

## Введение

Стремительная эволюция информатики и информационно-коммуникационных технологий (ИКТ) в последние 30 лет является определяющей. Информационные технологии в образовательной сфере предоставляют широкие возможности для исследовательской, а так же творческой работы в таких областях, как компьютерная графика, Web-дизайн, обработка видео, 3D-технологии, но прежде всего информационные технологии открывают поистине необъятные возможности в преподавании школьного курса программирования. Прогресс интеллектуального творчества, выявление талантливых и интеллектуально одаренных учеников, умения работать с научной литературой и информацией - все это составляет основные задачи развития исследовательской деятельности учеников старшего звена школы. Поскольку в большей степени именно для старшеклассников характерны развитые формы теоретического мышления. Творческая, основанная на исследовательском подходе работа школьников ориентирует их на получение высшего образования в сфере информационных технологий с последующим трудоустройством в этом секторе экономики. ИКТ всё больше находят свое применение в различных предметных областях, помогая лучшему усвоению как отдельных тем, так и изучаемых дисциплин в целом. Овладение навыками ИКТ и программирования помогает учащимся увереннее смотреть в будущее, применять свои знания в практической деятельности, творчески решать различные задания. На сегодняшний день, с учётом оснащённости компьютерной техникой школ, целесообразность применения компьютерных технологий в среднем и старшем звеньях школы становится всё более обоснованной. На сегодняшний день без, математических расчетов, логических схем и структур не обходится ни одно серьезное исследование или разработка конструкций. Изучение и практическое применение вышеизложенных подходов требует более глубоких знаний в предметной области "Информатика и информационные технологии", чем содержание базового курса информатики. Очевидно, таким образом, что возникает необходимость в организации профильного обучения отдельным разделам информатики и информационных технологий в школе или в иной образовательной системе. Реализовать решение проблемы более глубокого изучения разделов информатики, связанных с программированием можно путём введения элективных курсов по выбору, например, "Олимпиадные задачи по программированию", "Web-дизайн и программирование", "Программирование графики и анимации", "Программирование во Flash". При этом необходимо изменить общий подход к преподаванию программирования в школе. С

развитием мощности персональных компьютеров (ПК) наблюдается явная тенденция перехода обучения от алгоритмических языков к объектно-ориентированным языкам программирования. Это, безусловно, объективный фактор, однако не следует забывать о том, что основой для формирования алгоритмического мышления, понимания и правильного конструирования алгоритмов, прежде всего, являются именно алгоритмические языки. Полагаю, что оптимальной схемой реализующей сбалансированный переход от консольного программирования к визуальному, является: начало- младшее звено КуМир, среднее звено - Pascal, старшее звено - Lazarus. Объектно-ориентированную среду разработки Lazarus можно с успехом использовать для реализации различных моделей физических процессов, а так же создания целого комплекса обучающих, контролирующих, демонстрационных программ. Следующим шагом, на пути изучения основ программирования, по крайней мере для наиболее заинтересованных учащихся, в индивидуальном порядке можно вести обучение языку C++, который в будущем пригодится для изучения курса “Программирование во Flash” или для создания исследовательских проектов, связанных с реализацией аппаратной части. Приоритет в области обучения программированию, несомненно, обеспечит формирование научного мировоззрения учащихся, реализацию целей и задач общего образования по информатике в его базовой и профильно-ориентированной частях, реализацию межпредметных связей информатики и других школьных дисциплин, профессиональной ориентации учащихся. Теперь, если мы обратим взоры на реальное положение дел в области обучения программированию в школе, то приходится признать, что здесь накопилось очень много проблем, нерешённость которых на протяжении последних лет неуклонно снижает обученность программированию среди школьников, о чём красноречиво говорит вовлечённость в соревнования по программированию в рамках всероссийских олимпиад, а также результаты показываемые участниками. Одной из многочисленных составляющих означенной выше проблемы, является крайне низкая обеспеченность программами обучающих программированию. Таким образом, очевидна актуальность темы данной квалификационной работы, в рамках которой будет представлен подход к созданию таких программных средств, а так же конкретный комплекс обучающих программ. Созданная в рамках темы данной квалификационной работы программа должна отличаться от аналогичных систем, разнообразием средств обучения, обеспечивающих теоретическую и практическую работу, удобным и простым интерфейсом, возможностью подключения к интерфейсу программы самостоятельно

разработанных информационных ресурсов. Исходя из вышесказанного, а также большой актуальностью затронутой в теме проблемы предстоит решить следующие задачи:

1. Исследовать принципы проектирования автоматизированных систем обучения, методологию построения обучающих систем.
2. Исследовать структуру знаний в автоматизированной системе знаний (АОС) и функциональную модель процесса обучения.
3. Разработать АОС.
4. Произвести апробацию программного продукта в учебном классе.
5. Оценить эффективность применения.

Логика исследования обусловила структуру квалификационной работы: введение 3 стр. страницы., разделы 1-4 25 страниц, заключение 1 страница, список использованных источников из 10 наименований, двух приложений (текст программы, DVD с программой). Общий объём 30 страниц.

## 1. Анализ принципа проектирования обучающих программ

Эффективные компьютерные дидактические программы (электронные учебники, компьютерные задачки, учебные пособия, гипертекстовые информационно-справочные системы - архивы, тестирующие и моделирующие программы-тренажеры и т.д.) разрабатываются на основе мультимедиа-технологий. Мультимедиа-системы позволяют сделать подачу дидактического материала максимально удобной и наглядной, что стимулирует интерес к обучению и позволяет устранить пробелы в знаниях. Кроме того, подобные системы могут и должны снабжаться эффективными средствами оценки и контроля процесса усвоения знаний и приобретения навыков. Ключевую роль в создании обучающих компьютерных программ играет методическое обеспечение разработок. Обучающие программы призваны выполнять все основные этапы обучения - от изложения учебного материала до контроля знаний. При этом весь обязательный учебный материал переводится в яркую, увлекательную, с разумной долей игрового подхода, мультимедийную форму с широким использованием графики, анимации, в том числе интерактивной, звуковых эффектов и голосового сопровождения, включением видеофрагментов, и т.п. Подобный радикальный подход весьма трудоемок, но только "тотально мультимедийная" методически замкнутая обучающая программа может преодолеть существующую пропасть, которая отделяет ожидания потребителей от возможностей компьютерных технологий в обучении, преодолеть скепсис по отношению к последним и стать реальным шагом вперед в развитии системы образования, в том числе и дистанционного. Обучающая компьютерная программа - это не только комплексная, но и целостная дидактическая, методическая и интерактивная программная система, которая позволяет изложить сложные моменты учебного материала с использованием богатого арсенала различных форм представления информации, а также давать представление о методах научного исследования с помощью имитации последнего средствами мультимедиа. Не последнее место здесь играют видеоуроки, поскольку ученик, решивший самостоятельно изучить некий раздел максимально мотивирован, нужно лишь доступно преподнести теоретический материал и предложить применить полученные знания на практике. При этом повышается доступность обучения за счет более понятного, яркого и наглядного представления материала. Дидактические аспекты, касающиеся наиболее общих закономерностей обучения, и методические аспекты, определяемые спецификой

обучения программированию, тесно взаимосвязаны между собой и с вопросами программной реализации электронного учебника. Программно-методический комплекс, призван обеспечить возможность самостоятельно или с помощью преподавателя освоение учебного курса или его раздела. Всё большая доступность интернет позволяет делать обучающие программы интерактивными. Следует выделить три составляющих эффективной обучающей программы: презентационная часть, в которой излагается основная информационная часть курса в виде текста или видеуроков, практические задания в форме тестов или тренажеров, с помощью которых закрепляются полученные знания, творческий задания в виде задач на программирование технических систем, математических закономерностей, позволяющие проводить объективную оценку знаний ученика. Обучающая программа должна соединять в себе свойства обычного учебника, справочника, задачника и лабораторного практикума - это система комплексного назначения, обеспечивающая непрерывность и полноту дидактического цикла процесса обучения: предоставляющая теоретический материал, обеспечивающая тренировочную учебную деятельность и контроль уровня знаний, а также информационно-поисковую деятельность. Электронный учебник должен обеспечивать выполнение всех основных функций, включая предъявление теоретического материала, организацию применения первично полученных знаний (выполнение тренировочных заданий), контроль уровня усвоения (обратная связь), задание ориентированные на самообразование. Реализация всех звеньев дидактического цикла процесса обучения посредством единой компьютерной программы существенно упрощает организацию учебного процесса, сокращает затраты времени учащегося на обучение и автоматически обеспечит целостность дидактического цикла в пределах одного сеанса работы с обучающей программой. Учебный процесс происходит на принципиально новом уровне, так как обучающая программа по определению дает возможность работать в наиболее приемлемом для обучаемого темпе, обеспечивает возможность многократных повторений и диалога между обучаемым и обучающим, в данном случае компьютером. Психологи [1] говорят, что каждый из нас обучается по-разному - некоторые лучше обучаются на слух, другие являются зрительными или тактильными обучающимися. Поэтому необходимо в максимальной степени использовать тот стиль обучения, который является предпочтительным для конкретного учащегося. Привлечение всех органов чувств [2] ведет к исключительному росту степени усвоения материала по сравнению с традиционными методами. Обучение с использованием аудиовизуальных средств комплексного предъявления информации является наиболее интенсивной формой обучения; учебный материал, дидактически

подготовленный специалистами, ориентируется на индивидуальные способности учащихся. Индивидуальная диалоговая коммуникация с помощью видео-графических, текстовых и музыкально-речевых вставок настолько интенсивна, что максимально облегчает процесс обучения. Решение проблемы соединения потоков информации разной модальности (звук, текст, графика, видео) делает компьютер универсальным обучающим и информационным инструментом по практически любой отрасли знания и человеческой деятельности.

### 1.1 Принципы построения системы.

Любая программа есть набор алгоритмов, которые взаимодействуя [3] между собой решают поставленную задачу. При этом программа будет являться программной системой, если она представляет собой совокупность взаимосвязанных компонентов, каждый из которых выполняет вполне определенные функции. В общем случае любая обучающая программа может считаться программной системой, так как в ней обязательно присутствует компонента интерфейса пользователя, и компонента, реализующая предлагаемую методику. Таким образом, её можно рассматривать как автоматизированную обучающую систему. Каждая АОС имеет определенную структуру на основе группы элементов с указанием связей между ними и дающее представление о системе в целом. Поэтому структура системы может быть охарактеризована по имеющимся в ней типам связей. Исходя из соображений, изложенных выше разработанная обучающая программа построена по принципу системы с обратной связью. Схематически указанный принцип изображён на рисунке 1:

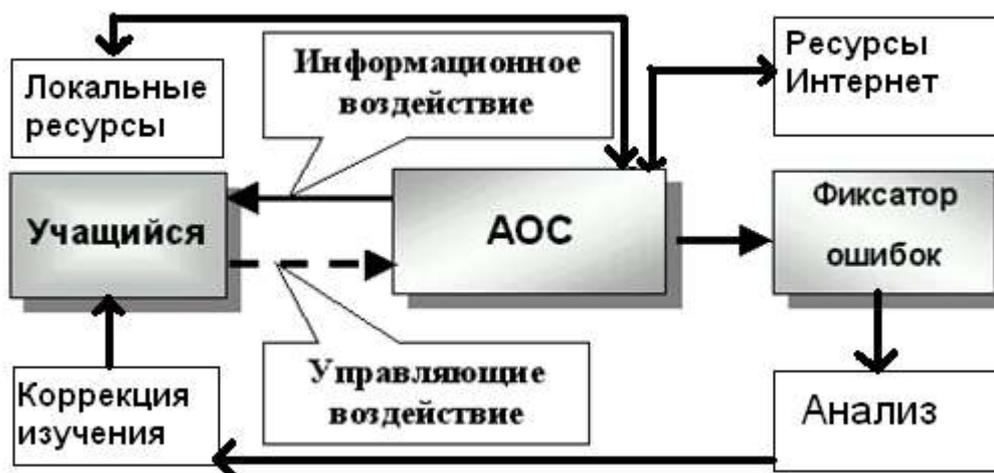


Рисунок 1. Структура автоматизированной обучающей системы

АОС взаимодействует с локальными ресурсами и ресурсами интернет, формируя информационное воздействие на обучаемого. В режиме тренинга от учащегося ожидается отклик в виде ответа (управляющего воздействия) на поставленный вопрос. Ответ фиксируется в блоке фиксатора ошибок. По результатам опроса [4] выставляется определенный балл, который служит критерием для результирующей оценки по степени усвоения учащимся требуемого учебного материала и рекомендаций обучаемому повторно рассмотреть тот или иной теоретический раздел.

## 1.2 Методология проектирования системы

В методологическом плане выделяют два направления в разработке и использование компьютерных средств поддержки обучения. [5] Первое направление опирается в своей основе на идеи программированного обучения. В его рамках разрабатываются и эксплуатируются автоматизированные обучающие системы (АОС) по различным учебным дисциплинам. Ядром АОС являются так называемые авторские системы, позволяющие преподавателю-разработчику вводить свой учебный материал в базу данных и программировать с помощью специальных авторских языков или других средств алгоритмы его изучения. Второе это отдельные программы, пакеты программ, элементы автоматизированных систем (АСУ, САПР, АСНИ, АСУП и др.), предназначенные для автоматизации трудоемких расчетов, оптимизации, исследования свойств объектов и процессов на математических моделях и т.п. АОС разработанная в рамках данной квалификационной работы относится к первому методологическому типу. Учитель может

подключать к интерфейсу АОС собственные материалы двух типов: видеоуроки в формате avi, текстовые файлы, содержащие информацию о заданиях, правильных ответах, подсказках. Выбор такого подхода к созданию обучающей системы обусловлен с одной стороны требованиями высокой эффективности применения, с другой универсальностью системы, возможностью её развития, расширения возможностей. На рисунке 2 представлена схема взаимодействия подключаемых к АОС ресурсов.

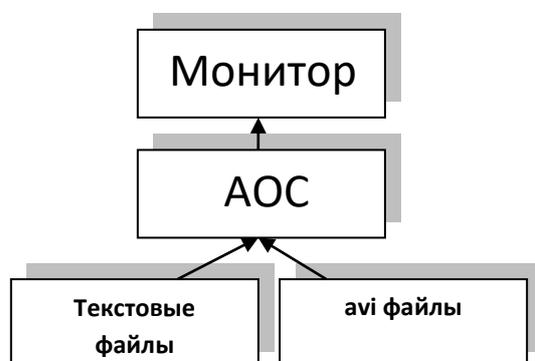


Рис.2 Подключаемые к АОС ресурсы.

## 2 Проектирование системы

В сфере компьютеризации обучения обычно выделяют два основных направления. Целью первого является обеспечение всеобщей компьютерной грамотности, в этом случае сам компьютер является объектом изучения. Второе направление, собственно компьютерное обучение, предполагает использование компьютера как технического средства, полностью или частично выполняющего в отношении обучаемых функции преподавателя. Основная цель второго направления компьютеризации - резко повысить эффективность обучения. К показателям оценки [6] эффективности компьютерного обучения обычно относят: качество усвоения обучаемыми программы обучения, время, затраченное ими на усвоение этой программы, материальные затраты, временные затраты преподавателей и т.д. Ведущим среди всех этих показателей является качество усвоения. Ясно, что никакое снижение материальных, временных и любых других затрат неприемлемо, если качество обучения ухудшается. При этом высокие материальные затраты на реализацию компьютерного обучения диктуют необходимость резкого роста этого главного показателя эффективности по сравнению с существующим традиционным обучением. Данная цель [4] может быть достигнута, если на этапах проектирования, создания и внедрения в учебный процесс соответствующих аппаратно-программных средств будут решены следующие задачи:

1. Гарантированное достижение всеми обучаемыми заданного качества усвоения программы.
2. Достижения каждым из обучаемых заданного качества усвоения наиболее рациональным для него путем, при условии работы преподавателя с достаточно большой группой обучаемых; обеспечение заданного эффекта обучения по отношению к индивидуальным способностям каждого из обучаемых.
3. Наиболее приемлемое преодоление противоречия между всё увеличивающимся объемом информации в изучаемой области человеческой деятельности и относительно фиксированным лимитом времени на ее усвоение.
4. Создания для преподавателя условий, которые бы при общей интенсификации процесса обучения не требовали от него все большей отдачи физических и моральных сил.

Совокупное решение этих задач представляет собой комплексную проблему, находящуюся на стыке целого ряда наук: педагогики, психологии, теории информации, теории системы, теории управления, информатики, системотехники и эргономики. Выше был отмечен целый ряд проблем создания АОС как человеко-машинной системы, в частности, проблемы разработки методологии проектирования АОС; обоснования

концепций системы обучения и системы обучения, основанной на использовании ЭВМ; обоснования и разработки объективных критериев оценки эффективности обучения и методик их определения в процессе педагогических экспериментов. Однако определяющий характер для решения как этих проблем, так и других проблем создания дидактически эффективной АОС, которые могут быть сформулированы только в процессе разработки и создания конкретных систем, имеют проблемы собственно педагогической науки как единственной сферы человеческих знаний, определяющей все закономерности и специфические особенности процесса обучения. Решение проблемы создания на базе ЭВМ человеко-машинных систем, не просто моделирующих обучающую деятельность преподавателя, а именно обеспечивающих гарантированно высокую эффективность этой деятельности, может основываться только на данных педагогической науки. При этом сами данные педагогики и практики обучения нуждаются в переосмыслении с позиций возможности и эффективности их реализации аппаратно-программными средствами современных ЭВМ. Это необходимо предполагает технологизацию данных педагогической науки в качестве единственной возможности решения и задачи обеспечения гарантированной дидактической эффективности обучения, и задачи реализации процесса обучения на основе ЭВМ как техническом средстве, обеспечивающем эту эффективность.

## 2.1 Исследование структуры знаний в системе

При разработке структуры, знания должны быть формализованы в виде совокупности взаимосвязанных элементов.

1. Множество информационных узлов, которым сопоставляются некоторые понятия (объекты, события, процессы, явления). Практически в узлах сети находятся различные порции информации о предмете изучения, представленные в произвольном виде (текст, аудио- и видеозапись, тестовые задания, рисунки, графики, анимация, цитаты из фильмов и т.д.). Одной из первоочередных задач разработки является выявление множества узлов и обеспечение его минимальной избыточности. Границы информационных узлов сети определяются двумя факторами: физиологически оптимальным объемом узла, смысловыми различиями между узлами в контексте обучения по данному предмету.

Физиологически обоснованным объемом узла можно считать информацию, эквивалентную 2—3 печатным страницам текста, при наличии не более 2—3 ссылок на экран. Однако чем ближе структура знаний к линейной, тем больший объем узла допустим. Внутренняя структура узлов сети ничем не регламентируется и может быть основана на разных моделях представления знаний.

2. Множество связей между узлами, которые отражают отношения любого типа между этими узлами и соответствующими понятиями. Связи между узлами (объектами гипермедиа среды) могут интерпретироваться как единство происхождения, степень взаимозависимости, удаленность по определенной шкале, а в целом, как степень общности двух объектов, двух порций информации. Чаще всего связи между узлами можно определить как степень близости узлов по смыслу (семантическая метрика). Такой подход создает возможность экспертной количественной оценки связей между узлами в интервале от полной независимости двух элементов знаний до их полной идентичности.

Первый этап разработки структуры знаний включает решение следующих задач:

1. Выявление основных понятий (информационных узлов), установление их границ.
2. Описание области применения понятий.
3. Частотные характеристики применения понятий.
4. Подробное описание понятий.

## 2.2 Функциональная модель процесса обучения

Целью функциональной модели процесса обучения является определение множества путей обучения, то есть последовательностей информационных узлов, которые предъявляются в ходе обучения. Полная свобода навигации (передвижения от узла к узлу) не оптимальный способ обучения. Свобода навигации по обучающей программе должна как-то быть ограничена, и определение степени ограничена, и определение степени ограничения навигации — важнейший результат разработки.

Определение оптимальных путей навигации наисложнейшая проблема, правильное решение, которой возможно лишь в результате практического применения с последующей неоднократной модификацией. Фактически, это точка зрения учителя, как лучше изучать представленный материал. Однако есть и иной способ выбора навигации — «снизу-вверх». В этом случае ученикам рекомендуются наиболее популярные, по данным статистики, пути достижения узлов, которые выявляются при анализе действий разных

групп обучающихся в условиях полной свободы навигации. Используя оба эти способа ограничения навигации по АОС, можно достичь эффективного уровня свободы навигации. Для любого уровня обучающегося должны быть выработаны критерии полноты знаний, в соответствии с которыми обучение может считаться успешным. Эти критерии можно представить как ограниченную область сетевой структуры знаний, которую ученик изучил с заданным уровнем рейтинга. Обучающая программа должна сопровождаться массивом справочной информации, доступным из любого режима работы программы, и именно эта информация для пользователя является определяющей в выборе пути обучения.

### 2.3 Информационная модель процесса обучения

Если сформулировать кратко, то информационную модель процесса обучения выглядит так "что, где, когда", т.е. могут быть получены ответы на вопросы: *какие* имеются информационные ресурсы, *где* они находятся и в какой последовательности, *когда* к ним возможен доступ.

### 2.4 Контроль уровня знаний

В традиционных системах обучения контроль знаний проводится с помощью нескольких вопросов (два – три в билете плюс несколько дополнительных). Полученные обучаемым оценки за ответы на эти вопросы распространяются и на не проконтролированные разделы учебного материала, использование тестового контроля минимизирует затраты рабочего времени экзаменатора и даёт хотя и поверхностный, но полный охват всего учебного [7] материала. В настоящее время в учебных заведениях, помимо традиционных методов контроля знаний учащихся, используется тестовый контроль, так как он является наиболее объективной формой оценки качества знаний. При разумной организации контролируемые (тестируемые) программы являются весьма эффективным средством контроля знания, так как большая часть учебного материала легко поддается машинному контролю. Возможность автоматизации проверки и уменьшения времени выполнения учащимися самих операций контроля приводит к снижению времени контрольной деятельности учащихся, что дает возможность увеличить частоту и регулярность

контроля. Однако проводить качественное тестирование без применения специальных средств практически невозможно из-за большого объема данных, который необходимо переработать преподавателю: во-первых, необходимо подготовить большое число вариантов тестов с неповторяющимися вопросами, а подготовка теста – весьма трудоемкий процесс, во-вторых, оценка результатов тестирования, особенно при использовании статистических методов, достаточно сложна. Для решения этих проблем используются программные средства, которые дают возможность преподавателю оперативно составлять множество вопросов по теме (курсу) и оценивать результаты тестирования. Поэтому неотъемлемым компонентом обучающей программы является функция контроля уровня знаний. В рассматриваемой программе контроль знаний осуществляется по двум критериям. Первый это количество правильных ответов и второй — это количество времени, затраченное на прохождение всего задания. Фрагмент интерфейса результата выполнения задания представлен на рисунке 3

### 3. Разработка программного обеспечения

Программирование всегда было достаточно сложной задачей. Как выбрать способ, наиболее подходящий для решения конкретной задачи, и как добиться максимальной производительности приложения. Прошли те времена, когда исследователю месяцами приходилось перерабатывать исходный код какой-нибудь бесполезной утилиты в практическом плане, забираться в недра исходного текста оригинально сделанного компонента. Конечно, небольшому проценту разработчиков по долгу службы нужны глубокие специфические знания, однако от подавляющего большинства программистов сегодня требуется, прежде всего, умение писать программы максимально быстро и без ошибок. При этом, чем меньше будут использоваться малоизвестные возможности Windows, тем лучше – ведь в исходных текстах, вполне возможно придется разбираться и усовершенствовать другим программистам, причем скорее всего разной квалификации. Графические интерфейсы пользователя, или GUIs, революционизировали микрокомпьютерную индустрию. Они продемонстрировали, что выражение "Лучше один раз увидеть, чем сто раз услышать" не потеряло своего смысла для большинства пользователей компьютеров. Вместо загадочной командной строки. C:>, которую так долго наблюдали пользователи DOS, теперь они смотрят на "рабочий стол" (desktop), заполненный значками программ, управляя ими при помощи мыши или посредством меню. Вероятно, при длительном использовании Windows то, как выглядят приложения, менее важно, чем-то, как они созданы. В Windows, как правило, программы имеют стандартный интерфейс пользователя. В то время как программисты испытывают смешанные чувства к GUIs, он нравится начинающим пользователям, поэтому от любых программ Windows ожидается соответствие модели GUI. Поэтому, если необходимо разработать программу для любой из версий Windows, программисту понадобится эффективное средство для разработки приложений с графическим интерфейсом пользователя. Это позволит программисту сосредоточиться только на логике решаемой задачи. В таком контексте система Visual Basic (VB) представляется средой, близкой к идеальной, а соответственно система VB наиболее приемлема для образования. VB – самая современная система программирования и в тоже время доступна для освоения даже новичками. VB является системой программирования высокого уровня. Она берет на себя значительную часть работы по управлению компьютером, что делает возможным в простых случаях обходиться без особых знаний о деталях его работы. В отличие от традиционных систем программирования.

### 3.1 Разработка пользовательского интерфейса.

Визуальное программирование как бы добавляет новое измерение при создании приложений, давая возможность изображать объекты пользовательского интерфейса на экране монитора до выполнения самой программы. Без визуального программирования процесс отображения требует написания фрагмента кода, создающего и располагающего объект «по месту». Увидеть закодированные объекты было возможно только в ходе исполнения программы. При таком подходе достижение того, чтобы объекты выглядели и вели себя заданным образом, становится утомительным процессом, который требует неоднократных исправлений программного кода с последующей прогонкой программы и наблюдения за тем, что в итоге получилось. Благодаря средствам визуальной разработки можно работать с объектами, держа их перед глазами и получая результаты практически сразу. Способность видеть объекты такими, какими они появляются в ходе исполнения программы, снимает необходимость проведения множества операций вручную, что характерно для работы в среде не обладающей визуальными средствами — вне зависимости от того, является она объектно-ориентированной или нет. После того, как объект помещен в форму среды визуального программирования, все его атрибуты сразу отображаются в виде кода, который соответствует объекту как единице, исполняемой в ходе работы программы. Размещение [8] объектов в VB связано с более тесными отношениями между объектами и реальным программным кодом. Объекты помещаются в вашу форму, при этом код, отвечающий объектам, автоматически записывается в исходный файл. Этот код компилируется, обеспечивая существенно более высокую производительность, чем визуальная среда, которая интерпретирует информацию лишь в ходе исполнения программы. Замечательным достоинством системы является и то, что размещение компонентов на экране, а также задание начальных значений их свойств (размеры, цвет, вид и др.) VB позволяет осуществлять на этапе конструирования формы без написания какой-либо программы. Для этой цели предусмотрено специальное окно, называемое Инспектором объектов, в котором перечислены все доступные в режиме проектирования свойства выделенного компонента и их текущие значения. Изменение свойства какого-либо объекта незамедлительно отразится на внешнем виде и коде программы. Это позволяет, уже до запуска программы видеть как будет выглядеть проектируемая форма. Такой способ работы с объектами, имеющими графическое представление, принято называть объектно-ориентированным программированием.

### 3.2 Разработка структуры подключаемых ресурсов

В программе используются подключаемые ресурсы двух типов это обычный текстовый файл и видео в формате avi. Главным принципом, определившим структуру подключаемого ресурса, является простота его создания. Действительно, мы живём в период бурной информатизации всех сфер жизни человека, и сохранить текст в текстовом файле не составляет труда. Что же касается видео, то видеокамера на сегодняшний день так же не редкость. Известная поговорка гласит: «Лучше 1 раз увидеть, чем 100 раз услышать». Тем самым, утверждается тезис, что «видеть и слышать», более эффективно по качеству освоения и скорости запоминания. Иными словами, использование видео уроков обеспечивает наглядное, глубокое и более быстрое освоение материала. Видео уроки являются дополнительной формой изучения-закрепления материала. При этом возможно их использование, как в рамках обучающих программ, так и вне этих рамок. Учебный процесс ограничен временными рамками, преподаватель заинтересован в предоставлении учащимся более глубоких знаний, рекомендует дополнительные источники. Изучение этой литературы достаточно трудоёмко и на практике редко применяется. Подчас учащиеся по тем или иным причинам отсутствуют на занятиях относительно длительное время. Наверстать упущенное, вполне по силам с помощью видеолекций. Видеоуроки это очень перспективное направление в развитии информационных ресурсов вообще и применительно к АОС в частности. Качество видеоурока зависит от целого ряда субъективных и объективных факторов. Для одного человека скажем внешний вид преподавателя не мешает воспринимать информацию, чего не скажешь о другом. Конечно же не последнее место занимает умение преподавателя подать материал доступно, без надрыва. Впрочем, создание хороших видеоуроков как и любых других продуктов требует профессионализма и настойчивости. Структура текстового файла определена логикой его чтения. Программа [9] тренажёр при загрузке задания из файла «считает», что первый фрагмент, заключённый в кавычки является заданием, второй, правильным ответом, третий, подсказкой, четвёртый, снова задание и т.д. Ниже показан фрагмент такого файла.

.....

"Напечатайте название логического типа переменных"

"boolean"

"boolean"

"Напечатайте название целочисленного типа переменных, имеющих диапазон -32 76 32 767"

"integer"

"integer"

"Напечатайте название типа переменной, максимальное значение которой может быть 255"

"byte"

"byte"

"Напечатайте название типа вещественной переменной, значение которой, число с точностью представления до 11-12 знака в дробной части"

"real"

"real"

"Напечатайте оператор присвоения"

":="

":=" .....

### 3.3 Описание программы.

Программа написана в среде разработки Visual Basic 6. **Microsoft Visual Basic** — средство разработки программного обеспечения, разрабатываемое корпорацией Microsoft и включающее язык программирования и среду разработки. Язык Visual Basic унаследовал дух, стиль и отчасти синтаксис своего предка — языка Бейсик, у которого есть немало диалектов. В то же время Visual Basic сочетает в себе процедуры и элементы объектно-ориентированных и компонентно-ориентированных языков программирования. Среда разработки VB включает инструменты для визуального конструирования

пользовательского интерфейса. Visual Basic считается хорошим средством быстрой разработки прототипов программы, для разработки приложений баз данных и вообще для компонентного способа создания программ, работающих под управлением операционных систем семейства Microsoft Windows. Первое признание серьёзными разработчиками Visual Basic получил после выхода версии 3 — VB3. Окончательное признание как полноценного средства программирования для Windows — при выходе версии 5 — VB5. Версия VB6, входящая в состав Microsoft Visual Studio 6.0, стала по-настоящему зрелым и функционально богатым продуктом. После этого разработчики из Microsoft существенно изменили направление развития данной технологии.

### 3.3.1 Общие сведения

Программа включает в свой состав 4 базовые формы с более чем пятьюдесятью объектами с программным кодом (приложение 1). Кроме того через интерфейс программы можно управлять sixtyю демонстрационными приложениями, в том числе среда разработки Pascal ABC с электронным учебником, а так же подключать видеолекции интернет университета Интуит и самостоятельно созданные видеолекции. Первая загружаемая форма (см. рис. 4) является титульной, кроме того выполняет функцию гипертекстового оглавления.

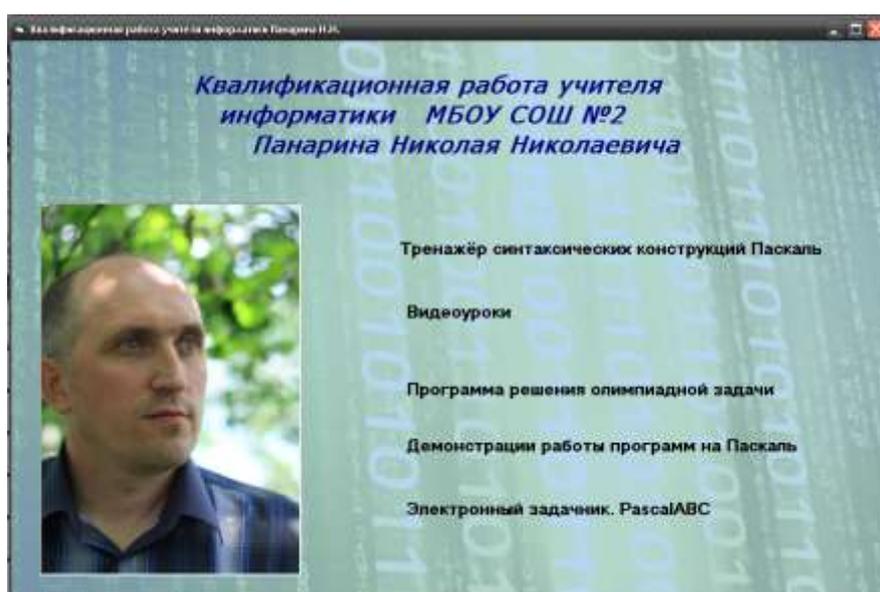


Рис. 4 Титульная страница приложения

На базе второй формы создан тренажёр синтаксических конструкций Pascal (см. рис. 5)

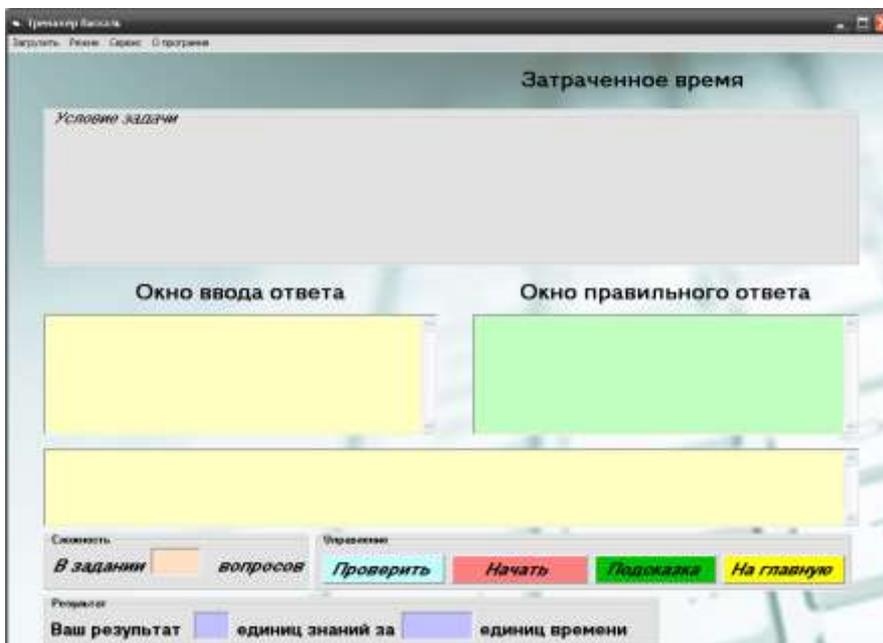


Рис. 5 Тренажёр синтаксических конструкций

В третьей форме построен интерфейс, позволяющий смотреть видеоуроки из локальных источников, а также видео лекции интернет университета Интуит (см. рис. 6)

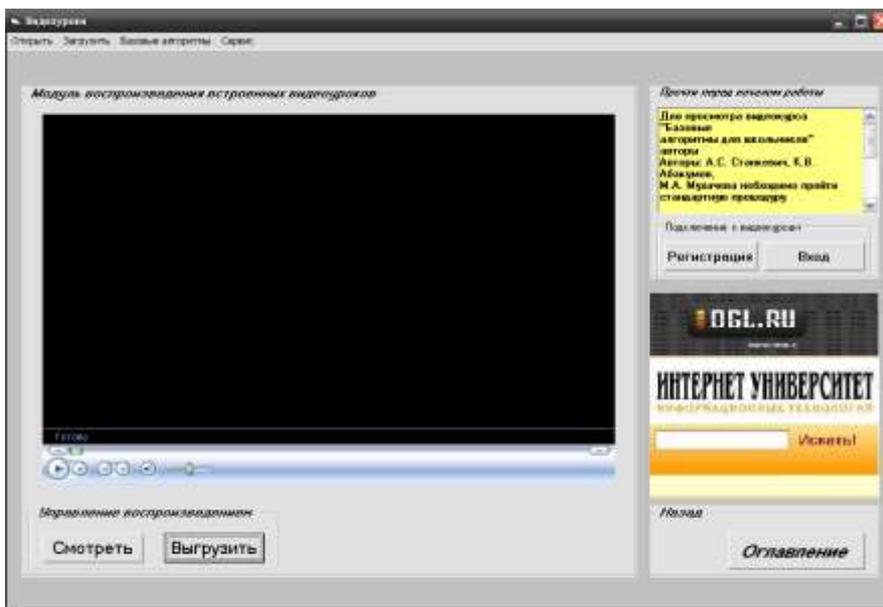


Рис. 6 Интерфейс просмотра видеоуроков

Четвёртая форма обеспечивает доступ к демонстрациям работы базовых алгоритмов реализованных на Pascal (см. рис. 7)

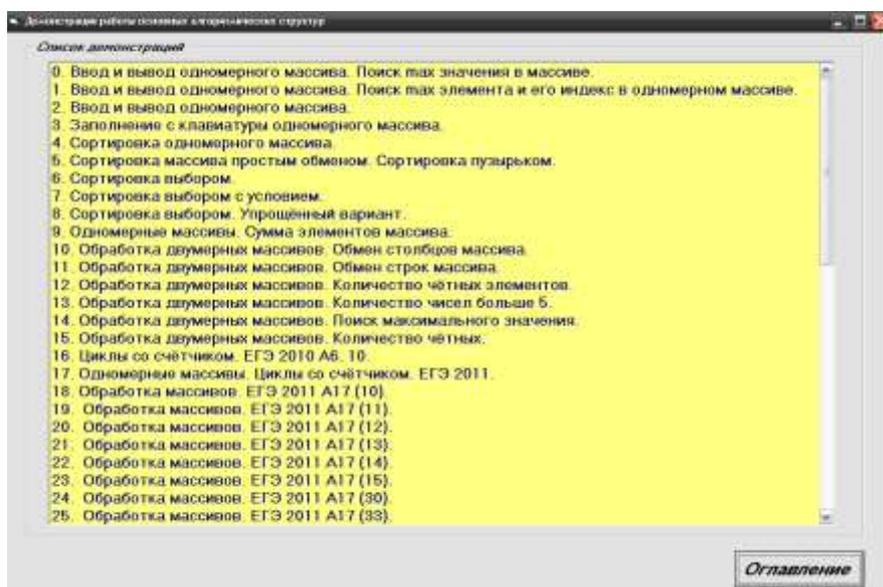


Рис. 7 Выбор демонстраций алгоритмов реализованных на Pascal

### 3.3.2 Описание логической структуры программы

На рисунке 8 представлена логическая схема работы программы.

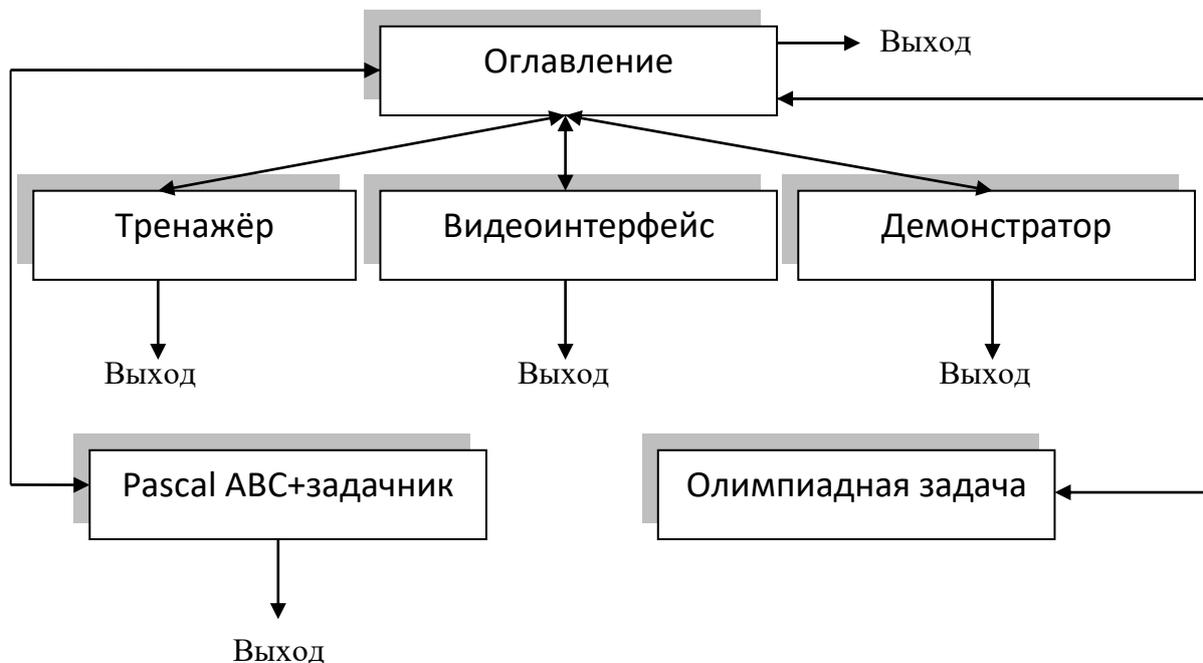


Рис. 8 Логическая схема работы программы

Навигация по программе осуществляется через титульную форму (см. рис. 4), где расположено гипертекстовое меню. Выход из приложения возможен из любой формы. Переходы между формами так же через титульную страницу.

### 3.3.3 Описание модулей и основных структур

Как отмечалось выше титульная страница выполняет функции навигации по программе. Переходы выполняются по одинарному клику на соответствующий переходу раздел. Выбранный раздел подсвечивается красным цветом. Выход из приложения осуществляется через стандартный элемент окна (см. рис. 9)



Рис. 9 Работа с титульной формой

После входа в тренажёр синтаксических конструкций, необходимо в меню **загрузить** выбрать задание и загрузить его (см. рис. 10)

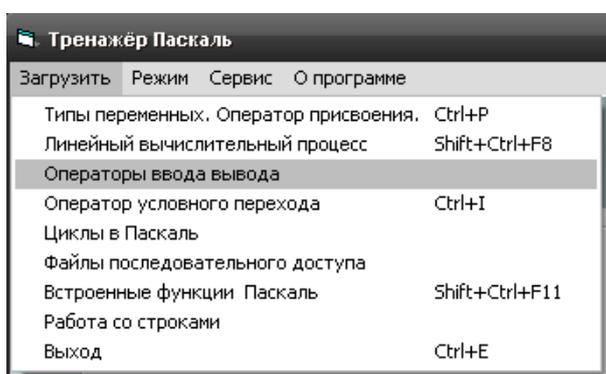


Рис. 10 Загрузка задания в тренажёр

Для начала работы необходимо нажать кнопку «Начать». После нажатия на эту кнопку выводится количество заданий, первое условие задачи и запускается отчёт времени (см. рис. 11)

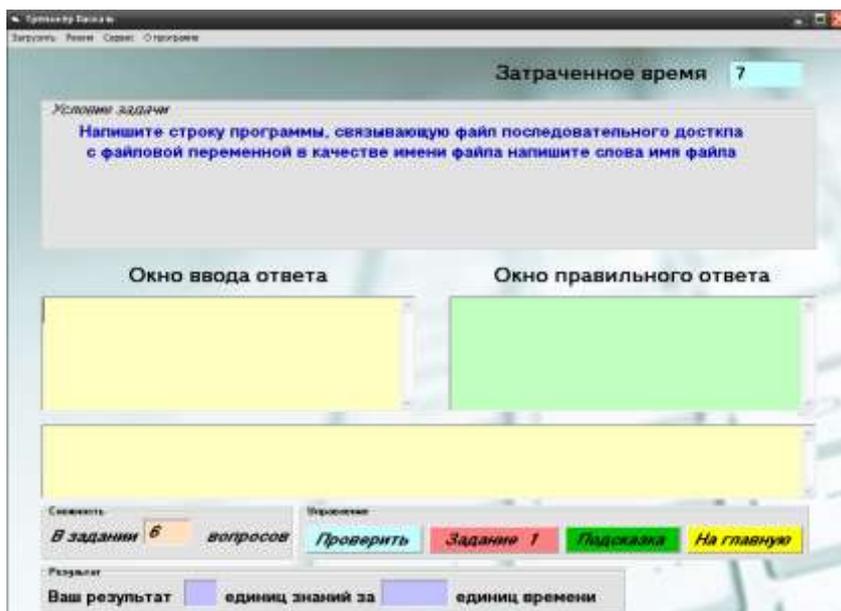


Рис. 11 Начало работы в тренажёре синтаксических конструкций

После ввода в поле «Окно ввода ответа» предполагаемого ответа необходимо нажать на кнопку «Проверить». В случае правильного ответа поле «Окно ввода ответа» подсвечивается зелёным светом, в противном случае красным. Если ученик затрудняется с ответом то он может вызвать подсказку нажатием соответствующей кнопки. Количество повторов неограниченно, однако балл знаний снижается, кроме того идёт время (см. рис.12).

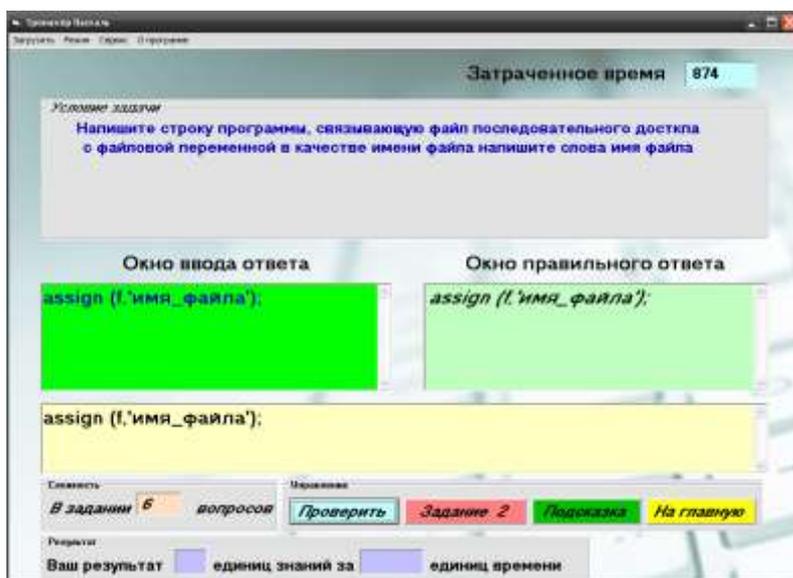


Рис. 12 Рабочий режим

В меню сервис можно сделать настройку заднего фона (см. рис 13)

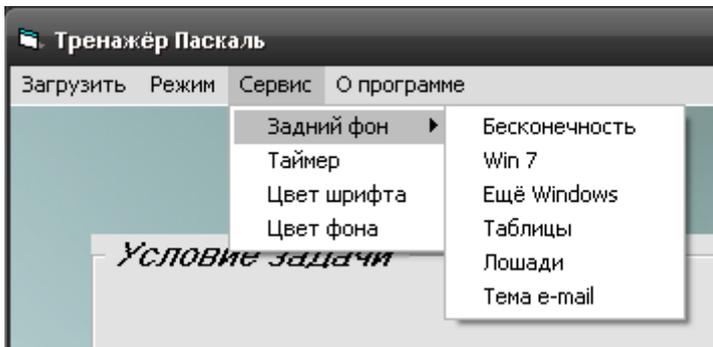


Рис. 13 Выбор заднего плана

Видеоинтерфейс реализован в третьей форме. С его помощью можно воспроизводить видеоматериал на локальном диске, а так же подключаться к видео лекциям интернет университета Интуит. Через меню открыть можно выбрать файл для воспроизведения (см. рис. 14). Кроме этого, можно воспроизвести встроенные видеоуроки, воспользовавшись меню – загрузить (см. рис. 15)

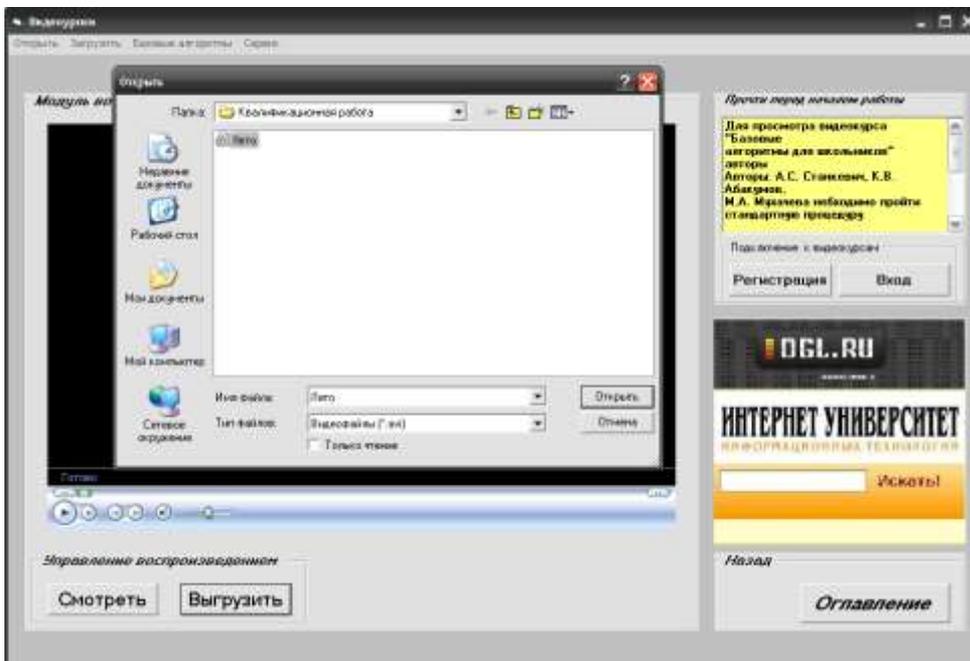


Рис. 14 Выбор файла для воспроизведения

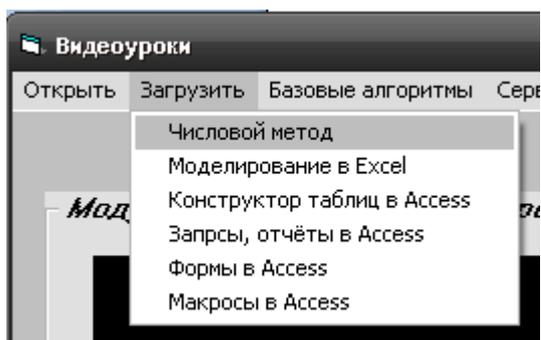


Рис. 15 Загрузка встроенных видеоуроков

Если компьютер подключён к сети интернет, можно посмотреть курс видео лекций «Базовые алгоритмы для школьников» Авторы: А.С. Станкевич, К.В. Абакумов, М.А. Мухачева. Для этого необходимо выполнить регистрацию на портале Интуит, нажав кнопку Регистрация. После чего необходимо осуществить вход на портал под своим именем и паролем, воспользовавшись кнопкой вход, после успешного входа на главную страницу портала, не закрывайте её, а сверните в панель задач, после чего можно приступить к выбору и просмотру лекций (см. рис. 16). Меню сервис позволяет выбрать фон загружаемый в форму.

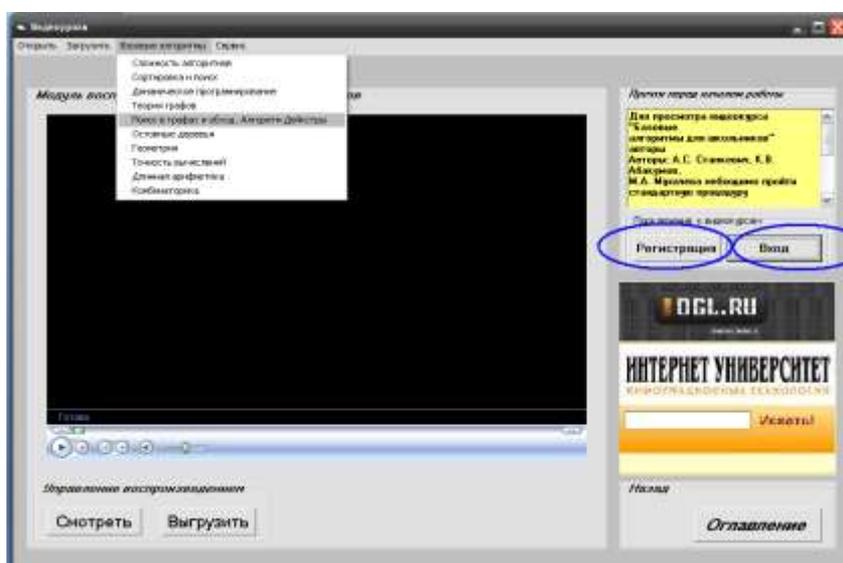


Рис. 16 Подключение к видео лекциям интернет университета Интуит

Для просмотра демонстрации работы основных алгоритмических структур реализованных на Pascal необходимо перейти по соответствующей ссылке из оглавления на форму выбора демонстрации (см. рис. 17)

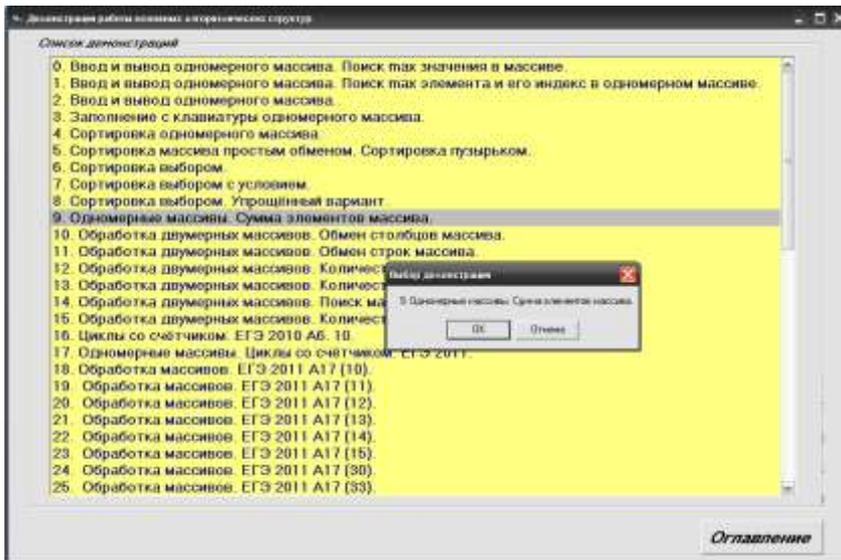


Рис. 17 Выбор демонстрации работы алгоритмов на Pascal

Завершающим пунктом оглавления является вызов среды разработки на Pascal в которую включён сборник задач с тестовыми входными и выходными параметрами (см. рис.18)

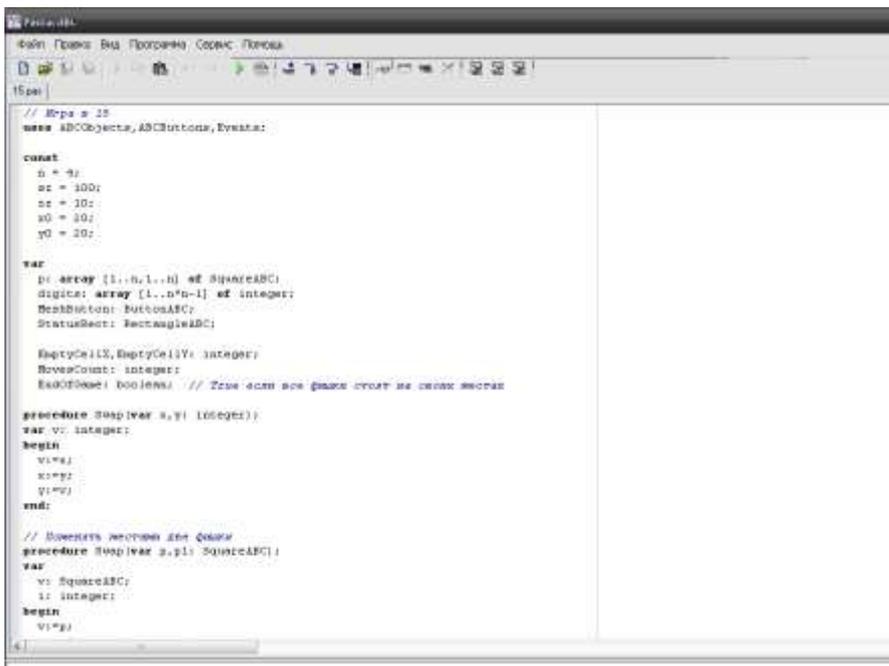


Рис. 18 Среда разработки на Pascal с встроенным задачником

#### 4. Оценка эффективности

Зачем надо оценивать эффективность учебных программ. Основная причина, по которой необходимо оценивать эффективность обучающих программ, – это выяснение того, в какой степени в итоге были достигнуты цели обучения. Менее эффективная система обучения, должна быть изменена или заменена другой. Даже хорошие программы могут потерпеть неудачу по многим причинам: могут быть поставлены нереалистичные или слишком общие цели обучения, может быть плохо организован сам процесс обучения, возможны срывы по причинам, находящимся в зоне непреодолимой силы (например, болезнь преподавателя, поломки оборудования), и др. Выявление причин, по которым данная программа обучения потерпела неудачу, и их анализ позволяют предпринять в будущем необходимые корректирующие меры. Оценка эффективности [10] обучения может проводиться с помощью тестов, вопросников, заполняемых обучающимися, экзаменов и т.п. Оценивать эффективность обучения могут преподаватели, эксперты или специально созданные целевые группы.

Можно выделить пять критериев, обычно используемых при оценке эффективности обучения:

- мнение обучающихся;
- усвоение учебного материала;
- поведенческие изменения;
- рабочие результаты.

Еще одним ценным источником информации об отношении слушателей к учебному курсу (программе) может быть неформальное общение с учащимися и наблюдение за ними как во время занятий, так и в перерывах, и после окончания занятий. Замечания, комментарии, вопросы, задаваемые обучающимися, обычно отражают общее отношение. К сожалению, с момента апробации программы в учебном классе прошло всего лишь месяц, однако, некоторые выводы уже можно сделать. Программа была предложена для работы ученикам 9 класса, по плану у которых начиналось программирование в Pascal. Мнение учеников по опросному листу приведу ниже. Опрошены были 12 человек.

Оцените по пятибалльной шкале:

Интерфейс 5баллов 10учеников, 4балла 2ученика;

Эффективность тренажёра 5баллов 9учеников, 4балла 3ученика;

Полезность видеоуроков 5баллов 11 учеников, 4 балла 1 ученик;

Полезность демонстраций 5 баллов 11 учеников, 4балла 1 ученик;

Общая оценка программы 5 баллов 12учеников.

В целях получения более или менее достоверных данных опрос письменно и анонимно проводил психолог школы. Мониторинг усвоения учебного материала по итогам месяца показал рост по сравнению с таким же периодом прошлого года, однако, эти данные не могут считаться объективными так как мониторинг уровня обученности учеников 9 класса этого года выше по сравнению с девятиклассниками прошлого года.

#### 4.1. Перспективы дальнейшего развития системы

Происходящую модернизацию образовательных технологий можно представить как развитие технологий классического образования (преподаватель, доска-прибор, библиотека, аудитория) в образование с использованием современных компьютерных технологий. Поэтому перспективы лишь за комплексными программными продуктами, вбирающими в себя все достоинства работы на локальном компьютере, а так же возможность интерактивной сетевой работы. Очевидна, таким образом, перспектива развития рассматриваемого в рамках данной квалификационной работы комплекса программ в сторону большей их универсальности (возможность применения для обучения другим дисциплинам), развития базы заданий с возможностью удобной разработки и интеграции в систему собственных задания для тренажёра, оптимизации навигации по карте знаний и заданий исходя из уровня начальной подготовленности обучаемого и обеспечения индивидуальных траекторий обучения, введения в логику взаимодействия различных алгоритмов программы здоровые развивающих идей и подходов, совершенствование и расширение базы знаний программы за счёт локальных и сетевых ресурсов, кроссплатформенность продукта .

## Заключение

Создание обучающих программ творческий процесс, требующий не только логического мышления, но и интуиции. Этот процесс еще изучен недостаточно и не может быть описан с помощью жестких нормативов-предписаний. Много опасностей и ловушек подстерегает разработчиков обучающих программ. Для педагогов самая большая опасность механический перенос особенностей обучения в классе (группе) на компьютерное обучение, стремление как можно более точно скопировать работу педагога. Хотелось бы отметить, что механический перенос в принципе недопустим по следующим причинам: Даже самый опытный педагог, мастер своего дела, далеко не всегда сможет описать свою деятельность и тем более объяснить каждое свое решение (многие решения принимаются педагогом интуитивно, они не полностью осознаются, и на вопрос, почему принято именно такое, а не иное решение в большинстве случаев отвечают: так подсказал опыт, это известно из практики и т.д.). Групповое, классное обучение, опыт которого приобретает педагог, не является адекватной моделью компьютерного обучения, которое обладает многими особенностями индивидуального обучения, существенно отличаются от группового. Компьютер не только накладывает определенные ограничения на реализацию учебного процесса, он раскрывает новые возможности в управлении учебной деятельностью. Это происходит прежде всего за счет неограниченных возможностей в предъявлении материала, применения разнообразных учебных задач, построения модели обучаемого путем накопления и переработки больших массивов данных, относящихся к учащемуся, неограниченного запаса знаний, относящихся к данной предметной области, и т.п. Кроме того следует иметь в виду, что разработка обучающих программ - это качественно иная, в сравнении с практической, деятельность педагога. Можно уметь решить задачу, но не уметь составить алгоритм. А ведь при разработке обучающей программы необходимо составить алгоритм работы компьютера, который отнюдь не копирует, а моделирует деятельность педагога и даже те же самые функции реализует иными способами. К тому же разработка обучающих программ требует более глубоких знаний не только в определенной предметной области, но и знаний об учебном процессе и учащихся. Мировой опыт убедительно показывает, что даже опытные практические работники, прошедшие специальную подготовку, нередко составляют весьма бледные обучающие программы, которые дают результаты значительно хуже, чем традиционное обучение. Справедливости ради стоит отметить, что далеко не все обучающие программы, составленные специалистами в области обучения, оказались эффективными. Многие из

них настолько скучные и неинтересные, что от них отказались как преподаватели, так и ученики. Составление обучающих программ — это наука и искусство. Оно требует и глубоких знаний, и педагогического таланта. Все задачи поставленные в данной квалификационной работе были успешно решены.

Список использованных источников.

[1] Печников А.Н. Теоретические основы психолого-педагогического проектирования автоматизированных обучающих систем. Петродворец: ВВМУРЭ им. А.С.Попова, 1995. - 322 с.

[2] Психологические аспекты общения и работа с информацией (М.М. Жинкин, И.А. Зимняя, В.А. Канкалин, Б.Ф. Ломов, А.А. Леонтьев, Б.Ф. Поршнев, З. Фрейд и др.). <http://www.dissercat.com/content/informatsionnaya-model-obucheniya-teoriya-i-praktika-postroeniya-i-realizatsii-v-vuze>

[3] *д.т.н. проф., Мельников А.В, асс. Цытович П.Л.* Принципы построения обучающих систем и их классификация. Южно-уральский государственный университет [pavel@comp.tu-chel.ac.ru](mailto:pavel@comp.tu-chel.ac.ru). [http://scholar.urfu.ac.ru/ped\\_journal/numero4/pedag/tsit3.html.ru](http://scholar.urfu.ac.ru/ped_journal/numero4/pedag/tsit3.html.ru)

[4] Процкий Э.С. “Проверка знаний умений навыков”// Москва – 1986

[5] История развития методологии проектирования АОС Чертова Екатерина <http://catalog.ru/nauka/istoria-razvitia-metodologii-proektirovaniya-aos.html>

[6] Теории информации и языковой коммуникации (О.Е. Бурый-Шмарьян, Н.Винер, Д.В. Демин, О.В. Елчанинова, А. Мартине, Дж. Миллер, Б.Ф. Поршнев, Г.Г. Почепцов, Е.П. Шубин и др.); *Говорящий и слушающий: СПб, 2001. с.70-75.*

[7] Принципы и методы системного подхода к анализу педагогических исследований (С.И. Архангельский, Ю.П. Сокольников и др.); <http://www.dissercat.com/content/informatsionnaya-model-obucheniya-teoriya-i-praktika-postroeniya-i-realizatsii-v-vuze>

[8] Использование Visual Basic 6. Брайан Сайлер и Джефф Споттс. Классическое издание.: Пер. с англ. – СПб.; Издательский дом «Вильямс», 2007. – 832с.

[9] Дмитрий Безуглый. Технология разработки программного обеспечения

Корпоративные системы №1 2002г

[10] Принципы и методы системного подхода к анализу педагогических исследований (С.И. Архангельский, Ю.П. Сокольников и др.);

<http://www.dissercat.com/content/informatsionnaya-model-obucheniya-teoriya-i-praktika-postroeniya-i-realizatsii-v-vuze>

## Приложение 1

### **Код объектов формы 1.**

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    If ((X < 14200) Or (X > 6200)) Or ((Y < 4950) Or (Y > 4100)) Then
```

```
        Label2.ForeColor = 255
```

```
    End If
```

```
    If ((X > 14200) Or (X < 6200)) Or ((Y > 4950) Or (Y < 4100)) Then
```

```
        Label2.ForeColor = 0
```

```
    End If
```

```
    If ((X < 14200) Or (X > 6200)) Or ((Y < 6750) Or (Y > 5300)) Then
```

```
        Label3.ForeColor = 255
```

```
    End If
```

```
    If ((X > 14200) Or (X < 6200)) Or ((Y > 6750) Or (Y < 5300)) Then
```

```
        Label3.ForeColor = 0
```

```
    End If
```

```
    If ((X < 14200) Or (X > 6200)) Or ((Y < 8300) Or (Y > 7100)) Then
```

```
        Label4.ForeColor = 255
```

```
    End If
```

```
    If ((X > 14200) Or (X < 6200)) Or ((Y > 8300) Or (Y < 7100)) Then
```

```
        Label4.ForeColor = 0
```

```
    End If
```

```
    If ((X < 14200) Or (X > 6200)) Or ((Y < 9200) Or (Y > 8200)) Then
```

```
        Label5.ForeColor = 255
```

```
    End If
```

```
    If ((X > 14200) Or (X < 6200)) Or ((Y > 9200) Or (Y < 8200)) Then
```

```
        Label5.ForeColor = 0
```

```
    End If
```

```
    If ((X < 14200) Or (X > 6200)) Or ((Y < 10600) Or (Y > 9700)) Then
```

```
        Label6.ForeColor = 255
```

```
    End If
```

```
    If ((X > 14200) Or (X < 6200)) Or ((Y > 10600) Or (Y < 9700)) Then
```

```
        Label6.ForeColor = 0
```

```
    End If
```

```
End Sub
```

```
Private Sub Label2_Click()
```

```
    Unload Form1
```

```
    Set Form1 = Nothing
```

```
    Form2.Show
```

```
End Sub
```

```
Private Sub Label3_Click()
```

```
    Unload Form1
```

```
    Set Form1 = Nothing
```

```
    Form3.Show
```

```
End Sub
```

```
Private Sub Label4_Click()  
Dim dTaskID As Double  
dTaskID = Shell(App.Path & "\\project3.exe", vbNormalFocus)  
End Sub
```

```
Private Sub Label5_Click()  
Unload Form1  
Set Form1 = Nothing  
Form4.Show  
End Sub
```

```
Private Sub Label6_Click()  
Dim dTaskID As Double  
dTaskID = Shell(App.Path & "\\PascalABC.exe", vbNormalFocus)  
End Sub
```

### ***Код объектов формы 2.***

```
Dim z(1 To 20) As String  
Dim o(1 To 20) As String  
Dim p(1 To 20) As String  
Dim t As Byte  
Dim i As Byte  
Dim ball As Integer  
Dim chtraf As Integer  
Dim nahalo As Double  
Dim flag As Integer
```

```
Private Sub besc_Click(Index As Integer)  
Form2.Picture = LoadPicture(App.Path & "\\2.JPG")  
End Sub
```

```
Private Sub cikl_Click(Index As Integer)  
Text4.Text = "00" 'сброс времени при загрузке задания  
Text4.Visible = False  
Text1.Text = ""  
Text2.Text = ""  
Text3.Text = ""  
Text5.Text = ""  
Label10.Caption = ("Тема занятия: Циклы.")  
Command3.Caption = ("Начать")  
Label2.Caption = ""  
t = 0  
i = 0  
Open App.Path & "\\4.txt" For Input As #1  
While Not EOF(1)  
i = i + 1  
Input #1, z(i)  
Input #1, o(i)  
Input #1, p(i)  
Wend  
Close #1
```

```

Text5.Text = i
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True

MsgBox "Загрузка завершена. Нажмите кнопку начать", vbInformation, "Загрузка задания"

End Sub

Private Sub Command1_Click()
Unload Form2
Set Form2 = Nothing
Form1.Show
End Sub

Private Sub Command2_Click()
If (t = 0) And (z(1) = "") Then
MsgBox "Сначала загрузите задание", vbOKOnly, "Предупреждение"
Exit Sub
End If
chtraf = chtraf + 2           'подсчёт штрафных баллов
If t = 0 Then
MsgBox "Нажмите кнопку начать", vbOKOnly, "Предупреждение"
Exit Sub
End If
Text3.Text = p(t)          'вывод подсказки
End Sub

Private Sub Command3_Click()   'кнопка задание - начать
If (Text1.Text = "") And (t > 0) Then
MsgBox "Вы не справились с предыдущим заданием, воспользуйтесь подсказкой", vbOKOnly,
"Предупреждение"
Exit Sub
End If

Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
If t = 0 Then
If z(1) = "" Then
MsgBox "Сначала загрузите задание", vbOKOnly, "Предупреждение"
Exit Sub
End If
Text4.Text = "00"           'начало отсчёта времени
Text4.Visible = True

nahalo = Timer
End If
If t = i Then               'выход из программы
Text7.Text = CInt(Timer - nahalo)
Text6.Text = ball - chtraf

```

```
flag = MsgBox("Повторить выполнение этого задания?", vbOKCancel, "Завершение работы с заданием")
```

```
If flag = 1 Then  
Command3.Caption = ("Начать")  
Label2.Caption = ""  
Text1.BackColor = vbWhite  
t = 0  
Text5.Text = ""  
Text6.Text = ""  
Text7.Text = ""  
Text1.BackColor = vbWhite  
Text1.Text = ""  
Label2.Caption = ""  
Text4.Visible = False  
Exit Sub  
Else  
Text1.Text = ""  
Text2.Text = ""  
Text3.Text = ""  
Command3.Caption = ("Начать")  
Label2.Caption = ""  
Command2.Visible = False  
Command3.Visible = False  
Command4.Visible = False  
Text4.Visible = False  
Text1.BackColor = vbWhite  
Text6.Text = ""  
Text7.Text = ""  
Text5.Text = ""  
t = 0  
Exit Sub  
End If  
End If  
Text1.SetFocus
```

```
Text1.BackColor = vbWhite  
t = t + 1  
Label2.Caption = z(t)
```

```
Command3.Caption = ("Задание ") & Str(t)  
End Sub
```

```
Private Sub Command4_Click()  
If (t = 0) And (z(1) = "") Then  
MsgBox "Сначала загрузите задание", vbOKOnly, "Предупреждение"  
Exit Sub  
End If  
If t = 0 Then  
MsgBox "Нажмите кнопку начать", vbOKOnly, "Предупреждение"  
Exit Sub
```

```

End If
If Text1.Text = "" Then
MsgBox "Вы ещё ничего не напечатали в окно ответов", vbOKOnly, "Предупреждение"
Exit Sub
End If
'кнопка проверить
If Text1.Text = o(t) Then
Text2.Text = o(t)
Text1.BackColor = vbGreen
ball = ball + 5
Else
Text1.BackColor = vbRed
End If
If t = i Then
'выход из программы
Text7.Text = Text4.Text
Text6.Text = ball - chtraf
flag = MsgBox("Повторить выполнение этого задания?", vbOKCancel, "Завершение работы с заданием")

If flag = 1 Then
Command3.Caption = ("Начать")
Label2.Caption = ""
Text1.BackColor = vbWhite
t = 0
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text1.BackColor = vbWhite
Text1.Text = ""
Label2.Caption = ""
Text4.Visible = False
Exit Sub
Else
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Command3.Caption = ("Начать")
Label2.Caption = ""
Command2.Visible = False
Command3.Visible = False
Command4.Visible = False
Text4.Visible = False
Text1.BackColor = vbWhite
Text6.Text = ""
Text7.Text = ""
Text5.Text = ""
t = 0
Exit Sub
End If
End If

Command3.Caption = ("Задание ") & Str(t + 1)
Label2.Caption = z(t)

```

End Sub

```
Private Sub files_Click(Index As Integer)
Text4.Text = "00" 'сброс времени при загрузке задания
Text4.Visible = False
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text5.Text = ""
Label10.Caption = ("Тема занятия: Файлы последовательного доступа.")
Command3.Caption = ("Начать")
Label2.Caption = ""
t = 0
i = 0
Open App.Path & "\7.txt" For Input As #1
While Not EOF(1)
i = i + 1
Input #1, z(i)
Input #1, o(i)
Input #1, p(i)
Wend
Close #1
Text5.Text = i
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
```

```
MsgBox "Загрузка завершена. Нажмите кнопку начать", vbInformation, "Загрузка задания"
```

End Sub

```
Private Sub Form_Load()
t = 0
ball = 0
chtraf = 0
flag = 0
```

End Sub

```
Private Sub funkcii_Click(Index As Integer)
Text4.Text = "00" 'сброс времени при загрузке задания
Text4.Visible = False
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text5.Text = ""
Label10.Caption = ("Тема занятия: Встроенные функции Паскаль.")
```

```

Command3.Caption = ("Начать")
Label2.Caption = ""
t = 0
i = 0
Open App.Path & "\5.txt" For Input As #1
While Not EOF(1)
i = i + 1
Input #1, z(i)
Input #1, o(i)
Input #1, p(i)
Wend
Close #1
Text5.Text = i
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True

MsgBox "Загрузка завершена. Нажмите кнопку начать", vbInformation, "Загрузка задания"

End Sub

Private Sub loch_Click(Index As Integer)
Form2.Picture = LoadPicture(App.Path & "\4.JPG")
End Sub

Private Sub mail_Click(Index As Integer)
Form2.Picture = LoadPicture(App.Path & "\1.JPG")
End Sub

Private Sub prisvoit_Click(Index As Integer)
Text4.Text = "00" 'сброс времени при загрузке задания
Text4.Visible = False
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text5.Text = ""
Label10.Caption = ("Тема занятия: Типы переменных. Оператор присвоения.")
Command3.Caption = ("Начать")
Label2.Caption = ""
t = 0
i = 0
Open App.Path & "\01.txt" For Input As #1
While Not EOF(1)
i = i + 1
Input #1, z(i)
Input #1, o(i)
Input #1, p(i)
Wend
Close #1
Text5.Text = i
Command2.Visible = True
Command3.Visible = True

```

```
Command4.Visible = True
```

```
MsgBox "Загрузка завершена. Нажмите кнопку начать", vbInformation, "Загрузка задания"
```

```
End Sub
```

```
Private Sub stroki_Click(Index As Integer)
```

```
Text4.Text = "00" 'сброс времени при загрузке задания
```

```
Text4.Visible = False
```

```
Text1.Text = ""
```

```
Text2.Text = ""
```

```
Text3.Text = ""
```

```
Text5.Text = ""
```

```
Label10.Caption = ("Тема занятия: Работа со строками.")
```

```
Command3.Caption = ("Начать")
```

```
Label2.Caption = ""
```

```
t = 0
```

```
i = 0
```

```
Open App.Path & "\8.txt" For Input As #1
```

```
While Not EOF(1)
```

```
i = i + 1
```

```
Input #1, z(i)
```

```
Input #1, o(i)
```

```
Input #1, p(i)
```

```
Wend
```

```
Close #1
```

```
Text5.Text = i
```

```
Command2.Visible = True
```

```
Command3.Visible = True
```

```
Command4.Visible = True
```

```
MsgBox "Загрузка завершена. Нажмите кнопку начать", vbInformation, "Загрузка задания"
```

```
End Sub
```

```
Private Sub tabl_Click(Index As Integer)
```

```
Form2.Picture = LoadPicture(App.Path & "\3.JPG")
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
Text4.Text = Str(Val(Text4.Text) + 1)
```

```
End Sub
```

```
Private Sub uchebnik_Click(Index As Integer)
```

```
Dim dTaskID As Double
```

```
dTaskID = Shell(App.Path & "\PascalABC.exe", vbNormalFocus)
```

```
End Sub
```

```
Private Sub vetvlenie_Click(Index As Integer)
```

```
Text4.Text = "00" 'сброс времени при загрузке задания
```

```
Text4.Visible = False
```

```

Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text5.Text = ""
Label10.Caption = ("Тема занятия: Оператор условного перехода.")
Command3.Caption = ("Начать")
Label2.Caption = ""
t = 0
i = 0
Open App.Path & "\2.txt" For Input As #1
While Not EOF(1)
i = i + 1
Input #1, z(i)
Input #1, o(i)
Input #1, p(i)
Wend
Close #1
Text5.Text = i
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True

MsgBox "Загрузка завершена. Нажмите кнопку начать", vbInformation, "Загрузка задания"

End Sub

Private Sub vvod_Click(Index As Integer)
Text4.Text = "00" 'сброс времени при загрузке задания
Text4.Visible = False
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text5.Text = ""
Label10.Caption = ("Тема занятия: Операторы ввода и вывода.")
Command3.Caption = ("Начать")
Label2.Caption = ""
t = 0
i = 0
Open App.Path & "\3.txt" For Input As #1
While Not EOF(1)
i = i + 1
Input #1, z(i)
Input #1, o(i)
Input #1, p(i)
Wend
Close #1
Text5.Text = i
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True

MsgBox "Загрузка завершена. Нажмите кнопку начать", vbInformation, "Загрузка задания"

```

End Sub

```
Private Sub vychisleniy_Click(Index As Integer)
Text4.Text = "00" 'сброс времени при загрузке задания
Text4.Visible = False
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text5.Text = ""
Label10.Caption = ("Тема занятия: Линейный вычислительный процесс.")
Command3.Caption = ("Начать")
Label2.Caption = ""
t = 0
i = 0
Open App.Path & "\6.txt" For Input As #1
While Not EOF(1)
i = i + 1
Input #1, z(i)
Input #1, o(i)
Input #1, p(i)
Wend
Close #1
Text5.Text = i
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True
```

MsgBox "Загрузка завершена. Нажмите кнопку начать", vbInformation, "Загрузка задания"

End Sub

```
Private Sub vyhod_Click(Index As Integer)
End
End Sub
```

```
Private Sub win_Click(Index As Integer)
Form2.Picture = LoadPicture(App.Path & "\6.JPG")
End Sub
```

```
Private Sub winda_Click(Index As Integer)
Form2.Picture = LoadPicture(App.Path & "\5.JPG")
End Sub
```

### ***Коды объектов формы 3***

```
Dim adrFiles As String
Dim FileName As String
```

```
Private Sub arifmetika_Click(Index As Integer)
WindowsMediaPlayer1.URL = ""
```

```
Open "C:\temp.url" For Output As #1
Print #1, "[InternetShortcut]"
Print #1, "URL=http://www.intuit.ru/department/algorithms/basicalgos/9/#video.1"
Close #1
Shell "rundll32.exe shdocvw.dll,OpenURL C:\temp.url", vbNormalFocus
```

```
End Sub
```

```
Private Sub chifra_Click(Index As Integer)
Form3.Picture = LoadPicture(App.Path & "\1234117488_8-digital-background.jpg")
```

```
End Sub
```

```
Private Sub chislovoi_Click(Index As Integer)
adrFiles = App.Path & "\лето.avi"
End Sub
```

```
Private Sub Command1_Click()
Unload Form3
Set Form3 = Nothing
Form1.Show
End Sub
```

```
Private Sub Command2_Click()
WindowsMediaPlayer1.URL = adrFiles
End Sub
```

```
Private Sub Command3_Click()
WindowsMediaPlayer1.URL = ""
End Sub
```

```
Private Sub Command4_Click()
WindowsMediaPlayer1.URL = ""
Open "C:\temp.url" For Output As #1
Print #1, "[InternetShortcut]"
Print #1, "URL=http://www.intuit.ru/user/registry/"
Close #1
Shell "rundll32.exe shdocvw.dll,OpenURL C:\temp.url", vbNormalFocus
End Sub
```

```
Private Sub Command5_Click()
WindowsMediaPlayer1.URL = ""
Open "C:\temp.url" For Output As #1
Print #1, "[InternetShortcut]"
Print #1, "URL=http://www.intuit.ru/"
Close #1
Shell "rundll32.exe shdocvw.dll,OpenURL C:\temp.url", vbNormalFocus
End Sub
```

```
Private Sub dereviy_Click(Index As Integer)
WindowsMediaPlayer1.URL = ""
Open "C:\temp.url" For Output As #1
```

```
Print #1, "[InternetShortcut]"
Print #1, "URL=http://www.intuit.ru/department/algorithms/basicalgos/6/#video.1"
Close #1
Shell "rundll32.exe shdocvw.dll,OpenURL C:\temp.url", vbNormalFocus
End Sub
```

```
Private Sub dinamicheskoe_Click(Index As Integer)
WindowsMediaPlayer1.URL = ""
Open "C:\temp.url" For Output As #1
Print #1, "[InternetShortcut]"
Print #1, "URL=http://www.intuit.ru/department/algorithms/basicalgos/3/#video.1"
Close #1
Shell "rundll32.exe shdocvw.dll,OpenURL C:\temp.url", vbNormalFocus
End Sub
```

```
Private Sub formy_Click(Index As Integer)
adrFiles = ""
WindowsMediaPlayer1.URL = ""
MsgBox "Извините ресурс в стадии разработки", vbOKOnly, "Предупреждение"
End Sub
```

```
Private Sub geometriy_Click(Index As Integer)
WindowsMediaPlayer1.URL = ""
Open "C:\temp.url" For Output As #1
Print #1, "[InternetShortcut]"
Print #1, "URL=http://www.intuit.ru/department/algorithms/basicalgos/7/#video.1"
Close #1
Shell "rundll32.exe shdocvw.dll,OpenURL C:\temp.url", vbNormalFocus
End Sub
```

```
Private Sub grafy_Click(Index As Integer)
WindowsMediaPlayer1.URL = ""
Open "C:\temp.url" For Output As #1
Print #1, "[InternetShortcut]"
Print #1, "URL=http://www.intuit.ru/department/algorithms/basicalgos/4/#video.1"
Close #1
Shell "rundll32.exe shdocvw.dll,OpenURL C:\temp.url", vbNormalFocus
End Sub
```

```
Private Sub kombinatorika_Click(Index As Integer)
WindowsMediaPlayer1.URL = ""
Open "C:\temp.url" For Output As #1
Print #1, "[InternetShortcut]"
Print #1, "URL=http://www.intuit.ru/department/algorithms/basicalgos/10/#video.1"
Close #1
Shell "rundll32.exe shdocvw.dll,OpenURL C:\temp.url", vbNormalFocus
End Sub
```

```
Private Sub konstruktor_Click(Index As Integer)
adrFiles = ""
WindowsMediaPlayer1.URL = ""
MsgBox "Извините ресурс в стадии разработки", vbOKOnly, "Предупреждение"
```

End Sub

```
Private Sub makros_Click(Index As Integer)
    adrFiles = ""
    WindowsMediaPlayer1.URL = ""
    MsgBox "Извините ресурс в стадии разработки", vbOKOnly, "Предупреждение"
End Sub
```

```
Private Sub model_Click(Index As Integer)
    adrFiles = ""
    WindowsMediaPlayer1.URL = ""
    MsgBox "Извините ресурс в стадии разработки", vbOKOnly, "Предупреждение"
End Sub
```

```
Private Sub otkrsnm_Click(Index As Integer)
    adrFiles = ""
    CommonDialog1.Filter = "Видеофайлы (*.avi) | *.avi"
    CommonDialog1.ShowOpen
    WindowsMediaPlayer1.URL = CommonDialog1.FileName
End Sub
```

```
Private Sub poiskVGrafah_Click(Index As Integer)
    WindowsMediaPlayer1.URL = ""
    Open "C:\temp.url" For Output As #1
    Print #1, "[InternetShortcut]"
    Print #1, "URL=http://www.intuit.ru/department/algorithms/basicalgos/5/#video.1"
    Close #1
    Shell "rundll32.exe shdocvw.dll,OpenURL C:\temp.url", vbNormalFocus
End Sub
```

```
Private Sub slognost_Click(Index As Integer)
    WindowsMediaPlayer1.URL = ""
    Open "C:\temp.url" For Output As #1
    Print #1, "[InternetShortcut]"
    Print #1, "URL=http://www.intuit.ru/department/algorithms/basicalgos/2/#video.1"
    Close #1
    Shell "rundll32.exe shdocvw.dll,OpenURL C:\temp.url", vbNormalFocus
End Sub
```

```
Private Sub sortirovka_Click(Index As Integer)
    WindowsMediaPlayer1.URL = ""
    Open "C:\temp.url" For Output As #1
    Print #1, "[InternetShortcut]"
    Print #1, "URL=http://www.intuit.ru/department/algorithms/basicalgos/2/#video.1"
    Close #1
    Shell "rundll32.exe shdocvw.dll,OpenURL C:\temp.url", vbNormalFocus
```

End Sub

```
Private Sub tochnost_Click(Index As Integer)
    WindowsMediaPlayer1.URL = ""
    Open "C:\temp.url" For Output As #1
```

```
Print #1, "[InternetShortcut]"
Print #1, "URL=http://www.intuit.ru/department/algorithms/basicalgos/8/#video.1"
Close #1
Shell "rundll32.exe shdocvw.dll,OpenURL C:\temp.url", vbNormalFocus
End Sub
```

```
Private Sub win7_Click()
Form3.Picture = LoadPicture(App.Path & "\6.JPG")
End Sub
```

```
Private Sub zaprosy_Click(Index As Integer)
adrFiles = ""
WindowsMediaPlayer1.URL = ""
MsgBox "Извините ресурс в стадии разработки", vbOKOnly, "Предупреждение"
End Sub
```

#### ***Код объектов формы 4***

```
Private Sub Command1_Click()
Unload Form4
Set Form4 = Nothing
Form1.Show
End Sub
```

```
Private Sub List1_Click()
Dim i As Byte
Dim dTaskID As Double
flag = MsgBox(List1.List(List1.ListIndex), vbOKCancel, "Выбор демонстрации")
i = List1.ItemData(List1.ListIndex)
Select Case i
Case 0
dTaskID = Shell(App.Path & "\0.exe", vbNormalFocus)
Case 1
dTaskID = Shell(App.Path & "\1.exe", vbNormalFocus)
Case 2
dTaskID = Shell(App.Path & "\2.exe", vbNormalFocus)
Case 3
dTaskID = Shell(App.Path & "\3.exe", vbNormalFocus)
Case 4
dTaskID = Shell(App.Path & "\4.exe", vbNormalFocus)
Case 5
dTaskID = Shell(App.Path & "\5.exe", vbNormalFocus)
Case 6
dTaskID = Shell(App.Path & "\6.exe", vbNormalFocus)
Case 7
dTaskID = Shell(App.Path & "\7.exe", vbNormalFocus)
Case 8
dTaskID = Shell(App.Path & "\8.exe", vbNormalFocus)
Case 9
dTaskID = Shell(App.Path & "\9.exe", vbNormalFocus)
```

Case 10  
dTaskID = Shell(App.Path & "\10.exe", vbNormalFocus)  
Case 11  
dTaskID = Shell(App.Path & "\11.exe", vbNormalFocus)  
Case 12  
dTaskID = Shell(App.Path & "\12.exe", vbNormalFocus)  
Case 13  
dTaskID = Shell(App.Path & "\13.exe", vbNormalFocus)  
Case 14  
dTaskID = Shell(App.Path & "\14.exe", vbNormalFocus)  
Case 15  
dTaskID = Shell(App.Path & "\15.exe", vbNormalFocus)  
Case 16  
dTaskID = Shell(App.Path & "\16.exe", vbNormalFocus)  
Case 17  
dTaskID = Shell(App.Path & "\17.exe", vbNormalFocus)  
Case 18  
dTaskID = Shell(App.Path & "\18.exe", vbNormalFocus)  
Case 19  
dTaskID = Shell(App.Path & "\19.exe", vbNormalFocus)  
Case 20  
dTaskID = Shell(App.Path & "\20.exe", vbNormalFocus)  
Case 21  
dTaskID = Shell(App.Path & "\21.exe", vbNormalFocus)  
Case 22  
dTaskID = Shell(App.Path & "\22.exe", vbNormalFocus)  
Case 23  
dTaskID = Shell(App.Path & "\23.exe", vbNormalFocus)  
Case 24  
dTaskID = Shell(App.Path & "\24.exe", vbNormalFocus)  
Case 25  
dTaskID = Shell(App.Path & "\25.exe", vbNormalFocus)  
Case 26  
dTaskID = Shell(App.Path & "\26.exe", vbNormalFocus)  
Case 27  
dTaskID = Shell(App.Path & "\27.exe", vbNormalFocus)  
Case 28  
dTaskID = Shell(App.Path & "\28.exe", vbNormalFocus)  
Case 29  
dTaskID = Shell(App.Path & "\29.exe", vbNormalFocus)  
Case 30  
dTaskID = Shell(App.Path & "\30.exe", vbNormalFocus)  
Case 31  
dTaskID = Shell(App.Path & "\31.exe", vbNormalFocus)  
Case 32  
dTaskID = Shell(App.Path & "\32.exe", vbNormalFocus)  
Case 33  
dTaskID = Shell(App.Path & "\33.exe", vbNormalFocus)  
Case 34  
dTaskID = Shell(App.Path & "\34.exe", vbNormalFocus)  
Case 35  
dTaskID = Shell(App.Path & "\35.exe", vbNormalFocus)

```
Case 36
  dTaskID = Shell(App.Path & "\36.exe", vbNormalFocus)
Case 37
  dTaskID = Shell(App.Path & "\37.exe", vbNormalFocus)
Case 38
  dTaskID = Shell(App.Path & "\38.exe", vbNormalFocus)
Case 39
  dTaskID = Shell(App.Path & "\39.exe", vbNormalFocus)
Case 40
  dTaskID = Shell(App.Path & "\40.exe", vbNormalFocus)
Case 41
  dTaskID = Shell(App.Path & "\41.exe", vbNormalFocus)
Case 42
  dTaskID = Shell(App.Path & "\42.exe", vbNormalFocus)
Case 43
  dTaskID = Shell(App.Path & "\43.exe", vbNormalFocus)
Case 44
  dTaskID = Shell(App.Path & "\44.exe", vbNormalFocus)
Case 45
  dTaskID = Shell(App.Path & "\45.exe", vbNormalFocus)
Case 46
  dTaskID = Shell(App.Path & "\46.exe", vbNormalFocus)
Case 47
  dTaskID = Shell(App.Path & "\47.exe", vbNormalFocus)
Case 48
  dTaskID = Shell(App.Path & "\48.exe", vbNormalFocus)
Case 49
  dTaskID = Shell(App.Path & "\49.exe", vbNormalFocus)
Case 50
  dTaskID = Shell(App.Path & "\50.exe", vbNormalFocus)
Case 51
  dTaskID = Shell(App.Path & "\51.exe", vbNormalFocus)
Case 52
  dTaskID = Shell(App.Path & "\52.exe", vbNormalFocus)
Case 53
  dTaskID = Shell(App.Path & "\53.exe", vbNormalFocus)
Case 54
  dTaskID = Shell(App.Path & "\54.exe", vbNormalFocus)
Case 55
  dTaskID = Shell(App.Path & "\55.exe", vbNormalFocus)
Case 56
  dTaskID = Shell(App.Path & "\56.exe", vbNormalFocus)
Case 57
  dTaskID = Shell(App.Path & "\57.exe", vbNormalFocus)
Case 58
  dTaskID = Shell(App.Path & "\58.exe", vbNormalFocus)
End Select
End Sub
```