2. Builder

После создания базы данных необходимо связать ее с программой. Суть программы будет заключаться в следующем. На экран выводится окно с товарами определенного раздела, среди которых пользователь должен выбрать необходимые ему. Кликнув по нужному товару, его стоимость будет прибавлена общей стоимости выбранных товаров в данном разделе. После выбора всех необходимых товаров, пользователь может перейти в следящий раздел и повторить процедуру выбора товаров. После прохождения всех разделов, будет посчитана общая стоимость всех товаров из всех разделов. Если пользователь хочет посчитать общую стоимость, не проходя дальнейшие разделы, то он может нажать на кнопку посчитать сейчас, которая моментально выведет общую стоимость товаров из всех пройденных разделов.

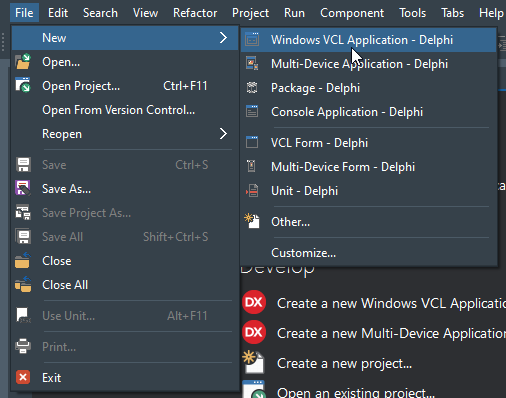
Итак, первым делом необходимо создать новый проект в Delphi. Для этого нажимаем на кнопку «file» далее «new» и «Windows VCL Application - Delphi» (рис. 2.1).

Рис. 2.1 Создание нового проекта

Откроется главная форма, которая будет первым что увидит пользователь при запуске программы. На данной форме будет располагаться краткая инструкция по использованию данной программы.

И все же первым делом необходимо соединить базу данных с данным проектом. Чтобы в дальнейшем при изменении данных было удобно обслуживать программу необходимо создать новую форму где будет храниться все необходимые компоненты для соединения с базой данных и переноса данных в таблицы на других формах.

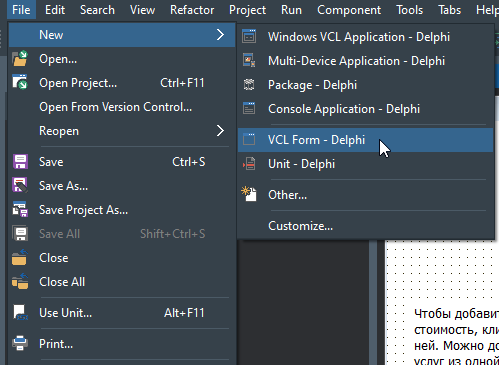
Создаем новую форму. Нужно нажать на «file», «new» и «VCl Form - Delphi» (рис. 2.2).

Рис. 2.2 Создание новой формы

После этого будет создана новая форма. Данная форма не будет видна пользователю, лишь разработчику. Располагаем на этой форме следящие компоненты:

* ADOConnection;
* ADOQuery;
* DataSource;
* StringGrid;
* Label;

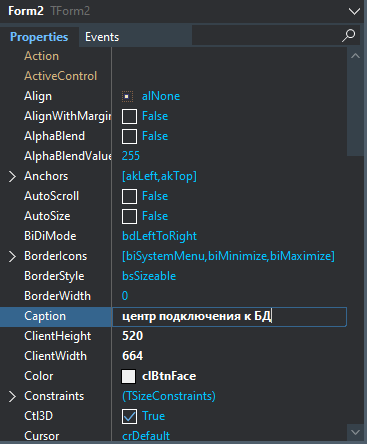
Компоненты Label, ADOQuery и DataSource должны быть в том же количестве, сколько и таблиц в базе данных. Label нужены нам лишь для удобства и могут быть не обязательными. Для того чтобы поменять отображаемое название компонента, например, формы, необходимо выделить компонент мышкой и во вкладке Properties в свойстве Caption ввести свое содержимое (рис. 2.3). Можно использовать кириллицу. После этого можно заметить, как в левом верхнем углу название формы с «Form 2» сменялась на «центр подключения к БД»

Рис. 2.3 Изменение названия формы

Так же необходимо поменять caption у компонентов lable. Можно расположить все компоненты как показано на (рис. 2.4).

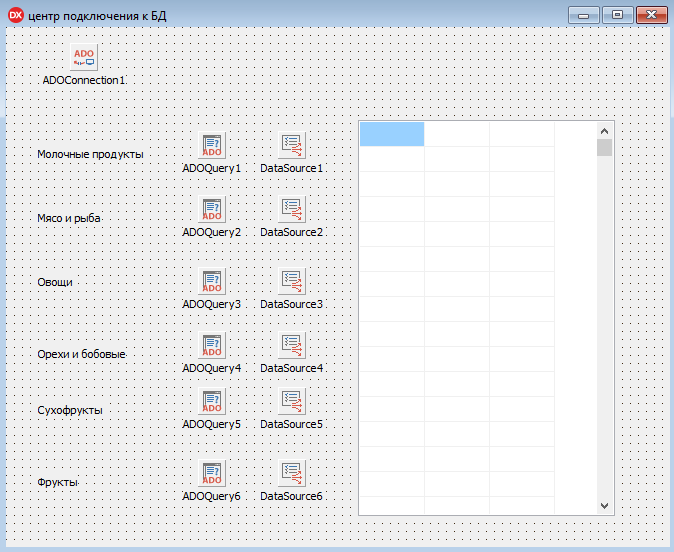


Рис. 2.4 Расположение компонентов на 2 форме

Необходимо произвести настойку компонента StringGrid. Нужно чтобы в нем было 68 строк (количество строк в данной таблице должно быть на одну больше чем строк в самой большой таблице из БД) и 3 столбца. Для этого нужно поменять свойства ColCount на 3 и RowCount на 68. Так же нужно убрать фиксированные ячейки, для этого ставим 0 в свойствах FixedRows и FixedCols.

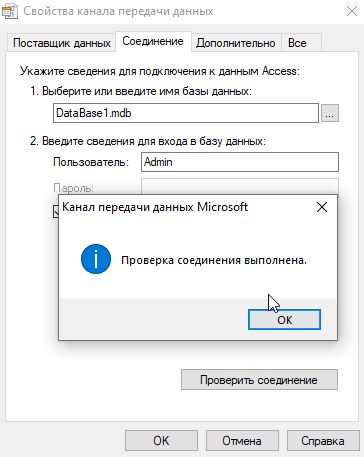
Дважды кликнув на компонент ADOConnection появиться окно, где нужно нажать на кнопку build далее нужно дважды кликнуть на «Microsoft Jet 4.0 OLE DB Provider». Важно понимать, что такое относительный путь и фактический. Относительный за начальную точку принимает папку где находится программа, а фактический начинается с диска и далее по папкам до файла. 2 способ не удобен при переносе программы в другое место или на другой компьютер, ведь путь будет необходимо прописывать заново. Гораздо проще просто перекинуть базу данных в папку с проектом и указать ее имя в поле «Выберите или введите имя базы данных». Сама база данных должна быть с расширением .mdb, если это не так-то необходимо запустить access и сохранить БД в нужном нам формате. После указания имени БД нажмите на кнопку «Проверить соединение» и если все как на (рис. 2.5), то можно переходить к следующему шагу.

Рис. 2.5 Проверка соединения БД

Далее выделяем все ADOQuery и в свойстве connection выбираем наш ADOConnection. После этого в свойстве SQL каждого ADOQuery нужно написать строку по формуле «select\*from [Имя таблицы]». После чего опять выделяем все ADOQuery и меняем значение свойства Active на true.

Далее нужно в каждом компоненте DataSource в свойстве DataSet указать соответствующий ADOQuery.

На этом с формой 2 пока все. Необходимо создать 3 форму. В общем проект должен будет содержать в себе 9 форм из которых 1 инструкция, 1 техническая (2 форма, где идет подключение access), 1 с итоговой ценой и 6 с товарами из различных отделов (соответственно 6 таблиц в базе данных). Формы от 3 по 8 не сильно отличаются друг от друга, поэтому здесь будет описано создание лишь одной из них.

Для начала необходимо расположить на ново созданной форме следующие компоненты:

* Label x4;
* Button x3;
* Memo;
* DBGrid.

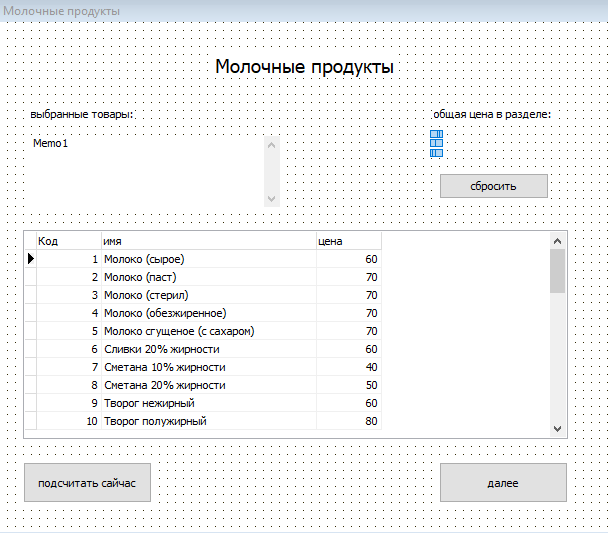
Расположить все эти компоненты можно следующим образом (рис. 2.6).

Рис. 2.6 Форма 3

Нажмите на DBGrid и в свойствах найдите DataSource. если нет никаких вариантов для подключения, нужно открыть код формы и в uses добавить «Unit2», после чего возвращаемся в дизайн и выбираем DataSource, соответствующий данному разделу (нам нужны молочные продукты, а значит выбираем DataSource1). После этих действий в таблице должны появиться товары.

Нам нужно чтобы при нажатии товар переносился в memo, проблема лишь в том, что c DBGrid невозможно работать так же, как и со StringGrid. То есть нельзя просто вынести какаю ни будь строку. Поэтому и была создана таблица StringGrid на 2 форме. Она будет дублировать таблицу, которая сейчас на экране (в зависимости от открытого раздела). С DBGrid нам нужно узнать лишь на какую строку пользователь нажал. Сделать это можно так. Выделите DBGrid и во вкладке Events дважды кликнуть напротив OnCellClick и вставить следующий код:

var

s: integer;

prise: integer;

begin

s := DBGrid1.DataSource.DataSet.RecNo;

memo1.Lines.Add(form2.StringGrid1.Cells[1,s-1]+': '+form2.StringGrid1.Cells[2,s-1]+' руб');

prise := StrToInt(form2.StringGrid1.Cells[2,s-1]);

sumF := sumF + prise; //это глобальная переменная, которая равна изначально 0

labelSumF.Caption := intToStr(sumF);

end;

Перейдем на 1 форму и создадим 2 кнопки: далее и выход. Теперь необходимо, чтобы при нажатии на кнопку далее на 1 форме, заполнялась StringGrid на форме 2. Для этого пишем следующий код кнопке на 1 форме

var i,j:Integer;

begin

//Form9.Memo1.clear; разкоментируте эту строку когда создадите 9 форму

// заполнения strggrid

Form2.StringGrid1.RowCount := Form2.ADOQuery1.RecordCount;

Form2.StringGrid1.ColCount := Form2.ADOQuery1.FieldCount;

Form2.ADOQuery1.First;

for i:=0 to Form2.ADOQuery1.RecordCount-1 do

begin

for j:=0 to Form2.ADOQuery1.FieldCount-1 do

begin

Form2.StringGrid1.Cells[j,i]:=Form2.ADOQuery1.Fields.Fields[j].AsString;

end;

Form2.ADOQuery1.Next;

end;

// переход к 3 форме

form3.show;

form1.Left:=10000;

end;

Можно проверить работоспособность этой кнопки, вывив на экран 2 форму (добавите код в кнопку далее form2.show). Это нужно лишь для проверки, не забудьте удалить эту строку перед продакшеном.

На 3 форме кнопка далее имеет код:

var i,j:Integer;

begin

Form1.sumO := Form1.sumO + sumF;

form9.Memo1.Lines.add(Memo1.text);

// заполнения strggrid

Form2.StringGrid1.RowCount := Form2.ADOQuery2.RecordCount;

Form2.StringGrid1.ColCount := Form2.ADOQuery2.FieldCount;

Form2.ADOQuery2.First;

for i:=0 to Form2.ADOQuery2.RecordCount-1 do

begin

for j:=0 to Form2.ADOQuery2.FieldCount-1 do

begin

Form2.StringGrid1.Cells[j,i]:=Form2.ADOQuery2.Fields.Fields[j].AsString;

end;

Form2.ADOQuery2.Next;

end;

form4.show;

form3.Hide;

end;

кнопка посчитать сейчас:

begin

form9.Memo1.Lines.add(Memo1.text);

Form1.sumO := Form1.sumO + sumF;

form9.label3.Caption := intToStr(Form1.sumO)+' руб';

form9.Show;

form3.Hide;

end;

Кнопка сбросить:

begin

sumF:=0;

labelSumF.Caption := intToStr(sumF);

memo1.clear;

end;

После всех манипуляции можно проверить работоспособность. Программа уже должна уметь добавлять строки в мемо, считать цену в разделе и удалять все при нажатии на кнопку стереть.