Министерство образования Российской Федерации

Уральский государственный педагогический университет

Кафедра информатики и ВТ

**Организация изучения основных алгоритмических конструкций в среде Лого Миры**

ВЫПУСКНАЯ РАБОТА

Исполнитель: студентка 5 курса

Заочного отделения

Информатики и ВТ

Факультета УрГПУ

Научный руководитель:

Кандидат педагогических наук

Доцент кафедры информатики

и вычислительной техники

**Грохульская Наталья Леонидовна**

Екатеринбург 2002

СОДЕРЖАНИЕ:

Введение 4

Глава 1. Язык Лого и его применение в преподавании информатики 7

История появления языка Лого 7

Распространенность Лого-сред в Екатеринбурге 9

Анализ сред Лого 10

Основные достоинства и недостатки среды Лого 13

Применение Лого в учебном процессе начальной школы 16

Применение Лого для внешкольной работы по информатике 18

Обзор публикаций о применении Лого в учебном процессе школы 20

Роль алгоритмизации в базовом курсе 23

Технология решения вычислительных задач 29

ГЛАВА 2. Решение вычислительных задач в среде Лого Миры 32

Описание возможностей среды Лого Миры 2.0 32

Понятия, которыми можно манипулировать в среде Лого Миры 33

Запуск Лого Миров 34

Графический экран системы Лого 34

Текстовый экран системы Лого 34

Режимы работы 34

Команды системы Лого 35

Базовые команды движения и рисования 36

Команды цикла (повторения) 37

Построение процедур (подпрограмм) 38

Команды изменения переменных 39

Команды и процедуры с параметрами 40

Команда выбора (ветвление) в Лого 40

Рекурсия 41

Функция 41

II. Команды управления системой 41

Запись основных алгоритмических конструкций в среде Лого Миры и их использование для решения вычислительных задач 43

Понятие алгоритма 43

Исполнитель алгоритма 43

Свойства алгоритма 44

Способы записи алгоритмов 45

Словесный способ записи алгоритмов 45

Графический способ записи алгоритмов 46

Базовые алгоритмические структуры 47

Решение задачи с использованием основных базовых структур и операторов языка Лого на примере экологической задачи 61

Методические рекомендации 64

Изучение базовых алгоритмических конструкций в языке Лого в учебном процессе школы 64

Примерный план разделов курса «Алгоритмизация и решение вычислительных задач» 66

Заключение. 68

Введение

Лого – это среда программирования и средство для моделирования различных задач и исследований. Объектами исследований преподавателей и учеников, в этом направлении могут быть их собственные исследования о животных и растениях, о мире психологии, мире физиологии человека, мире биологии клетки, а также химии, физики, математике, изобразительного искусства и других сфер деятельности.

В любой среде программирования реализуются основные алгоритмические конструкции, развивающие алгоритмический стиль мышления, важность которого отмечена Н.М. Амосовым, Н.Н. Моисеевым, А.Н. Лонда и другими учеными.[ ] Ими подчеркивалась необходимость разработки алгоритмов для развития мышления школьников. Они показывали, что с помощью алгоритмов можно не только организовывать мыслительную деятельность, но и описывать процессы.

Алгоритмы возникают не только в ходе описания какого-либо процесса (физического, химического, биологического, математического), но и в управлении, воспитании, во всей социальной сфере жизни человека. Именно это и доказывает необходимость их введения в обучение. Таким образом, алгоритм – это не программа-шаблон, а механизм, согласно которого функционирует, развивается любая самоорганизующая система. Некоторые алгоритмы человек осваивает самостоятельно, другие требуют обучения.

Фундаментальное понятие информатики - «алгоритмизация», имеет большое значение не только в теории информатики, но и в теории самореализации в развитии ученика.

Объем минимального содержания базового курса информатики включает в себя блок «Алгоритмы и исполнители». Алгоритмизация - одно из мощных средств развития мышления учащихся.

Изучать основные базовые конструкции можно по-разному: с помощью исполнителей [ ], специально разработанных учебных алгоритмических языков [ ], реальных языков программирования [ ].

Одно из перекрестных средств знакомства учащихся с основными алгоритмическими конструкциями является язык Лого.

В последнее время язык программирования Лого завоевывает все большую популярность и не только в начальной или средней школе, но и в колледжах, университетах, да и просто в среде любителей интеллектуального досуга.

Лого – один из самых доступных языков программирования для персональных компьютеров. Это диалект ЛИСП, языка, применяемого в сложнейших исследовательских проектах в области вычислительной техники и искусственного интеллекта. Не случайно Лого оказался более гибким языком, чем Паскаль или С++.

Язык Лого – это язык программирования и вместе с тем особая обучающая сфера. Язык этот по синтаксису предельно прост и близок к естественному. В то же время он обладает мощными современными средствами, формирующими культуру мышления и позволяющими создавать программы очень лаконичные, прозрачные по структуре и эффективности.

Поэтому целесообразно использовать этот язык при изучении базового курса информатики. В связи с этим становится очевидным актуальность предлагаемой работы.

Объект исследования – процесс обучения базовому курсу информатики.

Предмет исследования – методика преподавания алгоритмизации на базе языка Лого.

Цель работы – определение роли и месте вычислительных задач, решаемых в среде Лого Миры при изучении алгоритмизации.

Основные задачи исследования:

1. Освоить среду Лого Миры
2. Определить банк традиционно решаемых вычислительных задач при изучении алгоритмизации.
3. Разработка программ для решения вычислительных задач на языке Лого.
4. Проанализировать решение вычислительных задач на языке Лого.

Данная выпускная работа состоит из:

* Введения.
* Двух глав.
* Заключения.
* Списка литературы, используемой при написании работы.
* Приложений.

К работе прилагается дискета со следующим содержанием.

Глава 1. Язык Лого и его применение в преподавании информатики

## История появления языка Лого

"Лого - это философия образования и непрерывно развивающееся семейство языков, реализующих эту философию".[ ]

Основные педагогические идеи применения Лого, а также первые версии Лого-систем были разработаны профессором Сеймур Пейпертом (Seymour Papert) в Массачусетском Технологическом Институте в 60-х годах. В 1967 году группой профессора Пейперта совместно с группой Уоллеса Фойрцайга работавшей в фирме Bolt, Beranek, and Newman, была создана первая версия Лого.

Широкое распространение Лого связано с развитием персональных компьютеров в конце 70-х годов. Ряд компаний, таких, как, например, Terrapin Software, начал коммерческое распространение различных версий Лого. Одним из лидеров в этой области является основанная в 1980 году фирма Logo Computer Systems Inc. (LCSI), совет директоров которой возглавляет профессор Пейперт.

В 1985 году компания Logo Computer Systems Inc. начала распространять новую версию Лого - программу LogoWriter. Эта программа была признана одной из самых удачных, достаточно сказать, что в 1990 году система LogoWriter была названа читателями журнала Classroom Computer Learning лучшей образовательной программой десятилетия. Программа LogoWriter была переведена на десятки языков мира и была реализована на разных компьютерных платформах. В 1987 году Институтом новых технологий образования (ИНТО) была создана русская версия этой программы для компьютеров IBM, которая успешно распространяется.

Принципиально новые идеи были реализованы в середине 80-х годов в системе LEGO TC Logo (совместной разработке фирмы LEGO и LCSI), которая являлась расширением системы LogoWriter, однако в ней можно было управлять не только черепашками на экране, но и реальными роботами, собранными из деталей LEGO и подключенными к компьютеру.

Активная работа в области Лого продолжается и в настоящее время, а программа является русской версией программы MicroWorlds, разработанной в LSCI, 1993 г. и ПервоЛого (совместная разработка LCSI и ИНТ, 1996 г.). Среда Control Lab (разработанная компанией LEGO Dacta, русская версия программы называется ЛЕГО-Лаборатория и распространяется Институтом новых технологий образования) является естественным развитием идей, заложенных в системе LEGO TC Logo. Возникают и новые перспективные области исследований: среди таких проектов, например, LEGO-роботы (Programmable Brick) - дальнейшее развитие систем LEGO TC Logo и Control Lab, позволяющее создавать автономные роботы, управляемые миниатюрным специальным компьютером.

Лого, конечно, отнюдь не сводится к созданию и распространению компьютерных программ. Существует мировое сообщество преподавателей и исследователей, заинтересованных в развитии и распространении педагогической философии Лого. Публикуется различная литература, посвященная как проблемам использования Лого в преподавании конкретных школьных дисциплин (таких как математика, язык и т.д.), так и общим вопросам использования Лого в школьном и внешкольном образовании. Уже более 10 лет в США существует специальная некоммерческая организация Logo Foundation, координирующая деятельность мирового Лого-сообщества.

Институтом новых технологий образования рассматривается поддержка и развитие Лого-сообщества в России в качестве одной из своих основных задач. Институт разрабатывает и распространяет программы, основанные на Лого, а также различные методические и учебные материалы, связанные с Лого, который является не только языком программирования, но и языком общения.

Распространенность Лого-сред в Екатеринбурге

C 1999 года в г. Екатеринбурге достаточно активно используются Лого Миры. Первые курсы для учителей провели Сопрунов С.Ф. и Яковлева Е.И. Всю работу по внедрению и использованию Лого Миров, курирует Центр новых информационных технологий (ЦНИТ), созданный несколько лет назад при Управлении образования г. Екатеринбурга. Была создана городская экспериментальная площадка, в совет которой вошли наиболее квалифицированные в Лого преподаватели, целью, которой является апробация среды в учебном процессе общеобразовательных учреждений, исследование методики использования среды Лого Миры в обучении информатике для разных возрастных категорий учащихся и для разных форм учебной работы.[ ]

В городе действует Лого-клуб для преподавателей, который собирается в ЦНИТе раз в месяц, где преподаватели обмениваются идеями проектов, обсуждают свои проблемы, передают опыт.

В весенние каникулы с 2000 года проводится научно-практическая конференция, где преподаватели делятся своими наработками и методиками по определенным темам или целыми комплексами.

Школы города, такие как лингвистическая гимназия № 13, школа № 168, школа № 144, гимназия «Полифорум» и «Менталитет» №180 и другие, работают с Лого Мирами на уроках информатики, а также в кружках, факультативах, в группах дополнительного образования. В ЦНИТе ведутся занятия с детьми в рамках дополнительных образовательных услуг и используют среду в основном для создания тематических проектов, посвященной той или иной теме по изучению языка Лого Миры, чтоб заинтересовать детей.

Были проведены командные турниры (2000 - 2002 года) для школьников, в которых участвовало 32 команды из 25 школ в 2 возрастных группах: 5-7 класс и 7-9 классы. [Суриф Т.И., Волкова С.Б.]

Анализ сред Лого

Для чего вообще изучать программирование, а следовательно среду программирования? Очевидно, что процесс проектирования программ - это процесс решения проблем человеком, подобно другим процессам решения проблем в науке и технике. [Кузнецов И.Н.]

Программирование в компьютере подразумевает описание некоторой проблемы на определенном языке и последующее многократное моделирование с целью проверки модели и решения проблемы. Эффективное описание проблемы для моделирования требует развитого объектного мышления.

Правильно организованное обучение с использованием среды программирования развивает алгоритмическое и логическое мышление в естественной для этого обстановке; дает опыт работы с разными моделями; знакомит с общими принципами и методами программирования, что позволяет учащимся адаптировать приобретенные навыки при освоении других программных сред. [КИН]

Не должно быть изучения программирования без изучения конкретной программной среды. Выбор языка программирования осуществлялся по следующим критериям: мощность, достаточность, простота и элегантность.

Мощность языка должна позволять при использовании его для решения проблемы сосредоточиться целиком на ней, а не на языковых конструкциях. В Лого описание проблемы формулируется буквально в терминах этой проблемы и не требует переключаться на языковую специфику. Отсутствие в Лого жесткого контроля за ошибками программиста, вызывающее незаслуженные нарекания, на самом деле является не недоработкой среды, а сознательно запланированной ее особенностью, направленной на предоставление максимальной свободы для творчества при формулировании проблемы. [КИН]

Характеристика достаточности свидетельствует о нежесткой настроенности на вычислительные ресурсы компьютера. Фактически здесь система полностью скрыта от использующего ее и не отвлекает от проблематики задачи.

Простота языка выражается в неограниченных возможностях реализации мнемоники, а естественная для него процедурность позволяет на деле формировать соответствующие черты мышления учащихся.

Лежащая в основе языка рекурсивность дает реальную возможность учащемуся создавать программы. Развитые возможности структур данных превосходят многие известные языки, и позволяет на первое место выводить вопрос о структуре данных при разработке алгоритмов.

Перечисленные характеристики хорошо укладываются в идею о разработке специального языка для решения конкретного класса задач, что позволяет говорить об объектном подходе к проектированию программ.

Выбор именно этого программного продукта обоснован следующими требованиями к среде, развивающей абстрактное и любое другое мышление:

* наличие очевидных инструментов манипулирования понятиями;
* возможность усложнения инструментов по мере усложнения понятий;
* наличие средств, увеличивающих эмоциональное восприятие (мультимедиа вложения);
* возможность использования объектного (системного) подхода;

Сам по себе объектный подход это - идея, метод познания. На любом языке программирования можно показать такой подход, если учитель владеет им и предъявляет соответствующие требования к технологии разработки программ.

Объектно-ориентированное программирование возможно только в средах, имеющих для этого встроенные инструменты.

Классическая среда, разработанная для ОС DOS, - система Logo Writer – очень хорошо приспособлена для реализации объектного подхода при минимальной сложности языка программирования, является универсальной учебной компьютерной средой на базе языка Лого для дошкольного и начального школьного образования. В систему входят: текстовый, графический и музыкальный редакторы, возможность записи звука с микрофона и образное программирование с помощью пиктограмм. Система сочетает эти богатые возможности с простым графическим интерфейсом, благодаря которому ее пользователями могут быть дети от 4 лет.

Другая версия программного продукта - Лого Миры является объектно-ориентированной средой, предназначенной для использования в средней школе, является мультимедийной средой, универсальным инструментом на базе языка Лого - распространенного и эффективного педагогического средства, развивающего умение решать задачи, познавательные способности и творческое мышление детей. Эта программа, интегрирующая новейшие мультимедийные технологии, программирование и Интернет - возможности, предназначена для создания собственных проектов на любую тему.

Основные достоинства и недостатки среды Лого

Язык программирования Лого стал широко использоваться, и приобрел популярность в начальной, средней школе, в колледжах, университетах и для людей, занимающихся самообразованием. Это справедливо не только для зарубежной системы образования, но и для российской. Чтобы убедиться в этом, достаточно просмотреть журналы "Информатика и образование" за последние пару лет.

Что же вкладывается в понятие «мощный язык программирования»? Это качество не означает, что язык позволяет писать программы, выполняющие что-либо, не доступное другим языкам. В этом смысле все языки схожи. Если можно написать программу на Лого, то ее можно написать на Паскале или Бейсике.

Мощность языка программирования - это мера того, насколько язык помогает сконцентрироваться, в первую очередь, на конкретной проблеме, которую нужно решить, а не на преодолении ограничений, накладываемых самим языком. Например, для. С, Паскаля, Java и всех других языков, «выросших» первоначально из Фортрана, программист должен очень точно определять, что, как, и где хранится в памяти компьютера. Если нужно собрать 20 чисел вместе и рассматривать их как один объект, для этого «определяют массив», заявив заранее, что в нем будет храниться в точности 20 чисел. Очень плохо, если в последствии массив будет состоять из 21 числа. Также решается заранее, что массив будет содержать только целые числа, или числа с дробной частью, или же это будет символьный массив, и нельзя смешивать данные разных типов в одном массиве. Для Лого процесс размещения данных в памяти компьютера является полностью автоматическим. Если программа создала список из 20 чисел, память для его хранения отведется автоматически. Если позже добавить 21-е число, это также делается автоматически. [ ]

Более мощные языки базируются на конкретных математических моделях и реализуют эти модели в полной мере. Например, АПЛ базируется на идее матричной обработки; Пролог - на исчислении предикатов (раздел математической логики). Лого, как и ЛИСП, базируется на идее композиции функций (composition of functions). [ ]

Черепашка Лого живет в компьютерном мире, но ее поведение управляется командами понятными каждому ребенку. Язык Лого дал начало целому семейству обучающих средств. В одном из ветвей этого семейства в последние годы произошло возвращение к физическим управляемым моделям. При этом качественное отличие от робота-родоначальника состоит в том, что сегодня разнообразные управляемые модели собираются самими учениками из наборов Лего.

Язык Лого – заместительное средство для моделирования чего угодно. В распространении от одного до четырех исполнителей – черепашек, которые могут менять свою форму, создавать рисунки, двигаться по любым траекториям с разными скоростями, сообщить данные о той области экрана, где они находятся.

К его преимуществам можно отнести:

* возможность написания команд и идентификаторов, как на русском, так и на английском языке, как в полной, так и в сокращенной форме;
* крупный шрифт, уменьшающий напряжение глаз при работе;
* легкость написания динамических программ и программ для работы с текстовой информацией;
* наличие не одной, а 4-х черепашек, что позволяет легко создавать несложные игровые программы с небольшим числом персонажей;
* возможность использования внешних процедур, что способствует развитию умения работать с библиотекой алгоритмов;
* низкие требования к аппаратуре и совместимость с широким кругом IBM-совместимых компьютеров.

Но, как у любого продукта, у этой системы проявились и отрицательные стороны:

* невозможность визуально отличить русские буквы от латинских, что порождает множество ошибок;
* не полная совместимость с IBM-стандартом на клавиатуру;
* отсутствие готовой помощи (хотелось бы, чтобы в новой версии была контекстно-зависимая помощь и гипертекстовый справочник);
* отсутствие строки подсказки (назначений ключей);
* ограниченное число черепашек;
* неестественность логических связок;
* наличие только одного вида циклов - цикла Repeat.

Лого является прекрасной средой, помогающей освоить основные элементы программирования - переменные, функции, типы данных и другие. Среда Лого играет большую роль при изучении в старших классах универсальных языков программирования, таких как Паскаль. [Лысиков А.И.] В тоже время Лого является прекрасным средством для развития мышления и самостоятельных исследований в самых разных интеллектуальных областях и с различными уровнями сложности.

Применение Лого в учебном процессе начальной школы

Компьютерная техника и связанные с ней информационные технологии проникают во все сферы деятельности человека и становятся естественной средой обитания. Понимание того, что может и (что не менее важно) чего не может делать компьютер, представление о сути управления, знания о типах информации и способах ее обработки необходимо сформулировать у детей уже в раннем возрасте. В свою очередь развитие логико-арифметического стиля мышления является, как известно, мощным фактором развития ребенка, способствующим повышению качества его учебной деятельности. Все это определяет актуальность изучения информатики уже в начальной школе (1 – 5 классы). Пропедевтические курсы, как сказано в стандарте, должны формировать навыки сознательного и рационального использования компьютера в учебной, а затем и в профессиональной деятельности.

В процессе создания творческой работы часто требуется не только собрать материал, красиво его оформить, но и построить компьютерную модель, изучаемого процесса, провести компьютерный эксперимент. Таким образом, просто необходимо, чтобы уже к 7-8 классу школьник владел не только простыми редакторами, но и мог разработать алгоритм, а затем воплотить его на конкретном алгоритмическом языке. Эта одна из причин, по которой в начальной школе нужно применять обработку различных типов информации (звук, графика, текст), изучение алгоритмических структур и создание простейших программ на компьютере с помощью объектно-ориентрованной среды Лого Миры. [Шелегова И.А.]

В соответствии с образовательным стандартом цель этих курсов – обеспечение прочного и сознательного освоения основ информатики, развитие устойчивого интереса к предмету, использование ПК как средства развития учащихся, овладение элементарными навыками использования компьютерных технологий. Все это формирует навыки учащихся сознательного использования ПК как инструмент в практической деятельности, обработки различных видов информации средствами современных компьютерных технологий, алгоритмического подхода к решению вычислительных задач.

Курсы не только прививают учащимся элементы информационной культуры, под которой понимается умение целенаправленно работать с информацией на ПК, но и помогают детям расширить свой кругозор, развить логическое мышление, творческий потенциал, связать с другими учебными предметами. [Кадочникова В.И., Бурнева Л.С., Молодых Н.Г.]

Достоинства курсов в том, что снимаются психологические барьеры во взаимодействии учащихся с компьютером, развивают потребность в творческой деятельности, создают условия для самовыражения ребёнка, усиливают мотивацию к обучению и т.д.

В то же время изучения Лого в начальной школе ориентировано на то, чтобы дать детям в руки инструмент, с помощью которого они смогли бы в среднем звене создавать проекты, помогающие исследовать окружающий мир и адаптироваться в нем.

Применение Лого для внешкольной работы по информатике

В соответствии с концепцией школьного образования в настоящее время созрела необходимость создания оптимальных условий для развития каждого ученика, для формирования человека с новым уровнем сознания, способного к самооценке и критическому мышлению. Одним из путей реализации является организация обучения, таким образом, чтобы учащийся имел возможность не только освоить обязательный курс базового образования, но и получить дополнительные знания высокого уровня, необходимо для продолжения обучения в ВУЗах. [Которов Н.П.].

Внеучебное время занимает значительное место в жизни школьников. Глубокий внутренний смысл внешкольной работы во всех ее формах состоит в привлечении детей к активному участию в общественно-полезной деятельности, в стимулировании инициативы и самостоятельности, в развитии индивидуальных интересов, склонностей и способностей.

Внешкольная работа с учащимися представляет собой органическую часть и важный элемент во всей учебно-воспитательной деятельности школы.

Занимаясь любимым делом с интересом и увлечением, учащиеся имеют возможность проявить большую активность, самостоятельность. Развитие этих качеств – необходимые условия для воспитания творческой инициативы. Поэтому опора на самостоятельность, активность, инициативу учащихся является важной особенностью внешкольной работы с детьми.

Очень важно, чтобы учащиеся во внеклассной работе в школах и внешкольных учреждениях занимались посильными для них делами, что характер творческой деятельности юных энтузиастов соответствовал их возрастным особенностям.

В результате систематического участия во внешкольной работе повышается уровень общей культуры учащихся. Школьники начинают понимать перспективы развития конкретной отрасли, ее место и роль в жизни. В личности школьника происходят большие изменения: улучшается поведение, отношение к коллективу, развивается творческая самостоятельность и профессиональная устремленность, умения использовать знания и опыт для решения практических задач.

Все это может служить общим критерием эффективности внешкольной работы с учащимися по информатике.

Одним из эффективных способов работы с учащимися является кружок.

Кружок – это форма организации творчества школьников во Внеучебное время. Это добровольное объединение учащихся, проявляющих общий интерес к той или иной конкретной области информатики и стремящихся занимать практической деятельностью в этой области.

Основы работы такого объединения составляют совместное изучение учащихся при создании мультфильмов на заданную тему, формирование навыков алгоритмизации в среде Лого Миры, формирование навыков работы с объектами мультимедийной среды Лого Миры.

Обзор публикаций о применении Лого в учебном процессе школы

В процессе исследования были проанализированы ряд статей, опубликованных в журналах «Информатика и образование» и в газетах «Информатика» (приложение к газете «Первое сентября») за последние 4 года с целью выявления информации, которая связана с: применением алгоритмов в среде Лого Миры, реализация их при решении задач, применение Лого Миры на уроках.

Приведем обзор статей, которые содержат данные материалы.

Автор статьи [1] рассматривает решение логических задач средствами алгебры логики. Описывается множество подходов к решению логических задач.

В публикации рассматривается только первый способ.

Статья [2] знакомит читателей с опытом работы со средой Лого Миры в школах Екатеринбурга. Рассматривается обучающая программная среда Лого Миры, разработанная на базе языка Лого. Говорится о методике использования среды Лого Миры для учащихся разных “возрастных категорий” (младших школьников и учащихся среднего звена) и для разных форм учебной работы.

Авторы приводят доводы в пользу применения среды Лого Миры в обучении информатике для разных возрастных категорий учащихся и для разных форм учебной работы, излагают некоторые наиболее интересные результаты работы (конкретные разработки, курс алгоритмизации с использование среды Лого Миры).

В статье А.Г. Юдиной [4] представлена рецензия на учебное пособие для общеобразовательных школ А.Г. Юдиной “Практикум по информатике в среде LogoWriter”, также представлен фрагмент учебного пособия с предисловием автора, в том числе глава 7 «построение графиков функций».

Рассказ о курсе алгоритмизации на основе использования Лого Миров, а также проекты и классно-поурочная система в обучении алгоритмизации представлено в статье И.А. Шелеговой и Н.А. Юнерман [5]. Авторы статьи утверждают, что “Основная цель – сформировать у учащихся уже в начальной школе навыки самообразования, помочь ребенку в самовыражении, развить его коммуникативные навыки”.

Статья А.А. Зайцева [6] рассказывает о методическом пособии, написанном в жанре увлекательной повести, посвященной языку Лого, работе с Лого-системой. Это – один из компонентов программного обеспечения. Распространяемого в свое время в рамках “Пилотные школы”.

Графика в Лого и поурочная методическая разработка по информатике для класса УНЦ в 7-8 классах рассмотрена в статье [7] Г.В. Луканина.

В статье [9] Н.Г. Ивановой, О.Л. Русаковой предлагаются конспекты занятий по информатике для учащихся 5-6 классов. Цель данного курса – дать первые представления о формах познания и законах логического мышления, развить алгоритмическое мышление, обучить приемам написания и отладки программ, использующих процедуры с параметром.

Курс рассчитан на 15 двухчасовых занятий.

Автор статьи [2д] Т.В. Головина знакомит читателей с тем, как в ее школе организована работа над индивидуальными и коллективными разработками на уроках информатики, на примере 7 класса при изучении темы Лого. Описываются сложности, с которыми может столкнуться учитель при осуществлении коллективного проекта, и приводится примерный ход уроков по изучению возможностей Лого.

Статья может помочь студентам и начинающим учителям в планировании своих уроков по данной теме, разобраться в некоторых тонкостях при работе с объектами среды, содержит примеры создания новых форм черепашки и некоторые кадры реализованной разработки.

В статье [18д] высказывается соображение, что учителю информатики следует добиваться наиболее тесной взаимосвязи уроков информатики и традиционных школьных дисциплин. По мнению авторов, уроки информатики должны обеспечить компьютерную поддержку традиционных дисциплин, учитель информатики должен координировать свою программу с программой других учителей. Далее предлагается набор разработок, основанных на использовании программирования Лого. Для каждой разработки указаны те традиционные школьные предметы и те компьютерные понятия, которые естественным образом затрагиваются при работе над этой разработкой. Хотя предложенные разработки рекомендуются для выполнения в начальной школе, в тоже время некоторые варианты помогут определиться студенту или начинающему учителю в выборе темы для разработки.

Общий вывод: публикаций достаточно много, темы их разнообразны, что говорит о популярности среды Лого Миры на занятиях по информатике в школе.

Роль алгоритмизации в базовом курсе

Содержание базового курса информатики, предусмотренное государственными стандартами образования, сочетает в себе три основных направления, отображающие важнейшие аспекты общеобразовательной значимости:

* мировоззренческая линия, связанная с формированием представлений о системно-информационном подходе к анализу окружающего мира;
* алгоритмическая линия, связанная с развитием логического мышления школьников;
* пользовательская линия, связанная с формированием компьютерной грамотности, подготовкой школьников к практической деятельности в условиях широкого использования информационных технологий.

Изучение учебного материала по алгоритмической линии обеспечивает учащихся возможностью:

* понять на основе анализа примеров, смысл понятия алгоритма, знать свойства алгоритмов, понять возможность автоматизации в деятельности человека при использовании алгоритмов;
* освоить основные алгоритмические конструкции, научиться применять алгоритмические конструкции для построения алгоритмов решения задач;
* получить представление о библиотеке алгоритмов, уметь использовать библиотеку для построения более сложных алгоритмов;
* получить представление об одном из языков программирования, использовать этот язык для записи алгоритмов решения задач.

Этим же стандартом предусмотрены следующие требования к подготовке учащихся.

**Учащийся должен:**

* понимать сущность понятия алгоритма, знать его основные свойства, иллюстрировать их на конкретных примерах алгоритмов;
* понимать возможности автоматизации деятельности человека при использовании алгоритмов;
* знать основные алгоритмические конструкции и уметь использовать их при построении алгоритмов;
* записать на учебном алгоритмическом языке алгоритм простой задачи;
* уметь составлять программы на одном из языков программирования для решения вычислительных задач.

При изучении основ алгоритмизации в средней школе основное внимание в первую очередь должно уделяться:

* выявлению общих закономерностей и принципов алгоритмизации;
* основным этапам решения задач при помощи современных информационных технологий;
* анализу поставленной задачи, методам формализации и моделирования реальных процессов и явлений;
* выбору исполнителя поставленной задачи, исходя из тех рассуждений, что он является определенным объектом с присущими ему свойствами и набором действий, которые нуждаются в анализе для правильного и эффективного их использования;
* методам и средствам формализованного описания действий исполнителя, современным средствам их конструирования и реализации при помощи компьютера.

Главной целью изучения основ алгоритмизации в школе является:

* развитие алгоритмического, конструктивного и логического мышления учеников;
* формирование операционного типа мышления, которое направлено на выбор оптимального решения определенной поставленной задачей;
* развитие интеллектуальных умений через изучение технологии программирования;
* применение как универсального средства, которое может создавать себе интеллектуальных партнеров в любой сфере, где ученик может проявить себя как автор сценария, как программист;
* развитие алгоритмического мышления путем создания программ средствами языка в конкретной среде программирования;
* формирования приемов умственной деятельности (анализ, синтез, обобщение);
* развитие памяти, фантазии, интуиции.

Развитие этих специфических видов мышления делает весомый вклад в развитие общего научного мировоззрения и умственных способностей личности учащихся. [Копаев А.В., Триус Ю.В.]

В тоже время алгоритмизация, как раздел информатики, который изучает процессы создания алгоритмов, традиционно относится к теоретической информатике вследствие своего фундаментального характера. Следовательно, при развитии новых информационных технологий, и в частности технологий программирования, появляется возможность в пределах раздела "Основы алгоритмизации" давать общенаучные понятия информатики, и в то же время формировать и развивать умение, и навыки необходимые пользователю при работе с современным программным обеспечением, [ст.24], т.е. появляется возможность сделать раздел "Основы алгоритмизации" мостиком между теоретической и практической информатикой. Первые шаги в этом направлении уже сделаны. Стоит вспомнить работы А. Г. Кушниренка, Ю. А. Первина, А. Л. Семенова по внедрению "конструктивистской" парадигмы при изучении теоретической информатики. Одним из принципов этой парадигмы является самостоятельное добывание учениками знаний, которые формируются при работе с реальными и виртуальными объектами. Реализация этого принципа основывается на использовании творческих сред, таких как, например, Лого, Кумир, Роботландия.

Традиционно считалось, что знание языков программирования и умение ими пользоваться просто необходимо при работе с компьютером. Поэтому в школе учащихся обучают какому-нибудь алгоритмическому языку. А т.к. алгоритмические языки достаточно сложны, то это преподавание осуществлялось в старших классах. Сейчас появились различные адаптированные программные среды, в которых процесс программирования и составления алгоритмов доступен даже младшим школьникам. Это позволяет обучать основам программирования и построения алгоритмов, не вдаваясь в подробности того или иного сложного языка программирования. Наиболее благоприятной средой для изучения темы "Алгоритмизация" является среда Лого Миров. Она позволяет плавно перейти от игры к разработке серьезных алгоритмов, к созданию серьезных проектов. Причем сама среда такова, что нет необходимости искать побудительные мотивы для занятий в ней. Работа в редакторе форм, создание анимированных проектов привлекает учащихся, развивает их творческие способности и не надоедает им от урока к уроку. Наибольший интерес и понимание эта тема вызывает у школьников 5 класса. Там используется язык программирования Лого, который признан учебным для детей младшего и среднего школьного возраста. Уже в 5 и 6 классе, с его помощью, дети изучают все конструкции для структурного программирования: вызов процедур, ветвление, циклы. [Глинка Н.В.]

Раздел "Алгоритмизация" является одной из важнейших тем школьного курса информатики. Она красной ниточкой проходит через весь курс, усложняясь от класса к классу. Если на начальном этапе - это игра в алгоритмы, то в дальнейшем - это самостоятельное составление алгоритмов разного вида с использованием различных сред, в том числе и языков программирования.

Выбор начального языка программирования имеет принципиальное значение, так как от этого во многом зависит методика изучения курса, содержание и последовательность предъявления учебного материала, система учебно-познавательных заданий и, главное, вся дальнейшая работа по овладению программированием для решения реальных практических задач на ЭВМ. [Иванова Л.В. Юрзанова Т.К.]

Структура Лого подчинена методической системе, позволяющей последовательно вводить новые понятия, формировать умения и навыки операционного стиля мышления. Среди достоинств этого языка выделяем такие, как близость синтаксиса к естественному языку, приспособленность к интерактивному режиму работы, ориентация на формирование самых общих представлений о программировании, наличие системы графических примитивов, возможность использования процедур, позволяющих реализовать идеи структурного программирования. Наличие в Лого присущих всем алгоритмическим языкам арифметических и логических операций, большого количества стандартных функций позволяет перейти на заключительном этапе обучения к более сложным элементам программирования.

Лого–среды позволяют совершенно в новом ракурсе взглянуть на преподавание основ программирования. Являясь прекрасным образцом мультимедийных сред, адаптированных для обучения, они позволяют организовать обучение программированию на основе математических задач, а на этапах мультипликации: от разработки сценария до “оживления” и озвучивания персонажей. В них не только сохранены и расширены возможности изучения основных алгоритмических конструкций (ветвления, различные циклы, организация диалога), но и имеется разнообразный спектр атрибутов увлекательного программирования различных вычислительных задач. [Иванова Л.В, Юрзанова Т.К.]

Тема "Алгоритмизация" в школах РФ изучается с 5-х по 11 класс. В 5-х - 7-х классах учащиеся знакомятся с основными алгоритмическими конструкциями.

Обучение строится на последовательности решаемых задач, которые подбираются по следующим принципам:

1. От простого к сложному;
2. Новизна;
3. Наследование.

Алгоритмические конструкции вводятся по мере необходимости при решении конкретных задач.

Например, при изучении циклических алгоритмов вначале дается построение многоугольников, увеличивая количество их сторон, и, подводя к мысли, что, чем больше сторон у многоугольника, тем он больше напоминает окружность. Затем учащиеся приходят к выводу, что, используя линейный алгоритм такой многоугольник построить невозможно, необходимы новые знания.

Как итог изучения темы "Алгоритмизация" в среде LOGO, в 7-ом классе несколько уроков отводится для анимации. В этом разделе внимание учащихся акцентируется на работе в программном режиме, закрепляются навыки работы с процедурами и другими алгоритмическими конструкциями. Рисуя, учащиеся не только усваивают алгоритмические конструкции и их реализацию в различных средах, но и лучше понимают ЭВМ как формального исполнителя и то, что из принципа формального исполнения следует, что ни исполнитель, ни ЭВМ не могут совершать ошибок. Все ошибки (синтаксические, семантические и логические) совершает человек. Так как в 8-х - 11-х классах школьники уже знакомы с алгоритмическими конструкциями, преподавание можно вести от теории к практике, дополняя данные ранее определения алгоритмов, и изучать их реализацию на языке программирования.

Технология решения вычислительных задач

Под процессами решения вычислительных задач на ЭВМ надо понимать совместную деятельность человека и компьютера. Условно данный процесс можно представить в виде нескольких последовательных этапов. На долю человека приходятся этапы, связанные с творческой деятельностью-постановкой, алгоритмизацией и программированием задач, анализом результатов, а на долю компьютера - этапы обработки информации в соответствии с разработанным алгоритмом.

Рассмотрим, из чего складывается процесс решения вычислительных задач на ЭВМ, какие пройти этапы, чтобы достичь конечной цели – решить задачу. Следует заметить, что приведенные ниже последовательность действий ориентирована на решение задач любой сложности. Для простейших задач некоторые этапы, возможно, не понадобятся. Для более сложных задач некоторые этапы могут существенно усложниться

**Этапы решения задачи на ЭВМ**

|  |  |
| --- | --- |
| 1. Постановка задачи 2. Математическая формализация (описание задачи) 3. Построение алгоритма | Работа без применения ЭВМ |
| 1. Составление программы на языке программирования 2. Отладка и тестирование программы 3. Проведение расчетов и анализ результатов | Работа на ЭВМ |

**Первый этап – постановка задачи**

На этом этапе участвует только человек, который хорошо представляет предметную область задачи. Он должен четко определить цель задачи, на основании анализа имеющейся информации, привести описание каждого исходного данного и указать место его хранения, дать словесное описание содержания задачи и предложить общий подход к ее решению.

**Второй этап – математическая формализация описания задачи**

Этот этап выполняет человек, способный разработать математическое описание поставленной задачи. Целью этого этапа – создать такую математическую модель решаемой задачи, которая может быть реализована в компьютере. Существует целый ряд задач, где математическая постановка сводится к простому перечислению формул и логических условий.

**Третий этап – построение алгоритма**

На основе математического описания необходимо разработать алгоритм решения. Чаще всего алгоритм изображается в виде блок-схемы с четко определенной последовательностью действий. Алгоритм следует составлять, используя обозначения переменных и избегая конкретных значений констант. Алгоритм должен быть понятным любому пользователю и быть пригодным для решения задач.

**Четвертый этап – программирование**

Этот этап выполняет пользователь, умеющий программировать. Программа – это представление алгоритма с помощью специальных символов, воспринимаемых компьютером. Составление программы обеспечивает возможность физической реализации алгоритма и соответственно поставленной задачи. Во многих задачах при программировании на алгоритмическом языке часто пользуются заменой блока алгоритма на один или несколько операторов.

**Пятый этап – отладка и тестирования программы**

При выполнении первых этапов часто бывают допущены ошибки. Тестовая задача (тест) или контрольный пример – это совокупность исходных данных, на основании которых заранее определяются значения выходных данных. Программа и исходные данные тестовой задачи вводятся в оперативную память компьютера. Результаты решения сравниваются с полученными расчетами значениями. По результатам сравнения пользователь делает вывод: программа работает правильно, если результаты совпали, в противоположном случае в программе есть ошибки. Ошибки надо найти, вновь рассчитать тестовую задачу, и так до тех пор, пока не будет обеспечено полного совпадения результатов.

**Шестой этап – проведение расчетов и анализ результатов**

После устранения всех ошибок, выявленных тестовой задачей, можно перейти к получению результатов решения поставленной задачи. Подготавливаются исходные данные этой задачи и вводятся в ЭВМ. Полученные в результате решения выходные данные анализируются постановщиком задачи. На основании этого анализа вырабатываются соответствующие решения, рекомендации, выводы и т. д.

Рассмотренные этапы процесса решения вычислительных задач на ЭВМ позволяют определить роль человека и компьютера в этом процессе. Все сказанное выше характеризует давно сложившийся и устоявшийся подход к процессу решения вычислительных задач, когда основную творческую работу и работу, связанную непосредственно с программированием, выполняет человек, а компьютер участвует только как вспомогательное средство.[Жигорев А.Н, Макарова Н.В., Путинцева М.А.]

ГЛАВА 2. Решение вычислительных задач в среде Лого Миры

2.1. Описание возможностей среды Лого Миры 2.0

Эту версию известной программы Лого Миры можно назвать мультимедийное Лого, а также - Интернет-Лого. Прежние возможности Лого Миров теперь дополнены новейшими технологиями, предоставляемые современными мультимедиа-компьютерами. В частности, можно вставлять в работы фрагменты аудио - и видео-записей, QuickTime-звуки и фильмы. Работу, сделанную в ЛогоМирах 2.0, можно легко "опубликовать" в Интернет. Специальное приложение (Web-плеер) позволит посетить страницу проекта прямо в сети через браузер. С момента своего рождения в 1967 году в Лаборатории Искусственного Интеллекта Массачусетского технологического института язык программирования Лого, сохраняя, неизменными свои основные черты, постоянно развивается и совершенствуется. Особенно популярен Лого как начальный язык программирования. Благодаря таким его качествам как доступность языка, быстрое написание программ, наглядность, т.е. к составлению программ учащийся может приступить, в начале обучения. Однако Лого используют и профессиональные программисты.

Язык системы Лого Миры представляет собой неформальное описание языка программирования Лого. Язык Лого, как и большинство языков программирования, имеет много диалектов. Примитивы Лого Миры включает в себя описание всех примитивов (основных команд и операций) языка Лого.

Понятия, которыми можно манипулировать в среде Лого Миры

*Объекты*: проекты, листы проекта, черепашки, рабочее поле, текстовые окна, кнопки, бегунки, формы, рисунки, звуковые фрагменты, внешние мультимедиа объекты.

*Общие свойства объектов*: цвет, размер, положение на Рабочем поле. Приватные свойства объектов – в зависимости от типа. Например, Рабочее поле может занимать часть экрана (режим разработки) или весь экран – (режим демонстрации).

*Действия объектов*: создавать, удалять, менять свойства (передвигать, менять цвет, размер и т.п.). Все общие действия могут выбираться визуально на панели инструментов или описываться программно. Встроенный редактор для создания рисунков, встроенный редактор для создания звукового фрагмента.

*Интерфейс*: обычный для любого приложения Windows – окно, основное меню, панели инструментов. Команды и программы выполняются в режиме интерпретации, поэтому на экране – Рабочее поле и Командный центр.

*Язык*: классический Лого, в который добавлены инструменты для работы с объектами.

*Типы данных*: числа, включая числа с плавающей точкой, слова, списки, объекты. Возможен импорт и экспорт текста, графики через буфер обмена Windows, а также связывание с другими мультимедиа объектами.

*Алгоритмические конструкции*: процедуры, функции, циклы разных видов, разветвление, описание процессов.

*Демонстрация проекта*: включен инструмент для демонстрации проектов - плеер, занимающий мало места.

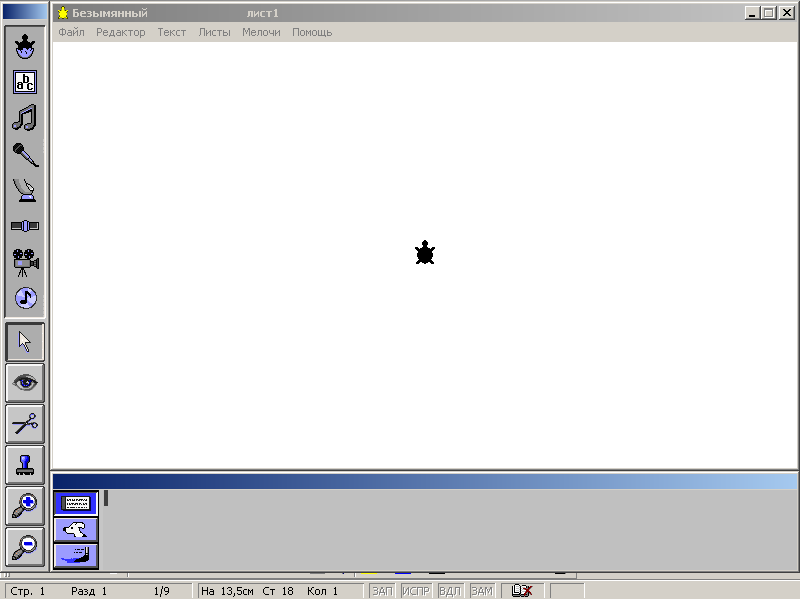
*Помощь*: база данных, содержащая справочную информацию, в формате HELP Windows.

*Дидактические материалы*: предлагаются готовые проекты для демонстрации, заготовки для дальнейшей разработки, некоторые идеи для реализации.

Запуск Лого Миров

Лого Миры можно запустить с помощью ярлыка на Рабочем столе Windows или через главное меню (Программы Лого Миры)

На экране дисплея появляется следующее изображение:



*Рис.1 Графический экран системы Лого Миры*

Графический экран системы Лого

Графический экран системы Лого состоит из 3-х окон:

Графическое окно - предназначено для действий "черепашки" и оставляемых ею рисунков.

Командное окно - предназначено для ввода команд адресованных только "черепашке", а также команд перехода в тот или иной режим команд, обращенных к системе Лого.

Приборное окно - показывает значения основных параметров, характеризующих состояние Черепашка:

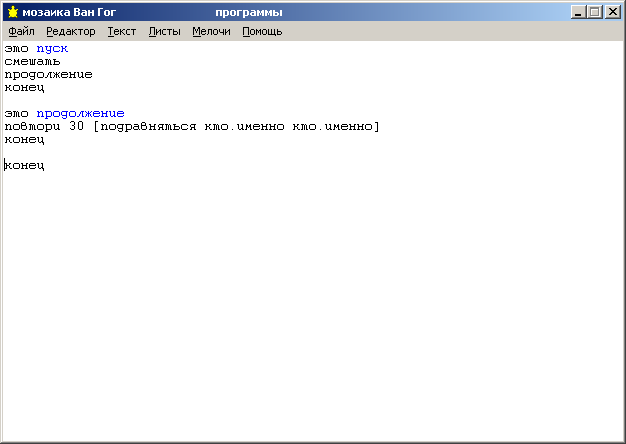
Текстовый экран системы Лого

Команды Листы→Программы осуществляют переход от графического к текстовому экрану.

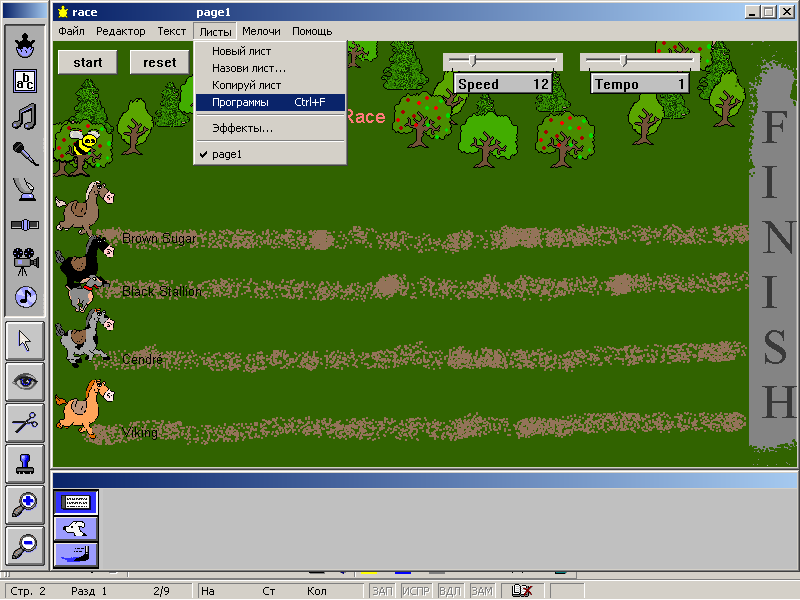
Режимы работы

В Лого предусмотрена работа в следующих режимах:

1. Режим непосредственного исполнения команд в графическом окне (рис.1), где работа ведется непосредственно с графическим окном. При этом команды задаются в самом нижнем окне.



*Рис.2. Тестовый экран системы Лого Миры*



*Рис.3.Меню Листы*

1. Режим подготовки и исправления текста программы в текстовом экране (рис.2), предназначен для создания и редактирования текста программ.

Переход от графического экрана к текстовому осуществляется с помощью меню (рис.3).

Команды системы Лого

Можно выделить две основные группы команд:

1. управления движением Черепашка;
2. управления системой.

3. Команды управления движением Черепашка

Могут задаваться как в графическом, так и в текстовом экране. К ним относятся:

1. Базовые команды движения и рисования;

2. Команда цикла (повторения);

3. Команда процедура (подпрограммы);

4. Команда изменения переменных;

5. Команда выбора (ветвления);

6. Рекурсивные команды.

##### Базовые команды движения и рисования

**Команды относительного перемещения**

Для перемещения Черепашки вперед и назад служат команды:

|  |  |  |
| --- | --- | --- |
| **FD** | FORWARD | ВПЕРЕД |
| **BK** | BACK | НАЗАД |

После команды нужно указать, на сколько шагов должна продвинуться Черепашка. Например, FD 50 (Черепашка продвинется на 50 шагов вперед) или BK 100 (Черепашка продвинется на 100 шагов назад). Между командой и числом - пробел.

**Команды управления пером**

|  |  |  |
| --- | --- | --- |
| **PU** | PEN UP | ПОДНЯТЬ ПЕРО |
| **PD** | PEN DOWN | ОПУСТИТЬ ПЕРО |

Двигаясь по рабочему полю Черепашка оставляет за собой прямую линию словно рисует пером. Пером можно управлять, используя вышеперечисленные команды.

**Команды поворота Черепашки**

|  |  |  |
| --- | --- | --- |
| **RT** | RIGHT | НАПРАВО |
| **LT** | LEFT | НАЛЕВО |

Эти команды требуют обязательного параметра - угла поворота в градусах. Например, LT 90 (Черепашка поворачивается налево на 90 градусов), RT 120 (Черепашка поворачивается направо на 120 градусов).

**Команды работы с цветом**

До сих пор Черепашка рисовала белым по черному. В нашем распоряжении есть еще 14 цветов - всего их 16, оттенков 123. Цвета пронумерованы от 0 до 139. Номер черного цвета - 9; номер белого - 0. Чтобы изменить цвет Черепашки (и цвет ее пера), нужно дать команду:

|  |  |  |
| --- | --- | --- |
| **SETC** | SET\_COLOR | НОВ\_ЦВЕТ |

и через пробел указать номер цвета. Например, после выполнения команды SETC 15 Черепашка станет красной, и будет оставлять такой же след. Для закраски частей экрана служит команда FILL (КРАСЬ). Черепашка, получив эту команду, заливает экран той же краской, какой покрашена сама (если перо Черепашки опущено). Чтобы Черепашка закрасила контур, необходимо выполнить следующие действия:

|  |
| --- |
| поднять перо |
| переместить Черепашку внутрь контура |
| опустить перо |
| дать команду FILL |

Черепашка может рисовать на экране. Чтобы наглядно представить это себе говорят, что она таскает за собой перо. Иногда говорят, что она чертит линии на песке своим хвостом. Вы можете использовать любое из этих представлений, говорите так, как вам больше нравиться. Если перо (хвост) опущено, то она оставляет след, если поднято - то при передвижении следа не остается.



*Рис.4. Поле команд*

Команда SETPENSIZE [ширина] - устанавливает размера пера. Перо всегда имеет прямоугольную форму. Например, setpensize 4

**Стирание графики**

Для того, чтобы стереть рисунок, но сохранить положение Черепашки, применяется команда:

|  |  |  |
| --- | --- | --- |
| CLEAN |  | СОТРИ |

Для восстановления исходного состояния графического экрана – рисунок стирается, Черепашка в исходном состоянии:

|  |  |  |
| --- | --- | --- |
| CLEARSCREEN | CS | СГ |

**Последовательность команд - программа для Черепашки**

Можно управлять Черепашкой, нажимая клавишу ввода после каждой команды. Если же записать подряд несколько команд (отделяя, их друг от друга пробелом) и только после этого нажать клавишу ввода, то Черепашка быстро выполнит команды одну за другой.

Например: FD 100 RT 90 FD 50 RT 90 FD 50 RT 90 FD 50.

Команды цикла (повторения)

**Команда REPEAT (повторить)**

При написании программ нередко приходится многократно повторять одни и те же команды. Для того, чтобы ускорить процесс написания программ предусмотрен специальный оператор цикла. Он позволяет строить программу из более крупных "блоков", избегая переписывания повторяющихся команд.

Циклом (повтором) называется такая форма организации действий, при которой одна и та же последовательность действий совершается несколько раз до тех пор, пока выполняется необходимое число раз.

В языке Лого оператором цикла является команда REPEAT (повторить).

**REPEAT число раз [ действие 1 , ... , действие n ]**

**ПОВТОРИ число раз [ действие 1 , ... , действие n ]**

Здесь можно выделить такие составные части:

* Собственно оператор REPEAT (повторить);
* Указатель количества повторений (число раз);
* Указатель начала цикла ( [ );
* Повторяющаяся цепочка команд (действие 1 , ... , действие n);
* Указатель конца цикла ( ] ).

Прочитав, в наборе команд REPEAT Черепашка ищет указание числа повторений. Запомнив его место, Черепашка будет выполнять команды, пока не встретит указатель конца цикла - правую квадратную скобку. После этого Черепашка возвращается к началу цикла и проверяет, выполнила ли она цикл заданное число раз. Если нет, то она продолжает выполнять цикл, в противном случае (если она уже выполнила цикл задуманное число раз) Черепашка начинает выполнять команды, расположенные после квадратных скобок. Таким образом, тело цикла выполняется задуманное число раз.

Построение процедур (подпрограмм)

Любое изображение, которое нарисует Черепашка, можно получить с помощью базовых команд. Но бывает необходимо выполнить несколько раз большие фрагменты программы. Чтобы избежать утомительного повтора, можно этот фрагмент оформить как (подпрограмму) процедуру для многократного использования. Для этого сначала этому фрагменту дают новое название, а затем используют его как имя команды для вызова в программе.

|  |  |
| --- | --- |
| **ТО** < имя команды >  < описание  фрагмента >  **END** | **ЭТО** < имя команды >  < описание  фрагмента >  **КОНЕЦ** |

Таким образом, можно определить любую процедуру, при этом следует соблюдать следующие правила:

* нельзя называть процедуры словами, за которыми в Лого закреплено определенное значение, такие слова называются служебными.
* название должно содержать не больше 15 символов и состоять из одного слова.

В остальном слова могут быть любыми.

Обязательно:

* Сначала идут все процедуры, потом - основная программа.
* ТО ... END - только для подпрограмм.
* Основная программа оформляется как подпрограмма. Для запуска ее необходимо указать только имя.

Команды изменения переменных

**Переменные величины**

В курсе школьной математики изучают переменные величины, и числа в математике часто обозначаются буквами.

Например: x=a+b, где a и b – переменные.

В языке Лого переменная - это любая последовательность символов (букв и цифр), начинающаяся с буквы. Она не должна содержать в себе пробел и должна быть больше 16 символов.

Признаком переменной является символ ":", с которого обязательно начинается переменная. Например: :A :N1 :INZ5

**Оператор присваивания**

: <переменная> = <арифметическое выражение>

Арифметическое выражение включает в себя константы, переменные, знаки арифметических действий ( +, -, \*, / ) и скобки.

|  |  |  |  |
| --- | --- | --- | --- |
| :А = 25 | :N = :N + 1 | :B = 125 \* ( :M + 1 ) / 100 | :C = 270 \* :N |

В Лого работает только целая арифметика. Поэтому, результатом деления является целая часть частного.

Команды и процедуры с параметрами

В Лого переменные могут использоваться в качестве параметров движения Черепашка, для управления последовательностью действий Черепашка, в процедурах с параметрами и в рекурсивных командах.

Базовые команды Черепашка (FORWARD, RIGHT, REPEAT и пр.) состоят из команды (оператора) и параметра (операнда). В качестве параметров использовали число (пример "квадрат", здесь три команды с параметром: REPEAT 4, FORWARD 50, RIGHT 90). Можно в качестве параметра команды использовать переменную, а не число. Допустим, нужно нарисовать квадраты с разной (переменной) стороной. В этом случае в команде FORWARD в качестве параметра используем переменную: FORWARD :X

Команда выбора (ветвление) в Лого

В некоторых случаях необходимо определить условие тех или иных действий, выполняемых Черепашка. Для этого используется команда выбора IF (ЕСЛИ)

|  |  |
| --- | --- |
| **IF** <условие>  [ < команда или серия команд, если условие выполняется > ]  [ < команда или серия команд, если условие не выполняется >] | **ЕСЛИ** <условие>  [ < команда или серия команд, если условие выполняется > ]  [ < команда или серия команд, если условие не выполняется >] |

Условие имеет вид отношения, при этом используются обозначения:

< - меньше, <= - меньше или равно,

- больше, >= - больше или равно,

= - равно, < > - не равно.

В случае записи условных операторов нужно:

* не забывать про квадратные скобки;
* записывать по одному оператору на строке.

В этом случае запись условного оператора становится более наглядной и человеку удобно читать и понимать программу, может иметь место многократная вложенность команд IF.

Рекурсия

В предыдущих разделах были использованы подпрограммы-процедуры, которые вызывались по мере необходимости. Рассмотрим процедуру, которая называется рекурсией. Суть рекурсии заключается в том, что в ходе ее выполнения процедура сама вызывает себя.

Заметим, что такая возможность имеется далеко не во всех языках программирования. К, примеру, Бейсик рекурсии не имеет, а в Лого - это наиболее употребляемый прием.

Функция

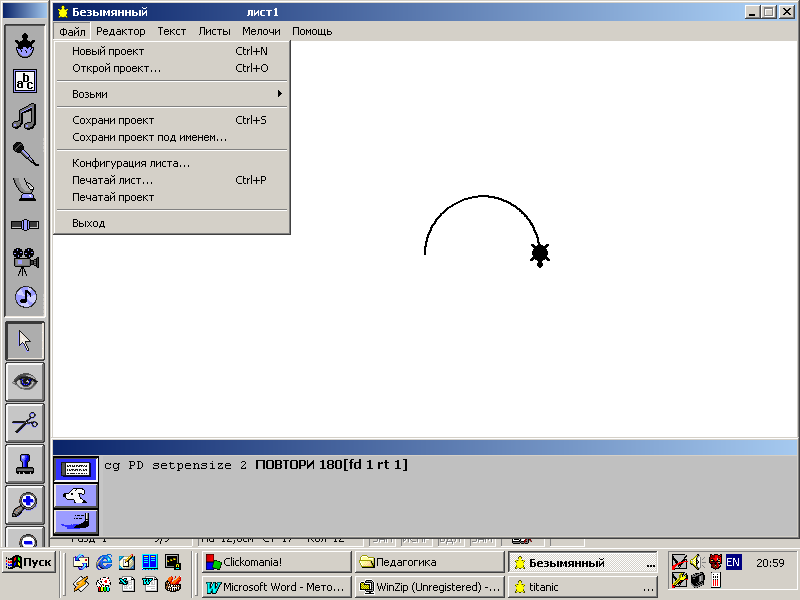
Можно провести аналогию с обычной записью математических выражений и порядком их вычисления.

В число Лого входят операции (стандартные функции) SQRT, SIN, COS (извлечение квадратного корня, синус и косинус от градусного аргумента). А вот стандартная функция тангенс отсутствует. Для описания функций служит специальная команда – OP (out put – вывести, считать результатом). Она прекращает выполнение процедуры и возвращает значение аргумента.

#### II. Команды управления системой

Управление системой Лого осуществляется в графическом экране. Соответствующие команды задаются в графическом (командном) окне.

**Команды выполнения**



*Рис.5. Меню Файл*

* Alt+R – выполняет выделенный текст или список инструкций.

**Команды сохранения и загрузки программы**

* Открой проект (Alt+N) - сохраняет на системном носителе текст записанной программы в файле <имя>.mw2 (рис.5).
* Alt+O – открывает существующий проект.
* Сохрани проект - считывает с системного носителя файл с именем <имя>.mw2 (рис.5).

**Команды печати созданных рисунков**

* Печатай проект - выводит созданный в Лого рисунок на принтер (рис.5);
* Печатай лист …. **–** выводит на принтер весь лист (рис.5);

**Диагностика ошибок**

Если при выполнении программы или команды возникла ошибка, то на экране появляется текст (в поле команд) - это диагностическое сообщение об ошибке, которое выводится на русском языке. Место ошибки в программе указывается после перехода в текстовый экран и нажатием клавиши <Enter>.

2.2. Запись основных алгоритмических конструкций в среде Лого Миры и их использование для решения вычислительных задач

### Понятие алгоритма

Понятие алгоритма такое же основополагающее для информатики, как и понятие информации. Именно поэтому важно в нем разобраться.

В математике для решения типовых задач используются определенные правила, описывающие последовательности действий. Например, правила сложения дробных чисел, решения квадратных уравнений и т. д. Обычно любые инструкции и правила представляют собой последовательность действий, которые необходимо выполнить в определенном порядке. Для решения задачи надо знать, что дано, что следует получить и какие действия и в каком порядке следует для этого выполнить. Предписание, определяющее порядок выполнения действий над данными с целью получения искомых результатов, и есть алгоритм.

*Алгоритм* **—** заранее заданное понятное и точное предписание возможному исполнителю совершить определенную последовательность действий для получения решения задачи за конечное число шагов.

Понятие алгоритма является не только одним из главных понятий информатики, но одним из главных понятий современной науки. Более того, с наступлением эры информатики алгоритмы становятся одним из важнейших факторов цивилизации [56].

### Исполнитель алгоритма

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

Исполнителя характеризуют:

1. среда;
2. элементарные действия;
3. система команд;
4. отказы.

*Среда* (или обстановка) — это "место обитания" исполнителя. Например, для исполнителя Черепашка 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 (в каких состояниях среды может быть выполнена команда) 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*.

*Отказы* исполнителя возникают, если команда вызывается при недопустимом для нее состоянии среды.

В информатике универсальным исполнителем алгоритмов является компьютер.

### Свойства алгоритма

Основные свойства алгоритмов:

*Понятность* — исполнитель алгоритма должен понимать, как его выполнять. Иными словами, имея алгоритм и произвольный вариант исходных данных, исполнитель должен знать, как надо действовать для выполнения этого алгоритма.

*Дискретность* (прерывность, раздельность) — алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов).

*Определенность* — каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

*Результативность* (или конечность) состоит в том, что за конечное число шагов алгоритм либо должен приводить к решению задачи, либо после конечного числа шагов останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов.

*Массовость* означает, что алгоритм решения задачи разрабатывается в общем, виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма. [Светозарова Г.И., Мельников А.А., Козловский А.В.]

### Способы записи алгоритмов

На практике наиболее распространены следующие формы представления алгоритмов:

1. словесная (запись на естественном языке);
2. графическая (изображения из графических символов);
3. программная (тексты на языках программирования).

#### *Словесный способ записи алгоритмов*

*Словесный способ* записи алгоритмов представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке.

Например. Записать алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел (алгоритм Эвклида). [диплом стр.21]

Алгоритм может быть следующим:

1. задать два числа;
2. если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;
3. определить большее из чисел;
4. заменить большее из чисел разностью большего и меньшего из чисел;
5. повторить алгоритм с шага 2.

Словесный способ не имеет широкого распространения, так как такие описания.

#### *Графический способ записи алгоритмов*

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным.

При *графическом представлении* алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.

Такое графическое представление называется схемой алгоритма или *блок-схемой.* В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде *блока*. Блоки соединяются *линиями переходов,* определяющими очередность выполнения действий. В таблице приведены наиболее часто употребляемые символы.

|  |  |
| --- | --- |
| ***Обозначение и пример заполнения*** | ***Пояснение*** |
|  | Вычислительное действие или последовательность действий |
|  | Проверка условий |
|  | Начало цикла |
|  | Вычисления по подпрограмме, стандартной подпрограмме |
|  | Ввод-вывод в общем виде |
|  | Начало, конец алгоритма, вход и выход в подпрограмму |
|  | Вывод результатов на печать |

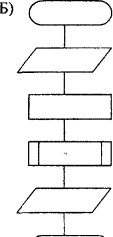
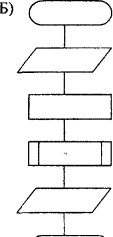
#### 

#### *Базовые алгоритмические структуры*

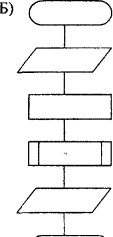
Алгоритмы можно представлять как некоторые структуры, состоящие из отдельных *базовых* (т.е. основных) *элементов*. Естественно, что при таком подходе к алгоритмам изучение основных принципов их конструирования должно начинаться с изучения этих базовых элементов.

Логическая структура любого алгоритма может быть представлена комбинацией трех *базовых структур*: следование, цикл, ветвление.

Характерной особенностью базовых структур является наличие в них одного входа и одного выхода.



*Рис.1*



**1. Базовая структура "следование".** Образуется последовательностью действий, следующих одно за другим, представлена на рис1.

начало

Ввод а, b

X=0-b/a

Вывод х

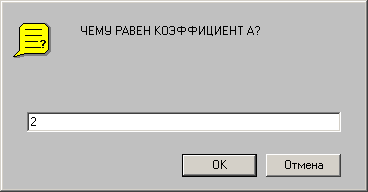
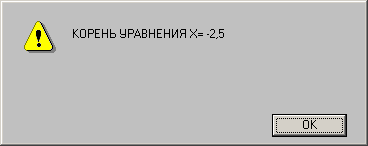
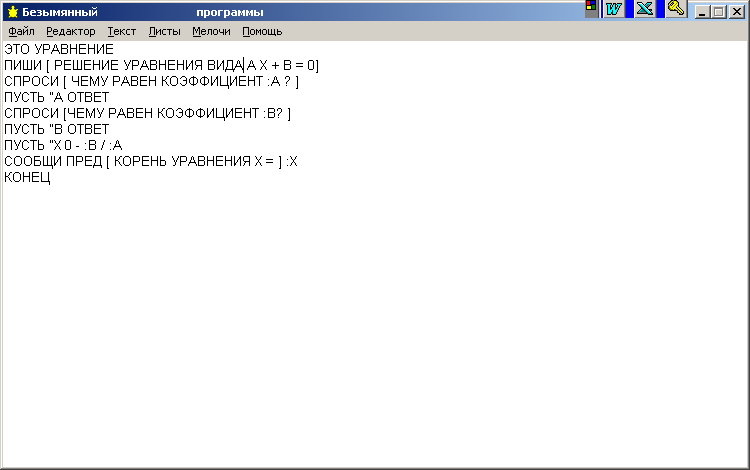
конец

*Рис.2*

Применение этой структуры рассмотрим на примере 1.

**Пример 1**. *Найти корень линейного уравнения вида ах+b=0.*

Для решения примера 1 составим блок-схему алгоритма (рис. 2.). Следуя последовательности блоков в алгоритме можно составить программу на языке Лого, которая может быть представлена, так как на рис. 3.



*Рис.3. Решение примера 1*

Для запуска программы в командном поле введем имя программы – УРАВНЕНИЕ и нажмем клавишу <Enter> .

В появившемся диалоговом окне введем значение для коэффициента А=2, во втором – В=5. В третьем диалоговом окне появится значение рассчитанного параметра Х=-2,5.

**Пример 2**. *Тело движется прямолинейно с ускорением а м/с2 и начальной скоростью υ м/с. Требуется определить, какой путь пройдет тело за Т секунд.* [Гейн А.Г., Житомирский В.Г., стр.15].

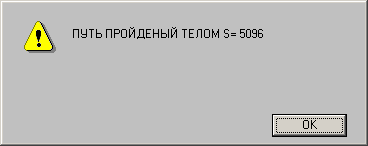
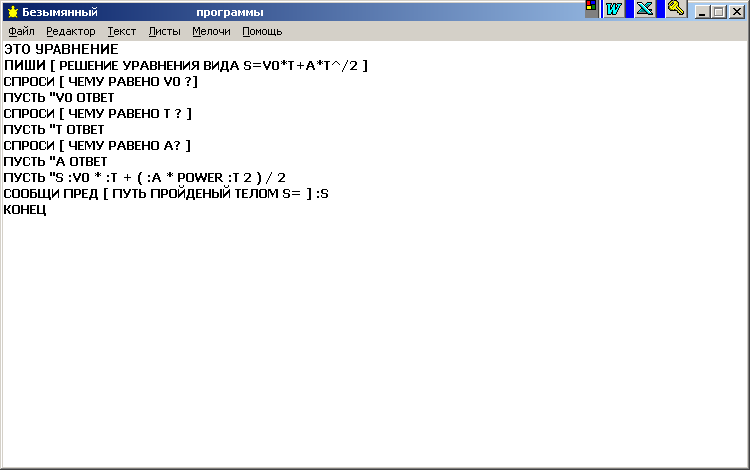
начало

Ввод v0, t, a

S=v0t+at^2/2

Вывод S

конец

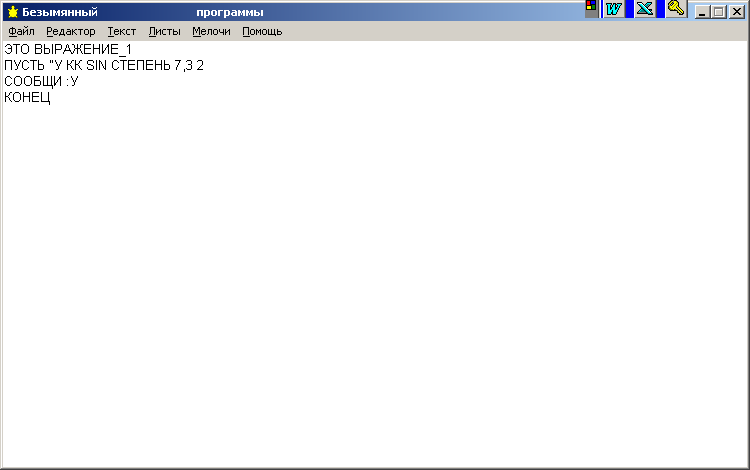


*Рис.4. Блок-схема решения примера 2, его реализация на языке Лого, вид окна с ответом в среде Лого;*

**Описание и использование функций**

В число Лого входят стандартные функции: SQRT - квадратный корень, SIN - синус, COS - косинус градусного аргумента, POWER - квадрат.

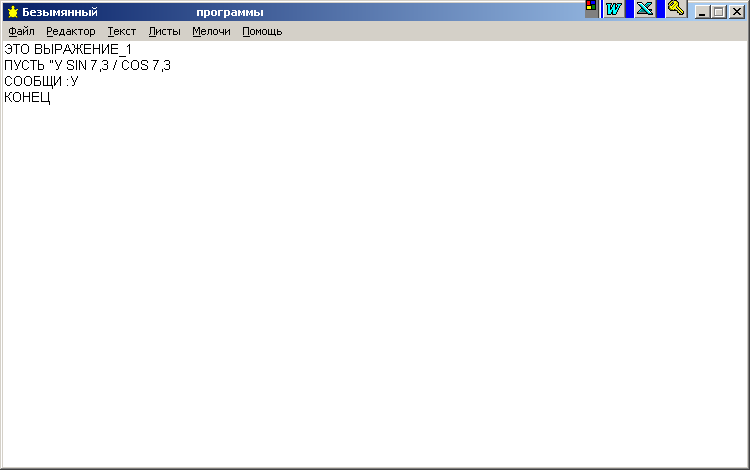
Вычислим пример:



*Рис.5 . Решение уравнения*



Для вычисления *тангенса* применяется соответствующая процедура.



*Рис. 6 . Решение уравнения*



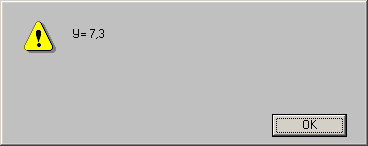
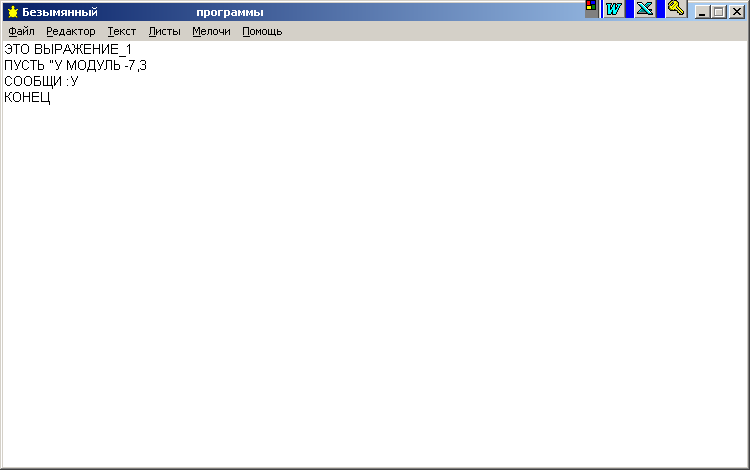
Произведем вычисления tg :



Для нахождения модуля числа решим уравнение вида: .



Использование функций может сделать программу короче и яснее. Кроме того, отпадает необходимость заводить вспомогательные переменные для хранения промежуточных результатов.



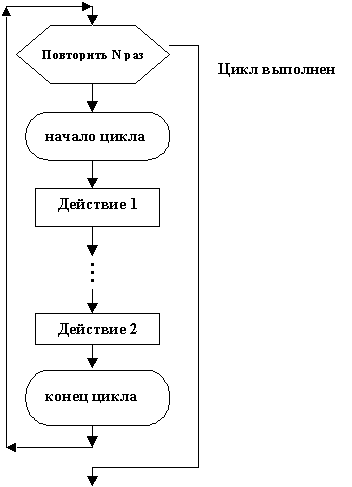
*Рис. 7 . Решение уравнения*



**2. Базовая структура "цикл"**

Обеспечивает *многократное выполнение некоторой совокупности действий,* которая называется *телом цикла.* Предписывает выполнять тело цикла до тех пор, пока выполняется условие, записанное после слова пока.

В языке Лого существует только один вид цикла, применяемый для повторения определенных действий. Базовую структуру «цикл» и команду REPEAT можно применять не только для перемещения Черепашки и рисования, но также и решения вычислительных задач, подразумевающих получения результата. Рассмотрим применение структуры Цикл на примере 3 (рис.8).



**Пример 3**. *Получить цифру 5 с помощью операции сложения и цифры 1.*

*Рис.8. Блок-схема решения примера 3, его реализация на языке Лого, вид окна с ответом в среде Лого;*

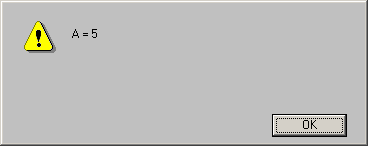
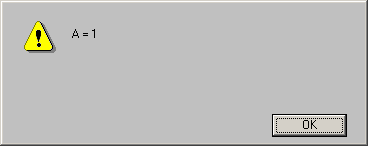
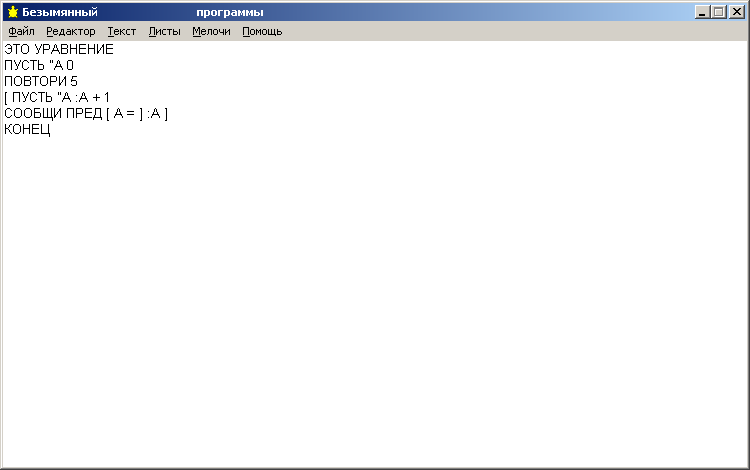
Повторить 5 раз

Начало цикла

а=а+1

Вывод а

Конец цикла



Можно заметить, что команды (пусть "a :a + 1, сообщи :a), повторяются 5 раз, следовательно применив структуру «Цикл» получим блок-схему применимо к рассматриваемому примеру 3 (рис.8). При вводе А=1, производится операция сложения А+1 и повторяется 5 раз, что в результате получаем А=5.

Принцип действия: прочитав, в наборе команд REPEAT Черепашка ищет указание числа повторений – указатель начала цикла «[». Запомнив его место, Черепашка будет выполнять команды, пока не встретит указатель конца цикла - «]». После этого Черепашка возвращается к началу цикла и проверяет, выполнила ли она цикл заданное число раз. Если нет, то она продолжает выполнять цикл, в противном случае (если она уже выполнила задуманное число раз) Черепашка начинает выполнять команды расположенные в квадратных скобках. Таким образом, тело цикла выполняется задуманное число раз.

Цикл с изменяющейся переменной для печатания числовых рядов могут быть построены по одной единой схеме. Сначала команды присваивания ПУСТЬ (MAKE) заводится новая переменная и ей присваивается начальное значение. Затем начинаются, повторения – многократно выполняется один и тот же набор действий. В этом наборе обязательно есть действие изменения значения переменной.

В такой программе команда присваивания ПУСТЬ (МАКЕ) встречается дважды. Первый раз при начальной установке переменной (а=0), второй раз – в цикле, при изменении значения переменной (а=а+1).

Первая команда присваивания выполняется один раз, вторая – многократно. При изменении переменной в цикле используется ее же предыдущее значение. Поэтому в команде ПУСТЬ "а :а + 1 (MAKE "а :а + 1) переменная упоминается дважды (присвоить переменной с именем “a” то значение, которое сейчас имеет переменная – “a”, увеличенное на единицу). Сначала упоминается имя, а затем – значение.

**Пример 4.** Вычислить уравнение: y=a+b четыре раза и получить результат.

Повторить 4 раза

Начало цикла

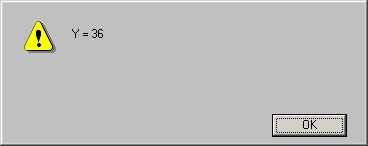
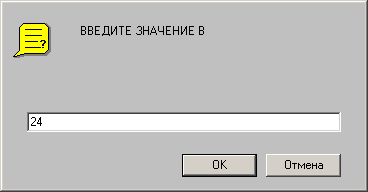
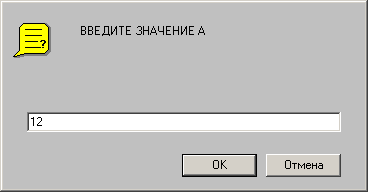
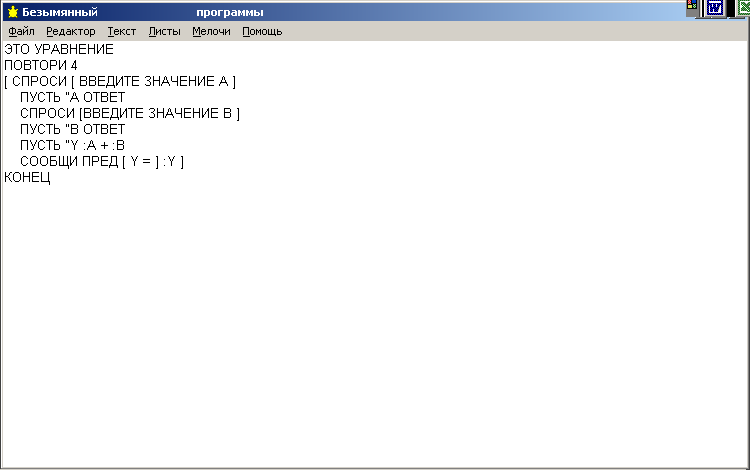
Ввод а, в

Y=a+b

Вывод Y

Конец цикла

*Рис.9. Блок-схема решения примера 4, его реализация на языке Лого, вид окна с ответом в среде Лого;*



Команда (пусть "y :a + b, сообщи :y), повторяются 4 раза, следовательно применив структуру «Цикл» получим блок-схему применимо к рассматриваемому примеру 4 (рис.9). При вводе А=12, а B=24 производится операция сложения А+B и повторяется 4 раза, что в результате получаем y=36.

**Пример 5.** *Вычислить 5 раз значения X=i3, Y=X/2 и вывести значения на экран.*

Сначала команды присваивания ПУСТЬ заводится новая переменная I и ей присваивается начальное значение. Затем начинаются, повторения – многократно выполняется один и тот же набор действий. В этом наборе обязательно есть действие изменения значения переменной. Команда (пусть "y :x / 2, сообщи :y, пусть “x :I \* :I \* :I ,сообщи :x ), повторяются 5 раз, следовательно применив структуру «Цикл» получим блок-схему применимо к рассматриваемому примеру 5 (рис.10). При выводе значений :y и :x производится операция возведения :x в степень, а :y=x/2 , что в результате получаем y=32, x=64.

***Пустой цикл***

Повторить 5 раза

Начало цикла

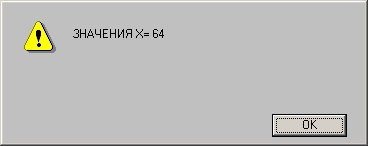
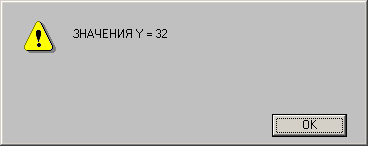
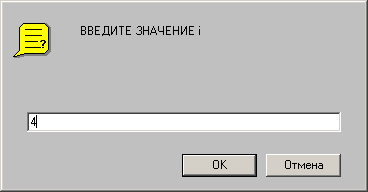
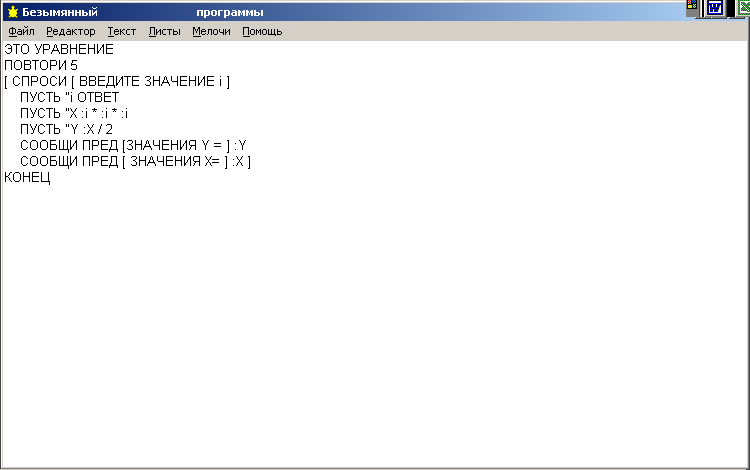
Ввод i

X=i3 , Y=X/2

Вывод Y

Конец цикла

*Рис.10. Блок-схема решения примера 5, его реализация на языке Лого, вид окна с ответом в среде Лого;*



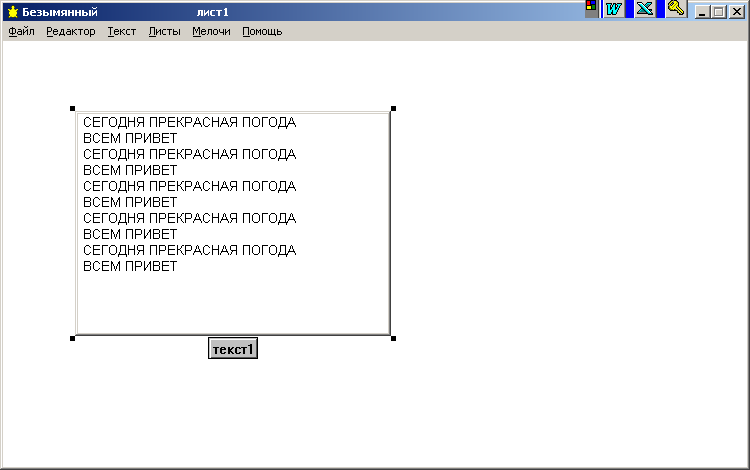
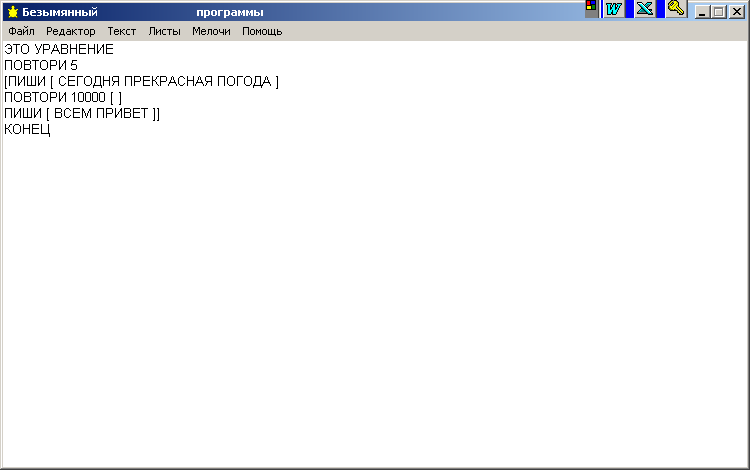
Это такой цикл, когда тело цикла отсутствует, т.е. после слова REPEAT в квадратных скобках ничего не стоит.

Например: REPEAT 1000 [ ]

Если ввести эту команду в ЭВМ, то Черепашка какое-то время не будет реагировать на команды, а будет 1000 раз выполнять НИЧЕГО. Иногда это полезно, потому что обеспечивает паузу между выполнением команд.

Это можно рассмотреть на примере 6.

**Пример 6.** Вывести две фразы “СЕГОДНЯ ПРЕКРАСНАЯ ПОГОДА” с паузой, затем фразу “ВСЕМ ПРИВЕТ”.



*Рис.11. Решения примера 6, его реализация на языке Лого, вид окна с ответом в среде Лого*

**3. Базовая структура "ветвление"**

Обеспечивает в зависимости от результата проверки условия (*да* или *нет*) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран. Структура *ветвление* существует в четырех основных вариантах: если—то; если—то—иначе;

|  |  |  |
| --- | --- | --- |
| 1. **если—то** | | |
|  | Обеспечивает ветвление (не полная форма).  Если условие истинно, то выполняются действия, указанные в списке. | **If**  **Если** |
| 2. **если—то—иначе** | | |
|  | Обеспечивает ветвление (полная форма).  Если условие истинно, то выполняется первое действие, если условие ложно – выполняется второе действие. | **Ifelse**  **Если\_иначе** |

**Пример 7.** *Если х>0, то вычислить значение y=sin(x).*

*Рис.12. Решения примера 7, его реализация на языке Лого.*

начало

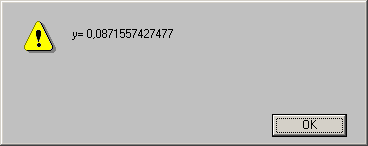
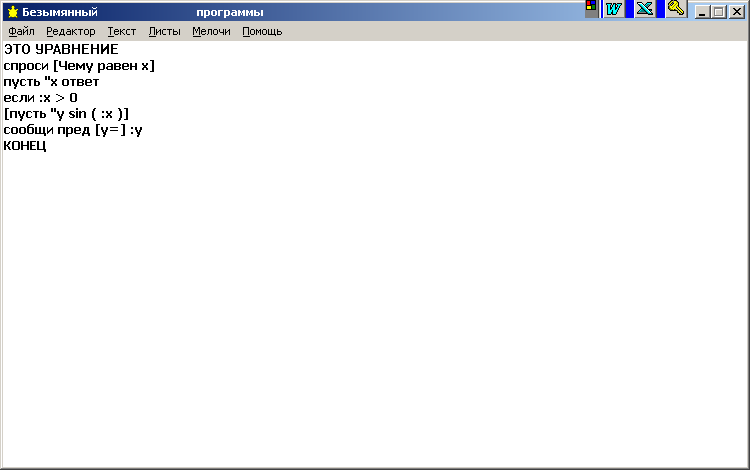
Ввод x

Y=sin(x)

Вывод y

конец

x>0



*Рис.11. Блок-схема решения примера 7, его реализация на языке Лого, вид окна с ответом в среде Лого;*

На примере 7 наглядно показано ветвление (“ЕСЛИ” не полная форма).

Если условие истинно, то выполняются действия, указанные в программе. Вычисления выполняются командной строкой [пусть :y sin (:x)],затем если условие выполнено, выводится “y”.

**Пример 8.** *Если a>b, то вычислить значение a=2\*a, если условие не выполняется, то вычислить b=2\*b*

начало

Ввод a, b

a=2\*a

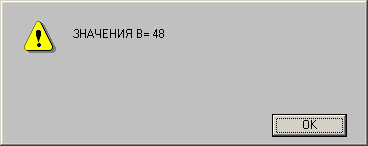
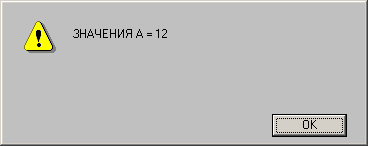
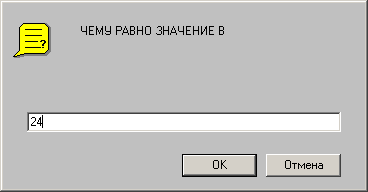
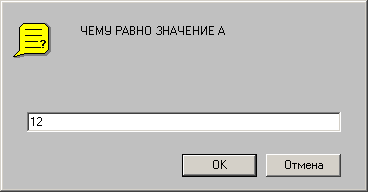
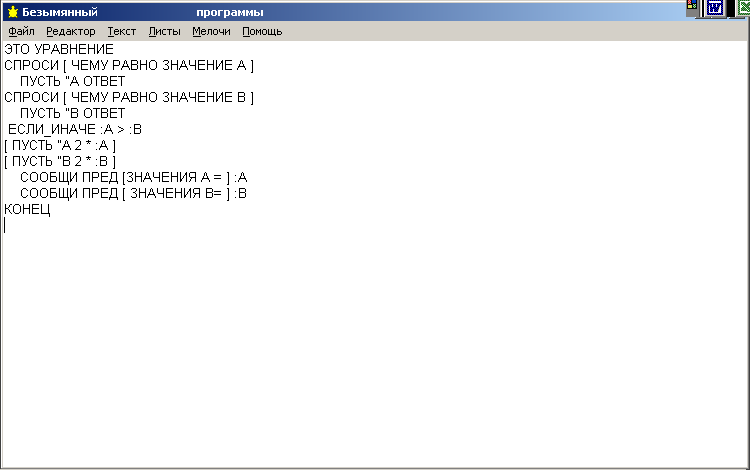
Вывод a, b

конец

a>b

b=2\*b

*Рис.13. Блок-схема решения примера 8, его реализация на языке Лого, вид окна с ответом в среде Лого;*

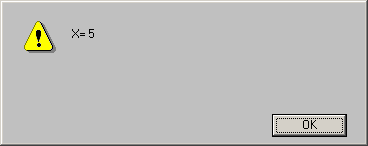
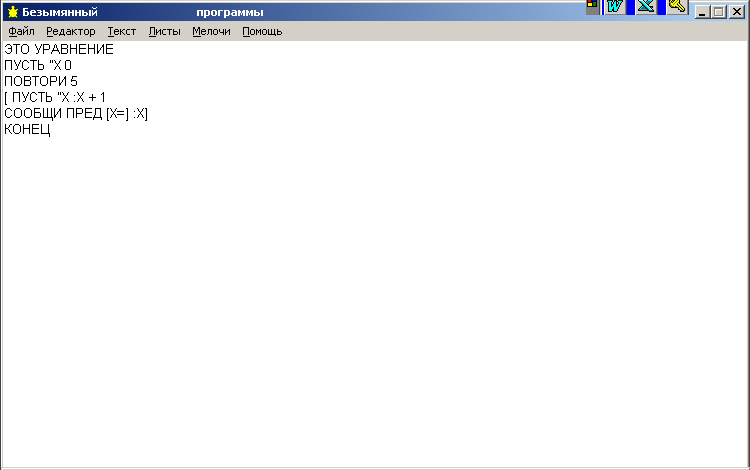


В примере 8 рассматривается полная форма ветвления. Если условие истинно, то выполняется первое действие, если условие ложно – выполняется второе действие.

**Процедуры с параметрами**

**1. Процедура без параметра**

Рассмотрим применение процедуры без параметра на уже решенном примере 3.

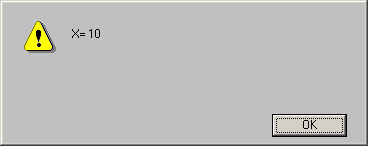
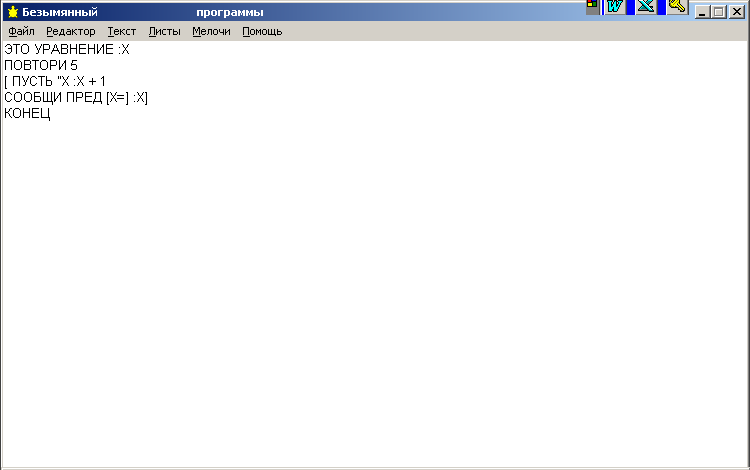


*Рис.13. Решение примера 3 с помощью языка Лого, вид окна с ответом в среде Лого.*

Процедура вызывается только с помощью имени, стоящего после ЭТО (ТО). Для этого используется процедура без параметра.

**2. Процедура с параметром**

Теперь в список команд Черепашки добавим переменную величину :х, которая позволит изменять размер начального значения :х.



*Рис.14. Решение примера 3 с помощью языка Лого, вид окна с ответом в среде Лого*

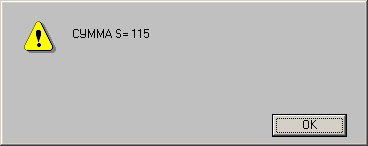
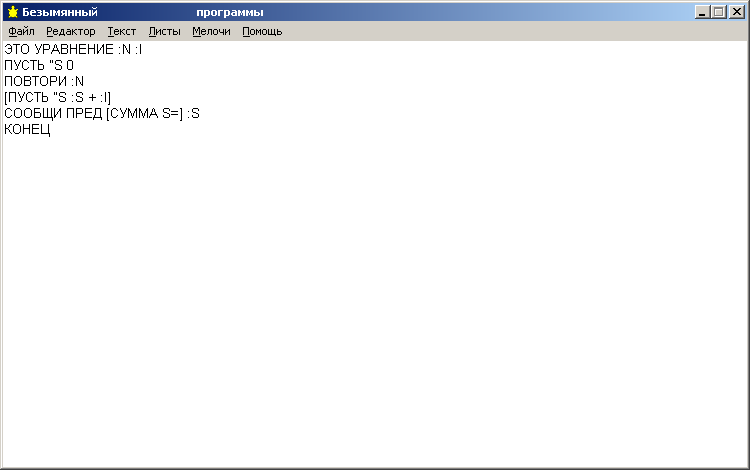
Такой параметр в заголовке процедуры *называется формальный параметр -* та же самая переменная "х". При вызове же процедуры нужно будет указать соответствующее значение - так называемый *фактический параметр.* Например, "уравнение 100". Следовательно, начальное значение переменной :х=100.

Рассмотрим чуть подробнее, что происходит при исполнении процедуры с параметром. Переменную можно представить в виде ящика, на котором наклеена табличка с именем. В этот ящик может поместиться одно значение. Компьютер в любое время может посмотреть, что там находится, а может и поменять хранящееся значение. Допустим, если дать команду уравнение 100.

В заголовке процедуры после ее имени стоит ":х". Поэтому число 100 компьютер положит в "ящик" с такой "табличкой". Теперь начинается выполнение самой процедуры. Черепашка начинает выполнять цикл. Встретившись с командой "пусть “х :х + 1", она смотрит, какое значение "лежит в ящике", и подставляет его в эту команду. Само значение переменной при этом не меняется.

Процедура может иметь не один параметр, а несколько. Рассмотрим пример 9.

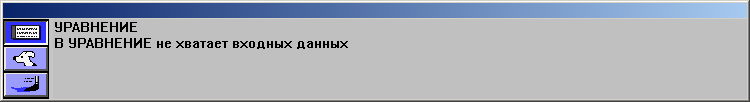
**Пример 9.** *Вычислить*



*Рис.15. Решение примера 9 с помощью языка Лого, вид окна с ответом в среде Лого*

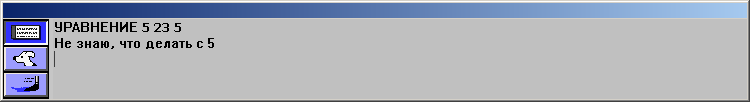


В этом случае при запуске процедуры количество фактических параметров и их порядок должны соответствовать формальным. Если их будет меньше, Лого выдаст сообщение:



Если больше –

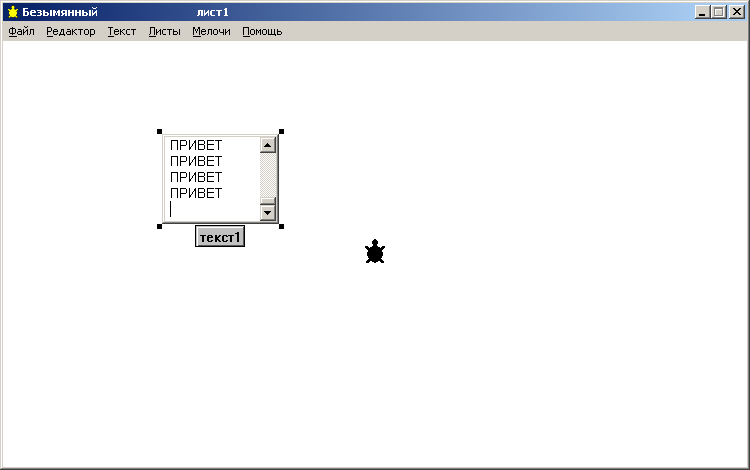
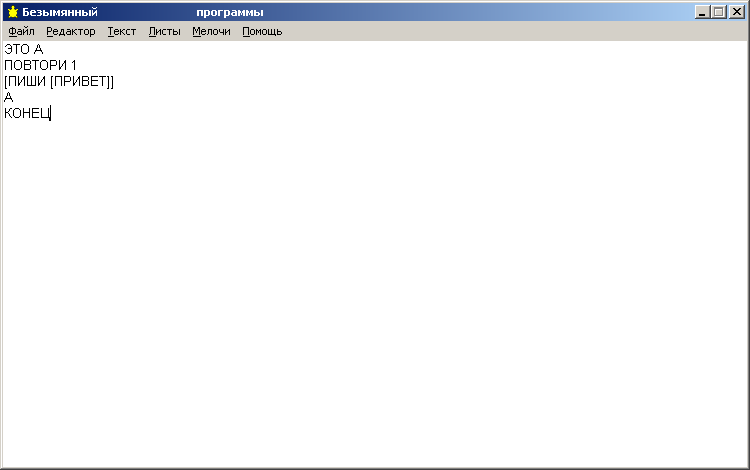
**Рекурсия**



В Лого все процедуры равноправны. Допускается, что одна процедура может вызывать вторую, вторая – третью и так далее. Кроме того, в Лого допускается, что процедура может вызывать процедуру со своим собственным именем. Такой вызов процедурой самой себя называется рекурсией.

**а) Процедура, обращающаяся сама к себе**

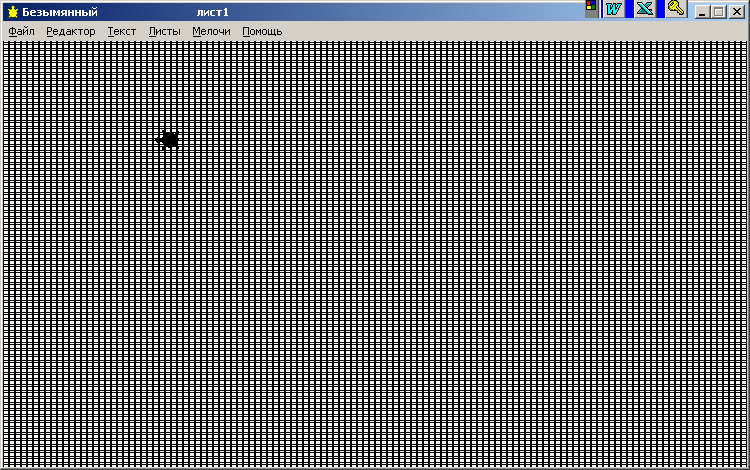
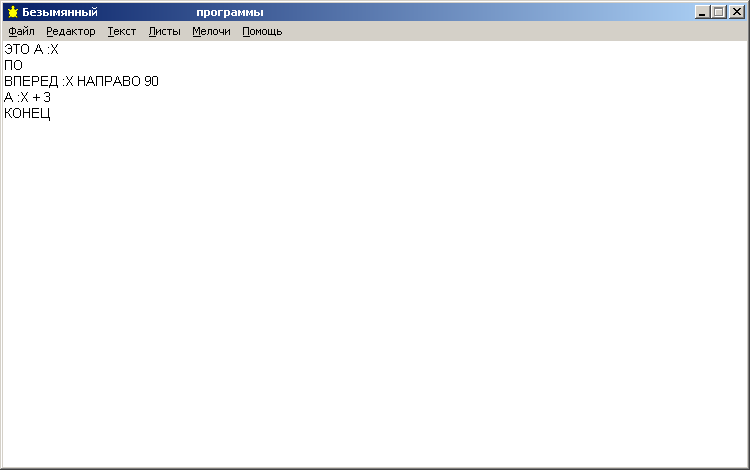
**Пример 10.** *Составить процедуру А, бесконечно печатающую слово «ПРИВЕТ»:*



*Рис.16. Решение примера с помощью языка Лого, вид окна с ответом в среде Лого*

Процедура А печатает в текстовом окне слово «ПРИВЕТ» и вызывает процедуру А, которая печатает в текстовом окне слово «ПРИВЕТ» и вызывает процедуру А и т.д. Слово «ПРИВЕТ» будет печататься до тех пор, пока пользователь не прервет программу. Такого типа программы называют программами *рекурсивного вызова самой себя или бесконечным циклом.* Более интересный результат получится, если при рекурсивном вызове изменить значение параметра.

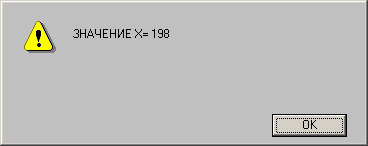
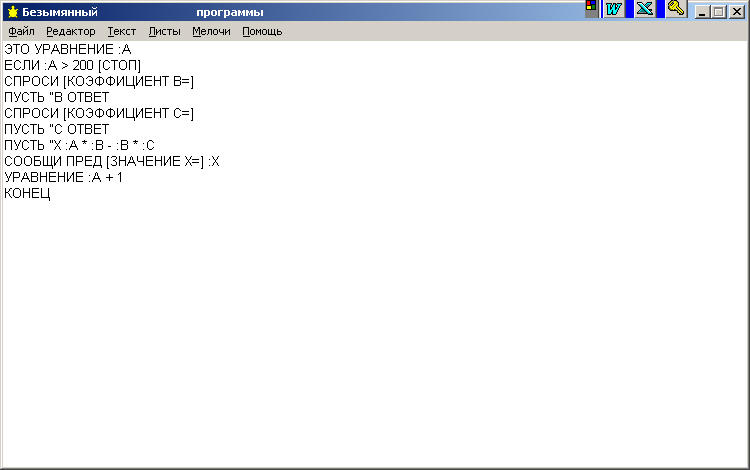
**Пример 11.** *Используя уравнение х+3 рассчитать и нарисовать траекторию движения Черепашки:*



**б) Управляемая рекурсия**

Для остановки программы в нужный момент можно применить условие и команду **STOP** (СТОП).

**Пример 12.** *Ввести коэффициенты А, В, С. Если А<200, вычислить Х=А.В-В.С. В противном случае остановить программу*.



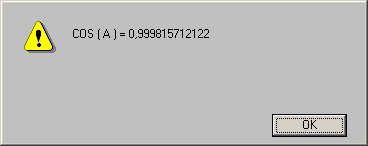
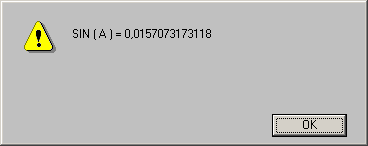
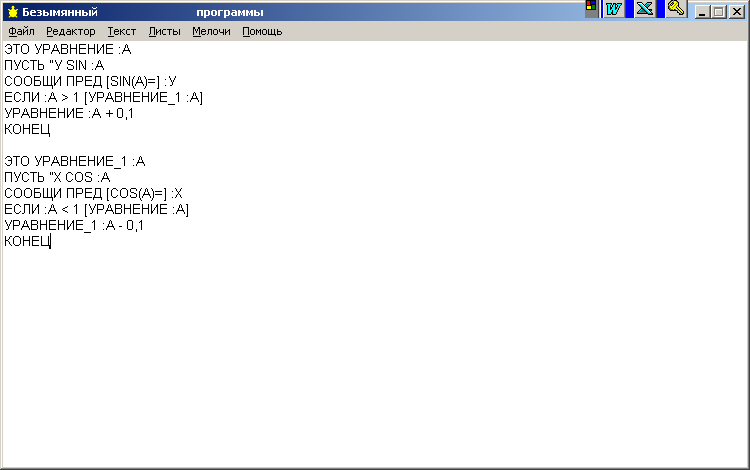
*Рис. 18. Решение примера 12*

Если при очередном вызове параметр А оказался больше 200, то программа остановится. В противном случае последует запрос и ввод коэффициентов А и В, расчет значения Х и новый вызов с увеличенным параметром.

**в) Косвенная рекурсия**

Если процедура вызывает сама себя не непосредственно, а через другую процедуру, то это называется косвенной рекурсией.

**Пример 13.** *Вычислить значения у=sin(a) и х=cos(a), установив между ними взаимосвязь и не превышать значения 1.*



*Рис.19. Решение примера 13*

Эти две процедуры (вычисления sin(a) b cos(a)) работают так: первая вызывает вторую, а вторая – первую. В результате получается один из примеров косвенной рекурсии.**Решение задачи с использованием основных базовых структур**

**и операторов языка Лого на примере экологической задачи**

**Пример 14.** *Расположенный на берегу реки металлургический завод осуществил сброс вод, в результате чего концентрация вредных веществ в реке резко увеличилось. С течением времени эта концентрация, естественно, уменьшается. Требуется сообщить, каков будет уровень загрязнения реки через N суток, до тех пор, пока концентрация не станет нормой.*

Сначала построим математическую модель изучаемого явления. Специалистам-экологам установлена следующая приближенная закономерность: в каждом конкретном случае можно указать такое число K>1, что концентрация примесей C уменьшается в К раз за сутки. При этом коэффициент К зависит от района, где протекает река, типа примесей и т. п. Значение К можно узнать из соответствующего справочника. Эту закономерность примем в качестве исходного предположения для рассматриваемой математической модели.

начало

конец

Начало цикла

Конец цикла

Ввод

C, D, K, Z, H

N=0

Повторять Z раз

N=N+H

C=C/K^H

Вывод

C, N

Вывод

N

C > D

C < D

Исходными данными будут начальная концентрация С вредных веществ в реке, предельно допустимая концентрация D и коэффициент К. Результат – последовательностью значений концентрации вредных веществ через сутки, двое суток и т. д. Связь между исходными данными и результатом дается следующими соотношениями:



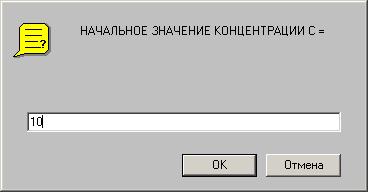
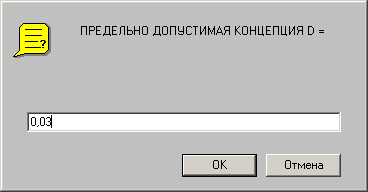
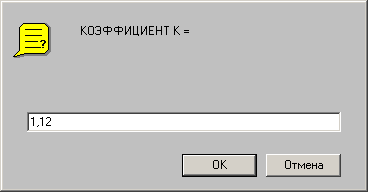
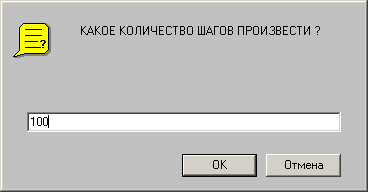
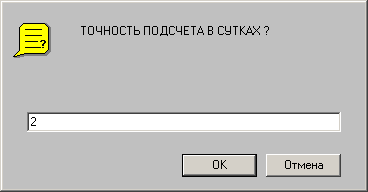
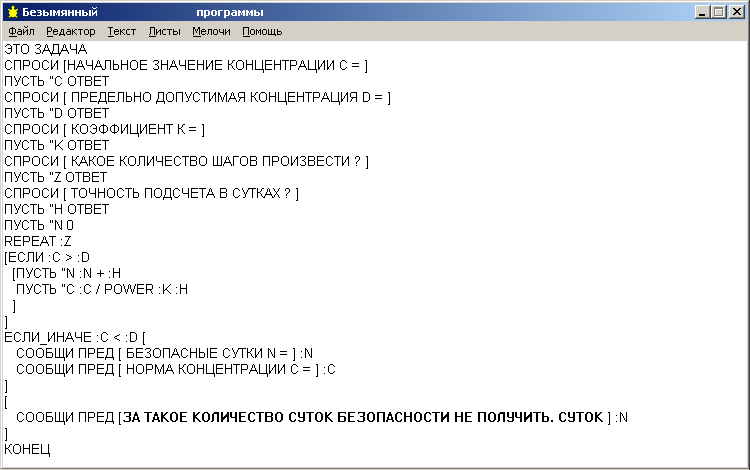
где - концентрация вредных веществ через N суток после сброса.



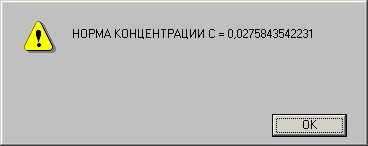
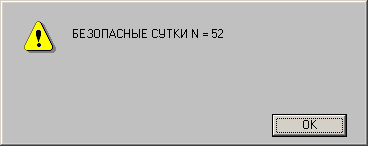
Руководствуясь этой математической моделью, составим блок-схему, где C - начальное значение концентрации,

D – предельно допустимая концентрация, K – коэффициент, Z – ограничения количества шагов для расчета, H – точность подсчета в сутках, N - количество суток.

Составим программу на языке Лого и произведем расчеты для свинца при C=10 мг/л, D=0.03 мг/л, K=1.12. [Гейн А.Г. стр.71]



Полученные расчеты дают возможность увидеть нормальную концентрацию свинца С= 0.027 в воде на 52 сутки.



2.3. Методические рекомендации

### Изучение базовых алгоритмических конструкций в языке Лого в учебном процессе школы

Данный курс предлагает введение в алгоритмизацию с помощью среды Лого Миры, основу которой составляет язык Лого. Во время обучения должны быть решены три главные задачи: приобретение детьми навыков обработки различных видов информации на ПК, усвоение основных алгоритмических конструкций, освоение работы с объектами языка Лого.

На изучение темы в учебном плане школы отводится достаточное количество академических часов, в течение которых рассматриваются ниже перечисленные понятия алгоритмизации и реализация соответствующих данным понятиям вычислительных задач:

1. Базовая структура «следование» в Лого Мирах
2. Базовая структура «цикл» в Лого Мирах
3. Базовая структура «ветвление» в Лого Мирах

На изучение каждого из пунктов отводится соответствующее количество часов.

После изучения раздела «Алгоритмизации» преподаватель получает хорошую возможность показать учащимся, зачем изучались принципы разработки алгоритмов, написание и отладка программ. Все эти принципы учащиеся могут применить уже сегодня, в своей главной деятельности – школьной учебе. Связав свою работу с основными школьными предметами, учитель информатики повышает свой внутришкольный статус. Включается в общее дело – решение базовых задач школы.

Кроме того, на уроках информатики по этой теме учащиеся знакомятся с учебной исследовательской работой. При этом они должны не только написать программу, но и научиться использовать ее для небольшого исследования: провести эксперименты, собрать данные (графические и числовые), проанализировать полученные результаты, сформулировать выводы.

Для успешного усвоения материала учащиеся должны вести тетрадь, куда помещают опорные конспекты, упражнения, помогающие ребенку освоить учебный материал. Тетрадь позволяет учителю организовать домашнюю работу учащихся, нацеленную не только на отработку знаний, умений и навыков, но и развитие познавательной активности, мышления, самостоятельности учащихся, организовать проверку усвоения новых знаний.

Данный курс не только прививает учащимся элементы информационной культуры, под которой понимается умение целенаправленно работать с информацией на ПК, но и помогает детям расширить свой кругозор, развивать логическое мышление, творческий и познавательный потенциал, связать с другими учебными предметами: математикой, биологией, искусством, языками, т.д.

В рамках поурочной системы компьютерные оболочки позволяют использовать Лого Миры как превосходное средство для организации развивающего, проблемного обучения, когда у детей вместе с количеством знаний возрастает самостоятельность. Для этого в каждом пункте имеется цепь заданий с вариациями, позволяющими детям проявить творческие возможности. Разделы имеют требования:

* + 1. Новая информация прелагается малыми порциями и появляется тогда, когда становится необходима для выполнения конкретного задания.
    2. Каждое задание опирается на изученный материал и приобретенные навыки, но обязательно содержит новую информацию или новые действия.
    3. Заданий в разделе больше, чем это необходимо на уроке, с тем, чтобы учащийся мог выбрать интересное ему задание и работать в своем индивидуальном темпе.
    4. Каждая часть раздела рассчитана на 15-20 мин работы на ПК. Непрерывная длительность занятия непосредственно на ПК не превышает санитарных норм, определенных документом «Гигиенические требования к видеодиспленным терминалам, персональным ЭВМ и организация работы».

При проведении занятий применяются различные формы обучения. Обучение учащихся должно иметь циклический характер. Раскрытие одного раздела может быть распределено по всему курсу обучения и идти поэтапно по мере подготовки учащихся.

### Примерный план разделов курса «Алгоритмизация и решение вычислительных задач»

* + - 1. Алгоритмы. Исполнитель алгоритмов. Система команд исполнителя. Свойства алгоритмов. Формирование исполнение алгоритмов. Способы записи алгоритмов.
      2. Линейный алгоритм.

Лого Миры. Три технологических способа программирования: в поле команд, в диалоговом окне черепашки, в диалоговом окне управляющей кнопки. Команды управления черепашкой: вперед, назад, налево, направо, жди, домой, нов\_форма, плавно.

Практическая работа.

* + - 1. Циклический алгоритм. Лого Миры. Свойства класса Черепашки. Команды повтори, всегда, отмени, возьми\_лист.

Практическая работа.

* + - 1. Величины. Переменные величины. Характеристики переменных: тип, имя, значение. Лого Миры. Команды пусть, спроси, покажи.

Практическая работа.

* + - 1. Библиотека алгоритмов. Встроенные функции. Вложенные циклы. Датчик случайных чисел. Лого Миры. Координатная плоскость. Команды (нов\_место, нов\_х, нов\_у, нов\_размер). Функции х\_коорр, у\_коор, место, случайный.

Практическая работа.

* + - 1. Вспомогательные алгоритмы. Процедура с параметрами. Лого Миры. Команды по, пп, пс, нрп.

Практическая работа.

* + - 1. Ветвления. Основные понятия формальной логики. Простые логические выражения. Лого Миры. Команды если, если\_иначе.

Практическая работа.

* + - 1. Операции формальной логики. Таблица истинности. Составленные логические выражения. Лого Миры. Логические операции или, и, не.

Практическая работа.

* + - 1. Рекурсия. Виды рекурсии. Лого Миры. Принципы построения рекурсии.

Практическая работа.

Заключение

В завершении хотелось бы отметить, что задачи, поставленные в выпускной работе, были решены, а именно:

* освоена среда Лого Миры;
* определен банк традиционно решаемых вычислительных задач при изучении алгоритмизации;
* разработаны алгоритмы и программы для решения вычислительных задач на языке Лого.
* проведен анализ решения вычислительных задач на языке Лого

Таким образом, цель данной выпускной работы достигнута.

В дальнейшем банк задач будет пополняться и данную методику решения вычислительных задач планируется использовать в средних общеобразовательных школах.

Данная работа будет полезна как студентам при подготовке к теоретическим, так и к практическим занятием по изучению среды Лого Миры и программированию на языке Лого, так и преподавателям, использующим пакет Лого Миры в своей работе. Так же поможет преподавателям разнообразить свои уроки, сделать их более эффективными в развитии познавательных процессов логического мышления, внимания, воображения, памяти, как младшего, так и старшего возраста.

**Аннотация**

Данная выпускная работа на тему: “Организация изучения основных алгоритмических конструкций в среде Лого Миры”.

Объект исследования – процесс обучения базовому курсу информатики.

Предмет исследования – методика преподавания алгоритмизации на базе языка Лого.

Цель работы – определение роли и месте вычислительных задач, решаемых в среде Лого Миры при изучении алгоритмизации.

Основные задачи исследования:

* Освоить среду Лого Миры
* Определить банк традиционно решаемых вычислительных задач при изучении алгоритмизации.
* Разработка программ для решения вычислительных задач на языке Лого.
* Анализ решение вычислительных задач на языке Лого.

Таким образом, цель данной выпускной работы достигнута.

Выпускная работа будет полезна как студентам при подготовке к теоретическим, так и к практическим занятием по изучению среды Лого Миры и программированию на языке Лого, так и преподавателям, использующим пакет Лого Миры в своей работе. Так же поможет преподавателям разнообразить свои уроки, сделать их более эффективными в развитии познавательных процессов логического мышления, внимания, воображения, памяти, как младшего, так и старшего возраста.