ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА АСТАНА

Политехнический колледж города Астана

010000 3706002

##### ДИПЛОМНЫЙ ПРОЕКТ

**АРМ менеджера в автосалоне "A-Motors"**

##### ДИПЛОМНЫЙ ПРОЕКТ

(Пояснительная записка)

ДП.3706.П401.31.06.07.ПЗ

Дипломник Хикимов Н.Б.

Руководитель проекта Лапенко С.А.

Консультант по технологическому разделу Лапенко С.А.

Консультант по экономическому разделу Приходько Л.И.

Консультант по разделу «ТБ и охрана труда» Старунов В.И.

Нормоконтролер Вотчал Г.К.

Рецензент

Дата защиты \_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_

Протокол №\_\_\_\_\_\_\_\_\_\_\_\_

2007

### СОДЕРЖАНИЕ

**Изм.**

**Лист**

**№ докум.**

**Подпись**

**Дата**

**Лист**

**4**

**ДП.3706.П401.**31.06.07.ПЗ

**Разраб.**

**Хикимов Н.**

**Провер.**

**Лапенко С.А.**

**Реценз.**

**Н. Контр.**

**Вотчал Г.К.**

**Утверд.**

Пояснительная записка

**Лит.**

**Листов**

94

**ПКА**

[ВВЕДЕНИЕ 6](#_Toc167351011)

[1. Постановочная часть 8](#_Toc167351012)

[1.1 Формулировка задачи 8](#_Toc167351013)

[1.2 Описание входной и выходной документации 8](#_Toc167351014)

[1.3 Требования к интерфейсу Windows-приложения 8](#_Toc167351015)

[2. Проектная часть 16](#_Toc167351016)

[2.1 Описание информационной базы 16](#_Toc167351017)

[2.2 Спецификации набора данных 18](#_Toc167351018)

[2.3 Спецификации набора данных 18](#_Toc167351019)

[2.4 Проект базы данных, используемой в задаче 19](#_Toc167351020)

[2.5 Разработка алгоритмов обработки данных 20](#_Toc167351021)

[2.6 Разработка SQL-запросов к базе данных 22](#_Toc167351022)

[2.7 Разработка форм приложения, меню, отчетов 23](#_Toc167351023)

[3. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ ДИПЛОМНОГО ПРОЕКТА 26](#_Toc167351024)

[3.1 Краткая характеристика операционных систем 26](#_Toc167351025)

[3.2 Краткая характеристика языка программирования Object Pascal и среды Delphi 26](#_Toc167351026)

[3.3 Краткая характеристика используемой СУБД 28](#_Toc167351027)

[4. ЭКСПЛУАТАЦИЯ 31](#_Toc167351028)

[4.1.Требования к аппаратному обеспечению 31](#_Toc167351029)

[4.2.Инструкция пользователю 31](#_Toc167351030)

[4.3 Инструкция программисту 39](#_Toc167351031)

[5. ЭКОНОМИЧЕСКАЯ ЧАСТЬ 40](#_Toc167351032)

[5.1. Определение затрат на создание программного продукта 40](#_Toc167351033)

[5.2 Расчет себестоимости и цены программного продукта 42](#_Toc167351034)

[5.3 Расчет экономической эффективности проекта 44](#_Toc167351035)

[5.4 Технико-экономические показатели проекта 46](#_Toc167351036)

[6. Мероприятия по технике безопасности и окружающей среды. 47](#_Toc167351037)

[6.1. Охрана труда 47](#_Toc167351038)

[6.2. Техника безопасности 49](#_Toc167351039)

[6.3 Охрана окружающей среды 56](#_Toc167351040)

[7. заключение 59](#_Toc167351041)

[8. Список использованных источников 60](#_Toc167351042)

[Приложение а. основные модули приложения 61](#_Toc167351043)

[А.1 Модуль формы окна «О программе» 61](#_Toc167351044)

[А.2 Модуль формы окна «Зарегистрировать автомобиль» 61](#_Toc167351045)

[А.3 Модуль формы «Удалить автомобиль» 69](#_Toc167351046)

[А.4 Модуль данных 71](#_Toc167351047)

[А.5 Модуль формы «Редактирование данных» 74](#_Toc167351048)

[А.6 Модуль заставки 78](#_Toc167351049)

[А.7 Модуль главной формы 79](#_Toc167351050)

[А.8 Модуль формы поиска 85](#_Toc167351051)

[А.9 Модуль формы менеджеров 89](#_Toc167351052)

[А.10 Модуль формы о владельцах 93](#_Toc167351053)

# ВВЕДЕНИЕ

Потоки информации, циркулирующие в мире, который нас окружает, огромны. Во времени они имеют тенденцию к увеличению. Поэтому в любой организации, как большой, так и маленькой, возникает проблема такой организации управления данными, которая обеспечила бы наиболее эффек­тивную работу. Некоторые организации используют для этого шкафы с папками, но большинство предпочитают компьютеризированные способы – базы данных, позволяющие эф­фективно хранить, структурировать и систематизировать большие объемы дан­ных. И уже сегодня без баз данных невозможно представить работу большинства финансовых, промышленных, торговых и прочих организаций. Не будь баз данных, они бы просто захлебнулись в информационной лавине.

Существует много веских причин перевода существующей информации на компьютерную основу. Сейчас стоимость хранения информации в файлах ЭВМ дешевле, чем на бумаге. Базы данных позволяют хранить, структурировать информацию и извлекать оптимальным для пользователя образом. Использование файл/серверных и клиент/серверных технологий позволяют сберечь значительные средства, а главное и время для получения необходимой информации, а также упрощают доступ и ведение, поскольку они основываются на комплексной обработке данных и централизации их хранения. Кроме того ЭВМ позволяет хранить любые форматы данных текст, чертежи, данные в рукописной форме, фотографии, записи голоса и т.д.

Для использования столь огромных объемов хранимой информации, помимо развития системных устройств, средств передачи данных, памяти необходимы средства обеспечения диалога человек-ЭВМ, которые позволяют пользователю вводить запросы, читать файлы, модифицировать хранимые данные, добавлять новые данные или принимать решения на основании хранимых данных. Для обеспечения этих функций созданы специализированные средства – системы управления базами данных (СУБД). Современные СУБД - многопользовательские системы управления базой данных, которые специализируется на управлении массивом ин­формации одним или множеством одновременно работающих пользователей.

### Актуальность и цель дпломного проекта

Данный программный продукт разработан для менеджеров по продажам в автосалоне «A-Motors». Программа предназначена для регистрации и ведения учета продаж автомобилей.

В данный момент у нас в городе существует несколько автомобильных рынков занимающихся продажей, в которой они выступают как посредники. Менеджеры по продажам, работающие на авторынках тратят массу времени на бумажную работу. Время на регистрацию автомобиля и поиск нужного бланка в куче папок на столе и полках очень обременяет, это неудобство побудило меня автоматизировать данный процесс.

Разрабатываемый программный продукт должен оперативно производить поиск, сортировки, составлять различного рода запросы по пожеланиям клиента, а также вести отчётность.

# 1. Постановочная часть

## 

## 1.1 Формулировка задачи

Задачей дипломного проекта является разработка программы для автоматизации процесса регистрации и учета продаж автомобилей, и с последующей возможностью распечатки.

Программа, создаваемая в данном дипломном проекте предназначена для производственно-технического отдела продаж автомобильного салона «А-Motors». Программа должна осуществлять управление данными об автомобилях принятых на реализацию автомобильным салоном, регистрировать характеристики автомобиля, осуществлять поиск в базе автомобилей по заданным критериям. Выводить на печать результаты поиска, а также отчет о проданных автомобилях.

## 

## 1.2 Описание входной и выходной документации

Входные документы

1. Технический паспорт автомобиля – содержит основные паспортные данные автомобиля – выдается при регистрации автомобиля в РЭО УДП УВД РК.
2. Графическое изображение автобусов (рисунки, фотографии)
3. Анкета владельца транспортного средства
4. Акт о приеме автомобиля на реализацию

Выходные документы

1. Карточка автомобиля
2. Список автомобилей удовлетворяющих критериям поиска
3. Список проданных автомобилей за заданный период
4. Перечень зарегистрированных марок автомобилей

## 1.3 Требования к интерфейсу Windows-приложения

Под графическим интерфейсом пользователя (Graphical User Interface — GUI) подразумевается тип экранного представления, при котором пользователь может выбирать команды, запускать задачи и просматривать списки файлов, указывая на пиктограм­мы или пункты в списках меню, показанных на экране. Дейст­вия могут, как правило, выполняться с помощью мыши, либо на­жатием клавиш на клавиатуре. Типичным примером графиче­ского интерфейса пользователя является Windows 95/98.

Delphi предоставляет разработчику приложения широкие воз­можности быстрого и качественного проектирования графиче­ского интерфейса пользователя — различных окон, кнопок, меню и т.д. Есть определенные принципы построения графического ин­терфейса пользователя, и пренебрегающий ими обречен на то, что его приложение будет выглядеть чужеродным объектом в среде Windows.

Для пользователя одним из принципиальных преимуществ работы с Windows является то, что большинство имеющихся приложений выглядят и ведут себя сходным образом. После того, как вы поработаете с несколькими приложениями, вы обна­ружите, что можете заранее почти наверняка сказать, где можно найти ту или иную функцию в программе, которую только что приобрели, или какие быстрые клавиши надо использовать для выполнения тех или иных операций.

Чаще всего сколько-нибудь сложное приложение не может ограничиться одним окном. Поэтому прежде всего вам нужно ре­шить вопрос управления окнами. Есть две различные модели приложений: с интерфейсом одного документа (SDI) и с интер­фейсом множества документов (MDI).

В большинстве случаев следует отдавать предпочтение интер­фейсу SDI. Этот интерфейс не обязательно предполагает наличие действительно только одного окна, как в приложениях Windows, типа «Калькулятор». Такое приложение, как «Проводник» Win­dows, также является SDI приложением, но в нужные моменты оно создает вторичные окна для поиска файлов или папок, зада­ния параметров, просмотра свойств файлов и других целей.

Основным элементом любого приложения является форма — контейнер, в котором размещаются другие визуальные и невизуальные компоненты. С точки зрения пользователя форма — это окно, в котором он работает с приложением.

К внешнему виду окон в Windows предъявляются определен­ные требования. К счастью, Delphi автоматически обеспечивает стандартный для Windows вид окон вашего приложения. Но вам надо продумать и указать, какие кнопки в полосе системного меню должны быть доступны в том или ином окне, должно ли окно допускать изменение пользователем его размеров, каким должен быть заголовок окна. Все эти характеристики окон обес­печиваются установкой и управлением свойствами формы.

Без особой необходимости не делайте окна приложения с изменяемыми по­льзователем размерами. При изменении размеров, если не применены специальные приемы, нарушается компоновка окна и пользователь ничего не выигрывает от своих операций с окном. Окно имеет смысл делать с изменяемыми размерами, только если это позволяет пользо­вателю изменять полезную площадь каких-то расположенных в нем компо­нентов отображения и редактирования информации: текстов, изображе­ний, списков и т.п.

Цвет является мощным средством воздействия на психику че­ловека. Именно поэтому обращаться с ним надо очень осторож­но. Неудачное цветовое решение может приводить к быстрому утомлению пользователя, работающего с вашим приложением, к рассеиванию его внимания, к частым ошибкам. Слишком яркий или неподходящий цвет может отвлекать внимание пользовате­ля или вводить его в заблуждение, создавать трудности в работе. А удачно подобранная гамма цветов, осмысленные цветовые акценты снижают утомляемость, сосредоточивают внимание поль­зователя на выполняемых в данный момент операциях, повыша­ют эффективность работы. С помощью цвета вы можете на что-то намекнуть или привлечь внимание к определенным областям эк­рана. Цвет может также связываться с различными состояниями объектов.

Надо стремиться использовать ограниченный набор цветов и уделять внимание их правильному сочетанию. Расположение яр­ких цветов, таких, как красный, на зеленом или черном фоне за­трудняет возможность сфокусироваться на них. Не рекомендует­ся использовать дополнительные цвета. Обычно наиболее прием­лемым цветом для фона будет нейтральный цвет, например, свет­ло-серый (используется в большинстве продуктов Microsoft). По­мните также, что яркие цвета кажутся выступающими из плос­кости экрана, в то время как темные как бы отступают вглубь.

Цвет не должен использоваться в качестве основного средства передачи информации. Можно использовать различные панели, формы, штриховку и другие методики выделения областей экра­на. Microsoft даже рекомендует разрабатывать приложение сна­чала в черно-белом варианте, а уже потом добавлять к нему цвет.

Нельзя также забывать, что восприятие цвета очень индиви­дуально. А по оценке Microsoft девять процентов взрослого насе­ления вообще страдают нарушениями цветовосприятия. Поэтому не стоит навязывать пользователю свое видение цвета, даже если оно безукоризненно. Надо предоставить пользователю возмож­ность самостоятельной настройки на наиболее приемлемую для него гамму. К тому же не стоит забывать, что может быть кто-то захочет использовать вашу программу на машине с монохром­ным монитором.

Статические цвета вы выбираете сами и они будут оставаться неизменными при работе приложения на любом компьютере. Это не очень хорошо, поскольку пользователь не сможет адаптиро­вать вид вашего приложения к своим потребностям. При выборе желательной ему цветовой схемы пользователь может руководст­воваться самыми разными соображениями: начиняя с практиче­ских (например, он может хотеть установить черный фон, чтобы экономить энергию батареи), и кончая эстетическими (он может предпочитать, например, шкалу оттенков серого, потому что не различает цвета). Все это он не может делать, если вы задали в приложении статические цвета. Но уж если по каким-то сообра­жениям вам надо их задать, старайтесь использовать базовый на­бор из 16 цветов. Если вы попытаетесь использовать 256 (или, что еще хуже, 16 миллионов) цветов, это может замедлить работу вашего приложения, или оно будет выглядеть плохо на машине пользователя с 16 цветами. К тому же подумайте (а, как прави­ло, это надо проверить и экспериментально), как будет выгля­деть ваше приложение на монохромном дисплее.

Исходя из изложенных соображений, везде, где это имеет смысл, следует использовать для своего приложения палитру си­стемных цветов. Это те цвета, которые устанавливает пользова­тель при настройке Windows. Когда вы создаете новую форму или размещаете на ней компоненты, Delphi автоматически при­сваивает им цвета в соответствии со схемой цветов, установлен­ной в Windows. Конечно, вы будете менять эти установки по умолчанию. Но если при этом вы используете соответствующие константы системных цветов, то, когда пользователь изменит цветовую схему оформления экрана Windows, ваше приложение также будет соответственно меняться, и не будет выпадать из об­щего стиля других приложений.

Не злоупотребляйте в приложении яркими цветами. Пестрое приложение — обычно признак дилетантизма разработчика, утомляет пользователя, рас­сеивает его внимание. Как правило, используйте системные цвета, которые пользователь может перестраивать по своему усмотрению. Из статических цветов обычно имеет смысл использовать только clBlack — черный, clWhite — белый и clRed — красный цвет предупреждения об опасности.

Использование шрифтов по умолчанию: System или MS Sans Serif, чаще всего позволяет избежать неприятностей. Впрочем, увы, не всегда. Если вы используете для надписей рус­ские тексты, то при запуске приложения на компьютере с неру­сифицированным Windows иногда возможны неприятности. Для подобных случаев все-таки полезно приложить файлы использо­ванных шрифтов к вашей программе.

Другой выход из положения — ввести в приложение команду выбора шрифта пользователем. Это позволит ему выбрать подхо­дящий шрифт из имеющихся в его системе. Проведенную пользователем установку можно запоминать в файле .INI, в реестре или в файле конфигурации и читать авто­матически информацию из этого файла при каждом запуске при­ложения (см. разделы 7.3 и 7.4).

Практически любое приложение должно иметь меню, поско­льку именно меню дает наиболее удобный доступ к функциям программы. Существует несколько различных типов меню: глав­ное меню с выпадающими списками разделов, каскадные меню, в которых разделу первичного меню ставится в соответствие спи­сок подразделов, и всплывающие или контекстные меню, появ­ляющиеся, если пользователь щелкает правой кнопкой мыши на каком-то компоненте.

Основное требование к меню — их стандартизация. Это требо­вание относится ко многим аспектам меню: месту размещения заголовков меню и их разделов, форме самих заголовков, клави­шам быстрого доступа, организации каскадных меню. Цель стан­дартизации — облегчить пользователю работу с приложением. Надо, чтобы пользователю не приходилось думать, в каком меню и как ему надо открыть или сохранить файл, как ему получить справку, как работать с буфером обмена Clipboard и т.д. Для осу­ществления всех этих операций у пользователя, поработавшего хотя бы с несколькими приложениями Windows, вырабатывает­ся стойкий автоматизм действий и недопустимо этот автоматизм ломать.

Начнем рассмотрение требований с размещения заголовков меню. Ко­нечно, состав меню зависит от конкретного приложения. Но раз­мещение общепринятых разделов должно быть стандартизиро­ванным. Все пользователи уже привыкли, что меню Файл разме­щается слева в полосе главного меню, раздел справки — справа, перед ним в приложениях MDI размещается меню Окно и т.д. Главное меню должно также снабжаться инструментальной па­нелью (см. рис. 1.5), быстрые кнопки которой дублируют наибо­лее часто используемые команды меню. На этих кнопках надо использовать, по возможности, привычные картинки.

По возможности стандартным должно быть и расположение разделов в выпадающих меню.

Группы функционально связанных разделов отделяются в вы­падающих меню разделителями.

Названия разделов меню должны быть привычными пользо­вателю. Если вы не знаете, как назвать какой-то раздел, не изоб­ретайте свое имя, а попытайтесь найти аналогичный раздел в ка­кой-нибудь русифицированной программе Microsoft для Win­dows. Названия должны быть краткими и понятными. Не испо­льзуйте фраз, да и вообще больше двух слов, поскольку это пере­гружает экран и замедляет выбор пользователя. Названия разде­лов должны начинаться с заглавной буквы.

Названия разделов меню, связанных с вызовом диалоговых окон, должны заканчиваться многоточием, показывающим поль­зователю, что при выборе этого раздела ему предстоит установить в диалоге еще какие-то параметры.

Разделы, к которым относятся каскадные меню должны заканчиваться стрел­кой, указывающей на наличие дочернего меню данного раздела.

В каждом названии раздела должен быть выделен подчерки­ванием символ, соответствующий клавише быстрого доступа к разделу (клавиша Alt плюс подчеркнутый символ). Хотя вряд ли такими клавишами часто пользуются, но традиция указания та­ких клавиш незыблема. В реальной работе, вероятно, они испо­льзуются только в случае, когда отказала мышь.

Многим разделам могут быть поставлены в соответствие «го­рячие» клавиши, позволяющие обратиться к команде данного раздела, даже не заходя в меню. Комбинации таких «горячих» клавиш должны быть традиционными. Например, команды вы­резания, копирования и вставки фрагментов текста практически всегда имеют «горячие» клавиши Ctrl-X, Ctrl-C и Ctrl-V соответст­венно. Заданные сочетания клавиш отображаются в заголовках соответствующих разделов.

Каждое окно, которое вы вводите в свое приложение, должно быть тщательно продумано и скомпоновано. Удачная компонов­ка может стимулировать эффективную работу пользователя, а неудачная — рассеивать внимание, отвлекать, заставлять тра­тить лишнее время на поиск нужной кнопки или индикатора.

Управляющие элементы и функционально связанные с ними компоненты экрана должны быть зрительно объединены в груп­пы, заголовки которых коротко и четко поясняют их назначение. Такое объединение позволяют осуществлять различные панели. Можно рекомендовать, как правило, размещать компоненты не непосредственно на форме, а на панелях. Но и внутри панелей надо продумывать размещение компонентов как с точки зрения эстетики, так и с точки зрения визуального отражения взаимоот­ношений элементов. Например, если имеется кнопка, которая разворачивает окно списка, то эти два компонента должны быть визуально связаны между собой: размещены на одной панели и в непосредственной близости друг от друга. Если же ваш экран представляет собой случайные скопления кнопок, то именно так он и будет восприниматься. И в следующий раз пользователь не захочет пользоваться вашей программой.

Каждое окно должно иметь некоторую центральную тему, ко­торой подчиняется его композиция. Пользователь должен пони­мать, для чего предназначено данное окно и что в нем наиболее важно. При этом недопустимо перегружать окно большим чис­лом органов управления, ввода и отображения информации. В окне должно отображаться главное, а все детали и дополнитель­ную информацию можно отнести на вспомогательные окна. Для этого полезно вводить в окно кнопки с надписью Больше..., много­точие в которой показывает, что при нажатии этой кнопки от­кроется вспомогательное окно с дополнительной информацией.

Помогают также разгрузить окно многостраничные компоненты с закладками. Они дают возможность пользователю легко пере­ключаться между разными по тематике страницами, на каждой из которых имеется необходимый минимум информации.

Еще один принцип, которого надо придерживаться при проек­тировании окон — стилистическое единство всех окон в прило­жении. Недопустимо, чтобы сходные по функциям органы управления в разных окнах назывались по-разному или разме­щались в разных местах окон. Все это мешает работе с приложе­нием, отвлекает пользователя, заставляет его думать не о сущно­сти работы, а о том, как приспособиться к тому или иному окну.

При проектировании приложения важно правильно опреде­лить последовательность табуляции оконных компонентов. Под этим понимается последовательность, в которой переключается фокус с компонента на компонент, когда пользователь нажимает клавишу табуляции Tab. Это важно, поскольку в ряде случаев по­льзователю удобнее работать не с мышью, а с клавиатурой. Пусть, например, вводя данные о каком-то сотруднике, пользователь должен в отдельных окнах редактирования указать фами­лию, имя и отчество. Конечно, набрав фамилию, ему удобнее на­жать клавишу Tab и набирать имя, а потом опять, нажав Tab, набирать отчество, чем каждый раз отрываться от клавиатуры, хватать мышь и переключаться в новое окно редактирования.

Приложение должно предельно облегчать работу пользовате­ля, снабжая его системой подсказок, помогающих сориентирова­ться в приложении. Эта система включает в себя:

* Ярлычки, которые всплывают, когда пользователь задержит курсор мыши над каким-то элементом окна приложения. В частности, такими ярлычками обязательно должны снабжать­ся быстрые кнопки инструментальных панелей, поскольку нанесенные на них пиктограммы часто не настолько вырази­тельны, чтобы пользователь без дополнительной подсказки мог понять их назначение.
* Более развернутые подсказки в панели состояния или в другом отведенном под это месте экрана, которые появляются при перемещении курсора мыши в ту или иную область окна приложения.
* Встроенную систему контекстно-зависимой оперативной справки, вызываемую по клавише F1.
* Раздел меню Справка, позволяющий пользователю открыть стандартный файл справки Windows.hlp, содержащий в виде гипертекста развернутую информацию по интересующим по­льзователя вопросам.

При работе программы могут возникать различного рода ошибки: переполнение, деление на нуль, попытка открыть несу­ществующий файл и т.п. При возникновении таких исключите­льных ситуаций программа генерирует так называемое исключе­ние я выполнение дальнейших вычислений в данном блоке пре­кращается. Исключение — это объект специального вида, харак­теризующий возникшую в программе исключительную ситуа­цию. Он может также содержать в виде параметров некоторую уточняющую информацию. Особенностью исключений является то, что это сугубо временные объекты. Как только они обработа­ны каким-то обработчиком, они разрушаются.

Программист должен принять все мыслимые меры, чтобы ни при каких ошибках пользователя и ни при каких сочетаниях данных приложение не за­канчивалось бы аварийно. Но если все-таки аварийное завершение проис­ходит, необходима полная зачистка «мусора» — удаление временных фай­лов, освобождение памяти, разрыв связей с базами данных и т.д.

# 2. Проектная часть

## 

## 2.1 Описание информационной базы

Данный программный продукт имеет шесть таблиц БД.

Таблица 2.1 Владельцы - *vladelec.dbf*

|  |  |  |  |
| --- | --- | --- | --- |
| *Наименование поля* | *Тип* | *Размер* | *Назначение* |
| \* Kod\_vlad | Number | 5 | Код владельца |
| Fam | Character | 20 | Фамилия |
| Name | Character | 20 | Имя |
| Oth | Character | 20 | Отчество |
| Adres | Character | 20 | Адрес |
| Mail | Character | 30 | Адрес электронной почты |
| Tel | Number | 20 | Номер телефон |
| Sot | Number | 20 | Номер мобильного телефона |

Таблица 2.2 Менеджеры - *sotrud.dbf*

|  |  |  |  |
| --- | --- | --- | --- |
| *Наименование поля* | *Тип* | *Размер* | *Назначение* |
| \* Kod\_sot | Number | 3 | Код сотрудника |
| Fam | Character | 20 | Фамилия |
| Name | Character | 20 | Имя |
| Oth | Character | 20 | Отчество |

Таблица 2.3 Автомобили - *avto.dbf*

|  |  |  |  |
| --- | --- | --- | --- |
| *Наименование поля* | *Тип* | *Размер* | *Назначение* |
| \* Kod\_avto | Number | 6 | Код автомобиля |
| Kod\_vlad | Number | 5 | Код владельца |
| Kod\_Marka | Number | 6 | Код марки |
| Model | Character | 20 | Модель автомобиля |
| V | Character | 5 | Объем двигателя |
| Gos\_nom | Character | 8 | Государственный номер |
| Cvet | Character | 20 | Цвет |
| Tip\_kuz | Character | 20 | Тип кузова |
| Foto\_1 | Character | 20 | Фотография 1 |
| Foto\_2 | Character | 20 | Фотография 2 |
| Cena | Number | 10 | Цена |

Таблица 2.4 Регистрация - *avto\_in.dbf*

|  |  |  |  |
| --- | --- | --- | --- |
| *Наименование поля* | *Тип* | *Размер* | *Назначение* |
| \* Kod\_in | Number | 6 | Регистрационный номер |
| Akt\_nom | Number | 6 | Номер акта |
| Kod\_avto | Number | 6 | Код автомобиля |
| Kod\_vlad | Number | 6 | Код владельца |
| Kod\_sot | Number | 6 | Код менеджера |
| Data | Date |  | Дата регистрации |

Таблица 2.5 Комплектность - *komplect.dbf*

|  |  |  |  |
| --- | --- | --- | --- |
| *Наименование поля* | *Тип* | *Размер* | *Назначение* |
| Kod\_avto | Number | 6 | Код автомобиля |
| Koleso | Character | 15 | Оформление колес |
| Zapaska | Logical |  | Наличие запасного колеса |
| Instrument | Logical |  | Наличие инструмента |
| Music | Logical |  | Наличие магнитолы |
| Medic | Logical |  | Наличие аптечки |
| Elect\_pak | Logical |  | Наличие электропакета |
| ABS | Logical |  | Наличие АБС |
| GUR | Logical |  | Наличие гидроусилителя рулевого колеса |
| Kondic | Logical |  | Наличие кондиционера |
| Korobka | Character | 15 | Тип КПП |
| Privod | Character | 15 | Вид привода |
| Info | Memo |  | Дополнительная информация |
| Defekt | Memo |  | Описание дефектов |

Таблица 2.6 Продажа - *avto\_out.dbf*

|  |  |  |  |
| --- | --- | --- | --- |
| *Наименование поля* | *Тип* | *Размер* | *Назначение* |
| \* Kod\_out | Number | 6 | Код продажи |
| Data | Date |  | Дата продажи |
| Marka\_out | Number | 6 | Код марки |
| Model\_out | Character | 20 | Модель автомобиля |
| V\_out | Character | 5 | Объем двигателя |
| Cvet\_out | Character | 20 | Цвет |
| Tipkuz\_out | Character | 20 | Тип кузова |
| Cena | Number | 10 | Цена |

Таблица 2.7 Марки автомобилей - *marka.dbf*

|  |  |  |  |
| --- | --- | --- | --- |
| *Наименование поля* | *Тип* | *Размер* | *Назначение* |
| \* Kod\_marka | Number | 6 | Код марки |
| Marka | Character | 20 | Марка автомобиля |

## 

## 2.2 Спецификации набора данных

Таблица 2.8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя набора данных в программе | Имя набора данных на внешнем носителе | Длина записи набора данных (в байтах) | Тип файла | Метод доступа |
| Vlad | vladelec.dbf | 155 | Dbase IV | Произвольный |
| Sot | sotrud.dbf | 63 | Dbase IV | Произвольный |
| Avto | avto.dbf | 140 | Dbase IV | Произвольный |
| Avto\_in | avto\_in.dbf | 36 | Dbase IV | Произвольный |
| Complect | komplect.dbf | >52 | Dbase IV | Произвольный |
| Auto\_out | avto\_out.dbf | 93 | Dbase IV | Произвольный |
| Marka | Marka.bdf | 26 | Dbase IV | Произвольный |

## 

## 2.3 Спецификации набора данных

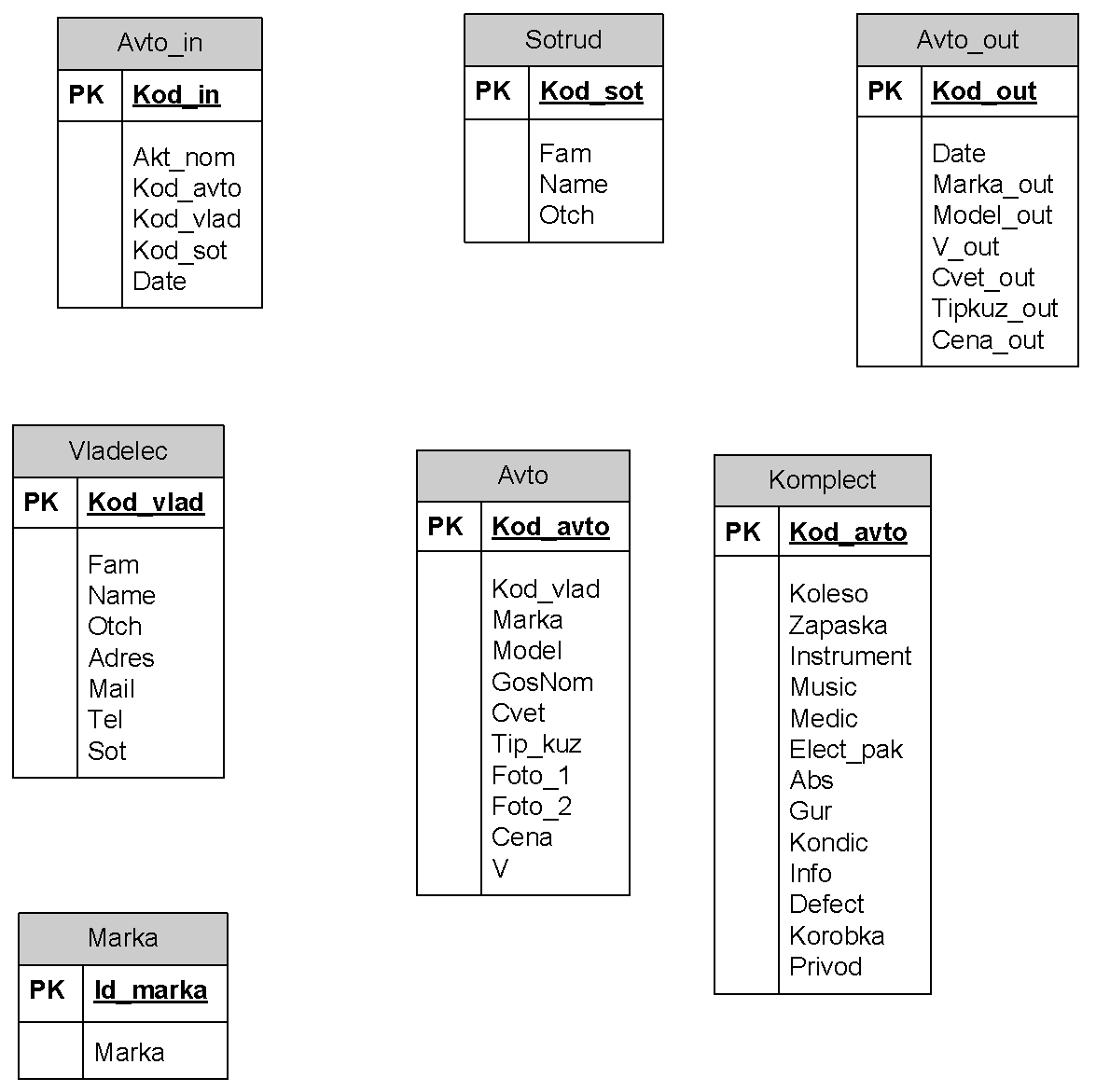
Таблица 2.9

|  |  |  |
| --- | --- | --- |
| Обозначение | Назначение | Примечание |
| 1 | 2 | 3 |
| vladelec.dbf | Информация о владельцах |  |
| sotrud.dbf | Информация о менеджерах |  |
| avto.dbf | Информация об автомобилях |  |
| avto\_in.dbf | Регистрация автомобиля |  |
| komplect.dbf | Комплектация автомобиля |  |

Продолжение таблицы 2.9

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| avto\_out.dbf | Проданные автомобили |  |
| Marka.bdf | Справочник марок автомобилей |  |

## 2.4 Проект базы данных, используемой в задаче



## 2.5 Разработка алгоритмов обработки данных

### 2.5.1 Алгоритм работы главной формы приложения

Начало

Отобразить соответ-ствующую форму

Выбать команду

Выход?

Определить команды главного меню

Конец

+

-

Схема 3.1

### 2.5.2 Алгоритм работы форм приложения

Начало

Выбрать действие

Закрыть форму?

Выполнить действие заданное формой

Конец

**+**

**-**

Схема 3.2

### 2.5 Таблица идентификаторов

Таблица 2.10

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Смысловое содержание | Тип | Разрядность |
| Main.path | Путь к папке программы | string |  |
| Add.pic1 | Путь к первому изображению | String |  |

Продолжение таблицы 2.10

|  |  |  |  |
| --- | --- | --- | --- |
| Add.pic2 | Путь ко второму изображению | string |  |
| Del. AppProgID | Идентификатор приложения Microsoft Excel | string |  |
| Del.App | Указатель на приложение Microsoft Excel | Variant |  |
| Del.Ke | Указатель на рабочую книгу Microsoft Excel | Variant |  |
| Del.Abc | Указатель на лист Microsoft Excel | Variant |  |
| Del.Result | Предназначена для проверки удачной активизации приложения Mcirosoft Excel | HResult |  |
| Del.i | Счетчик цикла | integer |  |

## 

## 2.6 Разработка SQL-запросов к базе данных

Запрос на выборку автомобилей по необходимым критериям формируется динамически, в зависимости от выбранных параметров поиска.

procedure TFSearch.LMDButton1Click(Sender: TObject);

begin

Query1.Close;

Query1.SQL.Clear;

try

Query1.Sql.Add('select a.kod\_avto, m.marka, a.model, a.cvet, a.tip\_kuz, a.cena, a.v, k.korobka,k.privod ');

Query1.SQL.Add('from avto a, komplect k, marka m ');

Query1.SQL.Add('where ');

if checkBox9.Checked then Query1.SQL.Add('a.marka = '+ Inttostr(DBLookupComboBox1.keyvalue)+' and');

if checkBox10.Checked then Query1.SQL.Add('a.tip\_kuz="'+ComboBox1.Text+

'" and ');

if checkBox11.Checked then Query1.SQL.Add('a.v<='+Edit1.Text+' and ');

if checkBox12.Checked then Query1.SQL.Add('a.cena<='+Edit2.Text+' and ');

if checkBox4.Checked then Query1.SQL.Add('k.elect\_pak = '+

BoolToStr(CheckBox17.Checked)+' and ');

if checkBox5.Checked then Query1.SQL.Add('k.kondic = '+

BoolToStr(CheckBox16.Checked)+' and ');

if checkBox6.Checked then Query1.SQL.Add('k.music = '+ BoolToStr(CheckBox15.Checked)+' and ');

if checkBox7.Checked then Query1.SQL.Add('k.gur = '+ BoolToStr(CheckBox14.Checked)+' and ');

if checkBox8.Checked then Query1.SQL.Add('k.abs = '+ BoolToStr(CheckBox13.Checked)+' and ');

if checkBox1.Checked then Query1.SQL.Add('k.koleso = "'+ComboBox2.Text+  
'" and ');

if checkBox2.Checked then Query1.SQL.Add('k.korobka = "'+ComboBox3.Text+  
'" and ');

if checkBox3.Checked then Query1.SQL.Add('k.privod = "'+ComboBox4.Text+  
'" and ');

Query1.SQL.Add('a.marka=m.id\_marka and a.kod\_avto=k.kod\_avto');

Query1.Open;

except

MessageDlg('Введены не все данные',mtWarning,[mbOk],0);

end;

end;

## 2.7 Разработка форм приложения, меню, отчетов

Разработка форм, меню и множество других графических элементов относятся к дизайну приложения. Дизайн приложения или интерфейс пользователя, должен быть прост, удобен.

Для написания данной программы я использовал как стандартный набор компонентов палитры Delphi 7, так и дополнительные (RXLib, GlobusLib, LmdTools v6.12.00 Professional For Delphi 7).

В программе реализованы такие функции как отчёт, он же экспорт в Excel. Данные которые отображаются в таблице БД по инициативе пользователя могут быть помещены в отчёт. Все отчёты созданные программой хранятся во вложенном каталоге «Reports». Название отчётов формируется из параметров: место в программе откуда был произведён отчёт и текущая дата.

Меню, спроектированное на главной форме можно представить следующим образом:

* Файл
  + Выход
* Справочники
  + Менеджеры
  + Владельцы
  + Марки
  + Проданные автомобили
* Операции
  + Зарегистрировать автомобиль
  + Редактировать данные
  + Продать автомобиль
  + Поиск
* Вид
  + Размер шрифта
    - Большой
    - Средний
    - Маленький
  + Цвет шрифта
* Помощь
  + Справка
  + О программе

Подробное описание назначения и использования команд главного меню описано в инструкции пользователю (п. 5.2).

Наиболее часто используемые команды вынесены на формы в виде кнопок, кроме того, на некоторые команды назначены горячие функциональные клавиши.

В приложении использованы стандартные системные цвета, что позволит пользователю использовать удобную для себя цветовую палитру.

Ко всем элементам интерфейса определены всплывающие подсказки, указывающие назначение данного элемента.

По возможности произведена обработка исключений, генерируемых при работе с базой данных.

# 3. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ ДИПЛОМНОГО ПРОЕКТА

## 

## 3.1 Краткая характеристика операционных систем

Windows 95, Windows 98, Windows NT, Windows 2000 - это последнее воплощение графических операционных систем, впервые представленных в ноябре 1985 и последующем развитии до 1999 года для использования на компьютерах типа IBM PC и совместимых с ним. По мере проникновения на рынок, за последнее десятилетие, Windows 95, Windows NT, Windows 98.Windows 2000 почти полностью вытеснили всех имевшихся конкурентов и стали, фактическими, эталонами операционной системы для персональных компьютеров. Теперь, если вы пишете программу для совместимых с IBM PC компьютеров, то вы пишете для Windows95, Windows NT, Windows 98,Windows 2000.

Windows обладает важными преимуществами и для пользователей, и для программистов по сравнению со средой MS-DOS. Выгоды для пользователей и выгоды для создателей программ на самом деле весьма схожи, поскольку задача создателя программы состоит в том, чтобы дать пользователю то, в чём он нуждается и то, что он хочет. Windows 95, Windows NT, Windows 98, Windows 2000 делает это возможным.

Операционная система не может реализовывать многозадачность без управления памятью. Так как одни программы запускаются, а другие завершаются, память фрагментируется. Система должна быть способной объединять свободное пространство.

## 

## 3.2 Краткая характеристика языка программирования Object Pascal и среды Delphi

Delphi - это комбинация нескольких важнейших технологий:

* Высокопроизводительный компилятор в машинный код
* Объектно-ориентированная модель компонент
* Визуальное (а, следовательно, и скоростное) построение приложений из программных прототипов
* Масштабируемые средства для построения баз данных

Повсеместное признание концепции объектно-ориентированного программирования и компонентного подхода к созданию оттеснило на второй план вопросы программирования, без которого невозможна разработка действительно серьезных, профессиональных программ.

Если рассмотреть все компоненты библиотеки визуальных компонентов Delphi с точки зрения реализуемых функций, то окажется, что выполняемая компонентом основная операция состоит из нескольких более простых. Причем этих простых операций не так уж и много – это ввод и вывод данных, работа с адресным пространством, обработка списков, использование объектов.

Поэтому разъезженная в десятках книг метафора приобретает новый смысл. Оказывается и в самом программировании в Delphi также можно выделить некую модульную структуру. При этом под программированием понимается создание собственного исходного кода приложения, а не перенос компонентов на форму и настройка их в Инспекторе объектов, - для этого процесса имеется хорошее и емкое определение – визуальное программирование.

Итак, в разработке программ – визуальное программирование обеспечивает быстрое и качественное выполнение стандартных задач. Программирование позволяет решать нестандартные и сложные задачи и увеличивать эффективность, хотя при этом используются те же самые программные инструменты.

Концепция Delphi была реализована в конце 1994 года, когда вышла первая версия среды разработки. В основу программного продукта легли концепции объектно-ориентированного программирования на базе языка Object Pascal и визуального подхода к построению приложений. Именно благодаря этому сочетанию каждая новая версия среды разработки становилась событием.

Среда Delphi 5 представляет собой оболочку разработчика, в которую входит набор специализированных программ, ответственных за разные этапы создания готового приложения. Исходный текст программы готовится в среде Delphi 5 с помощью встроенного редактора исходных текстов. Этот редактор специализирован. Он отличается гибкими возможностями цветового выделения различных элементов текста программы и предоставляет возможность быстрого ввода часто встречающихся конструкций.

Левая панель редактора представляет собой Проводник, позволяющий быстро перемещать между частями исходного текста и по структуре создаваемой программе.

Важнейшая характеристика разрабатываемой программы – удобство ее пользовательского интерфейса, наличие и доступность необходимых элементов управления. В системе Delphi 5 имеется специальный проектировщик форм, с помощью которого окна 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.

Любая работа в Delphi начинается с создания нового проекта. Как правило, из одного проекта получается одно приложение. Таким образом, проект представляет собой фундамент или каркас, на который “крепятся” все нужные детали. Самый маленький проект представляет собой только один файл с исходным кодом проекта, имеющий расширение DPR, он называется главным файлом проекта. Серьезный большой проект содержит десятки разнообразных форм, модулей, а также динамические библиотеки и ресурсы и т.д.

В главном файле содержится описание всех модулей и форм, входящих в состав проекта. Кроме того, в секции begin…end может располагаться любой код, который должен выполняться перед началом работы приложения или в процессе его закрытия.

Модулем называется текстовый файл с расширением PAS , в котором содержится некоторый исходный код на языке Object Pascal. С модулем может быть связана форма, которая представляет собой отдельно взятое окно программы с расположением на нем интерфейсными элементами. Служебная информация о формах хранится в файлах с расширением DFM. В общем случае в состав проекта могут входить модули, не имеющие связанных с ними форм.

В проекте Delphi помимо модулей и форм могут присутствовать различные служебные файлы. Среди них важное место занимают ресурсы (расширение RES), которые хранят значки, курсоры, строки сообщений, информацию о версии программы и т.д.

Группа проектов объединяет для совместной разработки несколько проектов. Файл группы проектов имеет расширение BPG и представляет собой текстовый файл с системной информацией и описанием проектов, входящих в группу.

## 3.3 Краткая характеристика используемой СУБД

Основные идеи современной информационной технологии базируются на концепции, согласно которой данные должны быть организованы в базы данных с целью адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей. Эти базы данных создаются и функционируют под управлением специальных программных комплексов, называемых системами управления базами данных (СУБД).

Увеличение объема и структурной сложности хранимых данных, расширение круга пользователей информационных систем привели к широкому распространению наиболее удобных и сравнительно простых для понимания реляционных (табличных) СУБД. Для обеспечения одновременного доступа к данным множества пользователей, нередко расположенных достаточно далеко друг от друга и от места хранения баз данных, созданы сетевые мультипользовательские версии СУБД. В них тем или иным путем решаются специфические проблемы параллельных процессов, целостности (правильности) и безопасности данных, а также санкционирования доступа.

### Формат dBase

Этот формат представляет собой обычный плоский файл \*.DBF. Записи расположены в порядке добавления, поскольку новые записи помещаются всегда в конец файла. Если Вы хотите просматривать записи в порядке, отличном от естественного, то Вы должны построить индекс. Индекс имеет структуру B-дерева, элементами которого являются ключевые значения и номера соответствующих записей в DBF-файле. Таким образом, при поиске или просмотре данных по индексу, из индекса берется номер записи, соответствующий конкретному значению ключа, и осуществляется выборка нужной записи из DBF-файла. Если в индексе ключевые значения расположены последовательно, то соответствующие им записи в DBF - произвольно, и при таком способе доступа происходят частые перемещения по файлу с записями, что безусловно замедляет работу

(например FILTER в Clipper) и более того, при больших объемах данных делает работу аппаратного или программного кэша практически бесполезной (кэш в этом случае помогает только индексу).

Для удаления записей применяется следующий подход - чтобы не оставлять "пустоты" в файле записей, при удалении первый байт записи принимает специальное значение, индицирующее, что запись удалена. Такие записи исключаются из просмотра, и как-бы отсутствуют. Безусловно, есть средства, позволяющие просматривать таблицу и с

удаленными записями, а также восстанавливать их (отменять флаг удаления). Номера записей остаются соответствующими физическим, и если удалена запись N 5, то при просмотре подряд идут записи с номерами 4 и 6.

При упаковке таблицы удаленные записи затираются записями, находящимися после них по порядку. Т.к. записи переместились (и изменились их физические порядковые номера), все индексы нужно перестроить.

Как результат, ориентация программы на номера записей DBF-файла имеет недостатки - номера записей никак не связаны с содержимым записей, и могут меняться после упаковки файла. Однако механизм закладок BDE для формата DBF использует именно номера записей, т.к. они стабильны в течение сеанса работы, и не зависят от используемого индекса.

# 4. ЭКСПЛУАТАЦИЯ

## 

## 4.1.Требования к аппаратному обеспечению

Чтобы комфортно эксплуатировать программу требуется, чтобы компьютер был сконфигурирован следующим образом:

1. Процессор Pentium 800MHz (рекомендуется 1500 MHz);

2. Оперативная память (ОЗУ) 128 MB (рекомендуется 256 Mb);

3. Свободного места на винчестере 100 Mб;

4. Window XР, MS Office (можно не весь пакет, а только Excel для отчётов)

## 4.2.Инструкция пользователю

После загрузки программы на экране появляется основное рабочее окно. В данном окне отображаются основные данные, об имеющихся в наличии автомобилях, и их характеристики. Из этого окна можно перейти к любому модулю программы по нажатию на соответствующую кнопку на панели кнопок.

По нажатию на кнопку «Менеджеры» - Открывается справочник менеджеров. Перейти к справочнику менеджеров так же можно при помощи меню: Справочники \ Менеджеры.

По нажатию на кнопку «Владельцы» - Открывается справочник владельцев. Перейти к справочнику владельцев так же можно при помощи меню: Справочники \ Владельцы.

По нажатию на кнопку «Проданные» - Открывается окно, содержащее основную информацию по проданным автомобилям. Перейти в окно проданных автомобилей так же можно при помощи меню: Справочники \ Проданные.

По нажатию на кнопку «Добавить» - Открывается окно регистрации автомобиля. Перейти к окну регистрации можно при помощи меню: Правка \ Зарегистрировать автомобиль.

По нажатию на кнопку «Редактировать» - Открывается окно редактирования записи. Перейти к окну редактирования можно при помощи меню: Правка \ Редактировать данные. ВАЖНО! В окне редактирования отображается текущая выделенная запись, для того чтобы внести корректировки в нужную запись, следует сначала выделить её.

По нажатии на кнопку «Продать» - Появляется запрос на подтверждение удаления. Если Вы выбрали «Yes», то в фоновом режиме происходит копирование основных данных в таблицу проданные, после чего запись в основной таблице удаляется. Продать автомобиль можно и через меню: Правка \ Продать автомобиль.

По нажатии на кнопку «Справка» - Открывается справочная система по программе. Так же вызвать справку можно при помощи меню: Помощь \ Справка, или же по нажатию на клавишу F1.

По нажатии на кнопку «Выход» - программа будет закрыта. Закрыть программу так же можно из меню: Файл \ Выход.

По нажатии на кнопку «Искать» - Происходит поиск по БД автомобили. Предварительно нужно ввести параметр для поиска и выбрать поле, по которому будет производиться поиск.

По нажатии на кнопку «Сортировать» - Происходит сортировка записей в таблице в алфавитном порядке (для цифровых полей – по возрастанию).

По нажатии на кнопку «Сформировать запрос» - Открывается окно формирования запросов. Так же открыть данное окно можно при помощи меню: Файл \ Создать запрос.

Справочник менеджеров. В этом окне отображены данные о персонале (фамилия, имя, отчество.

По нажатию на кнопку «Искать» - Происходит поиск по БД. Предварительно в поле необходимо ввести параметр поиска (искомое значение), затем выбрать где искать (имя, фамилия) и нажать на кнопку "Искать". Если в БД содержится запись с таким значением курсор таблицы переместится на неё, если же нет, то программа выдаст сообщение о том что нет такой записи.

По нажатию на кнопку «Редактировать» - Внизу окна появляется панель, куда автоматически переносятся данные из выделенной записи. После внесения необходимых корректировок нужно нажать на кнопку «Сохранить», в противном случае «Отмена».

По нажатию на кнопку «Новый» - Внизу окна появляется панель для ввода данных о новом сотруднике. После ввода следует нажать на кнопку «Сохранить», в противном случае «Отмена». Панель добавления нового менеджера.

Па нажатию на кнопку «Удалить» - Появляется запрос на подтверждение удаления. Если Вы выбрали «Yes» то выделенная запись будет удалена. Запрос на подтверждение удаления

Справочник владельцев. В этом окне отображены данные о владельцах автомобилей (фамилия, имя, отчество, адрес, E-mail, телефон, сотовый) которые зарегистрированных в базе данных.

По нажатию на кнопку «Искать» - Происходит поиск по БД. Предварительно в поле необходимо ввести параметр поиска (искомое значение), затем выбрать где искать (фамилия, № телефона) и нажать на кнопку "Искать". Если в БД содержится запись с таким значением курсор таблицы переместится на неё, если же нет, то программа выдаст сообщение о том что нет такой записи.

По нажатию на кнопку «Сортировать» - Происходит сортировка записей в таблице БД в алфавитном порядке (для числовых полей – в порядке возрастания). Предварительно нужно указать поле по которому будет происходить сортировка.

По нажатию на кнопку «Редактировать» - Внизу окна появляется панель, куда автоматически переносятся данные из выделенной записи. После внесения необходимых корректировок нужно нажать на кнопку «Сохранить», в противном случае «Отмена».Панель редактирования.

Справочник проданных автомобилей. В данном окне отображаются данные о проданных автомобилях (марка, модель, объём, свет, тип кузова, цена, дата продажи), которых в данный момент физически нет.

По нажатию на кнопку «Искать» - Происходит поиск по БД. Предварительно в поле необходимо ввести параметр поиска (искомое значение), затем выбрать где искать (марка, модель) и нажать на кнопку "Искать". Если в БД содержится запись с таким значением курсор таблицы переместится на неё, если же нет, то программа выдаст сообщение о том что нет такой записи.

По нажатию на кнопку «Сортировать» - Происходит сортировка записей в таблице БД в алфавитном порядке (для числовых полей – в порядке возрастания). Предварительно нужно указать поле по которому будет происходить сортировка.

По нажатию на кнопку «Отчёт» - Происходит экспорт данных из таблицы БД в MS Excel. После окончания создания отчёта откроется окно MS Excel, в котором будут помещены данные.

По нажатию на кнопку «Назад» - Закрывается справочник проданных автомобилей, происходит возврат к основному рабочему окну.

Окно регистрации автомобиля. Это окно предназначено для добавления новой записи об автомобиле и его владельце, комплектности и фотографий в соответствующие базы данных.

После заполнения всех полей и прикрепления фотографий следует нажать на кнопку «Записать», после чего если все поля заполнены корректно происходит запись в БД.

При необходимости можно отчистить все поля разом, для этого необходимо нажать на кнопку «Очистить» .

Окно редактирования записи. Это окно предназначено для редактирования выбранной (выделенной) записи в таблице базы данных.

Пользователю предлагается внести необходимые корректировки в полях и если всё готово, нажать на кнопку "Записать".

Окно поиска. Это окно предназначено для формирования запросов по различным критериям базе данных.

В данном окне пользователю предлагается ввести (выбрать) параметры запроса и нажать кнопку "Найти". По нажатию на эту кнопку происходит обработка запроса и результат его выполнения отображается в таблице.

## 

## 4.3 Инструкция программисту

Для того чтобы программа работала, на компьютере пользователя должно быть установлено ядро BDE, и желательно Windows XР. Все фотографии которыми пользуется программа должна находится во вложенной папке «FOTO». Все файлы базы данных должны находится в корневом каталоге.

Инсталляция программы не требуется.

# 5. ЭКОНОМИЧЕСКАЯ ЧАСТЬ

В экономической части дипломного проекта должна быть определена:

* себестоимость-цена,
* экономическая эффективность проекта.

Для подсчета себестоимости продукции и экономической эффективности данного программного продукта, нужно знать следующие составляющие:

* затрат на техническое и программное обеспечение проекта;
* расчет затрат на материальные ресурсы;
* расчет затрат на энергоресурсы;
* амортизационные отчисления;
* расчет фонда заработной платы;
* прочие расходы.

## 

## 5.1. Определение затрат на создание программного продукта

Трудоемкость разработки программного продукта можно определить следующим способом:

 (6.1)

где

 - затраты труда на подготовку описание задачи;

 - затраты труда на разработку алгоритма решение задачи;

 - затраты труда на разработку блок-схемы алгоритма решение задачи;

 - затраты труда на составление программы по готовой блок-схеме;

 - затраты труда на подготовку документации задачи;

 - затраты труда на отладку программы на ЭВМ при комплексной отладке задачи.

Составляющие затрат, в свою очередь можно вычислить через условное число операторов Q. В нашем случае число операторов в отлаженной программе Q=4.200 творческим характером работы, в место этого оценим затраты труда на изучение описания задачи с учетом уточнения описания и квалификации программиста определяются:

 (6.2)

где

*В*- коэффициент увеличения затрат труда вследствие недостаточного описания задачи, уточнение и некоторой доработки. (*В* = от 1,2 до 5). *В* = 1,2

*К*- коэффициент квалификации разработчика (*К* = от 0,8 до 1,5) *К* = 1,1

### 5.1.1 Трудоемкость работ на разработку программы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №№ | Этапы проектирования | В | К | Формула расчета | Трудоемкость чел. час. |
| 1 | Подготовительный | 1,2 | 1,1 |  | 39,27 |
| 2 | Алгоритм решения задачи |  |  |  | 40,91 |
| 3 | Блок-схема алгоритма |  |  |  | 40,91 |
| 4 | Составление программы по блок-схеме |  |  |  | 40,91 |
| 5 | Отладка программы на ЭВМ |  |  |  | 92,05 |
| 6 | Подготовка документации |  |  |  | 16,36  12,27 |
|  | Итого: |  |  |  | *Т*н=282,68 |

Примечание:

 (6.3)

где

 - затраты труда на подготовку материалов и рукописи.

-затраты на редактирование, печать и оформление документации.

Для разработки программы принимается один программист:

Инженер-программист 12 разряда, с трудоемкостью работ 

## 

## 5.2 Расчет себестоимости и цены программного продукта

### 5.1.2 Расчет фонда заработной платы

Работа программиста по условиям труда квалифицируется вредной. Согласно положению об оплате труда доплата за вредность составляет до 12% в зависимости от аттестации рабочего места.

Для расчета заработной платы, тарифной ставки учитывается поправочный коэффициент за условие труда 1,12. Тарифный коэффициент 2,2 разряд 12.

Дополнительный фонд заработной платы – 10%

Отчисления на социальные нужды- 19,5%   
Определение фонда заработной платы инженера-программиста 12 разряда.

Часовая тарифная ставка 2,2 разряда, с учетом доплаты за вредность, составляет:



Тарифная заработная плата за отработанное время:



Дополнительная заработная плата-10% от *З*тар



Оплата труда программиста составит:

 40050,10+ 4005 = 44055,11

Фонд оплаты труда на разработку программы составит:

 44055,11

Отчисления на социальные нужды:



Общий фонд заработной платы на разработку программного обеспечения составит:

 44055,11+8811 = 52866,13

Среднемесячная заработная плата одного специалиста составит:

 31718,18

### 5.1.3 Расчет затрат на техническое обеспечения проекта

Для разработки программного обеспечения необходимы технические средства:

* монитор - 30000 тенге
* системный блок - 95000 тенге
* клавиатура – 1500 тенге
* мышь – 1200 тенге
* принтер – 20000 тенге
* программный продукт - «Delphi» (лицензионный) - 150000 тенге
* Windows XP (лицензионный) - 10000 тенге

Итого затрат на техническое и программное обеспечение *С*оф = 307700 тенге

Основные фонды переносят свою стоимость на создаваемую продукцию в виде амортизационных отчислений.

Срок использования технических средств 3 года.

Амортизация- это денежное выражения износа основных фондов.



### 5.1.4 Прочие затраты

Затраты на оплату машинного времени при отладке программы определяются путем умножения фактического времени отладки программы на цену машино-часа арендного времени:



где

*С*час- цена машино-часа арендного времени, тен/час;

*t*эвм- фактическое время отладки программы ЭВМ;

Фактическое время отладки вычислим по формуле:

= 40,91+92,05+28,64 = 161,59

По усредненным расчетам фирм цена машино-часа арендуемого времени составляет 22,8 тг.

Общие затраты на оплату машинного времени составляют:

= 3684,27

### 5.1.5 Расчет затрат на текущий ремонт

Затраты на текущий и профилактический ремонт принимаются равными 5% от стоимости ЭВМ

/12\*1,5= 1923,13 тенге

Затраты на материалы, необходимые для обеспечения нормальной работы ПЭВМ составляют около 1% от стоимости ЭВМ:

тенге

Итого прочие затраты: Зпр = 1923,13+3684,27+384,63 = 5992 тенге

### 5.1.6 Затраты на материальные ресурсы

1. Компакт-диск - 1 \* 100 = 100 тенге
2. Вспомогательная литература 3000 тенге
3. Канцтовары – 800 тенге

Итого материальные ресурсы – 3900 тенге

### 5.1.7 Расчет затрат на энергоресурсы

Для выполнения программного проекта используется электроэнергия на технологические цели:

 тенге

*N* - мощность технических средств- 0,5квт

*n* - количество используемых технических средств.

*Ц* - цена одного киловатта электроэнергии – 4,84 тг.

*t*- время работы в часах. 35\*8 = 280

## 

## 5.3 Расчет экономической эффективности проекта

Калькуляция себестоимости и отпускной цены

|  |  |  |
| --- | --- | --- |
| № П/П | Статьи и затраты | Сумма (тенге) |
| 1 | Материалы | 3900,00 |
| 2 | Энергоресурсы | 2032,80 |
| 3 | Основная заработная плата | 44055,11 |
| 4 | Дополнительная заработная плата | 4005,00 |
| 5 | Отчисление на социальные нужды | 8811,00 |
| 6 | Амортизационные нужды | 13687,95 |
| 7 | Прочие затраты | 5992,00 |
| 8 | Расходы на подготовку и осваивание производства  0,5% от  17 | 412,42 |
| 9 | Общехозяйственные расходы 3 % от 1/7 | 2474,52 |
| 10 | Производственная себестоимость 1/9 | 85370,80 |
| 11 | Прибыль предприятия 20% от гр 10 | 17074,16 |
| 12 | Цена продукции гр 10+11 | 102444,96 |
| 13 | НДС- 14% от гр 12 | 14342,29 |
| 14 | Отпускная цена гр 12+гр 13 | 116787,25 |

Эф = (Ц-С)/К

Где Эф- экономическая эффективность.

Готовая эффективность создания системы определяется из выражения:

Эг = З/(Ен+Кр) = 57132,29/0,44 = 130445,87

Где З- изменение размера затрат на создание и обслуживание информационной системы определяется из выражения:

З =И∑ - К\*(Ен+Кр) = 57132,29

где И∑ - абсолютное изменение издержек работы с использованием программ.

Кр- коэффициент реновации, принимается как норма амортизации с учетом срока службы существующего оборудования ( срок службы системы Т сл – 3года) и определяется по формуле:

Кр = Ен/((1+Ен) тел -1),

Кр = 0,15/((1+0,15)3 – 1) = 0,29

К- единовременное капиталовложение

К = 97876,64 тенге (программные и технические затраты, фонд заработной платы, прочие

расходы, материальные затраты, затраты на энергоносители и прочие расходы)

Тогда изменение размера затрат составит:

З = 57132,29 тенге.

Точная эффективность планируется в размере

Эг = 130445,87 тенге.

Теперь можно определить срок окупаемости Ток создания новой системы из выражения:

Ток = К/Эг = 0,75

Данный срок окупаемости учитывает только материальные преимущества создания новой системы.

|  |  |  |  |
| --- | --- | --- | --- |
| В целом система считается рентабельной если ее срок окупаемости меньше года.№  П/П | Показатели | Единица  измерения | Количество |
| 1 | Общая трудоемкость работ в том числе | Чел/час | 282,68 |
| 1.1 | Техническое задание | Чел/час | 39,27 |
| 1.2 | Техно-рабочий проект | Чел/час | 122,73 |
| 1.3 | Внедрение | Чел/час | 92,05 |
| 2 | Срок разработки проекта по плану | Дни | 35,00 |
| 3 | Численность персонала | Чел. | 1,00 |
| 4 | Себестоимость продукции | Тенге | 85370,80 |
| 5 | Прибыль предприятия | Тенге | 17074,16 |
| 6 | Отпускная цена продукции | Тенге | 116787,25 |
| 7 | Среднемесячная заработная плата | Тенге | 31718,18 |
| 8 | Производительность труда | % | 100,96 |
| 9 | Срок окупаемости затрат | год | 0,75 |

## 

## 5.4 Технико-экономические показатели проекта

ПТ = Тн/Тпл\*100% = 282,68/280\*100% = 101

Тн- общая трудоемкость на программу

Тф- фактическая трудоемкость

Тф = ч\*т\*8 = 1\*35\*8 = 280

Где:

Ч - численность программистов - 1

t - плановый срок выполнения работ - 35

8 - продолжительность рабочего дня

# 6. Мероприятия по технике безопасности и окружающей среды.

## 

## 6.1. Охрана труда

Охрана труда - это система законодательных актов, социально-экономических, организационных, технических, гигиенических, и лечебно-профилактических мероприятий и средств, обеспечивающих безопасность, сохранение здоровья и работоспособности человека в процессе труда.

В практической деятельности принято рассматривать законодательные акты и социально-экономические мероприятия в понятиях трудовое законодательство и управление охраной труда; технические мероприятия и средства - техника безопасности и система противопожарной защиты; гигиенические и лечебно-профилактические мероприятия и средства - гигиена труда и производственная санитария: организационные мероприятия - в понятиях как техники безопасности и производственной санитарии, так и управления охраной труда.

Все многообразие законодательных актов, мероприятий и средств, включенных в понятие охраны труда, направлено на создание таких условий труда, при которых исключено воздействие на работающих опасных и вредных производственных факторов.

Трудовое законодательство - это совокупность действующих законов и нормативных актов, регулирующих общественно-трудовые отношения рабочих и служащих. Законодательство о труде регулирует трудовые отношения всех рабочих и служащих, содействуя росту производительности труда, повышению эффективности производства и подъему на этой основе материального и культурного уровня жизни трудящихся, укреплению трудовой дисциплины. Оно устанавливает временную охрану права на труд рабочих и служащих и неотъемлемое право на создание им высокого уровня безопасности и здоровых условий труда.

Право граждан на труд реализуется путем заключения рабочими и служащими трудового договора о роботе на соответствующем предприятии, в учреждении и организации. Трудовой вой договор есть соглашение между трудящимися и предприятием, по которому рабочий или служащий обязуется выполнять работу по определенной специальности, квалификации или должности с подчинением внутреннему распорядку.

Предприятие по этому соглашению обязуется выплачивать трудящемуся заработную плату и обеспечивать условия, предусмотренные законодательством о труде коллективным договором и соглашением сторон.

В соответствии с трудовым законодательством на всех предприятиях, в учреждениях и организациях должны быть созданы здоровые и безопасные условия труда. Обеспечение таких условий возлагается на администрацию, которая обязана внедрять современные средства техники безопасности, предупреждающие травматизм и создавать санитарно - гигиенические условия, предотвращающие возникновение профессиональных заболеваний.

Важнейшим условием обеспечения безопасности труда является обучение, инструктаж и проверка знаний по охране труда. Лица, принимаемые на работу или переводимые на другую, проходят следующие виды, инструктажа: вводный, первичный на рабочем месте, повторный, внеплановый и текущий. Должностные лица, виновные в нарушении трудового законодательства, правил по технике безопасности и производственной санитарии, в невыполнении обязанностей по коллективным договорам и соглашениям по охране труда или в воспрепятствии деятельности профсоюзов, несут за это ответственность. В зависимости от степени вины они могут быть привлечены к дисциплинарной, административной, материальной и уголовной ответственности.

### Требования к рабочему месту программиста

Столы при этом должны быть достаточно больше - от 1200х800 мм до 1600х800 мм. На таком столе вы сможете разместить всю свою технику. Именно на столе, а не под столом как мы обычно это делаем, потому что под столом должно быть достаточно просторно ногам да и лазать под стол , чтобы вставить в дисковод дискету, тоже весьма утомительно. Сейчас модно иметь под столом подставку с наклоном от 5 до 15 градусов для ног, которая обеспечивает комфорт и покой ногам во время длительного сидения за компьютером. Стул при этом необходимо иметь на колёсиках, высота сидения от пола 42-53 см, высота стола – 72см. Усаживаться за компьютер нужно так, чтобы угол между туловищем и бёдрами, туловищем и предплечьем равнялся 90 градусов. Монитор должен стоять от глаз на расстоянии не менее 45см, притом желательно наклонить его так, чтобы вы глядели с монитором не «глаза в глаза», а взирали на него несколько сверху под углом от 5 до 35 градусов к поверхности экрана. Угол наклона клавиатуры к столу рекомендуется устанавливать в пределах 5-10 градусов.

## 

## 6.2. Техника безопасности

Техника безопасности - система организационных мероприятий и технических средств, предотвращающих воздействие на работу в опасных производственных зонах или факторах.

Существуют 5 видов инструктажей по технике безопасности:

* вводный инструктаж (для всех независимо от образования или стажа работы);
* первичный - на рабочем месте (в день приема на работу);
* повторный (не реже одного раза в 6 месяцев);
* неплановый (при изменении правил охраны труда);
* текущий (проводят с работниками перед производством работ, на которых оформляется наряд-допуск).

Во время работы на ПЭВМ в помещении повышается температура и снижается относительная влажность воздуха, ухудшается ионный и качественный состав воздуха; увеличивается содержание в воздухе органических веществ и двуокиси углерода. Содержание в воздухе указанных веществ может в несколько раз предельно превышать допустимой величины. По этой причине необходимо поддерживать следующие оптимальные параметры микроклимата: температура воздуха от 18 до 21° C; относительная влажность воздуха соответственно 55-62%; скорость движения воздуха менее 0,1м/с. Также должно осуществляться проветривание помещения, в зависимости от погодных условий, длительность должна быть не менее 10 мин. Наилучший обмен воздуха осуществляется при сквозном проветривании, Другой путь обеспечения воздухообмена, может, быть, достигнут установлением в оконных проемах автономных кондиционеров. Звукоизоляция ограждающих конструкций КВТ должна также отвечать определенным требованиям. Для снижения уровня шума потолок или стены выше 1.5 - 1.7 метра от пола должны облицовываться звукопоглощающим материалом с максимальным коэффициентом звукопоглощения в области частот 63-8000 Гц. Дополнительным звукопоглощением в КВТ могут быть занавески, подвешенные в складку на расстоянии 15-20 см. от ограждения, выполненные из плотной, тяжелой ткани. Для уменьшенного поглощения света потолок и стены выше панелей (3,5 - 1.7м.), если они не облицованы звукопоглощающим материалом, окрашиваются белой водоэмульсионной краской (коэффициент отражения должен быть не менее 0,7). Для окраски стены панелей рекомендуется отдавать предпочтение светлым краскам.

В осветительных установках (ОУ) КВТ следует использовать систему общего освещения, выполненную потолочным или подвесным люминесцентными светильниками, равномерно размещенными по потолку рядами параллельно светопроемам так, чтобы экран видеомонитора находился в зоне защитного угла светильника, и его проекции не приходились на экран.

Работающие на ПЭВМ не должны видеть отражение светильников на экране. Применять местное освещение при работе на ПЭВМ не рекомендуется.

Работа на ПЭВМ может осуществляться при следующих видах освещения:

Общим люминесцентном освещении, когда видеомониторы располагаются по периметру помещения или при центральном расположении рабочих мест в два ряда по длине класса с экранами, обращенные в противоположные стороны;

Совмещенном освещении (естественное + искусственное) только при одном и трех рядном расположении рабочих мест, когда экран и поверхность рабочего стола находятся перпендикулярно светонесущей стене.

Естественное освещение, когда рабочие места с ПЭВМ располагаются в один ряд по длине помещения на расстоянии 0,8 - 1,0м от стены с оконными проемами, и экраны находятся перпендикулярно этой стены. Основной поток естественного света при этой должен быть слева. Не допускается направление основного светового потока естественного света справа, сзади и спереди работающего на ПЭВМ. Оптимальное расстояние глаз до экрана видеомонитора должно составлять 60-70 см, допустимое не менее 50 см. Рассматривать информацию ближе 50 см не рекомендуется.

Рекомендации по использованию вычислительной техники:

-необходимо соблюдать ограничения на работу с персональными компьютерами для служащих, страдающих заболеваниями опорно-двигательного аппарата, глаз, кожи, а также для беременных женщин;

-предпочтительнее использовать дисплеи с высокой разрешающей способностью и размером экрана не менее 14" (Hi-Resolution, Non-Interlaced);

-лучше выбирать видеоадаптеры с высоким разрешением и частотой кадровой развертки не менее 70~72Гц;

-обязательно ставить на дисплеи экранные фильтры с антистатическим покрытием, в несколько раз снижающие утомляемость глаз и концентрацию пылевых частиц в близи экрана монитора;

-сидеть не ближе 70 см от дисплея:

- экран дисплея должен быть ориентирован таким образом, чтобы исключить блики от источников света;

-не следует располагать дисплей непосредственно под источником освещения или вплотную с ним;

-желательно, чтобы освещенность рабочего места оператора не превышала 2/3 нормальной освещенности помещения;

-стена позади дисплея должна быть освещена примерно так же, как его экран.

-при размещении в одной комнате нескольких персональных компьютеров расстояние от рабочего места каждого оператора до задних и боковых стенок соседних персональных компьютеров должно составлять не менее 1.2м;

- рабочее место должно быть оборудовано так, чтобы исключить неудобные позы и длительные статические напряжения тела;

-общее время работы с дисплеем не должно превышать 50% всего рабочего времени оператора;

- не следует превышать темп работы порядка 10 тысяч нажатий клавиш в час (примерно 1500 слов);

- при обычной рабою с компьютером необходимо делать 15-минутные перерывы через каждые 2 часа, а при интенсивной работе - через каждый час.

Помимо безопасности пользователя, необходимо сказать несколько слов и о безопасности компьютера и, что особенно важно, безопасности данных, хранящихся в нем. ПК, используемые для хранения особо важной информации необходимо оборудовать устройствами бесперебойного питания, поддерживающими питающее напряжение в течение некоторого времени при аварийных ситуациях в электрической сети. Нельзя загораживать заднюю стенку системного блока или ставить персональный компьютер вплотную к стене - это приводит к «тяжелому» режиму охлаждения системного блока и его перегреву. То же самое относится к дисплею - нельзя класть на него бумаги, книги и вообще все, что может закрыть его вентиляционные отверстия. Пыль и электроника плохо совместимы друг с другом, поэтому необходимо поддерживать в помещении приемлемый пылевой режим.

При появлении запаха гари немедленно прекратить работу, выключить аппаратуру.

Перед началом работы убедитесь в отсутствии видимых повреждений рабочего места, сядьте так, чтобы линия взора приходилась в центр экрана, чтобы, не наклоняясь пользоваться клавиатурой и воспринимать переданную на монитор информацию.

Во время работы выполняйте указанные выше правила, следите за исправностью аппаратуры и немедленно прекратите работу при появлении постороннего звука или самопроизвольного отключения аппаратуры.

### Освещенность КВТ в учебных аудиториях

Освещенность КВТ также имеет большое значение при работе на ПЭВМ. Она во многом определяется цветовой и сетевой обстановкой. Для уменьшенного поглощения света потолок и стены выше панелей (1,5-1,7м.), если они не облицованы звукопоглощающим материалом, окрашиваются белой водоэмульсионной краской (коэф. отражения должен быть не менее 0,7). Для окраски стены панелей рекомендуется отдавать предпочтение светлым краскам. В осветительных установках (ОУ)КВТ следует использовать систему общего освещения, вы полненную потолочным или подвесным люминесцентными светильниками, равномерно размещенными по потолку рядами параллельно светопроемам так, чтобы экран видеомонитора находился в зоне защитного угла светильника и его проекции не приходилась на экран.

Работающие на ПЭВМ не должны видеть отражение светильников на экране. Применять местное освещение при работе на ПЭВМ не рекомендуется.

Основной поток естественного света при этой должен быть слева. Не допускается направление основного светового потока естественного света справа, сзади и спереди работающего на ПЭВМ.

Работа на ПЭВМ во время занятий или при производственной практике может осуществляться при следующих видах освещения:

общим люминесцентном освещение – видеомониторы располагаются по периметру помещения или при центральном расположении рабочих мест в два ряда по длине класса с экранами, обращенными в противоположные стороны;

совмещенном освещение, (естественное + искусственное) только при одном или трех рядном расположении рабочих мест, когда экран и поверхность рабочего стола находятся перпендикулярно светонесущей стене;

естественное освещение, когда рабочие места с ПЭВМ располагаются в один ряд по длине класса на расстоянии 0,8-1,0м от стены с оконными проемами, и экраны находятся перпендикулярно этой стены. Основной поток естественного света при этом освещение должен быть слева.

Оптимальное расстояние глаз до экрана видеомонитора должно

составлять 60-70 см, допустимое не менее 50 см. Рассматривать информацию ближе 50 см не рекомендуется.

### Пожарная безопасность при работе на персональном компьютере.

Пожары в вычислительном центре представляют особую опасность, так как поряжены с большими материальными потерями. Характерная особенность небольшие площади помещений. Как известно, пожар может возникнуть при взаимодействии веществ, окислителя, и источников зажигания. Горючими компонентами являются: строительные материалы для акустической и эстетической отделки помещений, перегородки, полы, двери, изоляция силовых, сигнальных кабелей и т.д. Для отвода теплоты от ЭВМ действует мощная система кондиционирования. Поэтому кислород, как  
окислитель процессов горения, имеется в любой точке помещений ВЦ.

Особенностью компьютеров является очень высокая плотность расположения микросхем. При прохождении электрического тока по проводникам и деталям выделяется тепло, что в условиях их высокой плотности может привести к перегреву. Надежная работа отдельных элементов и микросхем в целом обеспечивается только в определенных интервалах температуры, влажности и при заданных электрических параметрах. При отклонении реальных условий эксплуатации от расчетных могут возникнуть пожароопасные ситуации.

Кабельные линии являются наиболее пожароопасным местом. Наличие горючего изоляционного материала, вероятных источников зажигания в виде электрических искр и дуг, разветвленность и недоступность делают кабельные линии местом наиболее вероятного возникновения и развития пожара. Для понижения воспламеняемости и способности распространять пламя кабели покрывают огнезащитными покрытиями.

Пожар - это неконтролируемое горение вне специального очага, который наносит огромный ущерб. Главная причина пожара неэлектрического характера является неосторожное обращение с огнем, а также взрывы газо-воздушных и паро-воздушных смесей. Электрическое горение - это замыкание, перегрузка электрического тока на электрическом оборудовании, грозовая молния.

Устранение причин пожара в электрических оборудования проводится в различных направлениях:

предупреждение замыкания осуществляется правильным выбором, монтажом эксплуатацией сетей;

применение защиты схем в виде быстродействующих реле, а также выключателей, плавких предохранителей, автоматических выключателей.

Важное внимание следует обратить на пожарную безопасность предприятия в целом и отдельных его помещений. В помещениях не должен скапливаться мусор, ненужные бумаги, хлам и др. вещи, не используемые в производственном процессе. Необходимо предусмотреть аварийный выход из пределов помещения в случае пожара. В помещении должны быть предусмотрены огнетушители. Они должны быть в рабочем состоянии и проверяться согласно нормам. В помещениях должна быть пожарная сигнализация. В случае возникновения пожара необходимо сообщить в ближайшую пожарную часть и по возможности предпринять некоторые шага по его устранению.

### Электробезопасность

Так как работа программиста прямо связана с электрооборудованием, необходимо уметь пользоваться им правильно и соблюдать меры безопасности от поражения электрического тока.

Существует множество мер от повреждения электрическим током. Одно из них защитное заземление. Защитное заземление - преднамеренно электрическое заземление с землей или ее эквивалентом металлических нетоковедущих частей, которые могут оказаться под напряжением вследствие замыкания на корпус и по другим причинам.

Назначение защитного заземления - устранение опасности поражением током в случае прикосновения к корпусу и другим токоведущим частям электроустановки,

оказавшимся под напряжением. Принцип действия защитного заземления - снижение до безопасных значений напряжений прикосновения и шага, обусловленных замыканием на корпус и др. причинами. Это достигается путем, уменьшения потенциала заземленного оборудования (за счет подъема потенциала основания, на котором стоит человек, до значения, близкого к значению потенциала заземленного оборудования).

Также есть и такая мера защиты от поражения электрического тока, называемая занулением. Опасность поражения током при прикосновении к корпусу и другим нетоковедущим металлическим частям электрооборудования, оказавшимся под напряжением вследствие замыкания на корпусе и по др. причинам, может быть устранена быстрым отключением поврежденной электроустановки от питающей сети и вместе с тем снижением напряжения корпуса относительно земли. Этой цели и служит зануление. Зануление - преднамеренное электрическое соединение с нулевым защитным проводником металлических нетоковедущих частей, которые могут оказаться под напряжением.

Нулевым защитным проводником называется проводник, соединяющий зануляемые части с глухо-заземленной нейтральной точкой источника тока или ее эквивалентом.

Принцип действия зануления - превращение замыкания с целью вызвать большой ток, способный обеспечить срабатывание защиты и тем самым автоматически отключить поврежденную электроустановку от питающей сети.

Также существует такое средство защиты как защитное отключение. Защитное отключение - быстродействующая защита, обеспечивающая отключение электроустановки при возникновении в ней опасности поражения человека током. Такая опасность может возникнуть, в частности, при замыкании фазы на корпус, снижения изоляции сети ниже определенного предела и, наконец, в случае прикосновения человека непосредственно к токоведущей части, находящейся под напряжением. Основными элементами устройства защитного отключения является прибор защитного отключения и автоматический выключатель.

Прибор защитного отключения - совокупность отдельных элементов, которые воспринимают входную величину, реагирует на ее изменения и при заданном значении дают сигнал на ее отключение выключателя. Этими элементами являются:

датчик - входное звено устройства, воспринимающие воздействия из вне и осуществляющее преобразование этого воздействия в соответствующий сигнал;

усилитель, предназначенный для усиления сигнала датчика, если он оказывается недостаточно мощным;

цепи контроля, служащие периодической проверки исправности защитного отключения;

вспомогательные элементы - сигнальные лампы и измерительные приборы, характеризующие состояние электроустановки.

Автоматический выключатель - аппарат, предназначенный для включения и отключения от цепей под нагрузкой и при коротких замыканиях. Он должен включать цепь автоматически при поступлении сигнала от прибора защитного отключения.

Также есть различные электрические защитные средства от поражения током. Защитные средства могут быть условно разделены на три группы: изолирующие, ограждающие и предохранительные.

Изолирующие - изолируют человека от токоведущих или заземленных частей, а также от земли. Они делятся на основные и дополнительные.

Основные - обладают изоляцией, способной длительно выдерживать рабочее напряжение электроустановки и поэтому ими разрешается касаться токоведущих частей, находящихся под напряжением. К ним относятся: в электроустановках до 1000 Вт – диэлектрические перчатки, изолирующие штанги, изолирующие и электро измерительные клещи и т.д.; свыше 1000Вт - изолирующие штанги, и электроизмерительные клещи, а также средства для ремонтных работ под напряжением свыше 1000Вт.

Дополнительные - обладают изоляцией неспособной выдержать рабочее напряжение электроустановки, и поэтому они не могут самостоятельно защищать человека от поражения током под этим напряжением. Их значение – усилить защитные действия основных и изолирующих средств, вместе с которыми они должны применяться, при чем при использовании основных защитных средств достаточно применения одного дополнительного защитного средства. К дополнительным относятся средства в электроустановках до 1000Вт - диэлектрические галоши н коврики, а также изолирующие подставки.

## 6.3 Охрана окружающей среды

Охрана окружающей среды - комплекс научно-технических, производственно-хозяйственных и административно-правовых мероприятии, направленных на сохранение и надлежащее использование земли и ее недр, водных ресурсов, растительного и животного мира, обеспечение чистоты воздуха и воды, воспроизводства природных богатств, гармоничные взаимоотношения между обществом и природой. Не сложно представить себе процветающую экономику, при которой люди из года в год становятся все более больными в результате неправильного подхода к своему здоровью и загрязненной окружающей среды. По мере того, как мы строим свое общество, необходимо применять нарастающие усилия в том, чтобы наши граждане были здоровыми на протяжении всей своей жизни, и их окружала здоровая природная среда.

Источником загрязнения атмосферного воздуха называется технологический агрегат, выделяющий в процессе эксплуатации вредные вещества.

Количество выделенных веществ зависит от типа и мощности производства, его технической оснащенности и определяется путем инструментальных замеров или расчетов с использованием специальных отраслевых методик.

Основными мероприятиями по снижению выбросов вредных веществ в, атмосферу являются совершенствование технологических процессов, включая снижение производственных выбросов; строительство новых и повышение эффективности существующих очистных устройств; ликвидация источников загрязнения, перепрофилирование производства.

Применительно к предприятиям машиностроительного производства наиболее значимым представляется газа - и пылеулавливание вентиляционных выбросов, особенно при открытой разливке металлов.

Очистку и обезвреживание газовых составляющих выбросов промышленных производств осуществляют методами, выбор которых определяется составом, концентрации загрязняющих веществ, типа производства, условиями выброса.

Очистку вентиляционных выбросов от механических примесей осуществляют аппаратами, мокрого и сухого пылеулавливания, волокнистыми фильтрами и электрофильтрами.

В качестве фильтров используют различные фильтрующие, тонкие грубо волокнистые материалы. Кроме того, на предприятиях машиностроения широкое применение получили электрофильтры, которые в зависимости от способа удаления осажденным на электродах частиц подразделяются на сухие и мокрые.

Защита вод питьевого назначения также имеет глобальное значение.

Степень очистки сточных вод устанавливается в зависимости от местных условий с учетом возможного использования очищенных сточных вод - для промышленных и сельскохозяйственных нужд.

На предприятиях металлургической и машиностроительной промышленности очистка сточных вод осуществляется, как правило, в отстойниках, шлаконакопителях, нефте-маслоловушках с использованием ряде случаев коагулянтов. Полученный шлак, содержащий большое количество металлов, утилизируется и включается в состав шихты. Очищенные воды в большинстве случаев используются в системах оборотного водоснабжения. При этом вода основного источника или из других циклов водопользования идет на компенсацию потерь оборотной воды.

Важным направлением охраны окружающей среды является охрана почв от водной и ветровой эрозии, борьба с их засолением путем введения соответствующих севооборотов, создания лесозащитных полос, закрепления и облесения оврагов и балок, использования средств мелиорации.

В целях снижения загрязнения почв различными промышленными отходами в практике охраны земельных ресурсов предусматриваются следующие мероприятия: утилизация, обезвреживание методом сжигания, захоронение на специальных полигонах, организация усовершенствованных свалок.

Выбор метода обезвреживания и утилизации отходов зависит от их химического состава и степени влияния на окружающую среду. В ряде случаев отходы машиностроительных и металлургических производств содержат значительное количество химических соединений, которые могут представлять ценность как сырье и использоваться в виде дополнения к шахте.

# 7. заключение

В результате дипломного проектирования была разработана программа «АРМ менеджера по продажам в автосалоне «A-Motors», предназначенная для для регистрации и ведения учета продаж автомобилей.

Программа предназначена для того, чтобы облегчить рутинный труд менеджера автосалона, тратящего ежедневно массу времени на бумажную работу.

У данного приложения удобный интерфейс, рассчитанный на не подготовительного пользователя.

Рекомендуется использовать программу на более мощных персональных компьютерах, согласно требованиям к аппаратному программному обеспечению.

# 8. Список использованных источников

1. Епанешников А.М., Епанешников В.А. «Delphi 4 Среда разработки». Учебное пособие М: Диалог – Мифи, 1999г.- 304с.
2. Культин Н. “Программирование на Object Pascal» СПБ: БХВ – Санкт-Петербург, 1999г. 480с.
3. Фаронов В.В., Шумаков П.В. «DELPHI 4 Руководство разработки баз данных”, М: Нолидж, 1999г., 560с.
4. Шумаков П.В. «DELPHI 3 и разработка приложений баз данных” М: Нолидж, 1998г., 704с
5. Драхведидзе М.Г., Марков Е.П. . «DELPHI 4 среда визуального программирования” СП8:BHV –Санкт – Петербург, 1999г.,352с.
6. Фаронов В.В. Delphi 4. Учебный курс. - Москва, 1998
7. Валерий Фаронов Профессиональная работа в Delphi 6 Москва 2002
8. Том Сван Delphi 4 Библия разработчика Москва 1998
9. Деви Хакер Delphi 5 Руководство разработчика. (электронный вариант в формате pdf) 16. Delphi 6 Учебный курс (электронный вариант в формате CHM)
10. Иллюстрированный самоучитель по Delphi 6 (электронный вариант в формате HTML) 18. Delphi 6 и технология COM Москва 2002
11. БорисовЕ.Ф..“Основы экономической теории”,
12. Сергеев И.В.“Экономика предприятия”
13. Райзберг Б.А., Лозовский Л.Ш., Стародубцев Е.Б. “Современный экономический словарь”
14. Скала В.И. “Охрана труда и техника безопасности”, – Алматы: «LEM», 2002. –276с
15. “Охрана труда ” -М.;1989

# Приложение а. основные модули приложения

## А.1 Модуль формы окна «О программе»

unit about;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, RxGIF, ExtCtrls;

type

TFAbout = class(TForm)

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

procedure FormClose(Sender: TObject; var Action: TCloseAction);

private

{ Private declarations }

public

{ Public declarations }

end;

var

FAbout: TFAbout;

implementation

{$R \*.dfm}

procedure TFAbout.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Action:=caFree;

end;

end.

## А.2 Модуль формы окна «Зарегистрировать автомобиль»

unit add;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Buttons, glLabel, DBTables,ExtCtrls, glBevel, StdCtrls, RxGIF,

DBCtrls, LMDControl, LMDBaseControl, LMDBaseGraphicButton,

LMDCustomSpeedButton, LMDSpeedButton, ExtDlgs,IDGlobal, ToolEdit, Mask,

LMDCustomControl, LMDCustomPanel, LMDCustomBevelPanel,

LMDBaseEdit, LMDCustomEdit, LMDCustomBrowseEdit, LMDCustomFileEdit,

LMDFileOpenEdit, jpeg;

type

TFadd = class(TForm)

Panel1: TPanel;

glLabel1: TglLabel;

### Продолжение приложения А

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

Edit6: TEdit;

Edit7: TEdit;

Panel2: TPanel;

Panel3: TPanel;

glLabel2: TglLabel;

glLabel3: TglLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

Edit8: TEdit;

Edit9: TEdit;

Edit10: TEdit;

Edit11: TEdit;

ComboBox1: TComboBox;

Edit12: TEdit;

CheckBox1: TCheckBox;

CheckBox2: TCheckBox;

CheckBox3: TCheckBox;

CheckBox4: TCheckBox;

CheckBox5: TCheckBox;

CheckBox6: TCheckBox;

CheckBox7: TCheckBox;

CheckBox8: TCheckBox;

Edit13: TEdit;

ComboBox2: TComboBox;

Label15: TLabel;

ComboBox3: TComboBox;

ComboBox4: TComboBox;

Label16: TLabel;

Label17: TLabel;

Memo1: TMemo;

Label18: TLabel;

Label19: TLabel;

Memo2: TMemo;

Image2: TImage;

Panel4: TPanel;

Panel5: TPanel;

LMDSpeedButton1: TLMDSpeedButton;

LMDSpeedButton2: TLMDSpeedButton;

LMDSpeedButton3: TLMDSpeedButton;

Label20: TLabel;

DBLookupComboBox1: TDBLookupComboBox;

Image3: TImage;

### Продолжение приложения А

glLabel4: TglLabel;

LMDSpeedButton4: TLMDSpeedButton;

LMDSpeedButton5: TLMDSpeedButton;

OpenPictureDialog2: TOpenPictureDialog;

Image4: TImage;

Image5: TImage;

glLabel5: TglLabel;

Label21: TLabel;

Label22: TLabel;

Edit14: TEdit;

DateEdit1: TDateEdit;

OpenPictureDialog1: TOpenPictureDialog;

Image1: TImage;

DBLookupComboBox2: TDBLookupComboBox;

procedure LMDSpeedButton4Click(Sender: TObject);

procedure LMDSpeedButton5Click(Sender: TObject);

procedure LMDSpeedButton2Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure LMDSpeedButton3Click(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure LMDSpeedButton1Click(Sender: TObject);

procedure OpenPictureDialog1CanClose(Sender: TObject;

var CanClose: Boolean);

procedure Edit6KeyPress(Sender: TObject; var Key: Char);

procedure Edit7KeyPress(Sender: TObject; var Key: Char);

procedure Edit13KeyPress(Sender: TObject; var Key: Char);

procedure Edit14KeyPress(Sender: TObject; var Key: Char);

procedure DBLookupComboBox2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Fadd: TFadd;

pic1,pic2:string;

implementation

uses dm;

{$R \*.dfm}

function gen\_id(table:TTable; var name:string):integer;

var n:integer;

begin

table.First;

n:=table.FieldByName(name).asinteger;

while not table.Eof do begin

if table.FieldByName(name).Value>n then

n:=table.FieldByName(name).Value;

table.Next;

end;

gen\_id:=n+1;

end;

procedure TFadd.LMDSpeedButton4Click(Sender: TObject);

begin

OpenPictureDialog1.Execute;

### Продолжение приложения А

pic1:=OpenPictureDialog1.FileName;

if pic1<>'' then Image4.Picture.LoadFromFile(pic1)

else begin

MessageBox(Handle,'Фотография кузова не выбрана!','',MB\_OK);

exit;

end;

end;

procedure TFadd.LMDSpeedButton5Click(Sender: TObject);

begin

OpenPictureDialog1.Execute;

pic2:=OpenPictureDialog1.FileName;

if pic2<>'' then Image5.Picture.LoadFromFile(pic2)

else begin

MessageBox(Handle,'Фотография салона не выбрана!','',MB\_OK);

exit;

end;

end;

procedure TFadd.LMDSpeedButton2Click(Sender: TObject);

begin

edit1.Text:='';

edit2.Text:='';

edit3.Text:='';

edit4.Text:='';

edit5.Text:='';

edit6.Text:='';

edit7.Text:='';

edit8.Text:='';

edit9.Text:='';

edit10.Text:='';

edit11.Text:='';

edit12.Text:='';

edit13.Text:='';

edit14.Text:='';

CheckBox1.Checked:=false;

CheckBox2.Checked:=false;

CheckBox3.Checked:=false;

CheckBox4.Checked:=false;

CheckBox5.Checked:=false;

CheckBox6.Checked:=false;

CheckBox7.Checked:=false;

CheckBox8.Checked:=false;

Memo1.Clear;

Memo2.Clear;

end;

procedure TFadd.FormCreate(Sender: TObject);

begin

// dm1.vlad.Close;

// dm1.vlad.Open;

Memo1.Clear;

Memo2.Clear;

end;

procedure TFadd.LMDSpeedButton3Click(Sender: TObject);

begin

close;

end;

### Продолжение приложения А

procedure TFadd.FormClose(Sender: TObject; var Action: TCloseAction);

begin

edit1.Text:='';

edit2.Text:='';

edit3.Text:='';

edit4.Text:='';

edit5.Text:='';

edit6.Text:='';

edit7.Text:='';

edit8.Text:='';

edit9.Text:='';

edit10.Text:='';

edit11.Text:='';

edit12.Text:='';

edit13.Text:='';

edit14.Text:='';

CheckBox1.Checked:=false;

CheckBox2.Checked:=false;

CheckBox3.Checked:=false;

CheckBox4.Checked:=false;

CheckBox5.Checked:=false;

CheckBox6.Checked:=false;

CheckBox7.Checked:=false;

CheckBox8.Checked:=false;

Memo1.Clear;

Memo2.Clear;

end;

procedure TFadd.LMDSpeedButton1Click(Sender: TObject);

var k1,k2,k3:integer;

st1,q1,st2,q2,v,a,a\_i:string;

i,u:integer;

begin

if (edit1.Text='') or (edit2.Text='') or (edit3.Text='') or (edit4.Text='')

or (DBLookupComboBox2.KeyValue=Null) or (edit9.Text='') or (edit10.Text='') or (edit11.Text='')

or (edit12.Text='') or (edit13.Text='') or (edit14.Text='') or (pic1='')

or (pic2='') then begin

MessageBox(Handle,'Возможно некоторые поля не заполнены!','Внимание',MB\_OK);

exit;

end

else begin

dm1.komplect.Append;

dm1.avto\_in.Append;

v:='KOD\_vlad';

k1:=gen\_id(dm1.vlad,v);

dm1.vlad.Append;

dm1.vlad.FieldByName('KOD\_vlad').Value:=k1;

dm1.vlad.FieldByName('FAM').Value:=edit1.text;

dm1.vlad.FieldByName('NAME').value:=edit2.Text;

dm1.vlad.FieldByName('OTH').value:=edit3.text;

dm1.vlad.FieldByName('adres').value:=edit4.text;

dm1.vlad.fieldByName('mail').value:=edit5.Text;

dm1.vlad.FieldByName('tel').value:=edit6.text;

dm1.vlad.FieldByName('sot').value:=edit7.text;

### Продолжение приложения А

a:='KOD\_avto';

k2:=gen\_id(dm1.avto,a);

dm1.avto.Append;

dm1.avto.FieldByName('KOD\_avto').Value:=k2;

dm1.avto.FieldByName('KOD\_vlad').Value:=k1;

dm1.avto.FieldByName('marka').Value:=DBLookupComboBox2.KeyValue;

dm1.avto.FieldByName('model').value:=edit9.Text ;

dm1.avto.FieldByName('cvet').value:=edit11.text;

dm1.avto.FieldByName('gos\_nom').value:=edit10.text ;

dm1.avto.fieldByName('tip\_kuz').value:=ComboBox1.Text;

dm1.avto.FieldByName('v').value:=edit12.text;

dm1.avto.FieldByName('cena').value:=edit13.text;

st1:='';

q1:='';

for i:=length(pic1) downto 1 do

if pic1[i]<>'.' then insert(pic1[i],q1,0) else break;

st1:=ExtractFilePath(Application.Exename);

ForceDirectories(st1+'FOTO');

if dm1.avto.FieldByName('foto\_1').AsString<>'' then

DeleteFile(st1+'FOTO\'+dm1.avto.FieldByName('foto\_1').AsString);

if copyfileto(pchar(pic1),pchar(st1+'FOTO\'+inttostr(k2)+'kuz' + '.'+q1)) then

else begin

MessageBox(Handle,'Невозможно скопировать файл!','Внимание',MB\_OK);

exit;

end;

dm1.avto.FieldByName('foto\_1').AsString:=inttostr(k2)+'kuz' + '.'+q1;

st2:='';

q2:='';

for u:=length(pic2) downto 1 do

if pic2[u]<>'.' then insert(pic2[u],q2,0) else break;

st2:=ExtractFilePath(Application.Exename);

ForceDirectories(st2+'FOTO');

if dm1.avto.FieldByName('foto\_2').AsString<>'' then

DeleteFile(st2+'FOTO\'+dm1.avto.FieldByName('foto\_2').AsString);

if copyfileto(pchar(pic2),pchar(st2+'FOTO\'+inttostr(k2)+'sal' + '.'+q2)) then

else begin

MessageBox(Handle,'Невозможно скопировать файл!','Внимание',MB\_OK);

exit;

end;

dm1.avto.FieldByName('foto\_2').AsString:=inttostr(k2)+'sal' + '.'+q2;

dm1.komplect.FieldByName('KOD\_avto').Value:=k2;

if CheckBox1.Checked=true then dm1.komplect.FieldByName('zapaska').Value:='true'

else dm1.komplect.FieldByName('zapaska').Value:='false';

if CheckBox2.Checked=true then dm1.komplect.FieldByName('instrument').Value:='true'

else dm1.komplect.FieldByName('instrument').Value:='false';

### Продолжение приложения А

if CheckBox3.Checked=true then dm1.komplect.FieldByName('music').Value:='true'

else dm1.komplect.FieldByName('music').Value:='false';

if CheckBox4.Checked=true then dm1.komplect.FieldByName('medic').Value:='true'

else dm1.komplect.FieldByName('medic').Value:='false';

if CheckBox5.Checked=true then dm1.komplect.FieldByName('elect\_pak').Value:='true'

else dm1.komplect.FieldByName('elect\_pak').Value:='false';

if CheckBox6.Checked=true then dm1.komplect.FieldByName('abs').Value:='true'

else dm1.komplect.FieldByName('abs').Value:='false';

if CheckBox7.Checked=true then dm1.komplect.FieldByName('gur').Value:='true'

else dm1.komplect.FieldByName('gur').Value:='false';

if CheckBox8.Checked=true then dm1.komplect.FieldByName('kondic').Value:='true'

else dm1.komplect.FieldByName('kondic').Value:='false';

dm1.komplect.FieldByName('koleso').Value:=ComboBox2.Text;

dm1.komplect.FieldByName('korobka').Value:=ComboBox3.Text;

dm1.komplect.FieldByName('privod').Value:=ComboBox4.Text;

dm1.komplect.FieldByName('info').Value:=Memo1.Text;

dm1.komplect.FieldByName('defekt').Value:=Memo2.Text;

a\_i:='KOD\_avto';

k3:=gen\_id(dm1.avto,a\_i);

dm1.avto\_in.FieldByName('KOD\_in').Value:=k3;

dm1.avto\_in.FieldByName('akt\_nom').Value:=Edit14.Text;

dm1.avto\_in.FieldByName('KOD\_avto').Value:=k2;

dm1.avto\_in.FieldByName('KOD\_vlad').Value:=k1;

dm1.avto\_in.FieldByName('KOD\_sot').Value:=DBLookupComboBox1.KeyValue;

dm1.avto\_in.FieldByName('data').Value:=DateEdit1.Date;

// dm1.vlad.Post;

// dm1.avto.Post;

dm1.komplect.Post;

dm1.avto\_in.Post;

edit1.Text:='';

edit2.Text:='';

edit3.Text:='';

edit4.Text:='';

edit5.Text:='';

edit6.Text:='';

edit7.Text:='';

edit8.Text:='';

edit9.Text:='';

edit10.Text:='';

edit11.Text:='';

edit12.Text:='';

edit13.Text:='';

### Продолжение приложения А

edit14.Text:='';

CheckBox1.Checked:=false;

CheckBox2.Checked:=false;

CheckBox3.Checked:=false;

CheckBox4.Checked:=false;

CheckBox5.Checked:=false;

CheckBox6.Checked:=false;

CheckBox7.Checked:=false;

CheckBox8.Checked:=false;

Memo1.Clear;

Memo2.Clear;

end;

end;

procedure TFadd.OpenPictureDialog1CanClose(Sender: TObject;

var CanClose: Boolean);

begin

MessageBox(Handle,'Вы не выбрали машину!','Внимание',MB\_OK);

end;

procedure TFadd.Edit6KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'0'..'9',#8: key:=key

else key:=#0;

end;

end;

procedure TFadd.Edit7KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'0'..'9',#8: key:=key

else key:=#0;

end;

end;

procedure TFadd.Edit13KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'0'..'9',#8: key:=key

else key:=#0;

end;

end;

procedure TFadd.Edit14KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'0'..'9',#8: key:=key

else key:=#0;

end;

end;

procedure TFadd.DBLookupComboBox2Click(Sender: TObject);

begin

// Edit8.Text:=DBLookupComboBox2.KeyValue;

end;

### Продолжение приложения А

end.

## А.3 Модуль формы «Удалить автомобиль»

unit del;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Grids, DBGrids, ExtCtrls, StdCtrls, LMDControl, LMDBaseControl,

LMDBaseGraphicButton,ActiveX,ComObj, ComCtrls, LMDCustomSpeedButton, LMDSpeedButton,DB;

type

TFdel = class(TForm)

Panel1: TPanel;

DBGrid1: TDBGrid;

Panel5: TPanel;

Panel2: TPanel;

Image1: TImage;

LMDSpeedButton1: TLMDSpeedButton;

Edit1: TEdit;

RadioGroup1: TRadioGroup;

Panel3: TPanel;

LMDSpeedButton2: TLMDSpeedButton;

Image2: TImage;

RadioGroup2: TRadioGroup;

Panel4: TPanel;

Image3: TImage;

LMDSpeedButton3: TLMDSpeedButton;

LMDSpeedButton4: TLMDSpeedButton;

procedure LMDSpeedButton1Click(Sender: TObject);

procedure LMDSpeedButton2Click(Sender: TObject);

procedure LMDSpeedButton4Click(Sender: TObject);

procedure LMDSpeedButton3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Fdel: TFdel;

App, Ke, Abc : Variant;

ExelOtch : String;

Result : HResult;

AppProgID : string;

ServerIsRunning :boolean;

Unknown :IUnknown;

i:integer;

implementation

uses dm;

{$R \*.dfm}

### Продолжение приложения А

procedure TFdel.LMDSpeedButton1Click(Sender: TObject);

begin

if edit1.Text<>'' then begin

if RadioGroup1.ItemIndex=0 then

if dm1.avto\_out.Locate('marka\_out',edit1.Text,[loCaseInsensitive]) then

else MessageBox(Handle,'Нет автомобиля такой марки...','',MB\_OK);

if RadioGroup1.ItemIndex=1 then

if dm1.avto\_out.Locate('model\_out',edit1.Text,[loCaseInsensitive]) then

else MessageBox(Handle,'Нет автомобиля такой модели...','',MB\_OK);

end

else MessageBox(Handle,'Введите значение для поиска!','',MB\_OK);

end;

procedure TFdel.LMDSpeedButton2Click(Sender: TObject);

begin

if RadioGroup2.ItemIndex=0 then

dm1.avto\_out.IndexName:='marka\_out';

if RadioGroup2.ItemIndex=1 then

dm1.avto\_out.IndexName:='V\_out';

if RadioGroup2.ItemIndex=2 then

dm1.avto\_out.IndexName:='tipkuz\_out';

end;

procedure TFdel.LMDSpeedButton4Click(Sender: TObject);

begin

close;

end;

procedure TFdel.LMDSpeedButton3Click(Sender: TObject);

begin

AppProgID := 'Excel.Application';

ServerIsRunning := False;

Result:=GetActiveObject(ProgIDToClassID(AppProgID),nil,Unknown);

try

if (Result = MK\_E\_UNAVAILABLE) then

App := CreateOleObject(AppProgID)

else

begin

App := GetActiveOleObject(AppProgID);

ServerIsRunning := True;

end;

ExelOtch:=ExtractFilePath(Application.EXEName)+'Проданные автомобили на '+DateToStr(Date)+'.xls';

App.WorkBooks.Add;

Ke:=App.WorkBooks[1];

Abc:=Ke.WorkSheets[1];

Abc.Name:='Проданные авто('+DateToStr(Date)+').xls';

except on EinOutError do begin

messageBox(handle,'Не удаётся создать файл!','Ошибка',MB\_ICONERROR);

exit;

end;

end;

App.DisplayAlerts:=False;

Abc.Cells[1,3]:='Проданные авто('+DateToStr(Date)+').xls';

### Продолжение приложения А

Abc.Cells[1,3].Font.Bold:=True;

Abc.Cells[1,3].Font.Size:=11;

dm1.avto\_out.first;

i:=3;

while not dm1.avto\_out.eof do begin

i:=i+1;

app.cells[i,1]:=vartostr(dm1.avto\_out['Marka\_out']);

Abc.Cells[i,1].ColumnWidth:=15;

app.cells[i,2]:=vartostr(dm1.avto\_out['Model\_out']);

Abc.Cells[i,2].ColumnWidth:=15;

app.cells[i,3]:=vartostr(dm1.avto\_out['v\_out']);

Abc.Cells[i,3].ColumnWidth:=5;

app.cells[i,4]:=vartostr(dm1.avto\_out['cvet\_out']);

Abc.Cells[i,4].ColumnWidth:=15;

app.cells[i,5]:=vartostr(dm1.avto\_out['tipkuz\_out']);

Abc.Cells[i,5].ColumnWidth:=15;

app.cells[i,6]:=vartostr(dm1.avto\_out['cena\_out']);

Abc.Cells[i,6].ColumnWidth:=10;

app.cells[i,7]:=vartostr(dm1.avto\_out['date']);

Abc.Cells[i,7].ColumnWidth:=8;

dm1.avto\_out.next;

end;

App.ActiveWorkBook.SaveAs(ExelOtch);

App.Quit;

App:=Unassigned;

AppProgID := 'Excel.Application';

App := CreateOleObject(AppProgID);

App.Visible := True;

ExelOtch:=ExtractFilePath(Application.EXEName)+'Проданные автомобили на '+DateToStr(Date)+'.xls';

App.WorkBooks.Open(ExelOtch);

end;

end.

## А.4 Модуль данных

unit dm;

interface

uses

SysUtils, Classes, DB, DBTables;

type

Tdm1 = class(TDataModule)

dsvlad: TDataSource;

dsavto: TDataSource;

dssot: TDataSource;

dskomplect: TDataSource;

dsavto\_in: TDataSource;

dsavto\_out: TDataSource;

### Продолжение приложения А

sot: TTable;

komplect: TTable;

avto\_in: TTable;

avto\_out: TTable;

sotKOD\_SOT: TSmallintField;

sotFAM: TStringField;

sotNAME: TStringField;

sotOTH: TStringField;

avto\_inKOD\_IN: TFloatField;

avto\_inAKT\_NOM: TFloatField;

avto\_inKOD\_AVTO: TFloatField;

avto\_inKOD\_VLAD: TFloatField;

avto\_inKOD\_SOT: TFloatField;

avto\_inDATA: TDateField;

komplectKOD\_AVTO: TFloatField;

komplectKOLESO: TStringField;

komplectZAPASKA: TBooleanField;

komplectINSTRUMENT: TBooleanField;

komplectMUSIC: TBooleanField;

komplectMEDIC: TBooleanField;

komplectELECT\_PAK: TBooleanField;

komplectABS: TBooleanField;

komplectGUR: TBooleanField;

komplectKONDIC: TBooleanField;

komplectINFO: TMemoField;

komplectDEFEKT: TMemoField;

komplectKOROBKA: TStringField;

komplectPRIVOD: TStringField;

avto\_outKOD\_OUT: TFloatField;

avto\_outDATE: TDateField;

avto\_outMARKA\_OUT: TStringField;

avto\_outMODEL\_OUT: TStringField;

avto\_outV\_OUT: TStringField;

avto\_outCVET\_OUT: TStringField;

avto\_outTIPKUZ\_OUT: TStringField;

avto\_outCENA\_OUT: TFloatField;

dsmarka: TDataSource;

marka: TTable;

markaID\_MARKA: TSmallintField;

markaMARKA: TStringField;

vlad: TTable;

vladKOD\_VLAD: TFloatField;

vladFAM: TStringField;

vladNAME: TStringField;

vladOTH: TStringField;

vladADRES: TStringField;

vladMAIL: TStringField;

vladTEL: TFloatField;

vladSOT: TFloatField;

t1: TTable;

t1KOD\_AVTO: TFloatField;

t1KOD\_VLAD: TFloatField;

t1MARKA: TSmallintField;

t1MODEL: TStringField;

t1GOS\_NOM: TStringField;

t1CVET: TStringField;

t1TIP\_KUZ: TStringField;

t1FOTO\_1: TStringField;

t1FOTO\_2: TStringField;

### Продолжение приложения А

t1CENA: TFloatField;

t1hoz: TStringField;

t1V: TStringField;

t1MMarka: TStringField;

avto: TTable;

avtoKOD\_AVTO: TFloatField;

avtoKOD\_VLAD: TFloatField;

avtoMARKA: TSmallintField;

avtoMODEL: TStringField;

avtoGOS\_NOM: TStringField;

avtoCVET: TStringField;

avtoTIP\_KUZ: TStringField;

avtoFOTO\_1: TStringField;

avtoFOTO\_2: TStringField;

avtoCENA: TFloatField;

avtoV: TFloatField;

avtohoz: TStringField;

avtoMMarka: TStringField;

procedure DataModuleCreate(Sender: TObject);

procedure avtoAfterScroll(DataSet: TDataSet);

procedure markaBeforeInsert(DataSet: TDataSet);

procedure markaAfterInsert(DataSet: TDataSet);

private

{ Private declarations }

public

{ Public declarations }

end;

var

dm1: Tdm1;

id : integer;

implementation

uses main;

{$R \*.dfm}

procedure Tdm1.DataModuleCreate(Sender: TObject);

begin

avto.Close;

avto.Open;

komplect.Close;

komplect.Open;

avto\_in.Close;

avto\_in.Open;

avto\_out.Close;

avto\_out.Open;

vlad.Close;

vlad.Open;

sot.Close;

sot.Open;

end;

procedure Tdm1.avtoAfterScroll(DataSet: TDataSet);

begin

try

Fmain.Image3.Picture.LoadFromFile(ExtractFilePath(path)+'FOTO\'+dm1.avto.fieldbyname('FOTO\_1').AsString);

### Продолжение приложения А

except

Fmain.Image3.picture:=nil;

end;

try

Fmain.Image4.Picture.LoadFromFile(ExtractFilePath(path)+'FOTO\'+dm1.avto.fieldbyname('FOTO\_2').AsString);

except

Fmain.Image4.picture:=nil;

end;

end;

procedure Tdm1.markaBeforeInsert(DataSet: TDataSet);

begin

id:=0;

Marka.First;

while not Marka.Eof do begin

if Marka.FieldByName('Id\_marka').asinteger > id

then id := Marka.FieldByName('Id\_marka').asinteger;

Marka.Next;

end;

id:=id+1;

end;

procedure Tdm1.markaAfterInsert(DataSet: TDataSet);

begin

Marka.FieldByName('id\_marka').asinteger:=id;

end;

end.

## А.5 Модуль формы «Редактирование данных»

unit edit;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ExtCtrls, LMDControl, LMDBaseControl, LMDBaseGraphicButton,

LMDCustomSpeedButton, LMDSpeedButton, StdCtrls, RxGIF, glLabel, Mask,

ToolEdit, DBCtrls,IDGlobal, ExtDlgs, Grids, DBGrids;

type

TFedit = class(TForm)

Panel3: TPanel;

glLabel2: TglLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

Image1: TImage;

ComboBox1: TComboBox;

Panel4: TPanel;

glLabel3: TglLabel;

Label15: TLabel;

### Продолжение приложения А

Label16: TLabel;

Label17: TLabel;

Label18: TLabel;

Label19: TLabel;

ComboBox2: TComboBox;

ComboBox3: TComboBox;

ComboBox4: TComboBox;

Panel1: TPanel;

glLabel4: TglLabel;

LMDSpeedButton8: TLMDSpeedButton;

LMDSpeedButton9: TLMDSpeedButton;

Image4: TImage;

Image5: TImage;

Panel5: TPanel;

LMDSpeedButton10: TLMDSpeedButton;

LMDSpeedButton12: TLMDSpeedButton;

DBEdit2: TDBEdit;

DBEdit3: TDBEdit;

DBEdit4: TDBEdit;

DBEdit5: TDBEdit;

DBEdit6: TDBEdit;

DBMemo1: TDBMemo;

DBMemo2: TDBMemo;

DBCheckBox1: TDBCheckBox;

DBCheckBox2: TDBCheckBox;

DBCheckBox3: TDBCheckBox;

DBCheckBox4: TDBCheckBox;

DBCheckBox5: TDBCheckBox;

DBCheckBox6: TDBCheckBox;

DBCheckBox7: TDBCheckBox;

DBCheckBox8: TDBCheckBox;

OpenPictureDialog1: TOpenPictureDialog;

OpenPictureDialog2: TOpenPictureDialog;

Image3: TImage;

DBLookupComboBox1: TDBLookupComboBox;

procedure LMDSpeedButton12Click(Sender: TObject);

procedure LMDSpeedButton10Click(Sender: TObject);

procedure LMDSpeedButton8Click(Sender: TObject);

procedure LMDSpeedButton9Click(Sender: TObject);

procedure FormActivate(Sender: TObject);

procedure DBEdit6KeyPress(Sender: TObject; var Key: Char);

procedure DBEdit1KeyPress(Sender: TObject; var Key: Char);

procedure DBEdit3KeyPress(Sender: TObject; var Key: Char);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Fedit: TFedit;

pic1,pic2:string;

pole1,pole2,pole1\_1,pole2\_2:string;

y,t:integer;

implementation

uses dm, DB;

{$R \*.dfm}

### Продолжение приложения А

procedure TFedit.LMDSpeedButton12Click(Sender: TObject);

begin

close;

end;

procedure TFedit.LMDSpeedButton10Click(Sender: TObject);

var st1,q1,st2,q2:string;

i,u,k2:integer;

begin

dm1.avto.Edit;

dm1.komplect.Edit;

dm1.avto.FieldByName('TIP\_KUZ').Value:=ComboBox1.text;

dm1.komplect.FieldByName('KOLESO').value:=ComboBox2.text;

dm1.komplect.FieldByName('KOROBKA').value:=ComboBox3.text;

dm1.komplect.FieldByName('PRIVOD').value:=ComboBox4.text;

k2:=dm1.avto.fieldbyname('KOD\_avto').Value;

if (pic1<>'') then begin

pole1\_1:='';

pole1:='';

pole1:=dm1.avto.fieldbyname('FOTO\_1').AsString;

for y:=1 to length(pole1) do

if pole1[y]='.' then pole1\_1:=copy(pole1,1,y-1);

st1:='';

q1:='';

for i:=length(pic1) downto 1 do

if pic1[i]<>'.' then insert(pic1[i],q1,0) else break;

st1:=ExtractFilePath(Application.Exename);

ForceDirectories(st1+'FOTO');

if dm1.avto.FieldByName('foto\_1').AsString<>'' then

DeleteFile(st1+'FOTO\'+dm1.avto.FieldByName('foto\_1').AsString);

if copyfileto(pchar(pic1),pchar(st1+'FOTO\'+pole1\_1+ '.'+q1)) then

else begin

MessageBox(Handle,'Невозможно скопировать файл!','Внимание',MB\_OK);

exit;

end;

dm1.avto.FieldByName('foto\_1').AsString:=pole1\_1 + '.'+q1;

end;

if (pic2<>'') then begin

pole2\_2:='';

pole2:='';

pole2:=dm1.avto.fieldbyname('FOTO\_2').AsString;

for t:=1 to length(pole2) do

if pole2[t]='.' then pole2\_2:=copy(pole2,1,t-1);

st2:='';

q2:='';

for u:=length(pic2) downto 1 do

if pic2[u]<>'.' then insert(pic2[u],q2,0) else break;

st2:=ExtractFilePath(Application.Exename);

ForceDirectories(st2+'FOTO');

if dm1.avto.FieldByName('foto\_2').AsString<>'' then

DeleteFile(st2+'FOTO\'+dm1.avto.FieldByName('foto\_2').AsString);

### Продолжение приложения А

if copyfileto(pchar(pic2),pchar(st2+'FOTO\'+pole2\_2+ '.'+q2)) then

else begin

MessageBox(Handle,'Невозможно скопировать файл!','Внимание',MB\_OK);

exit;

end;

dm1.avto.FieldByName('foto\_2').AsString:=pole2\_2 + '.'+q2;

end;

dm1.avto.Post;

// dm1.komplect.Post;

end;

procedure TFedit.LMDSpeedButton8Click(Sender: TObject);

begin

OpenPictureDialog1.Execute;

pic1:=OpenPictureDialog1.FileName;

if pic1<>'' then Image4.Picture.LoadFromFile(pic1)

else begin

MessageBox(Handle,'Фотография кузова не выбрана!','',MB\_OK);

exit;

end;

end;

procedure TFedit.LMDSpeedButton9Click(Sender: TObject);

begin

OpenPictureDialog2.Execute;

pic2:=OpenPictureDialog2.FileName;

if pic2<>'' then Image5.Picture.LoadFromFile(pic2)

else begin

MessageBox(Handle,'Фотография салона не выбрана!','',MB\_OK);

exit;

end;

end;

procedure TFedit.FormActivate(Sender: TObject);

begin

if (dm1.avto.fieldbyname('FOTO\_1').Value<>'') then

Image4.Picture.LoadFromFile(ExtractFilePath(Application.Exename)+'FOTO\'+dm1.avto.fieldbyname('FOTO\_1').AsString)

else MessageBox(Handle,'Фотографии кузова НЕТ!','',MB\_OK);

if (dm1.avto.fieldbyname('FOTO\_2').Value<>'') then

Image5.Picture.LoadFromFile(ExtractFilePath(Application.Exename)+'FOTO\'+dm1.avto.fieldbyname('FOTO\_2').AsString)

else MessageBox(Handle,'Фотографии салона НЕТ!','',MB\_OK);

end;

procedure TFedit.DBEdit6KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'0'..'9',#8: key:=key

else key:=#0;

end;

end;

procedure TFedit.DBEdit1KeyPress(Sender: TObject; var Key: Char);

begin

### Продолжение приложения А

case key of

'A'..'Z','a'..'z','а'..'я','А'..'Я',#8: key:=key

else key:=#0;

end;

end;

procedure TFedit.DBEdit3KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'A'..'Z','a'..'z','а'..'я','А'..'Я',#8: key:=key

else key:=#0;

end;

end;

end.

## А.6 Модуль заставки

unit logo;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ExtCtrls, StdCtrls, ComCtrls, RxGIF;

type

TFlogo = class(TForm)

Timer1: TTimer;

Image1: TImage;

ProgressBar1: TProgressBar;

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure Timer1Timer(Sender: TObject);

procedure FormActivate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Flogo: TFlogo;

i : integer;

implementation

{$R \*.dfm}

procedure TFlogo.FormClose(Sender: TObject; var Action: TCloseAction);

begin

if i=100 then

Action:=caFree

else

Action := caNone;

end;

procedure TFlogo.Timer1Timer(Sender: TObject);

begin

i:=i+1;

### Продолжение приложения А

ProgressBar1.Position:=i;

if i=100 then begin

Timer1.Enabled:=false;

close;

end;

end;

procedure TFlogo.FormActivate(Sender: TObject);

begin

i:=0;

end;

end.

## А.7 Модуль главной формы

unit main;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ComCtrls, Menus, Grids, DBGrids, Buttons, LMDSpeedButton,

StdCtrls, XPMan, LMDBaseGraphicButton, LMDCustomSpeedButton, LMDControl,

LMDBaseControl, DBCtrls, ImgList, LMDBaseGraphicControl,

LMDGraphicControl, LMDBaseMeter, LMDCustomProgress,

ExtCtrls, DBTables, DB, LMDLEDCustomLabel, LMDLEDLabel,IDGlobal;

type

TFmain = class(TForm)

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N6: TMenuItem;

N8: TMenuItem;

N9: TMenuItem;

N11: TMenuItem;

N10: TMenuItem;

N12: TMenuItem;

N13: TMenuItem;

N14: TMenuItem;

N15: TMenuItem;

N81: TMenuItem;

N82: TMenuItem;

N16: TMenuItem;

N17: TMenuItem;

ColorDialog1: TColorDialog;

XPManifest1: TXPManifest;

ImageList1: TImageList;

Panel1: TPanel;

DBGrid2: TDBGrid;

DBGrid1: TDBGrid;

DBGrid3: TDBGrid;

Timer1: TTimer;

N7: TMenuItem;

### Продолжение приложения А

N18: TMenuItem;

N19: TMenuItem;

N20: TMenuItem;

Panel9: TPanel;

Panel6: TPanel;

Image3: TImage;

Image4: TImage;

Panel5: TPanel;

DBText1: TDBText;

DBText2: TDBText;

DBText3: TDBText;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

DBCheckBox1: TDBCheckBox;

DBCheckBox2: TDBCheckBox;

DBCheckBox3: TDBCheckBox;

DBCheckBox4: TDBCheckBox;

DBCheckBox5: TDBCheckBox;

DBCheckBox6: TDBCheckBox;

DBCheckBox7: TDBCheckBox;

DBCheckBox8: TDBCheckBox;

DBMemo1: TDBMemo;

DBMemo2: TDBMemo;

Panel2: TPanel;

RadioGroup1: TRadioGroup;

Panel7: TPanel;

Panel10: TPanel;

LMDSpeedButton3: TLMDSpeedButton;

LMDSpeedButton4: TLMDSpeedButton;

LMDSpeedButton5: TLMDSpeedButton;

LMDSpeedButton6: TLMDSpeedButton;

LMDSpeedButton7: TLMDSpeedButton;

LMDSpeedButton8: TLMDSpeedButton;

Edit1: TEdit;

LMDSpeedButton1: TLMDSpeedButton;

Label7: TLabel;

Label6: TLabel;

N21: TMenuItem;

N22: TMenuItem;

N23: TMenuItem;

procedure FormShow(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure N14Click(Sender: TObject);

procedure N81Click(Sender: TObject);

procedure N15Click(Sender: TObject);

procedure N82Click(Sender: TObject);

procedure N8Click(Sender: TObject);

procedure N9Click(Sender: TObject);

procedure N11Click(Sender: TObject);

procedure DBGrid1CellClick(Column: TColumn);

procedure N12Click(Sender: TObject);

procedure LMDSpeedButton1Click(Sender: TObject);

procedure LMDSpeedButton3Click(Sender: TObject);

procedure LMDSpeedButton4Click(Sender: TObject);

procedure LMDSpeedButton5Click(Sender: TObject);

procedure LMDSpeedButton6Click(Sender: TObject);

### Продолжение приложения А

procedure LMDSpeedButton9Click(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

procedure LMDSpeedButton10Click(Sender: TObject);

procedure N7Click(Sender: TObject);

procedure N19Click(Sender: TObject);

procedure LMDSpeedButton7Click(Sender: TObject);

procedure N10Click(Sender: TObject);

procedure DBGrid1KeyPress(Sender: TObject; var Key: Char);

procedure FormCreate(Sender: TObject);

procedure DBGrid1TitleClick(Column: TColumn);

procedure N22Click(Sender: TObject);

procedure N23Click(Sender: TObject);

procedure N17Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Fmain: TFmain;

path : string;

implementation

uses logo, dm, vladel, sotr, add, edit, del, query, search, marka, about;

{$R \*.dfm}

function gen\_id(table:TTable; var name:string):integer;

var n:integer;

begin

table.First;

n:=table.FieldByName(name).asinteger;

while not table.Eof do begin

if table.FieldByName(name).Value>n then

n:=table.FieldByName(name).Value;

table.Next;

end;

gen\_id:=n+1;

end;

procedure TFmain.FormShow(Sender: TObject);

begin

flogo.showmodal;

flogo.Free;

Image3.Picture.LoadFromFile(ExtractFilePath(Application.Exename)+'FOTO\'+dm1.avto.fieldbyname('FOTO\_1').AsString);

Image4.Picture.LoadFromFile(ExtractFilePath(Application.Exename)+'FOTO\'+dm1.avto.fieldbyname('FOTO\_2').AsString);

end;

procedure TFmain.N5Click(Sender: TObject);

begin

close;

end;

procedure TFmain.N14Click(Sender: TObject);

begin

### Продолжение приложения А

ColorDialog1.Execute;

DBGrid1.Font.Color:=ColorDialog1.color;

end;

procedure TFmain.N81Click(Sender: TObject);

begin

DBGrid1.Font.Size:=10;

end;

procedure TFmain.N15Click(Sender: TObject);

begin

DBGrid1.Font.Size:=12;

end;

procedure TFmain.N82Click(Sender: TObject);

begin

DBGrid1.Font.Size:=8;

end;

procedure TFmain.N8Click(Sender: TObject);

begin

fvlad.showmodal;

end;

procedure TFmain.N9Click(Sender: TObject);

begin

fsotr.showmodal;

end;

procedure TFmain.N11Click(Sender: TObject);

begin

fadd.showmodal;

end;

procedure TFmain.DBGrid1CellClick(Column: TColumn);

begin

Image3.Picture.LoadFromFile(ExtractFilePath(Application.Exename)+'FOTO\'+dm1.avto.fieldbyname('FOTO\_1').AsString);

Image4.Picture.LoadFromFile(ExtractFilePath(Application.Exename)+'FOTO\'+dm1.avto.fieldbyname('FOTO\_2').AsString);

end;

procedure TFmain.N12Click(Sender: TObject);

begin

Fedit.showmodal;

end;

procedure TFmain.LMDSpeedButton1Click(Sender: TObject);

begin

if edit1.Text<>'' then begin

if RadioGroup1.ItemIndex=0 then

if dm1.avto.Locate('marka',edit1.Text,[loCaseInsensitive]) then

else MessageBox(Handle,'Нет автомобиля такой марки...','',MB\_OK);

if RadioGroup1.ItemIndex=1 then

if dm1.avto.Locate('model',edit1.Text,[loCaseInsensitive]) then

else MessageBox(Handle,'Нет автомобиля такой модели...','',MB\_OK);

if RadioGroup1.ItemIndex=2 then

### Продолжение приложения А

if dm1.avto.Locate('gos\_nom',edit1.Text,[loCaseInsensitive]) then

else MessageBox(Handle,'Нет автомобиля c таким номером...','',MB\_OK);

end

else MessageBox(Handle,'Введите значение для поиска!','',MB\_OK);

end;

procedure TFmain.LMDSpeedButton3Click(Sender: TObject);

begin

fsotr.showmodal;

end;

procedure TFmain.LMDSpeedButton4Click(Sender: TObject);

begin

fvlad.showmodal;

end;

procedure TFmain.LMDSpeedButton5Click(Sender: TObject);

begin

fadd.showmodal;

end;

procedure TFmain.LMDSpeedButton6Click(Sender: TObject);

begin

Fedit.showmodal;

end;

procedure TFmain.LMDSpeedButton9Click(Sender: TObject);

begin

//close;

Panel9.Visible := not Panel9.Visible;

end;

procedure TFmain.Timer1Timer(Sender: TObject);

begin

Label6.Caption:=TimeToStr(Time);

end;

procedure TFmain.LMDSpeedButton10Click(Sender: TObject);

begin

fquery.showmodal;

end;

procedure TFmain.N7Click(Sender: TObject);

begin

fdel.showmodal;

end;

procedure TFmain.N19Click(Sender: TObject);

begin

fquery.showmodal;

end;

procedure TFmain.LMDSpeedButton7Click(Sender: TObject);

var v:string;

k:integer;

begin

dm1.komplect.Edit;

dm1.avto.Edit;

### Продолжение приложения А

v:='KOD\_out';

k:=gen\_id(dm1.avto\_out,v);

dm1.avto\_out.Append;

dm1.avto\_out.FieldByName('KOD\_out').Value:=k;

dm1.avto\_out.FieldByName('marka\_out').Value:=dm1.avto.FieldByName('marka').Value;

dm1.avto\_out.FieldByName('model\_out').value:=dm1.avto.FieldByName('model').value;

dm1.avto\_out.FieldByName('V\_out').value:=dm1.avto.FieldByName('V').value;

dm1.avto\_out.FieldByName('cvet\_out').value:=dm1.avto.FieldByName('cvet').value;

dm1.avto\_out.FieldByName('tipkuz\_out').value:=dm1.avto.FieldByName('tip\_kuz').value;

dm1.avto\_out.FieldByName('cena\_out').value:=dm1.avto.FieldByName('cena').value;

dm1.avto\_out.FieldByName('date').value:=Date;

if messagedlg('Продать?!',mtconfirmation, [mbYes, mbNo],0)=mrYes then

begin

while not dm1.komplect.Eof do dm1.komplect.Delete;

dm1.avto.Delete;

end;

DeleteFile(ExtractFilePath(Application.Exename)+'FOTO\'+dm1.avto.FieldByName('foto\_1').AsString);

DeleteFile(ExtractFilePath(Application.Exename)+'FOTO\'+dm1.avto.FieldByName('foto\_2').AsString);

//dm1.avto.Post;

//dm1.komplect.Post;

dm1.avto\_out.Post;

end;

procedure TFmain.N10Click(Sender: TObject);

begin

LMDSpeedButton7Click(Sender);

end;

procedure TFmain.DBGrid1KeyPress(Sender: TObject; var Key: Char);

begin

Image3.Picture.LoadFromFile(ExtractFilePath(Application.Exename)+'FOTO\'+dm1.avto.fieldbyname('FOTO\_1').AsString);

Image4.Picture.LoadFromFile(ExtractFilePath(Application.Exename)+'FOTO\'+dm1.avto.fieldbyname('FOTO\_2').AsString);

end;

procedure TFmain.FormCreate(Sender: TObject);

begin

path:=Application.ExeName;

end;

### Продолжение приложения А

procedure TFmain.DBGrid1TitleClick(Column: TColumn);

begin

if Column.Title.Caption='Марка' then

dm1.avto.IndexName:='marka';

if Column.Title.Caption='Объём' then

dm1.avto.IndexName:='V';

if Column.Title.Caption='Тип кузова' then

dm1.avto.IndexName:='tip\_kuz';

end;

procedure TFmain.N22Click(Sender: TObject);

begin

FSearch.showmodal;

end;

procedure TFmain.N23Click(Sender: TObject);

begin

FMarka.Showmodal;

end;

procedure TFmain.N17Click(Sender: TObject);

begin

Fabout.ShowModal;

end;

end.

## А.8 Модуль формы поиска

unit search;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, DBCtrls, StdCtrls, Mask, glLabel, ExtCtrls, Grids, DBGrids,

LMDCustomButton, LMDButton, DB, DBTables, RXDBCtrl;

type

TFSearch = class(TForm)

Panel3: TPanel;

glLabel2: TglLabel;

Label8: TLabel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

ComboBox1: TComboBox;

Panel4: TPanel;

glLabel3: TglLabel;

Label15: TLabel;

Label16: TLabel;

Label17: TLabel;

ComboBox2: TComboBox;

ComboBox3: TComboBox;

ComboBox4: TComboBox;

### Продолжение приложения А

CheckBox1: TCheckBox;

CheckBox2: TCheckBox;

CheckBox3: TCheckBox;

Panel1: TPanel;

DBLookupComboBox1: TDBLookupComboBox;

CheckBox4: TCheckBox;

CheckBox5: TCheckBox;

CheckBox6: TCheckBox;

CheckBox7: TCheckBox;

CheckBox8: TCheckBox;

CheckBox9: TCheckBox;

CheckBox10: TCheckBox;

CheckBox11: TCheckBox;

CheckBox12: TCheckBox;

Edit1: TEdit;

Edit2: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

CheckBox13: TCheckBox;

CheckBox14: TCheckBox;

CheckBox15: TCheckBox;

CheckBox16: TCheckBox;

CheckBox17: TCheckBox;

Query1: TQuery;

LMDButton1: TLMDButton;

DataSource1: TDataSource;

RxDBGrid1: TRxDBGrid;

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure CheckBox9Click(Sender: TObject);

procedure CheckBox10Click(Sender: TObject);

procedure CheckBox11Click(Sender: TObject);

procedure CheckBox12Click(Sender: TObject);

procedure CheckBox4Click(Sender: TObject);

procedure CheckBox5Click(Sender: TObject);

procedure CheckBox6Click(Sender: TObject);

procedure CheckBox7Click(Sender: TObject);

procedure CheckBox8Click(Sender: TObject);

procedure CheckBox1Click(Sender: TObject);

procedure CheckBox2Click(Sender: TObject);

procedure CheckBox3Click(Sender: TObject);

procedure LMDButton1Click(Sender: TObject);

procedure DBGrid1CellClick(Column: TColumn);

private

{ Private declarations }

public

{ Public declarations }

end;

var

FSearch: TFSearch;

implementation

uses dm, edit;

{$R \*.dfm}

### Продолжение приложения А

procedure TFSearch.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Action:=caFree;

end;

procedure TFSearch.CheckBox9Click(Sender: TObject);

begin

Label8.Enabled:=not Label8.Enabled;

DBLookupComboBox1.Enabled:=not DBLookupComboBox1.Enabled;

end;

procedure TFSearch.CheckBox10Click(Sender: TObject);

begin

Label12.Enabled:=not Label12.Enabled;

ComboBox1.Enabled:=not ComboBox1.Enabled;

end;

procedure TFSearch.CheckBox11Click(Sender: TObject);

begin

Label14.Enabled:=not Label14.Enabled;

Edit1.Enabled:=not Edit1.Enabled;

end;

procedure TFSearch.CheckBox12Click(Sender: TObject);

begin

Label13.Enabled:=not Label13.Enabled;

Edit2.Enabled:=not Edit2.Enabled;

end;

procedure TFSearch.CheckBox4Click(Sender: TObject);

begin

Label1.Enabled:=not Label1.Enabled;

CheckBox17.Enabled:=not CheckBox17.Enabled;

end;

procedure TFSearch.CheckBox5Click(Sender: TObject);

begin

Label2.Enabled:=not Label2.Enabled;

CheckBox16.Enabled:=not CheckBox16.Enabled;

end;

procedure TFSearch.CheckBox6Click(Sender: TObject);

begin

Label3.Enabled:=not Label3.Enabled;

CheckBox15.Enabled:=not CheckBox15.Enabled;

end;

procedure TFSearch.CheckBox7Click(Sender: TObject);

begin

Label4.Enabled:=not Label4.Enabled;

CheckBox14.Enabled:=not CheckBox14.Enabled;

end;

procedure TFSearch.CheckBox8Click(Sender: TObject);

begin

Label5.Enabled:=not Label5.Enabled;

CheckBox13.Enabled:=not CheckBox13.Enabled;

end;

### Продолжение приложения А

procedure TFSearch.CheckBox1Click(Sender: TObject);

begin

Label15.Enabled:=not Label15.Enabled;

ComboBox2.Enabled:=not ComboBox2.Enabled;

end;

procedure TFSearch.CheckBox2Click(Sender: TObject);

begin

Label16.Enabled:=not Label16.Enabled;

ComboBox3.Enabled:=not ComboBox3.Enabled;

end;

procedure TFSearch.CheckBox3Click(Sender: TObject);

begin

Label17.Enabled:=not Label17.Enabled;

ComboBox4.Enabled:=not ComboBox4.Enabled;

end;

function BoolToStr(value : boolean):string;

begin

if value then result:='true' else result:='false';

end;

procedure TFSearch.LMDButton1Click(Sender: TObject);

begin

Query1.Close;

Query1.SQL.Clear;

try

Query1.Sql.Add('select a.kod\_avto, m.marka, a.model, a.cvet, a.tip\_kuz, a.cena, a.v, k.korobka,k.privod ');

Query1.SQL.Add('from avto a, komplect k, marka m ');

Query1.SQL.Add('where ');

if checkBox9.Checked then Query1.SQL.Add('a.marka = '+ Inttostr(DBLookupComboBox1.keyvalue)+' and');

if checkBox10.Checked then Query1.SQL.Add('a.tip\_kuz="'+ComboBox1.Text+'" and ');

if checkBox11.Checked then Query1.SQL.Add('a.v<='+Edit1.Text+' and ');

if checkBox12.Checked then Query1.SQL.Add('a.cena<='+Edit2.Text+' and ');

if checkBox4.Checked then Query1.SQL.Add('k.elect\_pak = '+BoolToStr(CheckBox17.Checked)+' and ');

if checkBox5.Checked then Query1.SQL.Add('k.kondic = '+BoolToStr(CheckBox16.Checked)+' and ');

if checkBox6.Checked then Query1.SQL.Add('k.music = '+BoolToStr(CheckBox15.Checked)+' and ');

if checkBox7.Checked then Query1.SQL.Add('k.gur = '+BoolToStr(CheckBox14.Checked)+' and ');

if checkBox8.Checked then Query1.SQL.Add('k.abs = '+BoolToStr(CheckBox13.Checked)+' and ');

if checkBox1.Checked then Query1.SQL.Add('k.koleso = "'+ComboBox2.Text+'" and ');

if checkBox2.Checked then Query1.SQL.Add('k.korobka = "'+ComboBox3.Text+'" and ');

if checkBox3.Checked then Query1.SQL.Add('k.privod = "'+ComboBox4.Text+'" and ');

Query1.SQL.Add('a.marka=m.id\_marka and a.kod\_avto=k.kod\_avto');

Query1.Open;

except

### Продолжение приложения А

MessageDlg('Введены не все данные',mtWarning,[mbOk],0);

end;

end;

procedure TFSearch.DBGrid1CellClick(Column: TColumn);

begin

if Query1.Active then begin

if dm1.avto.Locate('kod\_avto',Query1.FieldByName('kod\_avto').AsInteger,[loCaseInsensitive])

then fedit.showmodal;

end;

end;

end.

## А.9 Модуль формы менеджеров

unit sotr;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Grids, DBGrids, ExtCtrls, LMDControl, LMDBaseControl,

LMDBaseGraphicButton, DB, LMDCustomSpeedButton, LMDSpeedButton, StdCtrls,

Mask, DBTables, DBCtrls, ImgList;

type

TFsotr = class(TForm)

Panel1: TPanel;

DBGrid1: TDBGrid;

Panel2: TPanel;

pedit: TPanel;

Panel4: TPanel;

RadioGroup1: TRadioGroup;

Edit1: TEdit;

LMDSpeedButton1: TLMDSpeedButton;

Image1: TImage;

LMDSpeedButton2: TLMDSpeedButton;

LMDSpeedButton3: TLMDSpeedButton;

LMDSpeedButton4: TLMDSpeedButton;

LMDSpeedButton5: TLMDSpeedButton;

pnew: TPanel;

Image2: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

Edit6: TEdit;

Edit7: TEdit;

LMDSpeedButton6: TLMDSpeedButton;

### Продолжение приложения А

LMDSpeedButton7: TLMDSpeedButton;

LMDSpeedButton8: TLMDSpeedButton;

LMDSpeedButton9: TLMDSpeedButton;

Image3: TImage;

Image4: TImage;

procedure LMDSpeedButton1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure LMDSpeedButton4Click(Sender: TObject);

procedure LMDSpeedButton5Click(Sender: TObject);

procedure LMDSpeedButton2Click(Sender: TObject);

procedure LMDSpeedButton7Click(Sender: TObject);

procedure LMDSpeedButton9Click(Sender: TObject);

procedure FormActivate(Sender: TObject);

procedure LMDSpeedButton8Click(Sender: TObject);

procedure LMDSpeedButton6Click(Sender: TObject);

procedure LMDSpeedButton3Click(Sender: TObject);

procedure Edit2KeyPress(Sender: TObject; var Key: Char);

procedure Edit3KeyPress(Sender: TObject; var Key: Char);

procedure Edit4KeyPress(Sender: TObject; var Key: Char);

procedure Edit5KeyPress(Sender: TObject; var Key: Char);

procedure Edit6KeyPress(Sender: TObject; var Key: Char);

procedure Edit7KeyPress(Sender: TObject; var Key: Char);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Fsotr: TFsotr;

implementation

uses dm;

{$R \*.dfm}

function gen\_id(table:TTable; var name:string):integer;

var n:integer;

begin

table.First;

n:=table.FieldByName(name).asinteger;

while not table.Eof do begin

if table.FieldByName(name).Value>n then

n:=table.FieldByName(name).Value;

table.Next;

end;

gen\_id:=n+1;

end;

procedure TFsotr.LMDSpeedButton1Click(Sender: TObject);

begin

if edit1.Text<>'' then begin

if RadioGroup1.ItemIndex=0 then

if dm1.sot.Locate('FAM',edit1.Text,[loCaseInsensitive]) then

else MessageBox(Handle,'Нет записи с такой фамилией...','',MB\_OK);

if RadioGroup1.ItemIndex=1 then

if dm1.sot.Locate('NAME',edit1.Text,[loCaseInsensitive]) then

### Продолжение приложения А

else MessageBox(Handle,'Нет записи с таким именем...','',MB\_OK);

end

else MessageBox(Handle,'Введите значение для поиска!','',MB\_OK);

end;

procedure TFsotr.FormCreate(Sender: TObject);

begin

dm1.sot.IndexName:='FAM';

RadioGroup1.ItemIndex:=-1;

end;

procedure TFsotr.LMDSpeedButton4Click(Sender: TObject);

begin

if messagedlg('Удалить?!',mtconfirmation, [mbYes, mbNo],0)=mrYes then begin

if dm1.sot.State=dsbrowse then dm1.sot.delete;

end;

end;

procedure TFsotr.LMDSpeedButton5Click(Sender: TObject);

begin

//dm1.sot.Edit;

//dm1.sot.Append;

pedit.Visible:=true;

Edit7.Text:=dm1.sot.fieldbyname('fam').Value;

Edit6.Text:=dm1.sot.fieldbyname('name').Value;

Edit5.Text:=dm1.sot.fieldbyname('oth').Value;

end;

procedure TFsotr.LMDSpeedButton2Click(Sender: TObject);

begin

pnew.Visible:=true;

end;

procedure TFsotr.LMDSpeedButton7Click(Sender: TObject);

begin

pedit.Visible:=false;

Edit5.Text:='';

Edit6.Text:='';

Edit7.Text:='';

dm1.sot.Cancel;

end;

procedure TFsotr.LMDSpeedButton9Click(Sender: TObject);

begin

pnew.Visible:=false;

Edit2.Text:='';

Edit3.Text:='';

Edit4.Text:='';

dm1.sot.Cancel;

end;

procedure TFsotr.FormActivate(Sender: TObject);

begin

pedit.Visible:=true;

pnew.Visible:=true;

pedit.Visible:=false;

pnew.Visible:=false;

end;

### Продолжение приложения А

procedure TFsotr.LMDSpeedButton8Click(Sender: TObject);

var v:string;

k:integer;

begin

if (Edit2.Text='') or (Edit3.Text='') or (Edit4.Text='') then begin

MessageBox(Handle,'Возможно некоторые поля не заполнены!','Внимание',MB\_OK);

exit;

end

else begin

v:='KOD\_sot';

k:=gen\_id(dm1.sot,v);

dm1.sot.Insert;

dm1.sot.FieldByName('KOD\_sot').Value:=k;

dm1.sot.FieldByName('FAM').Value:=edit2.text;

dm1.sot.FieldByName('NAME').value:=edit3.Text;

dm1.sot.FieldByName('OTH').value:=edit4.text;

dm1.sot.Post;

pnew.Visible:=false;

Edit2.Text:='';

Edit3.Text:='';

Edit4.Text:='';

end;

end;

procedure TFsotr.LMDSpeedButton6Click(Sender: TObject);

begin

if (Edit5.Text='') or (Edit6.Text='') or (Edit7.Text='') then begin

MessageBox(Handle,'Возможно некоторые поля не заполнены!','Внимание',MB\_OK);

exit;

end

else begin

dm1.sot.Edit;

dm1.sot.FieldByName('FAM').Value:=edit7.text;

dm1.sot.FieldByName('NAME').value:=edit6.Text;

dm1.sot.FieldByName('OTH').value:=edit5.text;

dm1.sot.Post;

pedit.Visible:=false;

Edit5.Text:='';

Edit6.Text:='';

Edit7.Text:='';

end;

end;

procedure TFsotr.LMDSpeedButton3Click(Sender: TObject);

begin

edit1.Text:='';

edit2.Text:='';

edit3.Text:='';

edit4.Text:='';

edit5.Text:='';

edit6.Text:='';

edit7.Text:='';

close;

end;

procedure TFsotr.Edit2KeyPress(Sender: TObject; var Key: Char);

begin

case key of

### Продолжение приложения А

'A'..'Z','a'..'z','а'..'я','А'..'Я',#8: key:=key

else key:=#0;

end;

end;

procedure TFsotr.Edit3KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'A'..'Z','a'..'z','а'..'я','А'..'Я',#8: key:=key

else key:=#0;

end;

end;

procedure TFsotr.Edit4KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'A'..'Z','a'..'z','а'..'я','А'..'Я',#8: key:=key

else key:=#0;

end;

end;

procedure TFsotr.Edit5KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'A'..'Z','a'..'z','а'..'я','А'..'Я',#8: key:=key

else key:=#0;

end;

end;

procedure TFsotr.Edit6KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'A'..'Z','a'..'z','а'..'я','А'..'Я',#8: key:=key

else key:=#0;

end;

end;

procedure TFsotr.Edit7KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'A'..'Z','a'..'z','а'..'я','А'..'Я',#8: key:=key

else key:=#0;

end;

end;

end.

## А.10 Модуль формы о владельцах

unit vladel;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Grids, DBGrids, ExtCtrls, StdCtrls, LMDControl, LMDBaseControl,

LMDBaseGraphicButton, LMDCustomSpeedButton, DB, LMDSpeedButton, Buttons;

### Продолжение приложения А

type

TFvlad = class(TForm)

Panel1: TPanel;

DBGrid1: TDBGrid;

Panel2: TPanel;

Panel3: TPanel;

RadioGroup1: TRadioGroup;

Edit1: TEdit;

LMDSpeedButton1: TLMDSpeedButton;

Image1: TImage;

RadioGroup2: TRadioGroup;

LMDSpeedButton2: TLMDSpeedButton;

Panel4: TPanel;

Image2: TImage;

Image3: TImage;

LMDSpeedButton4: TLMDSpeedButton;

pedit: TPanel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

LMDSpeedButton8: TLMDSpeedButton;

LMDSpeedButton9: TLMDSpeedButton;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label7: TLabel;

Edit5: TEdit;

Edit6: TEdit;

Edit7: TEdit;

Edit8: TEdit;

Image4: TImage;

procedure FormCreate(Sender: TObject);

procedure LMDSpeedButton1Click(Sender: TObject);

procedure LMDSpeedButton2Click(Sender: TObject);

procedure FormActivate(Sender: TObject);

procedure LMDSpeedButton8Click(Sender: TObject);

procedure LMDSpeedButton4Click(Sender: TObject);

procedure LMDSpeedButton9Click(Sender: TObject);

procedure Edit2KeyPress(Sender: TObject; var Key: Char);

procedure Edit3KeyPress(Sender: TObject; var Key: Char);

procedure Edit4KeyPress(Sender: TObject; var Key: Char);

procedure Edit7KeyPress(Sender: TObject; var Key: Char);

procedure Edit8KeyPress(Sender: TObject; var Key: Char);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Fvlad: TFvlad;

implementation

uses dm;

### Продолжение приложения А

{$R \*.dfm}

procedure TFvlad.FormCreate(Sender: TObject);

begin

dm1.vlad.IndexName:='FAM';

RadioGroup1.ItemIndex:=-1;

RadioGroup2.ItemIndex:=-1;

end;

procedure TFvlad.LMDSpeedButton1Click(Sender: TObject);

begin

if edit1.Text<>'' then begin

if RadioGroup1.ItemIndex=0 then

if dm1.vlad.Locate('FAM',edit1.Text,[loCaseInsensitive]) then

else MessageBox(Handle,'Нет записи с такой фамилией...','',MB\_OK);

if RadioGroup1.ItemIndex=1 then

if dm1.vlad.Locate('TEL',edit1.Text,[loCaseInsensitive]) then

else MessageBox(Handle,'Нет записи с таким телефоном...','',MB\_OK);

end

else MessageBox(Handle,'Введите значение для поиска!','',MB\_OK);

end;

procedure TFvlad.LMDSpeedButton2Click(Sender: TObject);

begin

if RadioGroup2.ItemIndex=0 then

dm1.vlad.IndexName:='FAM';

if RadioGroup2.ItemIndex=1 then

dm1.vlad.IndexName:='NAME';

if RadioGroup2.ItemIndex=2 then

dm1.vlad.IndexName:='TEL';

end;

procedure TFvlad.FormActivate(Sender: TObject);

begin

pedit.Visible:=true;

pedit.Visible:=false;

end;

procedure TFvlad.LMDSpeedButton8Click(Sender: TObject);

begin

if (Edit5.Text='') or (Edit6.Text='') or (Edit7.Text='')or (Edit8.Text='')

or (Edit2.Text='')or (Edit3.Text='') or (Edit4.Text='')then begin

MessageBox(Handle,'Возможно некоторые поля не заполнены!','Внимание',MB\_OK);

exit;

end

else begin

dm1.vlad.Edit;

dm1.vlad.FieldByName('FAM').Value:=edit2.text;

dm1.vlad.FieldByName('NAME').value:=edit3.Text;

dm1.vlad.FieldByName('OTH').value:=edit4.text;

dm1.vlad.FieldByName('ADRES').value:=edit5.text;

dm1.vlad.FieldByName('MAIL').value:=edit6.text;

dm1.vlad.FieldByName('TEL').value:=edit7.text;

dm1.vlad.FieldByName('SOT').value:=edit8.text;

dm1.vlad.Post;

### Продолжение приложения А

pedit.Visible:=false;

Edit2.Text:='';

Edit3.Text:='';

Edit4.Text:='';

Edit5.Text:='';

Edit6.Text:='';

Edit7.Text:='';

Edit8.Text:='';

end;

end;

procedure TFvlad.LMDSpeedButton4Click(Sender: TObject);

begin

pedit.Visible:=true;

edit2.text:=dm1.vlad.FieldByName('FAM').Value;

edit3.Text:=dm1.vlad.FieldByName('NAME').value;

edit4.text:=dm1.vlad.FieldByName('OTH').value;

edit5.text:=dm1.vlad.FieldByName('ADRES').value;

edit6.text:=dm1.vlad.FieldByName('MAIL').value;

edit7.text:=dm1.vlad.FieldByName('TEL').value;

edit8.text:=dm1.vlad.FieldByName('SOT').value;

end;

procedure TFvlad.LMDSpeedButton9Click(Sender: TObject);

begin

pedit.Visible:=false;

Edit2.Text:='';

Edit3.Text:='';

Edit4.Text:='';

Edit5.Text:='';

Edit6.Text:='';

Edit7.Text:='';

Edit8.Text:='';

dm1.vlad.Cancel;

end;

procedure TFvlad.Edit2KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'A'..'Z','a'..'z','а'..'я','А'..'Я',#8: key:=key

else key:=#0;

end;

end;

procedure TFvlad.Edit3KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'A'..'Z','a'..'z','а'..'я','А'..'Я',#8: key:=key

else key:=#0;

end;

end;

procedure TFvlad.Edit4KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'A'..'Z','a'..'z','а'..'я','А'..'Я',#8: key:=key

else key:=#0;

end;

end;

### Продолжение приложения А

procedure TFvlad.Edit7KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'0'..'9',#8: key:=key

else key:=#0;

end;

end;

procedure TFvlad.Edit8KeyPress(Sender: TObject; var Key: Char);

begin

case key of

'0'..'9',#8: key:=key

else key:=#0;

end;

end;

procedure TFvlad.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Edit1.Text:='';

RadioGroup1.ItemIndex:=-1;

RadioGroup2.ItemIndex:=-1;

close;

end;

end.