# Курсовая работа на тему “Базы данных в Delphi”

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 1](#_Toc6644194)

[ВВЕДЕНИЕ 2](#_Toc6644195)

[1. ОБЩАЯ ЧАСТЬ 9](#_Toc6644196)

[1.1. Цель разработки 9](#_Toc6644197)

[1.2. Анализ разработки 9](#_Toc6644198)

[2. СПЕЦИАЛЬНАЯ ЧАСТЬ 10](#_Toc6644199)

[2.1. Постановка задачи 10](#_Toc6644200)

[2.1.1. Назначение задачи 10](#_Toc6644202)

[2.1.2. Технико-математическая формулировка задачи 10](#_Toc6644203)

[2.1.3 Требования к программе 13](#_Toc6644205)

[2.1.3.1. Требования к функциональным характеристикам 13](#_Toc6644206)

[2.1.3.2. Требования к надежности 14](#_Toc6644208)

[2.2. Описание алгоритма 15](#_Toc6644209)

[2.3. Листинг программы 15](#_Toc6644210)

[2.4. Описание процесса отладки программы 15](#_Toc6644211)

[ЗАКЛЮЧЕНИЕ 16](#_Toc6644212)

[Список использованной литературы 17](#_Toc6644213)

[Приложение 1. Листинг программы 18](#_Toc6644216)

[Часть 1. Вычисление и вывод лучшей группы: 18](#_Toc6644217)

[Часть 2. Заполнение базы данных. 22](#_Toc6644218)

# ВВЕДЕНИЕ

С начала 60-70хх началось бурное развитие компьютеризации всех крупных промышленных центров. Ближе к 80-ым, компьютеры стали появляться и в учебных заведениях, предмету «Информатика» стали уделять огромное внимание. Стали появляться языки программирования высокого уровня такие, как Delphi, C++.

Далее пойдет речь о языке программирования Delphi и о базах данных.

В России Borland Delphi появляется в конце 1993 г. и сразу же завоевывает широкую популярность. Новые версии выходят практически каждый год. В них реализуются все новые мастера, компоненты и технологии программирования.

Действительно, процесс разработки в Delphi предельно упрощен. В первую очередь это относится к созданию интерфейса, на который уходит 80% времени разработки программы. Вы просто помещаете нужные компоненты на поверхность Windows-окна (в Delphi оно называется формой) и настраиваете их свойства с помощью специального инструмента (Object Inspector). С его помощью можно связать события этих компонентов (нажатие на кнопку, выбор мышью элемента в списке и т.д.) с кодом его обработки - и вот простое приложение готово. Причем разработчик получает в свое распоряжение мощные средства отладки (вплоть до пошагового выполнения команд процессора), удобную контекстную справочную систему (в том числе и по Microsoft API), средства коллективной работы над проектом, всего просто не перечислить. Вы можете создавать компоненты ActiveX без использования Microsoft IDL, расширять возможности web-сервера (скрипты на стороне сервера), практически ничего не зная об HTML, XML или ASP. Можно создавать распределенные приложения на базе СОМ и CORBA, Интернет- и intranet-приложения, используя для доступа к данным Borland DataBase Engine, ODBC-драйверы или Microsoft ADO. Появившаяся, начиная с Delphi 3, поддержка многозвенной технологии (multi-tiered) доступа к данным позволяет создавать масштабируемые приложения (относительно слабо зависящие от сервера БД) за счет перенесения методов обработки информации (бизнес-правил) на среднее звено.

Как уже говорилось ранее, в Delphi используется язык Object Pascal, который постоянно расширяется и дополняется Borland. Язык в полной мере поддерживает все требования, предъявляемые к объектно-ориентированному языку программирования. Как и положено строго типизированному языку, классы поддерживают только простое наследование, но зато интерфейсы могут иметь сразу несколько предков. К числу особенностей языка следует отнести поддержку обработки исключительных ситуаций (exceptions), а также перегрузку методов и подпрограмм (overload) в стиле C++. К числу удачных, на взгляд автора, относится также поддержка длинных строк в формате WideChar и AnsiChar. Последний тип (AnsiStrmg) позволяет использовать все прелести динамического размещения информации в памяти без всяких забот о ее выделении и сборке мусора Delphi делает это автоматически. Для поклонников свободного стиля программирования имеются открытые массивы, варианты и вариантные массивы, позволяющие размещать в памяти все, что душе угодно и смешивать типы данных.

Вы можете создавать свои собственные компоненты, импортировать ОСХ-компоненты, создавать <шаблоны> проектов и <мастеров>, создающих <заготовки> проектов. Мало того, Delphi предоставляет разработчику интерфейс для связи ваших приложений (или внешних программ) с интегрированной оболочкой Delphi (IDE).

Таким образом, вы можете использовать Delphi для создания как самых простых приложений, на разработку которых требуется 2-3 часа, так и серьезных корпоративных проектов, предназначенных для работы десятков и сотен пользователей. Причем для этого можно использовать самые последние веяния в мире компьютерных технологий с минимальными затратами времени и сил.

Ну, и одна из самых последних новостей от Inprise обещает, что в ближайшем будущем вы сможете переносить приложения, разработанные в Delphi, на платформу Linux. Более подробную информацию о Delphi можно получить на сайтах www.inprise.com и www.inprise.ru. Существует большое количество сайтов, посвященных Delphi, например <Королевство Delphi> - delphi.vitpc.com, Torry's Delphi Pages - www.torry.ru. Последний содержит большое число ссылок на ресурсы, связанные с Delphi. Ну а если вы уже давно знакомы с этим продуктом, то на сайте www.brainbench.com можно бесплатно протестироваться в качестве программиста Delphi 3 и получить по почте сертификат.

Мощность и гибкость Delphi при работе с базами данных основана на низкоуровневом ядре - процессоре баз данных Borland Database Engine (BDE). Его интерфейс с прикладными программами называется Integrated Database Application Programming Interface (IDAPI). В принципе, сейчас не различают эти два названия (BDE и IDAPI) и считают их синонимами. BDE позволяет осуществлять доступ к данным как с использованием традиционного record-ориентированного (навигационного) подхода, так и с использованием set-ориентированного подхода, используемого в SQL-серверах баз данных. Кроме BDE, Delphi позволяет осуществлять доступ к базам данных, используя технологию (и, соответственно, драйверы) Open DataBase Connectivity (ODBC) фирмы Microsoft. Но, как показывает практика, производительность систем с использованием BDE гораздо выше, чем оных при использовании ODBC. ODBC драйвера работают через специальный “ODBC socket”, который позволяет встраивать их в BDE.

Все инструментальные средства баз данных Borland - Paradox, dBase, Database Desktop - используют BDE. Все особенности, имеющиеся в Paradox или dBase, “наследуются” BDE, и поэтому этими же особенностями обладает и Delphi.

**Алиасы**

аблицы сохраняются в базе данных. Некоторые СУБД сохраняют базу данных в виде нескольких отдельных файлов, представляющих собой таблицы (в основном, все локальные СУБД), в то время как другие состоят из одного файла, который содержит в себе все таблицы и индексы (InterBase). Например, таблицы dBase и Paradox всегда сохраняются в отдельных файлах на диске. Каталог, содержащий dBase .DBF файлы или Paradox .DB файлы, рассматривается как база данных. Другими словами, любой каталог, содержащий файлы в формате Paradox или dBase, рассматривается Delphi как единая база данных. Для переключения на другую базу данных нужно просто переключиться на другой каталог. Как уже было указано выше, InterBase сохраняет все таблицы в одном файле, имеющем расширение .GDB, поэтому этот файл и есть база данных InterBase.

Удобно не просто указывать путь доступа к таблицам базы данных, а использовать для этого некий заменитель - псевдоним, называемый алиасом. Он сохраняется в отдельном конфигурационном файле в произвольном месте на диске и позволяет исключить из программы прямое указание пути доступа к базе данных. Такой подход дает возможность располагать данные в любом месте, не перекомпилируя при этом программу. Кроме пути доступа, в алиасе указываются тип базы данных, языковый драйвер и много другой управляющей информации. Поэтому использование алиасов позволяет легко переходить от локальных баз данных к SQL-серверным базам (естественно, при выполнении требований разделения приложения на клиентскую и серверную части).

Для создания алиаса запустите утилиту конфигурации BDE (программу bdeadmin.exe), находящуюся в каталоге, в котором располагаются динамические библиотеки BDE.

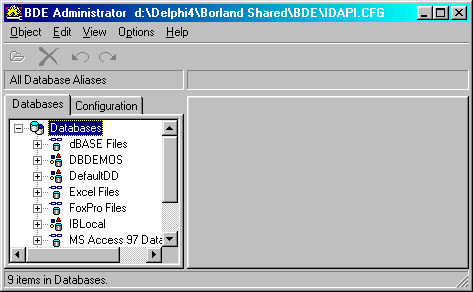


Рис. 1: Главное окно утилиты конфигурации BDE

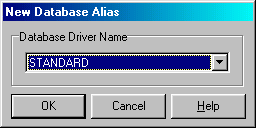


Рис. 2: В диалоговом окне добавления нового алиаса можно указать тип базы данных

Главное окно утилиты настройки BDE имеет вид, изображенный на рис.1. Для создания алиаса выберите в меню “Object” пункт “New”. В появившемся диалоговом окне выберите имя драйвера базы данных. Тип алиаса может быть стандартным (STANDARD) для работы с локальными базами в формате dBase или Paradox или соответствовать наименованию SQL-сервера (InterBase, Sybase, Informix, Oracle и т.д.).

После создания нового алиаса следует дать ему имя. Это можно сделать с помощью подпункта “Rename” меню “Object”. Однако просто создать алиас не достаточно. Вам нужно указать дополнительную информацию, содержание которой зависит от типа выбранной базы данных. Например, для баз данных Paradox и dBase (STANDARD) требуется указать лишь путь доступа к данным, имя драйвера и флаг ENABLE BCD, который определяет, транслирует ли BDE числа в двоично-десятичном формате (значения двоично-десятичного кода устраняют ошибки округления):

|  |  |
| --- | --- |
| TYPE | STANDARD |
| DEFAULT DRIVER | PARADOX |
| ENABLE BCD | FALSE |
| PATH | c:\users\data |

SQL-сервер InterBase и другие типы баз данных требуют задания большого количества параметров, многие из которых можно оставить установленными по умолчанию.

**Системная информация утилиты настройки BDE**

Итак, мы познакомились с наиболее важной возможностью утилиты настройки BDE - созданием и редактированием алиасов, определяющих параметры доступа к базам данных. Однако, утилита настройки BDE позволяет специфицировать не только алиасы, но и драйверы для доступа к базам данных, а также различную системную информацию, составляющую операционное окружение этих самых алиасов.

Рассмотрим, например, системную информацию драйвера PARADOX:

NET DIR. Параметр содержит расположение каталога сетевого управляющего файла. Он нужен для того, чтобы обратиться к таблице PARADOX на сетевом диске.

VERSION. Номер версии драйвера.

TYPE. Тип драйвера.

LANGDRIVER. Языковой драйвер, определяющий множество допустимых символов.

BLOCK SIZE. Размер блока на диске, используемого для запоминания одной записи.

FILL FACTOR. Содержит процент от блока на текущем диске. Параметр нужен для создания индексных файлов.

LEVEL. Параметр определяет тип формата таблицы, используемой для создания временных таблиц.

STRICTINTEGRTY. Параметр использования ссылочной целостности. Если он равен TRUE, то вы не можете изменить таблицу с ссылочной целостностью, а если FALSE, то можете, но рискуете нарушить целостность данных.

Как уже отмечалось выше, утилита настройки BDE сохраняет всю конфигурационную информацию в файле IDAPI.CFG. Этот файл с предустановленными ссылками на драйверы и некоторыми стандартными алиасами создается при установке Delphi. Кроме того, он создается при установке файлов редистрибуции BDE (т.е. когда Вы переносите BDE и SQL Links на другие компьютеры). (1)

# 1. ОБЩАЯ ЧАСТЬ

### 1.1. Цель разработки

Целью создания данного программного продукта является выявление лучшей группы колледжа по оценкам аттестационной или экзаменационной сессии.

### 1.2. Анализ разработки

Данная программа универсальна, так как может применяться в любом среднем специальном или высшем учебном заведении. Программа работает с Базами данных и вероятность ошибки в ней практически исключена.

# 2. СПЕЦИАЛЬНАЯ ЧАСТЬ

### 2.1. Постановка задачи

# Написать на языке программирования Delphi программу, определяющую лучшую группу колледжа по оценкам аттестационной и экзаменационной сессии.

Задача разработана в соответствии с ГОСТом 19.201-78 (ст. СЭВ 1627-79).

### 2.1.1. Назначение задачи

Определение лучшей группы колледжа путем вычисления среднего арифметического оценок по каждой группе и сравнение полученного результата с результатами остальных групп.

### 2.1.2. Технико-математическая формулировка задачи

Входными данными в этой программе являютсябазы данных

stud.db с полями:

* Zachetka –номера зачетки
* Student – данные о студенте
* Grup – группа
* Kurs – Курс

sessia.db с полями:

* Zachetka – номер зачетки
* Kodpr – код предмета
* Sessia – Тип сессии
* Ocenka – оценка

predmet.db с полями:

* kodpr – код премета
* predmet – название предмета

созданные с помощью компонента для работы с базами данных Borland Delphi Database Desktop. Для создания использовался драйвер баз данных Paradox 7.0. Для администрирования этих баз данных, в Borland Delphi Administrator был создан Alias (Псевдоним) kurs.

Выходными данными является название лучшей группы. Лучшая группа определяется следующим образом:

По полям базы данных формируется массив оценок одной группы, затем программа считает среднее арифметическое и сохраняет результат в массив с именами групп. После прохождения программой всех групп элементы массива с названиями групп сравниваются, группа имеющая наиболее высокий показатель считается лучшей группой.

Лучшая группа колледжа вычисляется по среднему арифметическому.

Арифметическое среднее, число (), получаемое делением суммы нескольких чисел (a1, a2, ..., an) на их число (n):



Например, А. с. чисел 3, 5, 7 равно (3 + 5 + 7)/3 = 5.

*(2)*

Арифметическое среднее, нескольких величин равны их сумме, деленной на их число. Если n измерений или наблюдений, одинаково заслуживающих доверия, дают n значений, a1, a2.-то вероятнейшее значение измеряемой величины есть А. среднее чисел a.

(3)

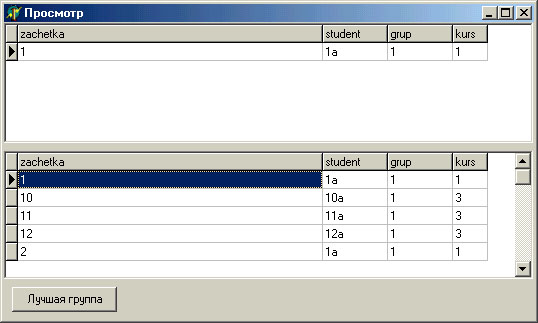


Рис. 3. Общий вид окна вычисления

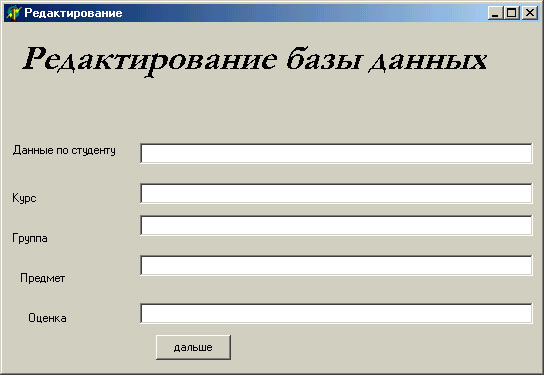


Рис. 4. Общий вид окна редактирования БД

# Блок-схема программы

1

начало

2

ввод

данных

3

вычисление

среднего

арифметического

4

ЦИКЛ

5

6

Сравнение

Лучшая

группа

7

Конец

1. Начало программы;
2. Просмотр очередного поля БД;
3. Вычисление среднего арифметического по оценкам каждой группы
4. Цикл от 1 до количества групп
5. Сравнение резултатов групп
6. Выявление лучшей группы
7. Выход из программы

### 2.1.3 Требования к программе

### 2.1.3.1. Требования к функциональным характеристикам

Созданная программа должна обеспечить вычисление и вывод лучшей группы колледжа.

# Структурная схема

Управляющая программа

Вычисление и вывод лучшей Редактирование базы данных

группы колледжа

### 2.1.3.2. Требования к надежности

Надежность программы зависит от работы всей системы и от условий ее хранения. Если программа записана гибком магнитном диске, то ее надежность зависит от правильного хранения дискеты. В этом случае рекомендуется создать копию дискеты.

В качестве постоянного хранения программы следует использовать жесткий магнитный диск, а для распространения программы, установку программы лучше записывать на CD-ROM.

2.1.3.3. требования к составу и параметрам технических средств

Программа была выполнена в оперционной системе Windows ME, она совместима с опеационными системами Windows 98, Windows XP, Windows 95, Windows ME, Windows 2000.

Для работы данной программы используется ПК имеющий следующий характеристики:

* Процессор не ниже Pentium 100
* Емкость ОЗУ не ниже 64 МБ
* Разрядность 32 бит
* Жесткий магнитный диск
* Диск 3,5
* CD-ROM

### 2.2. Описание алгоритма

1. Начало
2. Инициализация окна приложения
3. Ввод данных в программу
4. Обработка данных
5. Выявление лучшей группы
6. Если дата обновления БД, позднее установленного – открытие окна редактирования БД
7. Выход из программы

### 2.3. Листинг программы

Текст программы приводится в **Приложении 1**

В программе использованы два Unit

Unit1 – определение лучшей группы

Unit2 – редактирование БД

### 2.4. Описание процесса отладки программы

В результате тестового запуска программы не наблюдалось никаких ошибок и сбоев. Программа работает корректно и полностью отвечает поставленной задаче.

Для запуска программы необходимо из меню «Пуск», раздел «Программы», выбрать «Колледж», «Колледж».

После запуска программы, если дата не просрочена, запускается окно приложения, с помощью которого можно получить результат о лучшей группе, иначе выводится окно приложения редактирования базы данных.

# ЗАКЛЮЧЕНИЕ

Данный курсовой проект является актуальным и отвечает предъявленым к нему требованиям.Была разработана и написана, на языке программирования высокого уровня Borland Delphi 5.0, с использованием баз данных, программа, позволяющая определить лучшую группу колледжа по среднему баллу аттестационной и экзаменационной сессии.

# Список использованной литературы

# Delphi 5.0, учебный курс, Фараонов В.В., ISBN 5-8952-020-4, 400 с.

1. Большая Советская Энциклопедия
2. [Малый энциклопедический словарь Брокгауза и Ефрона](http://encycl.yandex.ru/yandsearch?enc_abc=%C0&rpt=encyc&how=enc_abc_rev&encpage=brokminor)
3. Delhphi 4.0, Дарахвелидзе П.Г., Марков Е. П. 1998, 816 с.

# Сайт <http://delhpi.mastak.ru>

# Приложение 1. Листинг программы

### Часть 1. Вычисление и вывод лучшей группы:

unit Unit1;

interface

uses

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

StdCtrls, Grids, DBGrids, Db, DBTables;

type

TForm1 = class(TForm)

Table1: TTable;

DataSource1: TDataSource;

Table2: TTable;

DataSource2: TDataSource;

DBGrid1: TDBGrid;

DBGrid2: TDBGrid;

Button1: TButton;

Label2: TLabel;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure FormShow(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

uses Unit2;

{$R \*.DFM}

procedure TForm1.Button1Click(Sender: TObject);

var

t, i, k, j, f: integer;

tmp: string;

gr: array of string;

tt: array [1..5, 1..5] of extended;

begin

table2.moveby(1);

i:=1;

for i:=1 to 3 do //3 это кол-во курсов

begin

for j:=1 to 2 do //2 кол-во групп на курсе

begin

table1.Filtered:=true;

tmp:tmp+gr[j];

table1.Filter:='kurs=1 and grup='+ tmp;

table2.Filtered:=true;

table2.Filter:='sessia=1';

k:=table2.RecordCount;

t:=0;

table2.moveby(1);

for f:=1 to k do

begin

t:=t+table2.fieldbyname('ocenka').value;

table2.next;

end;

tt[1, 1]:=t/k;

//table1.Filtered:=false;

end;

end;

label1.caption:=(inttostr(t));

end;

procedure TForm1.Button2Click(Sender: TObject);

var

i:integer;

sum:extended;

begin

table2.filter:='grup=2';

for i:=1 to Table2.recordcount do

begin

Table2.moveby(i);

sum:=sum + strtoint(Table2.Fieldbyname('ocenka').value);

end;

sum:=sum/Table2.recordcount;

//abel1.caption:=floattostr(sum)+ label1.caption;

end;

procedure TForm1.FormShow(Sender: TObject);

begin

Form2.show;

end;

end.

### Часть 2. Заполнение базы данных.

unit Unit2;

interface

uses

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

StdCtrls;

type

TForm2 = class(TForm)

Label1: TLabel;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Edit5: TEdit;

Button1: TButton;

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

{$R \*.DFM}

end.