

Министерство образования и науки Российской Федерации
Пензенский Государственный университет
Факультет вычислительной техники
Кафедра «Информационно-вычислительные системы»

ТАБЛИЦЫ ГЛАВНАЯ/ПОДЧИНЕННАЯ В ПРИЛОЖЕНИЯХ

Отчет о лабораторной работе №4
по дисциплине «Базы данных»

Выполнил: ст-т гр. 18ВО1

Кравчук М.В.

Проверил: к.т.н., доцент

Долгова И. А.

1 Цель работы

Ознакомиться с принципами построения реляционной базы данных и обеспечения целостности данных в базе, а также с разработкой приложения для таблиц «Главная/Подчиненная»

2 Задание

В учебных планах института имеются факультативные дисциплины. Студенты подают заявки на изучение этих дисциплин, учет которых производит деканат факультета. Один студент может изучать несколько дисциплин. Один преподаватель может проводить занятия по нескольким дисциплинам.

Использовать:

- имя файла с БД — FACULT026;
- логин — TEAM026;
- пароль — slave026.

В базе данных деканата должна храниться следующая информация:

- о студентах - Фамилия, Номер группы, Изучаемая дисциплина;
- о дисциплинах - Название, Фамилия преподавателя, Количество часов;
- о преподавателях - Фамилия, Имя, Отчество, Ученая степень, Звание.

3 Выполнение работы

3.1 Используемые приложения

- 1) CASE-средство Open ModelSphere;
- 2) редактор «OpenOffice Writer» для создания отчёта;
- 3) Delphi Builder 2009;
- 4) GUI-оболочка «IBExpert», предназначенная для разработки и администрирования баз данных InterBase и FireBird, а также для выбора и изменения данных, хранящихся в базах.

3.2 Перечень файлов, полученных при выполнении лабораторной работы

- БД: sqledu02:C:\Users\79053\Desktop\JOB\7 семак\БД\СУБД\database for DBMS\FACULT026.FDB;
- Отчет по лабораторной работе №4;
- Папка с файлами приложения - C:\Users\79053\Desktop\JOB\7 семак\БД\СУБД\lab4.

4 Создание приложения в среде Delphi

Необходимо создать для набора данных dstPD поле подстановки. Для этого необходимо перейти на модуль данных и выделить компонент dstPD, а затем щелкнуть дважды по нему - откроется редактор полей этого набора, в котором будет показан список полей этого набора. Щелкнув правой кнопкой мыши в пределах окна редактора полей из всплывающего меню выбрать строку «New Field». После чего откроется окно для создания нового поля.

В поле Name необходимо ввести "NameDis", выбираем качестве типа String, далее устанавливаем размер поля равный 30. После этого в группе Field type выбирается радиокнопка Lookup. В группе Lookup definition в качестве значения поля Key Fields выбираем имя поля NDIS (вместо числового значения

этого поля будет извлекаться значение поля NAME набора данных dstDiscipline). В качестве значения поля Dataset выбираем имя набора данных dstDiscipline, для значения поля Lookup Keys - имя поля NDIS набора данных dstDiscipline, а для значения поля Result Field - имя результирующего поля NAME. После создания поля подстановки нажимается кнопка «Ок» редактора нового поля. После этого в редакторе полей набора, появится имя нового, созданного нами поля «NameDis» (Рисунок 1).

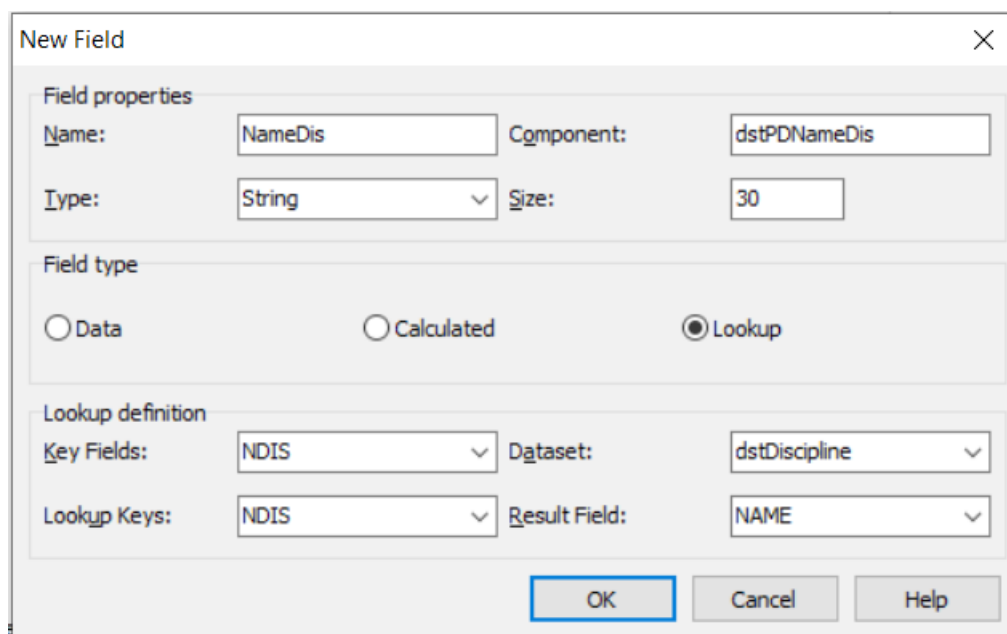


Рисунок 1 — Создание нового поля

После создания поля подстановки необходимо, для его вывода на экран создать ещё один столбец «Наименование преподаваемой дисциплины» в компоненте DBGrid1 (Рисунок 2).

Номер Преподавателя	Фамилия	Имя	Отчество	Научная степень	Научное звание
1	Долгова	Ирина	Анатольевна	кандидат наук	доцент
2	Чувькин	Борис	Викторович	Доктор наук	профессор
3	Зимин	Алексей	Петрович	ст. преп.	
5	Боброва	Вера	Васильевна	к.т.н.	доцент

№	Наименование преподаваемой дисциплины
1	Базы данных
17	ОТУ

Рисунок 2 — Добавления столбца для поля подстановки

Для изменения содержимого таблицы «Преподаватели/Дисциплины» используется способ дополнительной формы. При нажатии на кнопку «Добавить» под нижней таблицей откроется дополнительная форма fmAddPD для добавления записей в таблицу (Рисунок 3). На форму fmAddPD с вкладки Standart помещены компоненты: buOk: TButton (OK), buCancel: TButton (Отмена), Panel 1: TPanel, Panel2: TPanel, Label1: TLabel (Фамилия преподавателя), Label2: TLabel (Наименование дисциплины), DBLCNDIS: TDBLookupComboBox, DBLCNTEACH: TDBLookupComboBox.

Рисунок 3 — Форма для добавления записей в таблицу «PD»

При нажатии на кнопку «Добавить» под нижней таблицей откроется дополнительная форма fmEditPD для изменения записей в таблице (Рисунок 4). На форму fmEditPD с вкладки с вкладки Standart помещены компоненты: buOk: TButton (OK), buCancel: TButton (Отмена), Panel 1: TPanel, Panel2: TPanel, Label1: TLabel (Фамилия преподавателя), Label2: TLabel (Наименование дисциплины), DBLCNDIS: TDBLookupComboBox, DBLCNTEACH: TDBLookupComboBox.

Рисунок 4 — Форма для изменения записей в таблице «PD»

На обеих формах для добавления и изменения добавлен компонент DBLookupComboBox с вкладки: Data Controls.

Настройка свойств данного компонента представлена на рисунке 5.

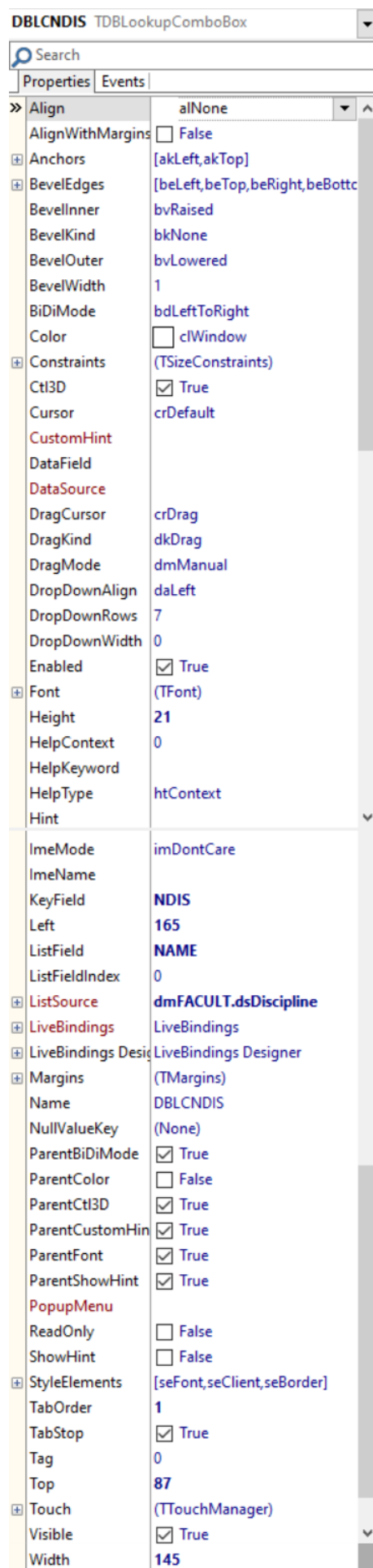


Рисунок 5 — Свойства компонента DBLookupComboBox1

Далее были разработаны обработчики событий всех форм. Результаты данных действий представлены в приложении А.

5 Описание выполнения работы

В ходе выполнения лабораторной работы было разработано приложение с возможностью ввода, редактирования и удаления данных в базе данных. Описание модулей приложения представлено в таблице 1, описание созданных обработчиков событий приведено в таблице 2, текст обработчиков представлен в приложении А.

Для связи с базой данных был разработан не визуальный модуль DM, доступ к которому имеют все модули. Схема взаимодействия всех модулей разработанного приложения представлена на рисунке 6.

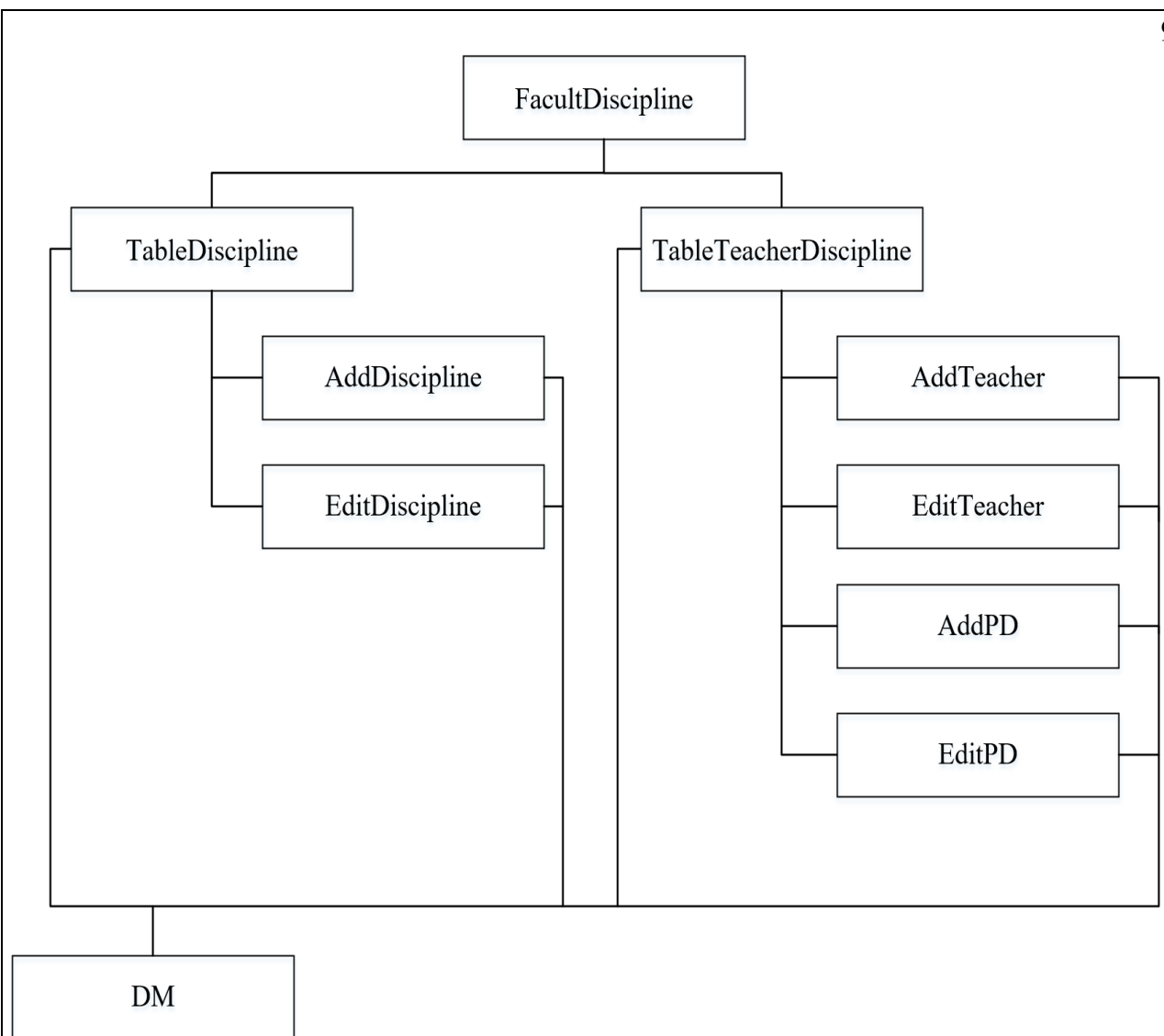


Рисунок 6 – Схема взаимодействия программных модулей

Таблица 1 — Описание модулей программы

Наименование модуля	Наименование формы	Описание
1	2	3
FacultDiscipline	fmFacultDiscipline	Отображает главную форму приложения
DM	dmFACULT	Модуль взаимодействия с базой данных
TableDiscipline	fmTableDiscipline	Отображает форму приложения с таблицей, содержащей информацию о товарах
AddDiscipline	fmAddDiscipline	Отображает форму добавления данных в таблицу «Discipline»

Продолжение таблицы 1

1	2	3
EditDiscipline	fmEditDiscipline	Отображает форму изменения данных в таблицу «Discipline»
TableTeacherDiscipline	fmTableTeacherDiscipline	Отображает форму приложения с таблицами, содержащими информацию о преподавателях и их дисциплинах
AddPD	fmAddPD	Отображает форму добавления данных в таблицу «PD»
EditPD	fmEditPD	Отображает форму изменения данных в таблицу «PD»
AddTeacher	fmAddTeacher	Отображает форму добавления данных в таблицу «TEACHER»
EditTeacher	fmEditTeacher	Отображает форму изменения данных в таблицу «TEACHER»

Таблица 2 — Описание обработчиков событий

Наименование модуля	Наименование формы	Наименование процедуры или функции	Описание
1	2	3	4
FacultDiscipline	fmFacultDiscipline	procedure TfmFacultDiscipline.FormShow(Sender: TObject);	Обработчик события отображения формы fmFacultDiscipline приложения программы.
		procedure TfmFacultDiscipline.FormClose(Sender: TObject; var Action: TCloseAction);	Обработчик события закрытия формы fmFacultDiscipline приложения.
		procedure TfmFacultDiscipline.DisciplineClick(Sender: TObject);	Процедура выполняет создание формы fmTableDiscipline и отображение её.
		procedure TfmFacultDiscipline.TablePDClick(Sender: TObject);	Процедура выполняет создание формы fmTableTeacherDiscipline и отображение её.

Продолжение таблицы 2

1	2	3	4
TableTeacherDiscipline	fmTableTeacherDiscipline	procedure TfmTableTeacherDiscipline. FormShow(Sender: TObject);	Обработчик события отображения формы приложения fmTableTeacherPD. Процедура открывает и устанавливает связь с базой данных.
		procedure TfmTableTeacherDiscipline. FormClose(Sender: TObject; var Action: TCloseAction);	Обработчик события закрытия формы fmTableTeacherPD приложения. Процедура разрывает связь с базой данных и закрывает форму.
		procedure TfmTableTeacherDiscipline.a cRefreshExecute(Sender: TObject);	Обработчик события обновления данных базы данных и отображения их на форме fmTableTeacherPD.
		procedure TfmTableTeacherDiscipline.a cQuitExecute(Sender: TObject);	Обработчик события отложенного закрытия формы fmTableTeacherPD. Форма закрывается после обработки всех сообщений.
		procedure TfmTableTeacherDiscipline.b uAddTeacherClick(Sender: TObject);	Обработчик события добавления данных в таблицу «TEACHER».
		procedure TfmTableTeacherDiscipline.b uDeleteTEACHERClick(Sen der: TObject);	Обработчик события удаления данных из таблицы «TEACHER».
		procedure TfmTableTeacherDiscipline.b uEditTeacherClick(Sender: TObject);	Обработчик события изменения данных в таблице «TEACHER».
		procedure TfmTableTeacherDiscipline.b uAddPDClick(Sender: TObject);	Обработчик события добавления данных в таблицу «PD».
		procedure TfmTableTeacherDiscipline.b uEditPDClick(Sender: TObject);	Обработчик события изменения данных в таблице «PD».
		procedure buDeletePDClick(Sender: TObject);	Обработчик события удаления данных из таблицы «PD».
AddTeacher	fmAddTeacher	procedure TfmAddTeacher.buOKClick(Sender: TObject);	Обработчик события добавления новой записи в таблицу «TEACHER».

Продолжение таблицы 2

1	2	3	4
		procedure TfmAddTeacher.FormClose(Sender: TObject; var Action: TCloseAction);	Обработчик события удаления формы fmAddTeacher из памяти компьютера.
EditTeacher	fmEditTeacher	procedure TfmEditTeacher.FormShow(Sender: TObject);	Обработчик события отображения формы редактирования данных fmEditTeacher, с полями заполненными значениями выделенной записи на главной форме.
		procedure TfmEditTeacher.buOKClick(Sender: TObject);	Обработчик события записи измененных данных в таблице «TEACHER».
		procedure TfmEditTeacher.FormClose(Sender: TObject; var Action: TCloseAction);	Обработчик события удаления формы fmEditTeacher из памяти компьютера.
AddPD	fmAddPD	procedure TfmAddPD.buOKClick(Send er: TObject);	Обработчик события добавления новой записи в таблицу «PD».
		procedure TfmAddPD.FormClose(Send er: TObject; var Action: TCloseAction);	Обработчик события удаления формы fmAddPD из памяти компьютера.
EditPD	fmEditPD	procedure TfmEditPD.FormShow(Send er: TObject);	Обработчик события отображения формы редактирования данных fmEditPD, с полями заполненными значениями выделенной записи на главной форме.
		procedure TfmEditPD.buOKClick(Send er: TObject);	Обработчик события записи измененных данных в таблице «PD».
		procedure TfmAddPD.buOKClick(Send er: TObject);	Обработчик события добавления новой записи в таблицу «PD».
TableDiscipline	fmTableDiscipline	procedure TfmTableDiscipline.FormSh ow(Sender: TObject);	Обработчик события отображения формы fmTableDiscipline приложения программы. Процедура открывает и устанавливает связь с базой данных.

Продолжение таблицы 2

1	2	3	4
		procedure TfmTableDiscipline.FormClose(Sender: TObject; var Action: TCloseAction);	Обработчик события закрытия формы fmTableDiscipline приложения. Процедура разрывает связь с базой данных и закрывает форму.
		procedure TfmTableDiscipline.acRefreshExecute(Sender: TObject);	Обработчик события обновления данных базы данных и отображения их на форме fmTableDiscipline.
		procedure TfmTableDiscipline.acQuitExecute(Sender: TObject);	Обработчик события отложенного закрытия формы fmTableDiscipline. Форма закрывается после обработки всех сообщений.
		procedure TfmTableDiscipline.buAddClick(Sender: TObject);	Обработчик события добавления данных в таблицу «Discipline».
		procedure TfmTableDiscipline.buEditClick(Sender: TObject);	Обработчик события изменения данных в таблице «Discipline».
		procedure TfmTableDiscipline.buDeleteClick(Sender: TObject);	Обработчик события удаления данных из таблицы «Discipline».
AddDiscipline	fmAddDiscipline	procedure TfmAddDiscipline.buOkClick(Sender: TObject);	Обработчик события добавления новой записи в таблицу «Discipline».
		procedure TfmAddDiscipline.FormClose(Sender: TObject; var Action: TCloseAction);	Обработчик события удаления формы fmAddDiscipline из памяти компьютера.
EditDiscipline	fmEditDiscipline	procedure TfmEditDiscipline.FormShow(Sender: TObject);	Обработчик события отображения формы редактирования данных fmEditDiscipline, с полями заполненными значениями выделенной записи на главной форме.
		procedure TfmEditDiscipline.buOKClick(Sender: TObject);	Обработчик события записи измененных данных в таблице «Discipline».
		procedure TfmEditDiscipline.FormClose(Sender: TObject; var Action: TCloseAction);	Обработчик события добавления новой записи в таблицу «Discipline».

Вывод

В результате выполнения лабораторной работы были изучены принципы построения реляционной базы данных и обеспечения целостности данных в базе данных, а также ознакомились с разработкой приложения для таблиц «Главная/Подчиненная».

ПРИЛОЖЕНИЕ А
(обязательное)
Текст программы

```
//Главная форма
```

```
unit FacultDiscipline;
```

```
interface
```

```
uses
```

```
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,  
    Vcl.Graphics,  
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.Menus,  
    Vcl.Imaging.jpeg;
```

```
type
```

```
    TfmFacultDiscipline = class(TForm)
```

```
        MainMenu1: TMainMenu;
```

```
        N2: TMenuItem;
```

```
        N3: TMenuItem;
```

```
        N4: TMenuItem;
```

```
        N5: TMenuItem;
```

```
        StudentApplic: TMenuItem;
```

```
        TablePD: TMenuItem;
```

```
        N8: TMenuItem;
```

```
        mnClose: TMenuItem;
```

```
        ListTeachers: TMenuItem;
```

```
        N10: TMenuItem;
```

```
        N11: TMenuItem;
```

```
        N12: TMenuItem;
```

```
        N13: TMenuItem;
```

```
        N14: TMenuItem;
```

```
        Image1: TImage;
```

```
        procedure TablePDClick(Sender: TObject);
```

```
        procedure DisciplineClick(Sender: TObject);
```

```
        procedure ListTeachersClick(Sender: TObject);
```

```
        procedure StudentApplicClick(Sender: TObject);
```

```
        procedure FormShow(Sender: TObject);
```

```
        procedure mnCloseClick(Sender: TObject);
```

```
    private
```

```
        { Private declarations }
```

```
    public
```

```
        { Public declarations }
```

```
end;
```

```
var
```

```
    fmFacultDiscipline: TfmFacultDiscipline;
```

```
implementation
```

```
{ $R *.dfm }
```

```
uses TableTeacherDiscipline, TableDiscipline, QueryListTeachers,  
    TableStudApplic;
```

```
procedure TfmFacultDiscipline.FormShow(Sender: TObject);
```

```
begin
```

```
    fmFacultDiscipline.Height := 550;
```



```
    fmFacultDiscipline.Width := 1090;
end;

procedure TfmFacultDiscipline.ListTeachersClick(Sender: TObject);
begin
    if (not Assigned(fmQueryListTeachers)) then
        fmQueryListTeachers := TfmQueryListTeachers.Create(Application);
    fmQueryListTeachers.Show;
end;

procedure TfmFacultDiscipline.mnCloseClick(Sender: TObject);
begin
    Close;
end;

procedure TfmFacultDiscipline.DisciplineClick(Sender: TObject);
begin
    fmTableDiscipline := TfmTableDiscipline.Create(Application);
    fmTableDiscipline.Show;
end;

procedure TfmFacultDiscipline.StudentApplicClick(Sender: TObject);
begin
    fmTableStudApplic := TfmTableStudApplic.Create(Application);
    fmTableStudApplic.Show;
end;

procedure TfmFacultDiscipline.TablePDClick(Sender: TObject);
begin
    fmTableTeacherDiscipline := TfmTableTeacherDiscipline.Create(Application);
    fmTableTeacherDiscipline.Show;
end;

end.
```

```
//Форма таблицы «Дисциплины»
```

```
unit TableDiscipline;
```

```
interface
```

```
uses
```

```
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,  
Vcl.Graphics,  
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, System.Actions, Vcl.ActnList,  
Vcl.Menus, Vcl.Grids, Vcl.DBGrids, Vcl.ComCtrls, Vcl.StdCtrls, Vcl.ExtCtrls;
```

```
type
```

```
TfmTableDiscipline = class(TForm)
```

```
Panel1: TPanel;
```

```
buAdd: TButton;
```

```
buEdit: TButton;
```

```
buDelete: TButton;
```

```
chbFiltr: TCheckBox;
```

```
edFilter: TEdit;
```

```
StatusBar1: TStatusBar;
```

```
buRefresh: TButton;
```

```
buExit: TButton;
```

```
DBGrid1: TDBGrid;
```

```
MainMenu1: TMainMenu;
```

```
mFile: TMenuItem;
```

```
mExit: TMenuItem;
```

```
mRefresh: TMenuItem;
```

```
ActionList1: TActionList;
```

```
acQuit: TAction;
```

```
acRefresh: TAction;
```

```
acQuery: TAction;
```

```
procedure buAddClick(Sender: TObject);
```

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

```
procedure buEditClick(Sender: TObject);
```

```
procedure buDeleteClick(Sender: TObject);
```

```
procedure FormShow(Sender: TObject);
```

```
procedure acRefreshExecute(Sender: TObject);
```

```
procedure acQuitExecute(Sender: TObject);
```

```
procedure DBGrid1TitleClick(Column: TColumn);
```

```
procedure chbFiltrClick(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
fmTableDiscipline: TfmTableDiscipline;
```

```
implementation
```

```
{ $R *.dfm }
```

```
uses DM, AddDiscipline, EditDiscipline;
```

```

procedure TfmTableDiscipline.acQuitExecute(Sender: TObject);
begin
    Application.Terminate;
end;

procedure TfmTableDiscipline.acRefreshExecute(Sender: TObject);
begin
    with dmFACULT do
        begin
            //Обновляем набор данных
            dstDiscipline.FullRefresh;
            //Выводим количество записей в таблице
            StatusBar1.Panels[1].Text := IntToStr(dstDiscipline.RecordCount);
        end;
    end;

procedure TfmTableDiscipline.buAddClick(Sender: TObject);
begin
    //Создаем форму Add
    fmAddDiscipline := TfmAddDiscipline.Create(Application);
    //Выполняем на этой форме добавление записи
    if fmAddDiscipline.ShowModal = mrOK then
        begin
            //Подтверждаем транзакцию
            dmFACULT.dstDiscipline.Post;
            StatusBar1.Panels[1].Text := IntToStr(dmFACULT.dstDiscipline.RecordCount);
        end
        //или откатываем ее
    else dmFACULT.dstDiscipline.Cancel;
end;

procedure TfmTableDiscipline.buDeleteClick(Sender: TObject);
begin
    with dmFACULT do
        begin
            if dstDiscipline.RecordCount= 0 then exit;
            if Application.MessageBox('Удалить текущую запись?', 'Удаление',
                MB_YESNO) = mrYes then dstDiscipline.Delete;
            StatusBar1.Panels[1].Text := IntToStr(dstDiscipline.RecordCount);
        end;
    end;

procedure TfmTableDiscipline.buEditClick(Sender: TObject);
begin
    //Проверить существуют ли записи в таблице
    if dmFACULT.dstDiscipline.RecordCount = 0 then exit;
    //Создать форму fmEdit
    fmEditDiscipline := TfmEditDiscipline.Create(Application);
    //Выполняем на этой форме изменение записи
    if fmEditDiscipline.ShowModal = mrOK then
        begin
            //Подтверждаем транзакцию
            dmFACULT.dstDiscipline.Post;

```

```

    StatusBar1.Panels[1].Text := IntToStr(dmFACULT.dstDiscipline.RecordCount);
end
//или откатываем ее
else dmFACULT.dstDiscipline.Cancel;
end;

procedure TfmTableDiscipline.chbFiltrClick(Sender: TObject);
var fn : string;
begin
    fn := "";
    if (chbFiltr.Checked) and (edFilter.Text<>") then
    begin
        //Если свойство checked установлено и есть образец для поиска
        fn := '(NAME LIKE "'+edFilter.Text+'%)';
        dmFACULT.dstDiscipline.Close;
        dmFACULT.dstDiscipline.MainWhereClause := fn;
        dmFACULT.dstDiscipline.Open;
    end
    else
    begin
        //Если свойство checked не установлено
        dmFACULT.dstDiscipline.Close;
        dmFACULT.dstDiscipline.MainWhereClause := fn;
        dmFACULT.dstDiscipline.Open;
    end;
end;

procedure TfmTableDiscipline.DBGrid1TitleClick(Column: TColumn);
var
    cn : string;
begin
    cn := Column.FieldName;
    if dmFACULT.dstDiscipline.OrderClause <> cn then
    begin
        dmFACULT.dstDiscipline.Close;
        dmFACULT.dstDiscipline.OrderClause := cn;
        dmFACULT.dstDiscipline.Open;
    end;
end;

procedure TfmTableDiscipline.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    with dmFACULT do
    begin
        //Запрос закрывается.
        dstDiscipline.Active := False;
        //Закрываем связь с базой данных
        dbFACULT.Connected := False;
    end;
    fmTableDiscipline := nil;
end;

procedure TfmTableDiscipline.FormShow(Sender: TObject);
begin

```

```
//Открыть базу данных
with dmFACULT do
begin
  //Устанавливаем связь с базой данных
  dbFACULT.Connected := True;
  //Активируем запрос
  dstDiscipline.Active := True;
  //Обновляем набор данных
  dstDiscipline.FullRefresh;
  //Выводим количество записей в таблице Phone
  StatusBar1.Panels[1].Text := IntToStr(dstDiscipline.RecordCount);
end;
end;

end.
```

//Форма «Добавить дисциплину»

unit AddDiscipline;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ExtCtrls;

type

TfmAddDiscipline = class(TForm)

Label1: TLabel;

Label2: TLabel;

Panel2: TPanel;

Panel1: TPanel;

buOk: TButton;

buCancel: TButton;

edName: TEdit;

edTIME: TEdit;

procedure buOkClick(Sender: TObject);

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

procedure buCancelClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

fmAddDiscipline: TfmAddDiscipline;

implementation

{ \$R *.dfm }

uses DM;

procedure TfmAddDiscipline.buCancelClick(Sender: TObject);

begin

//Удаляем форму из памяти компьютера

fmAddDiscipline := nil;

end;

procedure TfmAddDiscipline.buOkClick(Sender: TObject);

begin

with dmFACULT do

begin

//Включаем набор данных в режим добавления записи

dstDiscipline.Insert;

//Заносим введенные значения в набор данных

dstDiscipline.FBN('NAME').AsString := edNAME.Text;

dstDiscipline.FBN('TIME').AsString := edTIME.Text;

end;

```
end;  
procedure TfmAddDiscipline.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
    //Удаляем форму из памяти компьютера  
    fmAddDiscipline := nil;  
    Action := cafree;  
end;  
  
end.
```

//Форма «Изменить информацию о дисциплине»

unit EditDiscipline;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ExtCtrls;

type

TfmEditDiscipline = class(TForm)

Label1: TLabel;

Label2: TLabel;

Panel2: TPanel;

Panel1: TPanel;

buOk: TButton;

buCancel: TButton;

edName: TEdit;

edTIME: TEdit;

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

procedure FormShow(Sender: TObject);

procedure buOkClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

fmEditDiscipline: TfmEditDiscipline;

implementation

{ \$R *.dfm }

uses DM;

procedure TfmEditDiscipline.buOkClick(Sender: TObject);

begin

with dmFACULT do

begin

//Заносим введенные значения в набор данных

dstDiscipline.Edit;

dstDiscipline.FBN('NAME').AsString := edNAME.Text;

dstDiscipline.FBN('TIME').AsString := edTIME.Text;

end;

end;

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

begin

//Удаляем форму из памяти компьютера

fmEditDiscipline := nil;

end;


```
procedure TfmEditDiscipline.FormShow(Sender: TObject);
begin
  with dmFACULT do
  begin
    //Включаем набор данных в режим изменения записи
    dstDiscipline.Edit;
    dstDiscipline.FBN('NAME').AsString := edNAME.Text;
    dstDiscipline.FBN('TTIME').AsString := edTTIME.Text;
  end;
  edName.SetFocus;
end;

end.
```

//Форма таблиц «Преподавательские дисциплины и преподаватели»

unit TableTeacherDiscipline;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, System.Actions, Vcl.ActnList,
Vcl.StdCtrls, Vcl.Grids, Vcl.DBGrids, Vcl.ExtCtrls;

type

TfmTableTeacherDiscipline = class(TForm)
 Panel1: TPanel;
 DBGrid2: TDBGrid;
 Panel2: TPanel;
 Button1: TButton;
 DBGrid1: TDBGrid;
 Panel3: TPanel;
 buAddTeacher: TButton;
 buDeleteTeacher: TButton;
 buEditTeacher: TButton;
 ActionList1: TActionList;
 acQuit: TAction;
 acRefresh: TAction;
 Panel4: TPanel;
 buAddPD: TButton;
 buDeletePD: TButton;
 buEditPD: TButton;
 buRefresh: TButton;
 procedure FormShow(Sender: TObject);
 procedure buAddTeacherClick(Sender: TObject);
 procedure buEditTeacherClick(Sender: TObject);
 procedure buDeleteTeacherClick(Sender: TObject);
 procedure buAddPDClick(Sender: TObject);
 procedure buEditPDClick(Sender: TObject);
 procedure FormClose(Sender: TObject; var Action: TCloseAction);
 procedure acQuitExecute(Sender: TObject);
 procedure acRefreshExecute(Sender: TObject);
private
 { Private declarations }
public
 { Public declarations }
end;

var

fmTableTeacherDiscipline: TfmTableTeacherDiscipline;

implementation

{ \$R *.dfm }

uses DM, AddTeacher, EditTeacher, AddPD, EditPD;

```

procedure TfmTableTeacherDiscipline.acQuitExecute(Sender: TObject);
begin
    Application.Terminate;
end;

```

```

procedure TfmTableTeacherDiscipline.acRefreshExecute(Sender: TObject);
begin
    with dmFACULT do
    begin
        //Обновляем набор данных
        dstPD.FullRefresh;
        dstTeacher.FullRefresh;
    end;
end;

```

```

procedure TfmTableTeacherDiscipline.formShow(Sender: TObject);
begin
    //Открыть базу данных
    with dmFACULT do
    begin
        //Устанавливаем связь с базой данных
        dbFACULT.Connected := True;
        //Активируем запрос
        dstPD.Active := True;
        dstTeacher.Active := True;
        dstDiscipline.Active := True;

        //Обновляем набор данных
        dstPD.FullRefresh;
        dstTeacher.FullRefresh;
    end;
end;

```

```

procedure TfmTableTeacherDiscipline.buAddTeacherClick(Sender: TObject);
begin
    //Создаем форму Add
    fmAddTeacher := TfmAddTeacher.Create(Application);
    //Выполняем на этой форме добавление записи
    if fmAddTeacher.ShowModal = mrOK then
    begin
        //Подтверждаем транзакцию
        dmFACULT.dstTeacher.Post;
    end
    //или откатываем ее
    else dmFACULT.dstTeacher.Cancel;
end;

```

```

procedure TfmTableTeacherDiscipline.buDeleteTeacherClick(Sender: TObject);
begin
    with dmFACULT do
    begin
        if dstTeacher.RecordCount = 0 then exit;
        if Application.MessageBox('Удалить текущую запись?', 'Удаление',
            MB_YESNO) = mrYes then dstTeacher.Delete;
    end;
end;

```

```

end;
end;

procedure TfmTableTeacherDiscipline.buEditPDClick(Sender: TObject);
begin
    //Проверить существуют ли записи в таблице
    if dmFACULT.dstPD.RecordCount = 0 then exit;
    //Создать форму fmEdit
    fmEditPD := TfmEditPD.Create(Application);
    //Выполняем на этой форме изменение записи
    if fmEditPD.ShowModal = mrOK then
    begin
        //Подтверждаем транзакцию
        dmFACULT.dstPD.Post;
    end
    //или откатываем ее
    else dmFACULT.dstPD.Cancel;
end;

procedure TfmTableTeacherDiscipline.buEditTeacherClick(Sender: TObject);
begin
    //Проверить существуют ли записи в таблице
    if dmFACULT.dstTeacher.RecordCount = 0 then exit;
    //Создать форму fmEdit
    fmEditTeacher := TfmEditTeacher.Create(Application);
    //Выполняем на этой форме изменение записи
    if fmEditTeacher.ShowModal = mrOK then
    begin
        //Подтверждаем транзакцию
        dmFACULT.dstTeacher.Post;
    end
    //или откатываем ее
    else dmFACULT.dstTeacher.Cancel;
end;

procedure TfmTableTeacherDiscipline.buAddPDClick(Sender: TObject);
begin
    //Создаем форму Add
    fmAddPD := TfmAddPD.Create(Application);
    //Выполняем на этой форме добавление записи
    if fmAddPD.ShowModal = mrOK then
    begin
        //Подтверждаем транзакцию
        dmFACULT.dstPD.Post;
    end
    //или откатываем ее
    else dmFACULT.dstPD.Cancel;
end;

procedure TfmTableTeacherDiscipline.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Action := cafree;
end;
end.

```

//Форма «Добавить преподавателя»

unit AddTeacher;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ExtCtrls;

type

TfmAddTeacher = class(TForm)

Panel1: TPanel;

buOk: TButton;

buCancel: TButton;

Panel2: TPanel;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

edFam: TEdit;

edName: TEdit;

edOtch: TEdit;

cbNauchStep: TComboBox;

cbNauchZvan: TComboBox;

procedure buOkClick(Sender: TObject);

procedure buCancelClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

fmAddTeacher: TfmAddTeacher;

implementation

{ \$R *.dfm }

uses DM;

procedure TfmAddTeacher.buCancelClick(Sender: TObject);

begin

//Удаляем форму из памяти компьютера

//Action := caFree;

fmAddTeacher := nil;

end;

procedure TfmAddTeacher.buOkClick(Sender: TObject);

begin

with dmFACULT do

begin

```
//Включаем набор данных в режим добавления записи
dstTeacher.Insert;
//Заносим введенные значения в набор данных
dstTeacher.FBN('SURNAME').AsString := edFam.Text;
dstTeacher.FBN('NAME').AsString := edName.Text;
dstTeacher.FBN('OTHESTVO').AsString := edOtch.Text;
dstTeacher.FBN('ACADDEGREE').AsString := cbNauchStep.Text;
dstTeacher.FBN('TITLE').AsString := cbNauchZvan.Text;
end;
end;
end.
```

//Форма «Изменить информацию о преподавателе»

unit EditTeacher;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ExtCtrls;

type

TfmEditTeacher = class(TForm)

Panel1: TPanel;

buOk: TButton;

buCancel: TButton;

Panel2: TPanel;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

edFam: TEdit;

edName: TEdit;

edOtch: TEdit;

cbNauchStep: TComboBox;

cbNauchZvan: TComboBox;

procedure buOkClick(Sender: TObject);

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

procedure FormShow(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

fmEditTeacher: TfmEditTeacher;

implementation

{ \$R *.dfm }

uses DM;

procedure TfmEditTeacher.buOkClick(Sender: TObject);

begin

with dmFACULT do

begin

//Заносим введенные значения в набор данных

dstTeacher.Edit;

dstTeacher.FBN('SURNAME').AsString := edFam.Text;

dstTeacher.FBN('NAME').AsString := edName.Text;

dstTeacher.FBN('OTHESTVO').AsString := edOtch.Text;

dstTeacher.FBN('ACADDEGREE').AsString := cbNauchStep.Text;

```
    dstTeacher.FBN('TITLE').AsString := cbNauchZvan.Text;
end;
end;

procedure TfmEditTeacher.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    //Удаляем форму из памяти компьютера
    Action := caFree;
    fmEditTeacher := nil;
end;

procedure TfmEditTeacher.FormShow(Sender: TObject);
begin
    with dmFACULT do
    begin
        //Включаем набор данных в режим изменения записи
        dstTeacher.Edit;
        dstTeacher.FBN('SURNAME').AsString := edFam.Text;
        dstTeacher.FBN('NAME').AsString := edName.Text;
        dstTeacher.FBN('OTHESTVO').AsString := edOtch.Text;
        dstTeacher.FBN('ACADDEGREE').AsString := cbNauchStep.Text;
        dstTeacher.FBN('TITLE').AsString := cbNauchZvan.Text;
    end;
    edFam.SetFocus;
    edName.SetFocus;
    edOtch.SetFocus;
    cbNauchStep.SetFocus;
    cbNauchZvan.SetFocus;
end;

end.
```


//Форма «Добавить дисциплину преподавателю»

unit AddPD;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.DBCtrls, Vcl.StdCtrls, Vcl.ExtCtrls;

type

TfmAddPD = class(TForm)

Panel1: TPanel;

buOk: TButton;

buCancel: TButton;

Panel2: TPanel;

Label1: TLabel;

Label2: TLabel;

DBLCNTEACH: TDBLookupComboBox;

DBLCNDIS: TDBLookupComboBox;

procedure buOkClick(Sender: TObject);

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

procedure buCancelClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

fmAddPD: TfmAddPD;

implementation

{ \$R *.dfm }

uses DM;

procedure TfmAddPD.buCancelClick(Sender: TObject);

begin

//Удаляем форму из памяти компьютера

fmAddPD := nil;

end;

procedure TfmAddPD.buOkClick(Sender: TObject);

begin

```
with dmFACULT do
begin
  //Включаем набор данных в режим добавления записи
  dstPD.Insert;
  //Заносим введенные значения в набор данных
                                dstPD.FBN('NTEACH').AsInteger      :=
DBLCNTEACH.Listsource.Dataset.FieldName('NTEACH').AsInteger;
                                dstPD.FBN('NDIS').AsInteger          :=
DBLCNDIS.Listsource.Dataset.FieldName('NDIS').AsInteger;
  end;
end;

procedure TfmAddPD.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  //Удаляем форму из памяти компьютера
  fmAddPD := nil;
end;

end.
```

```
//Форма «Изменить дисциплину преподавателю»
```

```
unit EditPD;
```

```
interface
```

```
uses
```

```
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,  
    Vcl.Graphics,  
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.DBCtrls, Vcl.StdCtrls, Vcl.ExtCtrls;
```

```
type
```

```
    TfmEditPD = class(TForm)
```

```
        Panel1: TPanel;
```

```
        buOk: TButton;
```

```
        buCancel: TButton;
```

```
        Panel2: TPanel;
```

```
        Label1: TLabel;
```

```
        Label2: TLabel;
```

```
        DBLCNTEACH: TDBLookupComboBox;
```

```
        DBLCNDIS: TDBLookupComboBox;
```

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

```
        procedure FormShow(Sender: TObject);
```

```
        procedure buOkClick(Sender: TObject);
```

```
    private
```

```
        { Private declarations }
```

```
    public
```

```
        { Public declarations }
```

```
end;
```

```
var
```

```
    fmEditPD: TfmEditPD;
```

```
implementation
```

```
{ $R *.dfm }
```

```
uses DM;
```

```
procedure TfmEditPD.buOkClick(Sender: TObject);
```

```
begin
```

```
    with dmFACULT do
```

```
    begin
```

```
        //Заносим введенные значения в набор данных
```

```
        dstPD.Edit;
```

```
                                dstPD.FBN('NTEACH').AsInteger :=
```

```
DBLCNTEACH.Listsource.Dataset.FieldName('NTEACH').AsInteger;
```

```
                                dstPD.FBN('NDIS').AsInteger :=
```

```
DBLCNDIS.Listsource.Dataset.FieldName('NDIS').AsInteger;
```

```
    end;
```

```
end;
```

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

```
begin
  //Удаляем форму из памяти компьютера
  fmEditPD := nil;
end;

procedure TfmEditPD.FormShow(Sender: TObject);
begin
  with dmFACULT do
  begin
    //Включаем набор данных в режим изменения записи
    dstPD.Edit;

                                dstPD.FBN('NTEACH').AsInteger      :=
DBLCNTEACH.Listsource.Dataset.FieldName('NTEACH').AsInteger;
                                dstPD.FBN('NDIS').AsInteger        :=
DBLCNDIS.Listsource.Dataset.FieldName('NDIS').AsInteger;
  end;
  DBLCNTEACH.SetFocus;
  DBLCNDIS.SetFocus;
end;

end.
```