Borland®

Использование компонент Delphi для работы с базами данных

Разработка приложений, использующих базы данных – сложный процесс. К тому же существует множество различных баз данных, как локальных, так и удаленных, со своими драйверами и компонентами.

Изучение всего этого займет очень много времени. Существует масса учебников и пособий по различным базам данных.

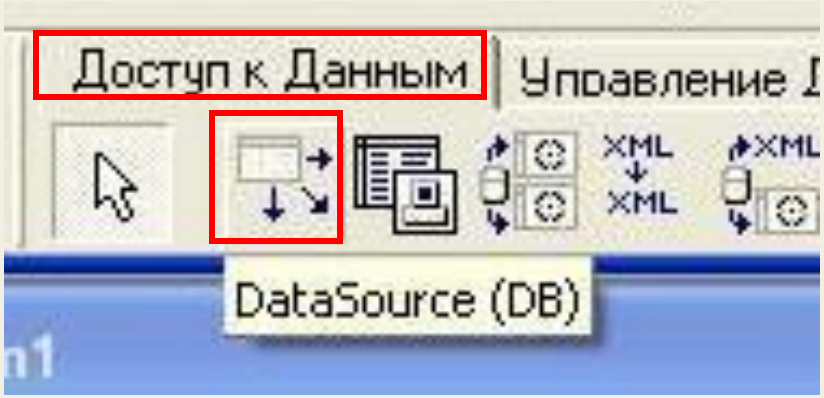
На этом уроке мы рассмотрим только использование и простые действия с локальными базами данных **Paradox** и **MS Access**. Предполагается, что Вы знаете основные определения и устройство реляционных баз данных, работу в Access, в противном случае сначала изучите этот материал (можно использовать цикл презентаций по базам данных, выложенных на сайте «**Базы данных – 1,2,3,4,5**»)

К презентации приложены электронные учебники по базам данных для углубленного изучения :

Delphi 7 - БД (там вы найдете, как создать простую базу данных Paradox, воспользовавшись входящей в состав Delphi утилитой Borland Database Desktop)

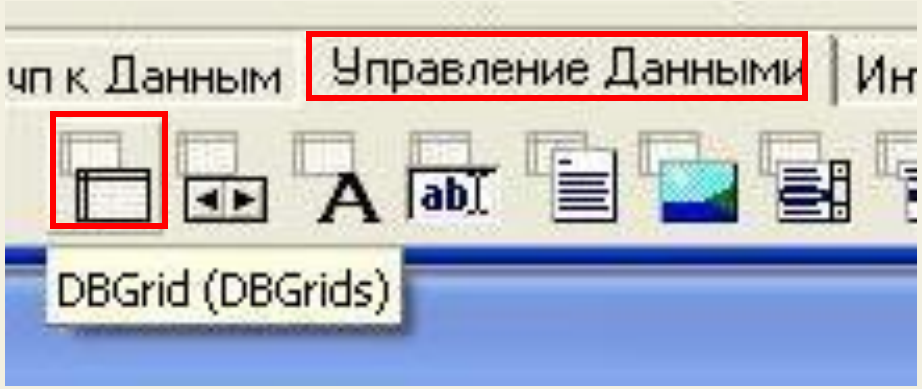
Начнем с обзора некоторых компонент, которые понадобятся нам для создания приложения, использующего локальную базу данных

1. Компонент **TTable** – основной компонент базы, через который идет обращение к конкретной таблице конкретной базы данных. Находится он на вкладке **BDE**

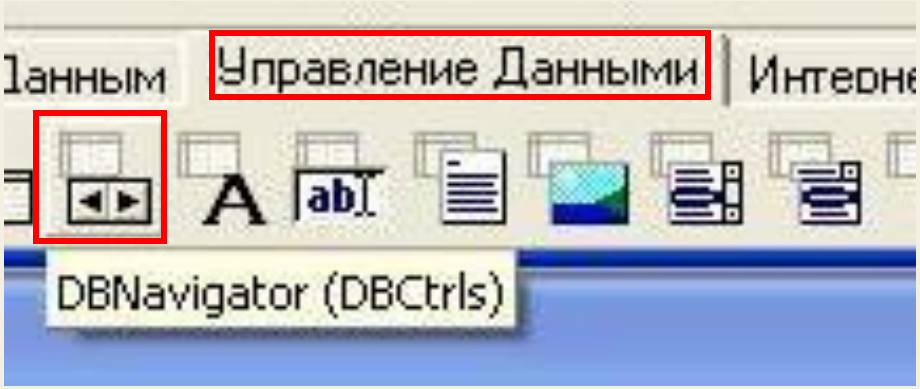


2. Компонент **TData Sourse** – он связывает наш компонент **TTable** с конкретной таблицей на нашем диске. Находится на вкладке **Доступ к Данным**

Начнем с обзора некоторых компонент, которые понадобятся нам для создания приложения, использующего локальную базу данных



3. Компонент **TDBGrid** – визуальный компонент, который отображает таблицу и ее содержимое на форме (без него мы не сможем увидеть таблицу, хотя программно можем с ней работать). Находится на вкладке **Управление данными**



2. Компонент **TDBNavigator** – визуальный компонент, который позволяет осуществлять навигацию по базе. Находится на вкладке **Управление данными**

После знакомства с компонентами начнем создание приложения, причем в качестве базы данных используем готовую базу с описанием и фото рыбок, имеющуюся в Delphi в качестве демонстрационной базы

Начнем по шагам

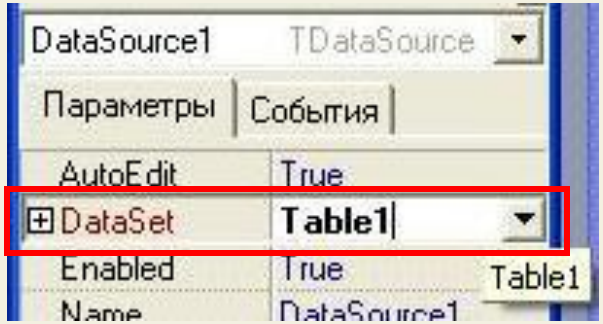
ШАГ 1

Положим на форму компонент **TTable** с вкладки BDE. Сейчас присоединим его к конкретной базе данных на нашем диске

Для этого раскрываем свойство **DataBaseName** и выбираем базу данных **DBDEMOS** – это демонстрационная база, входящая в состав Delphi

Дальше раскроем свойство **TableName** и выберем среди нескольких входящих таблиц файл **biolife.db** – это и есть таблица, которая содержит описание и фото рыбок

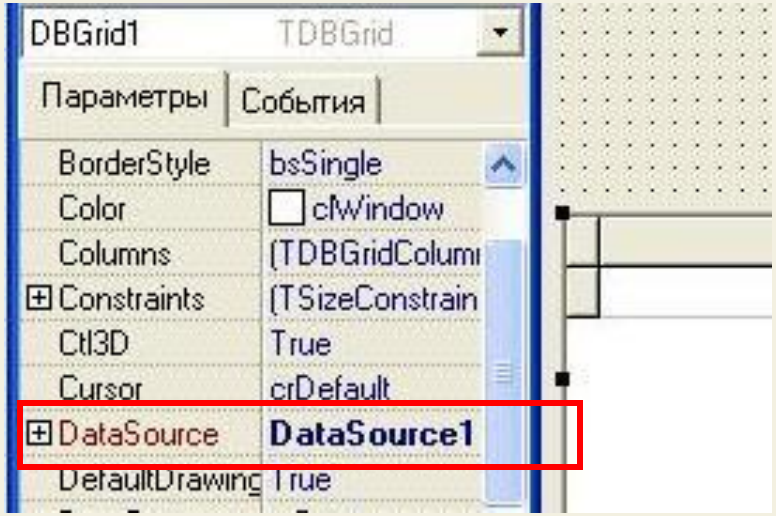
ШАГ 2



Ложим на форму компонент **TDataSourse** – он будет связывать визуальные компоненты, которые отображают содержимое таблицы с компонентом **TTable**

Находим свойство **DataSet** у этого компонента и в выпадающем списке указываем на **Table1**

ШАГ 3



Ставим на форму компонент **DBGrid** – он и будет отображать нашу таблицу с рыбками

В свойстве **DataSourse** выбираем источник данных – **DataSourse1**

ШАГ 4

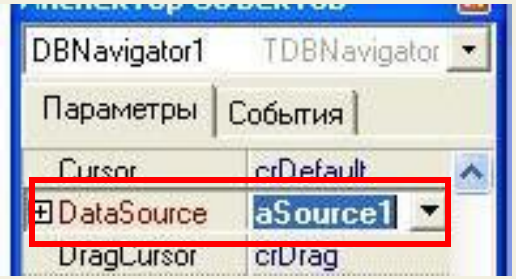
А сейчас **делаем таблицу активной**: у компонента **TTable** устанавливаем свойство **Active** в **True**.

В результате в **DBGrid** мы видим содержание таблицы :

Species No	Category	Common_Name	Species Name	Length (cm)	Length_In	Notes	Graphic
90030	Snapper	Red Emperor	Lutjanus sebae	60	0472440945	(MEMO)	(GRAPHIC)
90050	Wrasse	Giant Maori Wrasse	Cheilinus undulatus	229	4803149606	(MEMO)	(GRAPHIC)
90070	Angelfish	Blue Angelfish	Pomacanthus nauarchus	30	0236220472	(MEMO)	(GRAPHIC)
90080	Cod	Lunartail Rockcod	Variola louti	80	6062992126	(MEMO)	(GRAPHIC)

Можно откомпилировать приложение и поработать с таблицей – мы можем просматривать и редактировать эту базу

ШАГ 5



Для удобства работы с таблицей поместим на форму элемент **DBNavigator** с вкладки **Управление данными** и в инспекторе объектов поставим его свойство **DataSource** указывающим на тот же **DataSource1**, что и для **DBGrid** - сейчас работать с таблицей стало гораздо удобнее

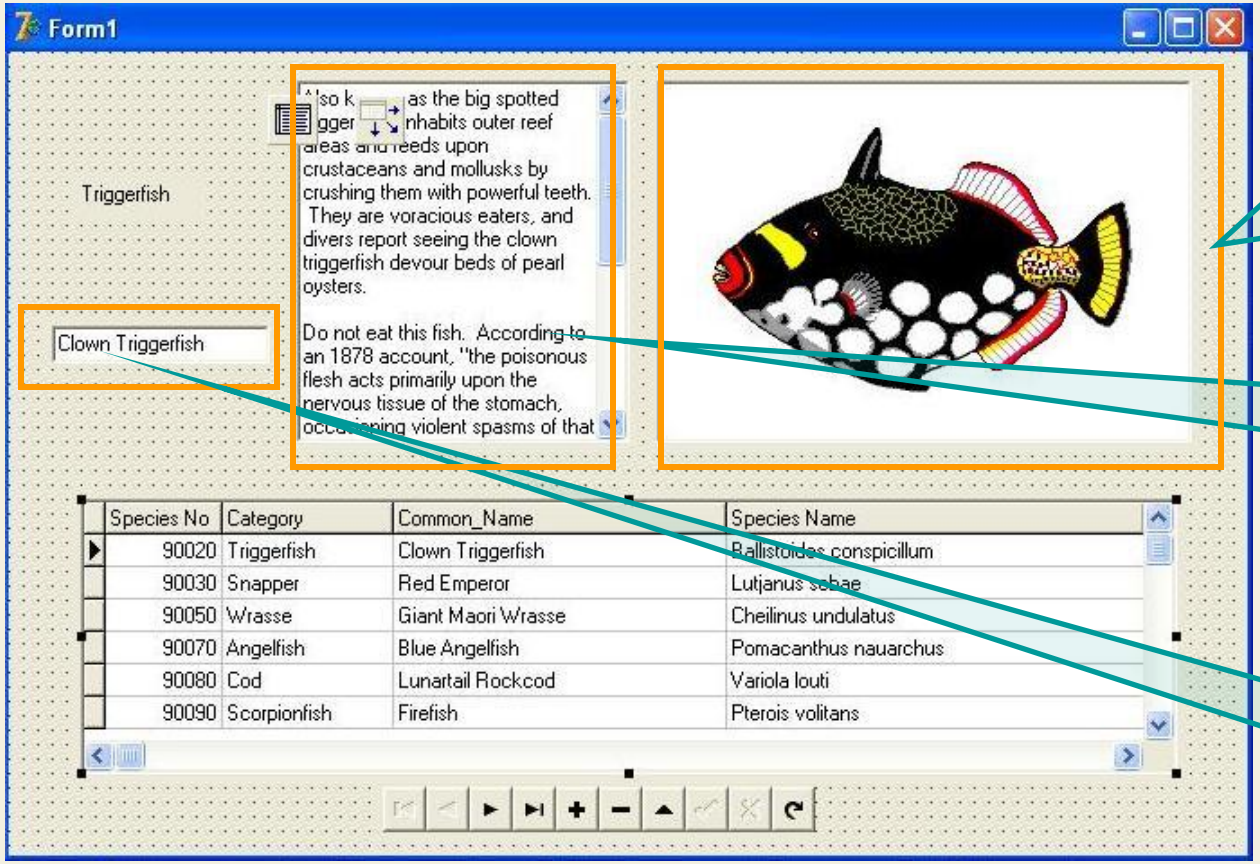
ШАГ 6

В Delphi имеется ряд компонент для отображения содержания отдельных ячеек – **DBEdit, DBMemo, DBImage** и др. Все они находятся на вкладке Управление данными . Поместим их на форму и свяжем с определенными столбцами таблицы: для каждого из этих компонентов укажем свойство **DataSourse** в **DataSourse1**, а свойство **DataField**, следующим образом:

DBEdit - ассоциируем с полем Common_name

DBMemo - ассоциируем с полем Notes

DBImage - ассоциируем с полем Graphic



DBImage, отображающий ячейку с рисунком рыбки

DBMemo, отображающий ячейку с описанием

DBEdit , отображающий ячейку с именем

Borland®

Создаем приложение с базой данных MS Access

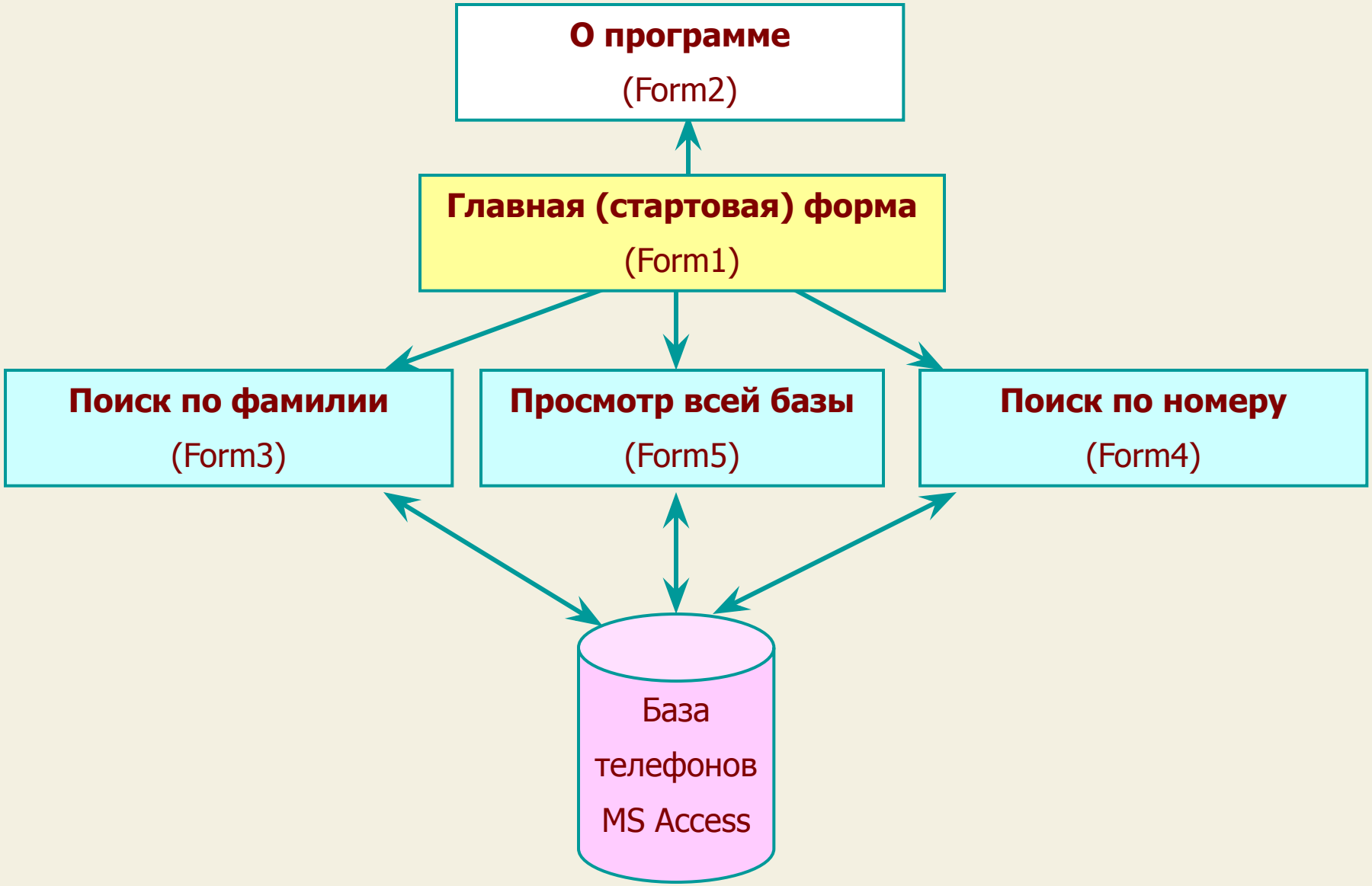
Для работы с базой данных **сначала создадим ее в MS Access**. Пусть это будет телефонный справочник с полями ФИО, ТЕЛЕФОН и АДРЕС

Таблица1 : таблица

ФИО	телефон	Адрес
Абрамов	50212	
Аскаров	51021	
Астахов	51236	
Ахтемзянов	51010	
Бандалетов	52441	
Бахвалов	52361	
Быков	52330	
Димиев	51787	
Домнин	51252	
Дукалис	51332	
Закиров	52110	
Зарипов	51233	
Захватаев	52447	
Иванов	51002	

Заполним таблицу произвольными значениями и сохраним ее. Сейчас можно приступать к созданию приложения. Нашим приложением будет **электронный телефонный справочник с функциями поиска по номеру или по фамилии**

Рассмотрим структуру нашей программы:



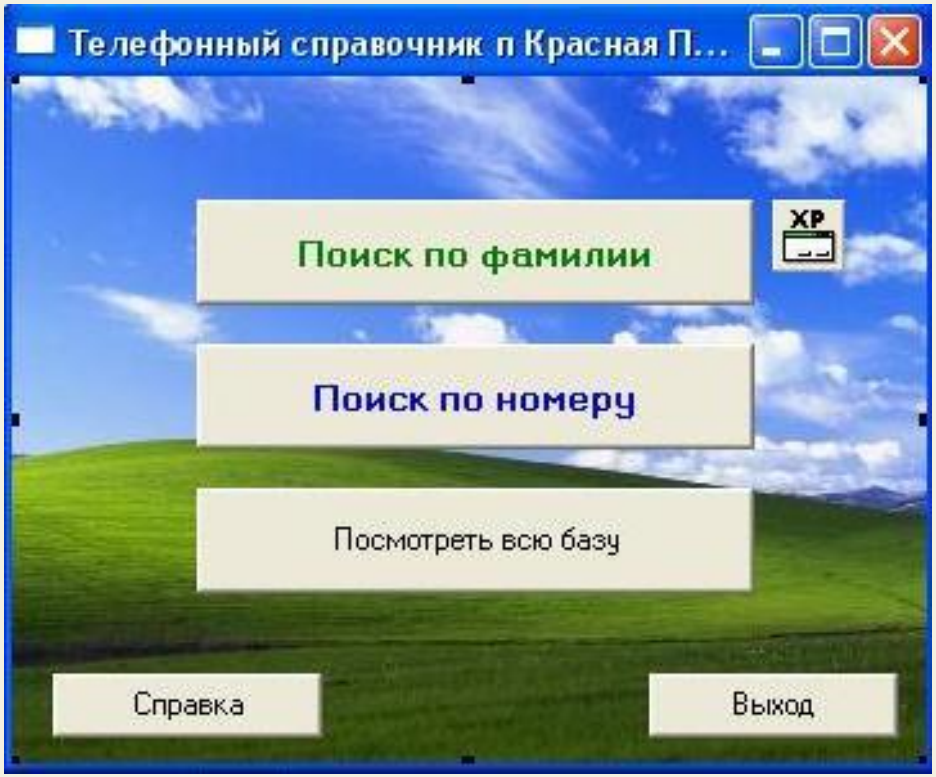
Как видно из схемы, наше приложение содержит 5 форм и опирается на базу Ms Access

ШАГ 1

Создадим 5 форм, познакомим их, назовем заголовки форм, выберем размеры и стили форм

ШАГ 2

На **главной** (стартовой) **форме** (Form1) расположим компоненты:

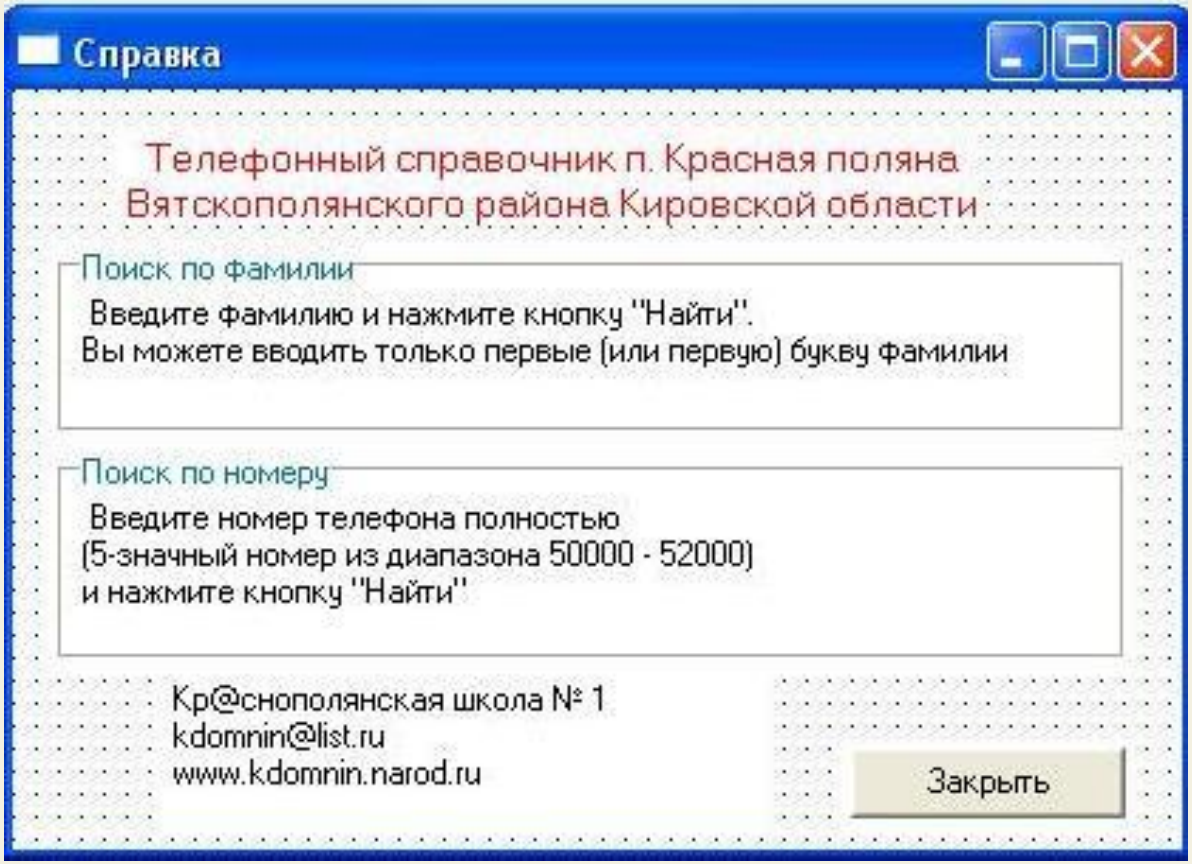


1. Кнопка – **Поиск по фамилии**
2. Кнопка – **Поиск по номеру**
3. Кнопка – **Посмотреть всю базу**
4. Кнопка – **Справка**
5. Кнопка – **Выход**
6. Манифест XP

Для всех кнопок запишем соответствующий код (открытие соответствующей формы – **ShowModal**, выход – **close**)

ШАГ 3

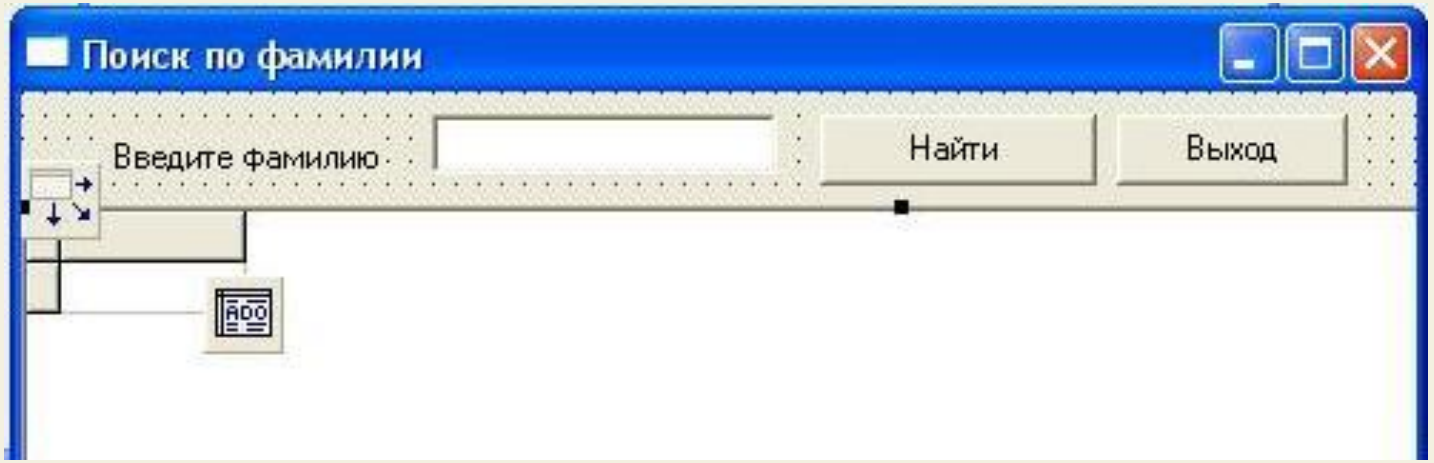
Разработаем дизайн формы **Справка**



Расположим на ней информацию по работе с программой с помощью соответствующих компонент и кнопку **Закрыть**, для которой запишем код выхода

ШАГ 4

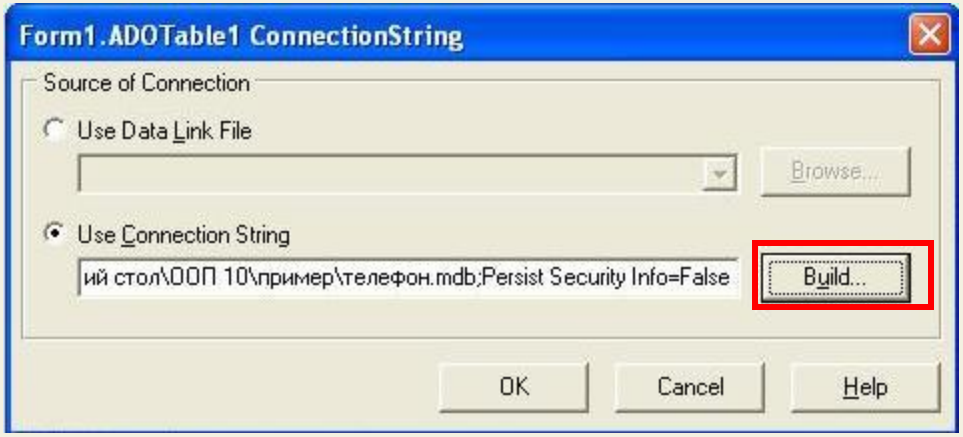
Разрабатываем форму **Поиск по фамилии**. Эта форма должна быть связана с базой телефонов MS Access



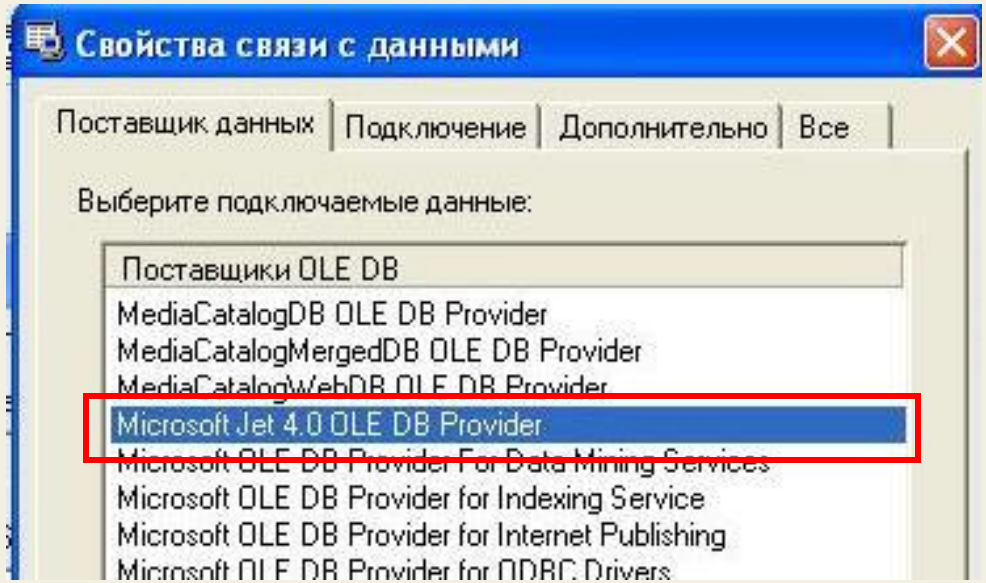
Для связи формы с Access используем следующие компоненты:

1. Вместо **TTable**, которую мы использовали в первом примере, для связи с Access служит «свой» компонент – **ADOTable**, который находится на вкладке ADO.

   Помещаем его на форму и привязываем к таблице телефонных номеров. Для этого раскрываем свойство **ConnectionString** и нажимаем кнопку **Build**

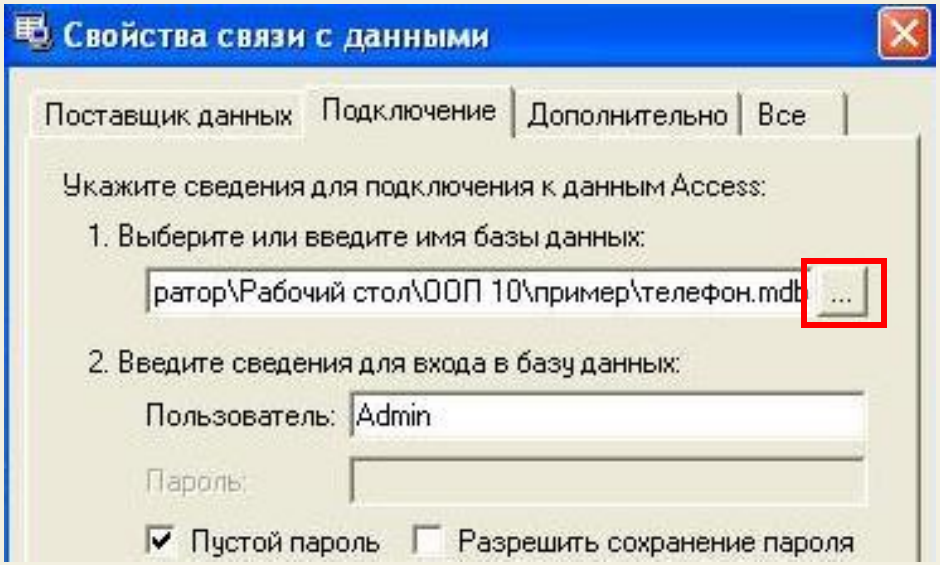


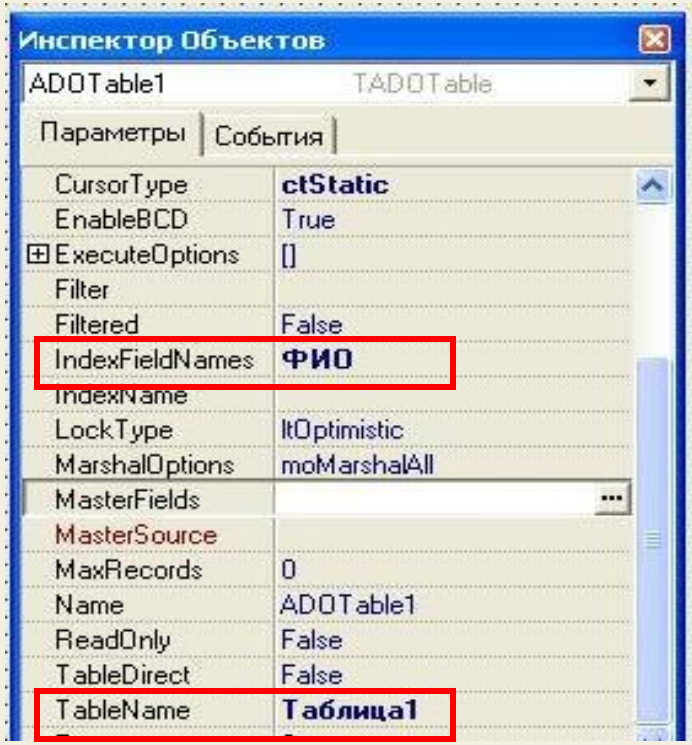
2. Открывается окно связи с данными, в котором на вкладке **Поставщик** данных выбираем **Microsoft Jet 4.0 Ole DB Provider**



3. Переходим на вкладку **Подключение** и выбираем через кнопку обзора нашу базу (**телефон.mdb**)

Здесь же можно проверить подключение, задать пароль и права доступа к базе



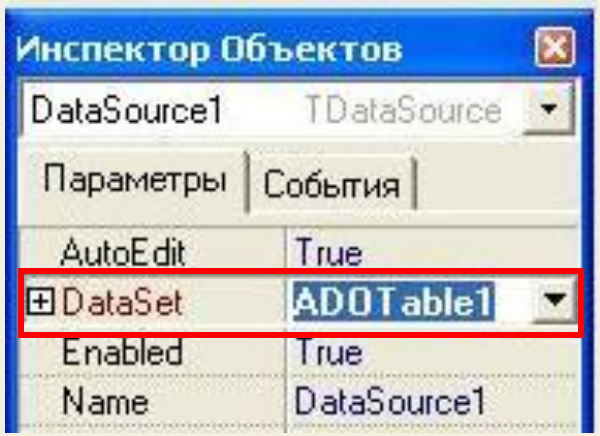


4. Устанавливаем свойства **ADOTable**:

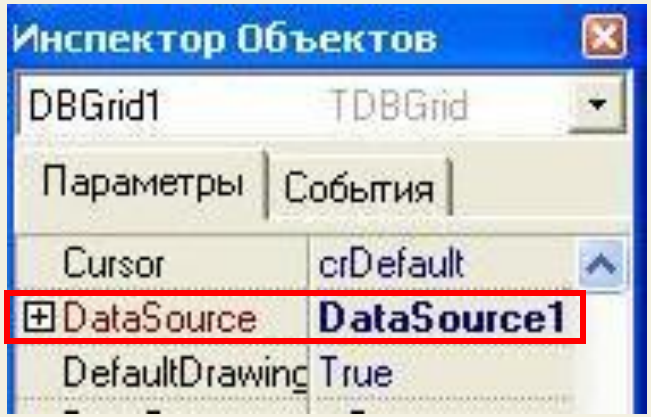
TableName – в раскрывающемся списке выбираем нашу таблицу (у меня она названа **Таблица1**

IndexFieldName – **ФИО** (данные будут сортированы по полю ФИО)

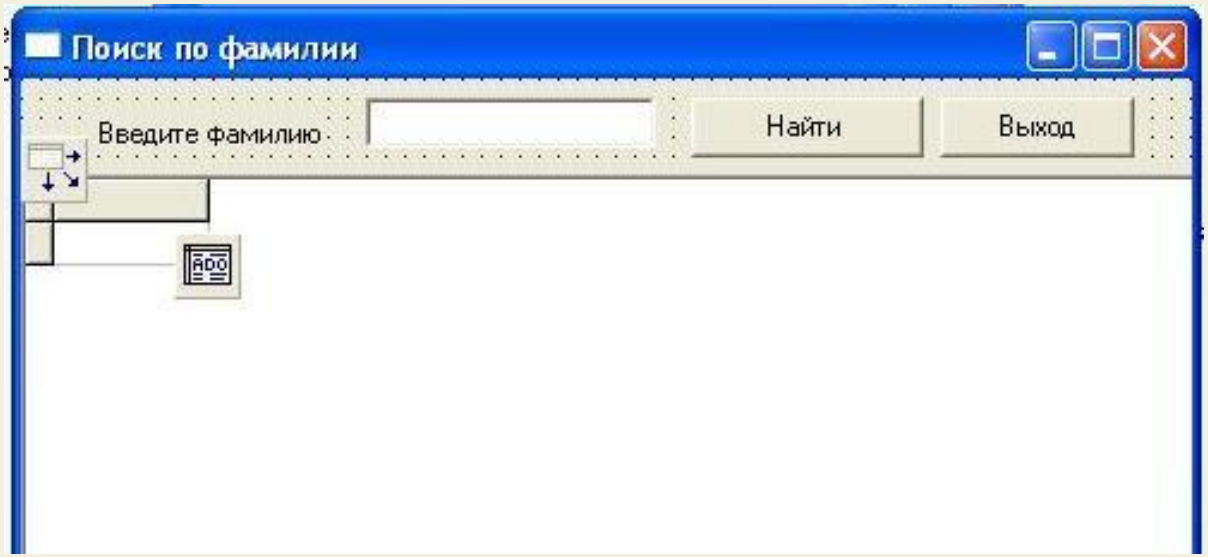
Свойство **Active** ставим в **True**



5. Помещаем на форму элемент **DataSourse** и его свойство **DataSet** устанавливаем через раскрывающийся список в **ADOTable1**



6. Ложим на форму компонент для отображения данных таблицы **– DBGrid**, у которого источником данных выбираем **DataSourse1** – и мы сразу видим, что в **DBGrid появились данные из нашей таблицы**



7. Помещаем на форму кнопки **Найти** и **Выход**, Label «**Введите фамилию**» и поле **Edit** для ввода фамилии

8. В результате мы имеем следующую форму

Свойство **Visible** у **DBGrid** – а сделаем **False**, чтобы при открытии формы поиска не было видно содержимого таблицы

Сейчас запишем код кнопки **Найти**

```
procedure TForm1.Button1Click(Sender: TObject);
begin
if edit1.Text='' then
showmessage('Нербходимо ввести хотя бы часть фамилии')
else
begin
dbgrid1.Visible:=true;
Adotable1.Active:=true;
AdoTable1.Locate('ФИО',Edit1.text,[loPartialKey])
end;
end;
```

Если фамилия не введена, то выводим об этом сообщение

Если фамилия введена, то видимость грида делаем True, активизируем ADOTable и осуществляем поиск по полю ФИО с помощью метода Locate

(Методы поиска описаны в Электронном учебнике по базам данных (Шумаков), прилагаемом к презентации)

Ищем по полю ФИО

Параметр поиска, позволяющий искать по первым буквам фамилии

Образец для поиска берем из Edit – а, в который вводится фамилия для поиска

И последнее – код кнопки **Выход**

```
procedure TForm1.Button2Click(Sender: TObject);
begin
edit1.Text:='';
  dbgrid1.Visible:=false;
form1.Close
```

Очищаем Edit для ввода фамилии

Делаем грид невидимым

Закрываем форму

ШАГ 5

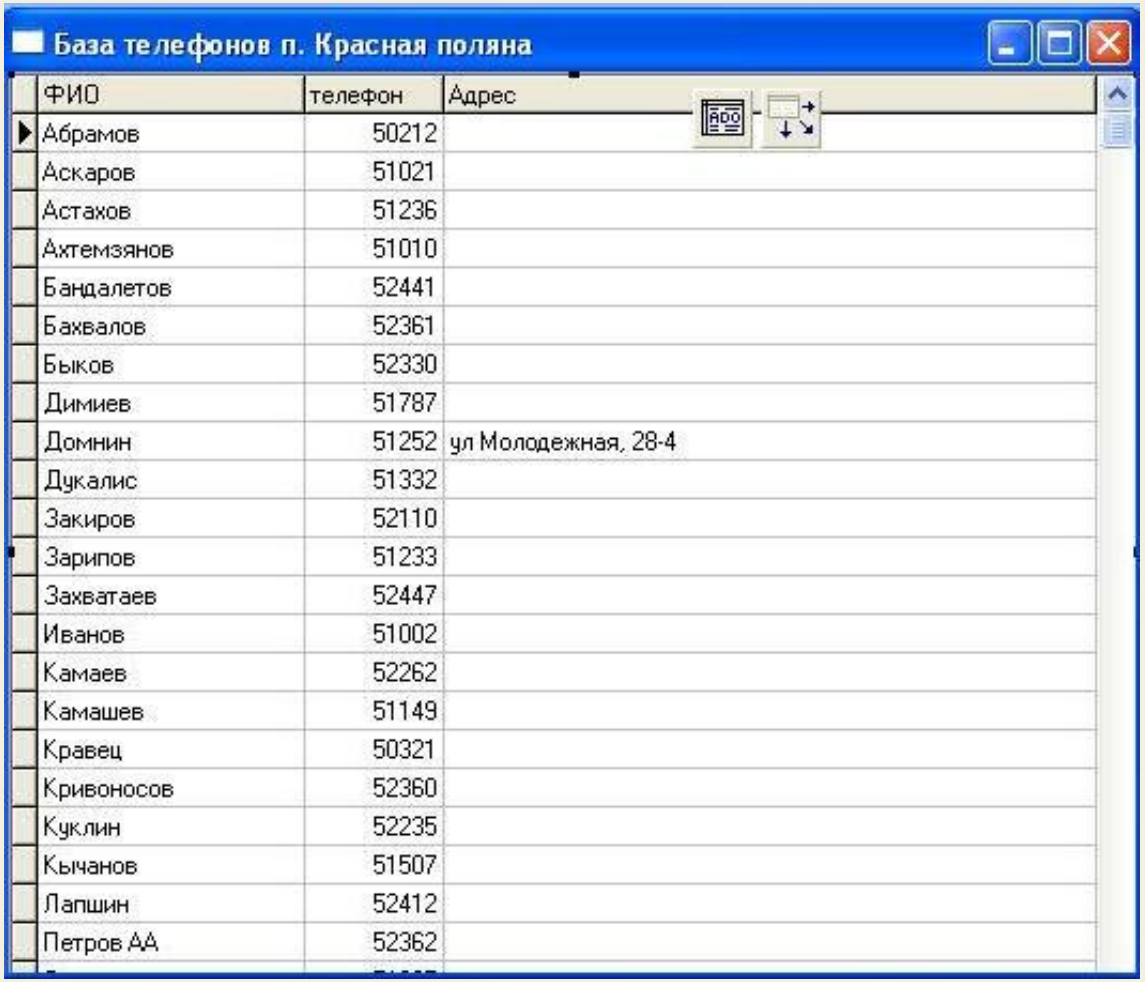
Разрабатываем форму **Поиск по номеру.**

Для этой формы размещаем точно такие же компоненты, связываем их с таблицей – т.е выполняем те же шаги, что и с формой Поиск по фамилии (смотри шаг 4)

Аналогичны и коды кнопок Найти и Выход. Отличие в коде кнопки Найти в том, что в качестве поля поиска указываем поле «Телефон»

ШАГ 6

Последняя форма отображает всю базу, поэтому **DBGrid** на ней можно развернуть на всю форму, а саму форму сделать побольше. Естественно те же компоненты для отображения данных (ADOTable, DataSourse и DBGrid, но кнопок никаких не ставим

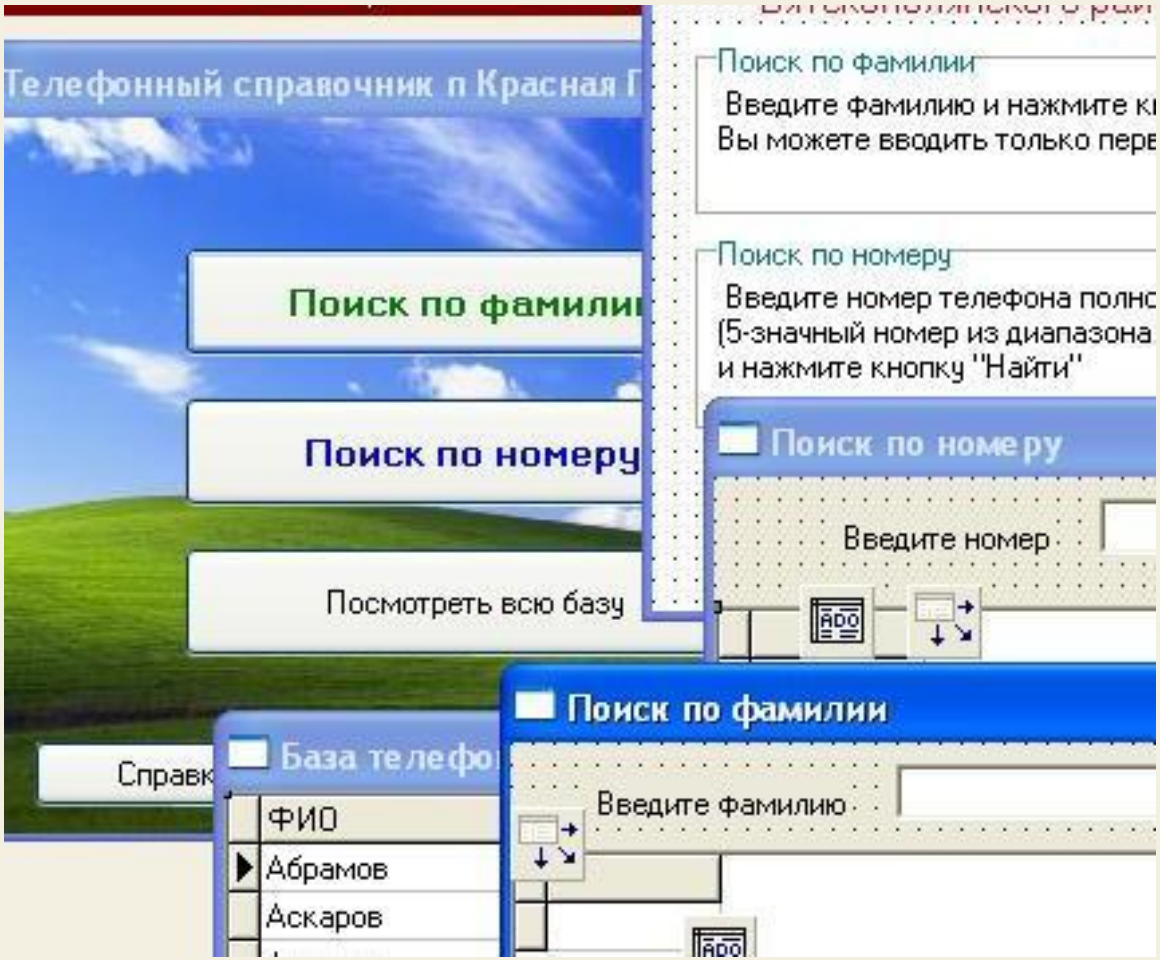


ФИО	телефон	Адрес
Абрамов	50212	
Аскаров	51021	
Астахов	51236	
Ахтемзянов	51010	
Бандалетов	52441	
Бахвалов	52361	
Быков	52330	
Димиев	51787	
Домнин	51252	ул Молодежная, 28-4
Дукалис	51332	
Закиров	52110	
Зарипов	51233	
Захватаев	52447	
Иванов	51002	
Камаев	52262	
Камашев	51149	
Кравец	50321	
Кривоносов	52360	
Куклин	52235	
Кычанов	51507	
Лапшин	52412	
Петров АА	52362	

ШАГ 7

Все сохраняем, компилируем и пробуем работу приложения

Пробуем ->



Итак, на этом уроке мы попытались использовать в Delphi две простые базы данных на базе Access.

Повторюсь, что баз данных множество со своими компонентами, драйверами … Мы ничего не сказали о клиент – серверных приложениях – заблудиться там очень просто, а найтись сложно