Тема: Разработка интерактивной системы обработки базы данных ведомости работы расчетно-кассового узла универсама

# Задание на выполнение курсовой работы

Разработать проект приложения, содержащего локальную базу данных для хранения и обработки данных заданной структуры (см. рис.1). системы обработки данных ведомости работы расчетно-кассового узла универсама.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование товара | Код товара | Количество в наличии | Количество проданного товара | Стоимость единицы товара | Общая стоимость проданного товара (В) | Остаток товара (В) |
|  |  |
|  | Код товара | Тип товара |  |

Рис. 1. Структура базы данных системы обработки данных ведомости работы расчетно-кассового узла универсама

Записи должны содержать данные различных типов.

База данных должна находиться в отдельном каталоге с именем KR, подкаталог Base.

# Возможности, состав и назначение системы

Система должна обеспечить:

1. ведение локальной базы данных, в том числе: просмотр, корректировку, удаление, дополнение базы данных;
2. использование меню для управления процессом обработки данных;
3. просмотр и корректировку данных в таблице и в окнах редактирования;
4. формирование значения вычисляемого поля (столбца);
5. отбор данных из базы данных по условию, определенному свойством Filtr компонента типа TTable;
6. сортировку данных по основному и дополнительным индексам;
7. поиск данных по заданным поисковым признакам;
8. непосредственный доступ к записям базы данных на примере формирования суммы столбца;
9. формирование и вывод во время выполнения приложения:

– графика зависимости данных, содержащихся в базе данных;

– отчета с помощью подсистемы QuickReport.

Для управления системой и решения поставленных задач система должна быть снабжена интерфейсом пользователя, включающим следующие элементы:

* систему меню для СУБД;
* элементы управления для отображения информации, хранящейся в таблицах;
* элементы управления для навигации по базе данных;
* элементы управления для ввода условий отбора записей;
* форм для ввода новых записей;
* формы для просмотра графика
* формы для просмотра и вывода на печать отчета.

# Процесс разработки интерфейса и проекта интерактивной системы

##

## 2.1 Подготовительный этап

На подготовительном этапе создания системы, удовлетворяющей требованиям технического задания, выполняются следующие действия:

1). Создается структура каталогов на жестком диске персонального компьютера для хранения файлов базы данных, проекта системы и исполняемого модуля. Для определенности будем полагать, что проектируемая система – файлы проекта и исполняемый модуль будет размещаться в папке \КР корневого каталога диска С:, а файлы базы данных – в подкаталоге \BASE данной папки.

2). Для создания проекта и структуры таблиц базы данных необходимо инсталлировать на компьютере разработчика средство управления базами данных MS Access, и средство разработки – Borland Delphi. При проектировании были использованы следующие версии программных продуктов: MS Access 2000 и Borland Delphi 6.0 с BDE 5.1

##

## 2.2 Создание структуры таблицы

Для создания заданной в техническом задании на проектирование системы структуры таблицы воспользуемся СУБД MS Access.

При запуске СУБД необходимо укать каталог, в котором будет помещен файл базы данных и имя этого файла: c:\kp\base.mdb.

Далее переходим в раздел таблицы и, с помощью конструктора, создаем две таблицы: Table1 и Table2. Состав и описание их полей приведены в табл. 1 и 2 соответственно.

После создания структуры таблиц определяем связи между ними в режиме просмотра схемы данных. При этом в качестве основной таблицы укажем Table1, поле связи T1KOD, в качестве связанной – Table2, поле связи T2KOD. Тип отношения – «один-ко-многим» (см. рис. 2).

Таблица 1. Состав и описание полей таблицы Table1 базы данных проектируемой системы

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Имя поля | Тип | Описание |
| 1 | KodZapisi | Счетчик | Автоинкрементируемое ключевое индексное поле целого типа для хранения информации о коде записи |
| 2 | Nazvanie | Текстовый | Текстовое поле длиной 50 символов для хранения информации о названии товара |
| 3 | Kod | Числовой | Длинное целое поле связи с таблицей Table2 |
| 4 | Prihod | Числовой | Числовое поле с плавающей точкой для хранения информации о количестве полученного товара |
| 5 | Rashod | Числовой | Числовое поле с плавающей точкой е для хранения информации о количестве проданного товара |
| 6 | Zena | Денежный | Числовое поле для хранения информации о стоимости единицы товара |

Таблица 2. Состав и описание полей таблицы Table2 базы данных проектируемой системы

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Имя поля | Тип | Описание |
| 1 | Kod | Счетчик | Автоинкрементируемое ключевое индексное поле целого типа для хранения информации о коде записи |
| 2 | Tip | Текстовый | Текстовое поле длиной 50 символов для хранения информации о типе товара |

## 2.3 Разработка проекта системы

До того как приступить к разработке системы, создадим в операционной системе компьютера разработчика с помощью средств администрирования баз данных псевдоним (Alias) используемой базы данных base.mdb. Это значительно упростит последующее написание кода приложения и облегчит перенос проектируемой системы в другой каталог, если в этом возникнет необходимость. Для создания псевдонима возможно использование различных средств: как встроенных в операционную систему (Инспектор «Источники данных (ODBC)»), так и поставляемых в комплекте со средой программирования Borland Delphi утилит (Administrator BDE или Database Destop). На рис. 3 показаны необходимые настройки при использования встроенного Инспектора OBDC.

Далее запускаем среду разработки приложений Borland Delphi и создаем стандартный исполняемый модуль (Application). Добавляем необходимые формы (Меню File, команда New 🡪 Form).

Сохраняем заготовку проекта в папке c:\kp. Имя файла проекта kp.dpr, имена используемых модулей: Unit1.pas…Unit5.pas, имена используемых форм: Unit1.frm…Unit5.frm.

Форма Form1 – основная форма проекта. На ней расположены основные элементы интерфейса пользователя и элементы управления. Их список представлен в табл. 3.

Таблица 3. Список элементов управления, расположенных на форме Form1

| № п/п | Имя элемента управления | Описание |
| --- | --- | --- |
| 1 | Database1 | Компонент, устанавливающий связь между базой данных base.mdb с помощью псевдонима base\_24 и приложением |
| 2 | MainMenu1 | Компонент, содержащий описание структуры меню |
| 3 | DataSource1 | Компонент, устанавливающий связь между источником данных DataBase1 и компонентом tTable1 |
| 4 | tTable1 | Компонент, устанавливающий связь между полями таблицы tTable1 источника данных DataSource1 и элементами их отображения, редактирования и навигации |
| 5 | DbGrid1 | Элемент отображения данных в таблице tTable1 |
| 6 | DataSource2 | Компонент, устанавливающий связь между источником данных DataBase2 и компонентом tTable2 |
| 7 | tTable2 | Компонент, устанавливающий связь между полями таблицы tTable2 источника данных DataSource2 и элементами их отображения, редактирования и навигации |
| 8 | DbGrid2 | Элемент отображения данных в таблице tTable2 |
| 9 | cmdNew | Командная кнопка открытия формы Form2 для ввода информации о новом товаре |
| 10 | cmdDel | Командная кнопка для удаления текущей записи и всей связанной с ней информацией |
| 11 | cmdFind | Командная кнопка вызова окна ввода информации для поиска |
| 12 | cmdChart | Командная кнопка для вызова формы Form4 с расположенным на ней компонентом DbChart1 |
| 13 | cmdReport | Командная кнопка для вызова формы Form5 с расположенными на ней компонентами QuickRep1, QRBand1…QRBand4, QRLabel1…QRLabel9, QRDBText1… QRDBText6, QRExpr1… QRExpr6, QRShape1, QR Shape2, Table1, Table2, DataSource1. |
| 14 | cmdType | Командная кнопка для вызова формы Form3 для ввода информации о типах товаров |
| 15 | RadioGroup1 | Радиокнопки, управляющие режимом сортировки |
| 16 | GroupBox1 | «Декоративный» элемент, объединяющий элементы DBLookupComboBox1 и CheckBox1 |
| 17 | DBLookupComboBox1 | Комбинированное поле для вывода условий отбора |
| 18 | CheckBox1 | Флажок, управляющий режимом фильтрации |
| 19 | StatusBar1 | Информационная строка |

Компоненты размещаются на форме посредством «перетягивания» их значков в определенное место формы из соответствующих разделов (Standart, DataAccess, DataControl, BDE, QReport) палитры компонентов среды разработки Borland Delphi.

Для всех вышеперечисленных компонентов необходимо определить их свойства, такие как геометрические размеры компонента (Width, Height), Имя (Name), Источник данных (DataSource), Отображаемые на экране надписи (Caption) и ряд других – в соответствии с требуемой задачей. Часть этих свойств удобно заполнять с помощью специализированных мастеров. Так, например, состав полей, отображаемых на экране с помощью компонента DBGrid, настраивается с помощью Редактора полей, вызываемого путем двойного щелчка по соответствующему компоненту. Контроль введенных свойств и доступ к компонентам осуществляется с помощью Окна свойств и Инспектора объектов. Размер формы и расположенных на ней компонентов выбирается исходя из наиболее часто встречающегося у потенциального пользователя разрешения экрана 800 х 600 точек, эргономических и эстетических соображений.

Форма Form2 предназначена для ввода информации о новом товаре. Ввод информации осуществляется посредством заполнения пользователем значений текстовых полей DBEdit1…3, связанных с базой данных через элементы управления DataSource, расположенные на форме Form1.

Форма Form3 предназначена для работы с таблицей типов товаров и содержит источник данных DataSource1, связанный с таблицей tTable2, расположенной на форме Form1, управляющий элемент DbGrid1 для отображения и редактирования полей таблицы, управляющий элемент DbNavigator1 для выполнения основных операций с записями в таблице: навигации, добавления, удаления и сохранения.

Форма Form4 содержит компонент DBChart1, источником данных которого являются данные из таблицы Query1. Все необходимые свойства этого компонента проще всего ввести с помощью мастера, также вызываемого двойным щелчком по компоненту.

Форма Form5 содержит отчет QuickRep1 и связанные с ним элементы: полосы данных QRBand1…QRBand4, надписи QRLabel1…QRLabel97, текстовые поля QRDBText1… QRDBText6, вычисляемые поля QRExpr1… QRExpr6, линии QRShape1, QRShape2. Источником данных формы являются связные таблицы Table1 и Table2, которые в свою очередь связаны с источником данных DataSource1.

Форма AboutBox содержит пять компонентов:

* командную кнопку OKButton, служащей для закрытия формы;
* панели Panel1 для группировки компонентов ProductName и Version;
* трех информационных надписей (ProductName и Version).

Форма Intro выполняет функции «заставки» при запуске программы и содержит два компонента: Image1 с графическим изображением (файл Intro.jpg) и Timer1, свойство которого Interval имеет значение 3000 мс, т.е. «заставка» будет отображаться на экране в течении 3 с.

Внешний вид форм проекта системы на этапе разработки приведен в прил.1

После подготовки компонентов форм необходимо приступить к написанию кода обработки событий, возникающих при работе приложения.

Код представляет собой программу на языке Object Pascal обработки событий объектов и состоит из нескольких модулей: основного (kp), и модулей обработки событий форм (Unit1…Unit7). В каждом модуле, в свою очередь, может содержаться несколько процедур, предусматривающих обработку тех или иных событий компонентов форм.

Листинги соответствующих программ приведены в прил.3, 4.

##

## 2.4 Запуск и отладка системы

После написания программного кода необходимо выполнить заключительный этап проектирования – запуск и отладку приложения. Запуск приложения на выполнение осуществляется путем выполнения команды Run (клавиша F9) среды проектирования Borland Delphi. На этом этапе происходит анализ кода на предмет наличия синтаксических ошибок и, при их отсутствии, компиляция в исполняемый модуль – kp.exe. При необходимости код программы корректируется разработчиком: исправляются допущенные синтаксические и логические ошибки, оптимизируются те или иные конструкции.

# Инструкция по эксплуатации системы

Сразу после запуска системы пользователю становится доступны главная форма системы, меню, командные кнопки, элементы навигации по базе данных и элементы ввода условий фильтрации. Внешний вид форм проекта на этапе выполнения приведен в прил. 2

В строке меню (см. Рис.П.2.1) расположены команды по управлению системой. Описание структуры меню приведено в табл. 4

Таблица 4. Структура меню системы

| Наименование раздела | Содержание раздела | Описание |
| --- | --- | --- |
| Файл | Выход | Команда выхода из системы. Во время выполнение команды выдается запрос на подтверждение и осуществляется корректный выход из системы. |
| Правка | Новый товар | Ввод информации о новом товаре |
| Удалить товар | Удаление информации о выбранном товаре |
| Поиск товара | Поиск товара по базе данных |
| Вид | Диаграмма | Просмотр сведений о продажах в графической форме |
| Отчет | Предварительный просмотр и распечатки отчета о продажах |
| Справка | Вызов справки | Вызов окна справочной системы программы |
| О программе | Вывод окна информации о разработчике |

Вспомогательная информация (подсказки по элементам управления) доступна пользователю в строке состояния.

Навигация по записям базы данных осуществляется путем выбора с помощью указателя мыши требуемого элемента в таблице товаров.

Пользователь имеет возможность изменить режим сортировки – для этого необходимо воспользоваться соответствующей радиокнопкой.

Для отбора сведений о товарах определенного типа необходимо включить режим фильтрации и указать искомый тип товара, выбрав его из списка.

Чтобы осуществить поиск информации о товаре в базе данных необходимо открыть с помощью кнопки «Поиск товара» форму ввода искомого значения (рис. .П.2.4) После нажатия кнопки «ОК» в системе будет осуществлен поиск искомых сочетаний знаков. При этом указатель будет помещен на первое найденное значение или будет выведено сообщение об отсутствии искомого значения.

Для ввода информации о новом товаре служит кнопка «Новый товар» (рис. .П.2.2). После её нажатия появляется форма ввода для внесения в базу новой записи.

Удаление информации о выбранном товаре осуществляется после нажатия кнопки «Удалить товар» (рис.П.2.3). При этом на экране появляется предупреждающее сообщение.

Вычисление общей суммы продаж по выбранному товару и остаток его на складе осуществляется автоматически.

Для просмотра сведений о продажах в графической форме служит кнопка «Диаграмма» (рис. .П.2.5).

Для работы с таблицей типов товаров необходимо нажать кнопку «Тип товара», что приведет к открытию соответствующей формы (см. рис.П.2.6).

Для предварительного просмотра и распечатки отчета о продажах служит кнопка «Отчет». В окне предварительного просмотра (рис. .П.2.7) пользователю доступны кнопки изменения масштаба изображения на экране, выбора и настройки принтера, отправки указанной страницы на печать.

# Перенос и установка системы на другой компьютер

Для нормальной работоспособности системы на другом компьютере необходимо воспользоваться программой InstallShield, поставляемой вместе с Delphi. Запустив мастер создания инсталляционного пакета и заполнив все необходимые поля карточки мастера (расположение на диске, метод установки, необходимость включения компонентов в пакет и т.п.) получаем на выходе исполняемый файл setup.exe. Этот файл необходимо передать пользователю программы.

Для установки системы на компьютере пользователя необходимо запустить указанный выше файл (setup.exe) и дождаться завершения установки. При этом будет произведено автоматическое копирование всех необходимых элементов, создан каталог KP на диске C компьютера пользователя, произведена настройка псевдонима базы данных.

Перед установкой программы необходимо:

1. Убедится в наличии установленных ОС Windows 98 или выше и администратора баз данных BDE версии 5.1 или выше.
2. Убедится в наличии установленной СУБД ACCESS 2000 или выше.
3. При необходимости установить указанные компоненты.

После завершения установки достаточно запустить из папки KP исполняемый файл kp.exe.

# Список литературы

1. Архангельский А.Я. Работа с локальными базами данных в Delphi 5. М.:Бином,2000.
2. Архангельский А.Я. Язык SQL в Delphi 5. М.: ЗАО «Издательство БИНОМ»,М.: 2000.
3. Архангельский А.Я. Object Pascal в Delphi 5. М.: ЗАО «Издательство БИНОМ», 1999.
4. Гофман В.Э., Хомоненко А.Д. Delphi 6. СПб, «БХВ-Петербург», 2002.
5. Гофман В.Э., Хомоненко А.Д.Работа с базами данных в Delphi. СПб,«Скт- Петербург», 2000.
6. Дарахвелидзе П., Марков Е. Программирование в Delphi 7. СПб, «Скт-Петербург», 2003.
7. Климова Л.М. Программирование в среде Delphi. Часть 2.Учебное пособие. М.: МГТУ ГА, 2003.
8. Фаронов В.В. Delphi 6. Учебный курс. М.: Издатель Молгачкова С.В., 2002.
9. Архангельский А. Я. Программирование в Delphi 7. М: « Издательство БИНОМ», 2003.

**Приложение**

Файл kp.dpr

program kp;

uses

Forms,

Unit1 in 'Unit1.pas' {Form1},

Unit2 in 'Unit2.pas' {Form2},

Unit3 in 'Unit3.pas' {Form3},

Unit4 in 'Unit4.pas' {Form4},

Unit5 in 'Unit5.pas' {Form5},

Unit6 in 'Unit6.pas' {Intro},

Unit7 in 'Unit7.pas' {AboutBox};

{$R \*.res}

begin

Application.Initialize;

Application.CreateForm(TForm1, Form1);

Application.CreateForm(TForm2, Form2);

Application.CreateForm(TForm3, Form3);

Application.CreateForm(TForm4, Form4);

Application.CreateForm(TForm5, Form5);

Application.CreateForm(TIntro, Intro);

Application.CreateForm(TAboutBox, AboutBox);

Application.Run;

end.

Файл unit1.pas

unit Unit1;

interface

uses

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

Dialogs, DB, Grids, DBGrids, DBTables, Menus, StdCtrls, ExtCtrls, DBCtrls,

dbcgrids, ComCtrls;

type

TForm1 = class(TForm)

Database1: TDatabase;

tTable1: TTable;

DBGrid1: TDBGrid;

DataSource1: TDataSource;

tTable2: TTable;

tTable1KodZapisi: TAutoIncField;

tTable1Nazvanie: TStringField;

tTable1Kod: TIntegerField;

tTable1Prihod: TFloatField;

tTable1Rashod: TFloatField;

tTable1Zena: TFloatField;

tTable1Stoimost: TCurrencyField;

tTable1Ostatok: TFloatField;

tTable2Kod: TAutoIncField;

tTable2Tip: TStringField;

tTable1Field111: TStringField;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

cmdNew: TButton;

cmdDel: TButton;

cmdType: TButton;

RadioGroup1: TRadioGroup;

GroupBox1: TGroupBox;

CheckBox1: TCheckBox;

DBLookupComboBox1: TDBLookupComboBox;

DataSource2: TDataSource;

StatusBar1: TStatusBar;

cmdChart: TButton;

cmdReport: TButton;

cmdFind: TButton;

N6: TMenuItem;

N7: TMenuItem;

N8: TMenuItem;

N9: TMenuItem;

N10: TMenuItem;

N11: TMenuItem;

N12: TMenuItem;

procedure tTable1CalcFields(DataSet: TDataSet);

procedure N5Click(Sender: TObject);

procedure cmdNewClick(Sender: TObject);

procedure cmdDelClick(Sender: TObject);

procedure tTable1AfterPost(DataSet: TDataSet);

procedure cmdTypeClick(Sender: TObject);

procedure tTable2AfterPost(DataSet: TDataSet);

procedure RadioGroup1Click(Sender: TObject);

procedure CheckBox1Click(Sender: TObject);

procedure DBLookupComboBox1Click(Sender: TObject);

procedure cmdChartClick(Sender: TObject);

procedure cmdFindClick(Sender: TObject);

procedure cmdReportClick(Sender: TObject);

procedure N6Click(Sender: TObject);

procedure N7Click(Sender: TObject);

procedure N8Click(Sender: TObject);

procedure N9Click(Sender: TObject);

procedure N12Click(Sender: TObject);

procedure N10Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);

procedure FormShow(Sender: TObject);

procedure N11Click(Sender: TObject);

private

{ Private declarations }

procedure DisplayHint(Sender: TObject);

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

uses Unit2, Unit3, Unit4, Unit5, Unit6, Unit7;

{$R \*.dfm}

//Вывод подсказки в строку состояния

procedure TForm1.DisplayHint(Sender: TObject);

begin

StatusBar1.SimpleText := Application.Hint;

end;

//Формирование значений вычисляемых полей

procedure TForm1.tTable1CalcFields(DataSet: TDataSet);

begin

tTable1Stoimost.AsCurrency:=tTable1Zena.AsCurrency\*tTable1Rashod.AsFloat;

tTable1Ostatok.AsFloat:=tTable1Prihod.AsFloat-tTable1Rashod.AsFloat;

end;

//Кнопка "Выход из программы"

procedure TForm1.N5Click(Sender: TObject);

begin

CloseQuery;

end;

//Кнопка "Новая запись"

procedure TForm1.cmdNewClick(Sender: TObject);

begin

Form2.ShowModal;

end;

//Кнопка "Удаление записи"

procedure TForm1.cmdDelClick(Sender: TObject);

begin

if MessageDlg('Удалить запись?',

mtConfirmation,[mbYes,mbNo],0) = mrYes then tTable1.Delete;

end;

procedure TForm1.tTable1AfterPost(DataSet: TDataSet);

begin

tTable1.Active:=False;

tTable1.Active:=True;

tTable1.Last;

end;

//Кнопка "Тип товара"

procedure TForm1.cmdTypeClick(Sender: TObject);

begin

Form3.ShowModal;

end;

procedure TForm1.tTable2AfterPost(DataSet: TDataSet);

begin

tTable2.Active:=False;

tTable2.Active:=True;

tTable2.Last;

end;

//Включение режимов сортировки

procedure TForm1.RadioGroup1Click(Sender: TObject);

begin

tTable1.Active:=False;

case RadioGroup1.ItemIndex of

0: tTable1.IndexName:='indNazvanie';

1: tTable1.IndexName:='indKod';

2: tTable1.IndexName:='';

end;

tTable1.Active:=True;

end;

//Управление фильтром

procedure FilterOn();

var v:Variant;

begin

with Form1 do

begin

if CheckBox1.Checked=False then

begin

DBLookupComboBox1.Enabled:=True;

v:=DBLookupComboBox1.KeyValue;

if VarIsNumeric(v) then tTable1.Filter:='Kod='+IntToStr(v)

else tTable1.Filter:='';

tTable1.Filtered:=True;

end

else

begin

DBLookupComboBox1.Enabled:=False;

tTable1.Filtered:=False;

end;

end;

end;

procedure TForm1.CheckBox1Click(Sender: TObject);

begin

FilterOn;

end;

procedure TForm1.DBLookupComboBox1Click(Sender: TObject);

begin

FilterOn;

end;

//Кнопка "Диаграмма"

procedure TForm1.cmdChartClick(Sender: TObject);

begin

Form4.ShowModal;

end;

//Кнопка "Поиск товара"

procedure TForm1.cmdFindClick(Sender: TObject);

var s:String;

f:Boolean;

begin

if InputQuery('Найти','', s) then

begin

f:=tTable1.Locate('Nazvanie',s,[loPartialKey,loCaseInsensitive]);

if f=False then Application.MessageBox('Значение не найдено',

'Результат поиска',MB\_ICONWARNING+MB\_OK);

end

else tTable1.Cancel;

end;

//Кнопка "Отчет"

procedure TForm1.cmdReportClick(Sender: TObject);

var bm: TBookmark;

begin

bm:=tTable1.GetBookmark;

Form5.QuickRep1.Preview;

tTable1.GotoBookmark(bm);

end;

//Работа с меню

procedure TForm1.N6Click(Sender: TObject);

begin

cmdNew.Click;

end;

procedure TForm1.N7Click(Sender: TObject);

begin

cmdDel.Click;

end;

procedure TForm1.N8Click(Sender: TObject);

begin

cmdChart.Click;

end;

procedure TForm1.N9Click(Sender: TObject);

begin

cmdReport.Click;

end;

procedure TForm1.N12Click(Sender: TObject);

begin

cmdFind.Click;

end;

procedure TForm1.N10Click(Sender: TObject);

begin

Application.HelpFile := 'kp.hlp';

Application.HelpCommand(HELP\_CONTENTS, 0);

end;

//Вывод подсказки в строку состояния

procedure TForm1.FormCreate(Sender: TObject);

begin

Application.OnHint := DisplayHint;

end;

//Выход из программы

procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);

begin

if Application.MessageBox('Вы уверены?',

'Подтверждение выхода',mb\_yesno+mb\_iconquestion)=idyes then CanClose:=True

else CanClose:=False;

end;

//Вывод заставки

procedure TForm1.FormShow(Sender: TObject);

begin

Intro.ShowModal;

end;

procedure TForm1.N11Click(Sender: TObject);

begin

AboutBox.ShowModal;

end;

end.

Листинг П. 4.2 Файл unit2.pas

unit Unit2;

interface

uses

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

Dialogs, StdCtrls, DBCtrls, DB, DBTables, Grids, DBGrids, Mask;

type

TForm2 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

cmdOK: TButton;

cmdCancel: TButton;

Label4: TLabel;

DBEdit1: TDBEdit;

DBEdit3: TDBEdit;

DBEdit4: TDBEdit;

DBLookupComboBox1: TDBLookupComboBox;

procedure cmdOKClick(Sender: TObject);

procedure cmdCancelClick(Sender: TObject);

procedure DBLookupComboBox1Click(Sender: TObject);

procedure FormActivate(Sender: TObject);

procedure DBEdit1Change(Sender: TObject);

procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses Unit1;

{$R \*.dfm}

procedure TForm2.cmdOKClick(Sender: TObject);

begin

Form1.tTable1.Post;

Form2.Close;

end;

procedure TForm2.cmdCancelClick(Sender: TObject);

begin

Form1.tTable1.Cancel;

Form2.Close;

end;

procedure TForm2.DBLookupComboBox1Click(Sender: TObject);

begin

if DBLookupComboBox1.Text<>'' then

if DBEdit1.Text<>'' then cmdOK.Enabled:=True;

end;

procedure TForm2.FormActivate(Sender: TObject);

begin

cmdOK.Enabled:=False;

Form1.tTable1.Append;

end;

procedure TForm2.DBEdit1Change(Sender: TObject);

begin

if DBLookupComboBox1.Text<>'' then

if DBEdit1.Text<>'' then cmdOK.Enabled:=True;

end;

procedure TForm2.FormCloseQuery(Sender: TObject; var CanClose: Boolean);

begin

Form1.tTable1.Cancel;

Form2.Close;

end;

end.

база данные ведомость приложение

Файл unit3.pas

unit Unit3;

interface

uses

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

Dialogs, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, DB;

type

TForm3 = class(TForm)

DBGrid1: TDBGrid;

DBNavigator1: TDBNavigator;

DataSource1: TDataSource;

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

uses Unit1;

{$R \*.dfm}

end.

Файл unit4.pas

unit Unit4;

interface

uses

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

Dialogs, TeEngine, Series, ExtCtrls, TeeProcs, Chart, DbChart, TeeFunci,

DB, DBTables, Grids, DBGrids;

type

TForm4 = class(TForm)

Query1: TQuery;

DBChart1: TDBChart;

Series1: TPieSeries;

procedure FormActivate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form4: TForm4;

implementation

uses Unit1;

{$R \*.dfm}

procedure TForm4.FormActivate(Sender: TObject);

begin

Query1.Active:=False;

Query1.Active:=True;

end;

end.

Файл unit5.pas

unit Unit5;

interface

uses

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

Dialogs, QuickRpt, ExtCtrls, DB, DBTables, QRCtrls;

type

TForm5 = class(TForm)

QuickRep1: TQuickRep;

QRGroup1: TQRGroup;

QRBand1: TQRBand;

Table1: TTable;

Table1KodZapisi: TAutoIncField;

Table1Nazvanie: TStringField;

Table1Kod: TIntegerField;

Table1Prihod: TFloatField;

Table1Rashod: TFloatField;

Table1Zena: TFloatField;

Table2: TTable;

Table2Kod: TAutoIncField;

Table2Tip: TStringField;

QRDBText1: TQRDBText;

DataSource1: TDataSource;

QRLabel3: TQRLabel;

QRLabel4: TQRLabel;

QRLabel2: TQRLabel;

QRLabel5: TQRLabel;

QRLabel6: TQRLabel;

QRLabel7: TQRLabel;

QRLabel8: TQRLabel;

QRShape1: TQRShape;

QRDBText2: TQRDBText;

QRDBText3: TQRDBText;

QRDBText4: TQRDBText;

QRDBText5: TQRDBText;

QRDBText6: TQRDBText;

QRExpr1: TQRExpr;

QRExpr2: TQRExpr;

QRBand3: TQRBand;

QRBand4: TQRBand;

QRLabel1: TQRLabel;

QRExpr3: TQRExpr;

QRExpr4: TQRExpr;

QRExpr5: TQRExpr;

QRExpr6: TQRExpr;

QRShape2: TQRShape;

QRLabel9: TQRLabel;

procedure FormActivate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form5: TForm5;

implementation

{$R \*.dfm}

procedure TForm5.FormActivate(Sender: TObject);

begin

Table1.Active:=True;

end;

end.

Файл unit6.pas

unit Unit6;

interface

uses

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

Dialogs, jpeg, ExtCtrls;

type

TIntro = class(TForm)

Image1: TImage;

Timer1: TTimer;

procedure Timer1Timer(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Intro: TIntro;

implementation

uses Unit1;

{$R \*.dfm}

procedure TIntro.Timer1Timer(Sender: TObject);

begin

Close;

end;

end.

Файл unit7.pas

unit Unit7;

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls,

Buttons, ExtCtrls;

type

TAboutBox = class(TForm)

Panel1: TPanel;

ProductName: TLabel;

Version: TLabel;

OKButton: TButton;

procedure OKButtonClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

AboutBox: TAboutBox;

implementation

{$R \*.dfm}

procedure TAboutBox.OKButtonClick(Sender: TObject);

begin

AboutBox.Close;

end;

end.