## Содержание

Введение……………………………………………………………………………..2

1. Техническое задание……………………………………………………………6

1.1 Постановка задачи……………………………………………………………..6

2. Технический проект……………………………………………………………..7

2.1 Выбор инструментария программирования…………………………………7

2.2 Тип операционной системы…………………………………………………..8

2.3 Требования к аппаратной части…………………………………………….12

2.4 Интерфейс конечного пользователя………………………………………..13

2.5 Внутренняя структура программного продукта…………………………..16

3. Рабочий проект………………………………………………………………….18

3.1 Описание применения программного продукта………………………….18

3.2 Руководство пользователя…………………………………………………..18

3.3 Руководство программиста………………………………………………….18

4. Экономический раздел…………………………………………………………20

4.1 Теоретическая часть………………………………………………………….20

4.2 Расчет себестоимости и цены программы………………………………....23

Заключение………………………………………………………………………...26

Литература…………………………………………………………………………28

Приложения………………………………………………………………………..29

## Введение

В нашем мире всё больше растет потребность в совершенствовании технологий и программного обеспечения к ним, а также основой этого совершенствования является потребность общаться с людьми из далеких уголков земли, рассчитывать траекторию полета самолета, исследовать космос так же не возможно без цифровых технологий и т.д. Мир цифровых технологий позволит вам не только услышать собеседника, но и увидеть. Вы можете сидя дома посмотреть, изучить различного рода информацию при помощи Интернета.

Сейчас в нашем мире очень важна техническая грамотность и знание компьютера в целом.

Данный программный продукт будет представлять собой базу данных по преподавателям Вяземского политехнического техникума, для отдела кадров. Сам программный продукт, я буду выполнять при помощи Delphi (языка программирования). Но об этом не много позже. Delphi мы изучаем по дисциплине «Технология разработки программных продуктов». Это предмет является ключевым в нашей специальности. Не зная его, мы будем, как пианист без рук, или певец без голоса. Главное понимать, что ты делаешь и для чего. Ведь в настоящее время идет очень стремительное развитие программных продуктов, которые позволят вам ощутить действия компьютерных технологий на окружающий мир и людей. Но когда ты будешь знать, что ты сам можешь сделать что - то подобное, и если это пригодится людям, это будет самой дорогой наградой для тебя.

В последние годы программирование выделилось в некоторую дисциплину, владение которой стало основным и ключевым моментом, определяющим успех многих инженерных проектов, а сама она превратилась в объект научного исследования. Из ремесла программирование перешло в разряд академических наук. Первый крупный вклад в ее становление сделали Э. Дейкстра и Ч. Хоар. Основное внимание в их работах уделяется построению и анализу программ, а более точно - структуре алгоритмов, представляемых текстом программы. Программы представляют собой конкретные, основанные на некотором реальном представлении и строении данных воплощения абстрактных алгоритмов. Для того чтобы разрабатывать собственные алгоритмы целесообразно сначала изучить уже существующие, методы анализа их параметров и эффективности. Тем более что мировой опыт программирования насчитывает их великое множество. Рассматривая различные методы решения одной и той же задачи, полезно проанализировать, сколько вычислительных ресурсов они требуют (времени работы, памяти), и выбрать наиболее эффективный. Конечно, в этом случае нужно учитывать какая модель вычислительной системы используется для их выполнения: однопроцессорная ЭВМ или многопроцессорный комплекс.

Алгоритм - это формально описанная вычислительная процедура, получающая исходные данные, называемые его аргументом, и выдающая результат вычислений на выход. Алгоритмы строятся для решения тех или иных вычислительных задач. Формулировка задачи описывает, каким требованиям должно удовлетворять решение задачи, а алгоритм, решающий эту задачу, представляет собой метод, применение которого позволяет получить объект, удовлетворяющий этим требованиям. В настоящее время слово «алгоритм» ассоциируется, в основном, с компьютерами и другими средствами вычислительной техники, хотя разработка алгоритмов началась на заре развития математики, задолго до появления вычислительных машин. В последние полвека творческий процесс создания вычислительных алгоритмов стал наиболее интенсивным, это связано с возникновением, совершенствованием и развитием информационных технологий и всей компьютерной индустрии.

Процесс подготовки программ для цифрового компьютера - это очень увлекательное занятие. И дело не только в том, что оно оправдывает себя с экономической и научной точек зрения, оно может вызвать эстетические переживания, подобные тем, которые испытывают творческие личности при написании музыки или стихов. Для написания программ используют алгоритмы.

Алгоритмы имеют ряд важных особенностей:

конечность. Алгоритм всегда должен заканчиваться после выполнения конечного числа шагов;

определенность. Каждый шаг алгоритма и его действия должны быть строго и недвусмысленно определены для каждого возможного случая;

ввод. Алгоритм должен иметь некоторое (возможно, равное нулю) число входных данных, т.е. величин, которые задаются до начала его работы или определяются динамически во время его работы, причем эти входные данные берутся из определенного набора объектов;

вывод. У алгоритма должно быть одно или несколько выходных данных, т.е. величин, имеющих вполне определенную связь с входными данными;

эффективность. Алгоритм обычно считается эффективным, если все его операторы достаточно просты для того, чтобы их можно было точно выполнить в течение конечного промежутка времени с помощью карандаша и бумаги;

Для того чтобы программа быстро и правильно работала, необходимы не просто алгоритмы, а хорошие (в широком смысле этого слова) алгоритмы. Критериями качества алгоритма являются:

время, необходимое для его выполнения;

адаптируемость алгоритма к различным компьютерам;

его простота;

изящество и т.д.

В программировании для решения конкретной задачи, существует несколько алгоритмов и необходимо знать, какой из них наилучший. Выбор подходящего алгоритма вызывает определенные трудности, связанные со следующими требованиями к алгоритмам:

алгоритм должен быть простым для понимания, перевода в программный код и отладки;

эффективно использовать вычислительные ресурсы (время работы, память);

выполняться по возможности быстро.

Мой дипломный проект будет состоять из нескольких частей.

Первая часть будет включать в себя техническое задание, в которой будет описана постановка задачи для моего программного продукта.

Вторая часть будет содержать технический проект, т.е. в ней будет описано, при помощи чего я создал программный продукт, каким языком программирования пользовался при его создании, требования к аппаратной части. В ней будет помещено описание требований к ПК, для оптимально работы с продуктом, описание операционной системы, интерфейс программного продукта.

В третьей части вы сможете найти описание применения программного продукта и руководство для пользователя.

Четвертая часть будет представлена в виде экономической части моего дипломного проекта. В нем вы сможете найти теоретическую часть дипломного проекта и расчет себестоимости программного продукта и его цену.

Так же вы увидите структурную схему алгоритма работы программы, конечно же, используемую литературу, программный язык.

## 1. Техническое задание

## 1.1 Постановка задачи

Мой дипломный проект будет представлять собой базу данных по преподавателям Вяземского политехнического техникума, для отдела кадров.

Моё будущее приложение, т.е. база данных, будет работать следующим образом. После того как пользователь осуществит двойной клик по ярлыку программы, появится, заставка. После чего нужно будет щелкнуть кнопкой мыши по заставке и перед пользователем откроется главный вид моего приложения. В нем вы сможете увидеть базу данных по преподавателям техникума, в которой будут занесены все интересующие вас данные. Также пользователь сможет узнать информацию о разработчике, т.е. обо мне, где я расскажу о себе и о людях, которые помогали мне в создании приложения.

В главном окне приложения будут содержаться стандартные элементы такие как - кнопка минимизации, распахивания и закрытия окна. Так же кроме стандартных элементов, будут и не стандартные.

## 2. Технический проект

## 2.1 Выбор инструментария программирования

Программирование можно осуществлять на языках различного уровня: Паскаль, Си, Пролог, Бейсик и т.д. На базе паскаля чаще всего осуществляется обработка навыков по объектно-ориентированному программированию. Роль бейсика в профессиональном программирование мала. Достаточно популярен объектно-ориентированный язык Visual Basic. На классических задачах поиска и сортировки и построения рекурсивных алгоритмов традиционно оттачиваются практические навыки будущего программиста в сфере алгоритмизации и программирования.

Возможности Delphi 7 подходят для создания систем любой сложности. Благодаря среде Delphi 7, которая представляет собой интегрированную оболочку разработчика, в которую входит набор специальных программ, ответственных за разные этапы создания готовности приложения, с ней легко работать. Не малое значение имеет компилятор Delphi. Что бы откомпилировать программу достаточно выполнить всего одну команду или нажать одну клавишу (F9). Всего лишь за одну секунду программа - компилятор анализирует и транслирует тысячи строк исходного кода. В Delphi 7 имеется специальный проектировщик форм, с помощью которого будущей программы подготавливаются в виде форм. Проектировщик позволяет подобрать оптимальные размеры окон, разместить и настроить всевозможные элементы управления и меню, добавить готовые изображения, указать заголовки, подсказки, подписи и так далее. На этапе проектирования форм программа как бы составляется из готовых компонентов. Компоненты располагаются на палитре компонентов разделённой на несколько самостоятельных панелей. Компоненты обладают наборами свойств, характеризующими их отличительные особенности. Некоторые свойства имеются практически у всех компонентов - таково, например, свойство Name (имя). Другие свойства, например Caption (Заголовок), имеются у большинства компонентов - ведь заголовок необходим и для окна, и для кнопки. Некоторые уникальны для конкретных компонентов, например свойство Simple Text (Простой текст), содержащий текст, для компонента Строка состояния. В системе Delphi 7 используется специализированная, постоянно совершенствуемая версия языка программирования Паскаль, которая называется Delphi (в шестой и более ранних вариантах в системы Delphi она называлась «Object Pascal», «объектный паскаль»). Эта версия включает набор расширений, ориентированных только на применение в рамках среды Delphi 7 и предназначенных для ускоренного создания приложений.

Сравнив язык программирования с системой программирования, отметив их положительные качества, а, также учитывая, что основные знания по системе программирования Delphi 7 были получены на занятиях по дисциплине ТРПП (Технология разработки программных продуктов). Свою базу данных, я буду разрабатывать в среде визуального программирования Delphi 7.

## 2.2 Тип операционной системы

Данная программа рассчитана для работы с операционной системой Windows XP Professional.

Объектно-ориентированный подход реализуется через модель рабочего стола. Пользователь работает с задачами и приложениями так же, как с документами на своём письменном столе.

Это удобно для людей, которые в первый раз увидели компьютер, но создаёт некоторые трудности «переходного периода» для тех, кто привык считать программу основой своего сущего в машине.

Подход к аппаратному обеспечению также кардинальным образом изменился. Теперь система использует стандарт Plug&Play, что облегчает и максимально автоматизирует процесс добавления новых периферийных устройств. Стандарт Plug&Play - Это совместная разработка фирм Intel и Microsoft. Основная его идея заключается в том, что каждое устройство, соответствующее этому стандарту, сообщает о себе определённую информацию, благодаря которой операционная система выполняет автоматическую конфигурацию периферийных устройств и разрешает аппаратные конфликты. Стандарту Plug&Play должен в первую очередь удовлетворять BIOS материнской платы и, разумеется, периферийные устройства.

Давайте рассмотрим характеристики всех операционных систем и вы увидите, почему я выбрал Windows XP

Windows 98 SE это программный продукт фирмы Microsoft который предназначен для максимального использование компьютерных ресурсов. Данный продукт поддерживает большое количество компьютерного «железа».

Многозадачность этой операционной системы предоставила большие возможности для программирования, а также много поточность и легкость в использовании программ. В данной операционной системе использование и применение программ становиться просто и доступно каждому человеку.

Минимальные требования к Windows 98 SE:

Pentium 133;

32 Мб ОЗУ;

540 Мб жесткого диска;

VGA-совместимая видео карта;

мышь;

клавиатура.

Оптимальная работа при данных требованиях этой системы будет обеспечивать надежность работы систем и прикладных программ к ней.

Windows Professional 2000:

Данная операционная система является одним из продуктов линии Windows 2000 и является наследником операционной системы Windows NT. Операционная система (О. С) Windows 2000 Professional предназначена для настольных и мобильных компьютеров в организации любого масштаба.

В процессе проектирования Windows 2000 Professional преследовались следующие цели:

упростить работу с системой (для этого используется привычный, но модифицированный интерфейс Windows 2000, в который интегрированы новые возможности);

сохранить традиционные достоинства системы NT (надежность, производительность, безопасность);

перенести в систему лучшие качества устройств и Plug& Play;

создать легко конфигурируемую настольную систему, позволившую снизить общую стоимость. Для этого обеспечивается возможность централизованного управления системой и приложениями, а также удаленная инсталляция системы.

Преимущества О.С. Windows 2000 Professional:

новый пользовательский интерфейс: Active Desktop;

браузер Internet Explorer, обеспечивающий наиболее полную;

интеграцию с системой;

direct 7.0 - новейшая версия графических интерфейсов API;

технология Plug and Play;

поддержка Universal Serial Bus;

AC PI Power - усовершенствованный стандарт управления конфигурацией и питанием;

расширенное восстановление системы;

улучшенная версия NT File System;

поддержка нового стандарта IEEE 1394;

сервер сценариев Windows Soripting Host;

планировщик задач Task Scheducler;

консоль управления MMS;

поддержка FAT 32;

драйвер NDIS 5.0;

поддержка ММХ, AGP;

дополнительные мониторы (до 4-х);

Active Directory;

IP Security;

поддержка Intelligent I/O Architecture (120), позволяющей разгрузить ЦП и передать часть задач по обслуживанию устройств ввода-вывода другим процессором;

поддержка смарт-карт;

Win Driver (WDM) - новая архитектура драйверов для 32-х разрядных систем;

поддержка сканеров, фотоаппаратов и фотокамер;

встроенная система дефрагментация дисков FAT 16, FAT 32, NTFS;

служба индексации содержимого и атрибутов документов;

поддержка Asynchronous Transfer Mode (ATM) средствами ОС, позволяющая обеспечить приложениям все возможности ATM;

новая версия существующих программных средств для обработки факсов.

Работать со своим приложением я буду в операционной системе Windows XP. Я выбрал именно эту операционную систему, потому что она является одной из последних разработок программного обеспечения персональных компьютеров компании Microsoft. Windows XP на данный момент является самой стабильной из всех ранее выпускаемых версий Windows. Версия Windows XP более простая версия по сравнению с версией Windows 2000, и менее подверженная аварийным отказам, нежели Windows Me. Windows XP представляет собой самую значительную модернизацию операционной системы, осуществленную компанией Microsoft за последние годы. И в тоже время - самую противоречивую из всех таких модернизаций. Эта операционная система самая функционально объёмная и многогранная версия. Она великолепно подходит для новейших ПК с их быстрыми процессорами, емкими жёсткими дисками и скоростными устройствами для записи компакт-дисков.

Операционная система Windows XP с помощью модуля Windows File Protection имеет возможность восстановления некоторых из используемых и системных файлов (\*. exe, \*. dll, \*. ocx и некоторые другие), которые оказались без всякой на то необходимости заменёнными (например, в процессе инсталляции какой либо программы), предотвращая тем самым возможные нарушения в работе операционной системы. Это связано с тем, что для защиты наиболее важных файлов Microsoft ввела их цифровую подпись, и в том случае, если обнаруживается, что происходит замена системного файла другим, не имеющим такой цифровой подписи, следует немедленное оповещение пользователя.

Операционная система Windows XP улучшила графический интерфейс, который значительно отличается оттого, что был в предыдущих операционных системах.

## 2.3 Требования к аппаратной части

Delphi 7 может работать в среде операционных систем от Windows 98 до Windows XP. Особых требований, по современным меркам, к ресурсам компьютера пакет не предъявляет: процессор должен быть типа Pentium или Celeron с тактовой частотой не ниже 166 МГц (рекомендуется Pentium II 400 МГц), оперативной памяти - 128 Мбайт (рекомендуется 256 Мбайт), достаточное количество свободного дискового пространства (для полной установки версии Enterprise необходимо приблизительно 475 Мбайт).

Спроектируем компьютерную модель базы данным по преподавателям техникума. Что такое «хорошее приложение» зависит от мощности аппаратуры, уровня развития программного обеспечения, вкусов пользователей и, конечно же, постоянно меняется. Мы все же постараемся нарисовать образ хорошего приложения, который будет соответствовать действительности, по крайней мере лет пять:

это 32-разрядное приложение для Windows 98 SE или Windows NT, a так же Windows XP и т.д.;

оно имеет простой, удобный, интуитивно понятный интерфейс со всеми присущими Windows атрибутами: динамическими окнами, кнопками, меню;

оно управляется как мышью, так и клавиатурой;

оно отказоустойчиво и корректно обрабатывает любые ошибки пользователя;

оно черпает информацию из баз данных;

оно работает быстро, не раздражая пользователя бесконечно медленной прокруткой;

оно хорошо документировано (как минимум имеет хороший Help);

оно поддерживает OLE (требования для всех продуктов, желающих получить сертификат от Microsoft);

оно, возможно, относятся к группе мультимедиа-приложение;

оно разрабатывается достаточно быстро, чтобы не устареть еще на

стадии разработки.

## 2.4 Интерфейс конечного пользователя

После запуска программного продукта внешний вид будет выглядеть так, как это показано на рисунке 2.1

Рис.2.1

После нажатия кнопкой мыши по этому интерфейсу, появится главное окно программного продукта. В главном окне располагается несколько листов, на которых есть все, что необходимо для составления отчетов и работы в отделе кадров. Главное окно выглядит так.

Рис.2.2

После заполнения всех необходимых данных, мы переходит к их отчетам, которые мы без особого труда можем вывести на печать.

Рис.2.3

Можно также вывести на бумажный носитель и заявление на аттестацию для выбранного вами преподавателя, который будет автоматически подключен к данному отчету

Рис.2.4

Еще на одном отчете, мы можем увидеть аттестационный лист, который пригодится при повышении квалификации преподавателя.

Рис.2.5

## 2.5 Внутренняя структура программного продукта

Рассмотрим внутренняя структура программного продукта. Ядро программы - это самая главная часть программы, которая отвечает за механизм работы и выполнение различных операций, таких как вызов модулей, ввод информации, вывод информации, интерфейс пользователя и работа с системными продуктами. Интерфейс - это внешний вид программы, где пользователь может обращаться к программе и данным. Ввод информации - это распознавание программой введенных вами данных. Вывод информации - вывод результата ваших действий на экран. ОС Windows XP - это использование программой системных библиотек операционной системы Windows XP. Модули uses - это использование системных модулей программного продукта, которые подключаются к главной программе и к другим модулям во время разработки программного продукта.

Структура программы

Рис.2.6

D

B Edit

Data Source

Tables

TabSheet

Labels

 Project1.

 exe

Main Menu

RV Project 1

RV Table Connection 1

## 3. Рабочий проект

## 3.1 Описание применения программного продукта

Данный программный продукт предназначен для отдела кадров. Он будет применяться, как база данных о сведениях преподавателя. Она очень легка в практическом применении, имеет доступный и приятный интерфейс, и разобраться в принципе её работы, для обычного пользователя не составит особого труда. Все что нужно - это заносить и сохранять данные.

## 3.2 Руководство пользователя

Для работы в данной программе вам потребуется небольшие навыки работы с Windows-приложениями. Внешний вид программы прост и понятен.

Программа запускается двойным щелчком по пусковому файлу программы - Project1. exe.

Главное меню представлено несколькими пунктами: файл, справка. При выборе пункта меню файл выпадает список: отчет, заявление, аттестационный лист, закрыть. Выбрав пункт меню Файл/Отчет вы сможете распечатать личные данные преподавателя, заполнить заявление на повышение квалификации и аттестационный лист. Также выбрав вкладку Закрыть, вы сможете выйти из программы. При выборе пункта меню Справка, вы сможете узнать информацию о разработчике.

## 3.3 Руководство программиста

Данная программа осуществляет работу с базой данных. При использовании компонентов DBEdit, TabSheet, Label, Table, MainMenu, Data Source, RV Project1, RV Table Connektion1. У всех данных компонентов осуществлена взаимосвязь, так как при общем их сочетании мы получаем желательный результат. Без этих компонентом работа данного продукта не осуществляется.

Программа состоит из модулей. Каждый модуль соответствует определённой форме и содержит процедуры. Модуль начинает работать сразу же по окончанию работы предыдущего модуля, либо по вызову его пользователем щелчком мыши по соответствующему объекту.

В программе использованы различные свойства компонентов: видимость (visible), доступность (enabled), заголовок (caption) и куча других настроек, необходимых для правильной работы программного продукта.

## 4. Экономический раздел

## 4.1 Теоретическая часть

Переход к рыночным отношениям в экономике и научно-технический прогресс чрезвычайно ускорили темпы внедрения во все сферы социально-экономической жизни российского общества последних достижений в области информатизации. Достижение Россией высоких результатов в экономике и социальной области и завоевание места полноправного партнера в мировой экономической системе в значительной степени зависит от того, каковы будут масштабы использования современных информационных технологий во всех аспектах человеческой деятельности, а также от того, какую роль будут играть эти технологии в повышении эффективности общественного труда.

Современное человеческое общество живет в период, характеризующийся небывалым увеличением информационных потоков. Это относится как к экономике, так и к социальной сфере. Наибольший рост объема информации наблюдается в промышленности, торговле, финансово-банковской сфере. В промышленности рост объема информации обусловлен увеличением объема производства, усложнением выпускаемой продукции, используемых материалов, технологического оборудования, расширением внешних и внутренних связей экономических объектов. Рыночные отношения предъявляют повышенные требования к своевременности, достоверности, полноте информации, без которой немыслима эффективная маркетинговая, финансово-кредитная, инвестиционная деятельность. Роль информации в общественной жизни существенно меняется. Информация приобретает преобразующий, определяющий характер. Создание индустрии информатики и превращение информационного продукта в товар приводит к глубинным социальным изменениям в обществе, трансформируя его из индустриального в информационное. Информация охватывает все стороны жизни общества - от материального производства до социальной сферы. Качественно новое обслуживание информационных процессов человеческой деятельности связано с использованием современной персональной электронно-вычислительной техники, систем телекоммуникаций, созданием сетей ЭВМ. В связи с этим общество на уровне государства должно решить ряд проблем информатизации. Пути решения наиважнейших из них были определены комплексом организационных, экономических и правовых мер в Указе Президента РФ от 20 января 1994 г. «Об основах государственной политики в сфере информатизации».

Стремительное развитие товарных и финансовых рынков в России явилось мощным толчком к интенсивному нарастанию процессов информатизации во всех сферах жизни общества. Изменились подходы к оценке роли информации и информационному обслуживанию производственно-хозяйственной, управленческой деятельности и различных категорий пользователей. Наше общество стало открытым, а потому возросла потребность в достоверных, современных, полных, представленных в доступной для пользователя форме сведений. Информация пронизывает все стороны деятельности субъектов экономики и производства, предоставляет высшему менеджменту безотказный механизм управления при условии правильной постановки задач и обоснованного выбора их технологического решения. В результате законотворческой и регулирующей деятельности государства возросли требования к достоверности, своевременности и законности отчетно-статистической информации, вырабатываемой на уровне микро - и макроэкономических объектов. Информационное сопровождение производства и бизнеса стало необходимым условием полноты и своевременности сбора налогов, что на сегодняшний день все в большей степени определяет существование и процветание государства - главного потребителя самой разнообразной, в том числе финансово-экономической информации. Для современных микроэкономических структур обладание информацией не менее важно, так как неопределенность в окружающей среде возросла в сто крат, что приводит нередко к сбоям в деятельности фирм, предприятий, корпораций, банков. Необоснованное принятие решений на любом уровне руководства может обернуться для хозяйствующих объектов непоправимыми отрицательными последствиями.

В России в условиях становления рыночных отношений скорость обновления информации на государственном, территориальном, отраслевом уровне и уровне предприятия очень высока. Спрос на достоверную, актуальную и полную информацию возрастает. Это обусловливает появление фирм, предоставляющих информационные услуги. Фирмы предлагают правовые общезначимые документы, постоянно необходимые юристам, аудиторам, бухгалтерам, работникам банковских и финансовых структур, государственным чиновникам и руководителям предприятий. Общий объем комплексов составляет десятки тысяч документов законодательного и нормативного характера. Обновление и пополнение информации осуществляется фирмами еженедельно на основании прямых договоров об информационном обмене с органами власти и управления Российской Федерации. Региональные банки правовой информации дают своим клиентам возможность использовать в работе труднодоступные документы органов власти и управления регионального уровня.

Цель моего дипломного проекта «Отдел кадров ВПТ» заключается в возможности в автоматизированном режиме вести базу данных работников Вяземского политехнического техникума, заполнять аттестационные документы, печатать отчеты.

## 4.2 Расчет себестоимости и цены программы

Таблица 4.2.1

Объем учебных часов по выбранным дисциплинам и тарифной сетки работников бюджетных организаций

|  |  |  |  |
| --- | --- | --- | --- |
| Перечень дисциплин по учебному плану, выбирает студент | Часы | Тарифный разряд | Фонд оплаты труда, руб.  |
| Экономика отрасли | 77 | 13 | 3080 |
| Математические методы | 101 | 14 | 4340,98 |
| Технология разработки программных продуктов | 122 | 14 | 5243,56 |
| Основы алгоритмизации и программирования | 192 | 7 | 4535,04 |
| Итого:  | 492 |  | 17199,58 |

Таблица 4.2.2

Тарифные разряды труда преподавателей и часовые тарифные ставки по оплате

|  |  |  |
| --- | --- | --- |
| Преподаватель | Тарифный разряд преподавателя | Стоимость нормо - часа(руб. - коп) Сч |
| Савченко Р.П.  | 13 | 40 |
| Контанистова Е.П.  | 14 | 42,98 |
| Коростелев М.К.  | 14 | 42,98 |
| Молотов Д.Н.  | 7 | 23,62 |

Средняя стоимость одного нормо-часа:

 руб/ч.

Время работы над программой - 60 ч.

Основная зарплата по стоимости труда программиста равна:

 руб.

Дополнительная зарплата (отпускные, учебный отпуск и т.д.) - 20% от основной зарплаты:

 руб.

Отчисление на социальное страхование - 26% от основной зарплаты:

 руб.

Оплата труда руководителя дипломного проекта (Никонов С. В) с начислениями за 12 часов.

 руб.

Стоимость работы компьютера:

 руб.

где 45 руб. /ч. - стоимость работы одного часа компьютера; 45 ч. - время работы.

Стоимость диска - 15 руб.

Стоимость инструкции - руководство по эксплуатации: 3 руб.

Итого общие затраты - 5758,31 руб.

Накладные расходы предприятия (амортизация помещения, освещения, отопления и т.д.) составляет 10% от всех затрат:

 руб.

Полная себестоимость программы:

5758,31+575,83=6334,14 руб.

Для определения стоимости программы начислим плановую прибыль в размере 15% (от 10 до 25%) по выбору студента:

 руб. руб.


## Заключение

Приложение баз данных, как следует уже из его названия, предназначено для взаимодействия с некоторым источником данных - базой данных (БД). Взаимодействие подразумевает получение данных, их представление в определенном формате для просмотра пользователем, редактирование в соответствии с реализованными в программе бизнес - алгоритмами и возврат обработанных данных обратно в базу данных.

В качестве источника данных могут выступать как собственно базы данных, так и обычные файлы - текстовые, электронные таблицы и т.д. Но здесь мы будем рассматривать приложения, работающие с базами данных.

Как известно, базы данных обслуживаются специальными программами - системами управления базами данных (СУБД), которые делятся на локальные, преимущественно однопользовательские, предназначенные для настольных приложений, и серверные - сетевые (часто удаленные), многопользовательские, функционирующие на выделенных компьютерах - серверах. Главный критерий такой классификации - объем базы данных и средняя нагрузка на СУБД.

Тем не менее, несмотря на разнообразие реализаций, общая архитектура приложения баз данных остается неизменной.

Само приложение включает механизм получения и отправки данных, механизм внутреннего представления данных в том или ином виде, пользовательский интерфейс для отображения и редактирования данных, бизнес-логику для обработки данных.

Механизм получения и отправки данных обеспечивает соединение с источником данных (часто опосредованно). Он должен "знать", куда ему обращаться и какой протокол обмена использовать для обеспечения двунаправленного потока данных.

Механизм внутреннего представления данных является ядром приложения баз данных. Он обеспечивает хранение полученных данных в приложении и предоставляет их по запросу других частей приложения.

Пользовательский интерфейс обеспечивает просмотр и редактирование данных, а также управление данными и приложением в целом.

Бизнес-логика приложения представляет собой набор реализованных в программе алгоритмов обработки данных.

Между приложением и собственно базой данных находится специальное программное обеспечение (ПО), связывающее программу и источник данных и управляющее процессом обмена данными. Это ПО может быть реализовано самыми разнообразными способами, в зависимости от объема базы данных, решаемых системой задач, числа пользователей, способами соединения приложения и базы данных. Промежуточное ПО может быть реализовано как окружение приложения, без которого оно вообще не будет работать, как набор драйверов и динамических библиотек, к которым обращается приложение, может быть интегрировано в само приложение. Наконец, это может быть отдельный удаленный сервер, обслуживающий тысячи приложений.

## Литература

1. Бобровский С.И. Delphi 7. - СПб.: Питер. 2004.

2. Гаевский А.Ю. Самоучитель работы на ПК. - М.: ТЕХНОЛОДЖИ. 2004.

3. Губарев В.Г. Программное обеспечение и операционные системы ПК. - Ростов н/Д.: Феникс. 2004.

4. Марченко А.И., Марченко Л.А. Программирование в среде Turbo Pascal 7.0. - СПб.: Корона принт. 2005.

5. Могилёв А.В., Пак Н.И., Хеннер Е.К. Практикум по информатике. М.: Академия. 2006.

6. Рудаков А.В. Технология разработки программных продуктов. - М.: Академия. 2006.

7. Кузнецова Т.В., Кузнец С.Л. Кадровое делопроизводство. - М.: ООО «Интел - синтез АПР». 2006.

## Приложения

## Приложение 1

да

Лист4

начало

Лист1

Занесение личных данных

нет

Лист2

Паспортные данные и прием на работу

да

нет

Лист3

Аттестация, повышение квалификации и проф. подготовка

да

нет

Отпуск и дополнитель-ные сведения

да

нет

да

нет

1

2

3

Отчеты

да

конец

1

2

нет

Печать

да

нет

3

Закрыть

## Приложение 2

program Project1;

uses

Forms,

Unit1 in 'Unit1. pas' {Form1},

Unit2 in 'Unit2. pas' {Form2},

Unit3 in 'Unit3. pas' {Form3};

{$R \*. res}

begin

Application. Initialize;

Application. CreateForm(TForm1, Form1);

Application. Run;

end.

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, XPMan, Buttons, jpeg, ExtCtrls;

type

TForm1 = class(TForm)

Image1: TImage;

Label1: TLabel;

procedure Image1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

uses Unit2;

{$R \*. dfm}

procedure TForm1. Image1Click(Sender: TObject);

begin

Form2: =TForm2. Create(Self);

try

Form2. ShowModal;

finally

Form2. Free;

end;

end;

end.

unit Unit2;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, XPMan, ComCtrls, Menus, Buttons, DB, DBTables, ExtCtrls, DBCtrls, Mask, RpDefine, RpCon, RpConDS, RpConBDE, RpRave;

type

TForm2 = class(TForm)

PageControl1: TPageControl;

TabSheet1: TTabSheet;

TabSheet2: TTabSheet;

XPManifest1: TXPManifest;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N6: TMenuItem;

TabSheet3: TTabSheet;

TabSheet4: TTabSheet;

DBNavigator1: TDBNavigator;

DataSource1: TDataSource;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Table1: TTable;

Table1Fio: TStringField;

Table1Im: TStringField;

Table1Ot: TStringField;

Table1Dd: TStringField;

Table1Dm: TStringField;

Table1Dg: TStringField;

Table1TRdn: TStringField;

Table1TRdd: TStringField;

Table1Obr: TStringField;

Table1NamObYch: TStringField;

Table1Nam: TStringField;

Table1Ser: TFloatField;

Table1Nom: TFloatField;

Table1GodOkonch: TStringField;

Table1ProfOsn: TStringField;

Table1ProfDr: TStringField;

Table1StajRabnad: TStringField;

Table1StajRabnam: TStringField;

Table1StajRabnag: TStringField;

Table1StajObd: TStringField;

Table1StajObm: TStringField;

Table1StajObg: TStringField;

Table1StajNepd: TStringField;

Table1StajNepm: TStringField;

Table1StajNepg: TStringField;

Table1Pas: TStringField;

Table1PasNom: TStringField;

Table1Pasd: TStringField;

Table1Pasm: TStringField;

Table1Pasg: TStringField;

Table1PasVid: TStringField;

Table1DatRegd: TStringField;

Table1DatRegm: TStringField;

Table1DatRegg: TStringField;

Table1NomerTel: TStringField;

Table1PrimnaRabd: TStringField;

Table1Strpod: TStringField;

Table1Dolg: TStringField;

Table1Stvka: TStringField;

Table1Osn1: TStringField;

Table1Podpes: TStringField;

Table1DatAtest: TStringField;

Table1ReshKom: TStringField;

Table1Docnom: TStringField;

Table1Docdat: TStringField;

Table1Osn2: TStringField;

Table1Datnach: TStringField;

Table1Datokoch: TStringField;

Table1Vidpovish: TStringField;

Table1NaimenObrYch: TStringField;

Table1Docnaimen: TStringField;

Table1DocSer: TStringField;

Table1Docd: TStringField;

Table1Osn3: TStringField;

Table1DataNach: TStringField;

Table1DataKon: TStringField;

Table1Special: TStringField;

Table1DokymNam: TStringField;

Table1DokymNOM: TStringField;

Table1Dokymdat: TStringField;

Table1Osn4: TStringField;

Table1KolD: TStringField;

Table1PerRabS: TStringField;

Table1PerRabPo: TStringField;

Table1KolDen: TStringField;

Table1DataNa: TStringField;

Table1DataOk: TStringField;

Table1Osn5: TStringField;

Table1Yvol: TStringField;

Table1DatYd: TStringField;

Table1DatYm: TStringField;

Table1DatYg: TStringField;

Table1PrikN: TStringField;

Table1Prikd: TStringField;

Table1Prikm: TStringField;

Table1Prikg: TStringField;

Table1RabKadSlD: TStringField;

Table1Pod: TStringField;

Table1Rashifr: TStringField;

Table1Rabotnic: TStringField;

DBEdit1: TDBEdit;

DBEdit2: TDBEdit;

Label4: TLabel;

DBEdit3: TDBEdit;

Label5: TLabel;

DBEdit4: TDBEdit;

Label6: TLabel;

DBEdit5: TDBEdit;

Label7: TLabel;

DBEdit6: TDBEdit;

Label8: TLabel;

Label9: TLabel;

DBEdit7: TDBEdit;

Label10: TLabel;

DBEdit8: TDBEdit;

Label12: TLabel;

DBEdit10: TDBEdit;

Label13: TLabel;

DBEdit11: TDBEdit;

Label14: TLabel;

DBEdit12: TDBEdit;

Label15: TLabel;

DBEdit13: TDBEdit;

Label16: TLabel;

DBEdit14: TDBEdit;

Label17: TLabel;

DBEdit15: TDBEdit;

Label18: TLabel;

DBEdit16: TDBEdit;

Label19: TLabel;

DBEdit17: TDBEdit;

Label20: TLabel;

Table1Grajdan: TStringField;

Table1MestoJ: TStringField;

Label11: TLabel;

DBEdit9: TDBEdit;

Label21: TLabel;

DBEdit18: TDBEdit;

Label22: TLabel;

DBEdit19: TDBEdit;

Label23: TLabel;

DBEdit20: TDBEdit;

Label24: TLabel;

DBEdit21: TDBEdit;

Label25: TLabel;

Label26: TLabel;

Label27: TLabel;

DBEdit22: TDBEdit;

Label28: TLabel;

DBEdit23: TDBEdit;

Label29: TLabel;

DBEdit24: TDBEdit;

Label30: TLabel;

Label31: TLabel;

Label32: TLabel;

Label33: TLabel;

DBEdit25: TDBEdit;

Label34: TLabel;

DBEdit26: TDBEdit;

Label35: TLabel;

DBEdit27: TDBEdit;

Panel1: TPanel;

Label36: TLabel;

Label37: TLabel;

DBEdit28: TDBEdit;

Label38: TLabel;

DBEdit29: TDBEdit;

Label39: TLabel;

DBEdit30: TDBEdit;

Label40: TLabel;

DBEdit31: TDBEdit;

Label41: TLabel;

DBEdit32: TDBEdit;

Label42: TLabel;

Label43: TLabel;

DBEdit33: TDBEdit;

Label44: TLabel;

Label45: TLabel;

DBEdit34: TDBEdit;

Label46: TLabel;

DBEdit35: TDBEdit;

Label47: TLabel;

DBEdit36: TDBEdit;

Label48: TLabel;

DBEdit37: TDBEdit;

Label49: TLabel;

Panel2: TPanel;

Label50: TLabel;

DBEdit38: TDBEdit;

Label51: TLabel;

DBEdit39: TDBEdit;

Label52: TLabel;

DBEdit40: TDBEdit;

Label53: TLabel;

DBEdit41: TDBEdit;

Label54: TLabel;

DBEdit42: TDBEdit;

Label55: TLabel;

DBEdit43: TDBEdit;

Label56: TLabel;

Panel3: TPanel;

Panel4: TPanel;

Panel5: TPanel;

Label57: TLabel;

Label58: TLabel;

Label59: TLabel;

Label60: TLabel;

Label61: TLabel;

Label62: TLabel;

Label63: TLabel;

DBEdit44: TDBEdit;

Label64: TLabel;

DBEdit45: TDBEdit;

Label65: TLabel;

DBEdit46: TDBEdit;

Label66: TLabel;

DBEdit47: TDBEdit;

Label67: TLabel;

DBEdit48: TDBEdit;

Label68: TLabel;

Label69: TLabel;

DBEdit49: TDBEdit;

Label70: TLabel;

DBEdit50: TDBEdit;

Label71: TLabel;

DBEdit51: TDBEdit;

Label72: TLabel;

DBEdit52: TDBEdit;

Label73: TLabel;

Label74: TLabel;

DBEdit53: TDBEdit;

Label75: TLabel;

DBEdit54: TDBEdit;

Label76: TLabel;

DBEdit55: TDBEdit;

Label77: TLabel;

DBEdit56: TDBEdit;

Label78: TLabel;

Label79: TLabel;

DBEdit57: TDBEdit;

Label80: TLabel;

DBEdit58: TDBEdit;

Label81: TLabel;

DBEdit59: TDBEdit;

Label82: TLabel;

DBEdit60: TDBEdit;

Label83: TLabel;

DBEdit61: TDBEdit;

Label84: TLabel;

DBEdit62: TDBEdit;

Label85: TLabel;

DBEdit63: TDBEdit;

Label86: TLabel;

Label87: TLabel;

Panel8: TPanel;

Label88: TLabel;

Label90: TLabel;

Panel6: TPanel;

Label89: TLabel;

DBEdit64: TDBEdit;

Label91: TLabel;

DBEdit65: TDBEdit;

Label92: TLabel;

DBEdit66: TDBEdit;

Label93: TLabel;

DBEdit67: TDBEdit;

Label94: TLabel;

DBEdit68: TDBEdit;

Label95: TLabel;

DBEdit69: TDBEdit;

Label96: TLabel;

DBEdit70: TDBEdit;

Label97: TLabel;

Label98: TLabel;

Label99: TLabel;

Label100: TLabel;

Label101: TLabel;

DBEdit71: TDBEdit;

Label102: TLabel;

DBEdit72: TDBEdit;

Label103: TLabel;

DBEdit73: TDBEdit;

Label104: TLabel;

Label105: TLabel;

DBEdit74: TDBEdit;

Label106: TLabel;

DBEdit75: TDBEdit;

Label107: TLabel;

DBEdit76: TDBEdit;

Label108: TLabel;

DBEdit77: TDBEdit;

Label109: TLabel;

DBEdit78: TDBEdit;

Label110: TLabel;

Label111: TLabel;

Label112: TLabel;

DBEdit79: TDBEdit;

Label114: TLabel;

DBEdit81: TDBEdit;

Label115: TLabel;

RvTableConnection1: TRvTableConnection;

RvProject1: TRvProject;

RvTableConnection2: TRvTableConnection;

RvProject2: TRvProject;

RvTableConnection3: TRvTableConnection;

RvProject3: TRvProject;

N7: TMenuItem;

N8: TMenuItem;

procedure N2Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure Label32MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

procedure Label32MouseLeave(Sender: TObject);

procedure Label32Click(Sender: TObject);

procedure Label36Click(Sender: TObject);

procedure Label49MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

procedure Label49Click(Sender: TObject);

procedure Label56Click(Sender: TObject);

procedure Label49MouseLeave(Sender: TObject);

procedure Label57Click(Sender: TObject);

procedure Label58Click(Sender: TObject);

procedure Label59Click(Sender: TObject);

procedure Label60Click(Sender: TObject);

procedure Label61Click(Sender: TObject);

procedure Label62Click(Sender: TObject);

procedure Label88Click(Sender: TObject);

procedure Label99Click(Sender: TObject);

procedure Label100Click(Sender: TObject);

procedure Label90Click(Sender: TObject);

procedure Label57MouseLeave(Sender: TObject);

procedure Label57MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

procedure Label58MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

procedure Label59MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

procedure Label88MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

procedure Label100MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

procedure Label58MouseLeave(Sender: TObject);

procedure Label59MouseLeave(Sender: TObject);

procedure Label88MouseLeave(Sender: TObject);

procedure Label100MouseLeave(Sender: TObject);

procedure N6Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N7Click(Sender: TObject);

procedure N8Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses Unit1, Unit3;

{$R \*. dfm}

procedure TForm2. N2Click(Sender: TObject);

begin

form1. Close;

end;

procedure TForm2. FormCreate(Sender: TObject);

var s: string;

begin

Table1. Active: =true;

Label36. Cursor: =crHandPoint;

Label32. Cursor: =crHandPoint;

Label49. Cursor: =crHandPoint;

Label56. Cursor: =crHandPoint;

Label57. Cursor: =crHandPoint;

Label58. Cursor: =crHandPoint;

Label59. Cursor: =crHandPoint;

Label60. Cursor: =crHandPoint;

Label61. Cursor: =crHandPoint;

Label62. Cursor: =crHandPoint;

Label88. Cursor: =crHandPoint;

Label90. Cursor: =crHandPoint;

Label99. Cursor: =crHandPoint;

Label100. Cursor: =crHandPoint;

end;

procedure TForm2. Label32MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

begin

Label32. Font. Color: =clBlue;

end;

procedure TForm2. Label32MouseLeave(Sender: TObject);

begin

Label32. Font. Color: =clBlack;

end;

procedure TForm2. Label32Click(Sender: TObject);

begin

Panel1. Visible: =true;

Label32. Visible: =false;

end;

procedure TForm2. Label36Click(Sender: TObject);

begin

Panel1. Visible: =false;

Label32. Visible: =true;

end;

procedure TForm2. Label49MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

begin

Label49. Font. Color: =clBlue;

end;

procedure TForm2. Label49Click(Sender: TObject);

begin

Panel2. Visible: =true;

Label49. Visible: =false;

end;

procedure TForm2. Label56Click(Sender: TObject);

begin

Panel2. Visible: =false;

Label49. Visible: =true;

end;

procedure TForm2. Label49MouseLeave(Sender: TObject);

begin

Label49. Font. Color: =clBlack;

end;

procedure TForm2. Label57Click(Sender: TObject);

begin

Panel3. Visible: =true;

Label57. Visible: =false;

end;

procedure TForm2. Label58Click(Sender: TObject);

begin

Panel4. Visible: =true;

Label58. Visible: =false;

end;

procedure TForm2. Label59Click(Sender: TObject);

begin

Panel5. Visible: =true;

Label59. Visible: =false;

end;

procedure TForm2. Label60Click(Sender: TObject);

begin

Panel3. Visible: =false;

Label57. Visible: =true;

end;

procedure TForm2. Label61Click(Sender: TObject);

begin

Panel4. Visible: =false;

Label58. Visible: =true;

end;

procedure TForm2. Label62Click(Sender: TObject);

begin

Panel5. Visible: =false;

Label59. Visible: =true;

end;

procedure TForm2. Label88Click(Sender: TObject);

begin

label88. Visible: =false;

Panel6. Visible: =true;

end;

procedure TForm2. Label99Click(Sender: TObject);

begin

label88. Visible: =true;

Panel6. Visible: =false;

end;

procedure TForm2. Label100Click(Sender: TObject);

begin

Label100. Visible: =false;

Panel8. Visible: =true;

end;

procedure TForm2. Label90Click(Sender: TObject);

begin

Label100. Visible: =true;

Panel8. Visible: =false;

end;

procedure TForm2. Label57MouseLeave(Sender: TObject);

begin

Label57. Font. Color: =clBlack;

end;

procedure TForm2. Label57MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

begin

Label57. Font. Color: =clBlue;

end;

procedure TForm2. Label58MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

begin

Label58. Font. Color: =clBlue;

end;

procedure TForm2. Label59MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

begin

Label59. Font. Color: =clBlue;

end;

procedure TForm2. Label88MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

begin

Label88. Font. Color: =clBlue;

end;

procedure TForm2. Label100MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

begin

Label100. Font. Color: =clBlue;

end;

procedure TForm2. Label58MouseLeave(Sender: TObject);

begin

Label58. Font. Color: =clBlack;

end;

procedure TForm2. Label59MouseLeave(Sender: TObject);

begin

Label59. Font. Color: =clBlack;

end;

procedure TForm2. Label88MouseLeave(Sender: TObject);

begin

Label88. Font. Color: =clBlack;

end;

procedure TForm2. Label100MouseLeave(Sender: TObject);

begin

Label100. Font. Color: =clBlack;

end;

procedure TForm2. N6Click(Sender: TObject);

begin

RvProject1. Execute;

end;

procedure TForm2. N4Click(Sender: TObject);

begin

Form3: =TForm3. Create(Self);

try

Form3. ShowModal;

finally

Form3. Free;

end;

end;

procedure TForm2. N7Click(Sender: TObject);

begin

RvProject2. Execute;

end;

procedure TForm2. N8Click(Sender: TObject);

begin

RvProject3. Execute;

end;

end.

unit Unit3;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, StdCtrls, ShellApi, jpeg, OleCtrls, SHDocVw, XPMan, Buttons;

type

TForm3 = class(TForm)

Panel1: TPanel;

Image1: TImage;

Label3: TLabel;

Label6: TLabel;

XPManifest1: TXPManifest;

Label7: TLabel;

Label1: TLabel;

Label2: TLabel;

Label4: TLabel;

procedure FormCreate(Sender: TObject);

procedure Label6Click(Sender: TObject);

procedure Label6MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

procedure Label6MouseLeave(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

{$R \*. dfm}

procedure TForm3. FormCreate(Sender: TObject);

begin

Image1. Picture. LoadFromFile('Мы 0085959595. jpg');

end;

Procedure AutoSendMail;

Var

EMailDestinationString,SubjectString,Line1String,Line2String,mailstring: String;

begin

EMailDestinationString: ='aleks\_bugaev@mail. ru';

SubjectString: ='Message Subject';

Line1String: ='This is the first line';

Line2String: ='This is the second line';

 // Можно использовать несколько адресов, разделяя их точкой с запятой //

mailstring: ='mailto: ' + EMailDestinationString +

'? subject=' + SubjectString +

'&body='+ Line1String +

'%0d' + Line2String;

if (ShellExecute(0,'open',PChar(mailstring),'','',SW\_SHOWNORMAL) <=32) then

ShowMessage('Auto method failed. ');

end;

procedure TForm3. Label6Click(Sender: TObject);

begin

AutoSendMail;

end;

procedure TForm3. Label6MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

begin

Label6. Font. Color: =clBlue;

Label7. Font. Color: =clBlue;

end;

procedure TForm3. Label6MouseLeave(Sender: TObject);

begin

Label6. Font. Color: =clBlack;

Label7. Font. Color: =clBlack;

end;

end.