

Самарский государственный аэрокосмический университет  
имени академика С.П. Королева

# Создание приложений в Delphi



Самара 2003

**Министерство образования Российской Федерации  
Самарский государственный аэрокосмический университет  
имени академика С.П. Королева**

## **Создание приложений в Delphi**

**Методические указания**

## Самара 2003

Составители: *Земляной Н.С., Стенгач М.С.*

УДК 681.3.06

**Создание приложений в Delphi:** Методические указания /Самар. гос. аэрокосм. ун-т; Сост.: *Земляной Н.С., Стенгач М.С.* Самара, 2003, 20 с.

Содержится указания по созданию различных приложений в популярной интерактивной среде программирования Delphi.

Предназначены для выполнения лабораторных работ по курсу «Информатика» для студентов всех специальностей.

Составлены на кафедре "Компьютерные системы".

Печатаются по решению редакционно-издательского совета Самарского государственного аэрокосмического университета им. академика С.П. Королева.

Рецензент О. В. П а в л о в

### 1. Создание простейшего приложения

Интерактивная среда программирования Delphi позволяет создавать различные приложения для операционной системы Windows. После запуска Delphi на экране появляются четыре окна (рис.1): Главное окно, Окно формы, Окно Инспектора объектов и Окно кода программы.

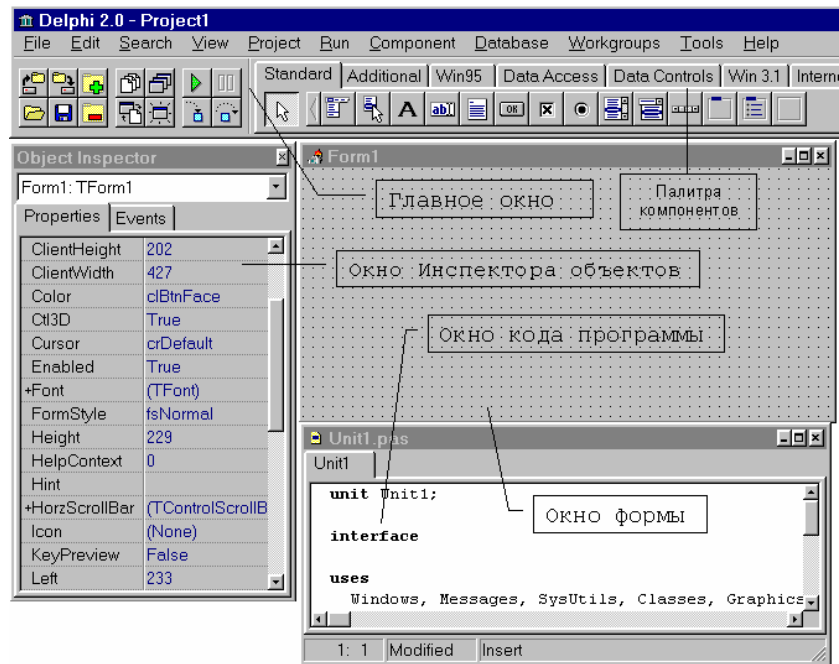


Рис.1. Окна Delphi

Главное окно Delphi содержит в себе главное меню Delphi, набор пиктографических командных кнопок и палитру компонентов. Компонентами в Delphi называются потомки класса TComponent.

Окно формы изначально пустое и содержит стандартные для Windows интерфейсные элементы – кнопки вызова системного меню, максимизации, минимизации и закрытия окна, полосу заголовка и очерчивающую рамку.

Окно инспектора объектов предназначено для изменения параметров компонентов.

Окно кода программы предназначено для создания и редактирования текста программы.

Размещать компоненты на форме очень просто. Требуется только щелкнуть на нужной вкладке палитры компонентов, затем на кнопке с пиктограммой соответствующего компонента и после этого щелкнуть на окне формы.

Создадим простейшее приложение, которое реализует следующую постановку задачи: "Ввести значения двух катетов и вычислить значение гипотенузы". Результат работы программы должен соответствовать окну, представленному на рис. 2.

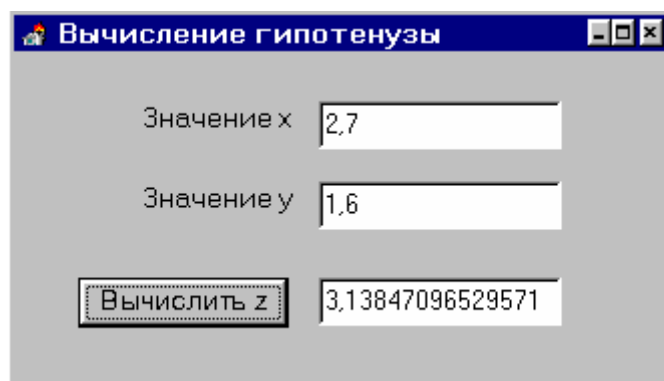


Рис.2.

Для разработки данного приложения необходимо выполнить следующие действия:

1. Выберите страницу Standard палитры компонентов.
2. Поместите указатель мыши на компонент с изображением буквы A.
3. Щелкните на кнопке, затем щелкните на вашей форме. Появится надпись Label1.
4. Для изменения стандартной надписи Label1 в строке Caption окна Инспектора объектов введите надпись Значение X.
5. Для изменения цвета и шрифта надписи щелкните мышью по свойству Font окна Инспектора объектов и с помощью кнопки в правой части строки раскройте диалоговое окно настройки шрифта.

В списке **Размер** этого окна выберите высоту шрифта, а с помощью списка **Цвет** выберите нужный цвет.

6. Аналогично пунктам 3...5 создайте надпись **Значение Y**.
7. Поместите указатель мыши на компонент **Edit** и создайте на форме три поля **Edit** (**Edit1**, **Edit2**, **Edit3**).
8. Поместите указатель мыши на компонент с изображением кнопки с надписью **OK**.
9. Щелкните по этой кнопке, а затем щелкните на форме. Появится кнопка с надписью **Button1**. Поменяйте название кнопки с помощью окна Инспектора объектов.
10. Измените название формы **Form1** на **Вычисление гипотенузы**.
11. Дважды щелкните мышью по вновь вставленному компоненту (кнопке). В ответ Delphi активизирует окно кода программы с процедурой

```
procedure TForm1.Button1Click(Sender: TObject);
```

12. Напишите в пустой процедуре фрагмент программы ввода исходных данных, вычисления и вывода на экран результатов расчета

```
var x,y,z:real;
begin
x:=strtofloat(Edit1.text);
y:=strtofloat(Edit2.text);
z:=sqrt(sqr(x)+sqr(y));
Edit3.text:=floattostr(z);
end;
```

13. Выполните команду **Save Project As...** из меню **File**, указав при этом свою рабочую папку.



В окне кода программы окончательный текст модуля будет выглядеть следующим образом:

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.DFM}
procedure TForm1.Button1Click(Sender: TObject);
var x,y,z:real;
begin
x:=strtofloat(Edit1.text);
y:=strtofloat(Edit2.text);
z:=sqrt(sqr(x)+sqr(y));
Edit3.text:=floattostr(z);
end;
end.
```

## 2. Создание приложений с компонентами страницы Standard

### 2.1. Приложение с переключателями

Для размещения на форме переключателей используются следующие компоненты страницы Standard:

1. **TCheckBox** – независимый переключатель или флаг выбора ;
2. **TRadioButton** – зависимые переключатели ;
3. **TRadioGroup** – контейнер зависимых переключателей.

Наличие выбора у двух первых компонентов определяет логическое свойство Checked:

if CheckBox1.Checked then ... – если флаг выбора установлен,

if not TRadioButton1.Checked then ... – если зависимый переключатель не выбран.

Так как переключатели TRadioButton зависимы, то на форме должно быть размещено как минимум два переключателя. Если в одном из компонентов TRadioButton свойство Checked принимает значение true, то у других компонентов оно автоматически принимает значение false.

Вместо переключателей TRadioButton удобно использовать контейнер зависимых переключателей TRadioGroup. Чтобы создать в контейнере переключатели, следует использовать свойство Items. Свойство ItemIndex по умолчанию имеет значение -1 (не выбран ни один переключатель). Если значение ItemIndex равно 0, то выбран первый зависимый переключатель, если равно 1, то второй и т. д.

Создадим приложение, которое реализует следующую постановку задачи:

1. Ввести значения двух переменных X и Y.
2. Вычислить значение Z в зависимости от предложенного выбора. Результат работы программы должен соответствовать окну, представленному на рис. 3.
3. В случае, если не включен флаг выбора, то после нажатия кнопки Вычислить Z в окне переменной Z (Edit3) должна появиться надпись 'Включи флажок!'.
4. В случае, если не выбран ни один из вариантов, то после нажатия кнопки Вычислить Z в окне переменной Z (Edit3) должна появиться надпись 'Выбери вариант!'.

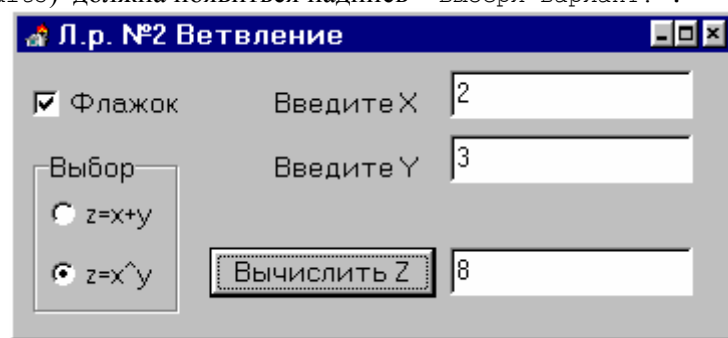


Рис. 3.

Окончательно текст модуля будет выглядеть следующим образом:

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, math;
type
  TForm1 = class(TForm)
    CheckBox1: TCheckBox;
    RadioGroup1: TRadioGroup;
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  end;
var Form1: TForm1;
implementation
procedure TForm1.Button1Click(Sender: TObject);
var x,y,z:currency;
begin
x:=strtocurr(Edit1.text);
y:=strtocurr(Edit2.text);
if CheckBox1.checked=true then
begin
if RadioGroup1.ItemIndex=0 then z:=x+y;
if RadioGroup1.ItemIndex=1 then z:=power(x,y);
Edit3.text:=currtostr(z);
if RadioGroup1.ItemIndex<0 then
Edit3.text:='Выбери вариант!';
end
else Edit3.text:='Включи флажок!';
end;
end.
```

## 2.2. Приложение с управлением значения числовой величины

Компонент **TScrollBar** страницы Standard реализует функцию управления значением числовой величины.

Основные свойства компонента:

Max: integer; - максимальное значение диапазона изменения числовой величины;

Min: integer; - минимальное значение диапазона изменения числовой величины;

Position: integer; - текущее значение числовой величины.

Создадим приложение изменения цвета формы.

Для отображения цветов на экране используется цветовая модель *RGB* (*Red, Green, Blue, Красный, зеленый, синий*). В системе координат RGB черный цвет представляется последовательностью 0, 0, 0, а белый — 255, 255, 255. Можно задать любую комбинацию из трех цифр и придумать произвольный цвет.

Для изменения значения цветов палитры RGB и, соответственно, цвета формы используем компонент **TScrollBar** с диапазоном изменения числовой величины от 0 до 255.

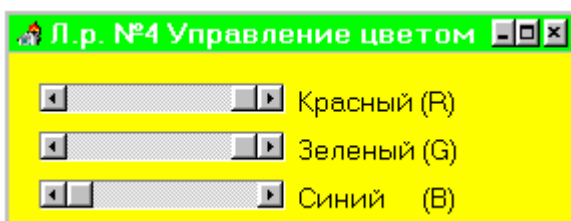


Рис. 4.

Окончательно текст модуля будет выглядеть следующим образом:

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
    ScrollBar1: TScrollBar;
    ScrollBar2: TScrollBar;
    ScrollBar3: TScrollBar;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    procedure ScrollBar1Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.DFM}
procedure TForm1.ScrollBar1Change(Sender: TObject);
begin
  Form1.Color:=RGB(ScrollBar1.Position,ScrollBar2.Position,
  ScrollBar3.Position);
end;
end.
```

Заметим, что процедура изменения значения величины `ScrollBar1Change` создается только для одного компонента `TScrollBar`. У двух других компонентов в окне инспектора объектов в разделе `Events` у события `OnChange` из списка выбирается значение `ScrollBar1Change`.

## 3. Создание приложений с компонентами страниц Additional и Win95 (Win32)

### 3.1. Редактор RTF-текста

Компонент **TRichEdit** (страница Win32 или Win95 у ранних версий) является редактором RTF-текста. Текст формата RTF хранит информацию, управляющую свойствами каждого абзаца и сменой шрифта по ходу текста. Для хранения атрибутов шрифта используются объекты класса `TTextAttributes`. Эти атрибуты распространяются на весь текст через свойство редактора `DefAttributes` или на выделенную

часть текста – через свойство SelAttributes. Атрибуты абзаца хранятся в объектах класса TParaAttributes.

Компонент **TSpeedButton** (страница Additional) представляет собой кнопку для инструментальных панелей.

Создадим приложение, реализующее работу с текстовым редактором. Редактируемый текст должен содержать такие атрибуты форматирования, как выделенный текст, наклонный текст, подчеркнутый текст.

Фрагмент работы такого приложения представлены на рис. 5.

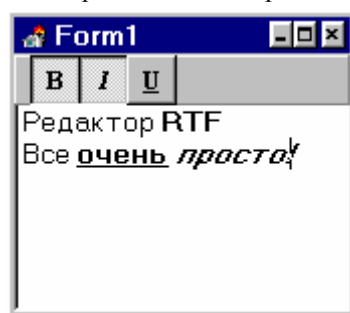


Рис. 5.

Для разработки данного приложения размещаем на форме компонент TRichEdit (страница Win95) – редактор текста в формате RTF. Свойство Align устанавливаем alBottom – заполнение нижней части формы.

В верхней части формы размещаем три кнопки TSpeedButton со страницы Additional. В свойстве Caption первой кнопки задаем латинскую букву B, у второй I, а у третьей – U. В свойстве кнопок Font задаем полужирный шрифт Times размером 10 пунктов, наклонный у второй кнопки и подчеркнутый у третьей. Свойство AllowAllUp у кнопок задается True, т.е. утопленную кнопку можно освободить повторным щелчком. Свойство GroupIndex у первой кнопки задаем равным 1, у второй равным 2, а у третьей равным 3. Таким образом, кнопки не будут принадлежать одной группе.

Окончательно окно формы приложения представлено на рис. 6.

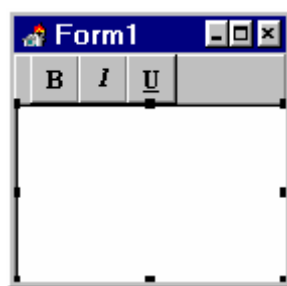


Рис. 6.

Двойным щелчком по кнопкам TSpeedButton создаем три события OnClick и формируем тексты соответствующих процедур.

Для кнопки SpeedButton1 (**B**):

```
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
  CurrText:=RichEdit1.SelAttributes;
  if SpeedButton1.Down then
    CurrText.Style := CurrText.Style + [fsBold];
  else
    CurrText.Style := CurrText.Style - [fsBold];
end;
```

Аналогично для кнопок SpeedButton2 (**I**) и SpeedButton3 (**U**) только параметр [fsBold] заменяется на [fsItalic] и [fsUnderline] соответственно.

Переменная CurrText, предназначенная для хранения атрибутов текста, описывается в разделе private:

```
CurrText: TTextAttributes;
```

На данном этапе программа будет работать следующим образом. В окне TRichEdit вводится произвольный текст. Нажатием кнопок задаются свойства текста: выделенный, наклонный и полужирный в любой комбинации.

Однако в программе пока не реализована одна очень удобная функция редактора Word – при выделении любого фрагмента текста положение кнопок TSpeedButton (нажатое или отжатое состояние)



показывает наличие соответствующих свойств у выделенного текста. Для реализации этой функции необходимо написать следующую процедуру:

```
procedure TForm1.SelectionChange(Sender: TObject);
begin
SpeedButton1.Down:=fsBold in RichEdit1.SelAttributes.Style;
SpeedButton2.Down:=fsItalic in RichEdit1.SelAttributes.Style;
SpeedButton3.Down:=fsUnderline in RichEdit1.SelAttributes.Style;
end;
```

Данная процедура описывается как метод класса TForm1 (до раздела private) и как событие OnSelectionChange компонента RichEdit1 (в Окне инспектора объектов на странице Events).

Метод SelectionChange должен активизироваться при создании формы. Двойным щелчком по форме создаем событие OnCreate:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
SelectionChange(Self);
end;
```

Запустите программу на выполнение.

## 4. Создание приложений для работы с базами данных

### 4.1. Создание базы данных типа InterBase

1. Активизируйте на своем компьютере локальный вариант IB-сервера запустив на выполнение ibserver.exe или файл приложения ibguard.exe, который находится в папке C:\Program Files\InterBaseCorp\InterBase\bin. В результате на панели задач должен появиться один из значков



2. Создайте новую базу данных (БД).
  - 2.1. Запустите утилиту InterBase Windows ISQL соответствующей командой из меню Пуск\Программы\InterBase.
  - 2.2. В появившемся окне InterBase Interactive SQL вызовите диалог Create Database из меню File (рис. 7)



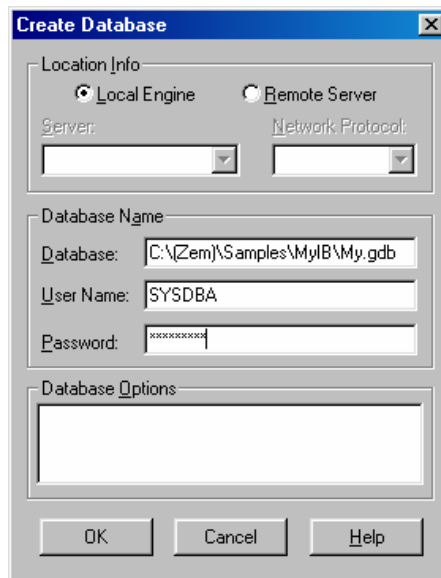


Рис. 7. Создание базы данных типа InterBase

- 2.3. В группе Location Info включите переключатель Local Engine, а в поле DataBase введите полный путь к создаваемому файлу БД C:\(Zem)\Samples\MyIB\My.gdb.
- 2.4. В поле Password введите пароль masterkey и нажмите кнопку ОК.
- 2.5. Убедитесь, что в результате этих действий в указанной папке появился пустой файл БД размером 230 КБ с заданным именем и расширением My.gdb.
3. Создайте алиас для БД My.gdb. Для этого из группы Borland Delphi5 в меню Пуск запустите программу BDE Administrator.
  - 3.1. В окне администратора BDE (рис. 8) выполните команду Object\New\INTRBASE (рис. 9), в результате которой появится новая строка INTRBASE1 на дереве алиасов, размещенном в левой части окна (см. рис. 8).
  - 3.2. Измените имя созданного алиаса INTRBASE1, заменив его на MyIBAlias.
  - 3.3. Введите полный путь к ранее созданному файлу C:\(Zem)\Samples\MyIB\My.gdb в строке SERVER NAME, которая расположена в правой части окна администратора.
  - 3.4. Выполните команду меню Object\Apply и закройте окно администратора.

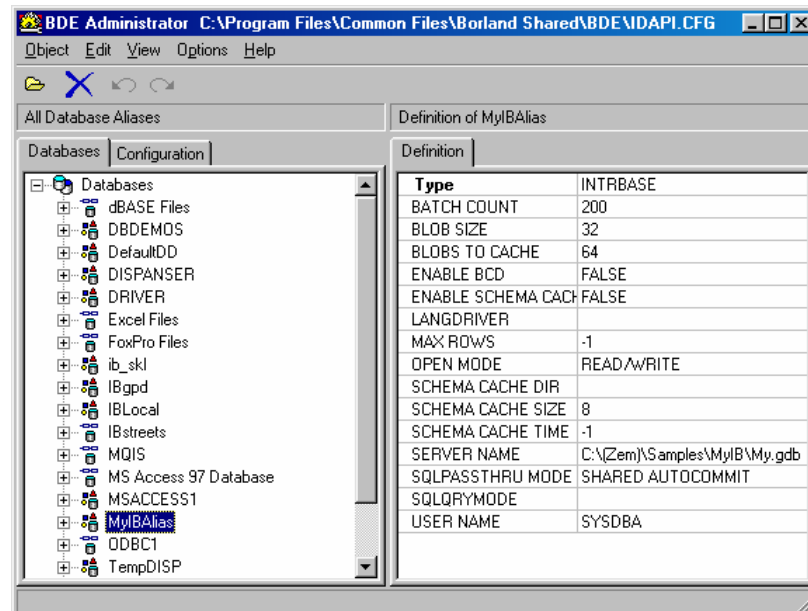


Рис. 8. Окно администратора BDE

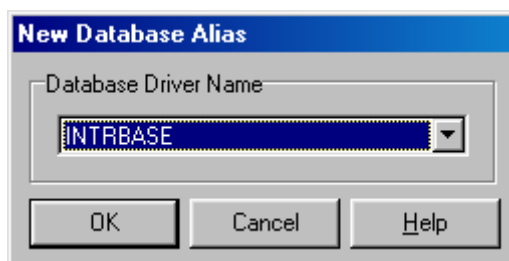


Рис. 9. Создание алиаса для базы данных

## 4.2. Создание таблицы базы данных

1. Запустите утилиту DBD (рабочий стол для работы с БД), выполнив команду Пуск\Программы\Borland Delphi5\Database Desktop.
2. Вызовите диалог Create Table командой File\New\Table (рис. 10).

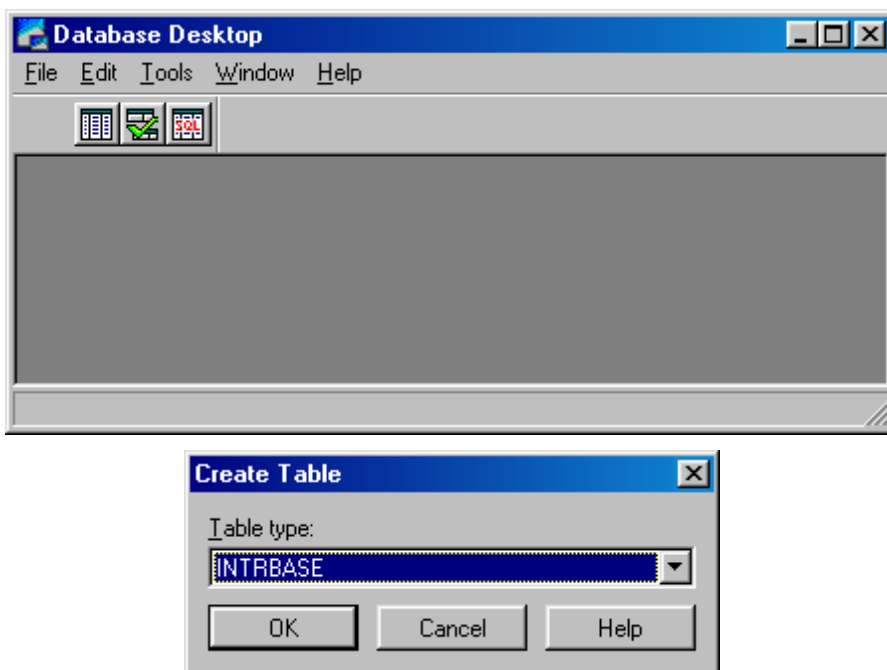


Рис. 10. Создание новой таблицы БД

- 2.1. Выберите из поля с раскрывающимся списком строку INTRBASE.
- 2.2. В появившемся окне Create INTRBASE Table создайте структуру таблицы (рис. 11), используя правую кнопку мыши для ввода типа столбца в графе Type, а поле STUDENT\_ID объявите как обязательное, включив переключатель Required field.

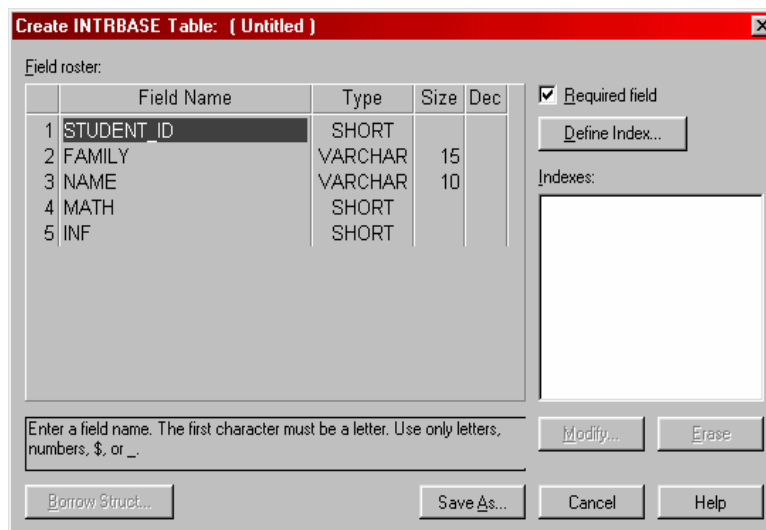


Рис. 11. Создание и формирование структуры таблицы БД

2.3. Сохраните введенные изменения нажатием кнопки Save As...

2.3.1. В поле Alias диалогового окна Save Table As выбором из списка алиасов введите MyIBAlias (рис. 12).

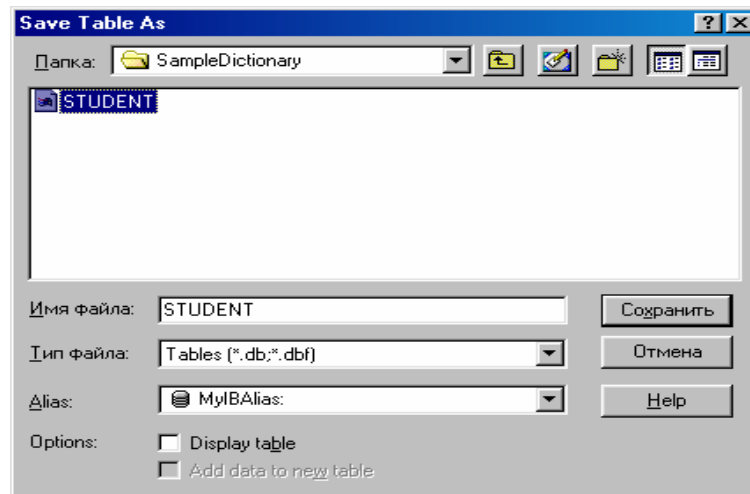


Рис. 12. Диалог сохранения таблицы в базе данных

2.3.2. В поле Имя файла диалогового окна Save Table As введите STUDENT в качестве имени для созданной таблицы .

2.3.3. Завершите процесс создания новой таблицы нажатием кнопки с надписью Сохранить.

2.3.4. Закройте окно DBD (рис. 13), в котором появилась созданная таблица STUDENT.

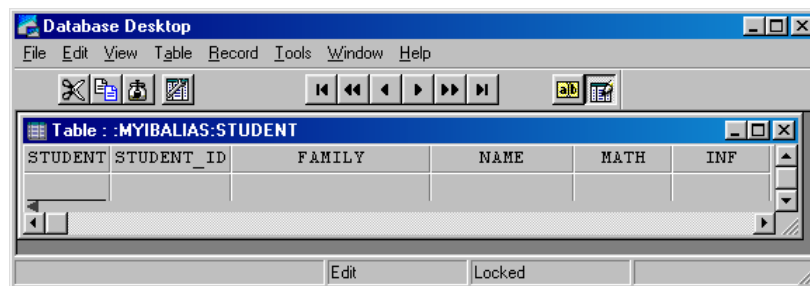


Рис. 13. Окно DBD с загруженной таблицей STUDENT

### 4.3. Создание приложения для работы с базой данных

Работа с БД возможна с помощью утилит SQL Explorer и Database Desktop, однако, более эффективна работа из специально разработанного приложения.

Для разработки таких приложений следует использовать невидимые компоненты со страницы Data Access и отображаемые в процессе работы приложения компоненты со страницы Data Controls.

Создадим приложение для работы с таблицей STUDENT базы данных MyIBAlias.

1. Разместите в верхней части формы компоненты со страницы Data Access, под ними добавьте сетку DBGrid и навигатор DBNavigator со страницы Data Controls, как показано на рис. 14.

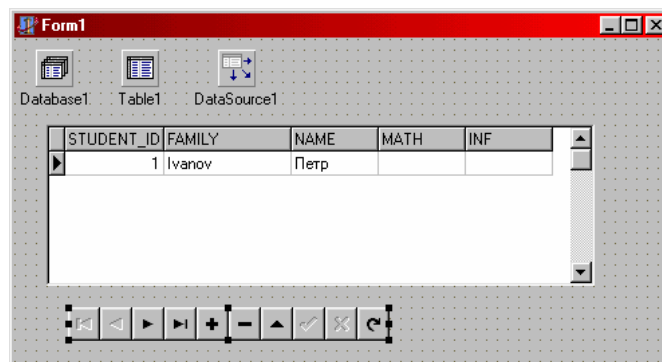


Рис. 14. Форма с компонентами DBGrid и DBNavigator для работы с БД

2. В окне Object Inspector введите произвольный идентификатор, например, DB, в строке свойства DatabaseName компонента Database1
3. Выбором из списка в окне Object Inspector задайте следующие значения других свойств:

```

Database1.AliasName:= MyIBAlias
Database1.Connected:=True
Table1.DatabaseName:=DB
Table1.TableName:=STUDENT
Table1.Active:=True
DataSource1.DataSet:=Table1
DBGrid1.DataSource:= DataSource1
DBNavigator1.DataSource:= DataSource1

```

4. Запустите приложение на выполнение и используйте кнопки навигатора для добавления (удаления) и редактирования записей, а также для перемещения в таблице.

#### 4.4. Визуализация данных с помощью компонента TDVChart

Для того чтобы построить диаграмму на основе информации из БД вызовите мастер диаграмм командой File\New\Business\TeeChartWisard и выполните его диалог. В результате будет создана форма Form2 с диаграммой. Добавьте на форму Form1 компонент Button и в обработчик события OnClick этой кнопки поместите вызов формы Form2.Show.

Запустив приложение на выполнение, правильно введите пароль в окне Database Login, вид которого представлен ниже на рис. 15. В появившемся окне щелчком мыши на кнопке с надписью Button1 вызовите окно, содержащее столбчатую диаграмму (рис. 16).

Рис. 15. Окно запроса пароля

STUDENT_ID	FAMILY	NAME	MATH	INF
1	Ivanov	Петр	5	4
2	Петров	Семен	3	3
3	Сидоров	Федор	2	3
4	Порох	Лабиринт	1	1
5	Дикий	Лука	3	3

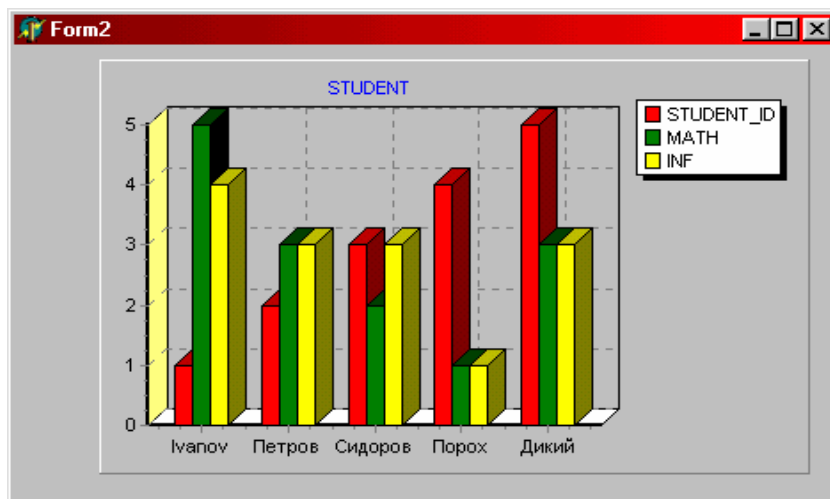


Рис. 16. Основное окно с сеткой и окно диаграммы

### Содержание

Ошибка! Недопустимо помещение оглавления в сноски, колонтитулы, примечания и надписи.

Учебное издание

СОЗДАНИЕ ПРИЛОЖЕНИЙ В DELPHI

Составители: *Земляной Николай Сергеевич*  
*Стенгач Михаил Сергеевич*

Редактор Н.С. Куприянова

Лицензия ЛР 020301 от 30.12.96 г.

Подписано в печать \_\_.\_\_.2003 г. Формат 60x84 1/16

Бумага офсетная. Печать офсетная.

Усл. печ.л. \_\_\_\_\_ Усл.кр.-отт. \_\_\_\_\_ Уч.-изд.л. \_\_\_\_\_

Тираж 100 экз. Арт. С-3(Д5)/2003. Зак.

Самарский государственный аэрокосмический университет имени академика С.П.  
Королева.

443086 Самара, Московское шоссе, 34.

ИПО СГАУ 443001 Самара, ул. Молодогвардейская, 151.