Содержание

Лабораторная работа № 2 2
Тема: Создание приложения для работы с базой данных формата Access 2
Лабораторная работа № 3 7
Тема: Разработка форм с использование различных компонентов отображения данных .. 7
Задание для самостоятельного выполнения 8
Лабораторная работа 4 9
Поиск записей. Сортировка и фильтрация записей в наборе данных. 9
Задание для самостоятельного выполнения 11
Лабораторная работа 5 12
Работа со связанными таблицами 12
Лабораторная работа 6 15
Компонент запроса AdoQuery 15
Лабораторная работа 7 18
Разработка документов и приложений MS Word и MS Excel в Delphi. Создание отчетов 18
Лабораторная работа 8 23
Разработка отчетов RaveReports 23

Лабораторная работа № 2

Тема: Создание приложения для работы с базой данных формата Access

Цель работы: Получить навыки построения таблиц формата Access и их подключения к проекту Delphi.

Описание хода выполнения работы

Задание 1. Создание базы данных «Склад»

Запустите MS Accecss создайте базу с названием sklad.accdb. В базе создайте три таблицы:

1. Таблица PRODUCTS (Продукты на складе), которая содержит следующие поля:

id (счетчик) – нужно задать как ключевое,

name (текстовый) –наименование продукта,

cena (тип данных – денежный);

id_kat (тип данных –числовой, в свойствах поля- размер поля- Длинное целое) – код группы продуктов,

kol (тип данных –числовой, в свойствах поля- размер поля- Длинное целое) – код группы продуктов

id_ed (тип данных –числовой, в свойствах поля- размер поля- Длинное целое) – код единицы измерения

srok (тип данных - Дата/время, в свойствах поля- формат поля Краткий формат даты) – дата истечения срока годности.

2. Таблица PRODUCT_GROUP (категория продукта), которая содержит следующие поля:

id (счетчик) – нужно задать как ключевое,

Kat_name (текстовый) –наименование категории продукта.

3. Таблица ED_IZMER (единицы измерения), которая содержит следующие поля:

id (счетчик) – нужно задать как ключевое,

ed_name (текстовый, в свойствах поля- размер поля- 10) – наименование единицы измерения.

Задание 2. Организация связи проекта Delphi с БД.

Создайте новый проект Delphi и сохраните его в той же директории, в которой вы сохранили базу данных sklad.accdb.

Выполните команду File-New-Data Module.

Далее размещаем на форме DataModule2 компонент TADOConnection вкладки ADO.

Выделяем компонент TADOConnection и в свойстве LoginPromt ставим на False. Это делаем для того, чтобы при подключении к БД пароль у нас не запрашивался, дальше в свойстве ConnectionString нажимаем на кнопку с «…» и в появившемся окне нажимаем на кнопку «Build…» и в следующем окне, на вкладке "Поставщики данных" в окне "Поставщики Ole DB" выбираем провайдера базы Access. Так как база данных создана в версии Access 2007, то выбрать следует "Microsoft Office 12.0 Access database Engine OLE DB Provider".

В открывшемся окне в поле "Источник данных" прописать имя базы данных sklad.accdb.

Далее кликнуть на кнопке "Проверить подключение". Если выдано сообщение: "Проверка подключения выполнена", значит все выполнено правильно и кнопкой ОК подтверждаем создание соединения. Если нет, то при создании соединения была допущена ошибка.

На форме DataModule2 разместите компонент ADOTable1 с закладки ADO. В свойстве Connection выберите ADOConnection1. В свойстве TableName выбери таблицу **PRODUCTS**. Свойство Active установить в True. Поместите на форме компонент DataSource1 с закладки DataAccess и в свойстве DataSet укажите ADOTable1.

Аналогично расположите на форме пары компонентов ADOTable2, DataSource2 для таблицы **ED_IZMER** и ADOTable3, DataSource3 для таблицы **PRODUCT_GROUP.**

Задание 3. Разработка форм для справочников

Вернитесь на главную форму приложения.

Добавьте на форму компонент MainMenu и создайте его структуру

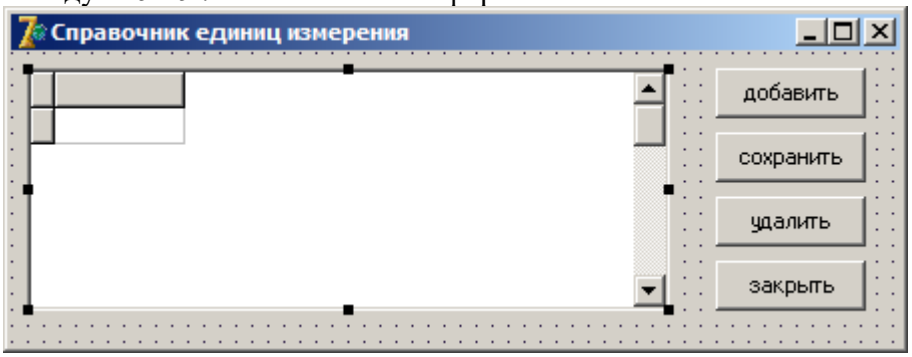
Справочники	**Выход**
Единицы измерения	
Категории продуктов	

Для пункта меню Выход напишите обработчик для завершения работы приложения.

Создадим форму для работы со справочником единиц измерения.

Выполните команду File-New-Form.

В свойстве caption созданной формы form3 пропишите Справочник единиц измерения. К модулю подключите модуль unit2. Расположите на форме компоненты DBGrid1 и четыре кнопки Button.



В его свойстве DataSource компонент DBGrid1 укажите DataModule2.DataSource2.

В обработчик события onClick кнопки Добавить впишите код:

```
datamodule2.ADOTable2.Append;
```

В обработчик события onClick кнопки Сохранить впишите код:

```
datamodule2.ADOTable2.post;
```

В обработчик события onClick кнопки Удалить впишите код:

```
if MessageDlg('Удалить данную запись?', mtConfirmation, mbYesNoCancel
, 0)=mrYes then  datamodule2.ADOTable2.Delete;
```

В обработчик события onClick кнопки Закрыть впишите код: `form3.Close;`

Вернитесь на главную форму. В пункт меню Справочники - Единицы измерения впишите код:

```
form3.show;
```

Запустите приложение. Откройте справочник единиц измерения, добавьте туда несколько записей, например:, кг, шт, л и т.д. Помните, что поле Id – счетчик, его заполнять нет надо.

Аналогично, создайте форму form4 для работы со справочником Категории продуктов, сделайте так, чтобы форма открывалась соответствующим пунктом меню, и добавьте в эту таблицу несколько записей, например, *бакалея, молочка, овощи* и т.д.

Задание 4. Разработка главной формы

4.1. Вернитесь на главную форму приложения. Подключите к нему модуль 2, указав его имя в разделе

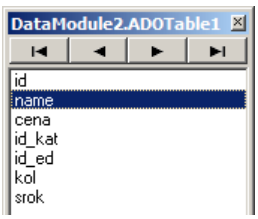
```
implementation
uses Unit2;
```

Расположите на форме компонент DBGrid1. В его свойстве DataSource укажите DataModule2.DataSource1.

Если все сделано правильно, содержимое таблицы **PRODUCTS** должно быть отображено на компоненте.

4.2. Зададим свой состав полей для набора данных adotable1. Для этого откройте форму DataModule2, и двойным щелчком левой кнопки мыши на adotable1 откройте Редактор полей.

Вызовите для Редактора полей контекстное меню и выберите пункт **Add all Fields** (Добавить все поля).



Далее изменим свойства полей.

Выделите в редакторе первое поле **Id**. Его свойству Visible присвойте значение False.

Выделяем поочередно остальные поля, и изменяем их свойства следующим образом:

Name	displayLabel – название продукта displayWidth - 20
cena	displayLabel – цена displayWidth - 10
kol	displayLabel – количество на складе alignment - taCenter
srok	displayLabel – срок годности displayFormat - dd.mm.yyyy alignment - taCenter
id_kat	displayLabel – код категории
id_ed	displayLabel – код единицы измерения

4.3. Создадим вычисляемое поле Stoim (стоимость всей партии данного товара).

Для этого в контекстном меню Редактора полей выбираем пункт **New Field** (Новое поле). Указываем для нового поля:

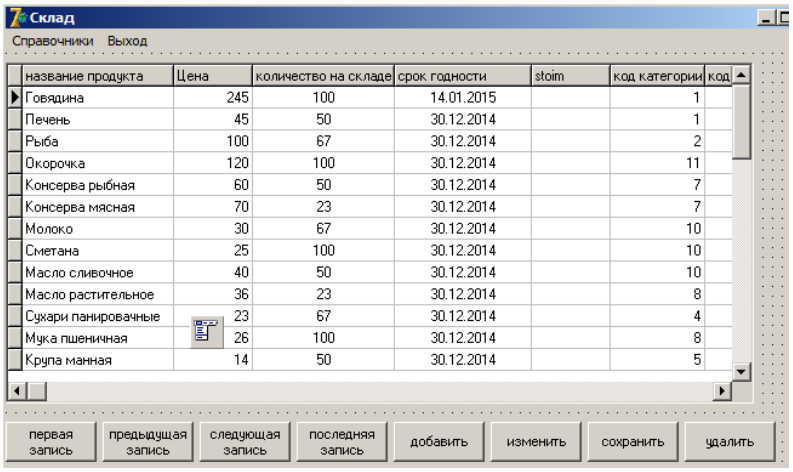
name – Stoim

type – Currency

fieldType – переключатель Calculated

Напишите обработчик события **OnCalcFields** для НД adoTable1, в котором производится расчет значения вычисляемого поля:

```
procedure TDataModule2.ADOTable1CalcFields(DataSet: TDataSet);
begin
adoTable1stoim.AsFloat:=adoTable1.FieldByName('Kol').AsFloat *
adoTable1.FieldByName('Cena').AsFloat;
end;
```



4.4. Добавим кнопки навигации и манипулирования данными таблицы **PRODUCTS.** Добавьте на форму 8 компонентов Button.

Для кнопки *Первая запись* напишите обработчик события для перехода к первой записи:

```
datamodule2.ADOTable1.First;
```

Аналогично распишите обработчики для кнопок Предыдущая запись (`datamodule2.ADOTable1.`**Prior**), Следующая запись (`datamodule2.ADOTable1.`**Next**), Последняя запись (`datamodule2.ADOTable1.` **Last**).

Процедуры для кнопок Добавить, Изменить, Удалить, Сохранить напишите самостоятельно.

Запустите приложение, проверьте его работу.

Контрольные вопросы

1. Навигация в наборе данных
2. Манипулирование данными в наборе
3. Доступ к полям.

Задание для самостоятельного выполнения

Спроектируйте базу данных *Access*, состоящую из нескольких таблиц по задаче выданного варианта. Спроектируйте проект, обеспечивающий работу с созданной базой данных.

При формировании набора данных постарайтесь добавить хотя бы одно вычисляемое поле.

Вариант 1.

Сведения об участниках конкурса бальных танцев. База данных должна содержать следующую информацию: фамилию, имя, отчество участника, город, фамилию тренера, оценки за каждый танец.

Вариант 2.

Сведения об успеваемости студентов. База данных должна содержать следующую информацию: фамилию, имя, отчество студента, номер группы, в которой обучается студент, название учебной дисциплины, номер задания, коэффициент сложности, оценку данного студента по данной дисциплине за данное задание от 0 до 1 (как доля сделанной работы).

Вариант 3.

Учет изделий, собранных в цехе за неделю. База данных должна содержать следующую информацию: фамилию, имя, отчество сборщика, количество изготовленных изделий за каждый день недели раздельно, название цеха, а также тип изделия и его стоимость.

Вариант 4.

Сведения об ассортименте игрушек в магазине. База данных должна содержать следующую информацию: название игрушки, ее цену, количество, возрастную категорию детей, для которых она предназначена, а также название фабрики и города, где изготовлена игрушка.

Вариант 5.

Учет рейтинга теннисистов за 5 лет. Каждая запись содержит поля: фамилия, имя, отчество спортсмена, пол, год рождения, фамилия, имя, отчество тренера, названия стран и пять полей с рейтингом.

Вариант 6.

Сведения о рейсах Аэрофлота. База данных должна содержать следующую информацию: номер рейса, пункт назначения, время вылета, время прибытия, количество свободных мест, тип самолета и его вместимость.

Вариант 7.

Сведения об ассортименте обуви в магазине. База данных должна содержать следующую информацию: артикул, наименование обуви, количество пар, стоимость одной пары, имеющиеся размеры, название фабрики и срок поставки обуви в магазин.

Вариант 8.

Учет лекарств в аптеке. База данных должна содержать следующую информацию: наименование лекарства, стоимость одной единицы, количество единиц, дату изготовления, срок годности, а также название фабрики, где производится данное лекарство, ее адрес.

Вариант 9.

Сведения о ветеранах спорта. Ассоциация ветеранов спорта проводит Всероссийские соревнования ветеранов. Для организации соревнований составляются списки участников, которые используются для размещения спортсменов в гостиницах. Для каждого спортсмена указывается гостиница, номер комнаты и количество мест в комнате. Для нужд самой ассоциации ветеранов спорта необходимо хранить информацию следующего вида: фамилию, имя, отчество спортсмена, возрастную группа, название города и вид спорта.

Вариант 10.

Учет рождаемости в роддоме. База данных должна содержать следующую информацию: фамилию, имя, отчество матери, пол ребенка, его вес, рост и дату рождения ребенка, а также ФИО лечащего врача и номер палаты, в которой находится мать ребенка.

Вариант 11.

Сведения об обучающихся на курсах повышения квалификации. База данных должна содержать следующую информацию: фамилию, имя, отчество слушателя, его пол и адрес, тип организации (коммерческая, государственная и т. д.), наименование организации, должность слушателя и оценки по прослушанным дисциплинам (маркетинг, финансы и кредит) для каждого слушателя.

Вариант 12.

Сведения о размере стипендии студента. База данных должна содержать следующую информацию: фамилию, имя, отчество студента, группу, адрес, по которому проживает студент, размер стипендии, а также фамилию преподавателя, читаемую дисциплину, дату экзамена и оценку, полученную студентом.

Вариант 13.

Учет поступления больных. База данных должна содержать следующую информацию: фамилию, имя, отчество больного, его пол, дату рождения, дата поступления, начальный диагноз, степень тяжести состояния больного, номер палаты, куда был помещен больной, и фамилию, имя, отчество лечащего врача.

Вариант 14.

Учет призывников. База данных должна содержать следующую информацию: фамилию, имя, отчество призывника, год его рождения, адрес, семейное положение, дату прохождения медкомиссии, заключение о пригодности к службе.

Вариант 15.

Учет золотых изделий в ювелирном магазине. База данных должна содержать следующую информацию: вид изделия, его вес, пробу, дату поступления и стоимость, а также фамилию, имя, отчество мастера-изготовителя, стаж его работы и разряд (1,2,3).

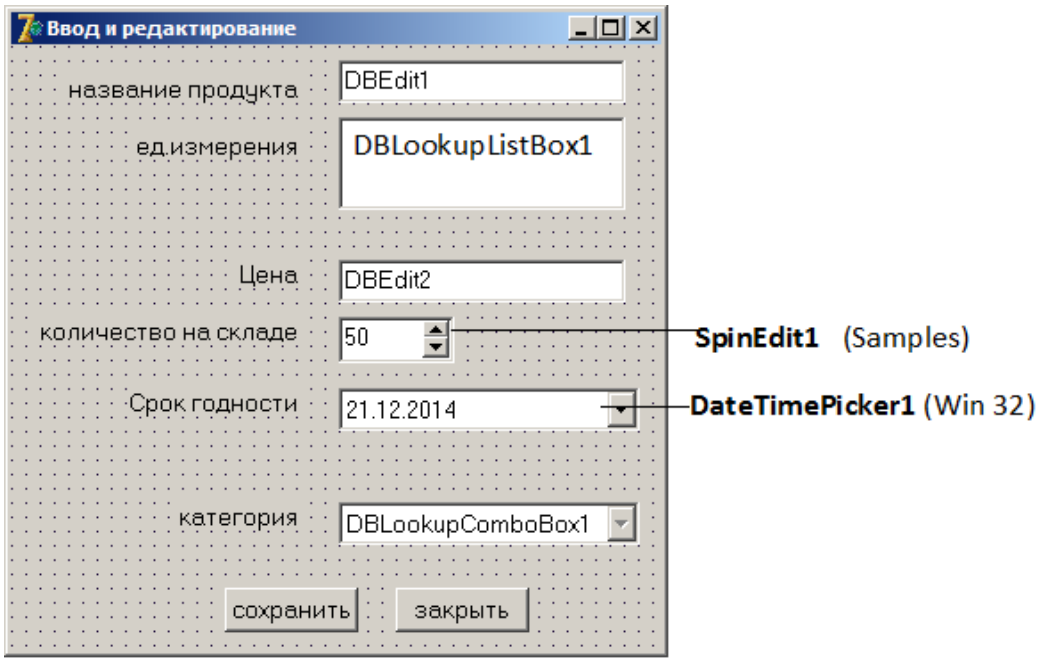
Лабораторная работа № 3

Тема: Разработка форм с использование различных компонентов отображения данных

Цель работы: Получить навыки использования разнообразных компонент отображения данных в проекте.

Задание 5. Разработка формы для ввода и редактирования данных

Создайте новую форму form5. Расположите на ней компоненты согласно приведенному рисунку.



Для компонентов установите следующие значения приведенных в таблице свойств.

Компонент	Свойство	Значение
DBEdit1	DataSource	DataModule2.DataSource1
	DataField	name
DBLookupListBox1	DataSource	DataModule2.DataSource1
	DataField	id_ed
	ListSource	DataModule2.DataSource2
	ListField	ed_name
	KeyField	id
DBEdit2	DataSource	DataModule2.DataSource1
	DataField	cena
SpinEdit1	Value	50
DBLookupComboBox1	DataSource	DataModule2.DataSource1
	DataField	id_kat
	ListSource	DataModule2.DataSource3
	ListField	Kat_name
	KeyField	id

Обработчик события OnClick для кнопки Закрыть:

```
form5.Close;
```

Обработчик события OnClick для кнопки Сохранить:

```
with datamodule2  do
```

```
begin
adotable1Kol.AsInteger:=spinedit1.value;
adotable1srok.AsDateTime:=datetimepicker1.Date;
adotable1.Post;
end;
```

Вернитесь на главную форму **Склад**. Обработчик события для кнопки Добавить преобразуйте следующим образом:

```
datamodule2.ADOTable1.Append;
form5.showmodal; {для ввода информации о новой записи открываем
модально форму 5}
```

Запустите приложение, проверьте его работу. Добавьте в таблицу несколько записей.

Задание для самостоятельного выполнения к лабораторной работе 3

В проект, созданный по заданию лабораторной работы №2, добавьте необходимое количество форм (по одной для каждой таблицы), которые будут использоваться при вводе и редактировании данных.

Для ввода и отображения данных ваших таблиц постарайтесь использовать различные компоненты.

Контрольные вопросы

1. Классификация компонентов отображения данных
2. Компоненты представление отдельных полей
3. Компоненты представление нескольких полей
4. Компоненты синхронного просмотра

Лабораторная работа 4

Поиск записей. Сортировка и фильтрация записей в наборе данных.

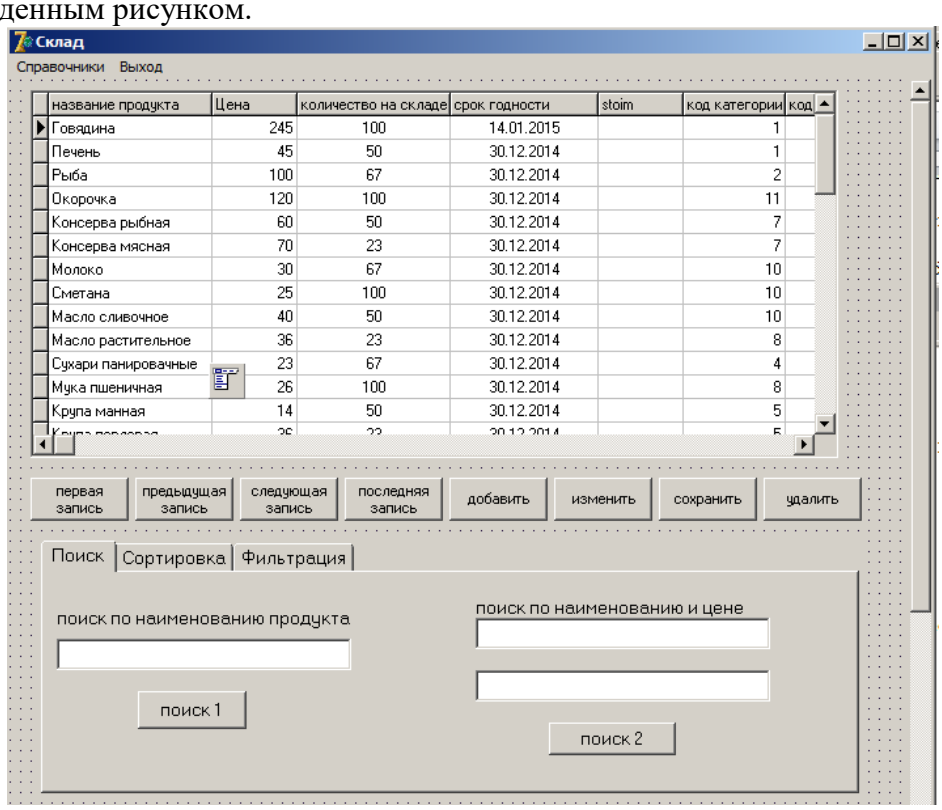
Задание 1. Организовать поиск записей по одному или нескольким полям.

Разместите на форме компонент PageControl1 (Win32). Выбирая пункт New Page из контекстного меню для компонента PageControl1, создайте для него три вкладки.

Выделите вкладку TabSheet1 и в его свойство caption впишите заголовок: *Поиск*.

Таким же образом озаглавьте две остальные вкладки, дав им названия: *Сортировка* и *Фильтрация.*

Выделите вкладку Поиск и поместите на ней компоненты edit1, edit2, edit3 и две кнопки, в соответствии с приведенным рисунком.



Для кнопки Поиск1 напишите обработчик события OnClick:

```
    if not datamodule2.ADOTable1.Locate('name',edit1.Text,[]) then
ShowMessage('Информация не найдена!');
```

Для кнопки Поиск2 напишите обработчик события OnClick:

```
if not datamodule2.ADOTable1.Locate('Name;Cena',VarArrayOf(
[edit2.text,edit3.text]),[]) then ShowMessage('Информация не найдена!');
```

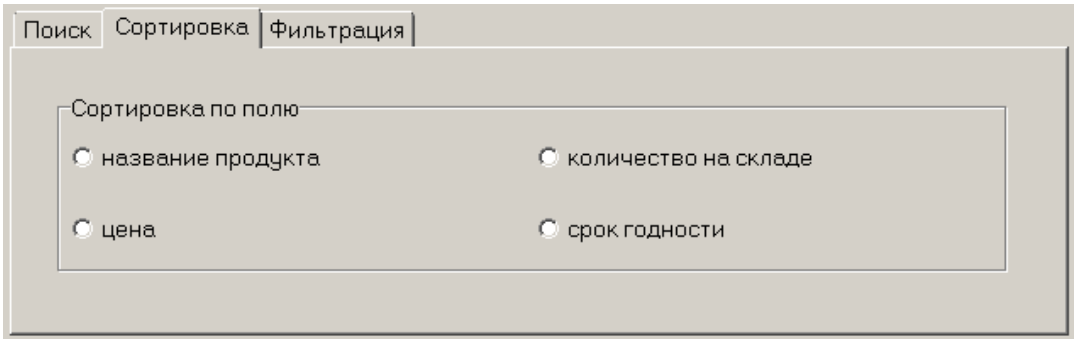
Задание 2. Организовать сортировку записей в таблице.

Активизируйте вкладку Сортировка.

Поместите на ней компонент RadioGroup1, выделив его свойство Items, добавьте в него пункты

название продукта
цена
количество на складе
срок годности

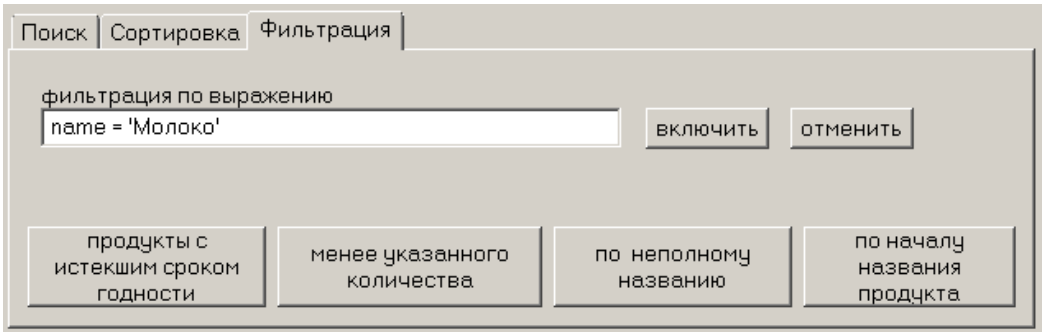
Выделите свойство Columns и присвойте ему значение 2.



Для RadioGroup1 напишите обработчик события OnClick:

```
procedure TForm1.RadioGroup1Click(Sender: TObject);
begin
case radiogroup1.ItemIndex of
0: datamodule2.ADOTable1.Sort:='name';
1: datamodule2.ADOTable1.Sort:='cena';
2: datamodule2.ADOTable1.Sort:='kol';
3: datamodule2.ADOTable1.Sort:='srok';
4: datamodule2.ADOTable1.Sort:='stoim';
end;
end;
```

Задание 3. Организовать фильтрацию записей набора данных.
Активизируйте вкладку Фильтрация.
Поместите на ней компоненты в соответствии с рисунком



Для всех кнопок должны быть написаны обработчики события OnClick.
Для кнопки Отменить:

```
datamodule2.ADOTable1.Filtered:=false;
```

Для кнопки Включить:

```
datamodule2.ADOTable1.Filter:=edit4.Text;
datamodule2.ADOTable1.Filtered:=true;
```

Сохраните и запустите проект. Проверьте, как будет отфильтрована таблица при следующих значениях фильтра:

name = 'Молоко' or name = 'Сметана'
srok<30.12.2014
name LIKE 'M%'

Для кнопки Продукты с истекшим сроком годности:

```
var s:string;
begin
s:=datetostr(now); //запоминаем текущую дату, преобразованную в строку
```

```
    datamodule2.ADOTable1.Filter:=' srok < '+#39+s+#39;
    datamodule2.ADOTable1.Filtered:=true;
```

Для кнопки Менее указанного количества:

```
    var s:string;
    begin
    InputQuery(' ввод данных ','введите количество', s);
     datamodule2.ADOTable1.Filter:=' kol < '+#39+s+#39;
    datamodule2.ADOTable1.Filtered:=true;
```

Для кнопки По неполному названию

```
    var s:string;
    begin
    InputQuery(' ввод данных ',' введите название, вместо пропущенных
частей слова вставьте *',s);
     datamodule2.ADOTable1.Filter:=' name Like '+#39+s+#39;
    datamodule2.ADOTable1.Filtered:=true;
```

Для кнопки По началу названия

```
    var s:string;
    begin
    InputQuery(' ввод данных ','введите название',s);
     datamodule2.ADOTable1.Filter:=' name Like '+QuotedStr(s+'%');
    datamodule2.ADOTable1.Filtered:=true;
    end;
```

Задание для самостоятельного выполнения к лабораторной работе 4

В проект, созданный по заданию лабораторных работ №№ 2, 3, добавьте объекты, которые будут использоваться для осуществления:

- Поиска информации по различным критериям;
- Сортировки записей по различным критериям;
- Фильтрации записей по различным критериям.

Контрольные вопросы

1. Поиск записей. Метод **Locate**
2. Сортировка набора данных. Свойство **Sort**
3. Фильтрация записей. Свойство **Filter**

Лабораторная работа 5
Работа со связанными таблицами

Задание 1. Добавление таблиц в существующую базу данных.

Запустите MS Accecss откройте созданную ранее базу с названием sklad.accdb. В базу добавьте две таблицы: Накладная и Состав накладной, структуры которых показаны на картинках.

Накладная

nakladnaya

Имя поля	Тип данных	
nomer	Счетчик	номер накладной
data	Дата/время	дата выписки накладной
summa	Денежный	сумма накладной

Состав накладной

sostav_nakladnoy

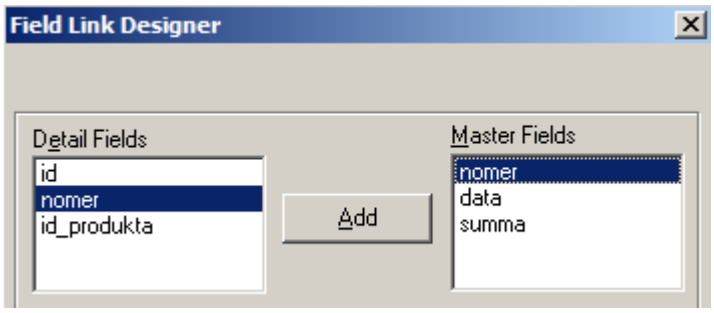
Имя поля	Тип данных	
id	Счетчик	код позиции
nomer	Числовой	номер накладной
id_produkta	Числовой	код продукта
kol	Числовой	количество продукта

Откройте модель datamodule2. Добавьте м модуль компоненты:

ADOTable4 и DataSource4, свяжите их с таблицей Накладная;

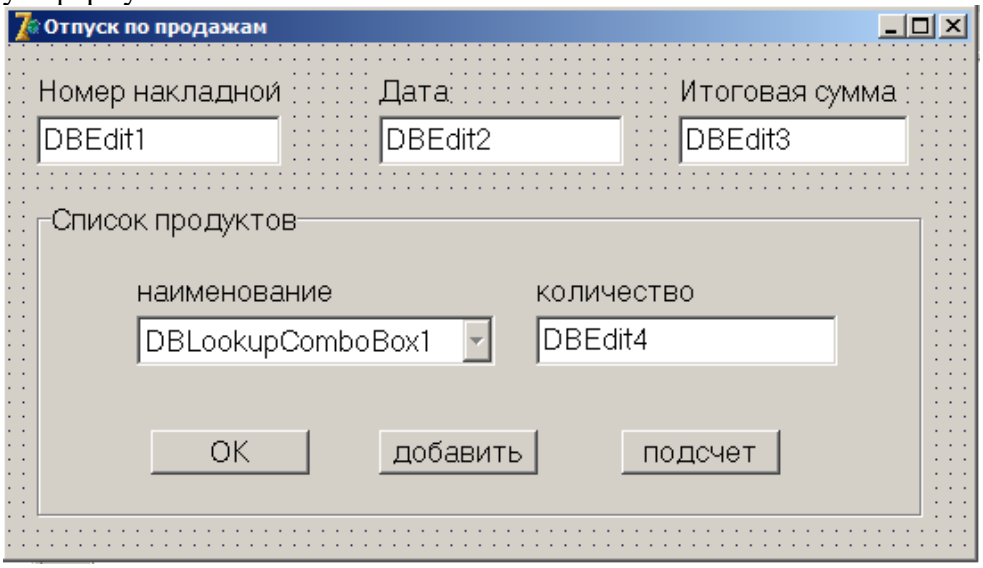
ADOTable5 и DataSource5, свяжите их с таблицей Состав накладной.

Для того чтобы связать эти таблицы по номеру накладной, в компоненте ADOTable5 свойству **MasterSource** присвойте значение DataSource4. Через свойство **MasterFeild** откройте редактор **Field Link Designer** выберите поля nomer в главной и подчиненной таблицах и кнопкой **Add** создайте связь между таблицами.



Задание 2.Создание формы для ввода данных в Накладные

Создайте новую форму Form6 и поместите на ней показанные компоненты.



В ориентированных на базы данных компонентах измените перечисленные в таблице свойства.

Компонент	Свойство	Значение
DBEdit1	DataSource	DataModule2.DataSource4
	DataField	nomer
DBEdit2	DataSource	DataModule2.DataSource4
	DataField	data
DBEdit3	DataSource	DataModule2.DataSource4
	DataField	summa
DBEdit4	DataSource	DataModule2.DataSource5
	DataField	kol
DBLookupComboBox1	DataSource	DataModule2.DataSource5
	DataField	id_produkta
	ListSource	DataModule2.DataSource1
	ListField	name
	KeyField	id

Создайте обработчики событий OnClick для кнопок:

ОК (сохранение введенной информации):

```
    datamodule2.adotable5.Post;
    {проданное количество необходимо вычесть от общего количества продукта
на складе}
    datamodule2.adotable1.Edit;
    datamodule2.adotable1.fieldbyname('kol').asfloat:=
    datamodule2.adotable1.fieldbyname('kol').asfloat -
datamodule2.adotable5.fieldbyname('kol').asfloat;
    datamodule2.adotable1.Post;
```

Добавить (добавление новой позицию в накладную)

```
    datamodule2.adotable5.Append;
```

Подсчет (расчет итоговой суммы проданной продукции)

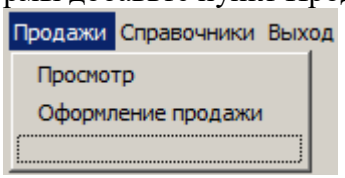
```
    var s:double;
    begin
    datamodule2.adotable5.First;   //переход к первой записи
    s:=0;//обнуляем значение суммы
    while not datamodule2.adotable5.eof do    //пока не конец таблицы
    begin
    {ищем в таблице Продукты выбранную позицию, указанную там цену
умножаем на указанное количество и добавляем к сумме}

   datamodule2.ADOTable1.Locate('id',datamodule2.ADOTable5.FieldValues['id
_produkta'],[]);
    s:=s+datamodule2.adotable5.fieldbyname('kol').asfloat*datamodule2.adot
able1.fieldbyname('cena').asfloat;
    datamodule2.adotable5.Next;//переход к следующей записи
    end;
    {подсчитанную сумму записываем в таблицу Накладные}

    datamodule2.adotable4.Edit;
    datamodule2.adotable4.FieldByName('summa').asfloat:=s;
    datamodule2.adotable4.post;
    end;
```

Откройте главную форму.

В компонент MainMenu1 главной формы добавьте пункт Продажи:



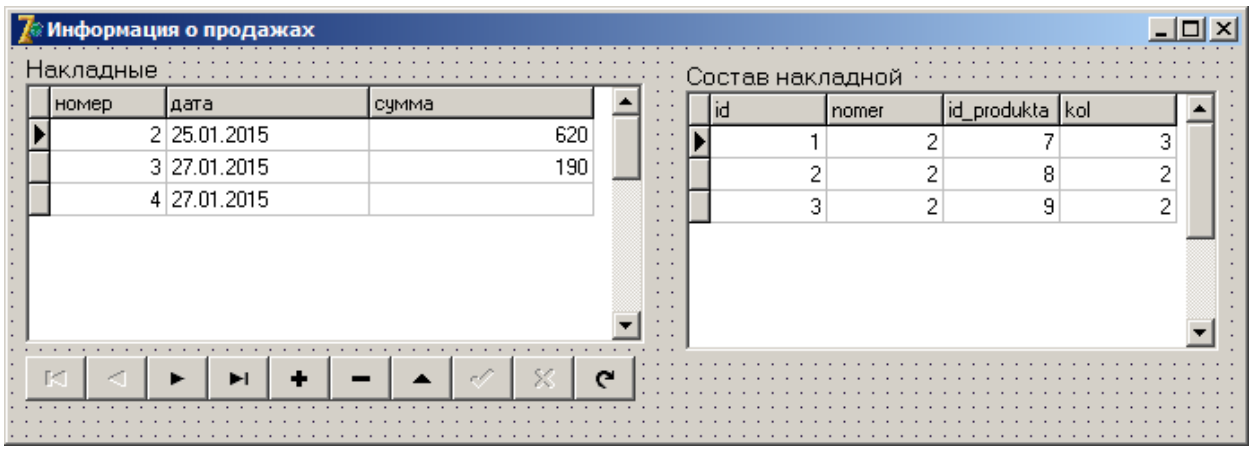
В пункт Оформление продажи впишите код обработчика события OnClick:

```
    datamodule2.adotable4.Append;
    {в таблицу Накладная добавляем новую запись и в поле дата записываем
текущую дату}
    datamodule2.adotable4.FieldValues['data']:=formatdatetime('dd.mm.yyyy'
,now);
    datamodule2.adotable4.Post;
    form6.showmodal; //открываем модально форму 6 - Отпуск по продажам
```

Запустите приложение и проверьте его работу.

Задание 3. Создание формы для просмотра данных

Создайте новую форму Form7 и разместите на ней компоненты:



Компонент DBGrid1 свяжите с источником данных DataSource4, а DBGrid2 - с источником данных DataSource5.

В компоненте DBGrid1 названия полей перепишите на русском языке.

Компонент DBNavigator1 свяжите с источником данных DataSource4.

Вернитесь на главную форму. В пункт меню Продажи – Просмотр впишите обработчик:

```
    form7.show;
```

Запустите приложение и проверьте его работу.

Оформите несколько накладных с разным перечнем и количеством продуктов, рассчитайте итоговую сумму. Просмотрите проделанные операции в Форме Информация о продажах.

Контрольные вопросы

1. Связывание таблиц при проектировании приложения.
2. Свойства MasterSource и MasterFeild
3. Доступ к полям набора данных.

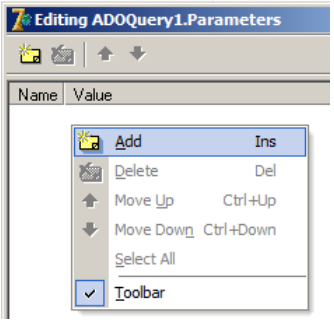
Лабораторная работа 6
Компонент запроса AdoQuery

Задание 1. Разработать запрос для просмотра информации о проданных товарах

На форме 7 разработанного проекта, компонент DBGrid2 отображает информацию о составе накладной на отпуск товаров (то есть список проданных товаров). Однако вся информация показана в кодах. Чтобы этого избежать, выполним следующее.

1. Расположите на форме DataModule2 компонент AdoQuery (вкладка Ado). В свойстве Connection укажите ADOConnection1. Добавьте на форму также компонент DataSource6, свойство DataSet которого укажите равным AdoQuery1.
2. Создайте новую форму Form8, подключите к модуль 2. Поместите на форме компонент DBGrid1, его свойство DataSource укажите равным DataModule2.DataSource6.
3. Запрос должен содержать данные из таблиц Продукты, Накладная и Состав накладной. Доля того чтобы обеспечить вывод только тех записей, которые соответствуют номеру текущей накладной, нам понадобится параметр, в который будет записываться указанный номер. Назовем этот параметр N.

Для добавления параметра вберите свойство Parameters компонента AdoQuery1.



В открывшемся окне вызовите контекстное меню и командой **Add** добавьте новый параметр. Перейдите в активизировавшееся окно инспектора объектов и укажите в свойстве **name** для параметра его имя – **N**. В результате параметр должен появиться в редакторе параметров запроса, который уже можно закрыть.

4. Для формирования запроса, используя структурированный язык запросов SQL, через свойство SQL компонента AdoQuery1, откройте редактор строк и занесите в него текст:

```
SELECT DISTINCT sostav_nakladnoy.nomer as [номер накладной],
products.name as [продукт] ,
products.cena as [цена],
sostav_nakladnoy.kol as [количество] ,
products.cena*sostav_nakladnoy.kol as [стоимость]
FROM sostav_nakladnoy, products
WHERE  (sostav_nakladnoy.id_produkta=products.id)
and  (sostav_nakladnoy.nomer= :N)
```

5. Для формы Form8 в обработчик события OnClose впишите команду:

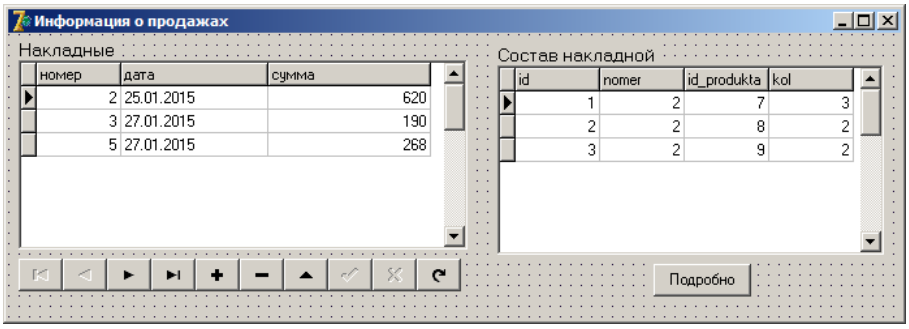
```
datamodule2.ADOQuery1.close;
```

Запрос каждый раз вместе с формой будет «закрываться», для того чтобы он при каждом открытии формы обновлялся.

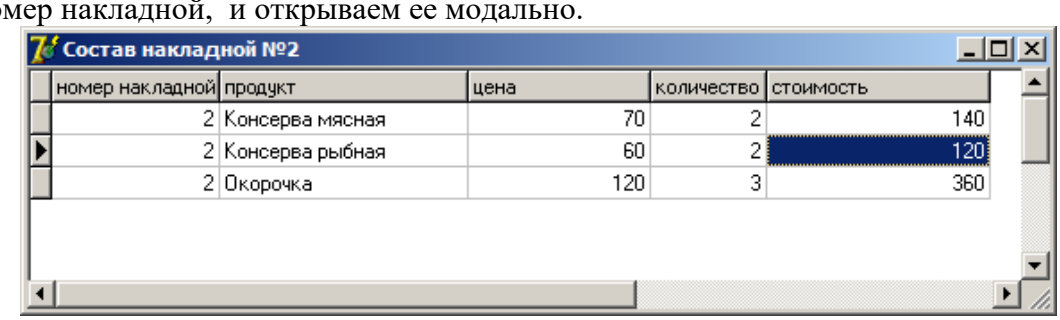
6. Вернитесь на форму Form7. Добавьте компонент Button1.

В обработчик события On Click кнопки **Подробно** впишите код:

```
datamodule2.adoquery1.Parameters.ParamByName('n').value:=
          datamodule2.ADOTable4.fieldbyname('nomer').asstring;
datamodule2.ADOQuery1.open;
form8.Caption:='Состав накладной'
          +datamodule2.ADOTable4.fieldbyname('nomer').asstring;
form8.showmodal;
```



Здесь мы вначале передаем значение параметра N, активизируем запрос, затем меняем заголовок Form8, указав номер накладной, и открываем ее модально.

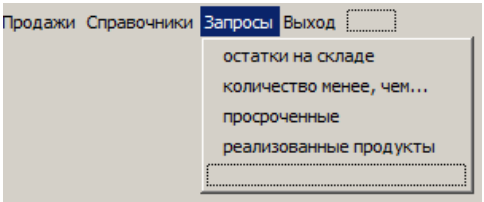


Задание 2. Формирование запросов к базе данных

На форму datamodule2 добавьте компоненты:

- adoQuery2, в свойстве Connection укажите ADOConnection1
- datasource7, в свойстве DataSet укажите adoQuery2.

Добавьте в компонент MainMenu1 главной формы раздел Запросы:



К созданным пунктам допишите обработчики событий OnClick.

Остатки на складе

```
begin
with datamodule2 do
begin
ADOQuery2.SQL.Clear;
adoQuery2.sql.add('SELECT products.name as [продукт], products.kol as
[количество], ed_izmer.ed_name as [единица измерения] ');
adoQuery2.sql.add(' FROM   products, ed_izmer WHERE
ed_izmer.id=products.id_ed');
form8.DBGrid1.DataSource:=datasource7;
Adoquery2.Open;
form8.Caption:='Остатки на складе ';
form8.showmodal;
end;
end;
```

Количество менее, чем …

```
var s:string;
begin
with datamodule2 do
begin
```

```
    InputQuery('Ввод данных','Введите количество ', s);
    ADOQuery2.SQL.Clear;
    adoQuery2.sql.add('SELECT products.name as [продукт], products.kol as
[количество], ed_izmer.ed_name as [единица измерения] ');
    adoQuery2.sql.add(' FROM   products, ed_izmer WHERE
ed_izmer.id=products.id_ed and products.kol <' +s);
    form8.DBGrid1.DataSource:=datasource7;
    Adoquery2.Open;
    form8.Caption:='Количество на складе меньше '+s;
    form8.showmodal;
    end;
    end;
```

Просроченные

```
    var s:string; SDate : TDateTime;
    begin
    InputQuery('Ввод данных','Введите количество дней до окончания срока
годности', s);
    sdate:=incday(now,strtoint(s));
    datamodule2.ADOTable1.Filter:=' srok < '+#39+datetostr(sdate)+#39;;
    datamodule2.ADOTable1.Filtered:=true;
```

Реализованные продукты

```
    var sum:real;
    begin
    with datamodule2 do
    begin
    {формируем запрос, в который выбираются проданные продукты}
    ADOQuery2.SQL.Clear;
    adoQuery2.sql.add('SELECT DISTINCT sostav_nakladnoy.nomer as [номер
накладной],   products.name as [продукт]   ,  products.cena as [цена], ');
    adoQuery2.sql.add(' sostav_nakladnoy.kol as [количество] ,
products.cena*sostav_nakladnoy.kol as [стоимость]');
    adoQuery2.sql.add(' FROM sostav_nakladnoy, products WHERE
(sostav_nakladnoy.id_produkta=products.id)');
    form8.DBGrid1.DataSource:=datasource7;
    Adoquery2.Open;
    adoquery2.First;
    {подсчитываем вырученную в итоге сумму}
    sum:=0;
    while not adoquery2.Eof do
      begin
      sum:=sum+adoquery2.fieldbyname('стоимость').asfloat;
      adoquery2.Next;
      end;
    {выводим подсчитанную сумму в заголовок формы}
    form8.Caption:='Всего продано на сумму: '+floattostr(sum)+' руб.';
    form8.showmodal;
    end;
    end;
```

Запустите приложение, проверьте его работу.

Контрольные вопросы

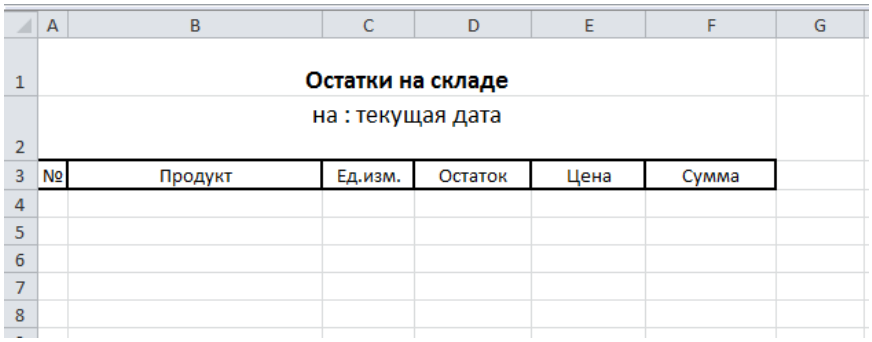
1. Назначение компонента AdoQuery.
2. Свойства компонента AdoQuery.
3. Формирование запросов.

Лабораторная работа 7

Разработка документов и приложений MS Word и MS Excel в Delphi. Создание отчетов

Задание 1. Разработать процедуру для вывода на печать информации об остатках на складе.

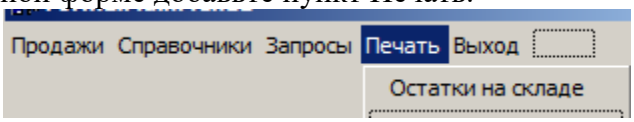
1. Откройте табличный процессор Microsoft Excel 2010. Создайте в нем табличный документ, состоящий из одного листа, со структурой, показанной на рисунке.



Сохраните созданный документ под именем **ostatki.xlsx** в папке, где хранится проект ваш проект.

2. Откройте Delphi - проект.

К главному меню на главной форме добавьте пункт Печать.



3. К модулю Unit1 подключите стандартный модуль comobj, прописав его имя в разделе Uses.

```
uses
```

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, Grids, DBGrids, Menus, StdCtrls, ComCtrls, Mask, DBCtrls, dbcgrids, ExtCtrls, dateUtils, **comobj**;

В разделе описаний переменных опишите переменные **excel, word:**

```
var
  Form1: TForm1;
  Excel, Word:variant;
```

4. В обработчик события OnClick для пункта меню **Печать-Остатки на складе** впишите код:

```
procedure TForm1.N12Click(Sender: TObject);
var i:integer;

begin
//создаем приложение excel
Excel := CreateOleObject('Excel.Application');
{указываем шаблон документа, в котором будем создавать отчет,
ExtractFilePath(ParamStr(0) – путь к каталогу, из которого запускается программа}
Excel.Workbooks.Open(ExtractFilePath(ParamStr(0)) + 'ostatki.xlsx');
//открываем Excel
Excel.Visible := True;
//вписываем в отчет текущую дату
excel.range['a2']:='на '+formatdatetime('dd.mm.yyyy',date);
```

```
i:=0;
datamodule2.ADOTable1.First;
while not  datamodule2.ADOTable1.Eof do
    begin
    inc(i);
    excel.range['a'+inttostr(3+i)]:=i;
    excel.range['b'+inttostr(3+i)]:=datamodule2.ADOTable1.FieldByName('name').asstring;
    datamodule2.ADOTable2.Locate('id',datamodule2.ADOTable1.FieldValues['id_ed'],[]);
    excel.range['c'+inttostr(3+i)]:=datamodule2.ADOTable2.FieldByName('ed_name').asstring;
    excel.range['d'+inttostr(3+i)]:=datamodule2.ADOTable1.FieldByName('kol').asinteger;
    excel.range['e'+inttostr(3+i)]:=datamodule2.ADOTable1.FieldByName('cena').asfloat;
    excel.range['f'+inttostr(3+i)]:=datamodule2.ADOTable1.FieldByName('kol').asinteger*
                                    datamodule2.ADOTable1.FieldByName('cena').asfloat;
    datamodule2.ADOTable1.next;
    end;
 //рисуем границы для заполненного диапазона
Excel.Range['A3:' + 'F' + IntToStr(i+3)].Borders.LineStyle := 1;     //стиль линии
Excel.Range['A3:' + 'F' + IntToStr(i+3)].Borders.Weight := 2;        //толщина линии
Excel.Range['A3:' + 'F' + IntToStr(i+3)].Borders.ColorIndex := 1;  //код цвета линии
 end;
```

5. Сохраните и запустите проект. Проверьте его работу. Создаваемый документ не сохранят\йте или сохраните под другим именем, чтобы ваш шаблон остался «чистым».

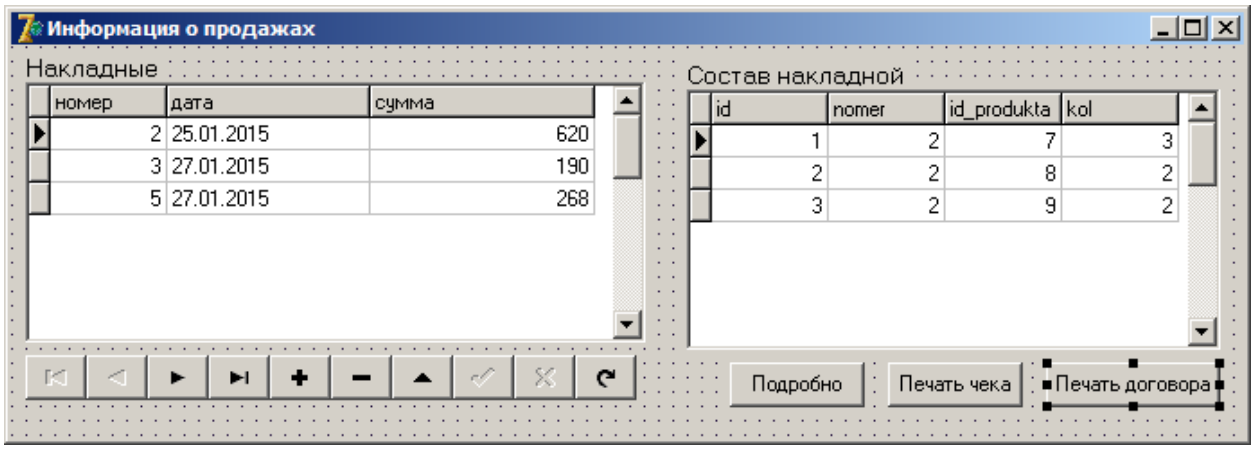
Задание 2. Разработать процедуру для вывода на печать Счета к оплате.

1. Создайте новый документ Excel по приведенной ниже структуре и сохраните его по именем **schet.xlsx**

	A	B	C	D	E	F
1		**ООО "Программка"**				
2						
3		**010101, г.Сайт, ул.Баз данных, дом 256**				
4						
5						
6		ИНН 0707078934564		КПП 45675		
7		Получатель				
8		ООО "Программка"		Сч. №	45764574543456	
9		Банк получателя		БИК	4586	
10		"Сбербанк"		Сч. №	38465346538437	
11						
12						
13						
14						
15			**Перечень товаров**			
17						
18		№	Наименование товара	Цена	Коли-чество	Сумма
19						
20						
21						

2.Откройте форму Firm7 и поместите на ней кнопки **Печать чека** и **Печать договора**.

Как и в предыдущем примере к модулю Unit7 подключите стандартный модуль **comobj**, прописав его имя в разделе Uses. Также опишите глобальную переменную **excel:variant**.



3. В обработчик события OnClick для кнопки Печать чека впишите код:

```
procedure TForm7.Button2Click(Sender: TObject);
var i:integer;
s:real;

begin
{активизируем запрос, в для просмотра информации о проданной продукции, разработанный в лабораторной работе 6}
datamodule2.adoquery1.Parameters.ParamByName('n').value:=datamodule2.ADOTable4.fieldbyname('nomer').asstring;
datamodule2.ADOQuery1.open;
{создаем объект  Excel }
Excel := CreateOleObject('Excel.Application');
{открываем документ Excel }
Excel.Workbooks.Open(ExtractFilePath(ParamStr(0)) + 'schet.xlsx');

{открываем Ecxel}
Excel.Visible := True;
{обнуляем начальные значения номера строки и суммы}
i:=0; s:=0;
datamodule2.ADOQuery1.First;
{открываем цикл по записям запроса}
 while not  datamodule2.ADOQuery1.Eof do
 begin
  inc(i);
excel.range['b'+inttostr(18+i)]:=i;
  excel.range['c'+inttostr(18+i)]:=datamodule2.ADOQuery1.Fields[1].asstring;
  excel.range['d'+inttostr(18+i)]:=datamodule2.ADOQuery1.Fields[2].asstring;
  excel.range['e'+inttostr(18+i)]:=datamodule2.ADOQuery1.Fields[3].asstring;
  excel.range['f'+inttostr(18+i)]:=datamodule2.ADOQuery1.Fields[4].asstring+' руб.';
  s:=s+datamodule2.ADOQuery1.Fields[4].asfloat;
  datamodule2.ADOQuery1.next;
  end;
{переходим к следующей строке, выписываем подсчитанную сумму}
 inc(i);
 excel.range['e'+ IntToStr(i+18)]:='ИТОГО';
```

```
excel.range['f'+ IntToStr(i+18)]:=floattostr(s)+' руб.';

{чертим рамки для заполненных ячеек таблицы}
Excel.Range['b19:' + chr(ord('f')) + IntToStr(i+18)].Borders.LineStyle := 1;
 Excel.Range['b19:' + chr(ord('f')) + IntToStr(i+18)].Borders.Weight := 2;
 Excel.Range['b19:' + chr(ord('f')) + IntToStr(i+18)].Borders.ColorIndex := 1;

{выписываем дату продажи}
excel.range['b12']:=СЧЕТ № '+datamodule2.ADOQuery1.Fields[0].asstring;
 excel.range['d'+ IntToStr(i+21)]:='Дата: '+
            formatdatetime('dd.mm.yyyy',datamodule2.ADOTable4.FieldValues['data']);

{выписываем области для подписей}
excel.range['d'+ IntToStr(i+23)]:='Покупатель _____';
 excel.range['d'+ IntToStr(i+25)]:='Продавец    _____';
 end;
```

4. Сохраните проект и проверьте его работу.

Задание 3. Разработать процедуру для вывода на печать Договора купли-продажи.

1. В текстовом процессоре Word 2010 создайте документ dogovor.docx и сохраните его в папке проекта.

ДОГОВОР № #номер#
КУПЛИ-ПРОДАЖИ ОПТОВОЙ ПАРТИИ ТОВАРА

г. Нальчик #**дата**#

ООО "Программка", именуемый в дальнейшем «Продавец», в лице Компиляторова А.С., и #**покупатель**#, именуемым в дальнейшем «Покупатель», с другой стороны, заключили настоящий Договор о нижеследующем:

1. ПРЕДМЕТ ДОГОВОРА

1.1. Продавец обязуется поставить и передать в собственность Покупателю товар, а Покупатель обязуется принять товар и оплатить его на условиях Настоящего Договора.

1.2. Развернутый ассортимент поставляемого товара определяется в Приложении № 1 к Настоящему Договору.

2. ЦЕНА

2.1. Общая цена по Настоящему Договору: #**цена**# **руб.**

3. СРОК ДЕЙСТВИЯ НАСТОЯЩЕГО ДОГОВОРА

3.1. Настоящий Договор вступает в силу с момента подписания его сторонами и действует до момента его окончательного исполнения.

4. ОТВЕТСТВЕННОСТЬ СТОРОН

4.1. За нарушение условий Настоящего Договора виновная сторона возмещает причиненные убытки, в том числе неполученную прибыль, в порядке, предусмотренном действующим законодательством.

Покупатель _____ / #**покупатель**#/

Продавец _____ / Компиляторова А.С/

2. Откройте форму form7. В разделе implementation опишите функцию poisk_zamena, которая будет фрагмент документа Word заменять предлагаемым программой текстом.

```
{$R *.dfm}

 function poisk_zamena(FindText, ReplacementText: String): Boolean;
  const wdReplaceAll=2;
begin
  Word.Selection.Find.Text:=FindText;
  Word.Selection.Find.Replacement.Text:=ReplacementText;
  poisk_zamena:=Word.Selection.Find.Execute(Replace:=wdReplaceAll);
end;
```

3. В обработчик события OnClick для кнопки Печать договора впишите код:

```
procedure TForm7.Button3Click(Sender: TObject);
var s:string;
begin

    {Вводим информацию о покупателе}
  s:=InputBox('Ввод данных', 'Введите покупателя', 'Основной покупатель');

    {создаем объект Word}
  Word:=CreateOleObject('Word.Application');
  Word.Visible:=true;

    {открываем документ dogovor.docx в режиме «только чтение»}
  Word.Documents.Open(ExtractFilePath(ParamStr(0)) + 'dogovor.docx',
readonly:=true);

    {заменяем требуемые фрагменты текста «оперативной» информацией или информацией из базы данных }
  poisk_zamena('#номер#',
                    datamodule2.ADOTable4.FieldByName('nomer').AsString);
  poisk_zamena('#дата#',
                    datamodule2.ADOTable4.FieldByName('data').AsString);
  poisk_zamena('#покупатель#', s);
  poisk_zamena('#цена# ',
                    datamodule2.ADOTable4.FieldByName('summa').AsString);
end;
```

4. Сохраните проект и проверьте его работу.

Контрольные вопросы

1. COM-объекты.
2. Создание приложения MS Word
3. Замена текста в документе Word
4. Создание приложения MS Excel
5. Импорт данных из базы данных в документ Excel

Лабораторная работа 8

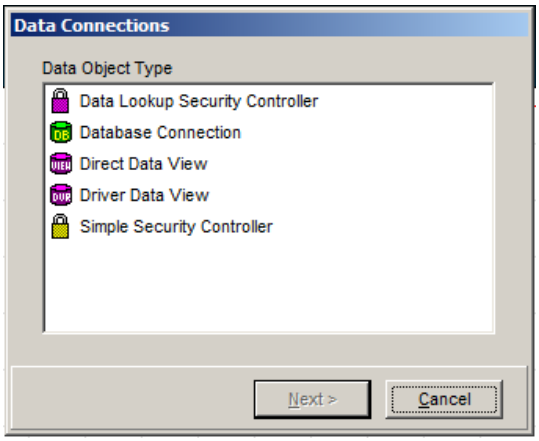
Разработка отчетов RaveReports

Задание 1. Разработать проект отчета для вывода на печать документа Прайс-лист.

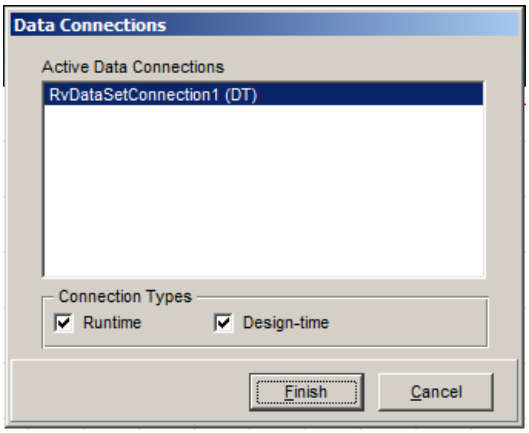
1. Разместить на форме datamodule2 компонент RvDataSetConnection1 , в свойстве DataSet укажите ADOTable1.

2. Создать компонент RvProject: TRvProject . Двойным щелчком мыши на этом компоненте вызвать визуальный дизайнер.

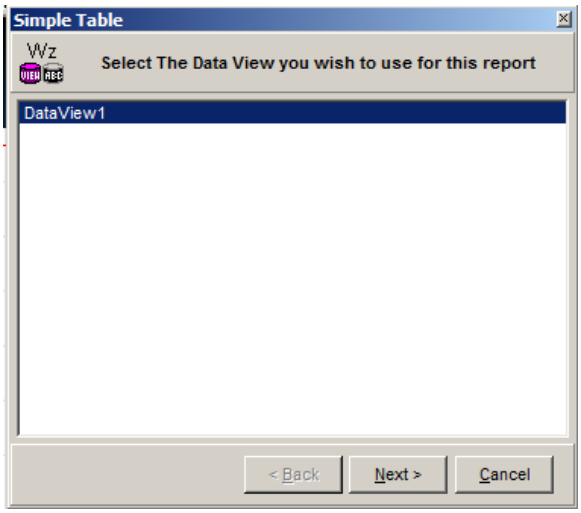
3. В окне дизайнера Rave Reports выполнить команду File – New Data Object. В появившемся окне Data Connections выбрать Direct Data View и нажать кнопку Next.



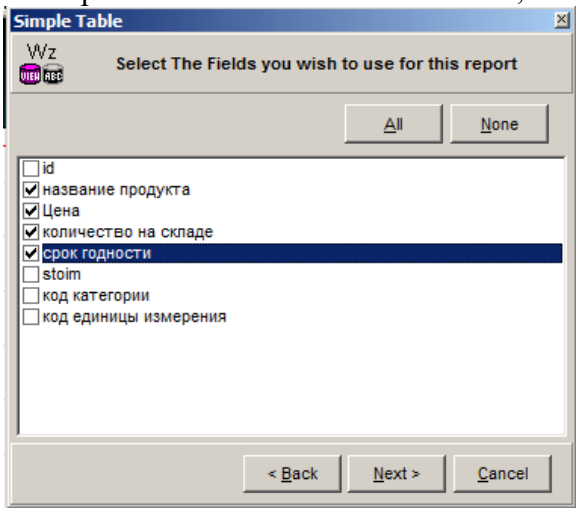
4. В окне Data Connection в списке Active Data Connections выделить имя RvDataSetConnection1 и кликните на кнопке Finish.



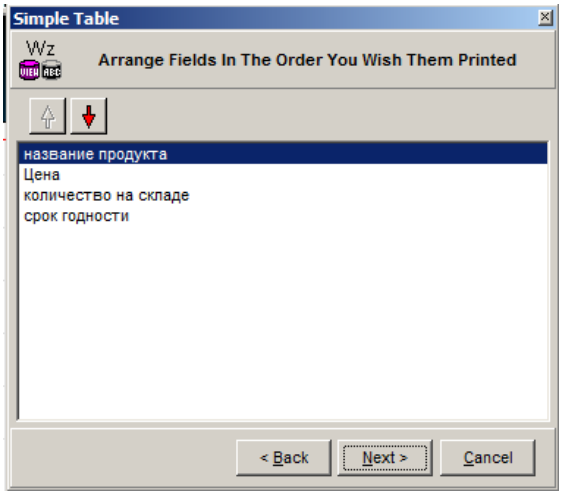
5. Выполнить команду Tools – Report Wizards – Simple Table для вызова мастера простой таблицы. Выделить имя DataView1 и нажать Next.



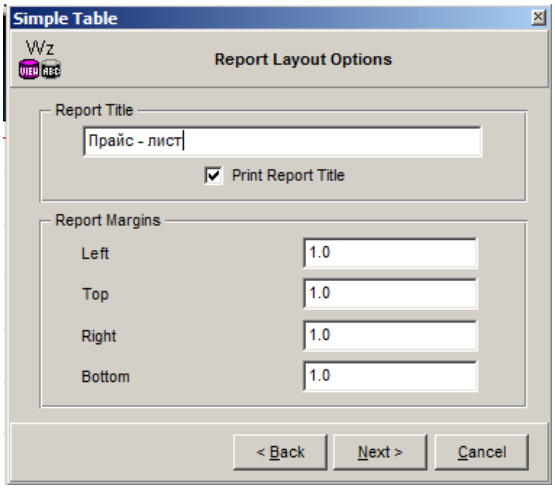
6. Выбрать нужные поля для отображения в отчете и нажать Next;



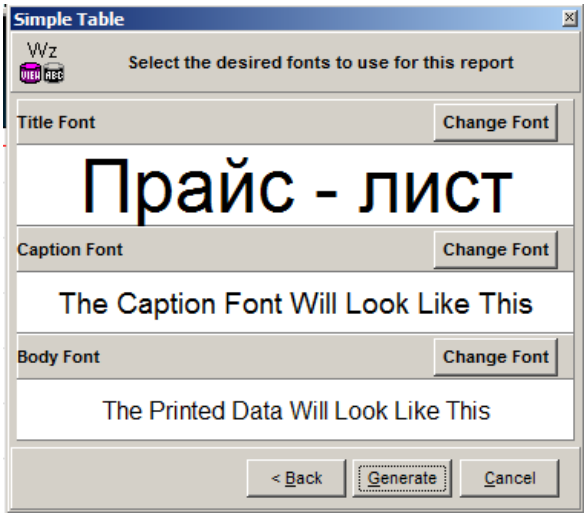
7.Если необходимо, изменить порядок полей и нажать Next;



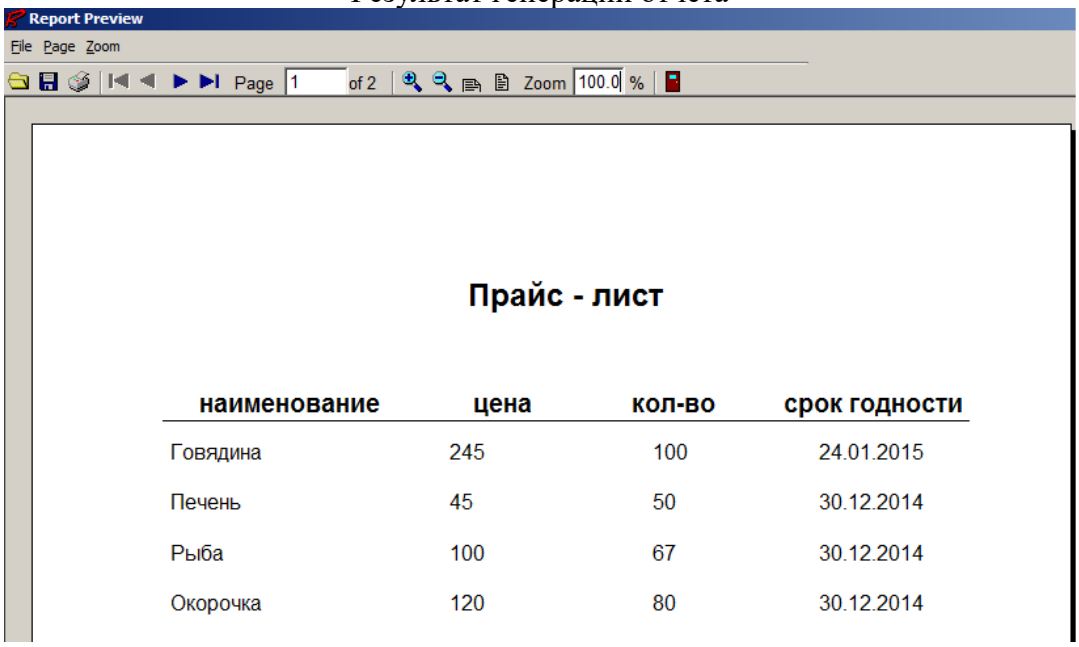
8. Задать имя отчета в поле Report Title.



9.Изменить размеры шрифта, если нужно, и нажать Generate для создания отчета.



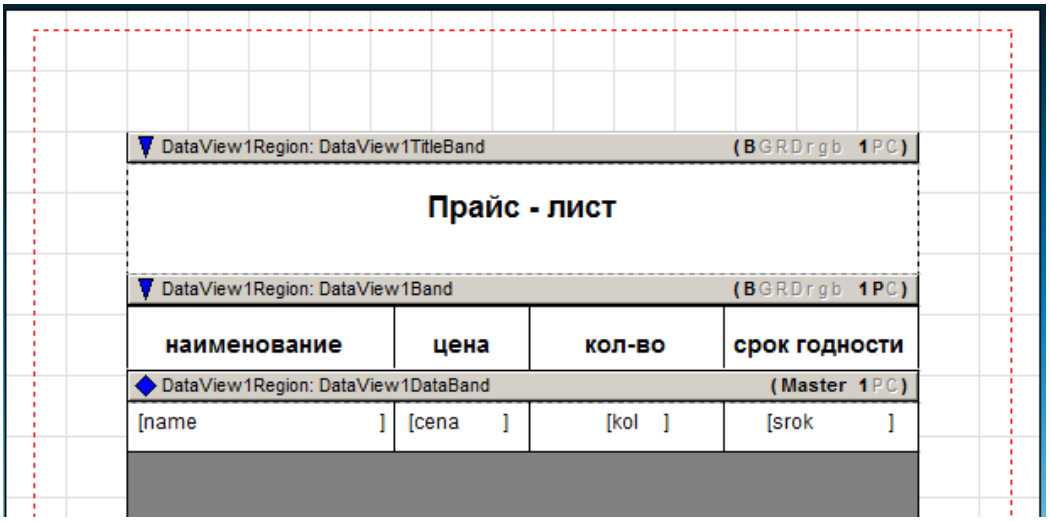
Результат генерации отчета



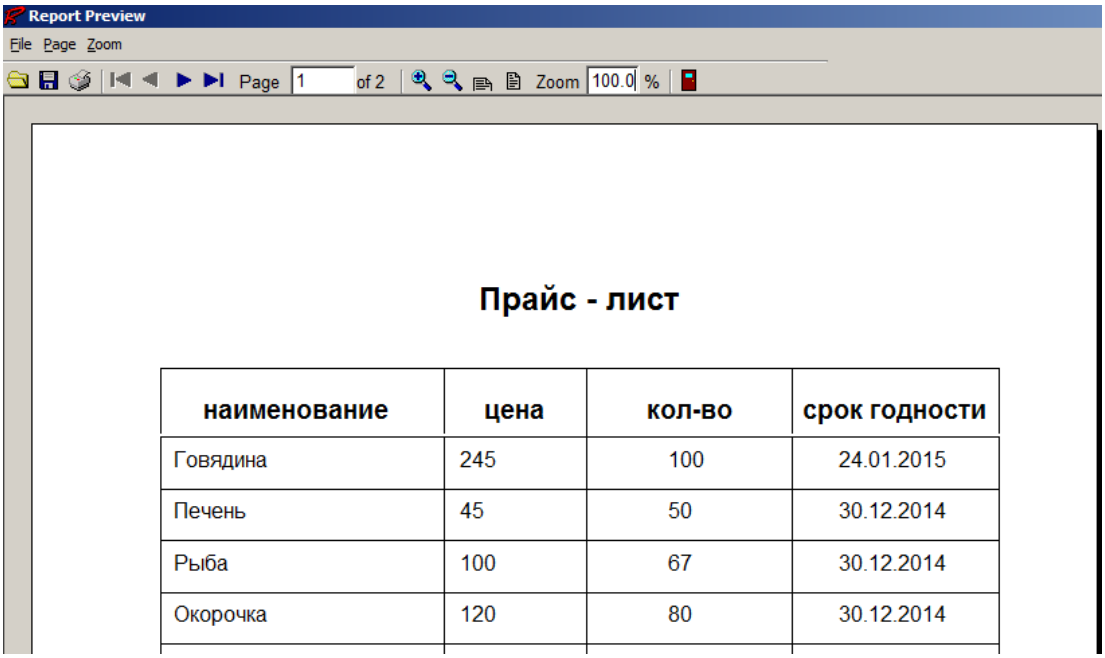
10. Компоненты с искаженной кодировкой удалите. Замените их компонентами Text из вкладки Standart

Используя вкладку Font, измените стиль и размер шрифта для текстовых компонент.

Используя компоненты вкладки Drawing, нарисуйте линии между полями выводимой таблицы



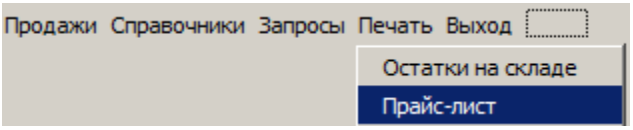
11.Для просмотра результатов выполнить команду File - Execute Report . В появившемся окне Output Options выбрать Preview и нажать Ok.



12.Сохраните созданный отчет под именем Prays.rav в директории проекта и закройте визуальную среду разработки отчетов RaveReport.

13.Откройте форму Datamodule2, выделите компонент RvProject1. В свойстве ProjectFile компонента через диалог открытия файла выберите файл Prays.rav.

14.Откройте главную форму. Дополните главное меню новым пунктом Прайс-лист.



15. В обработчик события OnClick данного пункта впишите код:

```
Datamodule2.RvProject1.Execute;
```

16.Сохраните проект и проверьте его работу.

Контрольные вопросы

1. Описание визуальной среды разработки отчетов Rave Reports.
2. Компоненты вкладки Standart среды разработки отчетов Rave Reports.
3. Компоненты вкладки Report среды разработки отчетов Rave Reports.
4. Компоненты Rave Reports в VCL в Delphi.