

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 08.02.2021 16:52:35
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabb75e945df4a4851fda56d089

1

МИНОБРАЗОВАНИЯ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра защиты информации и систем связи

УТВЕРЖДАЮ

проректор по учебной работе

О. Г. Локтионова

2015 г.

РАЗРАБОТКА ПРИЛОЖЕНИЙ ДЛЯ УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ НА ЯЗЫКЕ DELPHI

Методические указания по выполнению
курсового проекта по дисциплине
«Технологии и методы программирования»,
для студентов
направления подготовки бакалавров
090900.62

Курск 2015

УДК 004.432 : 004.65(075)

Составитель: К.А. Тезик

Рецензент

Кандидат технических наук, доцент Т. И. Лапина

Разработка приложений для управления базами данных на языке Delphi: методические указания по выполнению лабораторных работ / Юго-Зап. гос. ун-т; сост.: К. А. Тезик, Курск, 2015. 33 с.: 6 ил., Библиогр.: с. 33.

Содержат варианты заданий для курсового проекта, указания к разработке программ управления базами данных на языке Delphi.

Методические указания соответствуют требованиям программы по направлению подготовки бакалавров: информационная безопасность.

Предназначены для студентов направления подготовки бакалавров 090900.62 дневной и заочной форм обучения.

Текст печатается в авторской редакции

Подписано в печать 23.09.15 Формат 60x84 1/16.
Усл. печ. л. 4,2. Уч. – изд. л. 4,4. Тираж 100 экз. Заказ. 790 Бесплатно.
Юго - Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Введение

Система Delphi завоевала себе репутацию эффективного средства разработки приложений для баз данных, то есть программ, обслуживающих электронные хранилища информации.

BDE – исторически первая технология, используемая в Delphi для разработки приложений баз данных. Она представляет собой совокупность динамических библиотек и драйверов, обеспечивающих доступ к данным. Для доступа к локальной БД процессор баз данных BDE использует стандартные драйверы, которые позволяют работать с форматами БД dBase, Paradox, FoxPro, а также с текстовыми файлами.

Технология Microsoft ActiveX Data Objects (ADO) представляет собой универсальный механизм доступа к различным источникам данных из приложений баз данных. Основу технологии ADO составляет использование набора интерфейсов общей модели объектов COM, описанных в спецификации OLE DB. При применении технологии ADO приложение БД может использовать данные из электронных таблиц, таблиц локальных и серверных баз данных, XML – файлов. В соответствии с терминологией ADO, любой источник данных (базу данных, файл, электронную таблицу) называют хранилищем данных. Приложение взаимодействует с хранилищем данных с помощью провайдера. Для каждого типа данных используется свой провайдер ADO. Провайдер обеспечивает обращение к данным хранилища с запросами, интерпретацию возвращаемой служебной информации и результатов выполнения запросов для передачи их приложению.

Альтернативный вариант разработки информационно-поисковой системы заключается в использовании компонента StringGrid (таблицы строк). Центральным свойством данного компонента является свойство Cells, представляющее собой двумерный массив строк. Данное свойство позволяет работать с содержимым ячейки таблицы, как с элементом двумерного массива.

Содержимое ячеек сохраняется только на время сеанса работы с программой. Поэтому необходимо организовать сохранение данных в файле и при каждом сеансе работы с программой записывать данные из файла в массив.

Данное учебно-методическое пособие предназначено для студентов 2 курса факультета Фундаментальной и прикладной информатики направления подготовки бакалавров «Информационная безопасность - 090900.62 »

Оно представляет собой руководство для выполнения курсового проекта по дисциплине «Технологии и методы программирования». Методическое пособие содержит 22 варианта заданий курсового проекта и примеры выполнения типового задания двумя способами:

- с помощью команд работы с внешним файлом на диске и массива строк;
- на основе управления базой данных.

1. Задание на курсовой проект

Направление курсовой работы

Разработка информационно-поисковой системы на языке программирования (название языка программирования) по предметной области (название предметной области)

Пример конкретной темы:

Разработка информационно-поисковой системы на языке программирования Delphi 7 по предметной области “библиотека”.

Задание

Разработать компьютерную программу, реализующую функции информационно-поисковой системы двумя способами:

1) **На основе использования массива строк и файла.**

Программа должна выполнять функции ввода и редактирования информации в массив строк, а также запись информации из массива в файл и считывание информации из файла в массив.

Программа должна выполнять поиск информации по заданному значению определенного поля.

Наиболее подготовленным студентам рекомендую дополнительно решить задачу защиты информации. Дело в том, что если

пользователь нажмет кнопку “Запись в файл” не считав предварительно информацию из файла, то информация будет потеряна.

2) **На основании управления базой данных.** Студент по своему усмотрению может выбрать технологию BDE или технологию ADO. Программа должна выполнять функции ввода и редактирования информации в режиме таблицы или в режиме карточки (отдельной записи).

Программа должна выполнять поиск информации по заданному значению определенного поля (на выбор студента либо в режиме таблицы, либо в режиме карточки).

Программа должна выполнять сортировку по значению определенного поля в режиме таблицы.

Программа должна формировать отчет и выводить его на печать (на выбор студента либо в режиме таблицы, либо в режиме карточки)

Варианты заданий:

1. Предметная область: зачетная книжка студента. Поля записи: дисциплина, оценка, дата сдачи, преподаватель. Поиск и сортировку записей выполнить по полю дисциплина.

2. Предметная область: преподаватели вуза. Поля записи: фамилия, должность, ученая степень, кафедра. Поиск и сортировку записей выполнить по полю фамилия.

3. Предметная область: библиотека. Поля записи: тематический раздел, автор, книга, шифр. Поиск записей выполнить по полю книга, а сортировку по полю автор.

4. Предметная область: склад. Поля записи: товар(материал), цена, количество, дата изготовления. Поиск и сортировку записей выполнить по полю товар(материал).

5. Предметная область: страны мира. Поля записи: страна, столица, площадь, население. Поиск и сортировку записей выполнить по полю страна.

6. Предметная область: касса железнодорожного вокзала. Поля записи: пункт отправления, время отправления, пункт прибытия, время прибытия, рейс (№ поезда). Поиск и сортировку записей выполнить по полю пункт отправления.

7. Предметная область: фирма по ремонту ЭВМ. Поля записи: устройство ЭВМ, технические характеристики, цена, количество. Поиск и сортировку записей выполнить по полю устройство ЭВМ

8. Предметная область: паспортный стол. Поля записи: фамилия, № и серия паспорта, дата выдачи, адрес. Поиск и сортировку записей выполнить по полю фамилия.

9. Предметная область: телефонный справочник. Поля записи: тематический раздел, организация (абонент), телефон, адрес. Поиск и сортировку записей выполнить по полю организация (абонент).

10. Предметная область: учебные заведения Курска. Поля записи: учебное заведение, адрес, телефон, директор. Поиск и сортировку записей выполнить по полю учебное заведение.

11. Предметная область: аптека. Поля записи: лекарство, дата изготовления, срок годности, цена, количество. Поиск записей выполнить по полю лекарство, а сортировку по полю цена.

12. Предметная область: спортивный комплекс. Поля записи: фамилия спортсмена, название секции(вид спорта), спортивный разряд, фамилия тренера. Поиск записей выполнить по полю

фамилия спортсмена, а сортировку по полю название секции (вид спорта).

13. Предметная область: отдел кадров. Поля записи: фамилия, отдел(подразделение), должность, адрес, телефон. Поиск и сортировку записей выполнить по полю фамилия.

14. Предметная область: бухгалтерия. Поля записи: фамилия, год, месяц, зарплата. Поиск записей выполнить по полю фамилия, а сортировку по полю зарплата.

15. Предметная область: туризм. Поля записи: пансионат (санаторий, турбаза), город(поселок), номера, телефон. Поиск записей выполнить по полю пансионат, а сортировку по полю город.

16. Предметная область: факультеты и специальности ВУЗа. Поля записи: факультет, специальность, формы обучения, оплата за коммерческое обучение. Поиск и сортировку записей выполнить по полю специальность

17. Предметная область: юридическая консультация. Поля записи: фамилия клиента, фамилия юриста, дата, время, оплата за услугу. Поиск записей выполнить по полю фамилия клиента, а сортировку по полю оплата за услугу.

18. Предметная область: тестирование студентов. Поля записи: фамилия, группа, дисциплина, дидактические единицы, средний балл.

Поиск записей выполнить по полю фамилия, а сортировку по полю средний балл.

19. Предметная область: города России. Поля записи: город, дата основания (год), отрасли промышленности (предприятия), достопримечательности. Поиск записей выполнить по полю город, а сортировку по полю дата основания.

20. Предметная область: больница. Поля записи: фамилия больного, дата поступления, номер палаты, фамилия врача. Поиск и сортировку записей выполнить по полю фамилия больного.

21. Предметная область: лаборатории ВУЗа (НИИ). Поля записи: название лаборатории, номер (аудитории), фамилия заведующего лабораторией, количество ЭВМ, количество принтеров. Поиск записей выполнить по полю название лаборатории, а сортировку по полю количество ЭВМ

22. Предметная область: кураторы ВУЗа. Поля записи: фамилия куратора, кафедра (где он работает), номер группы, фамилия старосты, количество студентов(в группе). Поиск записей выполнить по полю номер группы, а сортировку по полю количество студентов.

Содержание курсового проекта

- Титульный лист
- Задание

Разработать информационно-поисковую систему по предметной области “название предметной области“ на языке высокого уровня Delphi двумя методами:

- 1) на основе использования массива строк;
- 2) на основе управления базой данных.

- Цель курсового проекта

изучить методы разработки информационно-поисковых систем на языке высокого уровня Delphi, разработать блок-схему алгоритма метода поиска информации по заданному значению определенного поля, создать структурные схемы программ, составить и протестировать программы на языке высокого уровня Delphi, предложить методику защиты информации (и реализовать ее в

программе), организовать сортировку записей в базе данных, сформировать отчет, в который выводится на печать информация из базы данных.

1. Введение

(здесь можно дать краткую характеристику языка программирования Delphi и описать его возможности, что нового появилось в современных версиях Delphi)

2. Теоретическая часть

2.1. Описание метода решения информационно- поисковых задач в Delphi на основе использования массива строк.

2.2. Описание методов разработки информационно-поисковых систем в Delphi на основе управления базами данных

(здесь желательно описать хотя бы две технологии, например BDE и ADO)

3. Практическая часть (разработка программы на основе использования массива строк)

3.1 Описание функций, процедур и переменных, используемых в программе (Здесь желательно предложить структурную схему программы)

3.2. Текст программы на ЯВУ

3.3. Описание интерфейса программы

3.4 Блок-схема алгоритма поиска информации

3.5 Описание алгоритма защиты информации

(Необходимо на методическом, алгоритмическом уровнях внести предложения по защите информации, а реализация этих предложений в программе по желанию и возможностям студентов.)

4. Практическая часть (разработка программы на основе управления базой данных)

4.1 Описание функций, процедур и переменных, используемых в программе
(здесь желательно предложить структурную схему программы)

4.2. Текст программы на ЯВУ

4.3. Описание интерфейса программы

4.4 Описание алгоритмов поиска и сортировки информации

4.5 Формирование отчета

(здесь желательно приложить пример отчета)

4.6 Описание алгоритма защиты от несанкционированного доступа к базе данных.

(Необходимо на методическом, алгоритмическом уровнях внести предложения по защите информации от несанкционированного доступа, а реализация этих предложений в программе по желанию и возможностям студентов. Продумать вопрос, чтобы доступ к информации осуществлялся через использование логина и пароля)

5. Заключение

(Здесь выводы о том, что сделано и достигнуто в работе, какие результаты получены, предложения по дальнейшему совершенствованию программы, например использование базы данных в локальной сети, когда сама база данных находится на сервере, какими путями, какими технологиями это можно сделать в Delphi)

6. Список использованной литературы

2. Разработка информационно-поисковой системы с помощью массива строк

Предлагаю вашему вниманию методические рекомендации для первого варианта разработки информационно-поисковой сис-

темы с помощью массива строк. Информация записывается из матрицы в файл и из файла в матрицу с помощью создания в процессе выполнения программы внешнего файла на диске. При этом после записи информации, содержащейся в ячейке матрицы, в файл записывается специальный символ, указывающий окончание информации в данной ячейке. Это может быть любой символ, который не используется при наборе текста записей, например (!, ?, *, #).

При чтении информации из файла сначала происходит запись в строку символов. При чтении информации из строки в матрицу чтение информации в текущие ячейки происходит **по подстрокам с помощью команды COPY**.

При этом признаком окончания подстроки является специальный символ. После чтения подстроки происходит смена ячейки матрицы.

Последовательность выполнения работы:

1. Перенесите на форму компоненты и в свойствах компонентов выполните надписи в соответствии с рис 1.

Здесь ЗАПИСЬ В ФАЙЛ, ЗАПИСЬ В МАТРИЦУ, ВВОД НАЗВАНИЙ ПОЛЕЙ, ОК – надписи на кнопках (компонент Button), число записей вводится в строку ввода (компонент Edit), информация из файла записывается в массив строк (компонент StringGrid).

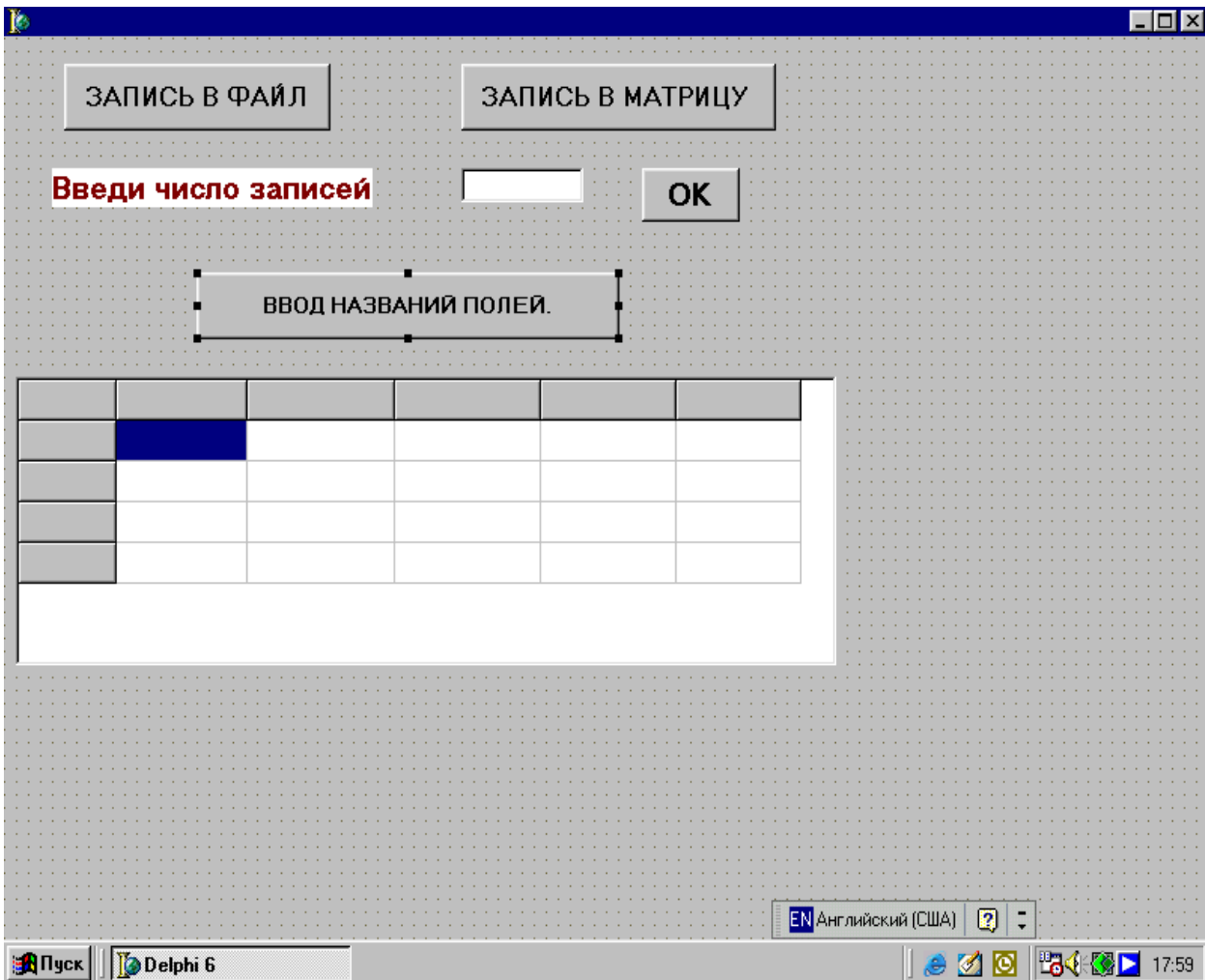


Рис1. Расположение элементов на форме.

2. Для определения количества строк в матрице и их автоматической нумерации необходимо выполнить следующие действия:

а) Выполните щелчок по кнопке **OK**.

б) Между операторами **begin** и **end** запишите текст программы:

```
n:=StrToInt(edit1.Text);
StringGrid1.RowCount:= n+1;
for j:=1 to n+1 do
begin
StringGrid1.Cells[0,j]:=IntToStr(j);
end;
```

Примечание: здесь i – номер столбца, j – номер строки, n – количество записей в матрице.

3. Для записи названий полей в матрицу необходимо выполнить следующие действия:

а) Выполните щелчок по кнопке **ВВОД НАЗВАНИЙ ПОЛЕЙ**

б) Между операторами **begin** и **end** напишите текст программы:

```
Stringgrid1.Cells[0,0]:= 'N записи';
Stringgrid1.Cells[1,0]:= 'Фамилия';
Stringgrid1.Cells[2,0]:= 'Курс';
Stringgrid1.Cells[3,0]:= 'Факультет';
Stringgrid1.Cells[4,0]:= 'Специальность';
Stringgrid1.Cells[5,0]:= 'Группа';
```

4. Для записи информации из массива строк в файл выполните следующие действия:

а) Выполните щелчок по кнопке **ЗАПИСЬ В ФАЙЛ**.

б) Между операторами **begin** и **end** запишите текст программы:

```
Rewrite(f);
For j:=1 to n do
begin
for i:=1 to 5 do
begin
write(f,stringgrid1.cells[i,j]);
write(f,'!');
end;
end;
write(f,'#');
closefile(f);
```

5. Для записи информации из файла в массив строк выполните следующие действия:

а) Выполните щелчок по кнопке **ЗАПИСЬ В МАТРИЦУ**.

б) Между операторами **begin** и **end** запишите текст программы:

```

nn:=1;
reset(f);
read(f,p);
for j:=1 to n do
begin
for i:=1 to 5 do
begin
kz:=0;
while p[nn]<>'!' do
begin
kz:=kz+1;
nn:=nn+1;
end;
stringgrid1.cells[i,j]:=copy(p,nn-kz,kz);
nn:=nn+1;
if p[nn]='#' then exit;
end;
end;

```

6) В программе необходимо установить связь между файловой переменной и файлом на диске. Выполните щелчок **по форме**. Между операторами **begin** и **end** запишите оператор:

```
AssignFile(f,'f');
```

7) Объявите переменные, используемые в программе и укажите их типы:

```

var
Form1: TForm1;
i,j,n,nn,kz: integer;
p:string;
f:textfile;

```

ВНИМАНИЕ!

В программе не предусмотрена защита информации. Если пользователь программы случайно нажмет кнопку **запись в файл**, предварительно не считав информацию из файла, то произойдет запись пустого файла и информация будет уничтожена. Предлагаю

студентам самостоятельно доработать этот вопрос, но не ставлю это обязательным условием для успешной защиты курсовой работы.

Также можно предложить студентам проявить творчество в следующих вопросах:

1) вместо того, чтобы в программе определять количество строк в матрице, можно сделать процедуру, которая добавляет в матрицу новую пустую строку;

2) поиск информации можно организовать таким образом, чтобы найденная строка выделялась маркером, а не выводилась в строки вывода Label.

3) Можно предложить альтернативный вариант записи из файла в матрицу, при котором информация будет считываться по-символьно непосредственно в файл, минуя промежуточную структуру данных – строку.

В курсовой работе можно предложить студентам **использовать две формы**. Пусть при запуске программы загружается дополнительная форма-заставка, на которой следует сделать надписи, поясняющие тему курсовой работы, например:

**Информационно-поисковая система
по предметной области «институт»
выполнил студент Иванов А. С.**

Также по своему усмотрению, студент может на данной форме расположить какой-либо рисунок или свою фотографию или может быть окно со справочной информацией о Delphi или какой-либо информацией о себе.

На форме - заставке следует расположить кнопку **Начать работу** или может быть **Войти в систему**. По нажатию этой кнопки будет происходить загрузка основной формы проекта, предназначенной для работы с информационно-поисковой системой.

Организовать работу с двумя формами не сложно. Для того, чтобы добавить форму в проект, необходимо использовать команду **File \ New \ Other**, затем выбрать форму. При этом к проекту будет присоединена новая форма **Form2** и соответствующий ей

модуль, например **Unit2**. Для того, чтобы работа в программе начиналась именно с этой формы, необходимо использовать команду **Project \ Options**, в открывшемся окне перейти на вкладку **Forms** и в списке **Main form** выбрать нужную форму, то есть **Form2**. Для того, чтобы по нажатию соответствующей кнопки происходил переход к основной форме проекта, необходимо по обработчику события нажатия кнопки между операторами **begin** и **end** добавить оператор

Form1. Show;

Кроме того, необходимо сделать ссылку на главный модуль. Для этого в модуле **Unit2** в предложении **Uses** необходимо дописать:

Implementation

Uses Unit1;

3. Разработка информационно-поисковой системы на основе управления базами данных

Предлагаю вашему вниманию методические рекомендации для второго варианта разработки информационно-поисковой системы на основе управления базами данных. Предлагаю студентам выполнить самостоятельно четыре лабораторные работы, которые с методической точки зрения являются базовыми для выполнения курсового проекта.

Лабораторная работа № 1

ПРОСМОТР ДЕМОНСТРАЦИОННОЙ БАЗЫ ДАННЫХ

Цель работы: Разработать приложение, предназначенное для просмотра демонстрационной базы данных, поставляемой вместе с Delphi.

Указания к работе

1. Перенесите на форму компонент Table со страницы BDE. Основные свойства компонента Table – DatabaseName (база данных) и TableName (таблица базы данных). Они указывают, с какой базой данных и с какой таблицей в ней мы хотим связаться.

2. Нажмите кнопку со стрелочкой справа от свойства DatabaseName в инспекторе объектов и выберите из выпадающего списка DBDEMOS – одну из демонстрационных баз данных, поставляемых с Delphi. В свойстве Table Name выберите таблицу biolife. Это таблица, содержащая информацию о рыбах.

3. Выберите поля базы данных с помощью редактора полей. Для этого выполните двойной щелчок по компоненту Table1. Вы увидите пустое поле этого редактора. Щелкните правой кнопкой мыши и из всплывающего меню выберите раздел Add fields (добавить поля). Выберите поля Category –класс, Common_Name – семейство, Species Name – вид, Length (см) –длина, Notes – описание, Graphic – изображение). Для этого щелкните мышью по данным полям, держа нажатой клавишу Ctrl.

4. В свойствах объектов, соответствующих выделенным полям выберите DisplayLabel. В этом свойстве запишите русские названия полей. Они будут отображаться в окне нашего приложения.

5. Переместите на форму компоненты, отображающие содержание полей таблицы. Для этого захватите поле в окне редактора полей курсором мыши и перетащите его на форму. На форме автоматически создадутся компоненты отображения данных. Обратите внимание: создан компонент (один на все поля) – DataSource1. Он связывает набор данных Table1 с компонентами отображения данных.

6. Повторите операцию перетаскивания для каждого поля и расположите компоненты примерно так, как показано на рис. 2.

Form1

ИЗОБРАЖЕНИЕ

КЛАСС
Triggerfish

СЕМЕЙСТВО
Clown Triggerfish

ВИД
Ballistoides conspicillum

ДЛИНА(СМ)
50

ОПИСАНИЕ

Also known as the big spotted triggerfish. Inhabits outer reef areas and feeds upon crustaceans and mollusks by crushing them with powerful teeth. They are voracious eaters, and divers report seeing the clown triggerfish devour beds of pearl oysters.

EN Английский (США)

Пуск Delphi 6 19:18

Рис 2. Форма к лабораторной работе №1

7. Выделите на форме компонент Table1. В свойстве Active установите значение true.

8. Выделите на форме компонент DBMemo. В свойстве ScrollBars установите значение ssVertical – вертикальная.

9. Перенесите на форму компонент DBNavigator со страницы DataControl. Он необходим для просмотра записей базы данных. В его основном свойстве **DataSource** установите свойство **DataSource 1**.

Лабораторная работа №2

РАЗРАБОТКА ИНФОРМАЦИОННО-ПОИСКОВОЙ СИСТЕМЫ С ПОМОЩЬЮ ТЕХНОЛОГИИ VDE

Цель работы: Разработать информационно-поисковую систему в заданной предметной области **на основе управления базой данных**, созданной в каком-либо формате СУБД.

Указания к работе

Рассмотрим пример разработки информационно-поисковой системы.

Поля записи: Фамилия, Курс, Факультет, Специальность, Группа. Организовать поиск записей по полю фамилия.

Последовательность выполнения работы:

1) Загрузите программу Delphi. С помощью главного меню дайте команду: Пуск\Borland Delphi 6\Database Desktop.

2) С помощью команды File\New\Table выберите тип таблицы в формате dBase for Windows.

3) Задайте структуру базы данных, определив имена, типы и длины полей следующим образом:

Field Name	Type	Size	Dec
Familia	C	15	
Kurs	N	1	
Fakultet	C	15	
Spez	C	7	
Gruppa	N	2	

Здесь тип C означает символьный тип, N – числовой тип.

4) Сохраните файл с помощью команды Save as под именем Stud в какую-либо произвольную папку, например в папку BAZA, находящуюся в корневом каталоге диска C.

5) На форму Delphi перенесите следующие 3 компонента:

а) компонент Dbgrid со страницы Data Controls;

б) компонент Data Source со страницы Data Access;

в) компонент Table со страницы BDE.

6) Установите связь между компонентами Dbgrid и Data Source. Для этого в компоненте Dbgrid свойству Data Source поставьте в соответствие Data Source1.

7) Установите связь между компонентами Data Source и Table. Для этого в компоненте Data Source свойству Data Set поставьте в соответствие Table1.

8) В компоненте Table в свойстве DatabasName укажите путь к файлу базы данных \ BAZA. В свойстве TableName укажите имя таблицы Stud.dbf. Значение свойства Active установите в True.

9) Перенесите на форму компонент Dbnavigator со страницы DataControls. Свойству Data Source поставьте в соответствие Data Source 1.

10) Запустите программу на исполнение. Введите 8 -10 записей (+ на компоненте Dbnavigator), удалите произвольную запись (- на компоненте Dbnavigator).

11) Перенесите на форму две кнопки. В свойствах Caption этих кнопок запишите **Поиск №1** и **Поиск №2**. Мы рассмотрим 2 способа поиска информации в базе данных. По кнопке **Поиск №1** реализуем метод **Lookup**, который записывает найденную запись в вариантный массив. По кнопке **Поиск №2** будет релизован метод **Locate**, который делает найденную запись текущей.

12) Перенесите на форму строки вывода **Label**, предназначенные для вывода найденной записи по методу **Lookup** (см. рис. 3).

FAMILIA	KURS	FAKULTET	SPEZ	GRUPPA
Иванов	1	Инженерный	Асу	1
Петров	2	Инженерный	Исит	2
Сидоров	3	Экономический	Фин	3
Медведев	4	Инженерный	Исит	5
Ткачев	5	Экономический	Бух	4

Рис 3. Форма к лабораторной работе №2

13. Для кнопок напишите следующие обработчики событий.

```

procedure TForm1.Button1Click(Sender: TObject);
var
  LookupResults:Variant;
begin
  LookupResults:=Table1.Lookup('FAMILIA',Edit1.text,'familia;
kurs; fakultet; spez; gruppa');
  if VarType(LookupResults)<>varNull then
  begin
    label7.caption:=lookupResults[0];
    label8.Caption:=lookupResults[1];
    label9.Caption:=lookupResults[2];
    label10.Caption:=lookupResults[3];
    label11.Caption:=lookupResults[4];
  end
end

```

```

else
begin
label7.caption:=' ';
label8.caption:=' ';
label9.caption:=' ';
label10.caption:=' ';
label11.caption:=' ';
end;
end;

```

```

procedure TForm1.Button2Click(Sender: TObject);
var
LocateSuccess: Boolean;
SearchOptions:TLocateOptions;
begin
SearchOptions:=[loPartialKey];
LocateSuccess:=Table1.Locate('familia',edit1.text, SearchOptions);
end;

```

14. Просмотрите возможность фильтра в базе данных. Для этого в свойстве Filter компонента Table1 поставьте условие, например Familia='Иванов'. Свойству Filtered поставьте в соответствие True. Отмените фильтр.

15. Запустите программу на исполнение. Просмотрите оба способа поиска записи в базе данных.

Лабораторная работа №3

РАЗРАБОТКА ИНФОРМАЦИОННО-ПОИСКОВОЙ СИСТЕМЫ С ПОМОЩЬЮ ТЕХНОЛОГИИ ADO

Цель работы: изучить методику применения технологии ADO для доступа к базам данных.

Назначение программы: программа позволяет в среде Delphi просматривать информацию в базе данных, созданной в среде Access, выполнять поиск информации на основе задания фильтра, выполнять сортировку по заданному полю.

Порядок выполнения работы :

1) Создадим в Access базу данных Contact.mdb а в ней таблицу с именем Contact. База данных будет иметь 3 поля: Familia, Telefon, Adres. Введите в базу данных 10 записей, содержащих информацию о фамилии, телефоне, адресе клиентов.

2) Установите на форму компоненты ADOConnection, ADODataset со страницы ADO; компонент DataSource со страницы DataAccess, компонент DBGrid со страницы DataControls.

3) Настраиваем компонент ADOConnection
Свойство ConnectionString, кнопка Build, вкладка поставщик данных, выбираем драйвер Microsoft Office 12.0 Access Database Engine OLE DB Provider.
Затем вкладка подключение: укажите источник данных – путь к базе данных, например c:\contact.accdb, далее проверяем подключение. Должно быть выведено сообщение “Проверка подключения выполнена”.

4) Настраиваем компонент ADODataset
Свойство Connection устанавливаем в ADOConnection1, В свойстве Command Text задаем SQL запрос: Select Adres, Familia, Telefon from Contact

Свойство Active устанавливаем в True.

5) Настраиваем компонент Data Source. Свойство Data Set устанавливаем в ADODataset1

6) Настраиваем компонент DBGrid. Свойство DataSource устанавливаем в DataSource1

По событию активизации формы обработчик события имеет вид:

```
procedure TForm1.FormActivate(Sender: TObject);
begin
  AdoConnection1.Open();
  AdoDataSet1.Active:=true;
end;
```

Устанавливаем на форму 2 кнопки: в свойствах Caption которых запишем “Поиск” и “Отмена поиска”. Обработчик события кнопки “Поиск” имеет вид:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  ADoDataSet1.Filtered:=true;
  //if length(Edit1.Text)<>0 then
  // ADODataset1.Filter:='Familia Like '+Edit1.Text+'%'
  // else
  // begin
  // ADoDataSet1.Filtered:=False;
  // showmessage('введи фамилию для поиска');
  // end;
  try
    ADODataset1.Filter:='Familia Like '+Edit1.Text+'%';
  except
    ADoDataSet1.Filtered:=False;
  showmessage('введи фамилию для поиска');
  end;
end;
```

В программе предложено 2 варианта проверки пустой строки: через условный оператор (операторы записаны в виде коммента-

рия) и через конструкции исключительных ситуаций. Во втором случае надо отключить интегрированный отладчик командой:

Tools – DebuggerOptions – Integrated Debugging.

Обработчик события кнопки “Отмена поиска” имеет вид:

```
procedure TForm1.Button2Click(Sender: TObject);
```

```
begin
```

```
  ADoDataSet1.Filtered:=False;
```

```
end;
```

Сортировка методом SQL-запроса.

Устанавливаем на форму 2 кнопки: в свойствах Caption которых запишем “Сортировка” и “Отмена сортировки”. Обработчик события кнопки “Сортировка” имеет вид:

```
procedure TForm1.Button3Click(Sender: TObject);
```

```
begin
```

```
  AdoDataSet1.Close();
```

```
  AdoDataSet1.commandtext:='select adres, familia, telefon from contact order by Familia';
```

```
  AdoDataSet1.Open();
```

```
end;
```

Обработчик события кнопки “Отмена сортировки” имеет вид:

```
procedure TForm1.Button4Click(Sender: TObject);
```

```
begin
```

```
  AdoDataSet1.Close();
```

```
  AdoDataSet1.commandtext:='select adres, familia, telefon from contact';
```

```
  AdoDataSet1.Open();
```

```
end;
```

Сортировка с помощью свойства IndexFieldNames компонента ADODataSet1

Устанавливаем на форму 2 кнопки: в свойствах Caption которых запишем “Сортировка” и “Отмена сортировки”. Обработчик события кнопки “Сортировка” имеет вид:

```
procedure TForm1.Button5Click(Sender: TObject);  
begin  
  AdoDataSet1.IndexFieldNames:='Familia';  
end;
```

Обработчик события кнопки “Отмена сортировки” имеет вид:

```
procedure TForm1.Button6Click(Sender: TObject);  
begin  
  AdoDataSet1.IndexFieldNames:="";  
end;
```

На рис 4 представлен внешний вид формы:

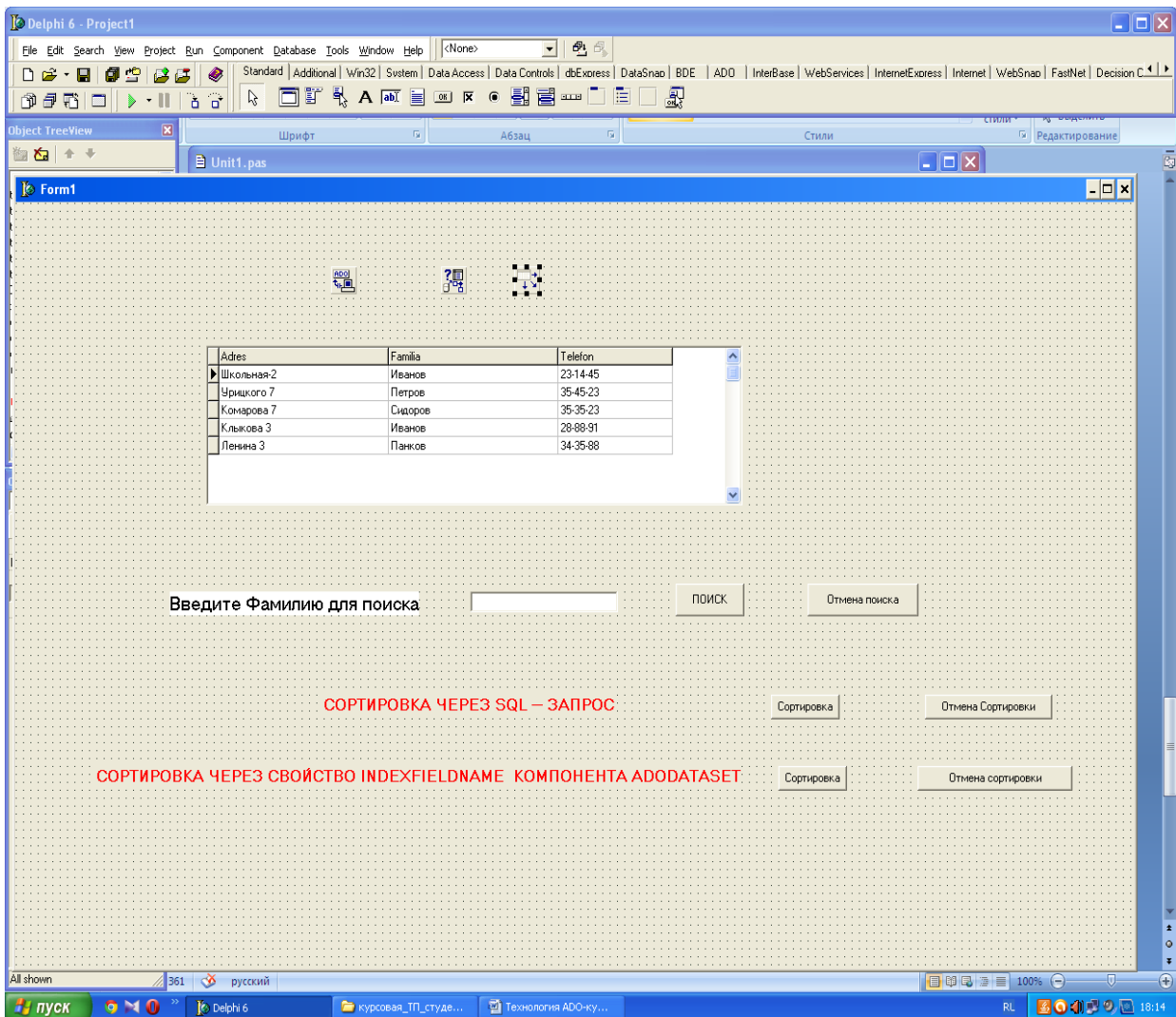


Рис 4. Форма к лабораторной работе №3

Лабораторная работа №4

ФОРМИРОВАНИЕ ОТЧЕТА

Цель работы: разработать информационную систему по предметной области библиотека на основе технологии BDE, ввод

информации организовать в режиме карточки, выполнить сортировку по фамилии автора, организовать вывод отчета таблицы на принтер.

1) Перенесите на форму компонент Table со страницы BDE, Data Source со страницы Data Access, 3 компонента DBEdit со страницы Data Controls, Dbnavigator со страницы Data Controls. Компоненты DBEdit служат для вывода информации, содержащейся в базе данных.

2) Используя технологию BDE, задайте структуру базы данных с полями Avtor, Kniga, Shifr. База данных будет содержать информацию об авторе книги, названии книги, шифре книги (см. лаб. 2).

3) Выполните настройку связи компонентов. (см. лаб. 2). Однако в данной работе также необходимо связать компоненты DBEdit с компонентом Data Source1, а затем связать компоненты DbEdit с полями базы данных по свойству DataField.

4) Выполните сортировку по полю Avtor. Для этого в режиме создания структуры базы данных (Пуск\ Borland Delphi б\ Database Desktop) укажите индексное поле. Для этого необходимо использовать команду **restructure**, затем опцию Define, затем выбрать индексное поле. Далее следует перенести на форму кнопку с надписью **Сортировка**, и по обработчику события кнопки записать следующий оператор:

```
Table1.IndexName:='Avtor';
```

5) Добавьте к проекту еще 1 форму, на которой будем формировать отчет в режиме таблицы (смотрите методические рекомендации для курсового проекта - метод использования внешнего файла на диске).

ФОРМИРОВАНИЕ ОТЧЕТА

б) Установите на вторую форму компонент Table. Установите связь данного компонента с таблицей базы данных. (см. лаб. 2)

7) Перенесите на форму компонент QuickRep со страницы Qreport. Данный компонент представляет собой лист отчета. Свяжите данный компонент с компонентом Table по свойству DataSet.

8) Установите на лист отчета 3 полосы в виде компонентов QRBand. В свойстве BandType данных компонентов выберите для верхней полосы rbTitle (заголовок), для средней полосы – rbColumnHeader (названия колонок), для нижней полосы Detail (здесь выводится информация базы данных).

9) Перенесите на верхнюю полосу компонент QRLabel с надписью Каталог Книг.

10) Перенесите на среднюю полосу 3 компонента QRLabel с надписями Книга, Автор, Шифр.

11) Перенесите на нижнюю полосу 3 компонента QRDBText и по свойству DataField свяжите данные компоненты с полями базы данных.

12) Проверьте, чтобы в составном свойстве DataSet компонентов QuickRep и QRDBText свойство Active было установлено в True.

13) Установите на листе отчета кнопку с надписью **ПРЕДВАРИТЕЛЬНЫЙ ПРОСМОТР**. По обработчику события нажатия кнопки запишите оператор QuickRep1.Preview();

14) Просмотрите отчет и распечатайте его на принтере, щелкнув по значку принтера в верхней части экрана.

На рисунках 5, 6 представлен внешний вид первой формы, на которой мы просматриваем информацию в режиме карточки и второй формы, на которой формируется отчет.

1. Предлагаю студентам самостоятельно сделать отчет в режиме карточки. Для этого можно по оператору присваивания связать строки вывода QRLabel в отчете со строками ввода DBEdit. На принтере должна распечатываться текущая карточка.

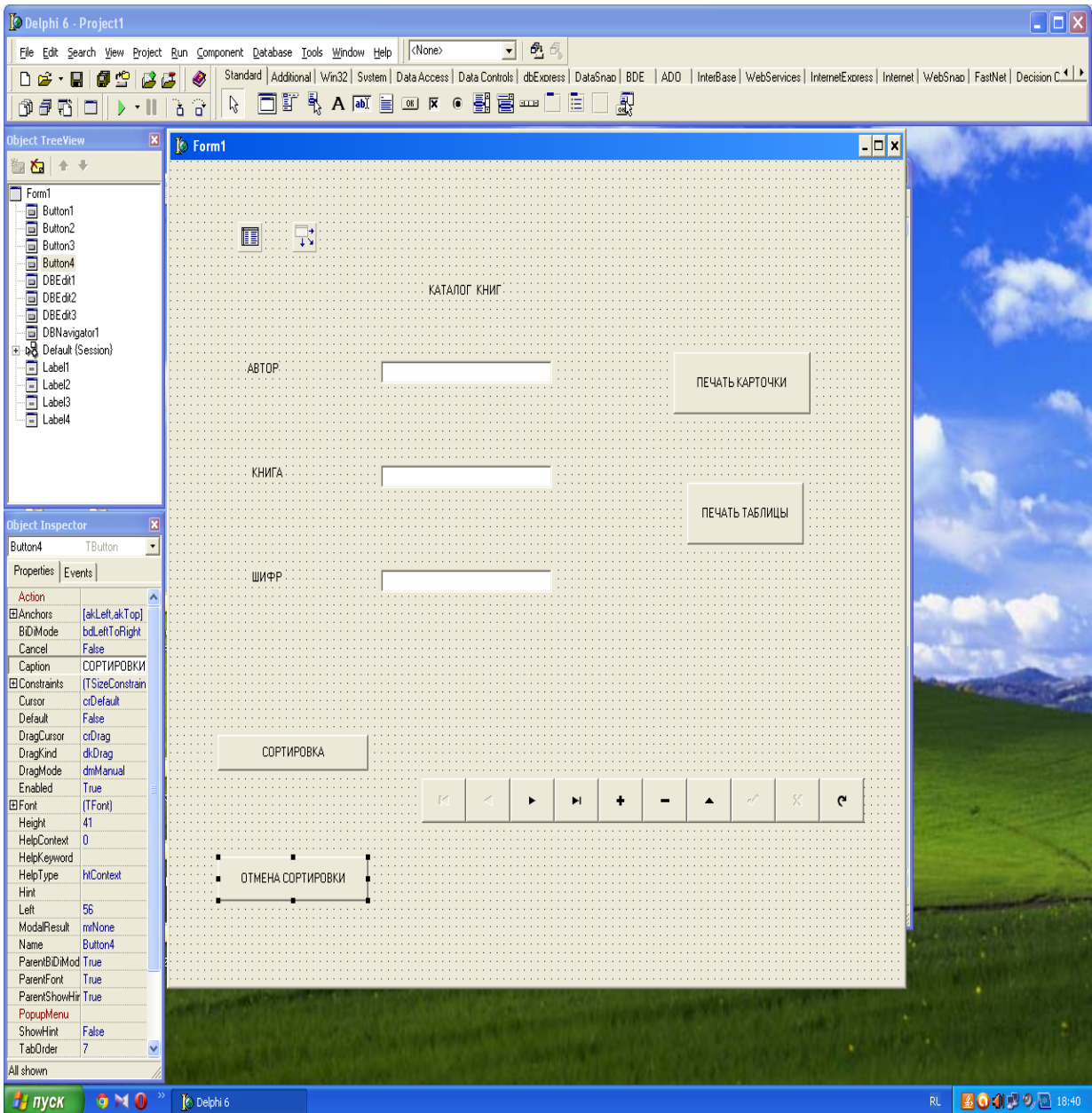


Рис 5. Форма к лабораторной работе №4

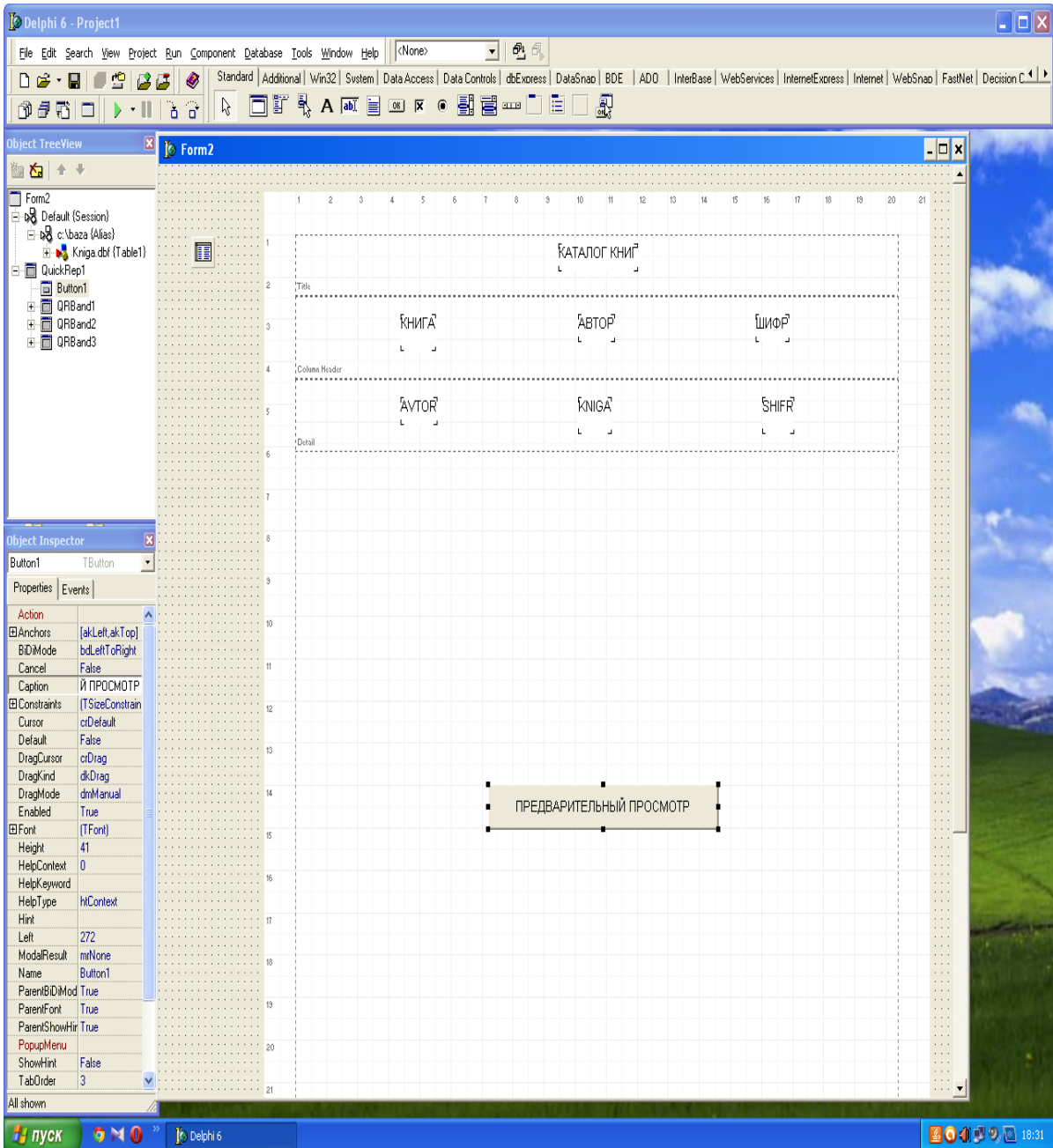


Рис 6. Форма для формирования отчета к лабораторной работе №4

Особенности создания отчета по технологии ADO по сравнению с технологией BDE

- 1) Компонент QuickRep1 связываем с компонентом ADODataSet по свойству DataSet.
- 2) Компоненты QRDbText также связываем с компонентом ADODataSet по свойству DataSet , а с конкретными полями базы данных связываем по свойству DataField.

СПИСОК КОНТРОЛЬНЫХ ВОПРОСОВ

1. Структура проекта, структура модуля Delphi.
2. Принцип объектно-ориентированного программирования. Взаимодействие формы и модуля. Обработчик события OnClick (щелчок по кнопке). Реакция на события мыши и клавиатуры.
3. Назначение компонент вкладки Standart.
4. Назначение компонент вкладки Additional.
5. Назначение компонент вкладок System, Dialogs.
6. Массивы в Delphi.
7. Записи в Delphi.
8. Строки в Delphi.
9. Формирование двумерного массива строк с помощью таблицы StringGrid.
10. Операции с файлами в Delphi.
11. Диалоги открытия и сохранения файлов в Delphi.
12. Оператор присваивания, составной оператор, условный оператор в Delphi.
13. Операторы, предназначенные для организации циклов в Delphi.
14. Операторы, предназначенные для досрочного выхода из циклов и процедур в Delphi.
15. Процедуры и функции в Delphi. Локальные и глобальные переменные. Понятие рекурсии.
16. Организация работы с несколькими формами в рамках единого проекта Delphi.
17. Какие компоненты необходимо занести на форму для просмотра базы данных при технологии BDE и как их настроить ?

18. Какие компоненты необходимо занести на форму для просмотра базы данных при технологии ADO и как их настроить ?

19. Какие методы поиска информации существуют в приложениях на языке Delphi для управления базами данных ? Объясните принципы их работы.

20. Какими методами можно сделать проверку того, что в поле поиска не введено никаких символов?

21. Какие компоненты необходимо установить на форму Delphi для вывода отчета на принтер и как их настроить?

СПИСОК ЛИТЕРАТУРЫ

1. Фленов М. Е. Библия Delphi. – 3-е изд. перераб. и доп. – СПб.: БХВ Петербург, 2012 г.

2. Хомоненко А. Д. Самоучитель Delphi.- 2-е изд., перераб. и доп. / А. Д. Хомоненко, В. Э. Гофман. – СПб.: БХВ-Петербург, 2008 г.

3. Иванова Г. С. «Объектно-ориентированное программирование»: изд. МГТУ, 2003 г.

4. Фаронов В. В. «Delphi 6. Программирование на языке высокого уровня: Учебник для вузов» - СПб.: Питер, 2005г.

5. Сорокин А. В. Delphi. Разработка баз данных.- СПб: Питер, 2005г.

6. Писарев Т. В. «Программирование в Delphi 6». М.: ДМК Пресс 2002 г.

7. Архангельский А. Я. «Интегрированная среда разработки Delphi от версии 1 до версии 5». М.: Изд. БИНОМ, 1999 г.

8. Бобровский С. И. «Delphi 7. Учебный курс» - СПб.: Питер, 2005 г.