

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ

**ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ АРМ ДИРЕКТОРА ЦЕНТРА
КОМПЬЮТЕРНОГО ОБУЧЕНИЯ**

Выпускная квалификационная работа
обучающейся по направлению подготовки 02.03.01
Математика и компьютерные науки
очной формы обучения, группы 07001303
Козлова Данила Александровича

Научный руководитель
к.т.н., доцент
Е.В. Бурданова

БЕЛГОРОД 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. АНАЛИЗ ПРОБЛЕМЫ И ПОСТАНОВКА ЗАДАЧИ.....	7
1.1 Разновидности компьютерных средств обучения и их возможности.....	7
1.2 Понятие автоматизированного рабочего места.....	16
1.3 Основные формы электронных образовательных ресурсов	19
1.4 Требования к автоматизированному рабочему месту	21
1.5 Структура автоматизированного рабочего места	25
2. РАЗРАБОТКА ЭЛЕКТРОННОГО ОБРАЗОВАТЕЛЬНОГО РЕСУРСА ..	28
2.1 Постановка задачи.....	28
2.2 Средства информационных технологий для проектирования автоматизированного рабочего места	29
2.3 Разработка структуры приложения	31
3. РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПО.....	35
3.1 Разработка пользовательского интерфейса.....	35
3.2 Разработка приложения.....	43
3.3 Тестирование	49
ЗАКЛЮЧЕНИЕ	52
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	55
ПРИЛОЖЕНИЕ	Ошибка! Закладка не определена.

ВВЕДЕНИЕ

Современные компьютерные обучающие системы разрабатываются с помощью мультимедиа-технологии. Данная технология возникла на стыке многих отраслей знания. Для ее успешного использования и разработки продуктов в ее среде нужна соответствующая программно-техническая платформа.

Интенсивное развитие мультимедиа-технологии началось в середине 80-х годов. Она применяется в различных сферах:

- развлечений (компьютерные игры, виртуальная реальность);
- рекламы (презентации, рекламные фильмы);
- телекоммуникаций (домашние страницы в World Wide Web);
- информационных системах (мультимедийные каталоги, архивы, справочники);
- моделировании (тренажеры);
- обучении.

Разработка обучающих курсов в среде мультимедиа (мультимедиа-курсов) является длительным и дорогостоящим процессом, поэтому важно хорошо представлять себе все основные этапы создания курса и возможные принимаемые на каждом этапе решения [1].

Обучение с помощью компьютера дает возможность организовать самостоятельные действия каждого ученика.

Как показывает российская практика, использование информационных технологий и сети Интернет в общеобразовательных школах и педвузах пока еще ограничено и слабо увязано с учебным процессом. В настоящее время это, как правило, занятия по информатике. Дистанционное обучение школьников находится в зародышевом состоянии. Применение ПК в

обучении информатики и компьютерным наукам значительно облегчает работу как ученика так и учителя.

В своей работе мы рассмотрим, что ПК может быть использован как вспомогательное средство, как любое другое техническое средство обучения или учебник.

Когда говорят о достоинствах компьютера в обучении, обычно имеют в виду, прежде всего дисплей (схемы, графики, чертежи, рисунки, движущиеся изображения в цвете и со звуковым сопровождением, причем эти изображения может создавать и сам учащийся). Часто указывают на возможность вести содержательную беседу, диалог с компьютером, причем учащийся не только отвечает на вопросы электронного педагога, но и сам может их ставить и даже вступать с компьютером в спор. Одно из наиболее плодотворных применений компьютера в обучении - использование его как средство управления учебной деятельностью. Именно в этом качестве он может наиболее существенно повысить эффективность обучения.

Чтобы эффективно использовать компьютер в учебном процессе, необходимо решить множество проблем, в первую очередь психолого-педагогических.

Научно-педагогическими предпосылками повышения качества обученности с помощью компьютерных программ являются результаты психолого-педагогических исследований.

Информационная система – это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели. Современное понимание информационной системы предполагает использование в качестве основного технического средства переработки информации компьютера. Кроме того, техническое воплощение информационной системы само по себе ничего не будет значить, если не учтена роль человека, для

которого предназначена производимая информация и без которого невозможно ее получение и представление.

Необходимо понимать разницу между компьютерами и информационными системами. Компьютеры, оснащенные специализированными программными средствами, являются технической базой и инструментом для информационных систем. Информационная система немыслима без персонала, взаимодействующего с компьютерами и телекоммуникациями.

В нормативно-правовом смысле информационная система определяется как «организационно упорядоченная совокупность документов (массив документов) и информационных технологий, в том числе и с использованием средств вычислительной техники и связи, реализующих информационные процессы» [2].

В данной работе анализируются обучаемые информационные системы.

Исходя из вышесказанного, мной была выбрана тема выпускной квалификационной работы «Проектирование и реализация Автоматизированного рабочего места директора Центра компьютерного обучения.

Цель: Создать автоматизированное рабочее место.

Задачи:

1. Анализа литературы по разработке АРМ
2. Раскрыть возможность использования АРМ в качестве средства обучения
3. Изучить средства информационных технологий для проектирования компьютерных программ учебного назначения и обосновать выбор средств, для разработки собственной компьютерной программы АРМ учебного назначения.
4. Разработать программу автоматизированного рабочего места
5. Экспериментально апробировать АРМ.

База исследования: Центр компьютерного обучения НИУ «БелГУ».

В дипломной работе содержится 58 страниц без приложения, 14 рисунков. В процессе создания было использовано 22 литературных источника.

1. АНАЛИЗ ПРОБЛЕМЫ И ПОСТАНОВКА ЗАДАЧИ

1.1 Разновидности компьютерных средств обучения и их возможности

Компьютерные обучающие и контролирующие программы – программные средства учебного назначения, которые широко используются в образовательном процессе и позволяют:

- индивидуализировать подход и дифференцировать процесс обучения;
- контролировать обучаемого с диагностикой ошибок и обратной связью;
- обеспечить самоконтроль и самокоррекцию учебно-познавательной деятельности;
- моделировать и имитировать процессы и явления;
- проводить лабораторные работы, эксперименты и опыты в условиях виртуальной реальности;
- повысить интерес к процессу обучения, используя игровые ситуации и многое другое [8].

С.А. Христочевский в своём учебнике «Электронные мультимедийные учебники и энциклопедии» пишет, что наше время все чаще в образовании применяются компьютеры, обеспечивающие приспособление процесса обучения к индивидуальным характеристикам обучаемых, разгружающие преподавателей от трудоемких и повторяющихся операций по представлению учебной информации и контролю знаний. При проведении

занятий в компьютерных классах возрастают активность обучаемых, самостоятельно изучающих большой объем учебной информации, и возможности преподавателя по управлению учебной деятельностью группы обучаемых. Например, с помощью компьютера можно иллюстрировать динамические процессы, которые с помощью обычных учебных средств осваиваются достаточно тяжело. Уже сейчас создаются виртуальные кафедры и виртуальные институты, обучаться в которых могут люди со всего мира. Созданы обучающие системы по самым различным учебным курсам, причем не только по точным, но и по гуманитарным дисциплинам.

Достоинствами компьютерных учебных программ можно считать [19]:

- 1) возможность почти полностью перевести курс обучения по какой-либо дисциплине на компьютер (лекции, практики, контроль знаний и умений);
- 2) избавление обучаемых от поиска и приобретения книг;
- 3) возрастание активности обучаемых, самостоятельно прорабатывающих большой объем учебной информации;
- 4) разгрузка преподавателей от ряда трудоемких и часто повторяющихся операций по представлению учебной информации и контроля знаний;
- 5) возможность оперативно изменять учебный материал;
- 6) возможность изучения учебного материала и выполнения практических работ дома;
- 7) сокращение времени выработки у обучаемых необходимых навыков;
- 8) адаптация к темпу работы обучаемого, облегчение поиска необходимой информации;
- 9) возможность моделировать на экране компьютера сложных процессов и явлений, создавать игровые познавательные ситуации;

10) возможность расширить обычный учебник, используя аудио– и видеовставки;

11) руководство школы или ВУЗа имеет возможность быстро просматривать результаты контроля усвоения учебного материала по самым разным критериям (по группам, по специальностям, по отдельным обучаемым и т.д.).

Однако следует заметить, что компьютерные программы в обучении не заменяет преподавателя, а является его помощником. Аналогично – компьютерный учебник не заменяет обычный учебник, а дополняет его, как средство педагогических приемов и методов, поэтому мы должны учитывать то, что использование учебных компьютерных программ при формировании темы «Алгоритм» способствует:

- развитию структурированного и наглядно-образного мышления у школьников;
- развитию умений построения логических цепочек и алгоритмов с помощью компьютерных учебных программ;
- создание условия для реализации у школьников выбора наиболее эффективных способов применения алгоритмов в зависимости от конкретных задач, применяемых в данной учебной компьютерной программе;
- реализации интеллектуального потенциала у школьников за счет применения учебной компьютерной программы.[1]

Это позволит педагогу сформировать у учащихся понятия по теме «Алгоритм», а так же практически закрепить полученные знания путем применения как педагогических так информационных технологий.

Рассмотрим один из возможных подходов к классификации компьютерных средств обучения.

Электронный ресурс можно рассматривать как набор текстовой, графической, цифровой, аудиоинформации, видеоинформации, фото и

другой информации, в электронном виде, записанной на носителе – либо размещённой в компьютерной сети,.

Электронные издания могут быть электронными справочниками, электронными словарями, энциклопедиями, электронные путеводители учебниками и так далее.

Рассмотрим электронные учебники как вид электронных ресурсов и программно–информационную систему, реализованную на ЭВМ, созданную для учебной деятельности. Информация структурирована и имеет в наличии функции упражнений для ее осмысления и закрепления.

Электронный учебник – как и традиционный учебник, содержит теоретический материал по определенному предмету и примеры (например, примеры решения задач).

Обучающая система – это человеко–машинный комплекс, работающий в диалоговом режиме и предназначенный для управления познавательной деятельностью. Обучающая система – более широкое понятие, чем электронный учебник, так как она должна обучать. Она должна включать в себя теоретический материал с примерами, а также средства для выработки практических навыков у обучаемых и средства контроля приобретенных знаний, умений и навыков.

Основное назначение обучения, а, следовательно, и обучающей системы, – овладение умениями, а не знаниями. Механизмом осуществления деятельности является решение задач. Следовательно, основная часть обучающей системы – тренирующая.

Интеллектуальная (адаптивная) обучающая система – обучающая система с элементами искусственного интеллекта. Такая обучающая система позволяет не просто тренировать обучаемого и контролировать его знания, но и по результатам деятельности обучаемого может определить, какие знания недостаточны или ошибочны и вернуть обучаемого на

соответствующий раздел теории или практики, либо дать дополнительные разъяснения.

Обзор имеющихся адаптивных средств, представлен в учебнике Брусиловского П.Л. Адаптивные обучающие системы в World Wide Web.

Дистанционная обучающая система – обучающая система, которая поддерживает удаленную работу через сеть. Таким образом, преподаватель и обучаемый разделены в пространстве и во времени: обучаемый занимается на своем компьютере, а преподаватель контролирует его деятельность на своем. Учебный материал, тесты, задачи и результаты обучения хранятся на сервере сети [6].

Однако при внедрении дистанционной обучающей системы возникают следующие проблемы [5, 4]:

- отсутствие реального доступа у массового пользователя;
- низкое качество связи, малая скорость и ненадежность связи, особенно на старых АТС и в отдаленных населенных пунктах;
- неверие и незнание возможностей всемирной сети Интернет.

Гипермедийная обучающая система – обучающая система, основывающаяся на использовании гипертекста для представления теоретического материала. Применение гипертекста позволяет объединять различные способы представления информации (текст, изображения, звук, видео и т.д.), легко связывать различные материалы между собой. Однако, обучаемый, переходя по ссылкам от одного документа к другому, может легко “потеряться” и забыть, откуда он пришел и с чего начинал обучение. Это явление называется эффектом “потери в гиперпространстве” [4]. Чтобы избежать этого эффекта, применяются способы возврата обучаемого к исходному пункту обучения.

Учебный курс – еще более широкое понятие, чем обучающая система. Если обучающая система проектируется для конкретной предметной области, то учебный курс может включать в себя несколько обучающих

систем по различным предметам (которые, в частности, могут быть связаны между собой), а обучаемый может изучать любой из предложенных предметов (какие конкретно предметы изучать выбирает он сам или это делает преподаватель).

Учебные курсы могут быть и адаптивными, и дистанционными, и гипермедийными.

Итак, создание обучающих систем на базе ЭВМ – это один из перспективных способов повышения эффективности процесса обучения.

Объектом управления при обучении является обучаемый. Субъектами управления выступают преподаватель и обучающая система, так как она не заменяет, а дополняет преподавателя.

Целью работы обучаемого с обучающей системой является получение:

- знаний по конкретной предметной области;
- умений применять различные методы и алгоритмы;
- навыков решения задач;
- оценки приобретенных знаний, умений и навыков.

Исходя из этого, учебная программа должна включать в себя 3 основные части [22]:

- теоретическую;
- тренирующую;
- контролирующую.

Система формирования задания предназначена для генерации задач в соответствии со знаниями обучаемого. Сгенерированное задание поступает обучаемому и на систему диагностики.

Система диагностики предназначена для контроля знаний обучаемого. Она осуществляет ввод ответа, сравнивает его с правильным ответом и принимает решение о правильности выполнения задания.

Система модели обучения принимает информацию о результатах обучения и принимает решение о продолжении обучения.

В интеллектуальных обучающих системах управление обучением определяется самой обучающей системой на основании результатов обучения.

Одной из основных функций учебной программы является управление познавательной деятельностью обучаемого. Для этого программа должна получать сведения о ходе процесса обучения, об усвоении обучаемым учебного материала, о результатах тестирования и выполнения практических заданий. То есть в системе “учебная программа – обучаемый” должна присутствовать обратная связь [6].

Обратная связь может быть двух видов: внутренняя и внешняя.

Внутренняя обратная связь – это информация, которая поступает от обучающей программы к обучаемому в ответ на его действия при выполнении заданий. Она предназначена для самокоррекции обучаемым своей учебной деятельности и дает возможность обучаемому сделать осознанный вывод об успешности или ошибочности учебной деятельности. Она является стимулом к дальнейшим действиям, помогает оценить и скорректировать результаты учебной деятельности. Различают консультирующую и результативную внутреннюю обратную связь. Консультация может быть разной: помощь, разъяснение, подсказка, наталкивание и так далее. Результативная обратная связь также может быть различной: от "верно – неверно" до демонстрации правильного результата или способа действия.

Внешняя обратная связь – это информация, которая поступает от обучающей программы к преподавателю. С ее помощью он корректирует учебную деятельность обучаемого, а также может скорректировать сценарий обучения.

Программа получает информацию о ходе обучения, анализирует ее, выставляет оценку обучаемому и принимает решение о дальнейшем ходе обучения.

В результате у обучаемого вырабатываются определенные знания, умения и навыки, которые будем называть результатом обучения.

Также в программу входит адаптирующая часть. Ее задача состоит в коррекции хода обучения, т.е. настройке его под конкретного обучаемого. Она получает информацию о цели и результате обучения, сравнивает их и корректирует работу обучающей программы и ход познавательной деятельности обучаемого.

Таким образом, можно сказать, что результат обучения есть функция от цели обучения и коррекции.

В этой схеме главное место занимает преподаватель. Он ставит цель обучения, получает информацию о ходе обучения и может вносить дополнительную коррекцию в работу программы и в ход познавательной деятельности обучаемого наравне с адаптирующей частью.

В современных электронно-вычислительных машинах программы решают самые различные задачи по содержанию и по народнохозяйственному значению В научно-исследовательских институтах и вузах во многих случаях программы создаются в единственном экземпляре для решения частных исследовательских задач, для ускорения вычислений, моделирования процессов, обработки экспериментального материала и т.д. Такие программы не имеют массового применения и доступны для использования только тем, кто их разработал. Они являются объектами научно-технического творчества и редко становятся промышленными изделиями [13].

Таким образом, проанализировав литературу по интересующей нас проблеме, мы **условия** отсутствие **необходимые** для **реализации автоматизированных рабочих мест** один из возможных подходов к классификации компьютерных средств обучения, который включает в себя:

- электронное издание;
- электронный учебник;

- обучающая система;
- интеллектуальная (адаптивная) обучающая система;
- дистанционная обучающая система ;
- гипермедийная обучающая система;
- учебный курс.

При создании учебной компьютерной программы мы будем сочетать виды электронного учебника и обучающей системы для эффективности процесса обучения по теме «алгоритм» у школьников. В структуру данной учебной компьютерной программы будут включены составные части:

- теоретическая;
- тренирующая;
- контролирующая.

Данные условия создания обучающей компьютерной программы соответствуют, как и целям, так и задачам изучаемой проблемы.

Общеобразовательные и дополнительные курсы должны учитывать особенности обучения, связанные с различным уровнем общей подготовки обучаемых (необходима оценка имеющихся знаний и подстройка системы для оптимального изложения), уровнем компьютерных знаний (необходим простой и понятный интерфейс). Важно обеспечить такое согласование тем обучения, чтобы изучение одной темы не шло в ущерб другой и не опережало изучения параллельных тем.

Курсы специального образования должны учитывать уровень подготовки, давать возможность не повторять уже известные темы, обеспечивать наличие самой последней информации в данной предметной области.

После того, как выбраны курс и аудитория, для которой он адресован, определяются цели обучения и степень сложности материала, который будет представлен в обучающей системе [1].

1.2 Понятие автоматизированного рабочего места

Автоматизированное рабочее место (АРМ) описывается как множества информационных, программных, технических ресурсов, которые дают пользователю возможность работы с данными и автоматизируют администрирование и управление в определённой предметной области.

Создание автоматизированных рабочих мест упрощает функции накопления, хранения переработки информации, и эти процессы выполняются компьютером, а пользователь (преподаватель) выполняет лишь некоторые простые операции. Персональные компьютеры дают пользователям возможность контролировать процесс обучения, просмотра конкретных результатов, изменения значений конкретных данных, таких как результаты тестирования, или ввода исходных данных в программу.

Автоматизированное рабочее место как средство для оптимизации и администрирования образовательной деятельности реализуется для обеспечения выполнения некоторой группы функций. Минимальной функцией автоматизированного рабочего места - это информационно-справочное обслуживание. Данный функционал свойственен многим АРМ, особенности ее создания зависят от категории пользователя.

АРМ создан для профессиональных сфер деятельности конкретным предметным областям. Автоматизированные рабочие места в профессиональной деятельности будут основным принципом взаимодействия с электронным ресурсом, играя роль автономных рабочих мест, терминалов специализированных больших ЭВМ, рабочих станций в сети. АРМ имеют доступную архитектуру и легко адаптируются в различных областях.

Локально АРМ позволяет оперативно обрабатывать информацию сразу после её введения в программу, обработанная информация хранится длительное время.

Внедряя процессы администрирования, цель АРМ интегрирование управления, рабочее место должно работать в многофункциональном режиме.

АРМ, созданное для персонального компьютера, — самый допустимый реализуемый вариант автоматизированного рабочего места для работников сферы образования. Такое АРМ является системой, предоставляющей в интерактивном режиме работы, доступ конкретному преподавателю (пользователю). Реализация этого компонента АРМ, для внутреннего пользования, должен находиться в личном распоряжении конкретного пользователя АРМ. Пользователь сам администрирует приложение и преобразовывает информацию.

Создание АРМ для персональных компьютеров характеризует:

- простоту, удобство и дружелюбность по отношению к пользователю;
- адаптировано к определённым функциям пользователя;
- компактно для размещения, имеет не большие требования к условиям эксплуатации;
- достаточно надёжно;
- имеет не сложную организацию тех обслуживания.

Работа среды в рамках персонального компьютера в качестве рабочей станции.

Выбор его конфигурации АРМ образовательного процесса носит определённый характер, выбранной специализации, целями, объемами работы. Но различные конфигурации АРМ должны удовлетворят требованиям реализации для информационного, технического, программного обеспечения.

Создание АРМ в диктуется условиями технических эксплуатационных характеристик компьютера, на которых они основываются. При создании АРМ ставятся условия к необходимым

техническим средствам обработки и вывода информации, набору комплектующих модулей, сетевым интерфейсам, эргономическим параметрам устройств и т.д.

Техническое обеспечение АРМ должно представлять высокую надежность технических средств, предоставления удобных пользователю режимов работы, таких как (автономный, с распределенной БД, информационный, с техникой верхних уровней и т.д., возможность проработать в определенное время нужный объем данных. Так как АРМ является индивидуальным пользовательским средством, оно должно обеспечивать высокие эргономические свойства и удобство обслуживания.

Информационное обеспечение АРМ сводится к одной, предметной области. Обработка материала представляет собой такое структурированное представление материала, дающим осуществлять необходимую обработку разными структурами, простую и быструю корректировку данных в массивах.

Программное обеспечение зависит от профессиональной направленности пользователя, его потребностями, квалификацией и специализацией. Пользователь, в программной среде должен чувствовать поддержки деятельности в разных режимах активно либо пассивно. Таким образом, при взаимодействии дано максимальное обеспечение удобств работы пользователя за счет развития программных средств.

Рассмотрим основные группы программных средств, применяемых для автоматизации наиболее часто встречающихся функций, выполняемых преподавателями. К подобным программным средствам относят следующие:

- системы подготовки текстовых документов (текстовые редакторы и настольные издательские системы);
- обработку информации ос обучающихся (табличные редакторы и другие подобные программы);
- системы управления базами данных;

- системы контроля проектирования;
- системы для личных нужд;
- администрирование электронной почты;
- системы интеллектуального проектирования и совершенствования систем управления;
- обработки изображений;
- системы сканирования символов;
- системы администрирования, документооборота, электронного документооборота.

Применение технологий обработки информации и разных технических средств представляет ряд требований к программному обеспечению АРМ, такие как надежность работы, эффективность использования ресурсов, построения модулей, удобный интерфейс. Программные средства оптимизируются с точки зрения удобства для обучения и администрирования обучения.

Программное обеспечение АРМ должно также обладать свойством адаптивности и возможностями настройки на применение конкретным пользователем для решения своих задач.

1.3 Основные формы электронных образовательных ресурсов

Реализуя любые компьютерные информационные системы, при подготовке электронного учебника автор руководствуется своим воображением и представлением о программе, но существуют определённые формы электронных ресурсов, отдельных их частей, формирующих учебник.

Тест – наиболее простая форма электронного ресурса. Самая длительная часть работы над тестом – заполнение его информацией, вопросами, а также толкование ответов на вопросы. Хороший тест может

дать объективно представление о знаниях, умениях и навыках, которыми владеет обучающийся в данной предметной области.

Энциклопедия –форма электронного ресурса, при которой в программе, информация отличается большей полнотой, нежели стандарты образования.

Ещё один вид ресурсов - задачник, в электронном учебнике ученик решает и разбирает конкретную программу, сопровождается теоретическим материалом, помощью. Должны содержаться задачи по всем разделам теории. Можно разобрать множество конкретных задач.

Современные электронные ресурсы имеют творческий подход, давать возможность учащимся проявлять свои способности к творчеству и развивать их. Творческая деятельность, содействует выработыванию и развитию определённых навыков и умений у учащегося. Креативная среда может быть коллективной работой обучающихся и преподавателя при работе с проектами.

Электронный ресурс может адаптироваться к учебному процессу, Среда будет поддерживать дополнительные материалы в электронном ресурсе и всегда можно будет расширить круг задач и теоретического материала, различных пособий. С помощью такого ресурса можно проектировать самостоятельный электронный учебник.

Электронные ресурсы излагают теорию в основном текстовом и графическом виде, что произошло от различных изданий. Электронный ресурс можно спроектировать более интерактивным, и вести активное взаимодействие с пользователем.

Эти многие другие формы электронных учебных ресурсов проектируются в виде, как отдельных электронных средств, так и множества, в зависимости от идеи автора. Автор же проектирующий электронный ресурс, должен знать требования к ресурсам и особенностям обучающихся, для которых он проектирует приложение.

1.4 Требования к автоматизированному рабочему месту

Электронные издания имеют собственные особенности.

В электронном учебнике информация может быть представлена на форме, или всплывающих окон, где продемонстрированы изображения, определения, ссылки, сноски, комментарии.

Чтобы лучше воспринять и запомнить материал можно воспользоваться техническими возможностями, использовать анимацию, аудио, изображения. Специальные модули могут выводить различные физические и технические процессы.

Главы в электронном ресурсе предлагается сделать небольшие, нежели в научной литературе, их можно разделить на отдельные элементы, включающие в себя конкретный вопрос по какому-либо разделу. Как правило, такой фрагмент состоит из двух-трех абзацев.

Содержание глав, должно быть доступно пользователю, для того, чтобы была возможность, не перелистывая страниц, ссылаться к необходимому разделу. Таким же образом, пользователь мог иметь возможность вернуться к исходной главе, либо к другому разделу.

В программе, для удобства пользователя, ключевые слова, термины, имена, должны быть гиперссылками либо всплывающими комментариями, с объяснением смысла этих понятий.

Для улучшения удобства ориентации в электронном ресурсе, можно использовать колонтитулы или заголовки.

В процессе проектирования электронного ресурса нужно принять, что множества методов, приемов, представление графической, аудио, видео, иной информации с учетом воспроизведения электронных ресурсов с дискового носителя, сети или из образовательных сайтов интернета. Нужно понимать, что во всех случаях слишком объёмный текст, лучше размещать в

традиционных бумажных изданиях. Представлять рекомендуется как можно меньше информации в виде текста.

При разработке электронных ресурсов необходимо:

- обращать внимание на современные формы обучения, совмещая при этом стандартные учебные ресурсы, на основании документов, регламентирующих содержание образования;

- создавать программы согласно возрастным особенностям обучающихся;

- максимально применять преимущества аудио и визуального вывода учебных материалов, элементов, объектов, явлений, процессов;

- применять средства компьютерного моделирования по определённой теме обучения, и моделей естественной окружающей среды и взаимодействия с ней ученика.

Дидактические требования к электронным изданиям.

Принцип научности должен давать необходимую глубину, научную достоверность представления знаний в программе, учитывая новейшие научные достижения. Закрепление знаний с помощью электронного ресурса должен основываться на современных средствах научного познания, таких как метод эксперимента, сравнения, наблюдения, абстрагирования, обобщения, анализа и синтеза, создания моделей, также математического, и системного подхода.

Доступность предполагает необходимость сопоставления степени сложности соответствуя возрастным и индивидуальным особенностям обучающихся. Слишком сложный и перегруженный материал, будет плохо восприниматься учениками.

Степень учебной деятельности определяется как требование к принципу проблемности. Обучающийся должен разрешить проблемную ситуацию, что активизирует его мысленную активность, уровень которой

будет возрастать, а при работе с электронными ресурсами в связке с традиционными методами, уровень же активности, значительно повышается.

Требования наглядности в процессе обучения, это чувственное восприятие изучаемых процессов и объектов. В электронных ресурсах наглядность можно применить как никогда, кстати, различные визуальные средства.

Также требования ставятся к самостоятельности выполнения учащимися заданий и изучения материала программы. В программе должна быть упорядоченная модель поведения и деятельности обучаемого. У программы, должна отслеживаться модель деятельности пользователя – обучаемого. Мотивы его деятельности должны соответствовать содержанию учебного материала. Повышая активность пользователя, в процессе обучения нужно применять электронном ресурсе вопросы по изучаемой теме, давать выбор процесса усвоения материала, возможность управлять ходом событий.

Требования системности даёт возможность учащимся осваивать определенные знания и систему знаний в конкретной предметной области. Знания, умения и навыки должны формироваться в конкретной системе, в определённом последовательном порядке и могли применять в жизненных ситуациях. Для этого необходимо:

- давать систематизированную теорию и структурировать её;
- брать в расчёт как старые, так и новые знания и практические способности организуя информацию;
- внимательно выбирать и структурировать учебный материал и влияние его на усвоение, оценивать каждый шаг по отношению к обучающемуся;
- процесс освоения знаний в последовательности, определяется логикой обучения;

– в электронных ресурсах связывать информацию с практическими навыками, связывать знания с личным опытом обучающегося, показывать примеры, в том числе и игровые примеры, показывать примеры практических навыков, эксперименты, модели реальных процессов и объектов.

Требования к электронным ресурсам в отдельных видах учебного процесса.

Электронные ресурсы, применяемые на занятиях, дают преподавателю различные визуальные и мультимедийные средства, которые показывают и объясняют не простые явления и процессы.

Электронные ресурсы, используемые на практике, могут содержать материал, предоставляющий возможность для индивидуальной работы ученика. Это могут быть и моделирующие моменты, виртуально создающие лаборатории, которые дают возможность понять явления или процессы в разных масштабах времени. Электронные ресурсы, используемые на практике, могут показывать имеющиеся автоматизированные средства контроля знаний, умений и навыков обучающихся.

Электронные ресурсы, применяемые на практике, дают ученику информацию занятии, порядке работы и т.д., позволяют проверять знания обучающегося, давать о информации пользователю о правильности ответа, показывать необходимый теоретический материал, методику решения задач, а также оценивать знания обучаемых, реализовывать обратную связь с преподавателем.

Содержание и структура электронных ресурсов, используемых в процессе практики, удовлетворяют требованиям дополнительной образовательной программы по конкретной теме, ориентируясь на детальное изучение материала. Электронные ресурсы должны иметь достаточно подробную систему помощи, комментария, всплывающих сообщений и т.д.

1.5 Структура автоматизированного рабочего места

Структура ресурса реализуется тем, что, электронные ресурсы используются для организации самостоятельной работы обучаемых и должны выделять, разделы и их последовательность должны быть изучены и взаимосвязаны между собой. Последовательности изучаемого материала: теоретическая часть, практическая, контрольные задания, демонстрации и материалы для дополнительного образования.

Любое электронное учебное пособие должно включать в себя следующие обязательные компоненты:

- средства изучения теоретических основ дисциплины;
- средства поддержки практических занятий;
- средства контроля знаний;
- средства взаимодействия между учителем и учащимися;
- методические рекомендации по изучению дисциплины;
- средства управления процессом изучения дисциплины.

Электронный ресурс пособие должно отвечать следующим требованиям:

- четкая структуризация предметного материала;
- наличие рекомендаций по изучению дисциплины;
- компактность представленного информационного материала;
- графическое оформление и наличие иллюстративного материала;
- включение промежуточного и текущего контроля знаний.

Электронные издания рассчитаны, прежде всего, на заочную форму обучения. Поэтому оно, кроме основного текста, содержит справочный материал, необходимый для самостоятельной работы. Для такого издания важно наличие гиперссылок, включающие в себя дополнительный текст, указатели, списки определений, мультимедиа.

Схема структуры выглядит примерно следующим образом:

Автор – Оглавление – Аннотация, – Атрибуты поиска в сети Интернет – Предисловие – Введение – Основной текст – Заключение – Справочный аппарат – Основа издания – Основные понятия – Вопросы – Тесты – Дидактический аппарат (Гиперссылки) – Приложения – Примечания – Комментарии – Дополнительный текст (Гиперссылки) – Глоссарий – Указатели – Библиографический список – Справочный аппарат (Гиперссылки) – Оглавление – Поисковый аппарат издания (Гиперссылки)

В Электронное издание могут входить следующие элементы:

1 Атрибуты поиска в сети Интернет.

2 Знак авторского права.

3 Основа издания.

4 Дополнительный текст.

5 Справочный аппарат

6 Дидактический аппарат.

7 Поисковый аппарат издания.

8 Иллюстрации.

9 Анимация.

10 Звук

11 Видео.

Электронный образовательный ресурс?

– упрощает понимание рассматриваемого материала за счет электронных ресурсов, аналогов стандартной учебной литературы, способов подачи материала: индуктивный подход, воздействие на слуховую и эмоциональную память и т.п.;

– адаптируется к потребностям учащегося, уровнем его знаний, интеллектуальными способностями;

– освобождает от громоздких вычислений и преобразований, позволяя сосредоточиться на предмете, рассмотреть большее количество примеров и решить больше задач;

- предоставляет возможности для самопроверки на всех этапах работы;
- выполняет роль наставника, предоставляя неограниченное количество разъяснений, повторений, подсказок и прочее.

На практических занятиях:

- помогает преподавателю проводить самостоятельную работу за компьютером, оставляя за собой роль руководителя и консультанта;
- позволяет преподавателю с помощью компьютера быстро и эффективно контролировать знания учащихся, задавать содержание и уровень;
- сложности контрольного мероприятия. позволяет использовать компьютерную поддержку для решения большего количества задач, освобождает время для анализа полученных решений и их графической интерпретации.

2. РАЗРАБОТКА ЭЛЕКТРОННОГО ОБРАЗОВАТЕЛЬНОГО РЕСУРСА

В этой главе описаны этапы, средства разработки электронных образовательных ресурсов.

2.1 Постановка задачи

Требуется разработать ПО на языке C++, выполняющее функции необходимые для обучающегося и для преподавателя.

Для обучающегося:

- Регистрация в системе автоматизированного рабочего места;
- Возможность изучить теоретический материал электронного образовательного ресурса;

Для преподавателя:

- Вход в личный кабинет;
- Ввод данных в программу;
- Возможность проверки знаний учащихся путём тестовых и практических заданий;
- Просмотр сохранённых в файл результатов тестовых заданий на форме.

Разработанное ПО должно обладать интуитивно понятным графическим интерфейсом.

2.2 Средства информационных технологий для проектирования автоматизированного рабочего места

Рассмотрим одно из инструментальных средств – язык программирования C++, являющийся наиболее удобным и современным языком программирования как для профессиональных программистов, работающих в сфере создания программного обеспечения, так и для учителей информатики.

Язык программирования C++ произошёл от языка Smalltalk. Первый вариант языка появился в 1980 году Бьярни Страуструпом и компанией «Белл Лабораториес». В отличие от своего предшественника C++ получил поддержку объектно-ориентированного программирования. Также была введена проверка и преобразование типов параметров функции, были включены производные классы управления открытым/закрытым доступом к унаследованным компонентам, конструктор и деструктор, а также дружественные классы [17, с. 113].

Язык программирования C++ является очень популярным, одним из основных факторов его популярности является доступность хороших и не дорогих компиляторов. C++ практически полностью совместим с его предшественником C. И несмотря на его недостатки, т. е. сложность и объёмность, этот язык имеет больше возможностей. В учебнике Роберта У. Себесты «Основные концепции», C++ сравнивается с языком Delphi, где автор выбирает Delphi с точки зрения простоты. Но чтобы понять, действительно ли C++ сложнее Delphi, достаточно рассмотреть базовые возможности языков программирования [17, с. 116]. На начальном этапе работы в одной из этих программных сред, разницу в сложности вы просто не заметите, т.к. первое время вы будете осваивать следующее:

использование переменных, циклы, ветвления, массивы, функции. Это самые важные вещи, без которых программирование вообще невысказимо. [23]. И в наши дни сравнивать два этих языка не актуально. Эти продукты занимают абсолютно разные доли рынка [17, с. 115].

Ещё один язык, который хотелось бы рассмотреть – это ActionScript, так как этот язык наиболее простой который может использовать педагог в обучении школьников.

ActionScript — объектно-ориентированный язык программирования, один из диалектов EcmaScript, который позволяет добавить в Flash-ролик логику, обработку данных и многое другое. Возможности языка ActionScript используются для создания интерактивных сайтов полностью сделанных на Flash, элементов сайтов, мультимедиа-приложений, flash-игр и много другого.

По своему диалекту язык ActionScript во многом схож с JavaScript, если Вы знаете JavaScript, ActionScript покажется Вам знакомым. ActionScript компилируется в байткод, который включается в SWF-файл. Исполнить swf-файл можно Flash-плеером, который существует в виде плагина к веб-браузеру или в виде самостоятельной программы [17, с. 117].

Язык ActionScript, который используется в программе Macromedia Flash MX, позволяет создавать видеоклипы. Вам не нужно знать все элементы ActionScript для того, что бы начать писать свои скрипты. Если цель ясна, то можно начать строить скрипты, используя простейшие операторы. Затем, по ходу работы, можно внедрять новые изученные элементы языка, что бы выполнять более сложные задачи [9].

Существует еще множество средств создания компьютерных программ, отличающихся друг от друга, и имеющих свои преимущества и недостатки. Это [C++](#), [Object Pascal](#), [Java Script](#), Delphi, [Oberon-2](#) [21].

Но выбранный нами для анализа Язык С++ позволят решить поставленную перед нами задачу по проектированию электронных образовательных ресурсов

2.3 Разработка структуры приложения

Написания программного обеспечения – сложный процесс, требующий особое внимание. Из всего этапа разработки можно выделить следующее:

1. Написание технического задания.
2. Кодирование программы.
3. Тестирование.
4. Внедрение и сопровождение.

Разработка программы начинается с описания технического задания. От этого зависит, насколько точным он будет и каким станет конечный результат. Требования должны быть качественно проанализированы и точно продуманы. Соберите как можно больше информации, для этого нужно разобрать все вопросы у программистов и у тех, кто будет непосредственно пользоваться программой.

По техническому заданию разрабатывается структура компьютерной программы, в которой отображаются основные блоки, представленные на рисунке 2.1.

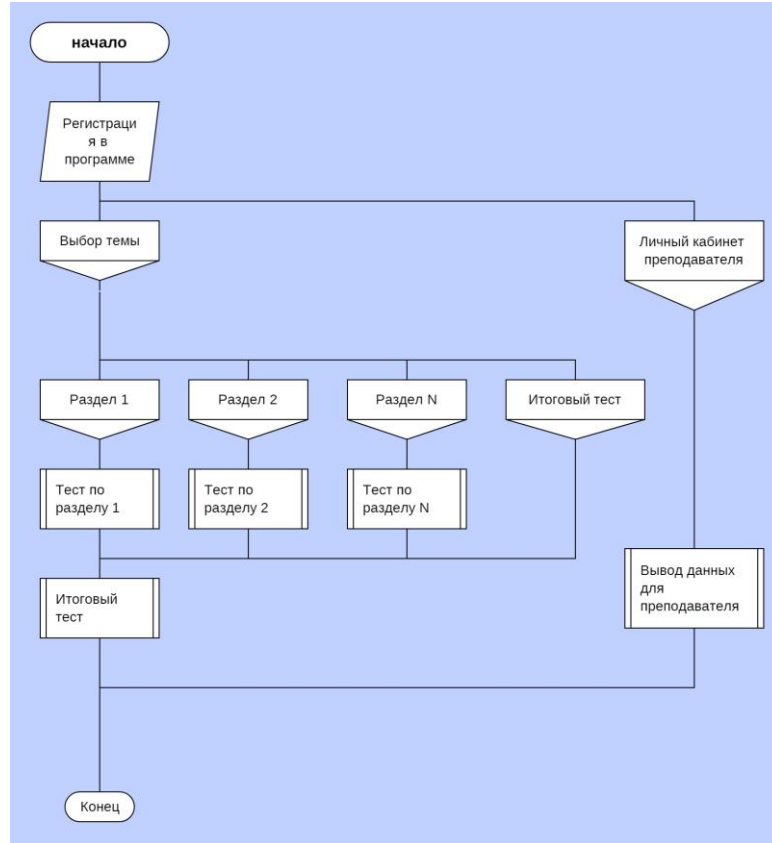


Рис. 2.1. Структура обучающей компьютерной программы

Следующий этап – разработка программы, или написания программистами кода программы.

Во время внедрения в редких случаях возникают ошибки, программное обеспечение прошло ряд экспериментов, и теперь готово к использованию. При сопровождении программы усовершенствуется программное обеспечение и обновляется в соответствии с требованиями [12].

Итак, основные этапы проектирования компьютерных программ описаны, теперь можно создать структуру будущей компьютерной программы.

Итак, основными этапами проектирования компьютерной программы являются:

- описание технического задания;
- разработка структуры программы;

– кодирование программы или отладка.

Определив основные этапы создания компьютерной программы, была разработана структура будущей учебной компьютерной программы. Таким образом, реализованы поставленные цели и задачи, связанные с проектированием учебной компьютерной программы.

1. Вход в систему / создание пользователя

Пользователь заполняет форму:

- Фамилия;
- Имя;
- Отчество.

Если введенные данные найдены в базе данных, система считает, что пользователь существует и запрашивает пароль (Рис.4). В случае правильно введенного пароля система переходит в режим предоставления материала. Если пользователь три раза введет пароль неверно, то система завершит работу.

Преподаватель входит в личный кабинет и получает доступ к результатам тестовых заданий, к спискам групп.

Если данные, введенные в форму, не найдены в базе данных, система предложит создать пользователя или пройти регистрацию снова.

2. Режим предоставления материала

Программа предоставляет пользователю меню и поле просмотра материала (Рис.5). Через меню пользователь может выбрать различные режимы работы программы (см. раздел 4).

Поле просмотра отображает информацию из текущего раздела курса, выбранного пользователем.

3. Режим тестирования

Пользователю задается вопрос с вариантами ответа. Пользователь может ответить на вопрос и перейти к следующему вопросу или завершить тестирование с последующим завершением программы (Рис.6).

3. РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПО

3.1 Разработка пользовательского интерфейса

Поскольку данная программа выполняет ограниченный набор функций, разработанный пользовательский интерфейс достаточно прост. Он состоит из окна регистрации и авторизации, главного окна, строки меню.

Окно регистрации содержит (рис. 3.2) поля для ввода фамилии, имени, отчества ученика. Для преподавателя содержатся поля ввода логина и пароля.

В основном окне пользователь выбирает раздел для изучения и посредством кнопок переходит к теоретическому материалу и тестам.

В отдельном окне личного кабинета преподавателя, выводятся списки групп и данные пользователей, вывод оценок по тестам.

Программа разработана как дополнение к традиционному процессу обучения, и даёт возможность повторить темы и проверить знания. В конце изучения тем, пользователь проходит итоговый тест. Ученик имеет возможность выбрать тему и изучить теоретический материал, после каждого раздела, ученику предлагается пройти тест и практические задания и проверить свои знания. В конце теста, на экран выводится оценка и предложение пользователю вернуться к теории, чтобы изучить тему ещё раз, либо вернуться к разделам электронного ресурса. Оценка записывается в файл базы данных.

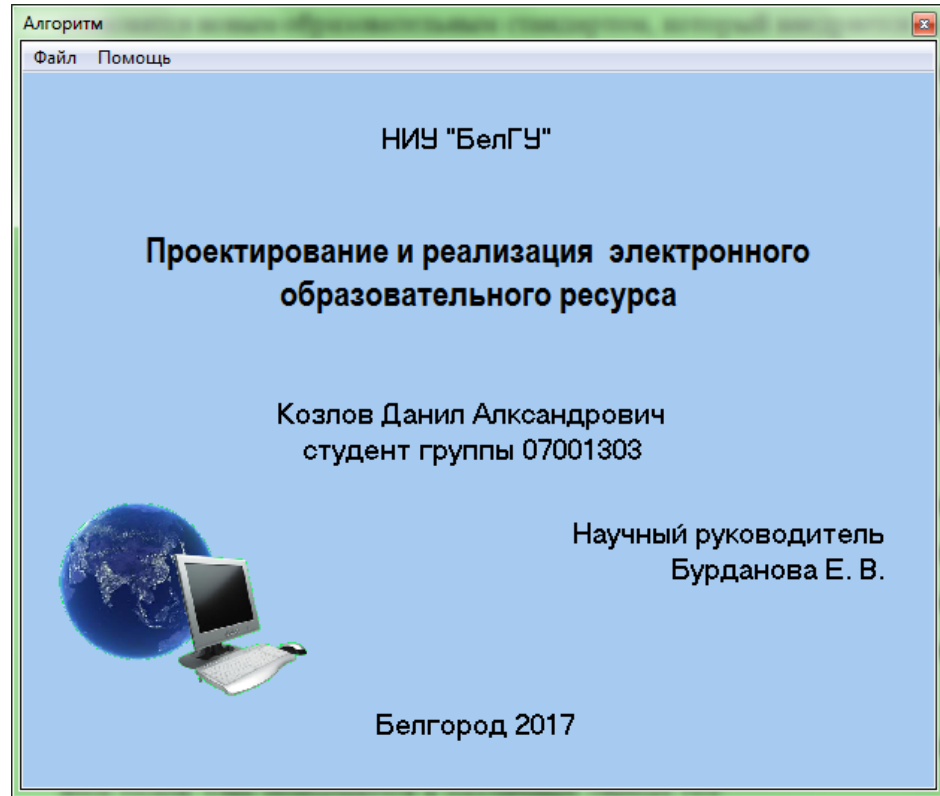


Рис. 3.1. Запуск программы

При входе в программу пользователю предлагается выбрать свою фамилию и группу, либо зарегистрироваться в системе, если их нет в списке. Выбранные данные будут сохранены в файл, в который далее будут записаны оценки за прохождения тестов. Когда пользователь завершит регистрацию, он перейдёт на форму выбора разделов (Рис. 3.2).

Рис. 3.2. Регистрация в системе

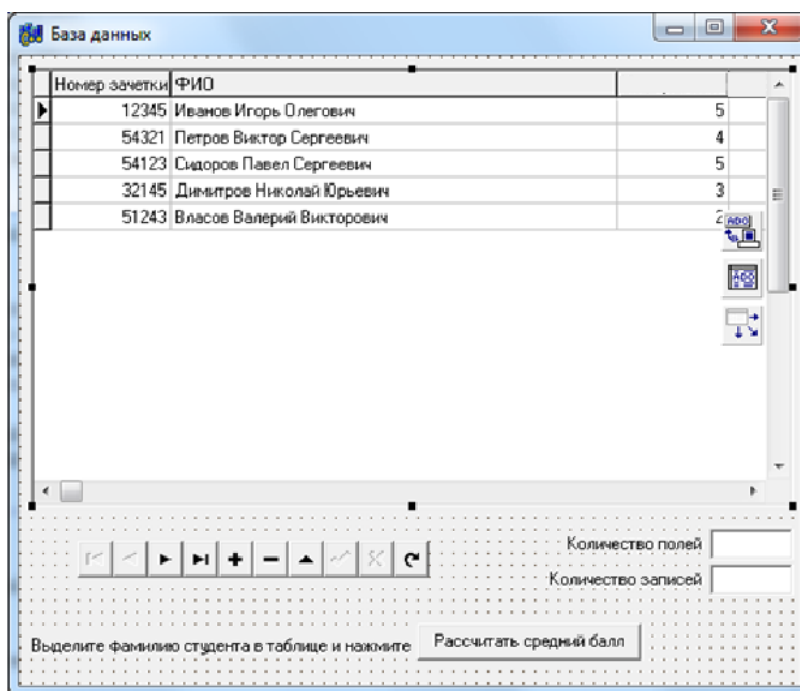


Рис. 3.3. Вывод оценок

Преподаватель имеет возможность войти в личный кабинет программы и получить доступ к результатам тестов. На форме результатов, преподавателю выводится группа, фамилия имя и оценка обучающегося.

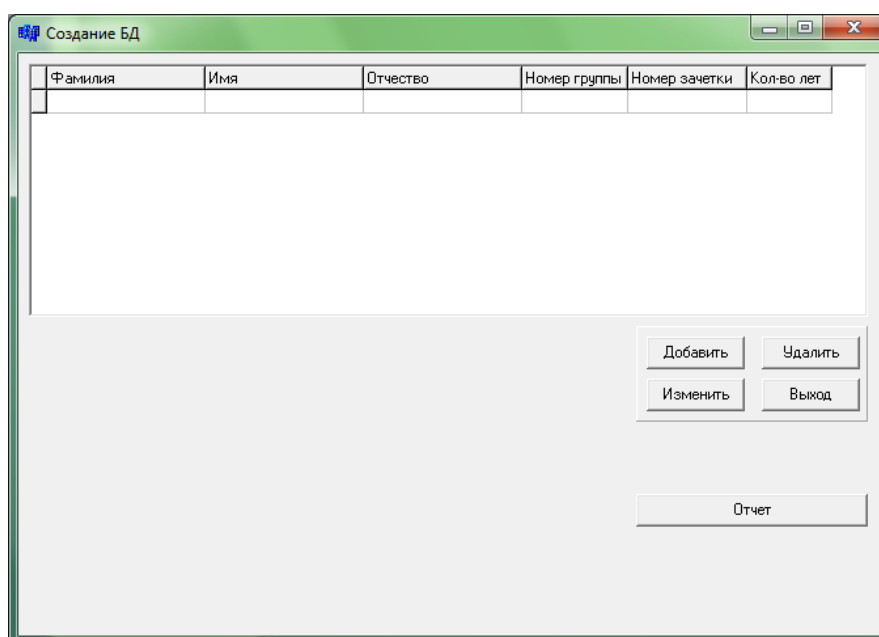


Рис. 3.4. Редактирование записей преподавателем

Пользователь может выбрать раздел из меню или строки меню и перейдёт к изучению выбранной темы.

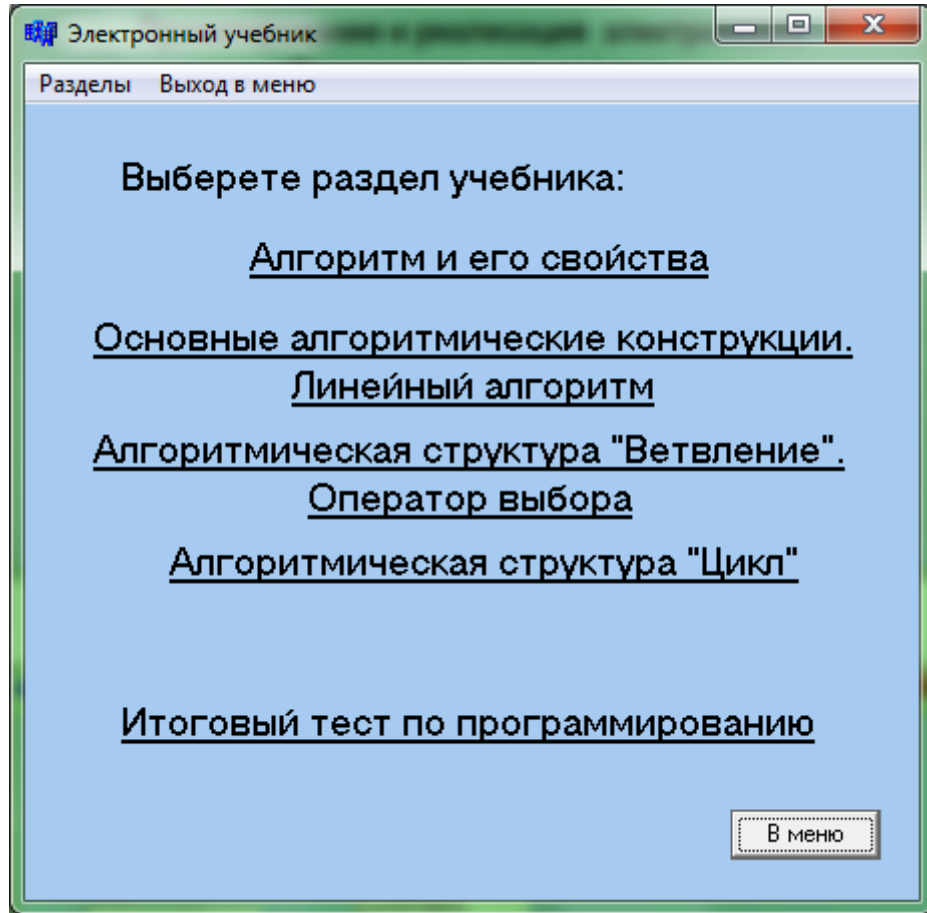


Рис. 3.5. Выбор раздела

Обучающийся, изучает тему раздела, переходя по кнопке далее, открывает новые страницы и проходит всю тему выбранного раздела (рис. 5). После изучения теоретического материала, пользователь переходит к тестовым и практическим заданиям. Если вариант ответа не выбран, то появляется окно с требованием выбрать один из вариантов ответа. В программе есть, как тесты где нужно ввести ответ с клавиатуры, так и тесты с выбором варианта ответа, множества правильных ответов (рис. 3.5 и рис. 3.6).

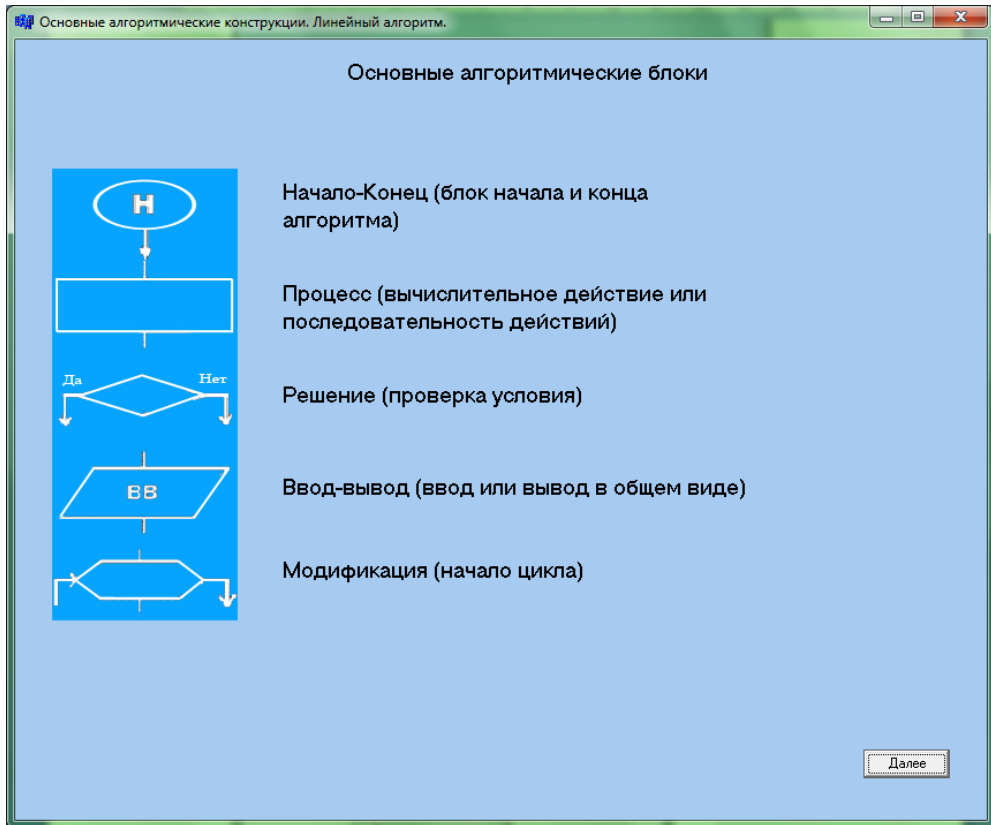


Рис. 3.6. Пример этапа изучения теории электронного ресурса

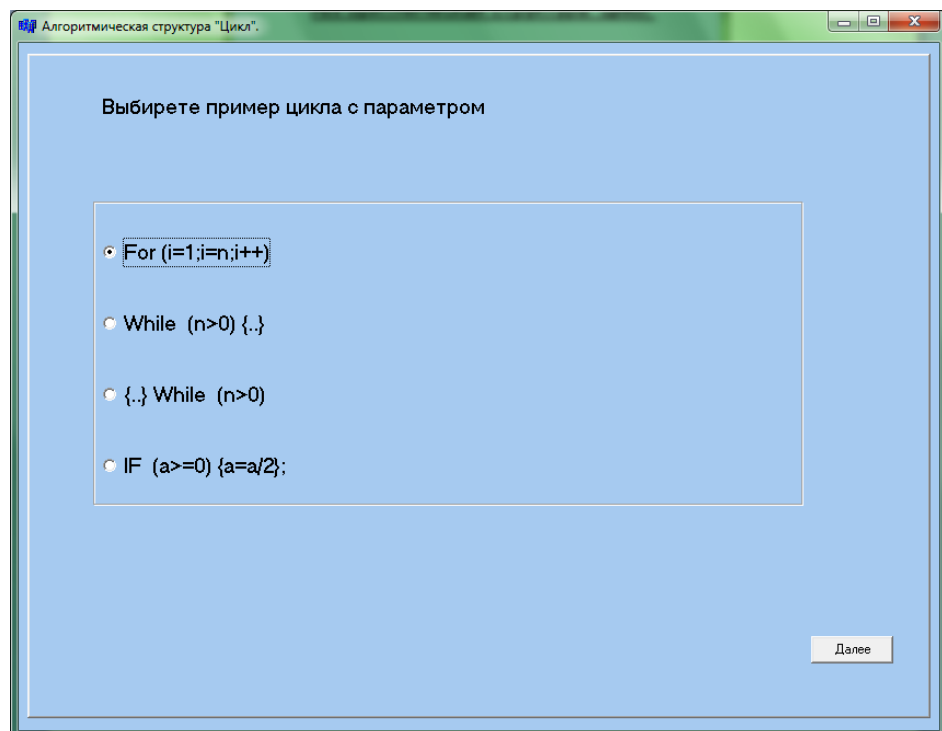


Рис. 3.7. Пример выполнения тестового задания электронного ресурса

Следующим этапом является ознакомление с учебным материалом по темам. В теме первого урока изучается: понятие алгоритма, свойства алгоритма, виды, способы записи, алгоритмические конструкции. Представлены изображения блок-схем для полного понимания материала. Так же еще представлена структура программы на языке C++, переменные, операторы, правила записи основных структур и операторов.

В конце изучения данной темы ученики приступают к проверке первичного понимания изученного материала с помощью теста по теме (Рисунок 4). Тест носит обучающий характер, учащимся предоставляется информация о том, на какие вопросы были даны верные ответы, а с какими вопросами они не справились. Если оценка учащихся ниже «4», то программа возвращает обучаемого к теории, если ученик получает «4» и выше, то он может по выбору закончить работу с тестом, либо снова повторить тему.

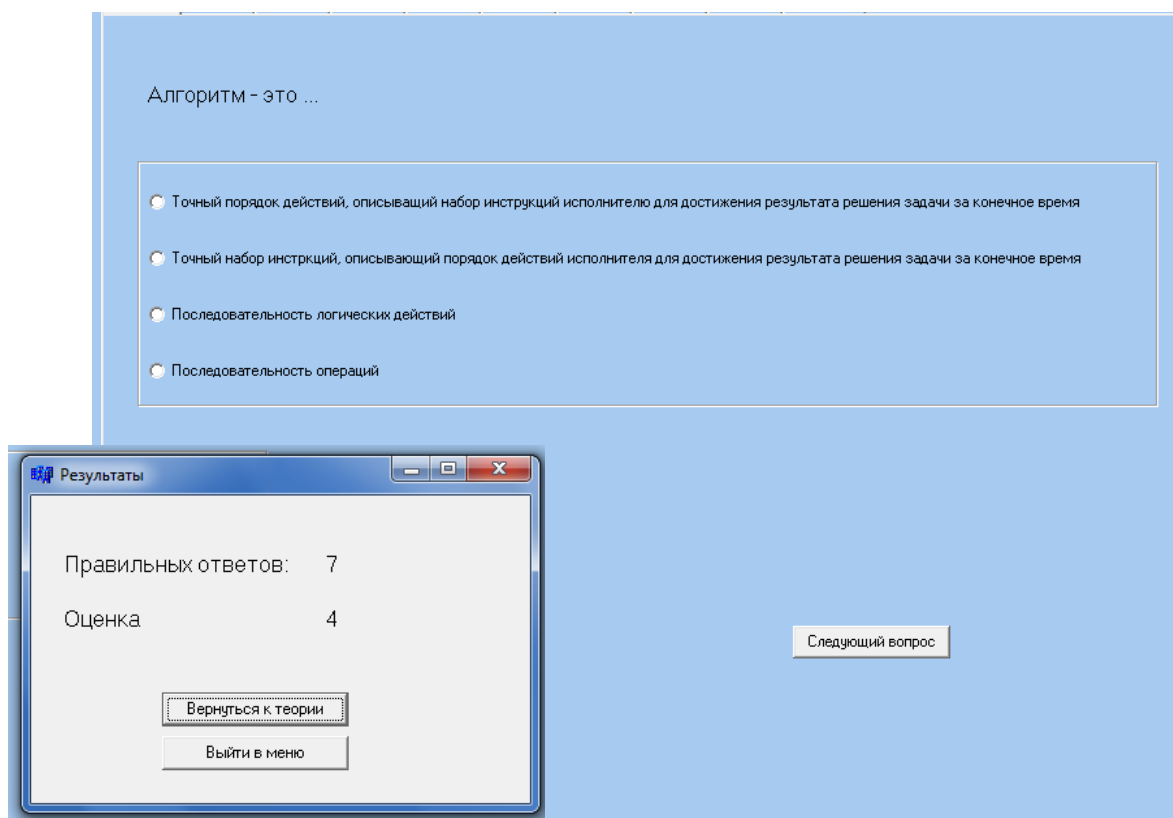


Рис. 3.8. Скриншот компьютерного теста обучающего характера.

Второй урок посвящён теме «Основные алгоритмические структуры. Линейный алгоритм». По этой теме осуществляется обзор всех алгоритмических структур: линейный алгоритм, алгоритмическая структура «Ветвление» и «Выбор», алгоритмическая структура «Цикл». Более подробно изучается структура «Линейный алгоритм».

Изучив теоретическую часть, ученики выполняют практическую работу, в которой нужно разместить по порядку представленные операторы и элементы блок-схем. После этого ученики проверяют свои знания с помощью теста, практических заданий.

На третьем уроке изучаются темы «Алгоритмическая структура «Ветвление». Оператор выбора». Учащиеся знакомились с графическим представлением алгоритмов в форме блок-схемы, решали задачи с использованием алгоритмической структурой «Ветвление» и оператора выбора. Практическая работа представляла собой задания с алгоритмом, записанном в форме блок-схемы или программы на языке Borland Pascal, учащимся необходимо было исполнить алгоритм и ввести в текстовое поле ответ. Так же она включала в себя задания на соответствие, в которых предлагалось расположить элементы блок-схем или операторов задач языка Borland Pascal в правильном порядке. После выполнения практической работы учащимся предлагалось компьютерное тестирование.

Четвёртый урок. Алгоритмическая структура «Цикл». Это наиболее обширная тема курса учебной компьютерной программы. В теме представлен обзор видов циклов «С параметром», «С предусловием», «С пост условием», после чего мы останавливаемся на каждом из них. Рассматриваются способы записи на языке Borland Pascal, блок-схемы, простые жизненные, бытовые примеры, для лучшего понимания повторений. Практическая работа состоит из задач, где необходимо посчитать повторения выведенной на экране программы, или ввести ответ, который получится в результате выполнения программы.

На контрольном этапе была проведена практическая работа и тестирование.

После завершения работы на экране учащимся был представлен результат, их оценка и количество правильных ответов. Кроме этого оценка сохранялась в защищённый файл.

После изучения теоретической части учащийся должен пройти контрольную точку. Данное окно служит для прохождения теста по изученному материалу. При выводе его на экран предыдущее окно с теоретическим материалом сворачивается. Пользователь выбирает тему, нажимает клавишу начать тест, видит вопросы с вариантами ответов и выбрав вариант ответа переходит к следующему посредством кнопки «Далее».

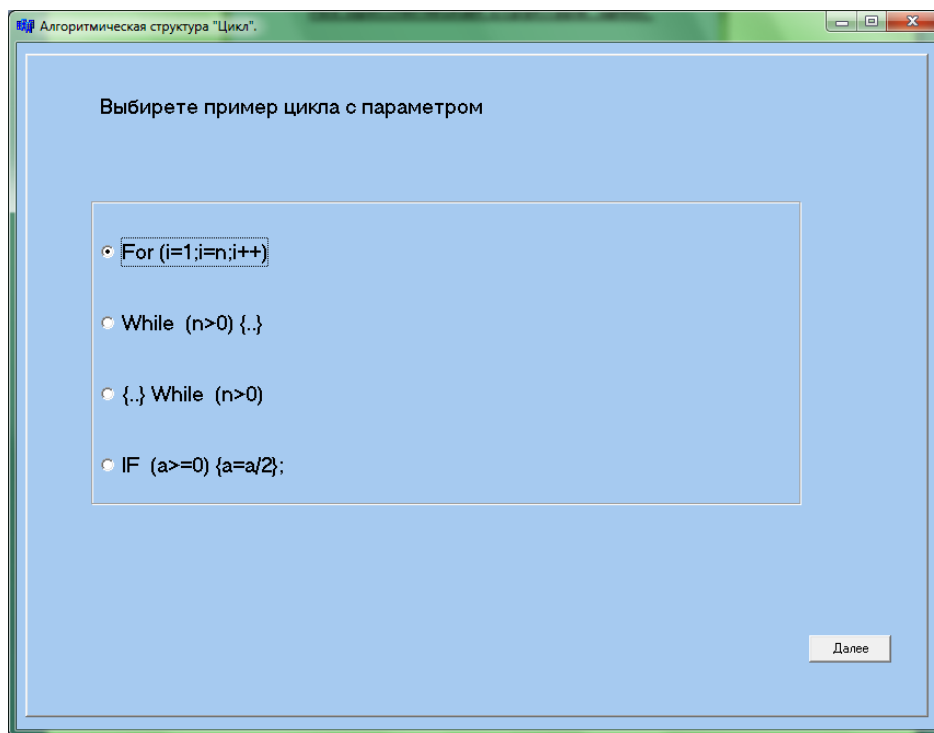


Рис. 3.9. Окно тестирования

В конце тестирования учащийся узнаёт количество верных и не верных своих ответов, преподаватель также может проверить результаты

тестирования каждого учащегося, в дальнейшем все результаты будут автоматически фиксироваться для каждого конкретного ученика.

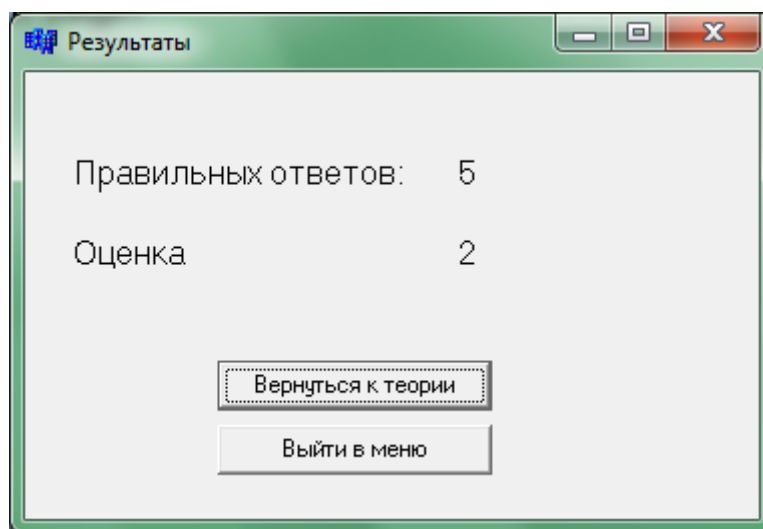


Рис. 3.10. Окно результатов тестирования

3.2 Разработка приложения

Программа автоматизированного рабочего места реализована средствами среды программирования Borland C++ Builder 6. Теоретический материал программы реализован посредством canvas, куда и вводится текст. Текст может быть занесён как вручную, так и с помощью формы в личном кабинете преподавателя.

Тесты в программе реализованы с помощью компонентов «RadioGroup», либо множественного выбора «ListBox». Подсчёт правильных ответов и оценки продемонстрирован в листинге 3.

Вопросы, где предлагается ввести правильный ответ, выполнены с помощью компонента «Edit», введённые ответы сравниваются со значением правильного ответа хранящегося в переменной.

Листинг 1. Вывод формы по центру экрана

```
{Form36->Position=poScreenCenter; }
```

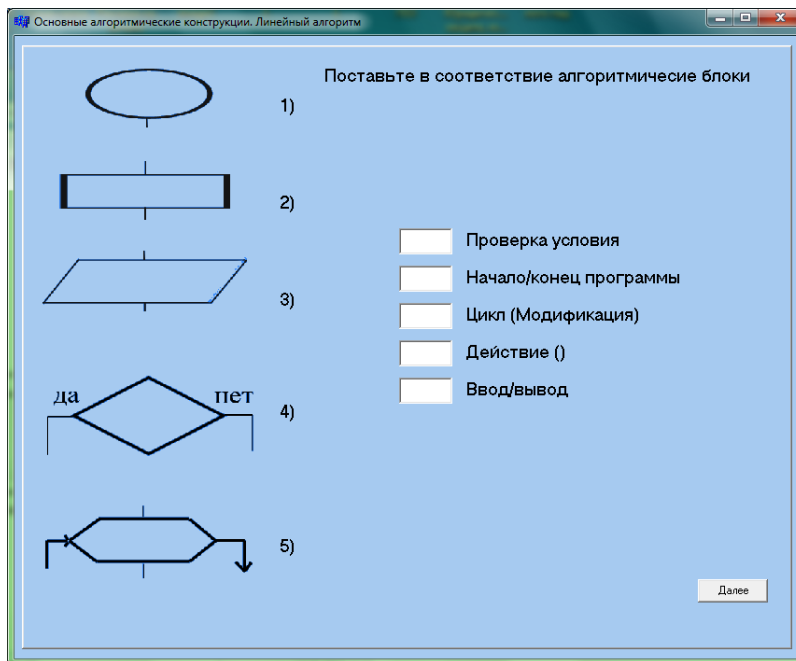


Рис. 3.11. Пример выполнения практического задания электронного ресурса

Так же в программе имеются практические задания, например одно из заданий, в котором ученику предлагается расположить алгоритмические конструкции в правильном порядке (рис. 3.11).

Листинг 3.1. Пример описания тестового вопроса в программе.

```
void __fastcall TForm36::Button1Click(TObject *Sender)
{
    k==0;
    if (RadioGroup1->ItemIndex==1) { ShowMessage("Выберете ответ"); } else {
        TabSheet2->Show();
        TabSheet1->Enabled=false;
        if (RadioGroup1->ItemIndex==1) { k++; } }
}
```

Листинг 3.2. Пример подсчёта правильных ответов в программе.

```
void __fastcall TForm36::Button15Click(TObject *Sender)
{
    if (Edit4->Text=="") { ShowMessage("Введите ответ"); } else {
        Form37->Show();
    }
}
```

Листинг 3.2. Продолжение.

```

TabSheet15->Enabled=false;
if (Edit4->Text=="14") {k++;}
if (k==0) {n=2;}
if (k==1) {n=2;}
if (k==2) {n=2;}
if (k==3) {n=2;}
if (k==4) {n=2;}
if (k==5) {n=2;}
if (k==6) {n=3;}
if (k==7) {n=3;}
if (k==8) {n=3;}
if (k==9) {n=3;}
if (k==10) {n=4;}
if (k==11) {n=4;}
if (k==12) {n=4;}
if (k==13) {n=4;}
if (k==14) {n=5;}
if (k==15) {n=5;}
Form37->Label1->Caption=k;
Form37->Label2->Caption=n;
}
}

```

Пользователь нажимает на один из объектов и перетягивает на панель в правой части формы. Как только на панели будет собрана блок-схема, появится сообщение о правильности выполнения задания (рис. 3.14). При неправильном выполнении задания, при нажатии на кнопку «далее», будет выведено сообщение с просьбой расположить блоки в правильном порядке. Если фигуры не будут расположены правильно, то появится сообщение о том, что блоки нужно расположить в правильном порядке (рис. 3.13).

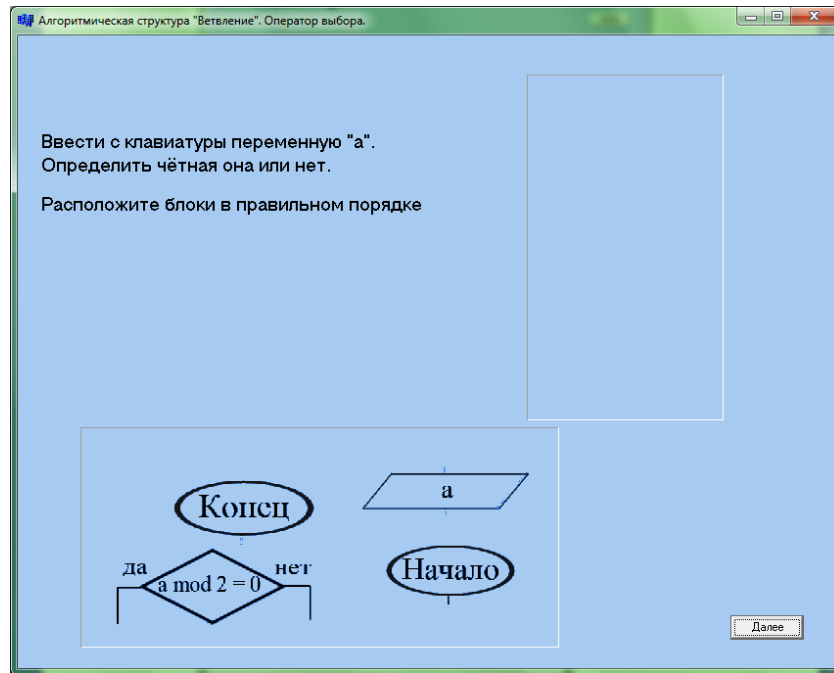


Рис. 3.12. Пример выполнения практического задания электронного ресурса.

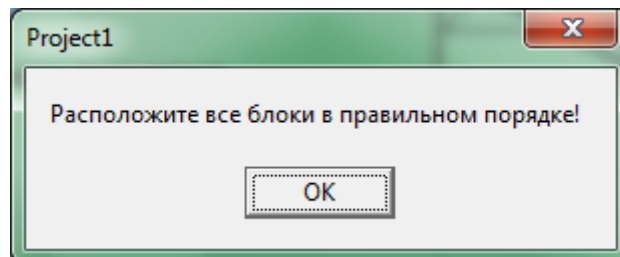


Рис. 3.13. Всплывающее окно

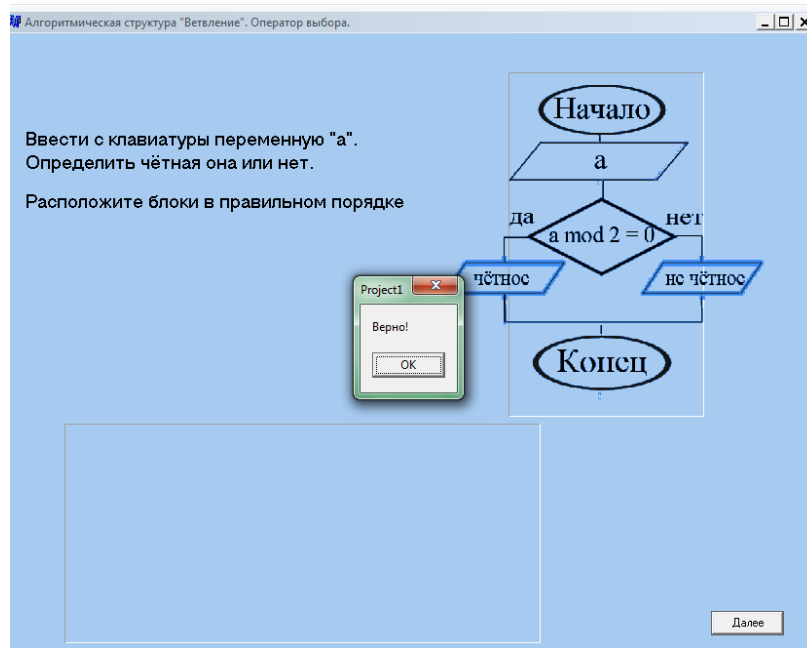


Рис. 3.14. Вывод результата теста или практического задания

Листинг 3.3. Описание переменных

```
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm25 *Form25;
int OldX;
int OldY;
bool IsMouseDown=false;
```

Листинг 3.4. Описание переменных

```
void __fastcall TForm25::Image1MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    OldX=X;
    OldY=Y;
    IsMouseDown=true;
    for (Image1->Left==440; Image1->Left==550; Image1->Left++)
    for (Image1->Top==60; Image1->Top==100; Image1->Top++)
    {IsMouseDown=false; Image1->Top=80; Image1->Left=480;}
}
```

Листинг 3.5. Движение мыши

```

void __fastcall TForm25::Image1MouseMove(TObject *Sender,
    TShiftState Shift, int X, int Y)
{
    if(IsMouseDown)
    {
        Image1->Left+=X-OldX;
        Image1->Top+=Y-OldY;
    }
}

```

Как только блок схема будет выстроена верно, появится соответствующее сообщение, и пользователь сможет перейти далее (рис 3.10). Для реализации данного модуля задаём значения координат x и y (листинг 3.3), Прописываем события для нажатия кнопки мыши (листинг 3.4), объект фиксируется при нажатии на него мышью, и мы сможем его перетащить, отпускание кнопки (листинг 3.6), и сравниваем с конечными координатами объекта (листинг 3.7).

Листинг 3.6. Действие при отпускании кнопки мыши

```

void __fastcall TForm25::Image1MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    IsMouseDown=false;
    if (Image1->Top>Image2->Top & Image1->Left>Image2->Left & Image1->
    Top<Image2->Top+Height & Image1->Left<Image2->Left+Width) {
        Image1->Visible=False;Image2->Visible=True;
    }
}

```

Листинг 3.7. Описание события нажатия кнопки, сравнение результатов

```

void __fastcall TForm25::Button1Click(TObject *Sender)
{

```



```

if (Image1->Top>Image2->Top & Image1->Left>Image2->Left & Image1-
>Top<Image2->Top+Height & Image1->Left<Image2->Left+Width & Image9->Top>Image8-
>Top & Image9->Left>Image8->Left & Image9->Top<Image8->Top+Height & Image9-
>Left<Image8->Left+Width & Image3->Top>Image4->Top & Image3->Left>Image4->Left &
Image3->Top<Image4->Top+Height & Image3->Left<Image4->Left+Width & Image5-
>Top>Image7->Top & Image5->Left>Image7->Left & Image5->Top<Image7->Top+Height &
Image5->Left<Image7->Left+Width) {
    ShowMessage("Верно!"); Form25->Hide(); Form3->Show();} else
ShowMessage("Расположите все блоки в правильном порядке!");
}

```

3.3 Тестирование

После разработки ПО требуется протестировать его, чтобы убедиться в отсутствии ошибок, а также в наличии всей функциональности, требуемой постановкой задачи.

Экспериментальная апробация автоматизированного рабочего места проходила на занятиях Центра компьютерного обучения.

В процессе обучения на формирующем этапе было использовано автоматизированное рабочее место как учебный электронный образовательный ресурс.

В ходе тестирования программы, программа была использована на занятиях по дополнительной образовательной программе «Основам программирования С++» Центра компьютерного обучения НИУ «БелГУ».

Для проверки правильности разработки ПО, необходимо протестировать его, проверить функциональность и наличие ошибок, выполнение требуемой постановкой задачи. Тестирование было проведено на базе группы № 17 Центра компьютерного обучения, изучающей курс «Основы программирования С++». На занятиях, ученикам было предложено изучить программу, теоретический материал программы, пройти тесты.

Содержание учебной компьютерной программы состоит из следующих разделов и соответствует требованиям стандарта общего образования по информатике и информационным технологиям:

1. Алгоритм и его свойства.
2. Основные алгоритмические структуры. Линейный алгоритм.
3. Алгоритмическая структура «Ветвление». Оператор выбора.
4. Алгоритмическая структура «Цикл».

Для организации контроля знаний в учебную компьютерную программу вошли задания и тесты по темам:

1. Алгоритм и его свойства.
2. Основные алгоритмические структуры. Линейный алгоритм.
3. Алгоритмическая структура «Ветвление». Оператор выбора.
4. Алгоритмическая структура «Цикл».
5. Контрольная работа.

На первом уроке с учащимися проводится инструктаж по работе с учебной компьютерной программой, демонстрируется ее содержание и возможности. Приобретается навык ориентации по учебной компьютерной программе. Программа легка и понятна в применении, то есть в программе применена удобная навигация, переходя с одной странице на другую, мы всегда можем вернуться назад, или сразу выйти из приложения, за исключением страницами с тестами и практическими заданиями. Кроме того, в программе имеется привлекательный и не раздражающий восприятие дизайн.

Обучающиеся, для проверки функциональности программы выполнили последовательность действий: Открыли программу, ввели свои данные в форму регистрации: Фамилию Имя Отчество и группу. Данные были занесены в файл базы данных. Далее Ученики перешли на форму выбора разделов, выбрав нужный раздел, приступили к изучению теоретического материала. В конце каждого раздела открывается тест,

который служит для закрепления изученного материала. Пройдя тест, обучающийся видит свою оценку, ему предлагается либо пройти тест заново, либо вернуться к выбору разделов. В тесте используются как вопросы с вводом ответа, так и выбор ответа. Кроме того в программе присутствуют практические задания, такие как вычисление результата выполнения программы, так и расположение в форме блок-схем в правильном порядке. В базе данных добавляется новая запись с результатом теста: группа, фамилия имя отчество, попытка и оценка. Преподаватель через личный кабинет имеет возможность посмотреть, за какие вопросы ученик получил баллы, а на какие вопросы ответил не верно. Такая же последовательность действий проходит и с остальными разделами. Раздел итоговый тест отличается от остальных тестов тем, что в программе меняется порядок вопросов. В конце выводится количество правильных ответов и оценка. Более подробные данные, например, какие вопросы, даны правильные ответы, а на какие нет может видеть только преподаватель.

Программа имеет дружелюбный интерфейс, удобна по функционалу при использовании учеником, соответствует возрастным особенностям ученика и удовлетворяет требованиям раздела курса. Таким образом, можно удостовериться в корректности работы заявленных функций ПО. Преподавателю программа даёт возможность следить за оценками ученика и управлять его учётной записью в базе данных программы

ЗАКЛЮЧЕНИЕ

Значение современных информационных и мультимедийных технологий в их универсальности и многофункциональности. Но при всех своих больших возможностях эти технологии предоставляют собой только средства, потенциально позволяющие сделать более эффективной деятельность обучающего. В том, как раскрыть этот потенциал для образовательного процесса, и состоит главная многоплановая проблема совершенствования образования на базе информационных технологий.

Электронные образовательные ресурсы являются очень эффективной поддержкой при обучении и приобретении знаний в школе, при использовании их в качестве инструментов познания для отражения того, что ученики выучили на данном уроке и что они знают в целом. Компьютерные технологии должны использоваться не только для распространения информации, компьютеры должны использоваться во всех областях знаний в качестве инструментов, которые помогают более наглядно представлять изучаемый материал и контролировать деятельность учащихся.

Использование компьютера в качестве средства обучения путем применения автоматизированного рабочего места в качестве среды представления знаний способствует более быстрому и более полному усвоению материала. Электронные образовательные ресурсы являются дополнительным стимулом для получения компьютерной грамотности. Поэтому в будущем само их применение в учебном процессе может стать самым первым средством для совершенствования образования. А автоматизированные рабочие места позволят упростить и ускорить работу преподавателя.

В работе приводятся виды средств создания автоматизированных рабочих мест на занятиях компьютерного обучения.

В ходе дипломного проектирования проведен анализ средств создания автоматизированных рабочих мест и влияние их на образовательный процесс. Рассмотрены средства объектно-ориентированного программирования и стандартные средства для создания учебных компьютерных программ и пути целесообразного использования их в образовательном процессе.

Изучив и проанализировав специальную литературу, ознакомившись с возможностями средств информационных технологий для проектирования компьютерных программ учебного назначения можно с уверенностью сказать, что проблема эффективного использования автоматизированных рабочих мест на сегодняшний момент является актуальной и значимой вследствие того, что качественная обучающая компьютерная программа явление не слишком повседневное и простое, особенно если для ее разработки использован объектно-ориентированный язык программирования C++, который соответствует общемировому стандарту. Программа, разработанная этим средством, может способствовать развитию структурированного и наглядно-образного мышления у обучающихся; развитию умений построения логических цепочек и алгоритмов с помощью компьютерных учебных программ; созданию условий для реализации у школьников выбора наиболее эффективных способов применения алгоритмов в зависимости от конкретных задач; реализации интеллектуального потенциала у школьников за счет применения учебной компьютерной программы.

Анализ источников предоставил нам возможность на основе опыта их авторов определить структуру обучающей программы, в которую можно включить теоретическую, контролирующую части и тренинг. и обосновать выбор средств для разработки собственной компьютерной программы учебного назначения.

В результате проделанной работы было рассмотрено понятие учебные компьютерные программы, автоматизированные рабочие места используемые на курсах Центра компьютерного обучения. Были рассмотрены возможности различных сред объектно-ориентированного программирования, их положительные и отрицательные стороны, и как их удачно использовать в учебной деятельности.

Так же были разработаны учебные компьютерные программы, разработанные в среде объектно-ориентированного программирования Borland C++ Builder 6.

При проектировании автоматизированных рабочих мест учитывались требования к их содержанию и оформлению. Учебные компьютерные программы регулярно использовались на занятиях. Это способствовало повышению уровня сформированности системы понятий у обучающихся.

В заключении хотелось бы отметить, что учебные компьютерные программы является дополнительным стимулом для получения компьютерной грамотности.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Асмолова А. Г. Как проектировать универсальные учебные действия в начальной школе: от действия к мысли: пособие для учителя (Стандарты второго поколения). / под ред. А. Г. Асмолова. – М.: Просвещение, 2008. – 152 с.
2. Закон РФ «Об информации, информатизации и защите информации» от 20.02.1995, №24-ФЗ
3. Автоматизированные учебные курсы по информатике.
<http://www.nntu.sci-nnov.ru/RUS/REC/inf1.htm>
4. Thomas Bruckner Ein WWW-basiertes Lernsystem zum Thema "Internet". – режим доступа:
<http://zemm.ira.uka.de/~brueckner/documents/da/index.html>, 1998 (Дата обращения: 11.01.2011).
5. Брусиловский П.Л. Адаптивные обучающие системы в World Wide Web: обзор имеющихся в распоряжении технологий. [Электронный курс] //Авт.пер.на сайте. – Режим доступа:
<http://ifets.ieee.org/russian/depository/WWWITS.html>1998 (Дата обращения: 23.04.2011).
6. Галеев И.Х. О систематизации учебных компьютерных средств [электронный ресурс] / И. Х. Галеев. – Режим доступа:
<http://ifets.ieee.org/russian> 1998 (Дата обращения: 17.03.2011).
7. Гейн А.Г. Основы информатики и вычислительной техники: Проб. учеб. для 10 - 11 кл. общеобразоват. Учреждений / А.Г. Гейн, В.Г. Житомирский, Е.В. Линецкий. - 5-е изд. - М.: Просвещение: Моск. учебники, 1996. - 254с.: ил.
8. Иванова В.А. Педагогика [Электронный ресурс] / В.А. Иванова, Т.В. // Левина электронный учебно-методический комплекс. – Режим

доступа: http://www.kgau.ru/distance/mf_01/ped-asp/08_01.html (Дата обращения: 23.04.2017).

9. Информационный ресурс для студентов ИУЗ.МГТУ [Электронный ресурс] . Режим доступа: <http://www.daflash.ru> (Дата обращения: 11.02.2017).

10. Каджаспиров А.Ю. Психологический словарь / ПМ. Коджаспирова, А.Ю. Коджаспиров. – М.: Академия., 2001. – 178с.

11. Кузнецов А. А. Основы информатики. 8 — 9 кл.: Учеб. для общеобразоват. учеб, заведений. - / А. А. Кузнецов, Н.В. Патапова. - М.: Дрофа, 2001. - 176с.:ил.

12. Культин Н.Б. Основы программирования в Delphi 7 . - / Н. Б. Культин. - 2-е изд. СПб.: БХВ-Петербург, 2008.

13. Липаев В. В. Проектирование программных средств: Учеб. пособие для вузов по спец. «Автом. сист. обр. информ. и упр ». - / В. В. Липаев. — М.: Высш. шк., 1990.— 303 с: ил

14. Лихачев Б. Т. Педагогика: Курс лекций: Учеб. пособие для студентов пед. учеб, заведений и слушателей ИПК и ФПК. - / Б. Т. Лихачев. - М.: Прометей, 1992.

15. Первин Ю. А. Формирование операционного стиля мышления как социальный заказ информационного общества общеобразовательной школе [Электронный ресурс] / Ю. А. Первин - <http://center.fio.ru/vio>

16. Примерная программа основного общего образования по информатике и информационным технологиям [Электронный ресурс]. – режим доступа: <http://www.ed.gov.ru/d/ob-edu/noc/rub/standart/pp/09-o.doc> (Дата обращения: 27.02.2017).

17. Себеста Роберт У. Основные концепции языков программирования, 5-е изд. : Пер. с англ. – М.: Издательский дом «Вильямс». 2001 - 672 с. : Парал. тит. англ.

18. Т.М. Полякова, Н.И. Лобова, В.О. Николаев, Д.С. Суслов.
Разработка обучающих курсов в среде мультимедиа.

<http://www.mesi.ru/joe/st056.html>

19. Дистанционное обучение. Определения.

<http://www.user.cityline.ru/~cd-media/definition.htm>

20. Системы дистанционного обучения.

<http://www.entel.kiev.ua/training/projects/remote.html>

21. Кречетников К.Г. Методология проектирования средств информационных технологий обучения. <http://ito.bitpro.ru/2001/ito/II/4/II-4-4.html>

22. А.И. Кириллов, Н.А. Сливина. Компьютерные обучающие программы. Классификация. Критерии качества.

http://www.karelia.ru/pgu/RussianWin/Conferences/Data/19950605/Abstract/sectionB_doc42.html

ПРИЛОЖЕНИЕ

Листинг 1. Запуск программы

```

#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#include "Unit12.h"
#include "Unit18.h"
#include "Unit40.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
Form1->Position=poScreenCenter;
}
//-----
void __fastcall TForm1::N7Click(TObject *Sender)
{
Close();
}
//-----
void __fastcall TForm1::N5Click(TObject *Sender)

```

```
{
Form2->Show();
Form1->Hide();
}
//-----
void __fastcall TForm1::N4Click(TObject *Sender)
{
AboutBox->Show();
}
//-----
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
Form18->Close();
}
//-----
void __fastcall TForm1::FormCanResize(TObject *Sender, int &NewWidth,
int &NewHeight, bool &Resize)
{
Resize = False;
}
//-----
void __fastcall TForm1::N3Click(TObject *Sender)
{
Form40->Show();
Form1->Hide();
}
//-----
```

Листинг 2. Форма авторизации

```

#include <vcl.h>
#include <fstream.h>
#pragma hdrstop
#include "Unit2.h"
#include "Unit3.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm2 *Form2;
//-----
__fastcall TForm2::TForm2(TComponent* Owner)
    : TForm(Owner)
{
Form2->Position=poScreenCenter;
}
//-----
void __fastcall TForm2::Button1Click(TObject *Sender)
{
char s[100];
strcpy(s, Edit1->Text.c_str());
ofstream out(s);
out.close();
ListBox1->Items->Add(Edit1->Text+" "+Edit2->Text+" "+Edit3->Text);
ListBox1->Items->SaveToFile(s);
if (Edit1->Text=="") {ShowMessage("Введите вашу фамилию");}
}
//-----
void __fastcall TForm2::Button2Click(TObject *Sender)
{

```

```

if(ListBox1->ItemIndex < 0)
ShowMessage("Вы не сделали свой выбор");
else {ListBox1->Items->Strings[ListBox1->ItemIndex];
char s[100];
strcpy(s,ListBox1->Items->Strings[ListBox1->ItemIndex].c_str());
Form3->Show();
Form2->Hide(); }
}
//-----
void __fastcall TForm2::FormCanResize(TObject *Sender, int &NewWidth,
int &NewHeight, bool &Resize)
{
Resize = False;
}
//-----
void __fastcall TForm2::Button3Click(TObject *Sender)
{
ShellExecute(NULL,"open","Univer.exe",NULL,"f:\\work\\temp",SW_SHOWNO
RMAL);
Form2->Hide();
}
//-----

```

Листинг 3. Выбор разделов. Раздел изучения теории

```

#include <vcl.h>
#pragma hdrstop
#include "Unit3.h"
#include "Unit4.h"
#include "Unit16.h"

```

```

#include "Unit1.h"
#include "Unit10.h"
#include "Unit26.h"
#include "Unit38.h"
#include "Unit41.h"

//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm3 *Form3;

//-----
__fastcall TForm3::TForm3(TComponent* Owner)
    : TForm(Owner)
{
Form3->Position=poScreenCenter;
}

//-----
void __fastcall TForm3::N13Click(TObject *Sender)
{
Form1->Show();
Form3->Hide();
}

//-----
void __fastcall TForm3::Button1Click(TObject *Sender)
{
Form3->Hide();
Form1->Show();
}

//-----
void __fastcall TForm3::Label2Click(TObject *Sender)

```

```

{
Form4->Visible=true;
setlocale(LC_ALL,"RUS");
Form4->Canvas->Font->Color=clBlack;
Form4->Canvas->Font->Size=12;
Form4->Canvas->TextOut(25,25,"    Алгоритм - точный набор инструкций,
описывающих порядок действий исполнителя для достижения");
Form4->Canvas->TextOut(25,45,"результата решения задачи за конечное
время.");
Form4->Canvas->TextOut(25,65,"    Само слово 'алгоритм' происходит от
'algorithmi' - латинской формы написания имени выдающегося");
Form4->Canvas->TextOut(25,85,"математика IX века аль-Хорезми, который
сформулировал правила выполнения арифметических ");
Form4->Canvas->TextOut(25,105,"операций.");
Form4->Canvas->TextOut(25,130,"Свойства алгоритмов.");
Form4->Canvas->TextOut(25,155,"    Дискретность. Примерами алгоритмов
являются кулинарные рецепты, в которых подробно описана");
Form4->Canvas->TextOut(25,175,"последовательность действий по
приготовлению пищи.");
Form4->Canvas->TextOut(25,195,"Алгоритмы кулинарных рецептов
состоят из отдельных действий, которые обычно нумеруются.");
Form4->Canvas->TextOut(25,215,"Каждый алгоритм состоит из
последовательности отдельных шагов. Если не выполнится один шаг,");
Form4->Canvas->TextOut(25,235,"то не выполнится и следующий. Это
является важным свойством алгоритма - дискретностью.");
Form4->Canvas->TextOut(25,255,"Приведём пример алгоритм приготовления
блюда быстрого питания:");
Form4->Canvas->TextOut(25,275,"    1. Высыпать в емкость содержимое
пакетика.");

```

```

Form4->Canvas->TextOut(25,295," 2. Налить в емкость 200 мл горячей
воды.");
Form4->Canvas->TextOut(25,315," 3. Тщательно перемешать.");
Form4->Canvas->TextOut(25,335,"    Результативность (конечность).
Исполнение алгоритма и получение результата должно");
Form4->Canvas->TextOut(25,355,"завершиться за конечное число шагов.");
Form4->Canvas->TextOut(25,380,"Пример:");
Form4->Canvas->TextOut(25,405,"Алгоритмами являются известные из
начальной школы правила сложения, вычитания, умножения");
Form4->Canvas->TextOut(25,425,"и деления столбиком. Применение этих
алгоритмов всегда приводит к результату.");
Form4->Canvas->TextOut(25,445,"1. Записать числа в столбик, так чтобы
цифры самого младшего разряда чисел (единицы) расположились одна под");
Form4->Canvas->TextOut(25,465,"другой (на одной вертикали).");
Form4->Canvas->TextOut(25,485,"2. Сложить цифры младшего разряда.");
Form4->Canvas->TextOut(25,505,"3. Записать результат под горизонтальной
чертой на вертикали единиц. Если полученная сумма больше");
Form4->Canvas->TextOut(25,525,"или равна величине основания системы
счисления (в данном случае 10), перенести десятки");
Form4->Canvas->TextOut(25,545,"в старший разряд десятков.");
Form4->Canvas->TextOut(25,565,"4. Повторить пункты 2 и 3 для всех
разрядов с учетом переносов из младших разрядов.");
Form3->Hide();
Form4->Show();
}
//-----
void __fastcall TForm3::Label7Click(TObject *Sender)
{
Form4->Hide();

```



```

Form10->Show();
}
//-----
void __fastcall TForm3::Label4Click(TObject *Sender)
{
Form16->Visible=true;
setlocale(LC_ALL,"RUS");
Form16->Canvas->Font->Color=clBlack;
Form16->Canvas->Font->Size=14;
Form16->Canvas->TextOut(25,35,"Алгоритмическая структура 'Ветвление");
Form16->Canvas->Font->Size=12;
Form16->Canvas->TextOut(25,75,"    В отличие от линейных алгоритмов,
в которых команды выполняются последовательно одна ");
Form16->Canvas->TextOut(25,100,"за другой, в разветвляющиеся
алгоритмы входит условие, в зависимости от выполнения или ");
Form16->Canvas->TextOut(25,125," невыполнения которого выполняется та
или иная последовательность команд (серий).");
Form16->Canvas->TextOut(25,150,"    В качестве условия в
разветвляющемся алгоритме может быть использовано любое понятное");
Form16->Canvas->TextOut(25,175,"исполнителю утверждение, которое
может соблюдаться (быть истинно) или не соблюдаться ");
Form16->Canvas->TextOut(25,200,"(быть ложно). Такое утверждение может
быть выражено как словами, так и формулой. Таким образом,");
Form16->Canvas->TextOut(25,225," команда ветвления состоит из условия и
двух последовательностей команд.");
Form16->Canvas->TextOut(25,250,"    Команда ветвления, как и любая
другая, может быть.");
Form16->Canvas->TextOut(25,275," о записана на естественном языке,");
Form16->Canvas->TextOut(25,300," о изображена в виде блок-схемы,");

```

```

Form16->Canvas->TextOut(25,325," о записана на алгоритмическом
языке;");
Form16->Canvas->TextOut(25,350," о закодирована на языке
программирования.");
Form3->Hide();
Form16->Show();
}
//-----
void __fastcall TForm3::Label5Click(TObject *Sender)
{
Form3->Hide();
Form26->Visible=true;
setlocale(LC_ALL,"RUS");
Form26->Canvas->Font->Color=clBlack;
Form26->Canvas->Font->Size=14;
Form26->Canvas->TextOut(25,35,"
Цикл");
Form26->Canvas->Font->Size=12;
Form26->Canvas->TextOut(25,100," При решении подавляющего
большинства задач в программе практически невозможно задать ");
Form26->Canvas->TextOut(25,125,"в явном виде все операции, которые
необходимо выполнить. Пусть необходимо вычислить сумму ");
Form26->Canvas->TextOut(25,150,"первых n членов гармонического ряда: ");
Form26->Canvas->TextOut(25,175,"  $Y = 1 + 1/2 + 1/3 + \dots + 1/n$  ");
Form26->Canvas->TextOut(25,200," Очевидно, что с использованием только
рассмотренных выше типов операторов можно составить");
Form26->Canvas->TextOut(25,225,"программу лишь для фиксированного
значения n. Например, при n=5 требуемые вычисления можно ");

```

```

Form26->Canvas->TextOut(25,250,"здать с помощью оператора
присваивания вида:");
Form26->Canvas->TextOut(25,275,"Y:= 1+1/2+1/3+1/4+1/5");
Form26->Canvas->TextOut(25,300," Если же значение n не фиксируется,
а является исходным данным, вводимым в процессе ");
Form26->Canvas->TextOut(25,325,"выполнения программы (и даже
константой, описанной в программе), то аналогичный оператор ");
Form26->Canvas->TextOut(25,350,"присваивания записать невозможно.
Ибо запись вида Y:= 1+1/2+1/3+...+1/ n в языках ");
Form26->Canvas->TextOut(25,375,"программирования недопустима.");
Form26->Show();
}
//-----
void __fastcall TForm3::Label6Click(TObject *Sender)
{
Form3->Hide();
Form38->Show();
Form38->TabSheet1->Show();
}
//-----
void __fastcall TForm3::FormCanResize(TObject *Sender, int &NewWidth,
int &NewHeight, bool &Resize)
{
Resize = False;
}
//-----

```

Листинг 4. Раздел 1.

```
//-----
```

```

#include <vcl.h>
#pragma hdrstop
#include<locale.h>
#include "Unit4.h"
#include "Unit9.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm4 *Form4;
int x;
int y;
//-----
__fastcall TForm4::TForm4(TComponent* Owner)
    : TForm(Owner)
{
Form4->Position=poScreenCenter;
}
//-----
void __fastcall TForm4::Button1Click(TObject *Sender)
{
Form9->Visible=true;
setlocale(LC_ALL,"RUS");
Form9->Canvas->Font->Color=clBlack;
Form9->Canvas->Font->Size=12;
Form9->Canvas->TextOut(25,35,"Массовость. Алгоритм применим к любой
конкретной формулировке задачи для решения которой он");
Form9->Canvas->TextOut(25,65,"предназначен.");
Form9->Canvas->TextOut(25,100,"Пример:");

```

```

Form9->Canvas->TextOut(25,135,"Алгоритмы сложения, вычитания,
умножения и деления могут быть применены для любых чисел, причем");
Form9->Canvas->TextOut(25,165,"не только в десятичной,но и в других
позиционных системах счисления (двоичной, восьмеричной,");
Form9->Canvas->TextOut(25,195,"шестнадцатеричной и др.");
Form9->Canvas->TextOut(25,225,"Понятность для исполнителя - т.е.
исполнитель алгоритма должен знать, как его выполнять.");
Form9->Canvas->TextOut(25,255,"Определенность - т.е. каждое правило
алгоритма должно быть четким, однозначным.");
Form4->Hide();
Form9->Show();
}

```

```
//-----
```

Листинг 5. Переход к тестовым заданиям.

```
//-----
```

```

#include <vcl.h>
#pragma hdrstop\
#include "Unit6.h"
#include "Unit7.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm6 *Form6;
//-----
__fastcall TForm6::TForm6(TComponent* Owner)
    : TForm(Owner)
{
Form6->Position=poScreenCenter;
}

```

```
//-----
void __fastcall TForm6::Button1Click(TObject *Sender)
{
Form7->Show();
Form7->TabSheet1->Show();
Form6->Hide();
}
//-----
```

Листинг 6. Тест по разделу 1.

```
#include <vcl.h> #include <vcl.h>
#include <fstream.h>
#pragma hdrstop
#include "Unit8.h"
#include "Unit7.h"
#include "Unit2.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm7 *Form7;
int n, k;
//-----
__fastcall TForm7::TForm7(TComponent* Owner)
: TForm(Owner)
{
Form7->Position=poScreenCenter;
}
//-----
void __fastcall TForm7::Button1Click(TObject *Sender)
```

```

{
k==0;
if (RadioGroup1->ItemIndex==-1) { ShowMessage("Выберите ответ"); } else {
TabSheet2->Show();
TabSheet1->Enabled=false;
if (RadioGroup1->ItemIndex==1) {k++;}
}
}
//-----
void __fastcall TForm7::Button3Click(TObject *Sender)
{
if (RadioGroup2->ItemIndex==-1) { ShowMessage("Выберите ответ"); } else {
TabSheet3->Show();
TabSheet2->Enabled=false;
if (RadioGroup2->ItemIndex==0) {k++;} }
}
//-----
void __fastcall TForm7::Button2Click(TObject *Sender)
{
if (RadioGroup4->ItemIndex==-1) { ShowMessage("Выберите ответ"); } else {
TabSheet4->Show();
TabSheet3->Enabled=false;
if (RadioGroup4->ItemIndex==3) {k++;} }
}
//-----
void __fastcall TForm7::Button8Click(TObject *Sender)
{
if (RadioGroup3->ItemIndex==-1) { ShowMessage("Выберите ответ"); } else {
TabSheet5->Show();
TabSheet4->Enabled=false;

```

```

if (RadioGroup3->ItemIndex==2) {k++;} }
}
//-----
void __fastcall TForm7::Button4Click(TObject *Sender)
{
if (RadioGroup8->ItemIndex==1) {ShowMessage("Выбирете ответ");} else {
TabSheet6->Show();
TabSheet5->Enabled=false;
if (RadioGroup8->ItemIndex==3) {k++;} }
}
//-----
void __fastcall TForm7::Button7Click(TObject *Sender)
{
if (RadioGroup5->ItemIndex==1) {ShowMessage("Выбирете ответ");} else {
TabSheet7->Show();
TabSheet6->Enabled=false;
if (RadioGroup5->ItemIndex==0) {k++;} }
}
//-----
void __fastcall TForm7::Button6Click(TObject *Sender)
{
if (RadioGroup6->ItemIndex==1) {ShowMessage("Выбирете ответ");} else {
TabSheet9->Show();
TabSheet8->Enabled=false;
if (RadioGroup6->ItemIndex==3) {k++;} }
}
//-----
void __fastcall TForm7::Button5Click(TObject *Sender)
{

```



```

if (RadioGroup7->ItemIndex==-1) { ShowMessage("Выбирете ответ");} else {
TabSheet8->Show();
TabSheet7->Enabled=false;
if (RadioGroup7->ItemIndex==2) {k++;} }
}
//-----
void __fastcall TForm7::Button9Click(TObject *Sender)
{
if (RadioGroup9->ItemIndex==-1) { ShowMessage("Выбирете ответ");} else {
TabSheet10->Show();
TabSheet9->Enabled=false;
if (RadioGroup9->ItemIndex==1) {k++;} }
}
//-----
void __fastcall TForm7::Button10Click(TObject *Sender)
{
if (RadioGroup10->ItemIndex==-1) { ShowMessage("Выбирете ответ");} else {
Form8->Show();
TabSheet10->Enabled=false;
if (RadioGroup10->ItemIndex==2) {k++;}
if (k==0) {n=2;}
if (k==1) {n=2;}
if (k==2) {n=2;}
if (k==3) {n=2;}
if (k==4) {n=2;}
if (k==5) {n=3;}
if (k==6) {n=3;}
if (k==7) {n=4;}
if (k==8) {n=4;}

```

```

if (k==9) {n=5;}
if (k==10) {n=5;}
Form8->Label1->Caption=k;
Form8->Label2->Caption=n;
}
}
//-----

```

Листинг 8. Переход к практическим заданиям

```

#include <vcl.h>
#pragma hdrstop
#include "Unit7.h"
#include "Unit8.h"
#include "Unit4.h"
#include "Unit1.h"
#include "Unit3.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm8 *Form8;
//-----
__fastcall TForm8::TForm8(TComponent* Owner)
    : TForm(Owner)
{
Form8->Position=poScreenCenter;
}
//-----
void __fastcall TForm8::Button1Click(TObject *Sender)
{

```

```

Form7->RadioGroup1->ItemIndex=-1;
Form7->RadioGroup2->ItemIndex=-1;
Form7->RadioGroup3->ItemIndex=-1;
Form7->RadioGroup4->ItemIndex=-1;
Form7->RadioGroup5->ItemIndex=-1;
Form7->RadioGroup6->ItemIndex=-1;
Form7->RadioGroup7->ItemIndex=-1;
Form7->RadioGroup8->ItemIndex=-1;
Form7->RadioGroup9->ItemIndex=-1;
Form7->RadioGroup10->ItemIndex=-1;
Form7->Close();
Form8->Close();
Form3->Show();
}
//-----
void __fastcall TForm8::Button2Click(TObject *Sender)
{
Form7->RadioGroup1->ItemIndex=-1;
Form7->RadioGroup2->ItemIndex=-1;
Form7->RadioGroup3->ItemIndex=-1;
Form7->RadioGroup4->ItemIndex=-1;
Form7->RadioGroup5->ItemIndex=-1;
Form7->RadioGroup6->ItemIndex=-1;
Form7->RadioGroup7->ItemIndex=-1;
Form7->RadioGroup8->ItemIndex=-1;
Form7->RadioGroup9->ItemIndex=-1;
Form7->RadioGroup10->ItemIndex=-1;
Form7->TabSheet1->Show();
Form7->Close();

```

```

Form8->Close();
Form1->Show();
}
//-----

```

Листинг 9. Теоретический раздел

```

//-----
#include <vcl.h>
#pragma hdrstop
#include "Unit10.h"
#include "Unit11.h"
#include "Unit14.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm10 *Form10;
//-----
__fastcall TForm10::TForm10(TComponent* Owner)
    : TForm(Owner)
{
Form10->Position=poScreenCenter;
}
//-----
void __fastcall TForm10::Button1Click(TObject *Sender)
{
Form14->Visible=true;
setlocale(LC_ALL,"RUS");
Form14->Canvas->Font->Color=clBlack;
Form14->Canvas->Font->Size=14;
}

```

Form14->Canvas->TextOut(25,45," Рассмотрим некоторые операторы, необходимые для решения задач ");

Form14->Canvas->Font->Size=12;

Form14->Canvas->TextOut(25,95," \ - используется для получения целой части от деления ");

Form14->Canvas->TextOut(25,125,"% - используется для получения остатка части от деления ");

Form14->Canvas->TextOut(25,155,"cout<< - вывод на экран. cout<<endl; - вывод на экран с новой строки ");

Form14->Canvas->TextOut(25,185,"Выводить на экран можно и сообщения, грубо говоря цитаты, они записываются в кавычках ");

Form14->Canvas->TextOut(25,215,"cout<<"Введите первое число :"); Также на экран выводятся переменные cout<<a; ");

Form14->Canvas->TextOut(25,245,"cin>>; - ввод с клавиатуры, Readln(); - ввод с клавиатуры в новой строке. ");

Form14->Canvas->TextOut(25,275,"С помощью этого оператора мы будем вводить значения, она будет присвоена переменной");

Form14->Canvas->TextOut(25,305,"abs() - модуль (абсолютная величина); abs(x1-x2); ");

Form14->Canvas->TextOut(25,335,"sqr() - квадрат; sqr(x);");

Form14->Canvas->TextOut(25,365,"sqrt() - квадратный корень; sqrt(a*a+b*b);");

Form14->Canvas->TextOut(25,395,"sin() - синус числа; sin(x)");

Form14->Canvas->TextOut(25,425,"cos() - косинус числа; cos(x+y);");

Form14->Canvas->TextOut(25,455,"arctan() - арктангенс числа; arctan(1-x/y)");

```

Form14->Canvas->TextOut(25,485,"log() - натуральный логарифм;
log10(z); ");
Form14->Canvas->TextOut(25,515,"exp(x) - экспонента натуральный
антилогарифм; exp(ln(x)/3); ");
Form14->Canvas->TextOut(25,545,"pow(n,q) - степень q, числа n");
Form14->Canvas->TextOut(25,575,"tg(x)=sin(x)/cos(x); функции тангенс в C++
нет ");
Form10->Hide();
Form14->Show();
}
//-----
void __fastcall TForm10::FormCanResize(TObject *Sender, int &NewWidth,
int &NewHeight, bool &Resize)
{
Resize = False;
}
//-----

```

Листинг 10. Теоретический раздел.

```

#include <vcl.h>
#pragma hdrstop
#include "Unit11.h"
#include "Unit13.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm11 *Form11;
//-----

```

```

__fastcall TForm11::TForm11(TComponent* Owner)
    : TForm(Owner)
{
Form11->Position=poScreenCenter;
}
//-----
void __fastcall TForm11::Button1Click(TObject *Sender)
{
Form13->Visible=true;
setlocale(LC_ALL,"RUS");
Form13->Canvas->Font->Color=clBlack;
Form13->Canvas->Font->Size=14;
Form13->Canvas->TextOut(25,15,"          Теперь напишем программу по
составленной блок-схеме");
Form13->Canvas->TextOut(25,45,"          Написать программу решения
примера c=a*b+d ");
Form13->Canvas->Font->Size=12;
Form13->Canvas->TextOut(25,80,"  Программа всегда начинается с
зарезервированного слова #Pragma.          #Pragma one; ");
Form13->Canvas->TextOut(25,105," Впрочем оно не является обязательным,
программа будет работать и          #include<iostream.h>; ");
Form13->Canvas->TextOut(25,130," без него, но для удобства мы можем
использовать '#Pragma'. Через          int a,b,c,d; ");
Form13->Canvas->TextOut(25,155," пробел мы записываем имя программы,
которое не должно совпадать          { ");
Form13->Canvas->TextOut(25,180," с операторами в коде программы и
другими зарезервированными словами.          cout>>'a= '; cin>>a; ");
Form13->Canvas->TextOut(25,205," Назовём программу #Pragma name;'
cout<<'b='; cin>>b;");

```

Form13->Canvas->TextOut(25,230," Можно подключить различные модули, '#include<>'; cout<>'d='; cin<>d; ");

Form13->Canvas->TextOut(25,255," необходим для операторов задержки экрана getch(); c=a*b+d ");

Form13->Canvas->TextOut(25,280," или оператор system("pause"). cout<<'c='; cin<>c; ");

Form13->Canvas->TextOut(25,305," Раздел описания переменных. Здесь описываются getch(); ");

Form13->Canvas->TextOut(25,330," абсолютно все переменные и их типы данных которые мы будем . } ");

Form13->Canvas->TextOut(25,355," использовать в программе ");

Form13->Canvas->TextOut(25,380," Например у нас все переменные целочисленные, мы не используем деление, вещественные");

Form13->Canvas->TextOut(25,405," (дробные числа) для которых нужен тип данных float (вещественный тип).");

Form13->Canvas->TextOut(25,430," Введём значения переменных с клавиатуры, с помощью оператора 'cin>>';, а для удобства");

Form13->Canvas->TextOut(25,455," пользователя, чтобы он видел что вводит выведем сообщение: 'введите c' или как в нашем");

Form13->Canvas->TextOut(25,480," случае 'c=', для этого используется оператор 'cout<<'; После введения всех трёх");

Form13->Canvas->TextOut(25,505," переменных выполним вычисления используя оператор присваивания 'c=a*b+d',");

Form13->Canvas->TextOut(25,530," выведем сообщения на экран ответ 'cout<<'c='<<endl; cin<>c;'.");

Form13->Canvas->TextOut(25,555," И завершиться выполнение программы оператором ");

Form13->Canvas->TextOut(25,580," '}'");


```

Form11->Hide();
Form13->Show();
}
//-----
void __fastcall TForm11::FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
Resize = False;
}
//-----

```

Листинг 11. Форма о программе

```

//-----
#include <vcl.h>
#pragma hdrstop

#include "Unit12.h"
//-----
#pragma resource "*.dfm"
TAboutBox *AboutBox;
//-----
__fastcall TAboutBox::TAboutBox(TComponent* AOwner)
    : TForm(AOwner)
{
AboutBox->Position=poScreenCenter;
}
//-----
void __fastcall TAboutBox::OKButtonClick(TObject *Sender)
{

```

```

AboutBox->Close();
}
//-----
Листинг 12. Теоретический раздел
#include <vcl.h>
#pragma hdrstop
#include "Unit13.h"
#include "Unit19.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm13 *Form13;
//-----
__fastcall TForm13::TForm13(TComponent* Owner)
    : TForm(Owner)
{
Form13->Position=poScreenCenter;
}
//-----
void __fastcall TForm13::Button1Click(TObject *Sender)
{
Form13->Hide();
Form19->Visible=true;
setlocale(LC_ALL,"RUS");
Form19->Canvas->Font->Color=clBlack;
Form19->Canvas->Font->Size=14;
Form19->Canvas->TextOut(25,25,"Ещё раз посмотрим код нашей програамы
с комментариями ");
}

```

```

Form19->Canvas->TextOut(25,50,"всё что записывается в программе после
знака / является комментарием ");
Form19->Canvas->TextOut(25,75,"и программой не учитывается ");
Form19->Canvas->TextOut(25,150," #Pragma one; / Имя программы");
Form19->Canvas->TextOut(25,175," #include<iostream.h> / подключение
модулей экрана ; ");
Form19->Canvas->TextOut(25,200," int a,b,c,d; / раздел описания
переменных");
Form19->Canvas->TextOut(25,225," { / начало программы");
Form19->Canvas->TextOut(25,250," cout<<'a='; cin>>a; / вывод на экран
сообщения: 'a=', и ввод числа в переменную 'a'");
Form19->Canvas->TextOut(25,275," cout<<'b='; cin>>b; / вывод на экран
сообщения: 'b=', и ввод числа в переменную 'b'");
Form19->Canvas->TextOut(25,300," cout<<'d='; cin>>d; / вывод на экран
сообщения: 'd=', и ввод числа в переменную 'd' ");
Form19->Canvas->TextOut(25,325," c=a*b+d / присваивание к переменной 'c'
выражения, вычисление ");
Form19->Canvas->TextOut(25,350," cout<<'c='; cin>>c; / вывод на экран
сообщения: 'c=' и вывод на экран ответа ");
Form19->Canvas->TextOut(25,375," system("pause"); / задержка экрана");
Form19->Canvas->TextOut(25,400," } / конец. ");
Form19->Canvas->TextOut(25,475," Задание: ");
Form19->Canvas->TextOut(25,500," 1. Наберите код этой программы в
среде Borland C++ Builder; ");
Form19->Canvas->TextOut(25,525," 2. Составьте блок-схему и напишите
программу решения примера  $P=2x(a+b)$ ; ");
Form19->Canvas->TextOut(25,550," 3. Напишите программу перевода
доллара в рубли. За курс доллара примим 30 рублей.");
Form19->Show();

```

```

}
//-----
void __fastcall TForm13::FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
    Resize = False;
}

```

Листинг13. Теоретический раздел

```

//-----
#include <vcl.h>
#pragma hdrstop

#include "Unit14.h"
#include "Unit11.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm14 *Form14;
//-----
__fastcall TForm14::TForm14(TComponent* Owner)
    : TForm(Owner)
{
    Form14->Position=poScreenCenter;
}
//-----
void __fastcall TForm14::Button1Click(TObject *Sender)
{
    Form11->Visible=true;
}

```

```

setlocale(LC_ALL,"RUS");
Form11->Canvas->Font->Color=clBlack;
Form11->Canvas->Font->Size=14;
Form11->Canvas->TextOut(25,45,"Алгоритм, в котором команды
выполняются последовательно одна за другой,");
Form11->Canvas->TextOut(25,75,"называется линейным алгоритмом.");
Form11->Canvas->TextOut(25,115,"      Давайте рассмотрим пример
решения задачи");
Form11->Canvas->TextOut(25,145,"      Написать программу решения
примера  $c=a*b+d$  ");
Form11->Canvas->Font->Size=12;
Form11->Canvas->TextOut(25,185,"      Прежде чем приступить к написанию
программы, сначала");
Form11->Canvas->TextOut(25,215," следует составить блок-схему      ");
Form11->Canvas->TextOut(25,245,"      Сначала мы должны
проанализировать задачу, сколько ");
Form11->Canvas->TextOut(25,275," нам нужно переменных. В данном случае
нам нужно 4 переменные.");
Form11->Canvas->TextOut(25,305,"      Чтобы работать с исходными данными,
их нужно ввести с кла-");
Form11->Canvas->TextOut(25,335," виатуры. В паралделограмме мы
записываем переменные, в которые");
Form11->Canvas->TextOut(25,365," будут вводиться значения, т.е. a, b, d.
");
Form11->Canvas->TextOut(25,395,"      Теперь можно выполнять действия.
Присваиваем к переменной");
Form11->Canvas->TextOut(25,425," 'c' выражение  $a*b+d$ . На языке
программирования C++");

```

```

Form11->Canvas->TextOut(25,455," это будет выглядеть так: c=a*b+d.
Необходимо выделить, что ");
Form11->Canvas->TextOut(25,485," действия в программе записываютс в
прямоугольнике.");
Form11->Canvas->TextOut(25,515,"    Осталось вывести ответ, вывод
данных на экран, записывается");
Form11->Canvas->TextOut(25,545," в параллелограмме. Выводим
переменную 'c'. Конец программы");
Form14->Hide();
Form11->Show();
}
//-----
void __fastcall TForm14::FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
    Resize = False;
}
//-----
Листинг 14. Теоретический раздел.
#include <vcl.h>
#pragma hdrstop
#include "Unit16.h"
#include "Unit17.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm16 *Form16;
//-----
__fastcall TForm16::TForm16(TComponent* Owner)

```

```

        : TForm(Owner)
    {
Form16->Position=poScreenCenter;
    }
//-----
void __fastcall TForm16::Button1Click(TObject *Sender)
{
Form17->Visible=true;
setlocale(LC_ALL,"RUS");
Form17->Canvas->Font->Color=clBlack;
Form17->Canvas->Font->Size=14;
Form17->Canvas->TextOut(25,25,"Рассмотрим в качестве примера
разветвляющийся алгоритм,");
Form17->Canvas->TextOut(25,50,"изображенный в виде блок-схемы.");
Form17->Canvas->Font->Size=12;
Form17->Canvas->TextOut(25,100,"    Аргументами этого алгоритма
являются две переменные А, В, а результатом - переменная Х");
Form17->Canvas->TextOut(25,125,"Если условие  $A \geq B$  истинно, то
выполняется команда  $X=A*B$ , в противном случае выполняется ");
Form17->Canvas->TextOut(25,150,"команда  $X=A+B$ . В результате
печатается то значение переменной Х, которое она получает");
Form17->Canvas->TextOut(25,175,"в результате выполнения одной из серий
команд.");
Form17->Canvas->TextOut(285,300,"#Pragma");
Form17->Canvas->TextOut(285,325,"#include<iostream.h>");
Form17->Canvas->TextOut(285,350,"int a,b,x;");
Form17->Canvas->TextOut(285,375,"{ ");
Form17->Canvas->TextOut(285,400,"cout<<'введите a,b'; ");
Form17->Canvas->TextOut(285,425,"cin>>a,b; ");

```

```

Form17->Canvas->TextOut(285,450,"if (a>=b){x=a*b;} else {x=a+b;}");
Form17->Canvas->TextOut(285,475,"cout<<'x= ', x;");
Form17->Canvas->TextOut(285,500,"system("pause");");
Form17->Canvas->TextOut(285,525,"}");
Form16->Hide();
Form17->Show();
}
//-----
void __fastcall TForm16::FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
    Resize = False;
}
//-----

```

Листинг 17. Теоретический раздел.

```

#include <vcl.h>
#pragma hdrstop
#include "Unit17.h"
#include "Unit22.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm17 *Form17;
//-----
__fastcall TForm17::TForm17(TComponent* Owner)
    : TForm(Owner)
{
    Form17->Position=poScreenCenter;

```



```

}
//-----
void __fastcall TForm17::Button1Click(TObject *Sender)
{
Form17->Hide();
Form22->Visible=true;
setlocale(LC_ALL,"RUS");
Form22->Canvas->Font->Color=clBlack;
Form22->Canvas->Font->Size=14;
Form22->Canvas->TextOut(25,25,"    Оператор выбора");
Form22->Canvas->Font->Size=12;
Form22->Canvas->TextOut(25,75,"    Оператор case используется для выбора
одного из нескольких направлений дальнейшего хода ");
Form22->Canvas->TextOut(25,100,"программы. Этот оператор имеет вид:");
Form22->Canvas->TextOut(25,125,"    switch (p) {  };
Form22->Canvas->TextOut(25,150,"        case a: s1; break;  ");
Form22->Canvas->TextOut(25,175,"        case b: s2; break;  ");
Form22->Canvas->TextOut(25,200,"        ..      ");
Form22->Canvas->TextOut(25,225,"        case n: sn;  break; ");
Form22->Canvas->TextOut(25,250,"        default:    ; break;");
Form22->Canvas->TextOut(25,275,"    }      ");
Form22->Canvas->TextOut(25,300,"Если значение выражения p не совпадает
ни с одной из констант выбора, выполняется оператор s n+1,");
Form22->Canvas->TextOut(25,325,"содержащийся после ключевого слова
default, причем ветвь else в операторе case необязательна.");
Form22->Canvas->TextOut(25,350,"    Нужно заметить, что оператор выбора
Switch Case, работает только с типами данных integer и char");
Form22->Show();
}

```

```
//-----
void __fastcall TForm17::FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
    Resize = False;
}
//-----
```

Листинг 15. Тестирование 2.

```
#include <vcl.h>
#pragma hdrstop
#include "Unit20.h"
#include "Unit21.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm20 *Form20;
int i,n,k;
//-----
__fastcall TForm20::TForm20(TComponent* Owner)
    : TForm(Owner)
{
    Form20->Position=poScreenCenter;
}
//-----
void __fastcall TForm20::Button1Click(TObject *Sender)
{
    k=0;
```

```

if ((Edit1->Text=="") && (Edit2->Text=="") && (Edit3->Text=="") && (Edit4-
>Text=="") && (Edit5->Text==""))
    {ShowMessage("Выберете ответ");} else {
TabSheet2->Show();
TabSheet1->Enabled=false;
if (Edit1->Text=="4" && Edit2->Text=="1" && Edit3->Text=="5" && Edit4-
>Text=="2" && Edit5->Text=="3") {k++;}
}
}
//-----
void __fastcall TForm20::Button2Click(TObject *Sender)
{
if (RadioGroup1->ItemIndex==-1) {ShowMessage("Выберете ответ");} else {
TabSheet3->Show();
TabSheet2->Enabled=false;
if (RadioGroup1->ItemIndex==3) {k++;} }
}
//-----
void __fastcall TForm20::Button3Click(TObject *Sender)
{
if (RadioGroup2->ItemIndex==-1) {ShowMessage("Выберете ответ");} else {
TabSheet4->Show();
TabSheet3->Enabled=false;
if (RadioGroup2->ItemIndex==0) {k++;} }
}
//-----
void __fastcall TForm20::Button4Click(TObject *Sender)
{
if (RadioGroup3->ItemIndex==-1) {ShowMessage("Выберете ответ");} else {

```

```

TabSheet5->Show();
TabSheet4->Enabled=false;
if (RadioGroup3->ItemIndex==4) {k++;} }
}
//-----
void __fastcall TForm20::Button5Click(TObject *Sender)
{
if (RadioGroup4->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet6->Show();
TabSheet5->Enabled=false;
if (RadioGroup4->ItemIndex==3) {k++;} }
}
//-----
void __fastcall TForm20::Button6Click(TObject *Sender)
{
if (RadioGroup5->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet7->Show();
TabSheet6->Enabled=false;
if (RadioGroup5->ItemIndex==1) {k++;} }
}
//-----
void __fastcall TForm20::Button7Click(TObject *Sender)
{
if (RadioGroup6->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet8->Show();
TabSheet7->Enabled=false;
if (RadioGroup6->ItemIndex==4) {k++;} }
}
//-----

```

```

void __fastcall TForm20::Button8Click(TObject *Sender)
{
if (Edit6->Text=="")
    {ShowMessage("Выбирете ответ");} else {
TabSheet9->Show();
TabSheet8->Enabled=false;
if (Edit6->Text=="8") {k++;}
}}
//-----
void __fastcall TForm20::Button9Click(TObject *Sender)
{
if (Edit7->Text=="")
    {ShowMessage("Выбирете ответ");} else {
TabSheet9->Enabled=false;
if (Edit7->Text=="2" ) {k++;}
if (k==0) {n=2;}
if (k==1) {n=2;}
if (k==2) {n=2;}
if (k==3) {n=2;}
if (k==4) {n=3;}
if (k==5) {n=3;}
if (k==6) {n=4;}
if (k==7) {n=4;}
if (k==8) {n=5;}
if (k==9) {n=5;}
Form21->Show();
Form21->Label1->Caption=k;
Form21->Label2->Caption=n;
Form20->Show();
}
}

```

```

}
}
//-----
void __fastcall TForm20::FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
Resize = False;
}
//-----

```

Листинг 16. Теоретический раздел.

```

#include <vcl.h>
#pragma hdrstop
#include "Unit22.h"
#include "Unit23.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm22 *Form22;
//-----
__fastcall TForm22::TForm22(TComponent* Owner)
    : TForm(Owner)
{
Form22->Position=poScreenCenter;
}
//-----
void __fastcall TForm22::Button1Click(TObject *Sender)
{
Form22->Hide();
}

```

```

Form23->Visible=true;
setlocale(LC_ALL,"RUS");
Form23->Canvas->Font->Color=clBlack;
Form23->Canvas->Font->Size=14;
Form23->Canvas->TextOut(25,25,"Рассмотрим пример. Введите номер месяца,
и выведите время года");
Form23->Canvas->TextOut(25,50,"в которое входит месяц. Т.е. если ввести 1,
2 или 12 вывести на экран: Зима.");
Form23->Canvas->Font->Size=12;
Form23->Canvas->TextOut(25,100,"#include <iostream.h>");
Form23->Canvas->TextOut(25,125,"int main()");
Form23->Canvas->TextOut(25,150," int month; //номер месяца");
Form23->Canvas->TextOut(25,175,"{ ");
Form23->Canvas->TextOut(25,200," cout<<'Введите номер месяца:;');");
Form23->Canvas->TextOut(25,225," cin>>month; ");
Form23->Canvas->TextOut(25,250," cout<<'Время года:; ');");
Form23->Canvas->TextOut(25,275," switch (month)");
Form23->Canvas->TextOut(25,300," case 1, 2, 12: cout<<'зима'; break; ");");
Form23->Canvas->TextOut(25,325," case 3..5: cout<<'весна'; break; ");");
Form23->Canvas->TextOut(25,350," case 6..8: cout<<'лето'; break; ");");
Form23->Canvas->TextOut(25,375," case 9..11: cout<<'осень'; break;");");
Form23->Canvas->TextOut(25,400," default: cout<<'число должно быть от
1 до 12'; break;");");
Form23->Canvas->TextOut(25,425," }");");
Form23->Canvas->TextOut(25,450," } ");");
Form23->Canvas->TextOut(25,475,"Ответ выведется в зависимости от
вводимых нами значений в переменную month.");");
Form23->Canvas->TextOut(25,500,"Заметьте что варианты вводимых
значений можно записывать по несколько через запятую, ");");

```

```

Form23->Canvas->TextOut(25,525,"а также через троиточие если значение
продолжают друг друг, т.е. что бы вывести 'весна,");
Form23->Canvas->TextOut(25,550,"нам нужно ввести 3,4,5, но поскольку
здесь наблюдается последовательность, то можно записать ");
Form23->Canvas->TextOut(25,575,"не записывать значения находящиеся
между первой и последней цифрой - 3..5.");
Form23->Show();
}
//-----
void __fastcall TForm22::FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
Resize = False;
}
//-----

```

Листинг 17. Практическое задание по размещению блоков блок-схемы.

```

#include <vcl.h>
#pragma hdrstop
#include "Unit3.h"
#include "Unit25.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm25 *Form25;
int OldX;
int OldY;
bool IsMouseDown=false;
//-----

```



```

__fastcall TForm25::TForm25(TComponent* Owner)
    : TForm(Owner)
{
Form25->Position=poScreenCenter;
}
//-----
void __fastcall TForm25::Image1MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
OldX=X;
    OldY=Y;
    IsMouseDown=true;
    for (Image1->Left==440; Image1->Left==550; Image1->Left++ )
    for (Image1->Top==60; Image1->Top==100; Image1->Top++ )
        {IsMouseDown=false; Image1->Top=80; Image1->Left=480;}
}
//-----
void __fastcall TForm25::Image1MouseMove(TObject *Sender,
    TShiftState Shift, int X, int Y)
{
if(IsMouseDown)
    {
        Image1->Left+=X-OldX;
        Image1->Top+=Y-OldY;
    }
}
//-----
void __fastcall TForm25::Image1MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)

```

```

{
    IsMouseDown=false;
    if (Image1->Top>Image2->Top & Image1->Left>Image2->Left & Image1-
>Top<Image2->Top+Height & Image1->Left<Image2->Left+Width) {
        Image1->Visible=False;Image2->Visible=True;
    } }
//-----
void __fastcall TForm25::Image3MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    OldX=X;
    OldY=Y;
    IsMouseDown=true;

}
//-----
void __fastcall TForm25::Image3MouseMove(TObject *Sender,
    TShiftState Shift, int X, int Y)
{
    if(IsMouseDown)
    {
        Image3->Left+=X-OldX;
        Image3->Top+=Y-OldY;
    }
}
//-----
void __fastcall TForm25::Image3MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{

```

```

IsMouseDown=false;
    if (Image3->Top>Image4->Top & Image3->Left>Image4->Left & Image3-
>Top<Image4->Top+Height & Image3->Left<Image4->Left+Width) {
        Image3->Visible=False;Image4->Visible=True;
    }
}
//-----

void __fastcall TForm25::Image5MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    OldX=X;
    OldY=Y;
    IsMouseDown=true;
}
//-----

void __fastcall TForm25::Image5MouseMove(TObject *Sender,
    TShiftState Shift, int X, int Y)
{
    if(IsMouseDown)
    {
        Image5->Left+=X-OldX;
        Image5->Top+=Y-OldY;
    }
}
//-----

void __fastcall TForm25::Image5MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{

```

```

IsMouseDown=false;
    if (Image5->Top>Image7->Top & Image5->Left>Image7->Left & Image5-
>Top<Image7->Top+Height & Image5->Left<Image7->Left+Width) {
        Image5->Visible=False;Image6->Visible=True;
    }
}
//-----
void __fastcall TForm25::Image9MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    OldX=X;
    OldY=Y;
    IsMouseDown=true;
}
//-----
void __fastcall TForm25::Image9MouseMove(TObject *Sender,
    TShiftState Shift, int X, int Y)
{
    if(IsMouseDown)
    {
        Image9->Left+=X-OldX;
        Image9->Top+=Y-OldY;
    }
}
//-----
void __fastcall TForm25::Image9MouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    IsMouseDown=false;

```

```

    if (Image9->Top>Image8->Top & Image9->Left>Image8->Left & Image9-
>Top<Image8->Top+Height & Image9->Left<Image8->Left+Width) {
        Image9->Visible=False;Image8->Visible=True;
    }
//-----
void __fastcall TForm25::Button1Click(TObject *Sender)
{
if (Image1->Top>Image2->Top & Image1->Left>Image2->Left & Image1-
>Top<Image2->Top+Height & Image1->Left<Image2->Left+Width & Image9-
>Top>Image8->Top & Image9->Left>Image8->Left & Image9->Top<Image8-
>Top+Height & Image9->Left<Image8->Left+Width & Image3->Top>Image4-
>Top & Image3->Left>Image4->Left & Image3->Top<Image4->Top+Height &
Image3->Left<Image4->Left+Width & Image5->Top>Image7->Top & Image5-
>Left>Image7->Left & Image5->Top<Image7->Top+Height & Image5-
>Left<Image7->Left+Width) {
    ShowMessage("Верно!"); Form25->Hide(); Form3->Show();} else
ShowMessage("Расположите все блоки в правильном порядке!");
}
//-----
void __fastcall TForm25::FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
Resize = False;
}
//-----

```

Листинг 18. Теоретический раздел.

```

#include <vcl.h>
#pragma hdrstop

```

```

#include "Unit27.h"
#include "Unit28.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm27 *Form27;
//-----
__fastcall TForm27::TForm27(TComponent* Owner)
    : TForm(Owner)
{
Form27->Position=poScreenCenter;
}
//-----
void __fastcall TForm27::Button1Click(TObject *Sender)
{
Form27->Hide();
Form28->Visible=true;
setlocale(LC_ALL,"RUS");
Form28->Canvas->Font->Color=clBlack;
Form28->Canvas->Font->Size=14;
Form28->Canvas->TextOut(25,25,"Составить программу вычисления значения
y=1+1/2+1/3+...+1/20.");
Form28->Canvas->TextOut(25,50," #include <iostream.h>");
Form28->Canvas->TextOut(25,75," Uses crt;");
Form28->Canvas->TextOut(25,100," int i;");
Form28->Canvas->TextOut(25,125," float y ");
Form28->Canvas->TextOut(25,150," { ");
Form28->Canvas->TextOut(25,175," y=0 ");
Form28->Canvas->TextOut(25,200," For (i=1; i=20; i++)");
}

```

```

Form28->Canvas->TextOut(25,225," y=y+1/i;    ");
Form28->Canvas->TextOut(25,250," cout<<'y='<<y<<endl;  ");
Form28->Canvas->TextOut(25,275," }    ");
Form28->Canvas->TextOut(25,300," }    ");
Form28->Canvas->Font->Size=12;
Form28->Canvas->TextOut(25,325,"    На самом деле вычисление этой суммы
можно осуществить по очень простому и компактному");
Form28->Canvas->TextOut(25,350,"алгоритму: предварительно положим
y=0 (с помощью оператора присваивания y=0), а затем ");
Form28->Canvas->TextOut(25,375,"выполним оператор присваивания
y= y+1/i для последовательных значений i= 1,2,..., n. ");
Form28->Canvas->TextOut(25,400,"При каждом очередном выполнении этого
оператора к текущему значению 'y' будет прибавляться ");
Form28->Canvas->TextOut(25,425,"очередное слагаемое. Как видно, в этом
случае процесс вычислений будет носить циклический ");
Form28->Canvas->TextOut(25,450,"характер: оператор y= y+1/i должен
выполняться многократно, т.е. циклически, при различных ");
Form28->Canvas->TextOut(25,475,"значениях i.");
Form28->Canvas->TextOut(25,500,"    В некоторых случаях бывает удобно,
чтобы параметр цикла C++ принимал последователь-");
Form28->Canvas->TextOut(25,525,"ные но не возрастающие, а убывающие
значения. Для таких случаев в C++ предусмотрен ");
Form28->Canvas->TextOut(25,550,"оператор цикла с параметром следующего
вида:");
Form28->Canvas->TextOut(25,575,"    For (V= E1;V=E2;i--)");
Form28->Canvas->TextOut(25,600,"где i-- (уменьшаясь к) - служебное
слово");
Form28->Show();
}

```

```
//-----
void __fastcall TForm27::FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
    Resize = False;
}

```

Листинг 19. Тестирование 3

```
#include <vcl.h>
#pragma hdrstop
#include "Unit36.h"
#include "Unit37.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm36 *Form36;
int n,k;
//-----
__fastcall TForm36::TForm36(TComponent* Owner)
    : TForm(Owner)
{
    Form36->Position=poScreenCenter;
}
//-----
void __fastcall TForm36::Button1Click(TObject *Sender)
{
    k==0;
    if (RadioGroup1->ItemIndex==-1) { ShowMessage("Выбирете ответ");} else {
        TabSheet2->Show();
    }
}

```



```

TabSheet1->Enabled=false;
if (RadioGroup1->ItemIndex==1) {k++;}
}
}
//-----

void __fastcall TForm36::Button2Click(TObject *Sender)
{
if (RadioGroup2->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet3->Show();
TabSheet2->Enabled=false;
if (RadioGroup2->ItemIndex==3) {k++;}
}
}
//-----

void __fastcall TForm36::Button3Click(TObject *Sender)
{
if (RadioGroup3->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet4->Show();
TabSheet3->Enabled=false;
if (RadioGroup3->ItemIndex==0) {k++;}
}
}
//-----

void __fastcall TForm36::Button4Click(TObject *Sender)
{
if (RadioGroup4->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet5->Show();
TabSheet4->Enabled=false;

```

```

if (RadioGroup4->ItemIndex==1) {k++;}
} }
//-----
void __fastcall TForm36::Button5Click(TObject *Sender)
{
if (RadioGroup5->ItemIndex==1) {ShowMessage("Выберите ответ");} else {
TabSheet6->Show();
TabSheet5->Enabled=false;
if (RadioGroup5->ItemIndex==2) {k++;}
} }
//-----
void __fastcall TForm36::Button6Click(TObject *Sender)
{
if (RadioGroup6->ItemIndex==1) {ShowMessage("Выберите ответ");} else {
TabSheet7->Show();
TabSheet6->Enabled=false;
if (RadioGroup6->ItemIndex==0) {k++;}
} }
//-----
void __fastcall TForm36::Button7Click(TObject *Sender)
{
if (RadioGroup7->ItemIndex==1) {ShowMessage("Выберите ответ");} else {
TabSheet8->Show();
TabSheet7->Enabled=false;
if (RadioGroup7->ItemIndex==2) {k++;}
} }
//-----
void __fastcall TForm36::Button8Click(TObject *Sender)
{

```

```

if (RadioGroup8->ItemIndex==-1) { ShowMessage("Выбирете ответ");} else {
TabSheet9->Show();
TabSheet8->Enabled=false;
if (RadioGroup8->ItemIndex==2) {k++;}
} }
//-----
void __fastcall TForm36::Button9Click(TObject *Sender)
{
if (RadioGroup9->ItemIndex==-1) { ShowMessage("Выбирете ответ");} else {
TabSheet10->Show();
TabSheet9->Enabled=false;
if (RadioGroup9->ItemIndex==3) {k++;}
} }
//-----
void __fastcall TForm36::Button10Click(TObject *Sender)
{
if (RadioGroup10->ItemIndex==-1) { ShowMessage("Выбирете ответ");} else {
TabSheet11->Show();
TabSheet10->Enabled=false;
if (RadioGroup10->ItemIndex==0) {k++;}
} }
//-----
void __fastcall TForm36::Button11Click(TObject *Sender)
{
if (Edit1->Text=="") { ShowMessage("Введите ответ");} else {
TabSheet12->Show();
TabSheet11->Enabled=false;
if (Edit1->Text=="-4") {k++;}
} }

```

```

//-----
void __fastcall TForm36::Button12Click(TObject *Sender)
{
if (RadioGroup11->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet13->Show();
TabSheet12->Enabled=false;
if (RadioGroup11->ItemIndex==3) {k++;}
} }
//-----
void __fastcall TForm36::Button13Click(TObject *Sender)
{
if (Edit2->Text=="") {ShowMessage("Введите ответ");} else {
TabSheet14->Show();
TabSheet13->Enabled=false;
if (Edit2->Text=="4") {k++;}
}
}
//-----
void __fastcall TForm36::Button14Click(TObject *Sender)
{
if (Edit3->Text=="") {ShowMessage("Введите ответ");} else {
TabSheet15->Show();
TabSheet14->Enabled=false;
if (Edit3->Text=="7") {k++;}
}
}
//-----
void __fastcall TForm36::Button15Click(TObject *Sender)
{

```

```

if (Edit4->Text=="") {ShowMessage("Введите ответ");} else {
Form37->Show();
TabSheet15->Enabled=false;
if (Edit4->Text=="14") {k++;}
if (k==0) {n=2;}
if (k==1) {n=2;}
if (k==2) {n=2;}
if (k==3) {n=2;}
if (k==4) {n=2;}
if (k==5) {n=2;}
if (k==6) {n=3;}
if (k==7) {n=3;}
if (k==8) {n=3;}
if (k==9) {n=3;}
if (k==10) {n=4;}
if (k==11) {n=4;}
if (k==12) {n=4;}
if (k==13) {n=4;}
if (k==14) {n=5;}
if (k==15) {n=5;}
Form37->Label1->Caption=k;
Form37->Label2->Caption=n;
}
}
//-----
void __fastcall TForm36::FormCanResize(TObject *Sender, int &NewWidth,
int &NewHeight, bool &Resize)
{
Resize = False;

```

```
}

```

Листинг 20. Тестирование 4

```
#include <vcl.h>
#pragma hdrstop
#include <fstream.h>
#include "Unit38.h"
#include "Unit39.h"
#include "Unit2.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm38 *Form38;
int k, n;
char s[100];
//-----
__fastcall TForm38::TForm38(TComponent* Owner)
    : TForm(Owner)
{
Form38->Position=poScreenCenter;
}
//-----

void __fastcall TForm38::Button1Click(TObject *Sender)
{
k=0;
if (RadioGroup1->ItemIndex==-1) { ShowMessage("Выбирете ответ");} else {
TabSheet2->Show();
```

```

TabSheet1->Enabled=false;
if (RadioGroup1->ItemIndex==2) {k++;}
}
}
//-----

void __fastcall TForm38::Button2Click(TObject *Sender)
{
if (RadioGroup2->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet3->Show();
TabSheet2->Enabled=false;
if (RadioGroup2->ItemIndex==1) {k++;}
}
}
//-----

void __fastcall TForm38::Button3Click(TObject *Sender)
{
if (RadioGroup3->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet4->Show();
TabSheet3->Enabled=false;
if (RadioGroup3->ItemIndex==1) {k++;}
}
}
//-----

void __fastcall TForm38::Button4Click(TObject *Sender)
{
if (RadioGroup4->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet5->Show();
TabSheet4->Enabled=false;

```

```

if (RadioGroup4->ItemIndex==0) {k++;}
}
}
//-----
void __fastcall TForm38::Button5Click(TObject *Sender)
{
if (RadioGroup5->ItemIndex==-1) {ShowMessage("Выберете ответ");} else {
TabSheet6->Show();
TabSheet5->Enabled=false;
if (RadioGroup5->ItemIndex==0) {k++;}
}
}
//-----
void __fastcall TForm38::Button6Click(TObject *Sender)
{
if (RadioGroup6->ItemIndex==-1) {ShowMessage("Выберете ответ");} else {
TabSheet7->Show();
TabSheet6->Enabled=false;
if (RadioGroup6->ItemIndex==2) {k++;}
}
}
//-----
void __fastcall TForm38::Button7Click(TObject *Sender)
{
if (RadioGroup7->ItemIndex==-1) {ShowMessage("Выберете ответ");} else {
TabSheet8->Show();
TabSheet7->Enabled=false;
if (RadioGroup7->ItemIndex==2) {k++;}
}
}

```



```

}
//-----
void __fastcall TForm38::Button8Click(TObject *Sender)
{
if (RadioGroup8->ItemIndex==-1) { ShowMessage("Выберите ответ"); } else {
TabSheet9->Show();
TabSheet8->Enabled=false;
if (RadioGroup8->ItemIndex==2) {k++;}
}
}
//-----
void __fastcall TForm38::Button9Click(TObject *Sender)
{
if (RadioGroup9->ItemIndex==-1) { ShowMessage("Выберите ответ"); } else {
TabSheet10->Show();
TabSheet9->Enabled=false;
if (RadioGroup9->ItemIndex==0) {k++;}
}
}
//-----
void __fastcall TForm38::Button10Click(TObject *Sender)
{
if (RadioGroup10->ItemIndex==-1) { ShowMessage("Выберите ответ"); } else {
TabSheet11->Show();
TabSheet10->Enabled=false;
if (RadioGroup10->ItemIndex==1) {k++;}
}
}
//-----

```

```

void __fastcall TForm38::Button11Click(TObject *Sender)
{
if (RadioGroup11->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet12->Show();
TabSheet11->Enabled=false;
if (RadioGroup11->ItemIndex==3) {k++;}
}
}
//-----
void __fastcall TForm38::Button12Click(TObject *Sender)
{
if (RadioGroup12->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet13->Show();
TabSheet12->Enabled=false;
if (RadioGroup12->ItemIndex==2) {k++;}
}
}
//-----
void __fastcall TForm38::Button13Click(TObject *Sender)
{
if (RadioGroup13->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet14->Show();
TabSheet13->Enabled=false;
if (RadioGroup13->ItemIndex==3) {k++;}
}
}
//-----
void __fastcall TForm38::Button14Click(TObject *Sender)
{

```

```

if (RadioGroup14->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet15->Show();
TabSheet14->Enabled=false;
if (RadioGroup14->ItemIndex==0) {k++;}
}
}
//-----
void __fastcall TForm38::Button15Click(TObject *Sender)
{
if (RadioGroup15->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet16->Show();
TabSheet15->Enabled=false;
if (RadioGroup15->ItemIndex==3) {k++;}
}
}
//-----
void __fastcall TForm38::Button16Click(TObject *Sender)
{
if (RadioGroup16->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet17->Show();
TabSheet16->Enabled=false;
if (RadioGroup16->ItemIndex==3) {k++;}
}
}
//-----
void __fastcall TForm38::Button17Click(TObject *Sender)
{
if (RadioGroup17->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet18->Show();

```

```

TabSheet17->Enabled=false;
if (RadioGroup17->ItemIndex==4) {k++;}
}}
//-----
void __fastcall TForm38::Button18Click(TObject *Sender)
{
if (RadioGroup18->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet19->Show();
TabSheet18->Enabled=false;
if (RadioGroup18->ItemIndex==3) {k++;}
}}
//-----
void __fastcall TForm38::Button19Click(TObject *Sender)
{
if (RadioGroup19->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet20->Show();
TabSheet19->Enabled=false;
if (RadioGroup19->ItemIndex==2) {k++;}
}}
//-----
void __fastcall TForm38::Button20Click(TObject *Sender)
{
if (RadioGroup20->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet21->Show();
TabSheet20->Enabled=false;
if (RadioGroup20->ItemIndex==3) {k++;}
}
}
//-----

```

```

void __fastcall TForm38::Button22Click(TObject *Sender)
{
if (RadioGroup21->ItemIndex==-1) {ShowMessage("Выбирете ответ");} else {
TabSheet23->Show();
TabSheet22->Enabled=false;
if (RadioGroup21->ItemIndex==0) {k++;}
}}
//-----
void __fastcall TForm38::Button21Click(TObject *Sender)
{
if (Edit1->Text=="") {ShowMessage("Выбирете ответ");} else {
TabSheet22->Show();
TabSheet21->Enabled=false;
if (Edit1->Text=="16") {k++;}
}}
//-----
void __fastcall TForm38::Button23Click(TObject *Sender)
{
if (Edit2->Text=="") {ShowMessage("Выбирете ответ");} else {
TabSheet24->Show();
TabSheet23->Enabled=false;
if (Edit2->Text=="11") {k++;}}
}
//-----
void __fastcall TForm38::Button24Click(TObject *Sender)
{
if (Edit3->Text=="") {ShowMessage("Выбирете ответ");} else {
TabSheet25->Show();

```

```
TabSheet24->Enabled=false;
if (Edit3->Text=="-2") {k++;}
}    }
//-----
void __fastcall TForm38::Button25Click(TObject *Sender)
{
if (Edit4->Text=="") {ShowMessage("Выбирете ответ");} else {
TabSheet25->Enabled=false;
if (Edit4->Text=="9") {k++;}
if (k==0) {n=2;}
if (k==1) {n=2;}
if (k==2) {n=2;}
if (k==3) {n=2;}
if (k==4) {n=2;}
if (k==5) {n=2;}
if (k==6) {n=2;}
if (k==7) {n=2;}
if (k==8) {n=2;}
if (k==9) {n=3;}
if (k==10) {n=3;}
if (k==11) {n=3;}
if (k==12) {n=3;}
if (k==13) {n=3;}
if (k==14) {n=3;}
if (k==15) {n=3;}
if (k==16) {n=4;}
if (k==17) {n=4;}
if (k==18) {n=4;}
if (k==19) {n=4;}
```

```

if (k==20) {n=4;}
if (k==21) {n=5;}
if (k==22) {n=5;}
if (k==23) {n=5;}
if (k==24) {n=5;}
if (k==25) {n=5;}
}
if(Form2->ListBox1->ItemIndex >= 0)
{Form2->ListBox1->Items->Strings[Form2->ListBox1->ItemIndex];
Form39->Label1->Caption=n;
strcpy(s,Form2->ListBox1->Items->Strings[Form2->ListBox1-
>ItemIndex+n].c_str());
Form39->Show();
} }
//-----
void __fastcall TForm38::FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
Resize = False; }
Листинг 21. Возврат в главное меню.
//-----

#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
#include "Unit40.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"

```

```

TForm40 *Form40;
//-----
__fastcall TForm40::TForm40(TComponent* Owner)
    : TForm(Owner)
{
Form40->Position=poScreenCenter;
}
//-----
void __fastcall TForm40::Button1Click(TObject *Sender)
{
Form40->Close();
Form1->Show();
}
//-----
void __fastcall TForm40::FormCanResize(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
Resize = False;
}
//-----

```

Листинг 22. База данных.

```

void __fastcall TForm1::New1Click(TObject *Sender)
{
Form2->Show();
}
void __fastcall TForm2::Button1Click(TObject *Sender)
{
const String ConnStr = "Provider=%s;Data Source=%s;Mode=%s";

```



```
Form1->ADOTable1->Active=true;
Form1->ADOTable1->Fields->Fields[0]->ReadOnly=false;
Form1->ADOTable1->Fields->Fields[1]->ReadOnly=false;
Form1->ADOTable1->Fields->Fields[2]->ReadOnly=false;
Form1->ADOTable1->Fields->Fields[3]->ReadOnly=false;
Form1->ADOTable1->Fields->Fields[4]->ReadOnly=false;
Form1->ADOTable1->Insert();
Form1->ADOTable1->Fields->Fields[0]->Value=Form1->ADOTable1-
>RecordCount+1;
Form1->ADOTable1->Fields->Fields[1]->Value=Form2->Edit1->Text;
Form1->ADOTable1->Fields->Fields[2]->Value=StrToInt(Form2->Edit2->Text);
Form1->ADOTable1->Fields->Fields[3]->Value=StrToInt(Form2->Edit3->Text);
Form1->ADOTable1->Fields->Fields[4]->Value=StrToCurr(Form2->Edit4-
>Text);
Form1->ADOTable1->UpdateBatch();
Form1->ADOTable1->Refresh();
Form1->ADOTable1->Active=false;
}
```