Всероссийский конкурс и конференция

«Юные техники и изобретатели» в Государственной Думе,

г. Москва, 2016 год



Исследовательская работа по номинации:

информационные технологии

по теме:

**«Разработка среды программирования ”Робот” и её использование при изучении алгоритмического языка на уроках информатики»**

**выполнил:**

ученик Школы программирования

IT-ПАРК

Синица Егор

**руководитель:**

Киреев А.М.,

руководитель Школы программирования

IT-ПАРК

2016 г.

**Оглавление**

[Введение .....3](#_Toc358633777)

[Глава 1. Разработка среды программирования для школьного алгоритмического языка на примере среды «Робот».. 5](#_Toc358633778)

[1.1. Постановка задачи разработки алгоритмической среды «Робот» 5](#_Toc358633779)

[1.2. Описание этапов разработки приложения «Робот» в среде C# 6](#_Toc358633780)

[Глава 2. Изучение алгоритмических конструкций на уроках информатики с использованием приложения «Робот» 8](#_Toc358633781)

## [2.2. Методические рекомендации по применению приложения «Робот» на уроках информатики 10](#_Toc358633782)

Результаты исследования [12](#_Toc358633784)

[Заключение 14](#_Toc358633784)

[Список используемых источников 16](#_Toc358633785)

# Введение

Настоящий этап развития общества отличается интенсивным внедрением во все сферы человеческой деятельности новых информационных и коммуникационных технологий, обеспечивающих более полную реализацию потенциальных способностей личности. Система общего и среднего образования призвана подготовить компетентного выпускника школы, способного жить, трудиться и непрестанно образовываться в современных условиях.

Процессы обучения и воспитания не сами по себе развивают человека, а лишь тогда, когда они способствуют формированию тех или иных типов деятельности.

Деятельность выступает как внешнее условие развития у ребенка познавательных процессов. Чтобы ребенок развивался, необходимо организовать его деятельность. Значит, образовательная задача состоит в организации условий, провоцирующих детское действие. Такую стратегию обучения легко реализовать на уроках информатики при обучении основам алгоритмизации с использованием алгоритмических программных сред.

Младший школьный возраст наиболее благоприятен для развития таких важных для всей последующей учебы и жизни школьника психических процессов, как рефлексия, внутренний план действий, которые, в свою очередь, являются основой для формирования алгоритмического стиля мышления. Если это время будет упущено, то в более старшем возрасте эти качества развить значительно труднее, а иногда и просто невозможно.

Многие авторы школьных учебников, разработчики учебных программ предлагают использовать для обучения алгоритмизации и программированию различные виды исполнителей. Причем каждый исполнитель предназначается для выполнения определенного круга задач.

Уча детей работе с исполнителем в программной среде, можно добиться большей заинтересованности на уроке, чем при чисто теоретическом объяснении темы. К сожалению, авторы новых учебников по информатике уделяют недостаточно внимания или вообще обходят стороной данных исполнителей. В то же время разработчики ОГЭ и ЕГЭ включают задания с использованием данных исполнителей в задания экзаменационных работ.

Актуальность моего исследования обусловлена необходимостью совершенствования методики преподавания алгоритмизации в пропедевтическом курсе информатики с использованием алгоритмических исполнителей.

Исходя из всего вышесказанного, я решил создать приложение, с помощью которого в игровой форме можно было бы развивать алгоритмическое мышление младших школьников и разработать методические рекомендации по его использованию на уроках информатики.

В моей работе рассматривается создание приложения «Робот», которое является компьютерной алгоритмической средой.

В качестве среды программирования я использую Visual Studio 2015 Express. Дополнительно использую Corel Draw – для создания и обработки графических изображений.

**Объект исследования**: методические подходы к обучению основам алгоритмизации в начальной школе.

**Предмет исследования**: методика обучения основам алгоритмизации в начальной школе с использованием алгоритмической среды «Робот».

**Цель исследования**: разработать приложение «Робот» в среде C# и методические рекомендации по его использованию на уроках информатики.

Для достижения поставленной цели необходимо решить ряд задач:

1. Изучить психолого-педагогическую и методическую литературу по проблеме развития алгоритмического мышления младших школьников.
2. Ознакомиться с библиографическими источниками по программированию в C# и проанализировать их.
3. Разработать удобный интерфейс приложения.
4. Описать программный код обработчиков событий.
5. Разработать методические рекомендации по использованию алгоритмической среды «Робот» на уроках информатики.
6. Рассказать пользователю об интересных фактах, связанных с IT.

Глава 1. Разработка среды программирования для школьного алгоритмического языка на примере среды «Робот»

## 1.1. Постановка задачи разработки алгоритмической среды «Робот»

Для создания приложения «Робот» определим круг задач. Пользователь, работающий с данным приложением, имеет возможность выполнять следующие операции:

* + Создавать новый уровень.
  + Загружать имеющийся в памяти уровень и редактировать его.
  + Выполнять созданные алгоритмы.
  + Вносить изменения в алгоритмы.

После запуска приложения на экране появляется форма приветствия.

Общий вид окна программы.

Графическая среда исполнителя Робот-исполнитель.

Роль человека - составление алгоритма, ввод программы и инициализация ее исполнения Роботом. Составленный алгоритм отображается рядом, с возможность редактирования.

Работа приложения происходит в 3 этапа.

* программирование: составление алгоритма.
* исполнение: работа исполнителя по заданному алгоритму.
* сброс: восстановление изначальной позиции исполнителя.

И, наконец, исполнение может проходить в 2 режимах:

* в автоматическом режиме (на экране автоматически выполняется алгоритм, написанный пользователем);
* в пошаговом режиме или режиме отладки (Робот выполняет каждый шаг алгоритма отдельно, по нажатию клавиши);

Режим чтения из файла.

Приложение выполняет чтение файлов уровней \*.datefile, также приложение считывает интересные факты из файла facts.txt.

Порядок работы в среде Робот:

1.Задать последовательность действий с помощью специальных клавиш.

2.Проверить алгоритм, если имеются ошибки, исправить, или же составить его заново.

3.Выполнить алгоритм, с помощью клавиш “Начать тест” или “Отладка”

Таким образом, среда «Робот-исполнитель» представляет собой игровой вариант ряда специализированных методик развития алгоритмического мышления у детей младшего школьного возраста. Эта среда, воссоединившая в себе элементы развлечения и развития, позволит в короткие сроки сформировать у школьников простейшие навыки программирования с использование линейной структуры.

## 1.2. Описание этапов разработки приложения «Робот» в среде C#

В качестве среды для разработки моего приложения я выбрал C#, поскольку данная среда разработки предоставляет все необходимые инструменты для решения поставленной задачи. C#, как и всякая современная среда разработки приложений, основана на объектно-ориентированном программировании (ООП). Эта технология программирования является той основой, которая и позволяет реализовать все функциональные возможности C#. При создании приложений на основе готовых компонентов с использованием свойств, методов и заранее определенных обработчиков событий можно обходиться программным кодом небольшого объема. Для разработчика это означает, что при разработке пользовательского интерфейса своих приложений, он может получить значительную экономию времени.

Прежде чем приступить к непосредственному программированию приложения – алгоритмической среды «Робот», я повторил, как записываются алгоритмы на алгоритмическом языке, так как мне предстояло реализовать аналогичные команды и алгоритмы в собственной среде.

В приложении реализована одна базовая структура:

Структура  "следование". Образуется последовательностью действий, следующих одно за другим.

Процесс создания приложения:

1. Мною изучена программа Графический Исполнитель «Стрелочка».
2. Мной написан свой собственный ГРИС.
3. Подготовлены графические материалы (картинки).
4. Написан код отрисовки поля, добавлена возможность добавления действий.
5. Вставлены картинки, добавлена возможность восстановления прогресса.
6. Создана отдельная форма с интерактивными подсказками.
7. Разработана программа создания уровней («LVLcreator»).
8. Улучшен графический интерфейс, доработан создатель уровней.
9. Полностью протестирована в обычных условиях, багов нет.
10. Проведены тесты для поиска неисправностей, которые затем исправлены.
11. Добавлены интересные факты, доработан код, исправлен интерфейс.
12. Протестировано на ПК Intel Pentium N3540, GeForce 720M, 2048 Mb.

# 

# Глава 2. Изучение алгоритмических конструкций на уроках информатики с использованием приложения «Робот»

## 2.1. Методические рекомендации по изучению основных алгоритмических конструкций в пропедевтическом курсе информатики

Главной целью курса раннего обучения информатике с первых лет существования школьной информатики было обозначено развитие алгоритмического (впоследствии установился термин «операционного») стиля мышления. Операционный стиль мышления представляется совокупностью навыков умственных действий, которые необходимы каждому современному образованному человеку, а именно: умение планировать структуру действий, умение строить информационные модели, умение организовать поиск информации, умение дисциплинировать и структурировать общение, умение инструментировать свою деятельность .

Одно из центральных умений операционного мышления – планирование структуры действий, в простейших случаях – планирование действий, безусловно, формируется под влиянием алгоритмических разделов курса. На бытовом уровне и дети, тем более взрослые понимают и вкладывают определенный смысл в слова «цикл», «циклические действия», в высказывания типа «если …, то …, иначе …».

Однако это не означает, что любой человек представляет существо базовых алгоритмических структур и модель их работы. Сформировать такое обобщение – значит, вооружить человека определенным инструментарием. Для этого необходимо движение от конкретных знаний и умений к абстрактному обобщенному знанию, к  дальнейшему применению его на практике, т.е. переносу знаний в иные сферы деятельности.

Поэтому естественно предполагать пропедевтику понятия алгоритма, нисходящего проектирования алгоритмов, базовых алгоритмических структур с самых первых лет обучения информатике в начальной школе.

В результате обучения основам алгоритмизации учащийся должен:

* понимать сущность понятия алгоритма, знать его основные свойства, иллюстрировать их на конкретных примерах алгоритмов;
* понимать возможности автоматизации деятельности человека при использовании алгоритмов;
* записать на учебном алгоритмическом языке алгоритм простой задачи;

Обучение основам алгоритмизации с использованием алгоритмических сред обладает рядом особенностей. Прежде всего, это относится к выбору самой среды, которая должна быть наделена всеми необходимыми функциями для изучения основных алгоритмических конструкций.

Исполнитель в данном приложении - робот, снабженный набором кнопок. Каждая кнопка соответствует одному действию, которое робот способен совершить. Нажатие кнопки вызывает добавление шага в алгоритм, которой затем выполнится роботом.

*Исполнитель* *алгоритма***-**это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.

Робот действует в определенной среде. Чтобы описать исполнителя, нужно задать среду, в которой он действует, и действия, которые он совершает при нажатии каждой из кнопок.

*Среда* *(или обстановка)* - это "место обитания" исполнителя. Например, для исполнителя Черепашка в Лого Мирах, среда - это бесконечное поле. Края поля тоже часть среды, а их расположение и положение самой Черепашки задают конкретное состояние среды.

*Система команд.* Каждый исполнитель может выполнять команды только из некоторого строго заданного списка - системы команд исполнителя. Для каждой команды должны быть заданы условия применимости и описаны результаты выполнения команды. Например, команда Черепашки "ВПЕРЕД" может быть выполнена, если рядом с Черепашкой нет края поля. Ее результат - смещение Черепашка на один шаг вперед.

После вызова команды исполнитель совершает *соответствующее элементарное действие*.

Все это позволяет школьнику наглядно представить себе процесс выполнения алгоритма, а не только его составления, что очень важно для понимания сущности алгоритмизации в целом.

Таким образом, общая схема изучения основ алгоритмизации с использованием алгоритмических сред следующая: от частного к общему, от примера к понятию.

## 

## 2.2. Методические рекомендации по применению приложения «Робот» на уроках информатики

Ученик должен получить представление о конкретном типе задач, решения которых можно записать в виде алгоритма, о задачах.

Должен узнать о том, как выполнять задачу в среде «Робот» и в других средах тоже. Для этого научиться оценивать задачу, составлять алгоритм, а затем его выполнять.

Приложения, которые используются на уроках в школах сейчас, позволяют ученику изучить алгоритмы, составлять эти алгоритмы, выполнять и редактировать алгоритмы. Однако процесс обучения в таком приложении – скучная и неинтересная задача.

Разработанная мной алгоритмическая среда «Робот» - это учебная среда, позволяющая изучить азы алгоритмизации и программирования на учебном исполнителе. При этом ученику совершенно не нужны знания программирования.

Обучение проходит в игровой форме, а значит конечному пользователю будет интересно, и к тому же познавательно изучать алгоритмы именно в моём приложении.

Опишу понятия, которыми можно манипулировать в среде «Робот».

Объекты: проекты, исполнитель «Робот», рабочее поле робота, текстовые окна, кнопки, бегунки, формы, рисунки.

Общие свойства объектов: цвет, размер, положение на Рабочем поле.

Действия объектов: создавать, удалять, выполнять, проверять.

.

Интерфейс: обычный для любого приложения Windows - окно, основное меню, панели инструментов. Процесс обучения выполняется в режиме интерпретации, поэтому на экране - Рабочее поле и кнопки задания и теста алгоритма.

Язык: школьный алгоритмический.

Система команд исполнителя: Шаг Вправо, Шаг Влево, Шаг Вниз, Шаг Вверх, Прыжок Вправо, Прыжок Влево, Прыжок Вниз, Прыжок Вверх.

Алгоритмические конструкции: простая последовательность действий.

Помощь: кнопка на главной форме с надписью “Помощь и Подсказки”.

В дополнение к данному приложению идет программа – создатель уровней, который может использоваться учителем, а также может быть представлен ученику. В последнем случае учащемуся необходимо будет также продумать эффективность и конечность (то есть возможность прохождения) созданного уровня.

# Результаты исследования

Проблема формирования алгоритмической культуры учащихся особенно актуальна в современном образовательном процессе. Совокупность знаний, умений и навыков работы с алгоритмами формируется у школьников при изучении разных дисциплин. Математике и информатике принадлежит ведущая роль в формировании алгоритмического мышления, воспитании умений действовать по заданному алгоритму и конструировать новые алгоритмы.

Целью обучения информатике является формирование алгоритмической культуры и компьютерной грамотности. В ходе изучения этой дисциплины систематически и последовательно формируются навыки умственного труда: планирование своей работы, поиск рациональных путей ее выполнения, критическая оценка результатов.

Данная программа была протестирована мною на уроках информатики в МБОУ Брянский Городской Лицей №2 им. М.В.Ломоносова:

* программа была представлена школьникам 3-го класса, которые еще не знакомились с понятиями "Алгоритм", "Программирование" и т.д. Для них это было что-то новое, интересное, не обычное, но для восприятия довольно сложное. Алгоритмы составлялись легко и быстро, однако часто допускались ошибки, в последовательности действий.
* Затем была представлена учащимся 5-го класса, которым алгоритмическая среда "Робот" показалась очень интересной и главное понятной в использовании, особенно дети отметили интерактивные подсказки по ходу игры.
* была предложена учащимся 9-го класса. Большая часть ребят заинтересовалась не интерфейсом программы, а самим кодом создания и языком программирования, поскольку раздел их изучения как раз был язык программирования Pascal ABC. Но, так как среда программирования C# не изучается в школе, преподаватель выделил лишь 1 урок для показа, знакомства и теста моего приложения.

Самое главное, у обучающихся не возникло вопросов по использованию программы. Это говорит о том что интерфейс предельно прост и понятен любому ученику. Это еще раз отмечает преимущество и необходимость использования в школе моей программы "Робот".

# Заключение

Изучив и проанализировав содержание библиографических источников, и выбрав из них нужную информацию, мы убедились, что для формирования алгоритмического мышления школьников необходимо выработать у них различные умения работы с алгоритмами. Наиболее эффективно эта работа может быть построена с использованием компьютерных исполнителей. Они позволяют решать задачи, наблюдать допущенные ошибки, исправлять их.

**В основу появления разработанной нами алгоритмической среды «Робот» легли следующие приложения:**

* Графический Исполнитель (ГРИС), изучаемый в 9 классе.
* Написанный мною Графический исполнитель.
* Основы графического интерфейса от игр Морской бой и Змейка.

Среда Робота позволяет так построить процесс обучения, что овладение языком программирования становится таким же естественным процессом, как и изучение родного языка. Слова «компьютерное обучение» обычно означает, что компьютер учит («программирует») ребенка. Занятия в среде Робота переворачивают традиционную ситуацию компьютерного обучения. В ней ребенок сам учит и программирует компьютер и, делая это, овладевает основами предмета.

Важную роль в новых направлениях обучения играет отношение к ошибкам и их устранение. Отладка программы (поиск и исправление в ней ошибок) обычно является трудоемким процессом. Язык Робота значительно упрощает его. Решение разбивается на множество мелких шагов, каждый из которых может быть проверен отдельно ( с помощью клавиши отладка). Если допущена ошибка, то исполнитель просто не выполнит этап алгоритма и остановится.Так как ученики могут самостоятельно найти ошибку, исправить и снова попробовать, то они перестают бояться ошибок. Школьники чувствуют себя исследователями, не боящимися творить. Они обретают уверенность в своих силах и могут выразить свое знание предмета в форме, отличной от вербальной.

Таки образом, задачи, поставленные в исследовании, выполнены, цель исследовательской работы достигнута.

В заключение, хочется отметить, что овладение учащимися среды Робот реализует новые подходы к обучению, направленные не на заучивание правил, а на формирование процесса мышления.

Следует учитывать и методические преимущества:

• возможность сглаживания и уменьшения противоречий, присущих классно-урочной системе обучения,

• повышение эффективности процесса обучения за счет его индивидуализации.

Поэтому использование компьютера и программных исполнителей является не целью обучения, а служит средством эффективного обучения.

# Список используемых источников

1. Yandex.ru
2. Cyberforum.ru
3. Code.org
4. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд.
5. Программирование для Microsoft Windows 8. 6-е изд.
6. Язык программирования C#. Классика Computers Science. 4-е изд.
7. Приложения для Windows 8 на C# и XAML
8. Изучаем C#. 3-е изд.
9. Гуриков С. Р. Введение в программирование на языке Visual C#;
10. Форум, Инфра-М, 2013. - 448 c.
11. Мартин Р. С., Мартин М. Принципы, паттерны и методики гибкой
12. разработки на языке C#; Символ-Плюс, 2011. - 768 c.
13. Пугачев С., Шериев А., Кичинский К. Разработка приложений для Windows 8 на языке C#; БХВ-Петербург, 2013. - 416 c.