



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

Изучение основ программирования во внеурочной деятельности по информатике

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

Направленность программы бакалавриата

«Информатика. Экономика»


Проверка на объем заимствований:

81,52 % авторского текста

Работа рекомендована к защите
рекомендована/не рекомендована.

«25» март 2018 г.

зав. кафедрой И, ИТ и МОИ

 Рузаков А.А.

Выполнила:

Студентка группы: ЗФ-513-~~094~~-5-1

Ахметжанова Марина Сергеевна



Научный руководитель:

доцент, кандидат педагогических наук,

Давыдова Надежда Алексеевна



Челябинск

2018



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 МЕТО-
ДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ**

Изучение основ программирования во внеурочной деятельности по информатике

**Выпускная квалификационная работа
по направлению 44.03.05, Педагогическое образование (с двумя профилями подго-
товки)**

Направленность программы бакалавриата

«Информатика. Экономика»

Выполнила:

Студентка группы ЗФ-513-094-5-1

Ахметжанова Марина Сергеевна

Проверка на объем заимствований:

_____ % авторского текста

Работа _____ к защите

рекомендована/не рекомендована

« ___ » _____ 20__ г.

зав. кафедрой И, ИТ и МОИ

_____ Рузаков А.А.

Научный руководитель:

к.п.н., доцент кафедры И, ИТ и МОИ

_____ Давыдова Н.А.

Челябинск

2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ В КУРСЕ ИНФОРМАТИКИ.....	7
1.1. Особенности изучения программирования в школе.....	7
1.2. Подходы к обучению программированию в начальной школе.....	12
1.3. Программные среды и языки программирования для младших школьников	14
Выводы по Главе 1	23
ГЛАВА 2. РАЗРАБОТКА ФАКУЛЬТАТИВНОГО КУРСА «ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ SCRATCH».....	24
2.1. Анализ нормативных документов	24
2.2. Факультативный курс «Основы программирования на языке Scratch»..	29
2.3. Программно–методическая поддержка курса.....	51
2.4. Апробация результатов исследования в школе	54
Выводы по главе 2.....	55
ЗАКЛЮЧЕНИЕ	57
СПИСОК ЛИТЕРАТУРЫ.....	58
Приложение	61

ВВЕДЕНИЕ

Информатика имеет очень большое и растущее число междисциплинарных связей, как на уровне концептуального аппарата, так и на уровне инструментария. Многие положения, разработанные в области информатики, рассматриваются как основа для создания и использования информационно-коммуникационных технологий (ИКТ) - одного из важнейших технологических достижений современной цивилизации. Особенность информатики заключается в том, что многие предметные знания и методы деятельности (включая использование инструментов ИКТ) имеют значение для других предметных областей и формируются, когда они изучаются. Вместе с математикой, физикой, химией, биологией, курсом информатики закладываются основы естественнонаучного мировоззрения.

В настоящее время одна из фундаментальных отраслей научного знания, информатики формирует системно-информационный подход к анализу окружающего мира, изучает информационные процессы, методы и средства получения, преобразования, передачи, хранения и использования информации; быстро развивающейся и постоянно расширяющейся области практической деятельности человека, связанной с использованием информационных технологий.

Задача современной школы - обеспечить вход студентов в информационное общество, научить каждого школьника использовать новые массовые ИКТ (текстовый редактор, графический редактор, электронные таблицы, электронная почта). Формирование пользовательских навыков для внедрения компьютера в образовательные мероприятия должно поддерживаться независимой творческой работой, лично значимой для обучаемого. Это достигается за счет информационно-объективной практической работы, сутью которой является заполнение задач в информатике с актуальным предметным содержанием. Только в этом случае проявляется индивидуальность, интеллекту-

альный потенциал стажера, проявляются знания, навыки и привычки, приобретенные в классах, фиксируются навыки самостоятельной работы.

Поэтому даже на самых ранних этапах обучения учащиеся должны получить представление о характере информационных процессов, рассмотреть примеры передачи, хранения и обработки информации в человеческой деятельности, живой природе и технологии, научиться классифицировать информацию, определять общие и специальные, установить ссылки для сравнения, провести аналогию. Это помогает ребенку понять окружающий мир, лучше ориентироваться в нем и сформировать основу научного мировоззрения. Внеурочная деятельность в информатике - это пропедевтический уровень информатики в средней и высшей школе. Целью внеурочной деятельности является - формирование основы компьютерной грамотности для учащегося и введение в терминологию предмета «Информатика и ИКТ» [4].

Основой внеурочной [24] деятельности в области информатики являются такие принципы, как:

- целостности и непрерывности, что означает, что этот этап является важным звеном в единой подготовке общей школы в области информатики и информационных технологий;

- ориентированная на практику, предоставляющая подборку контента, направленную на решение простейших практических задач планирования деятельности, использование языка программирования, поиск необходимой информации;

- принцип обучения в области развития (обучение ориентировано не только на получение новых знаний в области информатики и информационных технологий, но и на активизацию процессов мышления, формирование и развитие у школьников) [24].

В настоящее время в начальной школе действует Федеральный государственный образовательный стандарт второго поколения (ФГОС начального общего образования). Этот документ направлен на формирование личных, метапредметных и предметных результатов обучения [23]. Однако

стандарт не предусматривает отдельного предмета «Информатика и ИКТ». Формирование компетенций студентов в области ИКТ организовано по всем предметам. Это требует от учителей использования новых образовательных методов и организации информационной и образовательной среды. После анализа ФГОС начального общего образования мы обнаружили, что цели развития ИКТ в начальной школе следующие:

- освоение основ логического и алгоритмического мышления, пространственного воображения и математической речи, измерения, перевода, оценки и оценки, визуализации данных и процессов, записи и выполнения алгоритмов;
- получение первоначальных представлений об компьютерной грамотности;
- Формирование первоначальных представлений о свойствах информации, способах работы с ней, в частности, с использованием компьютера;
- формирование информационной культуры ребенка.

Таким образом, мы видим, что изучение основ программирования, начиная с начальной школы, является важным звеном в процессе обучения информатике.

Цель работы: разработать факультативный курс «Основы программирования на языке Scratch».

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 формы и методы обучения основам программирования в начальной школе.
2. Разработать факультативный курс по программированию в среде Scratch для начальной школы.
3. Разработать программно–методическую поддержку курса в виде электронного пособия.

4. Апробировать курс в рамках педагогической практики в МОУ «Тюбукская СОШ №3».

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

Выпускная квалификационная работа состоит из введения, двух глав, заключения и списка литературы.

В процессе выполнения работы основные результаты были апробированы в период педагогической практики в МОУ «Тюбукская СОШ №3» Каслинского муниципального района Челябинской области.

ГЛАВА 1. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ В КУРСЕ ИНФОРМАТИКИ

1.1. Особенности изучения программирования в школе

Компьютерный курс информатики появился в школе вместе с 8-битными бытовыми ПК и BASIC в ROM. В сознании многих из них связана концепция программирования с программами на Бейсике. Позже, когда появились другие возможности, ни авторы учебников, ни учителя не смогли отказаться от программной среды BASIC.

Несмотря на активную критику и формальное неприятие BASIC, большинство задач программирования решаются одинаково - строка в строке - на трех «официальных» языках: BASIC, Pascal, школьный алгоритмический язык. И это свидетельствует о механической передаче задач, придуманных BASIC, для последующих разработок. Даже в новом обязательном минимуме наряду с упоминанием объектного программирования есть фраза: «Знакомство с одним из языков программирования. Переменные: тип, имя, значение. Массивы (таблицы) как способ представления информации». Не строки, списки, стеки, очереди, а именно массивы, поскольку BASIC - это первый пример сложного (структурного) типа данных. Таким образом, подразумевается, что программирование в школе должно изучаться по проторенному пути, повторяя ошибки десятилетий назад.

Особую роль в программировании играют учебные курсы, основанные на языке Logo. Несмотря на всемирное признание LogoTechnologies, в нашей стране этот язык изучается по своему усмотрению для личного использования на официальном уровне (на уровне школы) не принимается [27]. Хотя под давлением «прогрессивного человечества» все еще есть некоторые уступки. Они касаются, в основном, внешней стороны вопроса - графики художника Logo. Возможно, это причина пренебрежения логотипом ортодоксальных программистов. Кроме того, авторы курсов логотипов не учитывают некоторые особенности массовой школы, например, трудности

использования англоязычных версий программ, а материальный аспект обучения не изменился с 1968 года, хотя теперь возможности для программ намного больше. Только при работе в Logo-среде возможно организовать независимую творческую деятельность учащихся среднего звена в массовом порядке. Все остальные программные продукты значительно отстают, по крайней мере, на порядок. Даже попытки использовать так называемые исполнители не могут сравниться с реальной черепахой из-за ограниченного набора задач. Однако, без помощи государства, продукты Logo не будут доступны в массовой школе в течение длительного времени, и учителя, которые не знакомы с ними, не будут заинтересованы в них.

С появлением объектно-ориентированного программирования большинство учителей, с одной стороны, не имели возможности познакомиться с ним, с другой стороны, ООП предназначено для решения проблем, которые не подходят для обучения: они объемны и сложны. Принимая во внимание единство информационных процессов в системах различного характера, можно утверждать, что объектный подход чрезвычайно важен для образования полноценного гражданина, поскольку он нацелен на конечный результат. Он тренирует аналитические и синтетические навыки, способность принимать решения, не говоря уже о том, что в реальной жизни мы не имеем дело с алгоритмами и процедурами, а с объектами, их свойствами и переменными связями между ними. Кстати, традиционный план урока является линейным алгоритмом, и поэтому он не позволяет описать ситуацию творчества на уроке, может возникнуть слишком много проблемных ситуаций, и почти невозможно предусмотреть реакцию на каждый из них. Возможно, это причина неудачи многих прогрессивных учителей составить план - конспекта для каждого урока, и описание опыта в этом стиле выглядит бледным и не передает сути, хотя все внешние стороны описываются с достаточным подробно.

Интересно, что попытки реализовать курс программирования в одной, даже лучшей программной среде, приводят к существенным

методологическим трудностям. Изучение одной и той же среды обучения - это роскошь, которую не могут позволить себе не все учителя.

Анализ ФГОС основного общего образования позволил выдвинуть требования к выпускникам в направлении «Алгоритмизация и программирование» [24]:

На базовом уровне выпускник должен:

- обладать навыками алгоритмического мышления и осознавать необходимость формального описания алгоритмов;
- уметь понимать программы, написанные на универсальном алгоритмическом языке высокого уровня, выбранном для изучения;
- знать основные конструкции программирования;
- иметь возможность анализировать алгоритмы с использованием таблиц;
- обладать стандартными методами написания программ на алгоритмическом языке для решения стандартных задач с использованием базовых программных конструкций и отладки таких программ;
- использовать готовые прикладные компьютерные программы для выбранной специализации.

На продвинутом уровне выпускник должен:

- владеть универсальным языком программирования высокого уровня (необязательно), представлением основных типов данных и структур данных;
- иметь возможность использовать основные контрольные структуры;
- обладать навыками и опытом разработки программ в выбранной среде программирования, включая программы тестирования и отладки;
- владеть элементарными навыками формализации прикладной задачи и документирования программ.

Тема «Алгоритмизация и программирование» [6] - одна из самых сложных тем в информатике. В настоящее время с этим направлением связано множество проблем. Каждый преподаватель информатики отвечает на множество вопросов при построении изучения темы «Алгоритмизация и

программирование»: недостаточные часы обучения информатике старших классов, фундаментальное отсутствие раздела программирования в федеральном стандарте по информатике (как такового, нет собственного раздела «Программирование», он включен в раздел «Алгоритмизация и программирование»). Несколько учебных пособий содержат этот раздел информатики. В то же время в ЕГЭ достаточно много вопросов, особенно в части «С», по пониманию и составлению программ. Кроме того, олимпиада по информатике состоит из задач, направленных исключительно на прикладное программирование. Отсюда возник вопрос: как мы можем подготовить учеников к участию в олимпиадах и сдаче Единого государственного экзамена. Вывод предполагает только одно - сократить время изучения приложений для публичного использования и использовать это время для изучения программирования. Все вышеперечисленные проблемы возникают из-за отсутствия четкого и полного изложения учебных 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.[5], Семанкин И.Г.[12,13,14], Угринович Н.Д.[21], Поляков К.Ю. [10] и Гейн А.Г.

Таблица 1

Анализ учебников по теме «Алгоритмизация и программирование»

Класс	Автор	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 VisualBasic 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. §46-52. ООП

	Семакин И.Г., углубленный уровень	§2.2.1. Паскаль – язык структурного программирования. §2.4.2. Система программирования Delphi.
--	-----------------------------------	---

Как видно из таблицы, авторы учебников используют для программирования 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 разнообразие языков программирования и связанных с ними программных сред.

Вывод возникает сам по себе: раз мы изучаем в школе не конкретные программные системы, а общие приемы и методы познавательной деятельности, то из всего множества доступных разработок следует выбирать только те, которые соответствуют уровню школы.

1.2. Подходы к обучению программированию в начальной школе

Согласно статистике [22], раздел «Алгоритмизация и программирование» является самым нелюбимым и трудным для изучения как школьниками, изучающими информатику, так и преподавателями, которые его учат. Социологические исследования в этой области показывают, что только 5% пользователей ПК работают над написанием разных программ, а остальные 95% просто используют результаты разработчиков.

Конечно, возможно и необходимо вовлечь детей в тему программирования. Как это сделать? Во-первых, при выборе методов и форм обучения необходимо опираться на возрастные характеристики школьников. Возраст младших школьников охватывает период жизни от 6 до 10 лет. В этом возрасте способность воспринимать и наблюдать внешнюю реальность не является абсолютной, они не воспринимают вещи окружающей среды не совсем, подчеркивая случайные особенности и признаки, которые их интересуют по какой-то причине. В связи с этими особенностями учителю необходимо пра-

вильно составить план урока, разбивая урок на отдельные этапы, характеризующиеся сменой действий (урок за партой, гимнастика, урок за ПК и т. Д.). Во-вторых, психологические особенности являются важным компонентом детей, посещающих начальную школу. Детям, которые недавно ходили в детский сад, очень сложно адаптироваться в новой роли ученика. Ведущей деятельностью в настоящее время является обучение, которое является огромной работой для ребенка и требует внимания, дисциплины, волевых усилий ученика. В этой связи учителю нужно отличное воображение в выборе форм и методов проведения занятий. Известно, что дети намного лучше поглощают информацию, если она представлена в виде игры. Следует помнить, что ученики начальной школы - новые выпускники детских садов, которым очень сложно сидеть в течение 40 минут и слушать лекцию по программированию, алгоритмам и т. д. Основные условия подходящей программы: видимость, мультимедиа и простота. В настоящее время существует большое количество различных программных сред и языков программирования, представленных в разном возрасте и уровня подготовки студентов. Основной задачей учителя является правильная выбранная программа и форма занятия. Другая проблема заключается в отсутствии отдельного предмета «Информатика и ИКТ» в некоторых учебных заведениях. В этом вопросе на помощь приходит внеурочная деятельность [24].

Информатика в сочетании с математикой закладывает в образование основные проявления человеческого интеллекта в образовании: способность действовать, размышлять и учиться. Главную роль в курсе информатики играет формирование у обучаемых способности выполнять действия на основе алгоритмического образа мышления, который он сформировал. Человек, который живет в современном информационном обществе, должен обладать алгоритмическим мышлением. Следует отметить, что многие считают, что главная задача учителя информатики - объяснить школьникам, как использовать современные информационные технологии. Конечно, это важно, но при освоении основ алгоритмизации формируется системно-информационная

картина мира, способность различать объекты, процессы и явления, а также способность самостоятельно решать задачи и решать их. Все перечисленные выше функции в некоторой степени влияют на планирование, подготовку и проведение занятия. Основная задача преподавателя информатики в начальных классах - обратить особое внимание на все компоненты: возраст, психологические характеристики младших школьников, а также время для проведения внеурочной деятельности. Только все в совокупности может привести к продуктивной деятельности школьника во внеурочной деятельности.

1.3. Программные среды и языки программирования для младших школьников

Образовательные языки программирования - это языки программирования и среды программирования, предназначенные для изучения основ программирования, алгоритмизации и разработки логико-алгоритмического (оперативного) мышления у детей. На сегодняшний день существует множество различных языков программирования - от самых простых (тех, которые учат детей младших классов), и заканчивается такими мощными инструментами разработки, как C ++, C #, Java, OpenGL и т. д. Однако, некоторые языки программирования не подходят для детей, которые только начинают изучать программирование, потому что они еще не сформировали абстрактного мышления. Языки программирования, предназначенные специально для младших школьников, учитывают их интеллектуальные и психофизические характеристики.

Перечислим некоторые из них:

1. Ligthbot(рис.1) – среда программирования более похожа на логическую игру-головоломку для детей в возрасте от 5 до 8 лет. Суть игры состоит в том, чтобы создать алгоритм действия робота с помощью различных команд, предлагаемых игрой, чтобы он попал на синие квадраты и зажег лампочку. Это действительно сложная задача. Сложность игры заключается в том, что вам необходимо минимизировать размер программы, поскольку па-

мать робота фиксирована и находит оптимальный маршрут для перемещения робота. Несмотря на простоту, с помощью LightBot можно сформировать оперативное мышление ребенка, а также сформировать концепцию программы, процедур, циклов и т. д. Обзор форумов показывает, что среда популярна не только среди детей, но и среди взрослых программистов. К недостаткам этой среды относятся подключение к Интернету [18]. Программа бесплатна, вы также можете играть в нее онлайн.

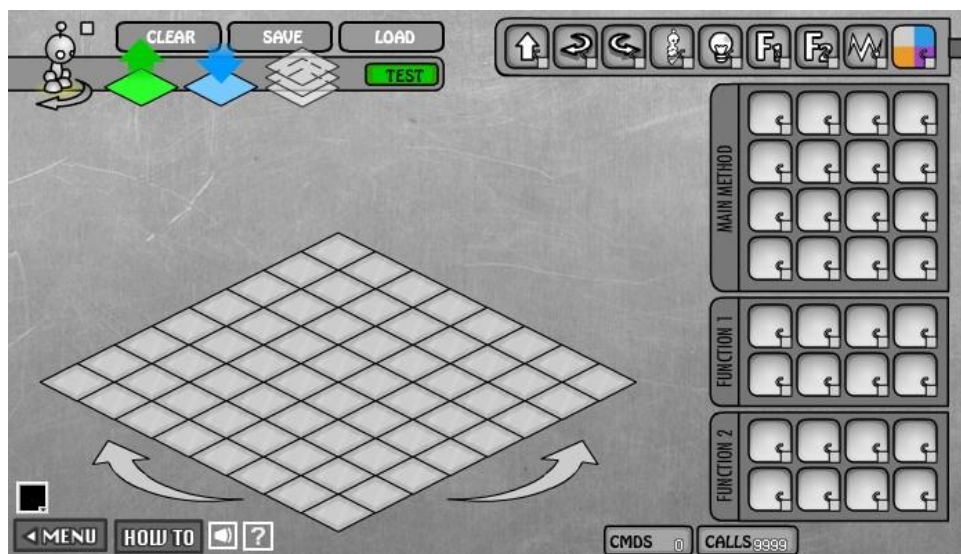


Рис.1.Lightbot

2. **RoboMind**(рис.2) – Это среда для обучения программированию, для детей 8-12 лет, реализована с использованием простого языка сценариев. В настоящее время среда программирования RoboMind локализована на русском языке и ориентирована на изучение языка программирования ROBO, который использует базовые команды и управляет конструкциями программирования для управления поведением роботов. Основные команды программы могут быть введены либо вручную, либо с помощью пульта дистанционного управления, который автоматически выполняет их. Поле для перемещения робота - это карта, которая загружается заранее для решения определенных задач. Результатом программ может быть рисунок, который робот рисует с помощью кисти, найденного «белого пятна» или тупика в стене лабиринта, робота, движущегося вдоль нарисованной линии или найденного выхода из лабиринта [16]. Для домашне-

го использования программный продукт свободно распространяется, для других целей доступна коммерческая лицензия.

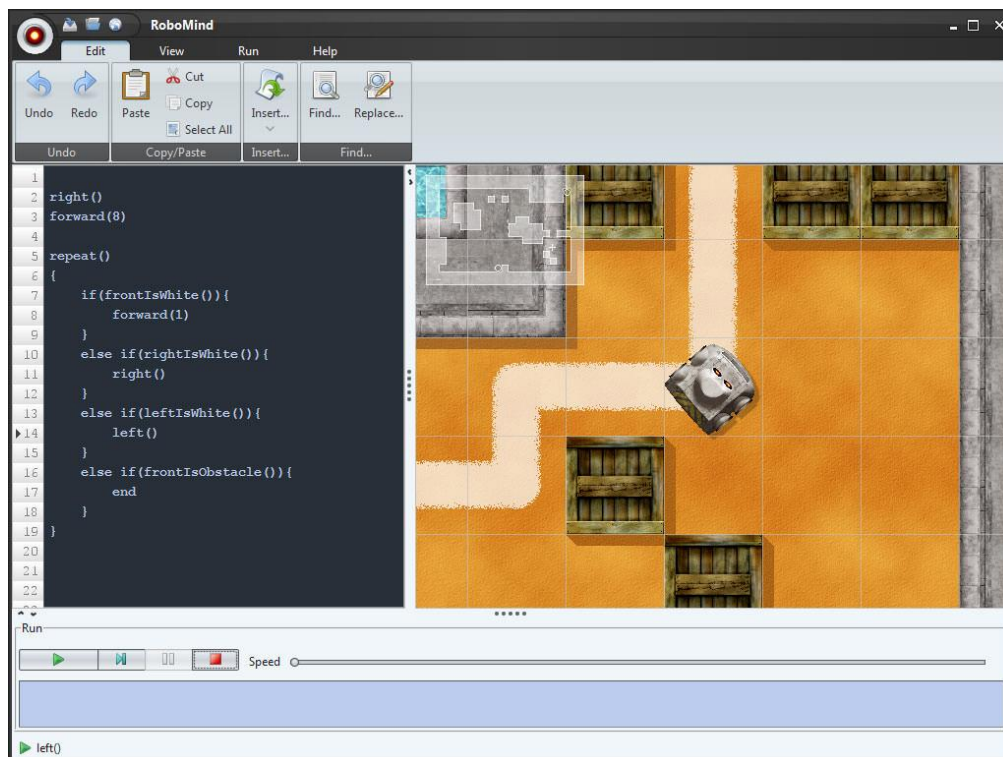


Рис.2. RoboMind

3.LittleWizard (рис.3) – среда разработки для будущих программистов - учащихся начальной школы [19]. LittleWizard служит для обучения детей основам знаний об основных элементах реальных компьютерных языков. Чтобы облегчить освоение базовых концепций программирования, каждый элемент языка является интуитивно понятным значком, который позволяет легко распознать его. Процесс программирования очень прост, так как элементы языка представлены в виде интуитивных символов. Программа бесплатна.

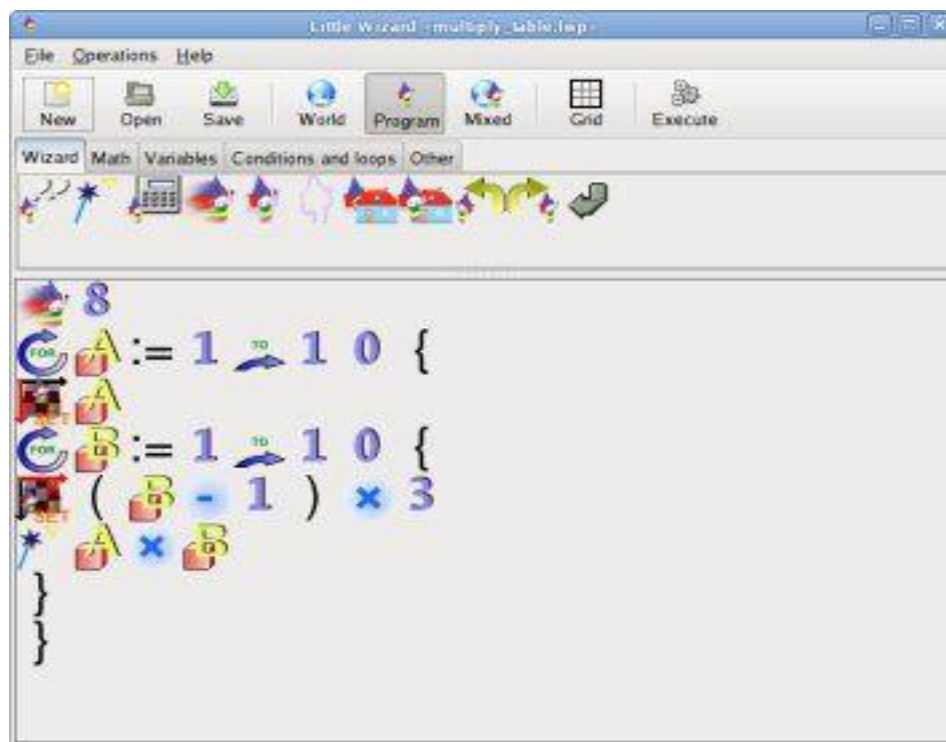


Рис. 3 LittleWizard

4. **Greenfoot**(рис.4) – Это интерактивная среда разработки для языка Java, предназначенная для начинающих программистов, чтобы дать им представление об основах объектно-ориентированного программирования. Она была создана в Университете Дикин, Мельбурн, Австралия и Университете Кента, Великобритания, при содействии Oracle. Программа позволяет создавать простые 2D-приложения, симуляции и игры. Greenfoot полностью бесплатна и распространяется под открытой лицензией GPL. Окно программы разделено на две части: так называемый «мир» слева, на котором расположены «актеры», а справа - дерево классов, в котором показаны все классы Java, участвующие в выбранном сценарии. Классы «мир» и «актер» всегда присутствуют, потому что они системные, другие зависят от выбранного сценария. Вы можете сами создавать сценарии и загружать их. Существует бесплатная версия программы на русском языке [20].

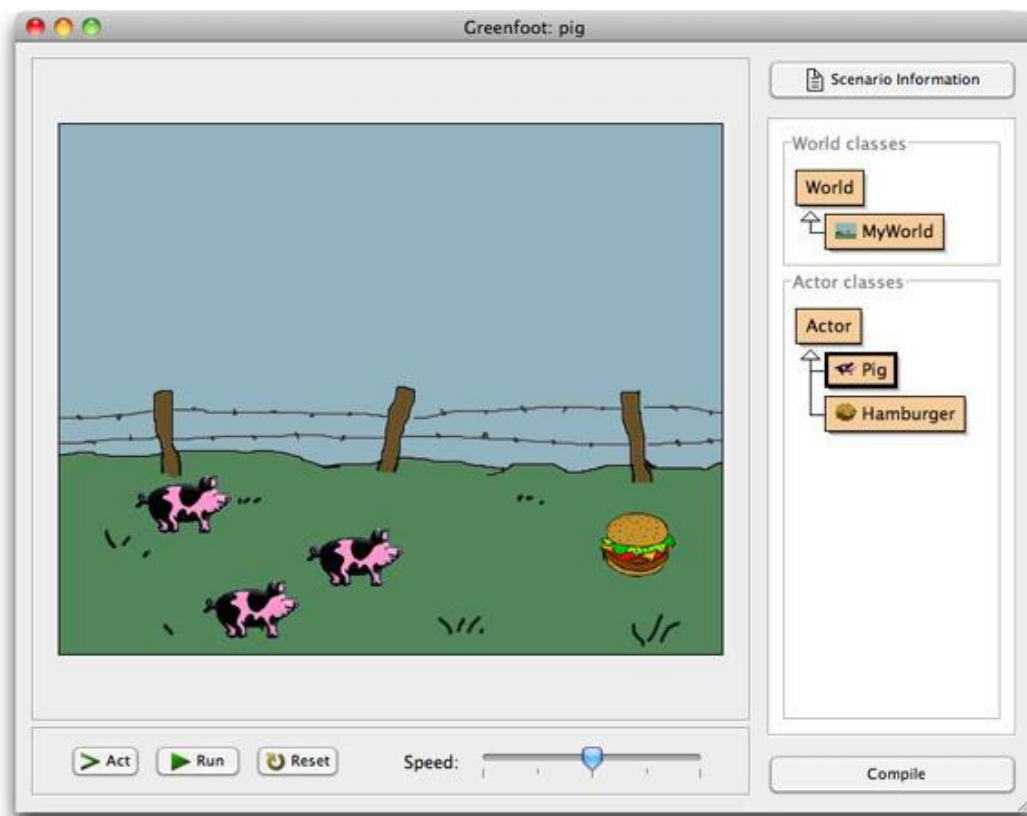


Рис.4 Greenfoot

5. Scratch(рис.5) – Визуальная объектно-ориентированная среда программирования, предназначенная для обучения детей в возрасте от 6 до 14 лет. Scratch создается как продолжение языков программирования Squeak и Logo. Это подойдет тем детям, которые больше не заинтересованы в выполнении знакомых и повторяющихся алгоритмов. В Scratch вы можете создавать интерактивные проекты, составляя блоки алгоритмов и назначая их каждому объекту. Программа включает в себя большую библиотеку животных, героев, различные предметы на любой вкус. Основная цель - познакомить детей с концепциями программирования в процессе создания проектов. Существует международное сообщество пользователей Scratch [17], что позволяет не только создавать проекты, но и делиться новыми идеями и событиями по всему миру. Программа бесплатна, легко устанавливается на любой компьютер.

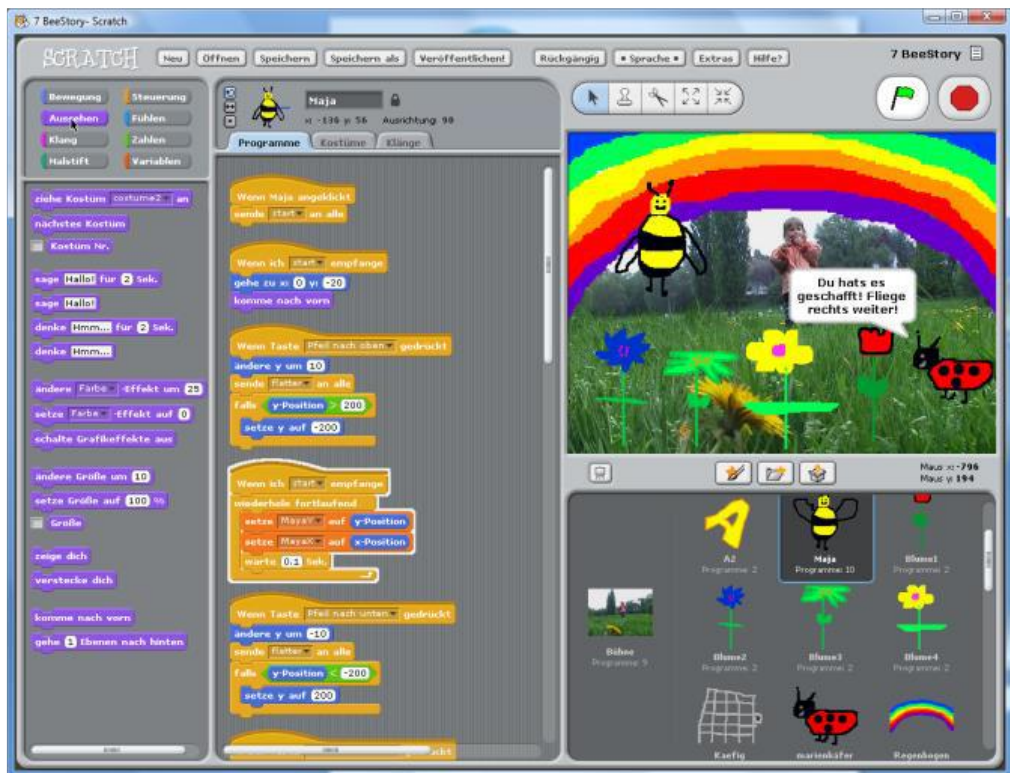


Рис.5 Scratch

6. **Squeak**(рис.6) — язык программирования, диалект Smalltalk. Имеет кросс-платформенную реализацию Windows, Linux, Macintosh. Первоначально Squeak был разработан группой программистов AppleComputer, в которую вошли некоторые разработчики Smalltalk-80. Разработка была продолжена той же группой уже в WaltDisneyImaging. На данный момент Squeak доступен абсолютно бесплатно для любого использования. Кроме того, Squeak полностью доступен в исходном коде (включая виртуальную машину). В Squeak [25] реализовано несколько графических подсистем (в том числе MVC, унаследованный от оригинального Smalltalk-80, в текущей версии не поддерживается, работает в версиях под 3.8). Однако основной является собственная графическая подсистема Morphic (портирована из Self). В дополнение к самой среде, по открытой лицензии, распространяется книга SqueakbyExample. Недостатки включают низкую производительность виртуальной машины, довольно своеобразный GUI и небольшую документацию.

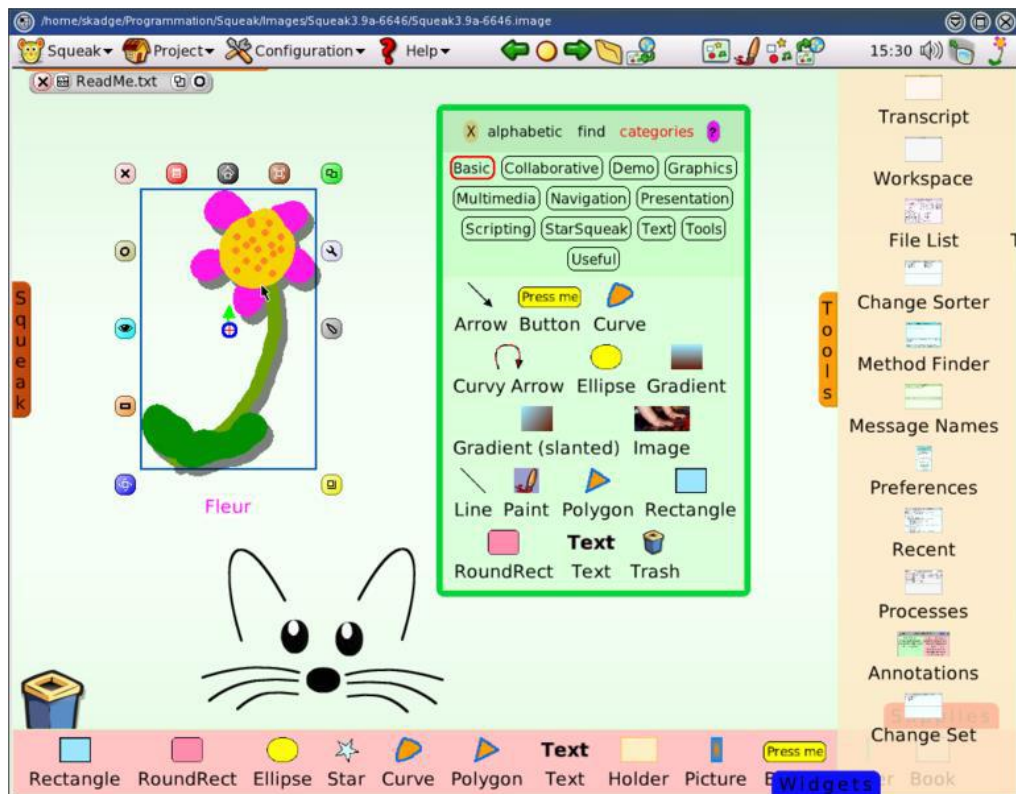


Рис.6 Squeak

7. **Baltie**(рис.7) — визуально преобразованный язык программирования, используемый в образовательных целях. Он основан на C-подобном языке Baltazar, но все команды заменяются значками, в результате чего среда разработки становится визуальной. Baltie 4 основан на C # и позволяет вам использовать текстовые команды вместо значков.

Все три версии языка Baltie [26] были разработаны SGP Systems с 1996 года. Язык полностью патентован; демо-версии Baltie 2 и Baltie 3 доступны бесплатно. «Версии» языка соответствуют уровню абстракции, используемому при создании программ. Baltie 2 дает программисту только ручное управление магическим спрайтом и является скорее игрушкой, чем полноценным языком программирования. В Baltie 3 базовое программирование доступно в процедурной парадигме с использованием значков команд из довольно обширного набора, включая математические и логические функции, функции для работы с файлами и строками, структуры управления и средства взаимодействия с пользователем. Baltie 4 позволяет «реально» программировать, с

текстовыми командами C # вместо значков и поддержкой объектно-ориентированной парадигмы и многопоточности.

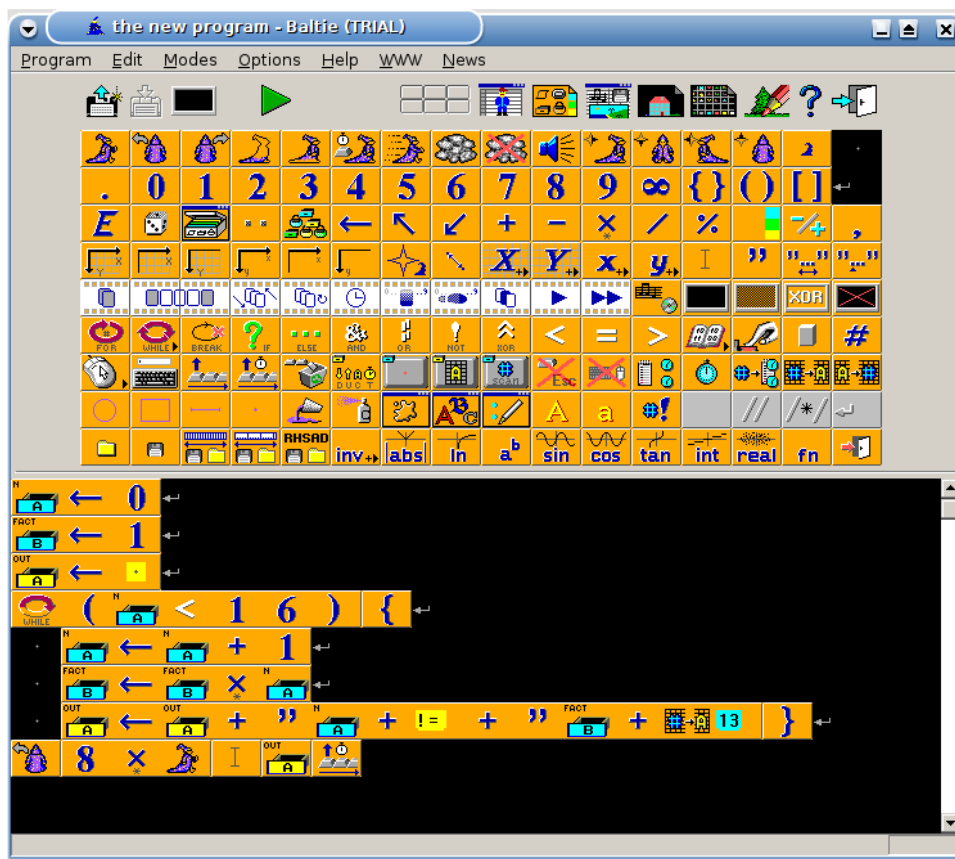


Рис.7 Baltie

8. Logo(рис.8) – язык программирования, предназначенный для детей дошкольного и младшего школьного возраста. Этот язык используется при работе в системе LogMira. С черепахой или любым другим персонажем, выбранным ребенком, легче понять основную логику программ из-за «направления», которое можно изменить. Обучение черепах для любого действия побуждает ребенка понимать свои действия и мысли, что создает более эффективный процесс обучения для программирования. Главный недостаток: отсутствие бесплатной версии, что приводит к значительному сокращению учебных заведений, которые могут позволить себе приобрести лицензию на этот продукт [27].

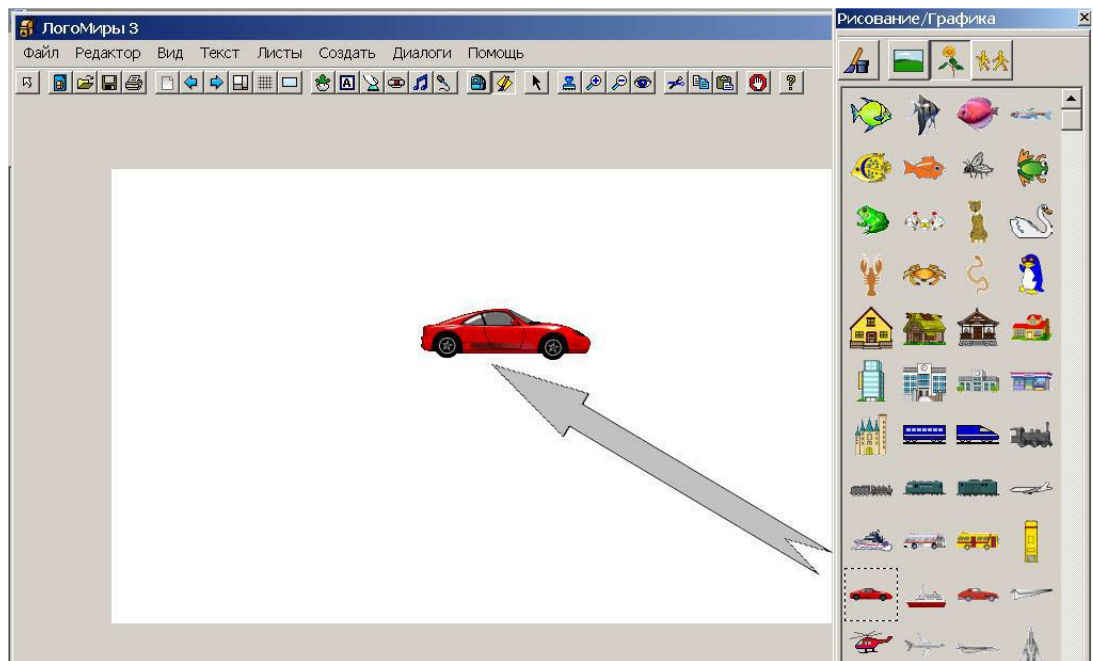


Рис.8 Logo

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 этой главе были рассмотрены вопросы изучения программирования в начальной школе, обоснованы возрастные и психологические характеристики младших школьников. В ходе написания этой главы был проанализирован ряд программных сред и языков программирования для изучения темы «Программирование» в начальной школе: Lightbot, RoboMind, LittleWizzard, Greenfoot, Logo, Batlie, Squeak, Scratch. Мы пришли к выводу, что желательно изучать программы на уровне младших школьников, постепенно включая соответствующие вопросы в тематическое планирование курса информатики. Лучшим программным обеспечением, на мой взгляд, является программа Scratch, простая в освоении, очень интересная и имеющая достаточную функциональность. Программа бесплатна. Теоретический материал, рассмотренный в первой главе, послужил основой для разработки факультативного курса «Основы программирования Scratch».

ГЛАВА 2. РАЗРАБОТКА ФАКУЛЬТАТИВНОГО КУРСА «ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ SCRATCH»

2.1. Анализ нормативных документов

Федеральные государственные образовательные стандарты [24] (далее именуемые ФГОС) включают в себя набор требований, которые являются обязательными для учебных заведений, имеющих государственную аккредитацию для реализации базовых 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 контентной линии основных образовательных программ начального общего, основного общего, среднего (полного) общего, начального профессионального, среднего профессионального и высшего профессионального образования. Воспитание считается одним из основных компонентов образовательного процесса наравне с обучением. Важной педагогической задачей является воспитание высокоморального, ответственного, компетентного и предприимчивого гражданина Российской Федерации.

В соответствии с федеральным государственным образовательным стандартом начального общего образования (далее – ФГОС НОО) базовая образовательная программа осуществляется учебным заведением, в том числе посредством внеурочной деятельности [23]. Внеурочная деятельность является не только неотъемлемой частью образовательного процесса, но и одной из форм организации свободного времени школьников, направленной на то, чтобы помочь ребенку каким-то образом преодолеть любые трудности в новой учебной деятельности и создать мотивацию для обучения, интерес в этой дисциплине [23].

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

Цель внеурочной деятельности: создание условий для раскрытия и развития их увлечений и интересов на основе свободного выбора, ознакомления с культурными традициями, моральных ценностей, создания условий для интеллектуального и физического развития детей.

Существуют некоторые принципы организации внеурочной деятельности:

1. Принцип гуманистической ориентации. Потребности, интересы и желания детей полностью учитываются при организации вне образовательной деятельности, кроме того, реализуются условия для формирования умений и навыков учащихся в области самопознания и самореализации.

2. Принцип системы. Создана система внеурочной деятельности школьников, которая устанавливает взаимосвязь между:

- всеми участниками учебного процесса: учащиеся, учителя, родители;
- урочной и внеурочной деятельностью;
- основными компонентами образовательного процесса: целевые, значимые, ориентированные на деятельность и оценочно-эффективные;
- региональные, муниципальные, школьные, классные, индивидуальные системы образования и дополнительного образования для школьников.

3. Принцип вариативности. Определены несколько направлений, форм, методов и методов организации внеурочной деятельности, с помощью которых дети имеют полную свободу выбора и добровольного участия в любой деятельности.

4. Принцип креативности(творчества). Задачи учителя - развивать творческие способности детей во внеурочное время, создавать благоприятные условия для занятий по индивидуальным и коллективным программам.

5. Принцип успеха и социального значения. Стремление учителя направлено на удовлетворение потребности ребенка в достижении успеха и самореализации. По сути, результаты обучения ребенка не только лично ценны, но и имеют отношение к другим, например, к одноклассникам, родителям, учителям, школьной организации и т. д. Время, затрачиваемое на внеурочную деятельность, составляет не более 1350 часов в течение 4 лет обучения [23]. Эти принципы предполагают различные способы организации внеурочной деятельности, которые тесно связаны с основным образованием.

Внеурочная деятельность в ФГОС организована по пяти направлениям личностного развития:

- Духовно-нравственное;
- общекультурный;
- общеинтеллектуальное;
- социальное;
- спортивно-оздоровительное;

Для достижения цели внеурочной деятельности определены следующие задачи:

1. Формирование личности каждого младшего ученика в творческой и художественной среде и самоопределение.
2. Формирование детской инициативы.
3. Воспитание патриотизма и уважение национальных ценностей.
4. Популяризация здорового образа жизни.
5. Привлечение школьников к выполнению самостоятельных задач.
6. Привлечение студентов к организации научной работы.
7. Формирование интереса к научно-исследовательской деятельности.

Методисты Д.В. Григорьев, П.В. Степанов, Давыдов В.В. [9] считают, что во внеурочную деятельность целесообразно включать:

1. Игровую деятельность.
2. Познавательную деятельность.
3. Активный отдых.
4. Проблемно-ценностное общение.
5. Художественное творчество.
6. Социальное творчество.
7. Рабочая деятельность.
8. Спортивные и развлекательные мероприятия.
9. Туристические и краеведческие мероприятия.

Ожидаемые результаты внеурочной деятельности. Самая сложная задача - определить продуктивности внеурочной деятельности. Однако, существуют различные методы, упрощающие анализ результатов (интервью, тестирование, опросы, конкурсы, конкурсы, конкурсы, анализ портфолио ребенка и др.).

В Федеральном стандарте [23] представлены следующие ожидаемые результаты обучения во внеурочное время:

- создание оптимальных условий для развития и воспитания учащихся;
- творческое самоопределение и самореализация детей;
- формирование коллективных и организационных навыков;
- психологический комфорт и социальное обеспечение каждого ребенка;
- реализация игровой истории в образовательной деятельности детей.

Предметная область «Математика и информатика»

Федеральный государственный образовательный стандарт начального общего образования[23] отражает требования к предметной области «Математика и информатика», обеспечивая изучение темы «Алгоритмизация и программирование». Рассмотрим основные положения этой области.

Метапредметные результаты:

- формирование информационной и алгоритмической культуры; формирование представления компьютера как универсального устройства для

обработки информации; развитие основных навыков и навыков в использовании компьютерных устройств;

- формирование идеи основных изучаемых понятий: информации, алгоритма, модели и их свойств;

- разработка алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; разработка навыков для составления и записи алгоритма для конкретного исполнителя; формирование знаний об алгоритмических построениях, логических значениях и операциях; знакомство с одним из языков программирования и базовыми алгоритмическими структурами - линейными, условными и циклическими;

- формирование навыков для формализации и структурирования информации, умение выбирать, как представлять данные в соответствии с задачами - таблицы, диаграммы, графики, диаграммы, используя соответствующие инструменты обработки данных программного обеспечения;

- формирование навыков и умений безопасного и надлежащего поведения при работе с компьютерными программами и в Интернете, умение соблюдать нормы информационной этики и права.

Предметные результаты:

- освоение основ 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. Факультативный курс «Основы программирования на языке Scratch»

Одним из фундаментальных компонентов в информатике является программирование, особенно для обучения и подготовки любого ИТ-специалиста. Кроме того, первоначальная информация о программировании включена в образовательные программы гуманитарных специальностей, такие как социология и психология. Это означает, что разработчики программ и образовательных стандартов не сомневаются, что необходимо изучать раздел «Программирование» даже гуманитариями, хотя и на начальном уровне. Но для обеспечения высоких результатов изучения программ в старшей школе или в высшем образовании необходимо заложить прочную основу на начальном этапе обучения, а именно в начальной школе. Это подтверждает актуальность факультативного курса «Основы программирования на языке Scratch».

В настоящее время компьютерная наука начинает изучаться уже в начальной школе. Возникает вопрос: возможно ли учить детей в начальной школе алгоритмизации и программирования?

Это возможно, но при соблюдении следующих условий:

- Для решения алгоритмических задач выбранная предметная область понятна и интересна ученикам;
- Программная реализация среды программирования имеет удобный интерфейс;
- Чтобы создать алгоритм для решения проблемы, визуальные средства используются для представления структур данных и структур управления, которые не требуют хранения большого количества вспомогательных слов и синтаксических правил для написания программы.

Такая программная среда, основанная на языке программирования Scratch, была разработана группой ученых из Массачусетского технологического института (Массачусетский технологический институт).

Обучение программистам начальной школы в среде Scratch должно выполняться поэтапно, от простого до сложного.

Для того чтобы дети могли эффективно усваивать знания в новой области, существует потребность в мотивации. Мотивация для учебной деятельности возникает, если они чувствуют личную заинтересованность в приобретении знаний для достижения своей цели. Scratch позволяет создавать проекты, интересные для разных возрастных групп от 8 до 16 лет и старше - интерактивные презентации, мультфильмы, компьютерные игры, учебные программы с использованием графики и мультимедиа.

Предлагаемый курс предназначен для организации внеурочных занятий в начальной школе при поддержке программной среды Scratch. Программа разработана с учетом возрастных характеристик учащихся начальной школы и предусматривает работу в классе компьютерного обучения.

Цель: сформировать у учащихся основные идеи о языках программирования, алгоритме, исполнителе, способах записи алгоритма.

Задания:

Обучающие:

- Обучение основным базовым алгоритмическим конструкциям.
- Обучение навыкам алгоритмизации задачи.
- Освоение основных этапов решения задачи.
- Обучение навыкам разработки, тестирования и отладки простых программ.

Развивающие:

- развивать познавательные интересы школьников;
- развивать творческое воображение, математическое и образное мышление студентов;
- развивать способность работать с компьютерными программами и дополнительными источниками информации;
- Развивать навыки планирования проектов, способность работать в группе.

Воспитывающее:

- воспитывать интерес к информатике;
- воспитывать культуру общения между учащимися;
- воспитывать культуру безопасной работы при работе на компьютере;
- воспитывать культуру работы в глобальной сети.

Занятия проводятся группами (от 10 до 15 человек), рассчитанные на 30 минут один раз в неделю. Срок освоения программы: 1 полугодие. Размер курса: 17 уроков.

Формы и методы обучения определяются возрастом учащихся. При проведении классов используются компьютеры с установленным Scratch, проектор, сканер, принтер, компьютерная сеть с доступом в Интернет. Теоретическая работа чередуется с практической работой, а также используются интерактивные формы обучения.

Формы проведения занятий: беседы, игры, практические занятия, самостоятельная работа, викторины и проекты.

Использование метода проекта позволяет обеспечить условия для развития навыков детей в самостоятельной постановке задач и выборе оптимального решения для их решения, анализ результатов, полученных с точки зрения решения задачи.

В программе предусмотрены методы обучения: пояснительно-иллюстративные, частично-поисковые (вариативные задачи), творческие, практические.

Программа реализуется на основе следующих принципов [23]:

1. Обучение активной познавательной деятельности. Все темы учащиеся учатся на практике, выполняют различные творческие задания, общаются в парах и группах.

2. Индивидуальное обучение. Обучение студентов работе на компьютере позволяет организовать деятельность студентов с индивидуальной скоростью и в индивидуальном порядке. Этот принцип реализуется посредством организации семинара по развитию навыков работы с компьютером.

3. Принцип природосообразности . Основная деятельность школьников - игра, в которой дети учатся окружающему миру, поэтому элементы игры включены в уроки, которые способствуют успешному развитию курса.

4. Преемственность. Программа курса структурирована так, что каждая последующая тема логически связана с предыдущей. Этот принцип помогает ученикам понять важность уже изученного материала и важность каждого отдельного урока.

5. Целостность и непрерывность, что означает, что этот этап является важным звеном в единой подготовке общеобразовательной школы в области информатики и информационных технологий. В рамках этого этапа подготовки продолжается вводная и ознакомительная подготовка школьников, предполагающая более глубокое изучение предмета в 8-9 (основной курс) и 10-11 (профильные курсы).

6. Практическая ориентация, обеспечивающая подбор контента, направленного на решение простейших практических задач планирования деятельности, поиск необходимой информации и инструментарий для всех видов деятельности на основе общепринятых информационных мероприятий, реализующих основные пользовательские возможности информационных технологий. В то же время отправной точкой является то, что компьютер может многократно использовать возможности человека, но не заменять его.

7. Принцип дидактической спирали как наиболее важный фактор структурирования в методе преподавания информатики: сначала общее знакомство с понятием, с учетом опыта обучаемых, затем его последующее развитие и обогащение, которое создает предпосылки для научного обобщения в старших классах.

8. Принцип обучения в области развития (обучение ориентировано не только на получение новых знаний в области информатики и информационных технологий, но и на активизацию процессов мышления, формирование и развитие обобщенных видов деятельности среди школьников и формирование навыков для самостоятельной работы).

Требования к результатам обучения

Личностные результаты:

- широкие познавательные интересы, инициативность и любопытство, мотивы познания и творчества; готовность и способность учащихся к саморазвитию и реализации творческого потенциала в духовной и субъектно-продуктивной деятельности за счет развития их образного, алгоритмического и логического мышления;
- готовность улучшить свой образовательный уровень и продолжить обучение с использованием средств и методов информатики и ИКТ;
- интерес к информатике и ИКТ, стремление использовать эти знания в процессе обучения другим предметам и жизни;
- способность связывать учебный контент с собственным жизненным опытом и личными значениями, понимать важность обучения информатике и ИКТ в контексте развития информационного общества;
- готовность к независимым действиям и действиям, ответственность за их результаты; готовность к индивидуальной и коллективной информационной деятельности;
- способность избирательно относиться к полученной информации благодаря навыкам ее анализа и критической оценки; ответственное отношение к информации с учетом правовых и этических аспектов ее распространения;
- Способность и готовность принимать ценности здорового образа жизни посредством знания основных гигиенических, эргономических и технических условий для безопасной эксплуатации объектов ИКТ.

Метапредметные результаты:

- овладение навыками организации своей собственной учебной деятельности, в том числе: постановка целей как постановка задачи обучения на основе соотношения того, что уже известно и что требуется для установления;
- планирование - определение последовательности промежуточных целей с учетом конечного результата, деление задачи на подзадачи, формиро-

вание последовательности и структуры действий, необходимых для достижения цели за счет фиксированного набора средств;

- прогнозирование - предвосхищение результата;
- контроль - интерпретация полученного результата, его корреляция с доступными данными для установления соответствия или несоблюдения (обнаружение ошибок);
- исправление - внесение необходимых дополнений и корректировок в план действий в случае ошибки;
- оценка - информированность учащихся о том, насколько хорошо выполнена образовательная и познавательная задача;
- владение основными универсальными информационными навыками: формулирование и постановка проблемы;
- поиск и выделение необходимой информации, применение методов поиска информации;
- структурирование и визуализация информации; выбор наиболее эффективных способов решения проблем в зависимости от конкретных условий;
- независимое создание алгоритмов деятельности в решении проблем творческого и исследовательского характера;
- владение основами продуктивного взаимодействия и сотрудничества со сверстниками и взрослыми: способность правильно, четко и недвусмысленно формулировать мысль в понятной форме;
- способность осуществлять совместную информационную деятельность в команде, в частности при осуществлении проекта;
- способность говорить с аудиторией, представляя ей результаты своей работы с помощью инструментов ИКТ;
- использование коммуникационных технологий в учебной деятельности и повседневной жизни.

Предметные результаты:

- возможность использования терминов «информация», «сообщение», «данные», «алгоритм», «программа»; понимание различий между использованием этих терминов в повседневной речи и в информатике;
- способность составлять линейные, ветвящиеся и циклические алгоритмы управления исполнителями на языке программирования Scratch;
- возможность использования с ними логических значений, операций и выражений;
- освоение понятий класса, объекта, обработки событий;
- возможность формально выполнять алгоритмы, описанные с использованием ветвящихся структур (условных операторов) и повторений (циклов), вспомогательных алгоритмов;
- возможность создавать и выполнять программы для решения простых алгоритмических задач в программе Scratch;
- возможность использования готовых прикладных компьютерных программ и услуг;
- навыки выбора способа представления данных в зависимости от задачи.

В результате образовательной деятельности будет сформирована и разработана необходимая универсальная образовательная деятельность и специальные образовательные навыки для решения различных учебных, познавательных и образовательных задач, которые лягут в основу успешной образовательной деятельности в средних и высших учебных заведениях. Формы подведения итогов реализации внеурочной деятельности: учебно-исследовательская конференция, защита проектов.

Описание учебно–методического и материально–технического обеспечения
курса внеурочной деятельности

Характеристики компьютерного класса

Количество РМУ: 11

Периферийные устройства: Сканер, принтер, проектор, локальная сеть.

Локальная сеть, выход в Интернет.

Операционная система: Windows, Linux

Основная программа: Scratch v 1.4.

Учебно–методическое обеспечения курса внеурочной деятельности.

Рабочая программа курса.

Практические работы.

Разработки игр, викторин.

В таблице 2 приведен календарно–тематический план факультативного курса «Основы программирования на языке Scratch».

Таблица 2

Календарно–тематический план

№	Тема занятия	Кол. часов	в том числе	
			теория	практика
1.	Компьютер в нашей жизни. ТБ	1	1	-
2.	Алгоритмы в жизни человека. Линейный алгоритм	1	1	-
3.	Алгоритмы ветвления	1	1	-
4.	Алгоритмы ветвления в задачах	1		1
5.	Циклы в алгоритмах	1	1	-
6.	Циклические алгоритмы в задачах	1		1
7.	Знакомство со средой программирования Scratch	1	1	
8.	Исполнитель Спрайт. Мини- проект «Аквариум»	1		1
9.	«Спрайт и радуга»	1		1
10.	«Рисуем со Спрайтом»	2		2
11.	«Безопасная дорога»	1		1
12.	«Узоры» (Циклический алгоритм)	2		2
13.	Работа с проектом (Создание соб-	2		2

	ственных проектов на свою тему)			
14.	Защита проекта	1		1
	Итого:	17		

Поурочное планирование курса

Урок 1. Компьютер в нашей жизни. 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 тест, направленный 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 в жизни человека. 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. На уроке была поставлена цель: пополнить базу знаний по теме «Алгоритмы». Ввести новое понятие – линейные алгоритмы, типы алгоритмов. Просмотр презентации по теме: «Типы алгоритмов», способы записи алгоритмов в виде блок-схем. Использовать задания из рабочей тетради: Горячев А.Д. «Информатика в играх и задачах» 4 класс. Домашнее задание: Составить линейный алгоритм в виде блок-схемы. Привести примеры из окружающего мира, где используются алгоритмы. Составить к этим примерам блок-схемы. Например,

«Собираюсь в школу»;

«Собираюсь на рыбалку»;

«Как построить скворечник»

«Как решить задачу по математике»

«Как сделать уборку дома».

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. Алгоритмы ветвления

Тип урока: урок усвоения новых знаний.

Цель урока: создать условия для формирования у учащихся представление об алгоритмах ветвления.

Задачи:

Образовательная: дать первоначальное представление об алгоритмах ветвления; научить формировать алгоритмы ветвления и формально исполнять их; научить выделять главную информацию, анализировать содержание алгоритма.

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

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

Основные понятия: алгоритм, алгоритм ветвления.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Цель урока - формировать понятие алгоритма с ветвлением; научить составлению алгоритмов с ветвлением в виде блок-схем. На данном уроке я постаралась показать следующее: обучающиеся закрепляли свои знания по предыдущим темам: алгоритм, свойства алгоритма, линейный алгоритм, способы записи алгоритма; приобретали новые знания по теме «Алгоритм с ветвлением»; работали над проектом блок-схема «Условный алгоритм в народных приметах». На каждом этапе урока учащиеся выполняли задание, поставленное в начале урока: повторяли основные понятия изученного материала, играя в «Вопрос-ответ», изучали новое понятия «Алгоритм с ветвлением», его запись с помощью блок-схем, углубление знаний по теме «Алгоритмы». На этапе «Физкультминутка» - на закрепление определения «Алгоритм с ветвлением», выполняли упражнение, после того как было проверена истинность или ложь высказывания. На этапе – применение новых знаний и умений в новой ситуации – ученикам предлагается в форме групповой работы над проектом блок-схема

«Условный алгоритм в народных приметах». За счёт привлекательности содержания заданий и подачи учебного материала, повысились возможности обучающихся в достижении поставленных целей на уроке.

Вопросы для контроля:

1. Какой алгоритм называется разветвляющимся?
2. Как начинается любой алгоритм?
3. Как заканчивается любой алгоритм?

Урок 4. Алгоритмы ветвления в задачах

Тип урока: урок применения предметных умений.

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

Задачи:

Образовательная: научить решать задачи с использованием алгоритмов ветвления.

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: алгоритм, ветвление в алгоритмах.

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

Вопросы для контроля:

1. Какой алгоритм называется разветвляющимся?
2. Виды ветвления?
3. Как начинается любой алгоритм?

4. Как заканчивается любой алгоритм?

Урок 5. Циклы в алгоритмах

Тип урока: урок усвоения новых знаний.

Цель урока: создать условия для формирования представления о циклических алгоритмах.

Задачи:

Образовательная: дать первоначальное представление о циклических алгоритмах; научить формировать циклические алгоритмы и формально исполнять их; научить выделять главную информацию, анализировать содержание цикла.

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: цикл, циклический алгоритм, цикл с предусловием, цикл с постусловием.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Ведется закрепление понятия цикла, видов циклических алгоритмов; формируются умения пользоваться операторами цикла, умения решать задачи с использованием цикла. Вводятся понятия: типы циклических алгоритмов, блок-схемы. Предлагается задание: Расставить команды на схеме циклического алгоритма. Используя PowerPoint, создаем презентацию: «Скакалочка», цель которой показать следующее: если установите флажок непрерывный цикл до нажатия клавиши {Esc} и щелкнуть на кнопке ОК, то в этом режиме происходит циклическое повторение показа слайдов. Домашнее задание: привести примеры циклических алгоритмов из литературы.

Вопросы для контроля:

1. Что такое цикл?
2. Чем отличается циклический алгоритм от других видов алгоритмов.
3. Как начинается любой алгоритм?
4. Как заканчивается любой алгоритм?

Урок 6. Циклические алгоритмы в задачах

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: 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. Как начинается любой алгоритм?

Урок 7. Знакомство со средой программирования 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. Исполнитель Спрайт. Мини- проект «Аквариум»

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. Учащимся было дано задание: создать проект «Аквариум». Для этого создали нового исполнителя «Рыба», используя скрипты блока «Движения», привели в движение рыбу. Затем поменяли фон сцены, выбрали подходящий фон. Используя встроенный графический редактор, нарисовали недостающие элементы аквариума: водоросли, камушки. Учащиеся, самостоятельно проделали всю работу по подсказкам из раздаточного материала, и добавив своих исполнителей в проект, разнообразили его.

Вопросы для контроля:

1. Какие существуют способы добавления новых спрайтов?
2. Как изменить угол наклона объекта?
3. Как изменить цвет объекта?

Урок 9. «Спрайт и радуга»

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. Учащимся на занятии было предложено выполнить проект, где Кот ходил по радуге. Для того, чтобы нарисовать первый круг радуги, необходимо взять блок Перо, установить размер пера – 50. После этого, обратимся к блоку Движение и зададим несколько команд на движение. При создании проекта учащимся помогает раздаточный материал для занятия. Самостоятельно, аналогично тому, что было сделано с помощью учителя, добавили еще несколько цветов радуги.

Вопросы для контроля:

1. Какие существуют способы добавления новых спрайтов?
2. Как изменить угол наклона объекта?
3. Как изменить цвет объекта?

Урок 10. «Рисуем со Спрайтом»

Урок 1.

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. Основная задача урока состоит в совершенствовании умений использовать конструкцию «цикл в цикле» при решении учебных задач. В качестве учебных задач можно использовать возможности блока команд ПЕРО. Как только исполнитель получает команду ОПУСТИТЬ ПЕРО, он оставляет отпечаток соответствующего текущему размеру пера. Эту возможность хорошо использовать при рисовании линий, отличных от сплошной. Первая задача может состоять в рисовании исполнителем квадрата, но линия будет не сплошной, а будет состоять из точек размером 15 на расстоянии 20 шагов друг от друга. Ещё одна задача для иллюстрации использования циклической конструкции — это рисование светофора. Решение задачи состоит из двух частей, в каждой из которых используется циклическая конструкция:

1. рисование контура светофора из трёх квадратов;
2. перемещение исполнителя по квадратам и печать костюмов, соответствующих цвету сигнала светофора.

При решении первой части задачи удобно использовать конструкцию «цикл в цикле» для рисования трёх квадратов в качестве основы светофора.

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 2.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для формирования 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. Продолжаю работу, начатую на прошлом занятии, учащимися самостоятельно было выполнено несколько заданий: рисование линий (использовался циклический алгоритм), квадраты, прямоугольники. Получался красивый орнамент из фигур.

Вопросы для контроля:

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

Урок 11.«Безопасная дорога»

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. В конце урока школьники выполняют самостоятельную работу. Путем перемещения исполнителя и многократного запуска программы, исполнитель рисует белые полосы на сером фоне, похожие на «зебру» – место пешеходного перехода. Для реализации этого мини-проекта обучающимся, нужно:

- изменить фон — нарисовать на нём прямоугольник и залить его серым цветом;

- изменить костюм исполнителя, нарисовав красный кружок, для удобства рисования белых линий.

Здесь учащиеся будут использовать ранее полученные знания по работе с инструментами встроеного растрового графического редактора.

В конце урока школьники выполняют самостоятельную работу и могут самостоятельно видоизменить работу.

Вопросы для контроля:

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

Урок 12. «Узоры» (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

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Используя знания, полученные на предыдущем занятии, учащимися были созданы проекты, с использованием фигур. Примеры проектов: «Рисование из кругов (маленького размера) фигуры квадрат», «Забор», «Играем в классики». На изучение данной темы отводится 2 ч.

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 13. Работа с проектом. (Создание собственных проектов на свою тему)

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 ка-

честве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу. На изучение данной темы отводится 2ч.

Темы проектов: «Аквариум», «Говорящий кот», «Зебра», «Светофор», «На поляне», «Воздушные шары», «Кот и радуга», «Лесенка».

Вопросы для контроля:

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

Урок 14. Защита проекта

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. Конечным продуктом проекта является, созданный в среде программирования Scratch – мультфильм. Учащиеся показывают свой мультфильм на компьютере, рассказывая поэтапно, как они создавали свой мультфильм.

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 факультативного курса «Основы программирования на языке Scratch» (рис.9) был разработан сайт с помощью системы "uCoz"[2] - бесплатный хостинг и конструктор сайтов. Одной из наиболее известных и распространенных систем управления дистанционным обучением является "uCoz". Эта система дистанционного обучения позволяет создавать единое учебное пространство для обучающихся и преподавателей курса. Это означает, что Ucoz охраняется законом об авторском праве, но пользователи имеют широкие возможности по его использованию. Основные возможности Ucoz: подходит для организации традиционного обучения; система дистанционного обучения Ucoz является простой, легкой, эффективной, совместимой с различными продуктами, предъявляя невысокие требования к браузеру. На рисунке 9 представлена главная страница программно-методической поддержки курса.

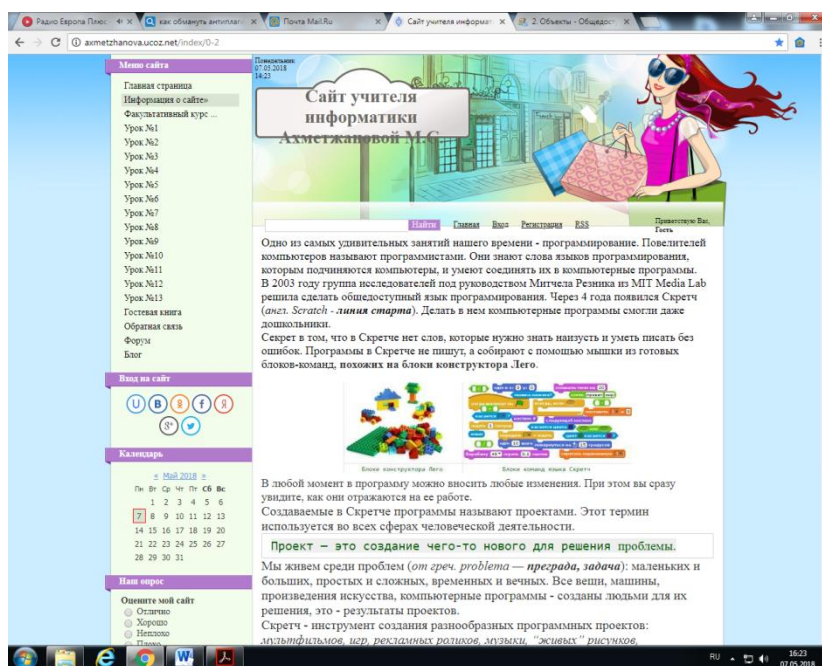


Рис.9 Электронное пособие «Основы программирования на языке Scratch»

На главной странице «Информация о сайте» - представлена статья, рассказывающая о среде программирования Scratch. Вложенная страница «Программные среды» - рассказывает о программах, которые научат ребенка про-

граммировать. Так же представлены такие разделы как о проекте, разработки уроков.

В разделе «О проекте» (рис. 10) находится информация о содержании данного сайта.

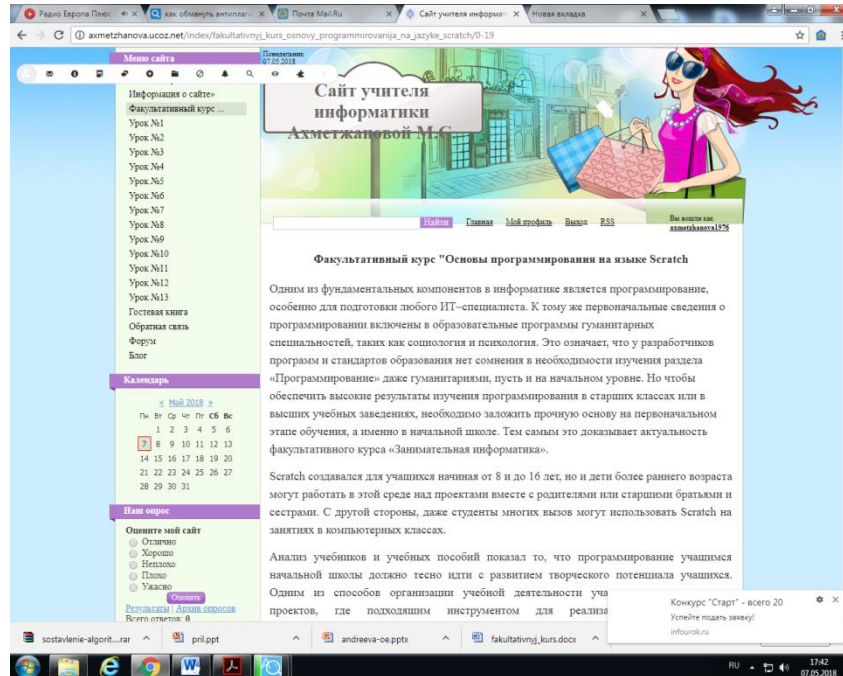


Рис.10 Факультативный курс «Основы программирования на языке Scratch»

В разделе «Материалы для учителя» представлены примерная программа и конспекты уроков в хронологическом порядке (рис.11).

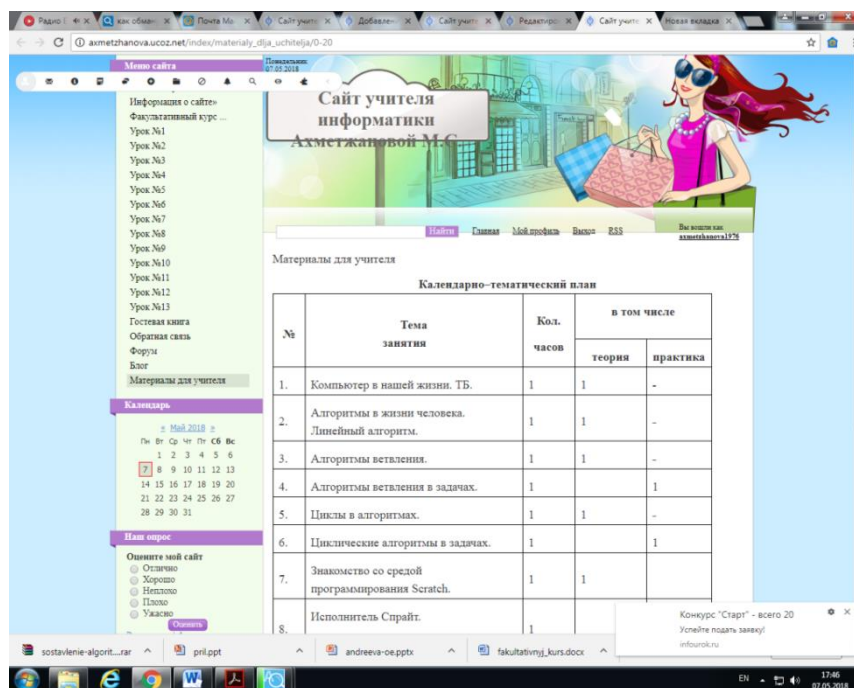


Рис. 11 Конспекты уроков

В разделе «Материалы для учащихся» расположены теоретические (рис.12) и практические сведения о программировании в Scratch.

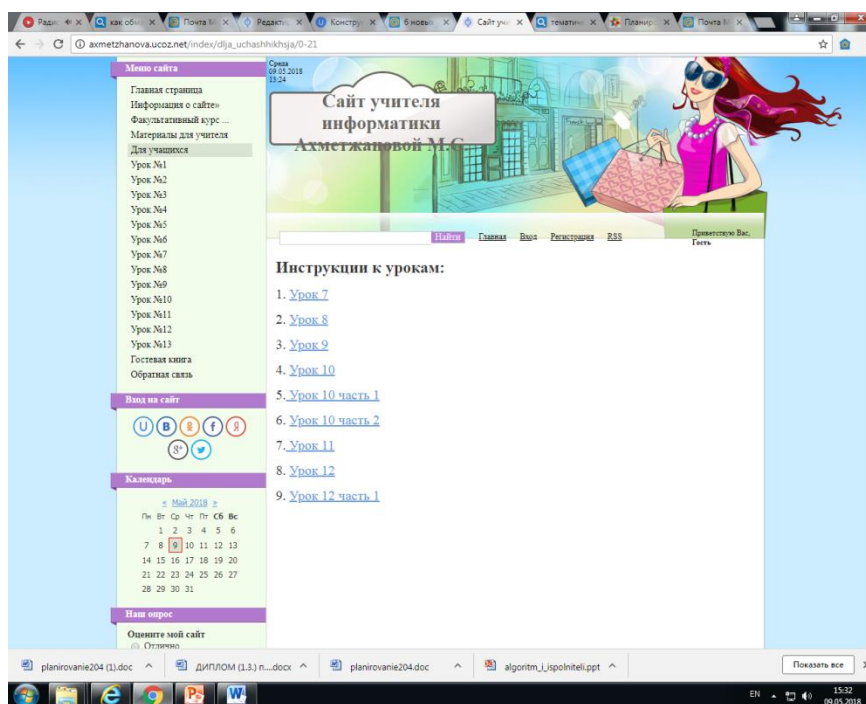


Рис.12 Материалы для учащихся

В разделах «Обратная связь» (рис.12) и «Гостевая книга» (рис.13) представлена информация для организации связи с разработчиком сайта.

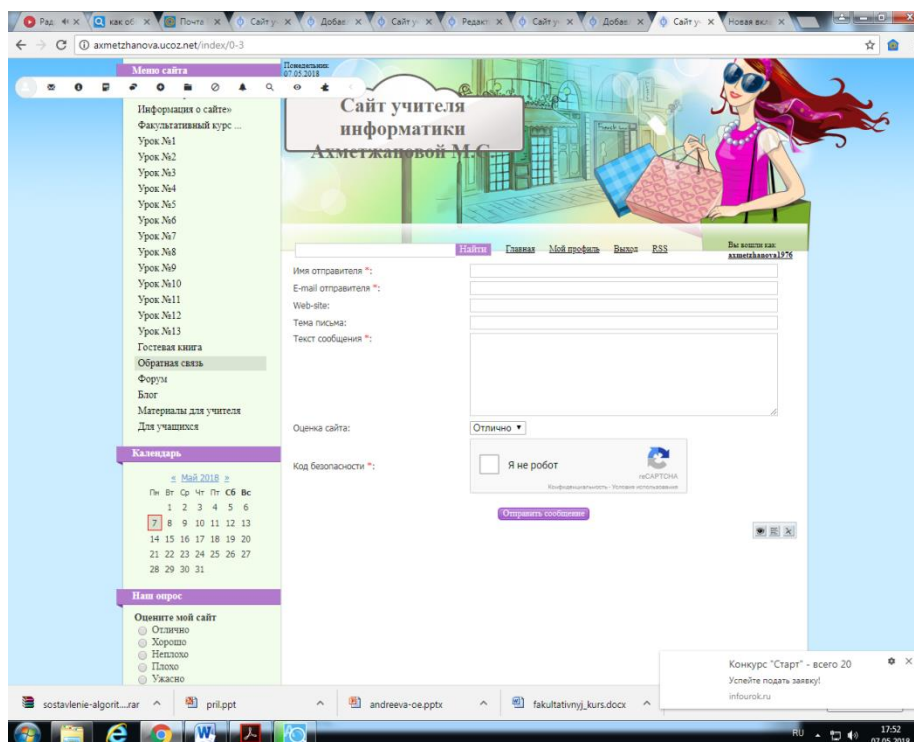


Рис.12 Обратная связь

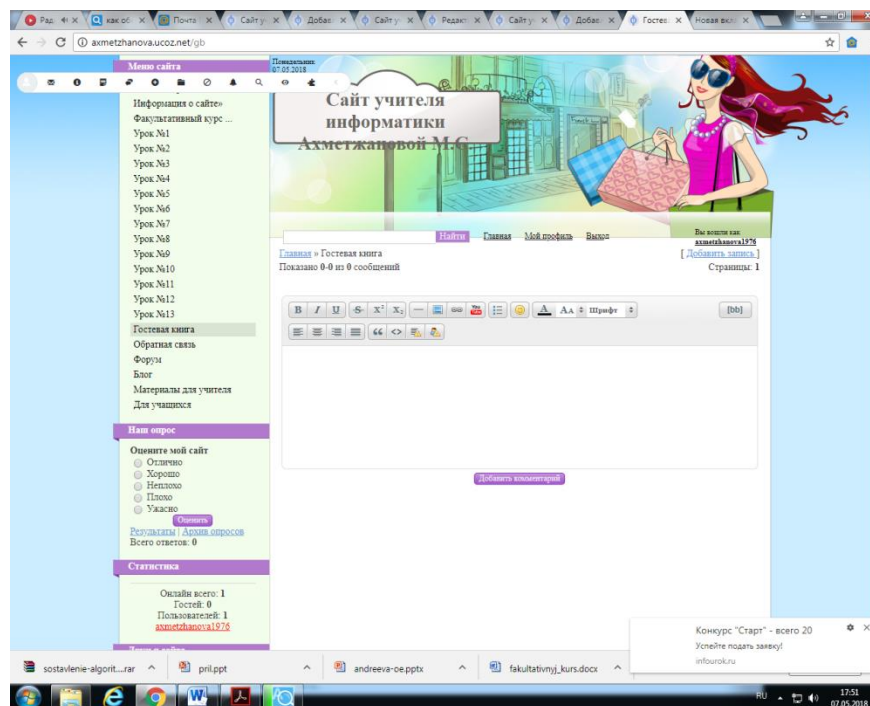


Рис. 13 Гостевая книга

Электронное пособие располагается по адресу:
<http://axmetzhanova.ucoz.net/index/0-2>

2.4. Апробация результатов исследования в школе

Педагогическая апробация проводилась в рамках исследовательской практики в МОУ «Тюбукская школа № 3». Курс изучался в 4 классах. Были изучены следующие темы: «Компьютер в нашей жизни. ТБ», «Алгоритмы в жизни человека», «Линейный алгоритм», «Введение в среду программирования Scratch». Апробация прошла успешно, чему способствовала правильная мотивация, цели и задачи изучения темы. Тема курса была незнакома студентам, но они быстро присоединились к этой работе. В этом возрасте дети особенно заинтересованы в получении знаний, особенно если информация ясна и проста.

Выводы по главе 2

Во второй главе были рассмотрены образовательные стандарты, описан дополнительный курс «Основы программирования Scratch». Этот курс предназначен для учащихся 4 классов. Во время реализации программы используются формы и методы обучения, такие как беседа, демонстрация, мастер-класс и создание самостоятельного проекта. Ученик выполняет задачу, которая имеет смысл, интересна и важна для него.

По теме «Введение в среду программирования Scratch» был создан первый анимационный проект. У учащихся была возможность углубить свои знания и навыки во время создания собственного проекта.

То, что учащиеся делают во время занятий в течение часа, не всегда получается с первого раза, мы должны искать причину неудачной программы для исполнителя, проверять несколько раз: правильно ли написана программа - все это указывает на то, что ребенок сознательно выполняет частичную поисковую деятельность. Я считаю, что во внеурочной деятельности мои ученики в среде программирования Scratch будут шаг за шагом изучать команды программирования и учиться создавать небольшие компьютерные анимации.

Изучение в среде Scratch заставляет учащихся проявлять повышенный интерес к предмету, развивает математическую интуицию и геометрические представления, является своего рода математическим симулятором, формирует алгоритмические, структурные, логические и комбинаторные типы мышления. Учитывая это, занятия внеурочной деятельности в области информатики для младших школьников, а именно их проектная деятельность в среде Scratch, позволят в будущем в тесном взаимодействии с другими предметами использовать интересные материалы по географии, истории, биологии, литературе в своих проектах.

Из анализа ФГОС НОО, мы выяснили, что внеурочная деятельность является не только неотъемлемой частью образовательного процесса, но и

одной из форм организации свободного времени учащихся, направленной на то, чтобы помочь ребенку каким-то образом преодолеть любые трудности в новой учебной деятельности и создание образовательной мотивации, интерес к этой дисциплине [23]. Апробация курса проводилась в рамках исследовательской практики МОУ «Тюбукская СОШ № 3», которая показала, что разработанная программа деятельности способствует развитию алгоритмического мышления учащихся начальной школы и мотивирует к изучению предмета информатики.

ЗАКЛЮЧЕНИЕ

Подводя итог проделанной работе, можно сделать вывод о том, что цель исследования была достигнута, а именно изучить особенности пропедевтики программирования и разработать факультативный курс «Основы программирования на языке Scratch», используя возможности внеурочной деятельности в информатике в начальной школе.

Разработанный курс, предназначенный для проведения занятий в начальной школе, с одной стороны, позволяет осуществить пропедевтику изучения программирования у школьников, проводить лекционные исследования программирования у школьников, содействовать разработке алгоритмического мышления, формированию логико-алгоритмического компонента; с другой стороны, обеспечить учителю освоение новой среды программного обеспечения Scratch.

В ходе исследования были решены следующие задачи:

1. изучены теоретические основы по проблеме исследования;
2. разработан курс по программированию в среде Scratch для начальной школы;
3. разработана программно-методическая поддержка курса в виде образовательного портала.

В подтверждение гипотезы можно сказать, что курс программирования приводит к тому, что ученики осваивают основы логического и алгоритмического мышления.

СПИСОК ЛИТЕРАТУРЫ

1. Аверкин А.Ю., Павлов Д.И. Примерная рабочая программа. Информатика 2–4 классы – М: Бином. Лаборатория знаний, 2016.
2. Бесплатный конструктор сайтов Ucoz [Электронный ресурс]. – <https://www.ucoz.ru/>
3. Босова Л., Босова А. Информатика и ИКТ [Текст]: поурочные разработки для VIII–IX классов // Информатика в школе. – 2013. – №7 (90), сентябрь.
4. Босова Л., Сорокина Т.Е. Методика применения интерактивных сред для обучения младших школьников программированию [Текст] // Информатика и образование. – 2014. – №7(256), сентябрь.
5. Босова Л., Босова А. Информатика [Текст]: Учебник для 6 класса / Л. Босова, А. Босова – М: БИНОМ. Лаборатория знаний, 2016. – 224с.
6. Босова Л., Босова А. Информатика [Текст]: Учебник для 9 класса / Л. Босова, А. Босова – М: БИНОМ. Лаборатория знаний, 2016. – 208с.
7. Газета «1 сентября» [Электронный ресурс]. – <http://festival.1september.ru/articles/534738>
8. Григорьев Д.В., Степанов П.В. Диагностика и мониторинг процесса воспитания в школе [Электронный ресурс]. – <https://studfiles.net/preview/1790554/>
9. Давыдов В.В. «Проблемы развивающего обучения [Текст]: Опыт теоретического и экспериментального психологического исследования» – М: Педагогика, 1986. – 240 с. – (Труды д.чл. и чл.-кор. АПН СССР)
10. Поляков К. Информатика [Текст]: 11 класс. Углубленный уровень: учебник в 2 ч. / К.Ю. Поляков Е.А. Еремин – М: БИНОМ. Лаборатория знаний, 2016. - 203с.
11. Рындак В. Г., Дженжер В. О., Денисова Л. В. Проектная деятельность школьника в среде программирования Scratch [Текст]: учебно-

методическое пособие / В. Г. Рындак, В. О. Дженжер, Л. В. Денисова.— Оренбург: Оренб. гос. ин-т. менеджмента, 2009. — 116 с.: ил.

12. 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. – 208с.

13. 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. – 208с.

14. 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.

15. Сорокина Т.Е. Визуальная среда Scratch как средство мотивации учащихся основной школы к изучению программирования [Текст] // Информатика и образование. – 2015. – №5(264), июнь.

16. Среда для обучения программированию Robomind [Электронный ресурс]. – <http://www.robomind.net/>

17. Среда для обучения программированию Scratch [Электронный ресурс]. – <http://scratch.mit.edu/>

18. Среда программирования Ligthbot [Электронный ресурс]. - <http://lightbot.com/>

19. Среда разработки LittleWizard [Электронный ресурс]. – <https://sourceforge.net/projects/littlewizard/>

20. Среда разработки будущих программистов Greenfoot [Электронный ресурс]. – <http://soft.mydiv.net/win/download-Greenfoot.html>

21. Угринович Н. Информатика и ИКТ [Текст]: Учебник для 9 класса / Н. Угринович. – М: БИНОМ. Лаборатория знаний, 2012. – 295с.

22. Федеральная служба государственной статистики [Электронный ресурс]. – http://www.gks.ru/wps/wcm/connect/rosstat_main/rosstat/ru/

23. Федеральный государственный образовательный стандарт второго поколения (ФГОС начального общего образования) 2009 г. [Электронный ресурс]. – <http://минобрнауки.рф/документы/543>
24. Федеральный государственный образовательный стандарт основного общего образования 2010 г. [Электронный ресурс]. – <http://минобрнауки.рф/документы/543>
25. Язык программирования Squeak [Электронный ресурс]. – <http://squeak.org/>
26. Язык программирования Baltic [Электронный ресурс]. – <http://www.sgpsys.com/en/>
27. Язык программирования Лого [Электронный ресурс]. – <http://myrobot.ru/logo/aboutlogo.php>

Тема урока: «Компьютер в нашей жизни. ТБ»

Тип урока: урок повторения знаний.

Цель урока: Прояснить знание детей о компьютере, о соблюдении правил ТБ с компьютером.

Задачи:

1. Образовательные:

- создать условия для освоения новых знаний; расширить горизонты детей, ознакомив их в игровой форме с компьютерными терминами и концепциями.

2. Развивающие:

- обогатить словарный запас детей;
- содействовать развитию внимания, памяти, мышления;

3. Воспитательные:

- развивать коммуникативную культуру, творческие способности учащихся;
- формировать способность к рациональному использованию времени;
- формировать способность планировать свои действия.

Формы работы: индивидуальные, групповые, коллективные.

Оборудование:

- проектор, компьютер
- альбом для рисования и цветных карандашей, магнитная доска, магниты.

Литература:

- методические рекомендации для учителя. - Москва: Баласс, 2011;
- видео "Компьютер в жизни человека"

Ход урока:

1. Организационный момент

Добрый день, ребята! Сегодня в классе мы поговорим с вами о предмете, который очень прочно укоренился в нашей жизни без него, временами мы чувствуем себя как без рук. Слушайте и попробуйте угадать: 30 лет назад, я и ваши родители были детьми, предмета этого у нас не было дома. 20 лет назад этот предмет появляется на крупных предприятиях, в офисах и банках. Прошло еще 10 лет, и этот предмет постепенно и прочно обосновался в школах, больницах, квартирах, домах. И сегодня он есть и даже не один у каждом из вас. Что это?

Ученики: компьютер.

2. Мотивация. Сообщение темы, цели и задач урока.

Сегодня мы поговорим о роли компьютера в вашей жизни. Когда и где вы впервые увидели компьютер? Что вы узнали об этом? Для каких целей они стали использоваться? Как ваши родители отреагировали на появление компьютера?

Вы уже много знаете о своем компьютере для своего молодого возраста. А, следующий этап нашей работы, мы будем называть «Анкета». Я дам вам опросники (рис.8), и вы будете честно и правдиво отвечать на них. Ваши анкеты потребуются для дальнейшей работы с вами на уроке.

Анкета «Компьютер в моей жизни»

- ✓ Твое имя _____
- ✓ Есть ли у тебя дома компьютер (ноутбук, телефон, планшет)? Подчеркни или напиши _____
- ✓ Он твой личный или всех членов семьи? _____
- ✓ Для чего тебе компьютер _____
- ✓ Как часто ты пользуешься ПК? _____
- ✓ Чем ты занимаешься на ПК? _____
- ✓ В каких программах работаешь? (Word, Point, Power Point, Excel)? Подчеркни _____
- ✓ Ты спрашиваешь разрешение у родителей перед использованием ПК или сам решаешь? _____
- ✓ Кто научил тебя пользоваться ПК? _____
- ✓ Имеешь ли ты свою страничку в соц.сетях? _____
Где? _____
- ✓ Хотел бы выбрать профессию, связанную с ПК? Какую? _____
- ✓ Нужен ли компьютер в школе? Для чего? _____

Рис.8 Анкета

3. Анкетирование

Чтобы понять, насколько вы знаете компьютер, знаете, как работать на нем, собственные компьютерные термины, для этого всего, мне нужно обработать результаты анкеты. И, вы, в это время можете нарисовать тематический рисунок «Компьютер в моей жизни».

4. Рисование на тему

Не забудьте подписать свои рисунки на обратной стороне, поместите готовые работы на доске для конкурса на лучший рисунок (рис.9).



Рис.9 Работы учащихся

5. Деление на группы

Итак, подведем итоги нашей анкеты, основываясь на результатах опросников, наш класс можно разделить на 2 группы. Я хочу назвать группу детей (Полина, Катя, Егор, Соня, другая Полина), которые не только могут играть на компьютере, но и создавать презентации, видео, которые часто представлены нам в классе. Эта группа продвинутых пользователей, а все остальные начинающие пользователи.

Мы переходим к задаче: слушаем задачу, которую мы с компьютером предлагаем выполнить этим группам.

Группе продвинутых пользователей необходимо создать коллективный проект до конца мероприятия - видеофильм «Компьютер в моей жизни», а начинающим пользователям, рисовать свои рисунки в программе Paint. По-

нения, непонятные для вас слова, которые вызовут трудности у вас, вы можете найти в Интернете в словаре компьютерных терминов.

6. Получение новых знаний

Для чего, по вашему мнению, вам нужен Интернет (рис.10)? Обратите внимание на плакат, в котором рассказывается о функциях Интернета.

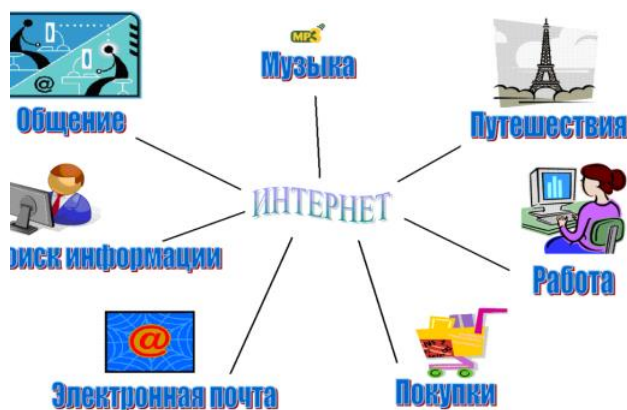


Рис.10 Зачем нам Интернет?

Прежде чем приступать к работе с компьютером, я предлагаю вам видео «Как правильно сидеть за компьютером», чтобы вы всегда помнили о своем здоровье.

7. Защита творческого проекта, созданного группой детей во время мероприятия.

- Давайте послушаем ответы ребят, зачем нам нужен Интернет?
- Давайте посмотрим рисунки, которые вы получили в Painte, и поставим оценки друг другу за урок!

8. Итог занятия

Ребята, вам понравилось наше мероприятие? Ответьте на мой вопрос, продолжая предложения

- Я узнал сегодня ...
- Я сегодня повторил ...
- Сегодня я усвоил...

С каким настроением вы идете домой? Если вы веселы и радостны, то хлопайте в ладоши. Те, кто немного грустен, помашите мне рукой.

До новых встреч с компьютером!

Конспект урока №2

Тема урока: «Алгоритмы в жизни человека. Линейный алгоритм»

Тип урока: Урок формирования новых знаний.

Задачи:

1. Образовательные:

- создавать условия для освоения новых знаний; содействовать развитию способности школьников составлять и реализовывать план (алгоритм) действий; определить правильность порядка шагов,

2. Развивающие:

- обогатить словарный запас детей; содействовать развитию внимания, памяти, мышления;

3. Образовательные:

- культивировать бережное отношение ко времени; формировать способность к рациональному использованию времени; формировать способность планировать свои действия.

Личностные цели: интерес к изучению информатики.

Метапредметные цели:

- регулятивные: развитие способности оценивать правильность реализации действия, контролировать себя, находить и исправлять свои собственные ошибки;
- познавательный: развитие навыков для извлечения и обработки информации, полученные из разных источников (таблицы, диаграммы);
- коммуникативные: развивать способность аргументировать свои ответы, выслушивать мнения других.

Предметные цели:

- научиться различать этапы (шаги) действия;
- научиться определять правильный порядок шагов;
- ознакомиться с понятиями «алгоритм действия», «командный алгоритм»;
- научиться, как создавать и выполнять алгоритмы;

- иметь первоначальное представление о вложенности алгоритмов;
- научиться, находить и исправлять ошибки в алгоритмах.

Формы работы: индивидуальные, групповые, коллективные.

Оборудование:

- проектор, компьютер
- модель Lego robot

Литература:

- методические рекомендации для учителя. - Москва: Баласс, 2011;
- видео "Компьютер в жизни человека"

Ход урока

1. Этап мотивации

Организация для урока.

2. Актуализация знаний.

1. Информатика. Человек и компьютер

- С изучением какого устройства, полезного инструмента связано с изучением информатики?

- Как можно использовать компьютер? (С помощью компьютера дети могут играть, рисовать.) Компьютер можно использовать:

- при создании мультфильма, рекламы;
- для вождения, космических аппаратов;
- для осмотра в кабинете врача;

- Считаете ли вы, что компьютер сам может рисовать или играть? Может ли он самостоятельно решить задачу или проблему без участия человека? (Компьютер не может действовать самостоятельно).

Чтобы управлять своими действиями, человек делает подробные задачи - компьютерные программы.

- Какие машины и устройства могут быть встроены в компьютер?

(Компьютер может находиться внутри стиральной машины, камеры, автомобиля, самолета, космического аппарата ...)

Демонстрация видео «Компьютер в нашей жизни».

3. Выявление места и причины затруднения

2. Демонстрация модели Лего-робота (Mindstorms)

- А, еще компьютер обязательно есть внутри каждого робота. Что такое робот? У него есть мозг? Кто контролирует действия робота?

(Робот - это машина, у нее нет мозга.) Человек контролирует действиями робота.

- Для робота, как для компьютера, очень важно правильно составить задачу. Один человек сделал себе робота- помощника и попросил его пойти на кухню, налить чай и принести. Робот принес кружку воды, а к вопросу владельца: «Что ты сделал?», ответил: «Я точно выполнил все инструкции». Давайте попробуем тренироваться в управлении роботом.

Физминутка

4. Этап построения проекта выхода из затруднения

После того, как игра закончена, подведите итоги:

1. Чтобы описать действие, необходимо выбрать отдельные этапы, из которых состоит это действие, и привести их в порядок;

2. Каждый шаг - это также действие, которое можно подробно описать
- выделить шаги и привести их в порядок.

3. Алгоритм. Команда алгоритма

- Если робот часто выполняет одну и ту же задачу, удобнее сохранить эту задачу в специальной электронной памяти робота. План действий (последовательность шагов) - это алгоритм действия. Скорее всего, вам уже пришлось действовать в соответствии с планом. Просто вы не знали, что план действий является алгоритмом.

Вспомните, какие действия вы проводили в соответствии с планом, в соответствии с описанием лица

Примеры ответов:

- найти путь в школу (магазин, автобусная остановка, бассейн);
- пришить пуговицу, убрать в комнате;
- добавить два двузначных числа ...)

Каждый шаг в алгоритме обычно называется командой. В начале алгоритма принято писать команду «Пуск», а в конце - команду «Конец».

- Конечно, ваши родители спрашивают вас: «Пойдите в магазин, купите еду».

- Или мама говорит: «Объясни младшей сестре или брату, как открыть дверь ключом».

- Мы предлагали друг другу во время зимних прогулок: «Слепой снеговик»

5. Этап реализации построенного проекта

4. Последовательность команд в алгоритме

Выполнить задание 1 (алгоритм «Посадить дерево»).

Пронумеровать рисунки по порядку, начиная с номера 3.

Попросите детей назвать команды для рисунков без подписи.

Напишите команды алгоритма в тетради.

Чтобы привлечь внимание детей: при составлении задачи для робота мы не полагаемся на его сообразительность и включаем такие команды, чтобы любое действие было завершено. Например, робот начинается с шага «принесите лопату, лейку и саженец» и заканчивается действием «отнесите лопату и лейку на место».

- С чего должен начинать робот, и как завершить действие «вырыть яму»? Какими будут первые и последние шаги? («Возьмите лопату» и «положите лопату».)

Физминутка

6. Первичное закрепление проекта

5. Ошибки в алгоритме.

Выполнить задание 2 (алгоритм «Ешь банан»).

Обсудите ошибки робота, показанные на трех верхних рисунках. (Он ест банан с кожурой, очищает банан, но ест кожуру, а не мякоть, не выбрасывает кожуру, где она должна была, поскользнулся и упал.)

Попросите детей самостоятельно закончить алгоритм «Съешь банан».

Проверьте ответы, отметьте, учащихся, которые включили в свой алгоритм, не обычные, но разумные шаги.

7. Этап включения в систему знаний

6. Выполнение алгоритма

«Как попасть в школу»

«Как одеться по погоде»

8. Этап рефлексии

1. Итоги урока

- С помощью чего вы можете описать действие? (Сегодня мы узнали, что действие можно описать с помощью алгоритма)
- Из чего состоит алгоритм? (Алгоритм состоит из команд)
- Вам нравилось делать и выполнять алгоритмы? (Да/нет)
- У кого были трудности?

Домашнее задание

Задача 3 - составить алгоритм «Возьмите банан».

Задача 4 - выполнить алгоритм «Раскрась рисунки».

Тема урока № 5. Знакомство со средой программирования 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:

- Доска, компьютеры для каждого ученика, программа Scratch.

Структура и ход урока

1. Организационный момент

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

2. Изучение нового материала

Но для начала рассмотрим интерфейс программы. Так выглядит экран программы, которая состоит из четырех частей (рис.11).

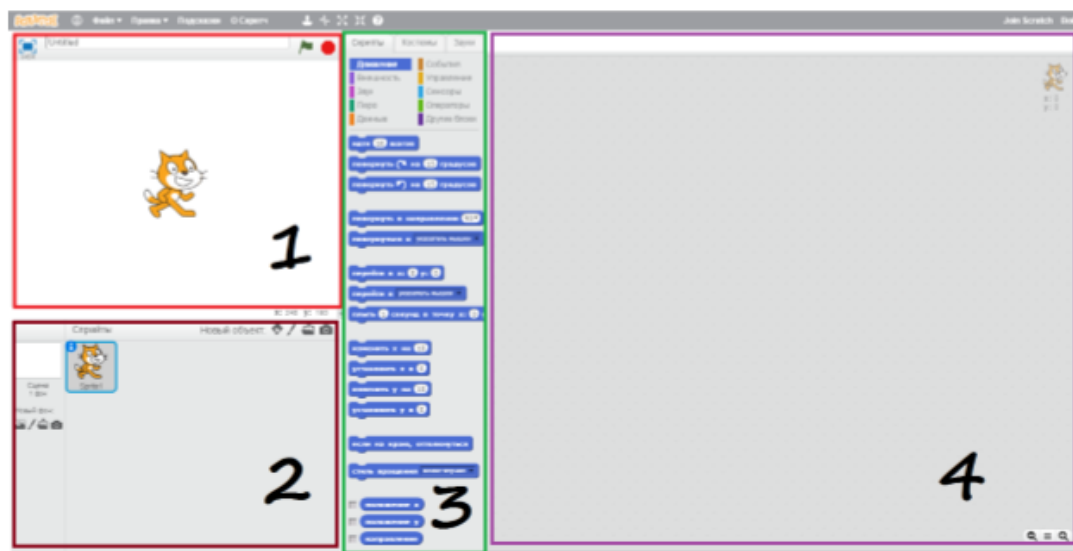


Рис. 11 Интерфейс программы Scratch

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

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

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

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 рассмотреть внимательней область под номером 3. В Scratch большое количество блоков разного цвета. В течение курса, мы познакомимся со всеми постепенно, но для начала будем использовать синие, фиолетовые, розовые и желтые.

Итак, желтый раздел (рис.12), который мы сегодня рассмотрим, называется события. Данные блоки 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.



Рис.12 Раздел События

Следующий раздел (рис.13), синий. Это раздел движение. Скажите мне, пожалуйста, что наш спрайт может делать, если мы воспользуемся этими блоками?

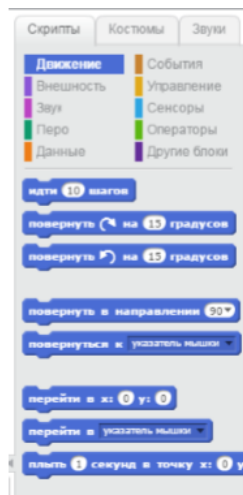


Рис.13 Раздел Движение

Учащиеся: 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, а 5 и т.д.

Розовый блок (рис.14), этот раздел отвечает за звуки. Прочитайте по очереди.

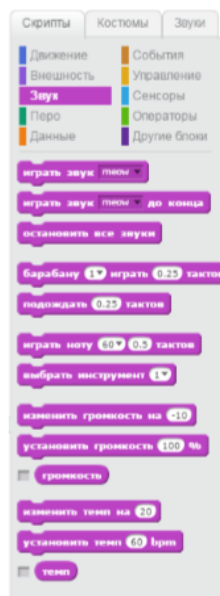


Рис.14 Раздел Звуки

Учащиеся: 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.

И, последний раздел - сиреневый (рис.15), внешность. Прочитайте по очереди.



Рис.15 Раздел Внешность

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

Учитель: переходим к практической части. Но перед тем как сесть за компьютеры, предлагаю немного размяться и выполнить несколько физических упражнений.

3.Физминутка

Молодцы, теперь садимся за компьютеры. Запускаем программу Scratch с рабочего стола.

Задание 1. Учим кота ходить. Выбираем вверху желтые блоки «события» (рис.16), находим блок «когда клавиша пробел нажата», нажимаем левой кнопкой мыши и, не отпуская, перетаскиваем блок вправо в область скриптов.

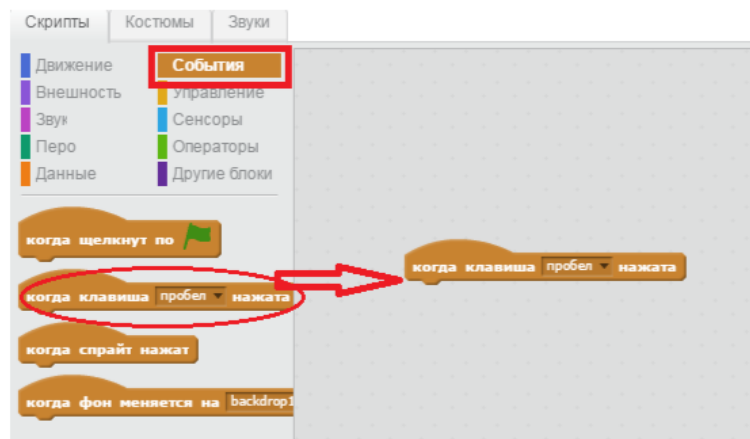


Рис.16 Блоки События

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 (рис.17).

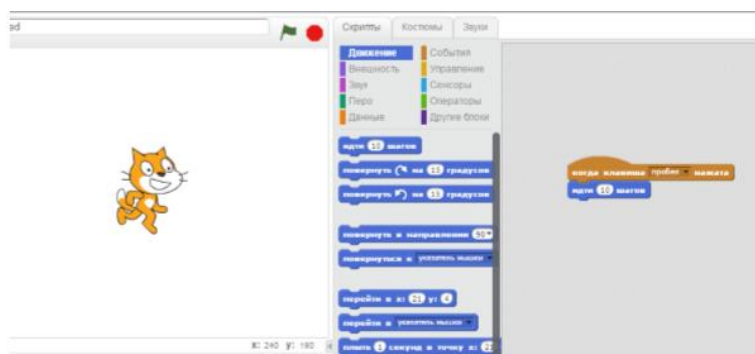


Рис.17 Блоки Движение

Если у вас возникают 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?

Учащиеся: Кот двигается вправо.

Учитель: Верно. Если много раз нажимать на пробел, то он дойдет до края экрана и почти скроется. Потянув за хвост, его можно вытащить.

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?

Учащиеся: 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 (рис.18).

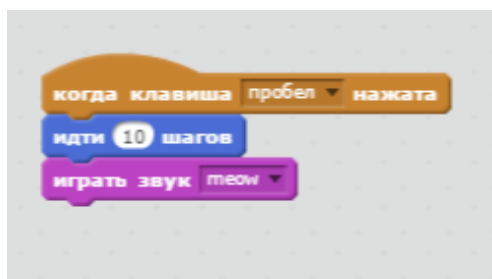


Рис.18 «Играть звук meow»

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» (рис. 19).

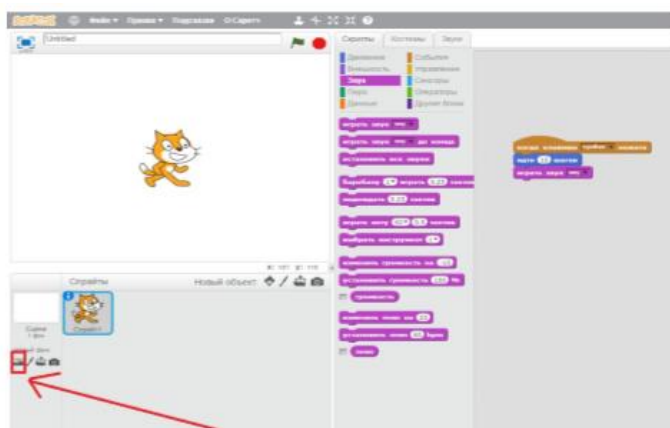


Рис. 19 Фон из библиотеки

Нажимаем на кнопку, и появляется окно – Библиотека фонов. Там выбирайте любой понравившийся фон (я выбрала пустыню) и нажимайте кнопку «ОК» в правом нижнем углу (рис. 20).

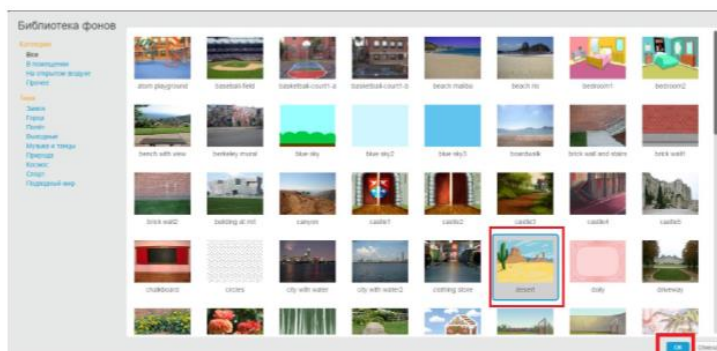


Рис. 20 Изменение фона

Фон изменился, согласитесь со мной, что так работа выглядит куда более интересной (рис. 21).



Рис. 21 Выбор фона

У всех получается? Если есть затруднения, то приглашайте меня, я помогу. Если вдруг вы нечаянно добавили не тот блок или просто захотели удалить блок, то необходимо открепить его и перетащить обратно.

Самостоятельно!

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

Учащиеся: 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 (рис. 22).

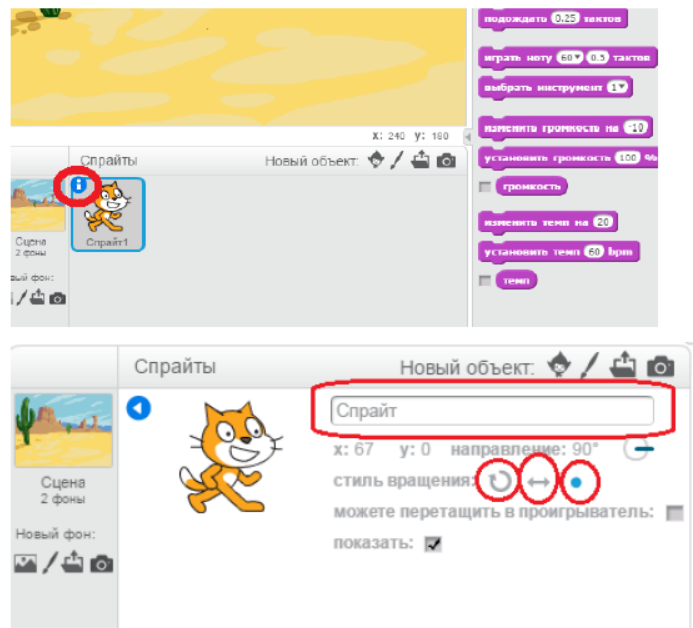


Рис. 22 Изменении имени спрайта

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.

Учащиеся: Кот по достижению края пойдет задом.

Учитель: Хорошо, а теперь, я 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?

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

Учитель: Всем спасибо, урок закончен.