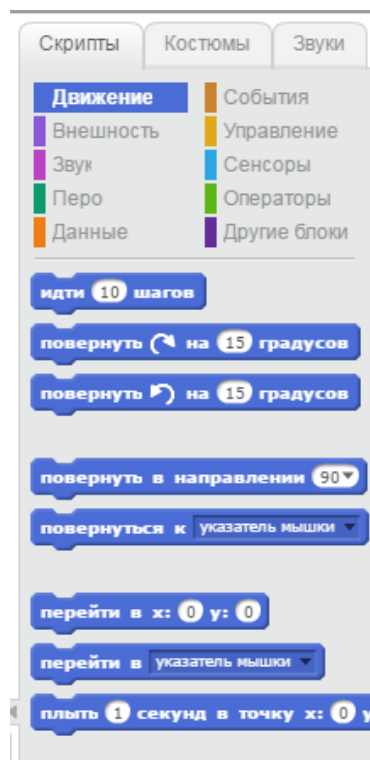
Скажите мне, пожалуйста, что наш спрайт может делать, если мы воспользуемся этими блоками?

Ученики:

Идти 10 шагов, повернуть направо/налево на 15 градусов, повернуть в направлении, повернуться к чему-либо, плыть, менять расположение, вращаться.

Учитель:

Верно. Хочу заметить числа в окошечках мы можем задавать сами, например идти не 10 шагов, а 7 или 40 шагов.



Перейдем к розовым блокам. Этот раздел отвечает за звуки. Прочитайте по очереди.

Ученики:

Играть звук, останавливать все звуки, подождать несколько тактов, изменять громкость, изменять темп.

Учитель:

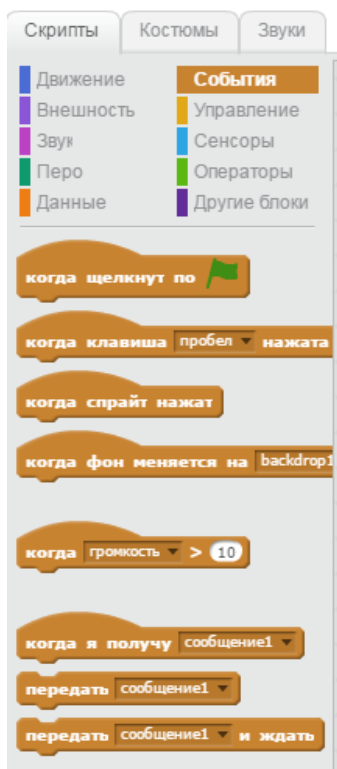
Молодцы. На самом деле мы не только можем выбирать различные звуки из библиотеки программы, но и записывать их сами.

Рассмотрим раздел внешность.

Прочитайте по очереди.

Ученики:

Говорить слово в течении 2 секунд, сказать слово, думать, показаться, спрятаться, сменить костюм, поменять фон, изменять размер.



Учитель:

Хорошо. И последний раздел, который мы сегодня рассмотрим, называется события. Данные блоки отличаются от предыдущих тем, что они являются вспомогательными к блокам из других разделов.

Например блок «когда спрайт нажат» мы не можем использовать самостоятельно, обязательно к нему прикрепим другой например «идти 1 шаг». В результате получится – когда спрайт нажат, идти 1 шаг.

Ребята, мы переходим к практической части. Но перед тем как сесть за компьютеры, предлагаю

немного размяться и выполнить несколько физических упражнений.

3. Физкультминутка 2 мин.

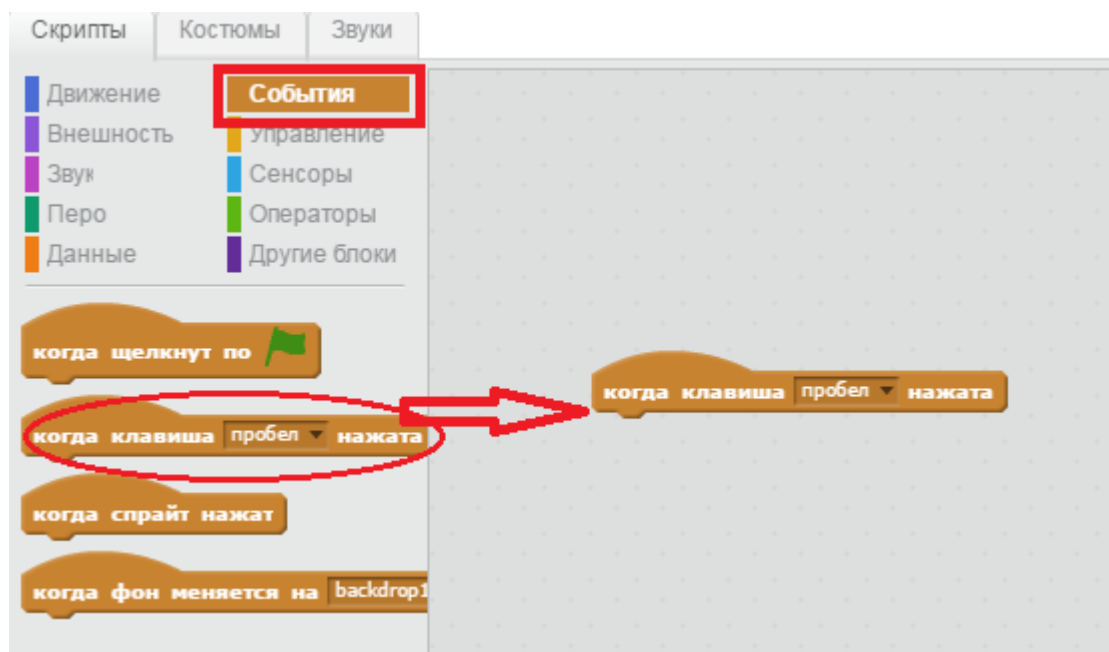
Встаем из за парт. Руки вверх, тянемся за ручками. Руки в стороны, затем вниз и расслаблено скрещиваем перед грудью, голову наклоняем вперед. Повторяем 5 раз (темп средний).

4. Закрепление изученного материала 17 мин.

Молодцы. Теперь садимся за компьютеры. Запускаем программу Scratch с рабочего стола.

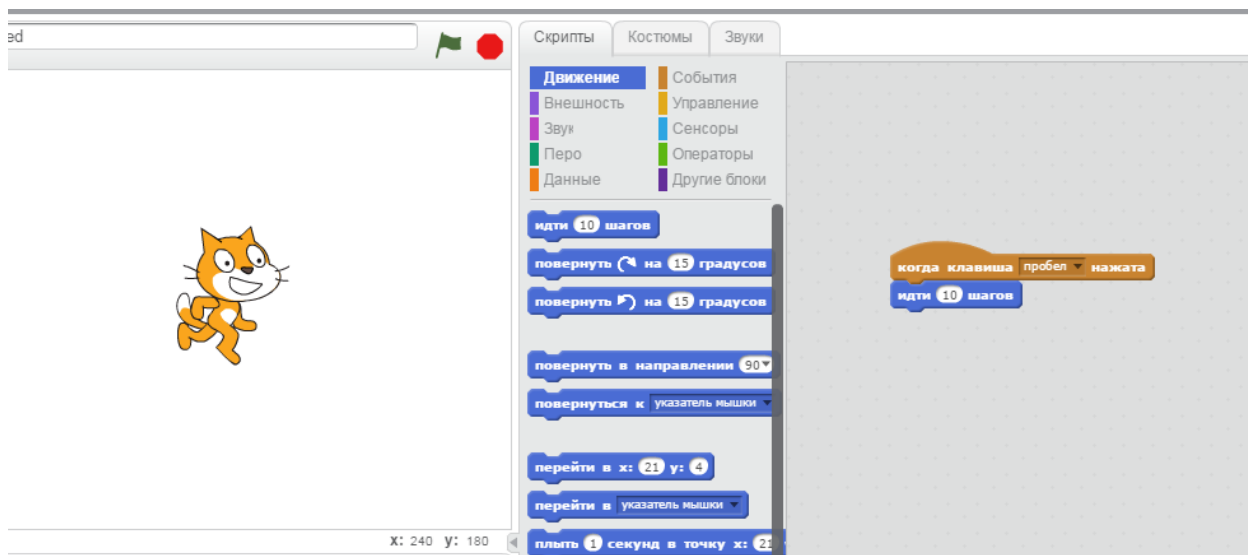
Задание 1. Учим кота ходить.

Выбираем вверху желтые блоки «события», находим блок «когда клавиша пробел нажата», нажимаем левой кнопкой мыши и, не отпуская, перетаскиваем блок вправо в область скриптов.



Затем выбираем синие блоки «движение», тянем блок «идти 10 шагов» в область скриптов и прицепляем под желтый блок «когда клавиша пробел нажата».

В результате у вас должно получиться так.



Если появились затруднения, поднимайте руку, и я подойду к вам.

Вот наша первая программа! Теперь необходимо протестировать ее.

Нажимайте на клавиатуре пробел и скажите что происходит с котиком?

Ученики:

Кот двигается вправо.

Учитель:

Верно. Если много раз нажимать на пробел, то он дойдет до края экрана и почти скроется.

Его можно вытащить, потянув за хвост.

Задание 2. Учим кота мяукать.

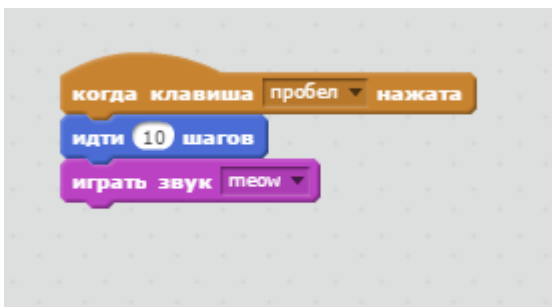
Чтобы наш кот начал издавать звуки, например мяукать, нужно добавить соответствующий блок. Как вы думаете в каком разделе он находится?

Ученики:

В разделе «звуки».

Учитель:

Верно. Переходим в раздел «звуки» и находим там блок «играть звук meow». Затем перетаскиваем его к уже имеющимся блокам и прикрепляем к синему блоку.

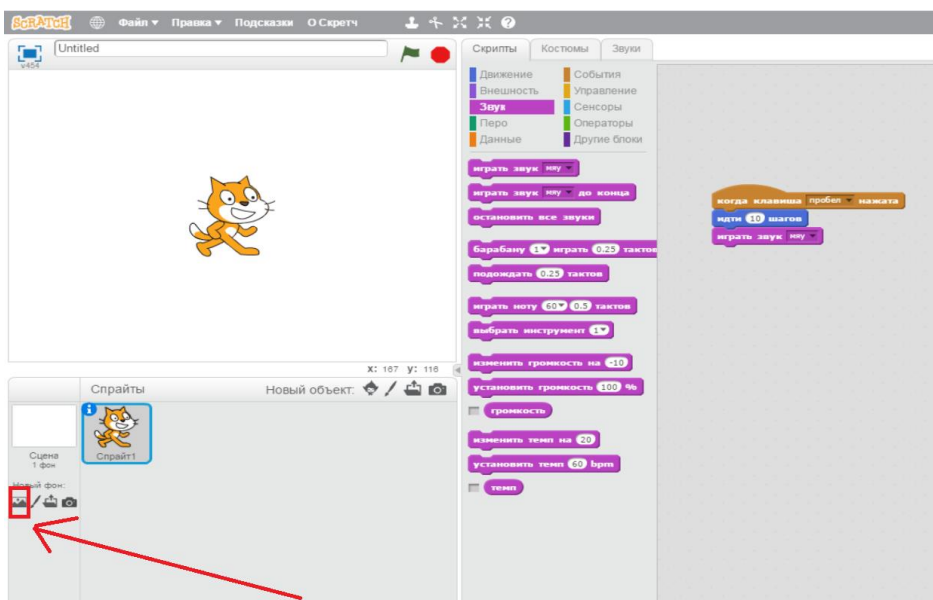


Теперь при нажатии на пробел кот не только идет, но и мяукает.

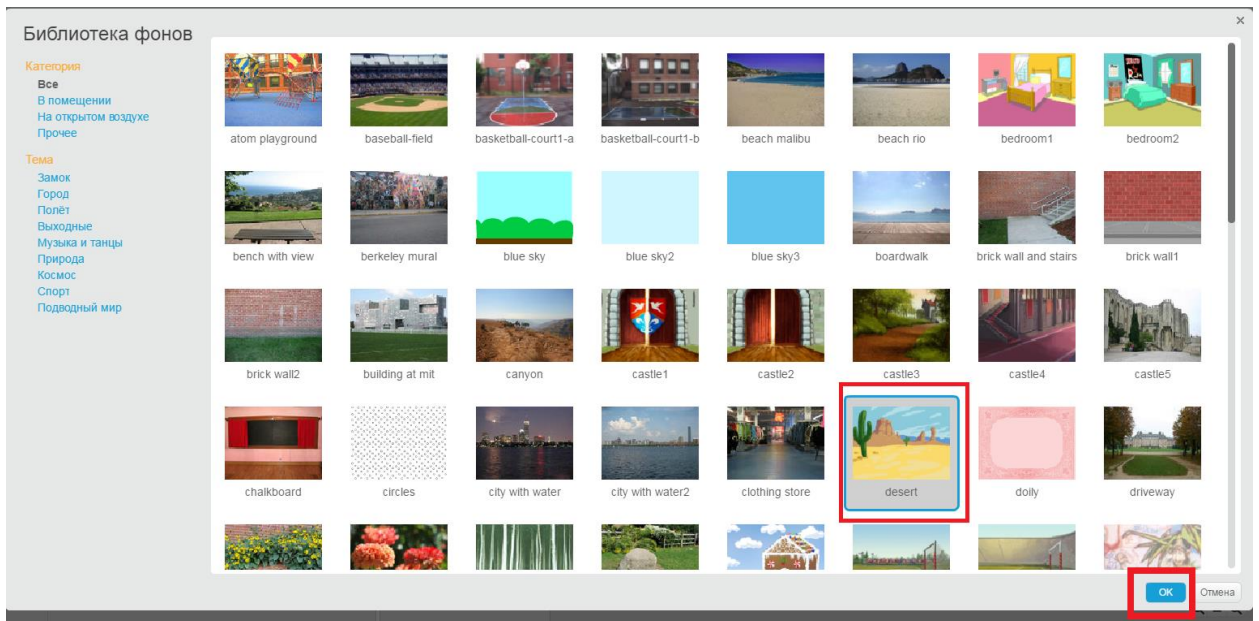
У кого есть вопросы, спрашивайте.

Задание 3. Меняем фон.

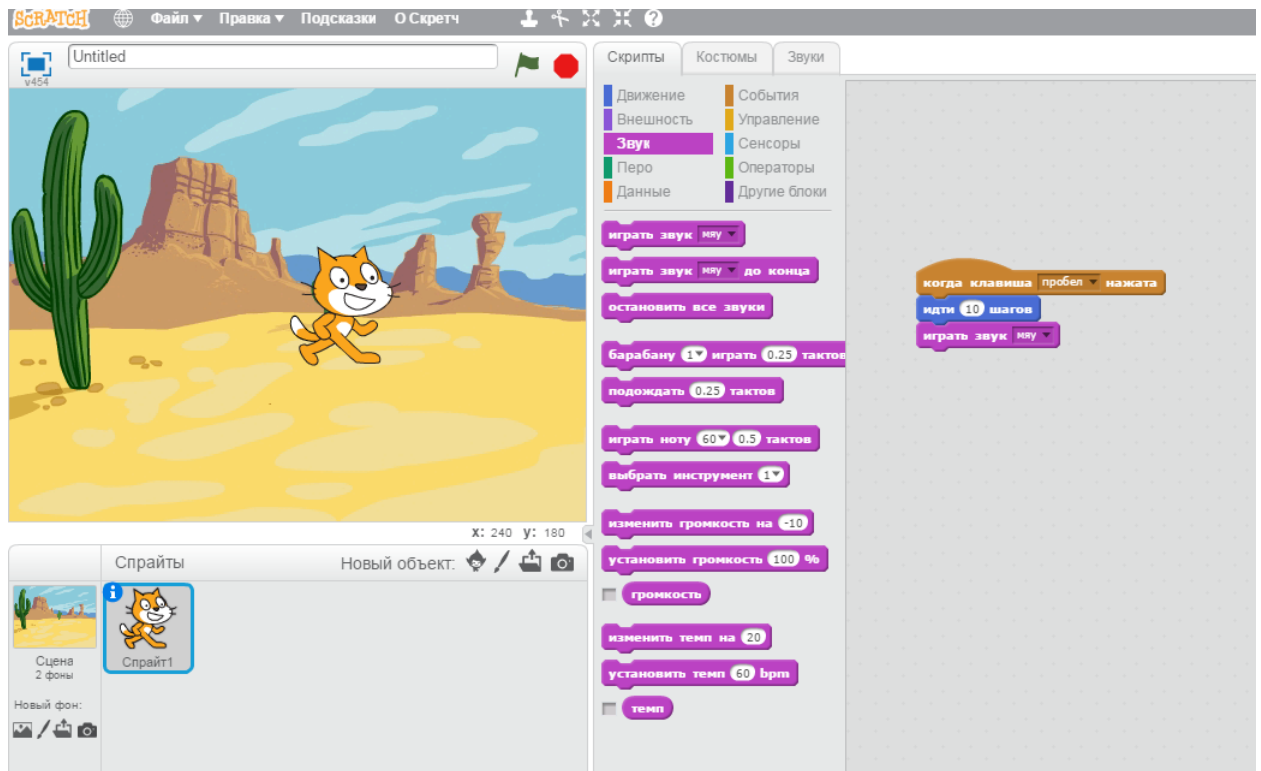
Ребята согласитесь, что белый фон – это скучно. Давайте выберем подходящий фон, например пустыню или пляж. Для этого необходимо в левом нижнем углу найти кнопку «Выбрать фон из библиотеки».



Нажимаем на кнопку, и появляется окно – Библиотека фонов. Там выбирайте любой понравившийся фон (я выбрала пустыню) и нажимайте кнопку «ОК» в правом нижнем углу.



Фон изменился. Согласитесь со мной, что так куда более интересней.



У всех получилось? Если есть затруднения, то учитель подходит и помогает ученику.

Если вдруг вы нечаянно добавили не тот блок или просто захотели удалить блок, то необходимо открепить его и перетащить обратно.

Самостоятельно!

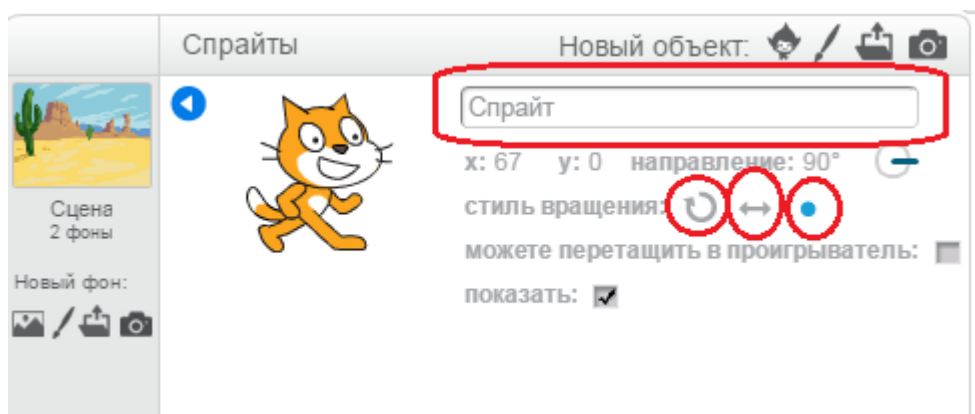
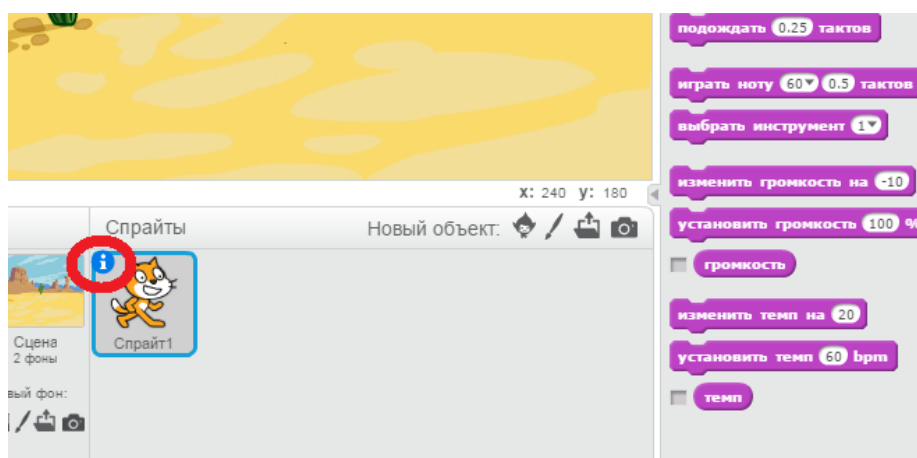
Попробуйте самостоятельно убрать блок «играть звук», а вместо него поставить новый «если на краю, оттолкнуться» из раздела движение. Что в итоге получилось?

Ученики:

Кот шагает при нажатии на пробел, но теперь не скрывается, когда доходит до края экрана, а поворачивается и движется в другую сторону и не мяукает.

Учитель:

Хорошо. В левом нижнем углу в поле спрайтов найдите кота и рядом значок «i». Нажмите на него.



Измените имя спрайта на кот. А так же поменяйте стиль вращения.

1 стиль означает, что кот будет ходить по кругу, когда достигнет края экрана.

2 стиль более правдив, ведь кот развернется, когда дойдет до края.

А что означает 3 стиль? Поэкспериментируйте.

Ученики:

Кот по достижению края пойдет задом.

Учитель:

Хорошо. А теперь предлагаю сохранить вашу программу на компьютере. Для этого в левом верхнем углу найдите кнопку «Файл» и нажмите «Сохранить как». В окне выбирайте «Рабочий стол» и «Сохранить».

А теперь закрываем программу, нажав на красный крестик в правом верхнем углу.

Учитель:

Ребята, вы сегодня все большие молодцы. Расскажите, чему вы научились?

Ученики:

Знаем что такое скрипт и спрайт, научились перетаскивать блоки, знаем, как поменять фон сцены, научились задавать звук спрайту.

Учитель:

Всем спасибо, урок закончен.

Конспект урока № 8

Тема урока: Анимация имени.

Тип урока: урок усвоения новых знаний.

Цель урока: научить создавать простейшую анимацию в среде Scratch.

Задачи:

Образовательная: создать условия для отработки навыков и умений работы в среде Scratch.

Развивающая: создать условия для развития внимания, памяти и воображения.

Воспитательная: содействовать развитию культуры речи и эстетического вкуса.

Этапы урока:

1. Организационный момент – 3 мин.
2. Практическая работа – 12 мин.
3. Физкультминутка – 2 мин.
4. Практическая работа (продолжение) – 13 мин.

Перечень средств ИКТ, используемых на уроке:

- Интерактивная доска, компьютеры для каждого ученика, программа Scratch.

СТРУКТУРА И ХОД УРОКА

1. Организационный момент 3 мин.

Учитель: *Приветствует учеников, отмечает присутствующих.*

Здравствуйте ребята. На предыдущем уроке мы с вами познакомились с простейшими действиями в программе Scratch. Давайте вспомним, чему мы учили нашего котика?

Ученики:

Учили кота ходить, мяукать, отталкиваться от края экрана. Задавали новый фон сцены.

Учитель:

Верно. Ответьте мне на вопрос что такое спрайт и скрипт?

Ученики:

Спрайт – это графический объект, выполняющий какие-либо действия в проекте (кот, машина, человек и т.д.)

Скрипт – это программа, которая описывает действия объекта (спрайта)

Учитель:

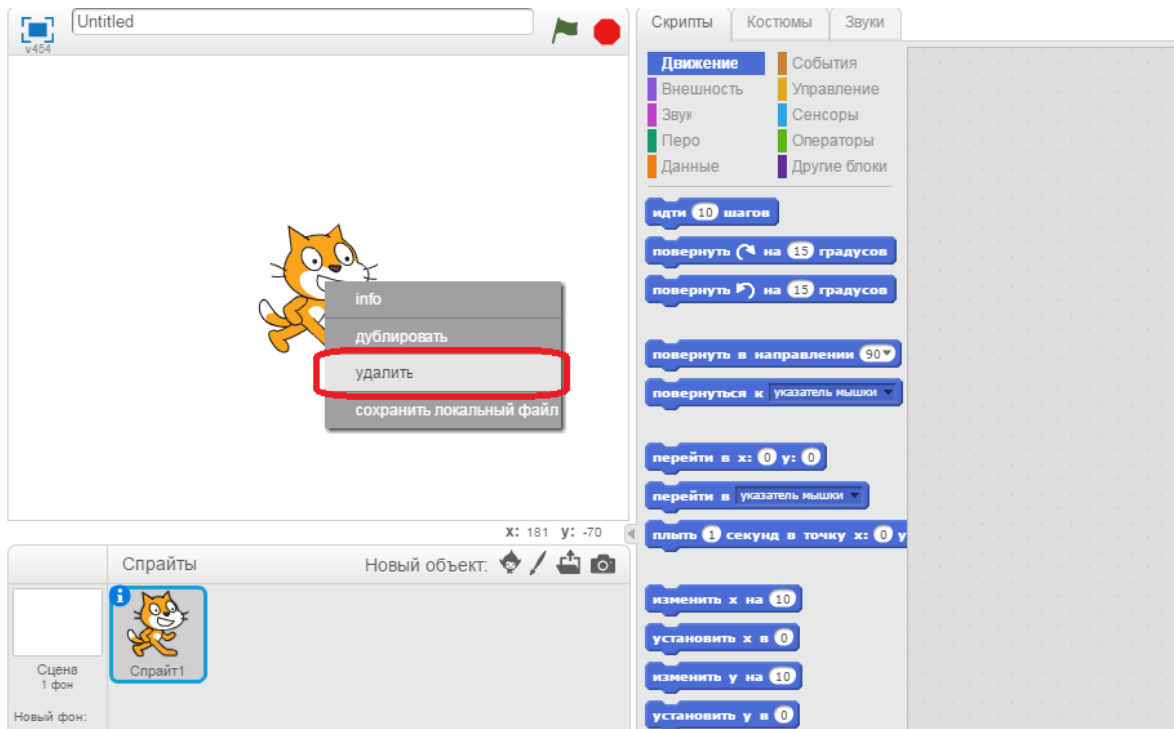
Молодцы. Сегодня мы продолжим работу в Scratch. Вы почувствуете себя настоящими волшебниками, ведь мы будем оживлять буквы.

Занимайте рабочие места за компьютерами.

2. Практическая работа 12 мин.

Запускайте программу.

Перед тем как писать буквы, нашего котика нужно удалить. В области сцены нажимаем на кота правой кнопкой мыши и нажимаем «удалить».



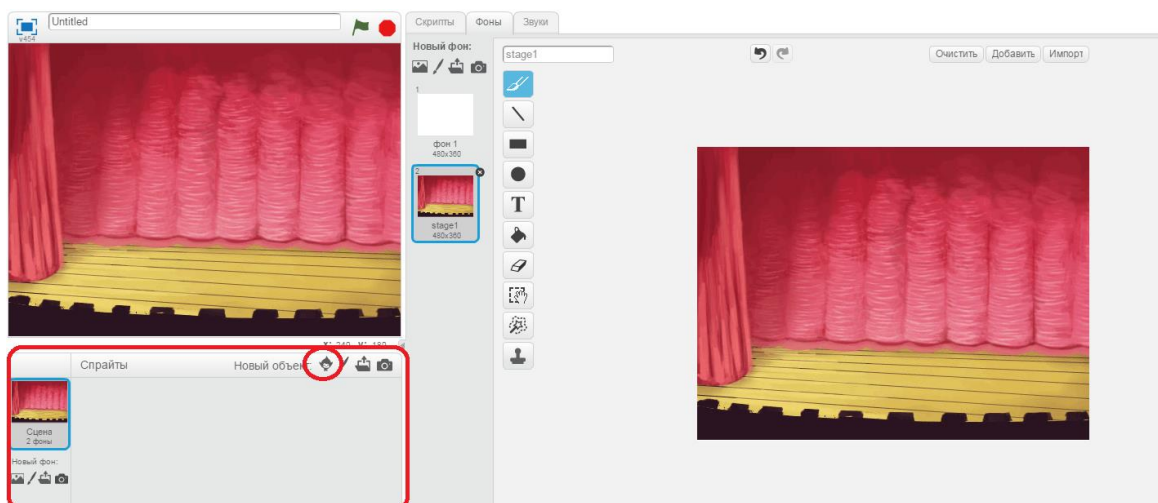
Теперь сцена пустая.

Задание 1. Фон и объекты

Задаем любой понравившийся фон. На предыдущем уроке мы делали это, вспоминайте. *Если есть затруднения, учитель подсказывает.*

Я выбрала розовую сцену.

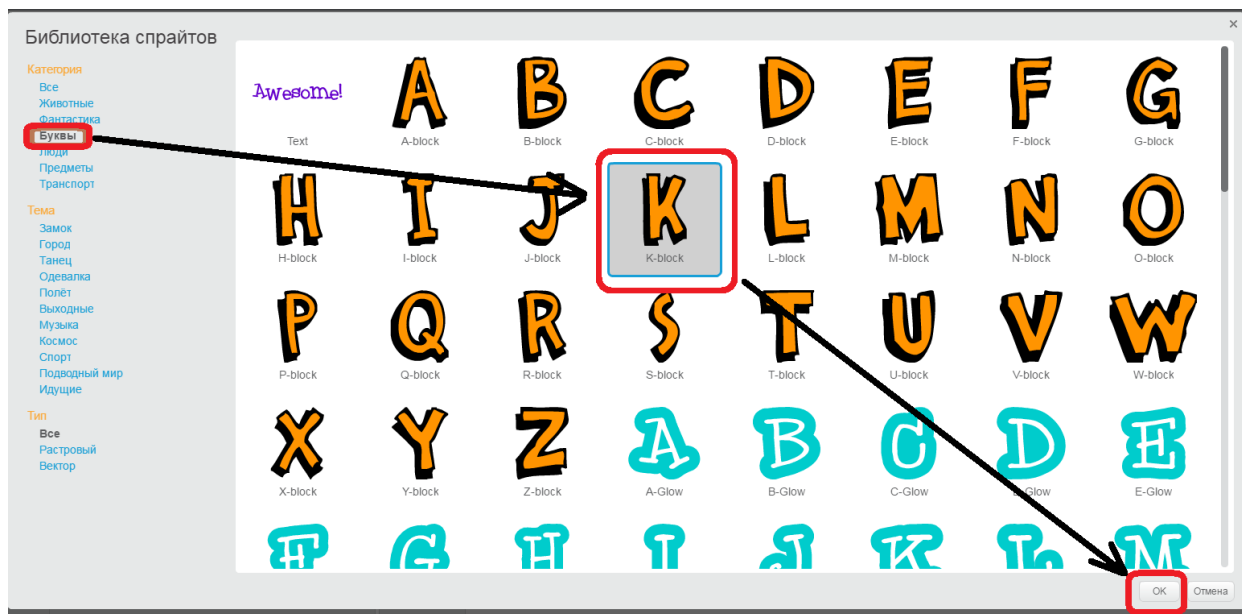
Далее добавляем объекты. Как это сделать? В левой нижней части находим окно спрайтов. Рядом с «Новый объект» находим человечка и кликаем на него.



Откроется библиотека спрайтов. Посмотрите, какие бывают графические объекты.

Слева из списка выбирайте «Буквы». У каждого из вас получатся разные слова. Давайте из букв составим ваши имена. *Учитель помогает детям написать имена на латинице. (Русских букв в программе нет).*

Выбираем букву и нажимаем кнопку «ОК».

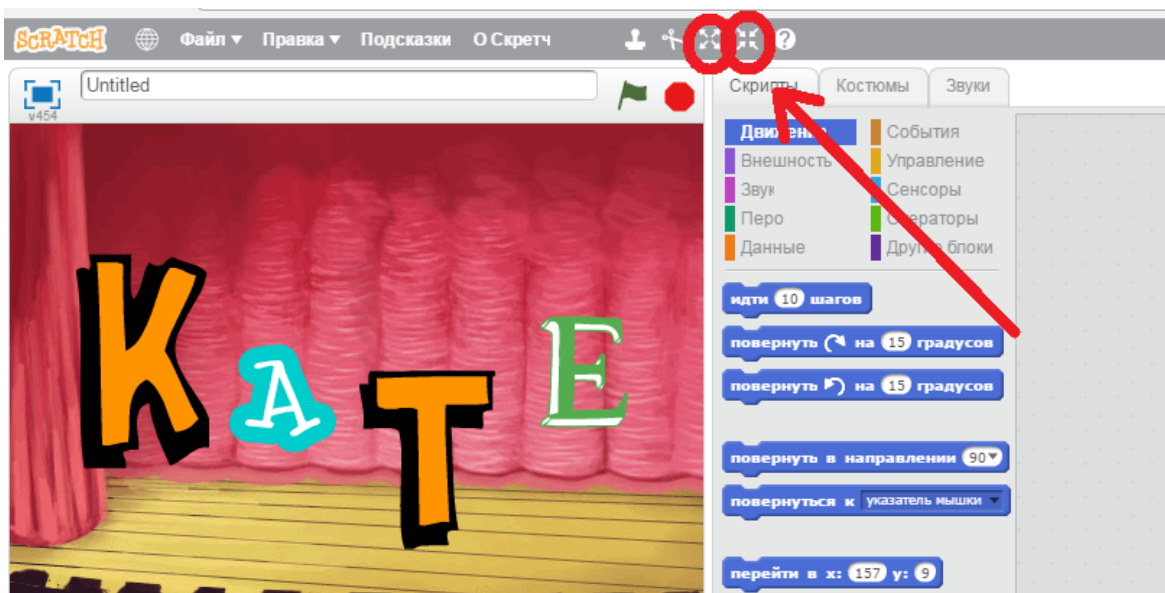


! Самостоятельно добавьте остальные буквы.

После того как все спрайты добавили, вы можете менять расположение и их размер.

Для того чтобы перетаскивать их с места на место, нужно зажимать левую кнопку мыши и не отпуская переместить на новое место.

Для того чтобы изменять размер спрайтов нужно вверху в меню найти кнопочки «Увеличить» и «Уменьшить».



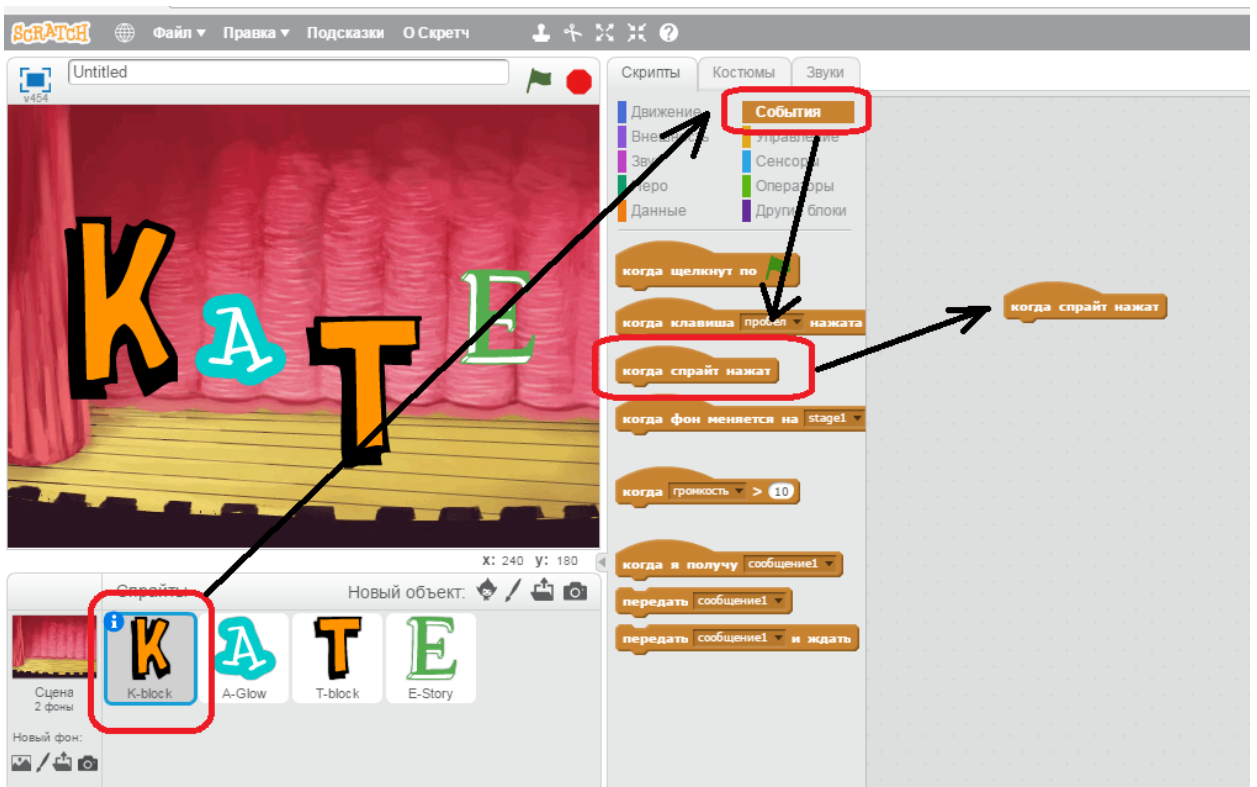
Нажимаем один раз на кнопку, например «Увеличить», ведем курсор на нужную букву и нажимаем на нее несколько раз, пока буква не станет подходящего размера. Поэкспериментируйте с кнопками.

У всех получилось? *Если есть затруднения, учитель помогает.*

Задание 2. Написание скрипта для первой буквы

Чтобы анимировать наши буквы, необходимо каждой написать свою программку (скрипт).

3. Выбираем в области скриптов первую букву (снизу слева).
4. Вверху посередине из раздела «события» выбираем блок «когда спрайт нажат» и перетаскиваем в рабочую область



5. Из раздела «внешность» добавляем блок «изменить цвет эффект на 25». Посмотрите что получается в результате нажатия на данную букву в области сцены?

Ученики:

Буква меняет свой цвет.

Учитель:

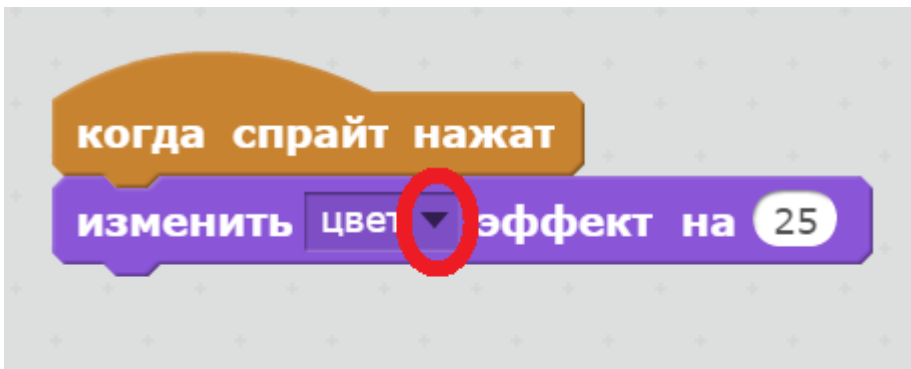
Верно. Попробуйте вместо числа 25 вести цифру 5. В чем разница?

Ученики:

Буква по-прежнему меняет цвет, но переход от цвета к цвету стал плавней.

Учитель:

Молодцы. Теперь на блок «изменить цвет эффект на 25», раскройте список, нажав на стрелочку.



Какие эффекты мы можем применять к спрайту?

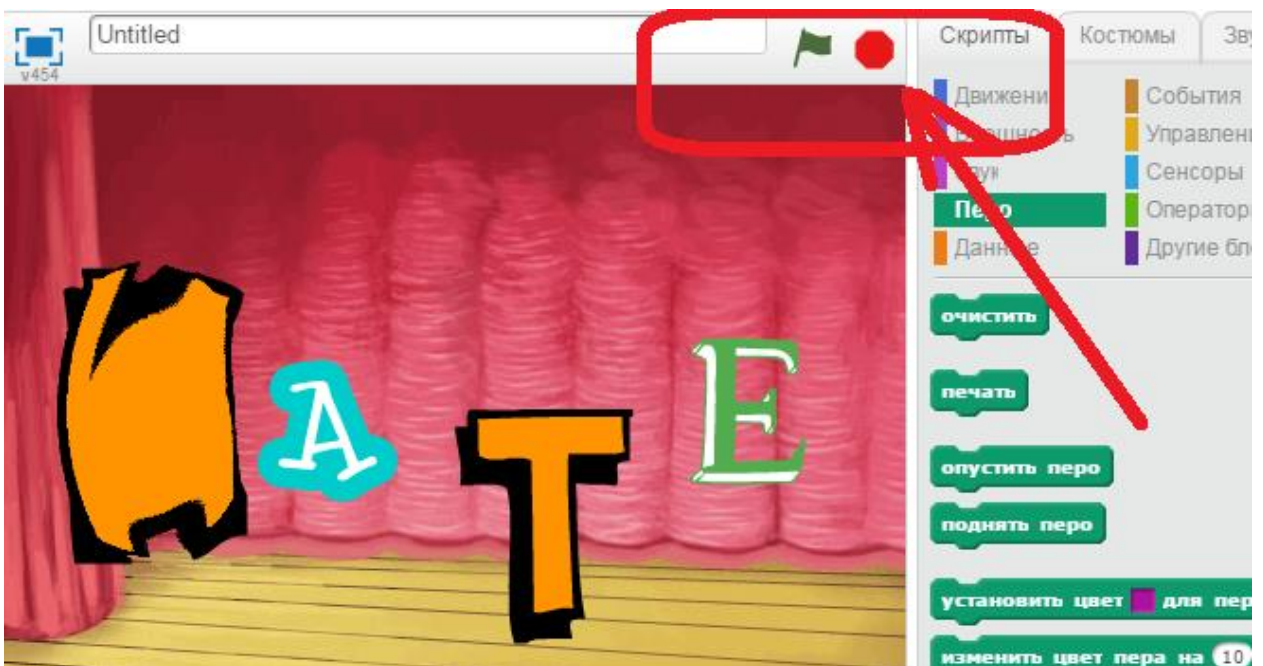
Ученики:

Рыбий глаз, завихрение, укрупнение пикселей, мозаика, яркость, прирак.

Учитель:

Поэкспериментируйте с каждым эффектом.

!Внимание. Чтобы вернуть букву в исходное состояние, необходимо нажать на красный кружок. Эта кнопка стирает все эффекты и действия.



Рядом с красной кнопкой – зеленый флажок. Эта кнопка запускает наш проект.

Давайте оставим из предложенных эффектов – цвет.

Ребята, давайте прервемся и проведем физкультминутку.

3. Физкультминутка 2 мин.

Встаем из-за парт. Повторяем за мной.

Ручки кверху поднимаем, а потом их опускаем.

А потом их развернем и быстрее к себе прижмем.

А теперь скорей скорей, хлопай хлопай веселей.

Приседай скорей со мной, держим руки за спиной.

Встали прямо, руки вбок, вправо - влево поворот.

Молодцы, давайте повторим еще раз, а вы постарайтесь повторять не только движения, но и слова.

4. Практическая работа (продолжение) 13 мин.

Продолжаем нашу работу, садимся за рабочие места.

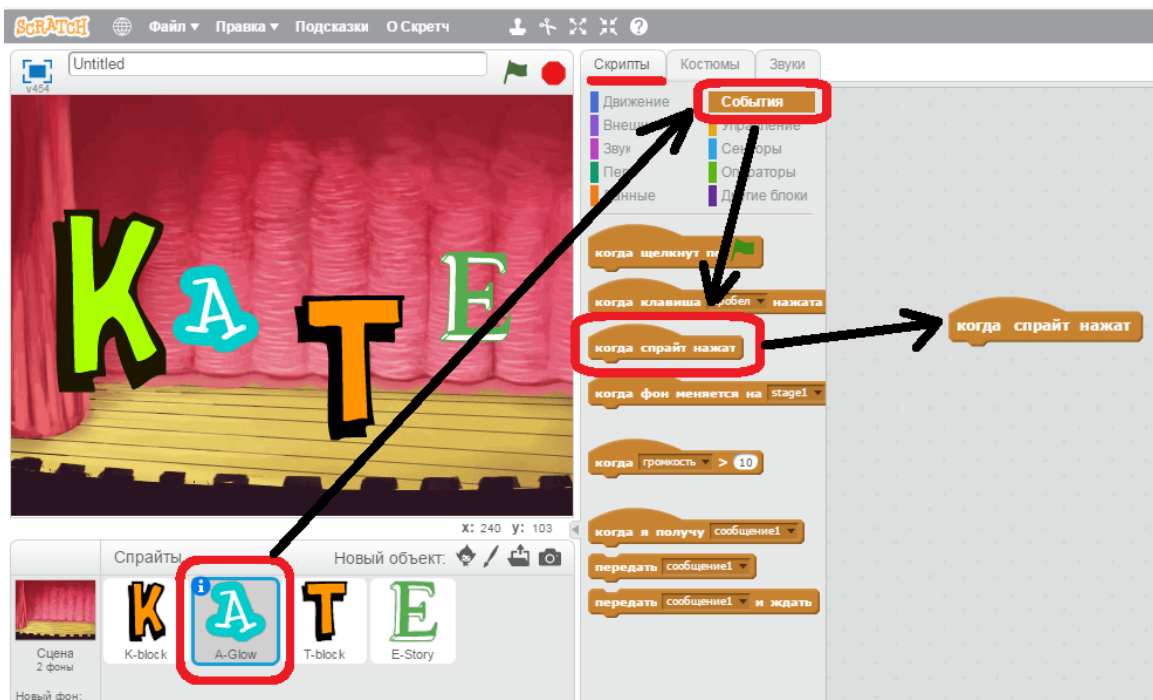
Мы остановились на том, что написали скрипт для первой буквы.

Теперь добавим скрипты для остальных букв.

Задание 3. Скрипт для второй буквы.

Скрипт для второй буквы будет отличаться. При нажатии на спрайт, он будет менять наклон.

1. Для этого выбираем в области спрайтов вторую букву.
2. В окне «Скрипты» выбираем раздел «событие» и блок «когда спрайт нажат». Перетаскиваем в рабочую область.



3. Из раздела «движение» выбираем блок «повернуть вправо на 15 градусов».

4. Посмотрите результат. Измените число 15 на 100. Что получится?

Ученики:

Буква сильно изменит положение.

Учитель:

Верно: 360 градусов – это полный круг. Мы не увидим результат.

180 градусов – это половина круга.

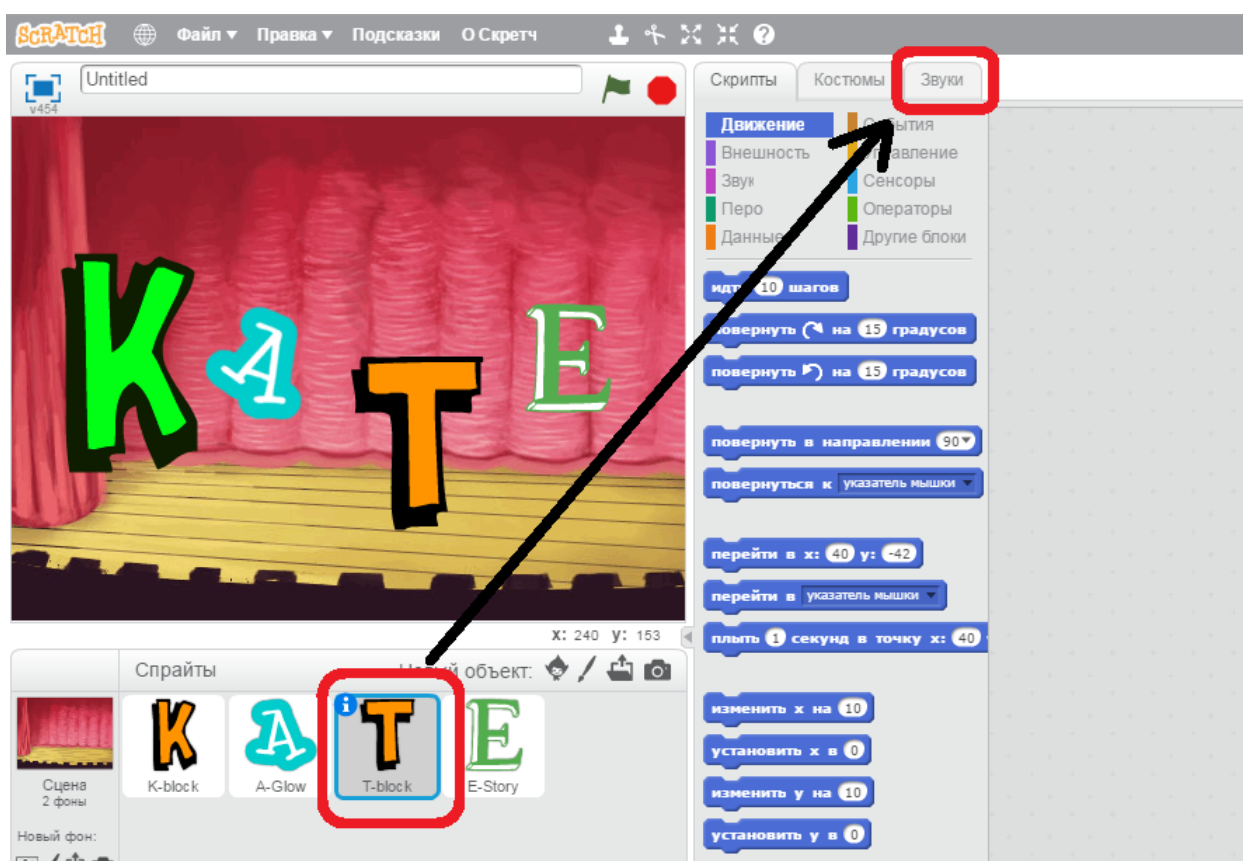
90 градусов – это четверть круга.

Поэкспериментируйте с градусами.

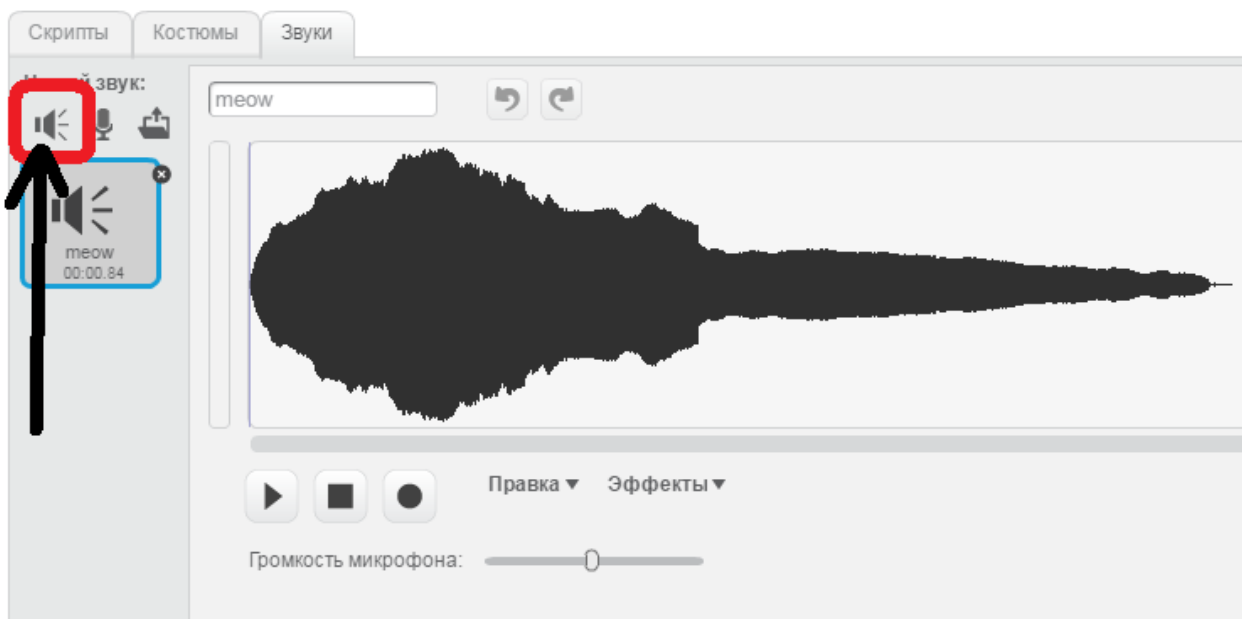
Задание 4. Скрипт для 3 буквы.

К третьей букве мы добавим звук. Помните, на предыдущем уроке мы учили кота мяукать?

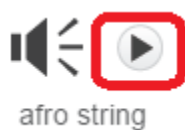
1. В области скриптов выбираем третью букву.
2. Переходим на вкладку «звуки».



3. У нас добавлен только звук «meow». Нажмите на кнопку «библиотека звуков».



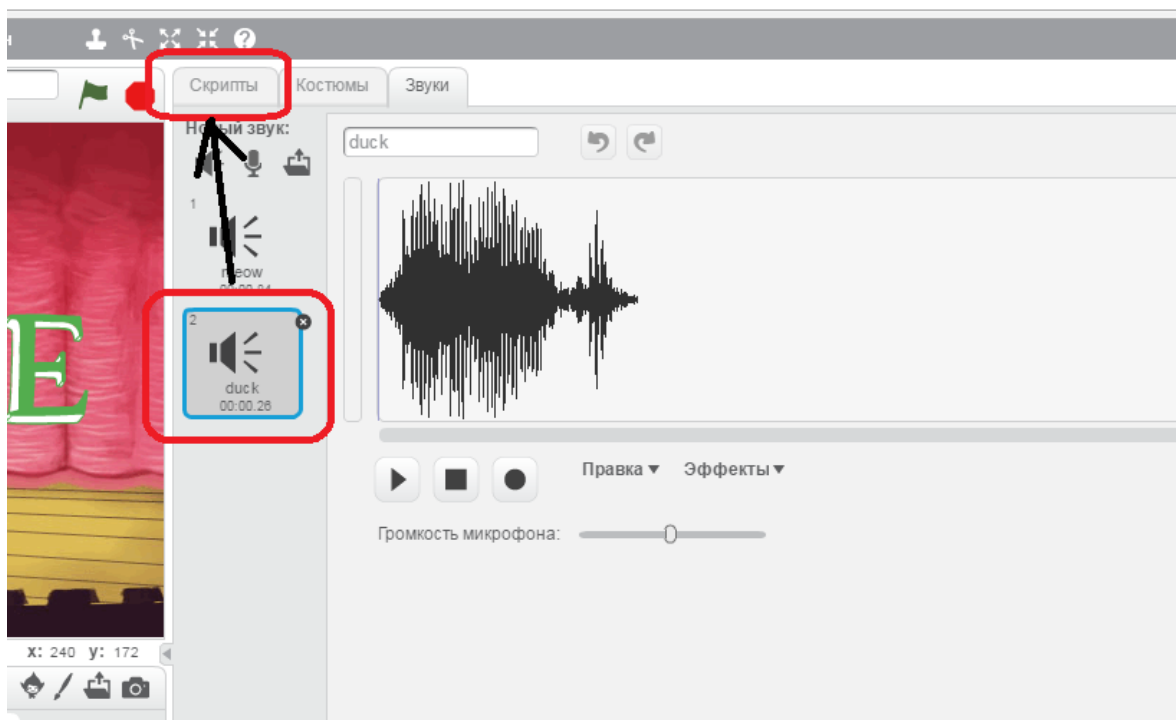
4. Посмотрите какие могут быть звуки. Вы можете их прослушать, нажав на треугольник.



5. Выберите любой понравившийся звук и запомните название. Нажмите на кнопку «ОК».



6. Посмотрите, кроме звука «meow» у вас добавился новый звук.



7. Переходим на вкладку скрипты. Из раздела «события» снова добавляем блок «когда скрипт нажат». Из раздела «звук» добавляем блок «играть звук». Попробуйте в области сцены нажать на третью букву.

Получилось? *Если есть затруднения, учитель помогает.*

! Самостоятельно.

Теперь самостоятельно выберете скрипты для оставшихся букв и напишите их. Изменить цвет буквы, изменить наклон или добавить звук.

После того, как все скрипты написаны, необходимо изменить имя проекта и сохранить на рабочий стол. Кнопка «Файл», «Сохранить как». В окне выбирайте «Рабочий стол» и «Сохранить».

А теперь закрываем программу, нажав на красный крестик в правом верхнем углу.

Учитель:

Ребята, вы сегодня все большие молодцы. Расскажите, чему вы научились?

Ученики:

Научились: добавлять новые объекты (спрайты), задавать звук, менять угол наклона спрайта, изменять цвет объекта.

Учитель: Всем спасибо, урок закончен.