

## Лабораторная работа № 6

### Создание отчетов с использованием генератора отчетов Rave Reports

**Цель работы:** изучить и получить навыки использования компонентов для работы с генератором отчетов Rave Reports, входящим в систему Embarcadero RAD Studio 2010 и доступным в C++Builder 2010.

**Продолжительность работы** - 4 ч.

#### Теоретические сведения

В лабораторной работе № 4 было показано, как получить доступ к информации, хранящейся в базе данных, и создать приложения для отображения и редактирования этой информации, используя удобный и понятный пользовательский интерфейс. Наряду с этим на практике требуется обеспечить вывод выбранной из базы данных информации не только в экранные формы, но и на печать.

Информацию, выводимую на печать или в файл и представленную в удобном для восприятия виде, называют отчетом. Создание отчетов является одной из основных функций приложений, работающих с базами данных.

Для создания отчетов используются специальные компоненты, которые значительно облегчают эту задачу, выполняя все основные функции по форматированию, предварительному просмотру и выводу на печать информации из некоторого набора данных. Поэтому разработка отчетов обычно сводится к определению их структуры и внешнего вида.

Разработка отчета во многом схожа с разработкой формы для отображения данных, поскольку выполнение основных функций уже предусмотрено в соответствующих компонентах, которые требуется только определенным образом разместить и настроить. Многие компоненты, используемые при создании отчетов, подобны компонентам отображения данных, применяемым при разработке форм.

Поэтому, приступая к разработке отчета, требуется определить следующие его параметры:

- 1) информацию, которая должна содержаться в отчете;
- 2) таблицы с необходимыми данными;
- 3) внешний вид создаваемого отчета;
- 4) поля, по которым производится упорядочение и группировка данных в отчете;
- 5) содержание итоговой части отчета, если в ней есть необходимость;
- 6) дополнительную информацию, отображаемую в отчете: заголовки, поясняющие надписи, разделительные линии, рисунки, номера страниц, дату формирования отчета и т.д.

Перед разработкой отчета целесообразно нарисовать его эскиз на бумаге, чтобы определить перечень компонентов, требующихся для создания отчета.

Компоненты для создания отчетов позволяют формировать отчеты, которые условно можно разделить на две группы: табличные отчеты и отчеты в свободной форме.

В табличном отчете информация упорядочивается по строкам и столбцам. Такие отчеты фактически повторяют структуру таблиц базы данных или выборки таблиц.

Отчеты, выполненные в свободной форме, содержат информацию, располагающуюся произвольно. Примером отчетов такого типа могут служить этикетки с почтовыми адресами для конвертов и письма, где поля таблицы базы данных должны размещаться в специально отведенных местах. При реализации отчетов в свободной форме данные также представляются в определенном порядке, но результат упорядочения не является таблицей.

Помимо значений полей из таблиц базы данных или вычисляемых полей отчет может содержать и другие объекты, например графики, диаграммы, рисунки, поясняющие надписи.

В системе C++Builder для создания отчетов имеются компоненты, собранные в группе Rave палитры компонентов и обеспечивающие связь с генератором отчетов Rave Reports. Процесс разработки структуры отчета отделен от процесса формирования отчета за счет применения промежуточного документа, называемого проектом отчета. *Проект отчета* создается с помощью утилиты Rave Reports Designer, которая вызывается из системы C++Builder и взаимодействует с программой, создаваемой в системе C++Builder. В создаваемую

программу внедряется так называемая машина генератора отчета, которая по данным, полученным из файла проекта отчета (\*.rav), формирует отчет.

Из всех компонентов генератора отчетов Rave Reports, имеющих в группе Rave, для выполнения лабораторной работы используются следующие компоненты:

- RvProject для установки связи с файлом проекта отчета, который был создан с помощью утилиты Rave Reports Designer. У этого компонента имеются, в частности, свойства Name (имя компонента) и ProjectFile (спецификация местонахождения файла проекта отчета);
- Data Connection (RvTableConnection, RvQueryConnection, RvDataSetConnection и т.п.) для связи между компонентами доступа к данным, включенными в программу, созданную в системе C++Builder, и компонентами, использованными в проекте отчета. Для компонентов RvTableConnection, RvQueryConnection, RvDataSetConnection устанавливается соответственно свойство Table, Query, DataSet, задающее ссылку на имеющийся в программе компонент доступа к данным типа TTable, TQuery, TADOTable, TADOQuery и т.д.

### ***Визуальный дизайнер отчетов***

В состав утилиты Rave Reports Designer входит визуальный дизайнер отчетов, который может запускаться из системы C++Builder командой главного меню Tools | Rave Reports Designer или из главного меню ОС Windows. После запуска дизайнера отчетов активизируется окно, показанное на рис.1 и состоящее из следующих частей:

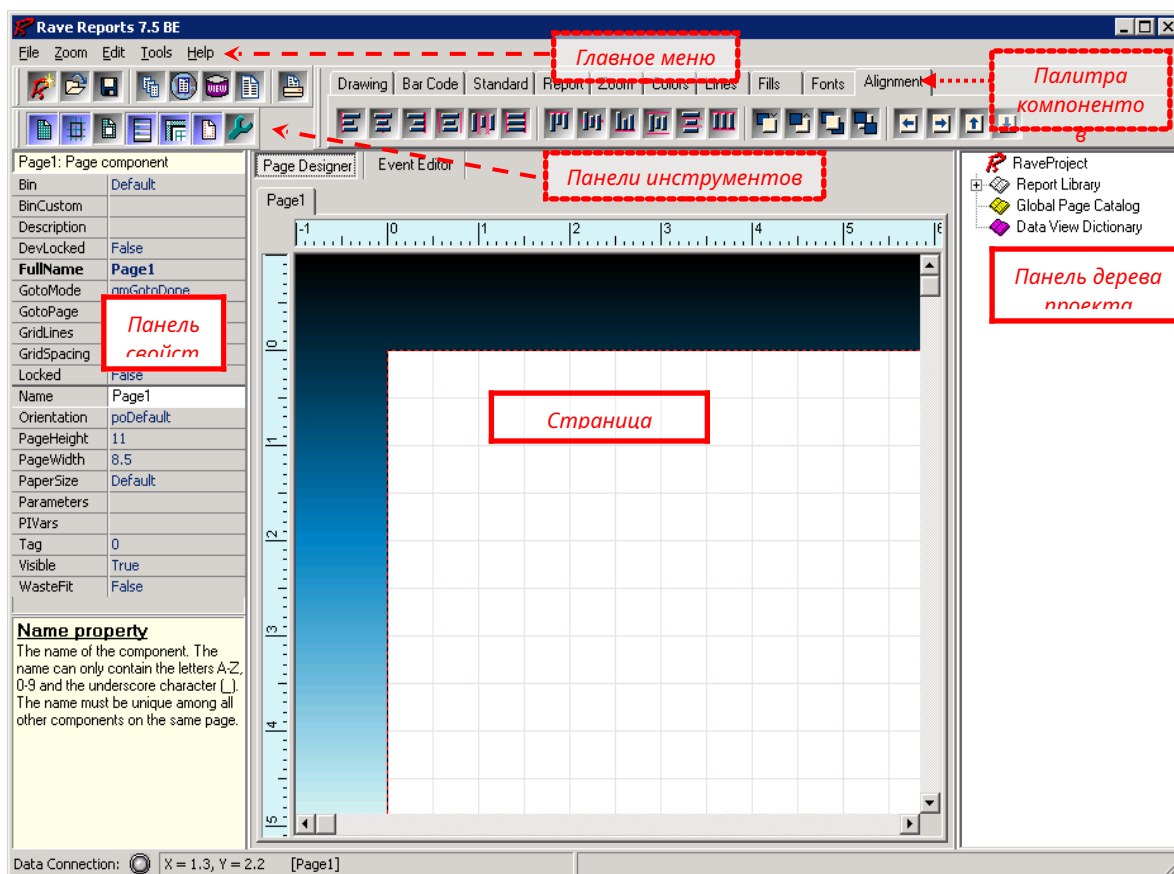


Рис.1. Окно дизайнера

- область навигации, содержащая главное меню, панели инструментов, дублирующие команды главного меню, и палитра компонентов с закладками;
- панель свойств, служащая для отображения и установки значений свойств выбранного компонента отчета;
- страница отчета, разбитая на ячейки, в виде сетки;

- панель дерева проекта, позволяющая быстро просматривать структуру проекта отчета и переходить к нужному объекту проекта.

Структура проекта отчета представлена в дереве проекта следующими узлами:

- RaveProject - контейнер всех прочих частей проекта;
- Report Library - библиотека отчетов;
- Global Page Catalog - каталог глобальных страниц
- Data View Dictionary - каталог объектов данных.

Каждый из узлов 2 - 4 может быть раскрыт для доступа к узлам нижнего уровня и выделения нужного объекта отчета.

### ***Технология создания отчетов***

**Создание простого отчета с помощью Мастера.** Рассмотрим пример построения отчета, содержащего информацию о служащих из таблицы Employee базы данных DBdemo, находящейся на сервере Nebula.

1. Создадим проект нового приложения командой File | New | VCL Forms Application и поместим в форму компонент TADOTable, свяжем его с таблицей Employee и откроем таблицу, задав для этого компонента, получившего имя ADOTable1, значение свойства Active равным True. Затем выберем в палитре компонентов группу Rave и поместим в форму компоненты TRvProject и TRvDataSetConnection, связав последний из компонентов с таблицей с помощью свойства DataSet. Далее разместим кнопку, выбрав компонент TButton в группе Standard и задав для ее свойства Caption значение ОТЧЕТ. Двойным щелчком по кнопке перейдем в окно текстового редактора и в строке отмеченной текстовым курсором наберем оператор, запускающий процесс формирования отчета в соответствии с имеющимся проектом отчета

```
RvProject1->Execute( );
```

2. Сохраним проект приложения в отдельной папке (например, Отчет1) командой File | Save Project As.

3. Запустим утилиту Rave Reports Designer, создадим новый проект отчета командой File | New и в новом проекте отчета командой File | New Data Object создадим объект данных Direct Data View, связав его с компонентом RvDataSetConnection1, ранее размещенным в форме сохраненного приложения.

4. Работая в дизайнере отчетов, вызовем Мастер создания отчета (командой Tools | Report Wizards | Simple Table) и с его помощью сгенерируем проект отчета, содержащего все столбцы таблицы Employee (рис.2).

5. Сохраним в папке, содержащей сохраненный проект приложения, файл проекта отчета командой File | Save, оставив предлагаемое по умолчанию имя Project1.rav

6. Осуществим предварительный просмотр сформированного отчета, нажав клавишу F9 или выбрав команду File | Execute Report (рис.3).

7. Вернувшись в систему C++Builder, установим в свойстве ProjectFile компонента RvProject1 ссылку на файл проекта отчета.

8. Запустим приложение на выполнение и сформируем отчет, воспользовавшись кнопкой ОТЧЕТ, имеющейся в форме.

Данные, выводимые в отчете, всегда сортируются в соответствии с текущим индексом, который задается с помощью свойства IndexName компонентов доступа к данным (в рассматриваемом примере это компонент ADOTable1). По умолчанию сортировка производится в соответствии с первичным ключом. Можно отсортировать строки отчета по фамилиям, если указать для свойства IndexName значение индекса Emp\_LastName, если предусмотреть его для таблицы Employee.

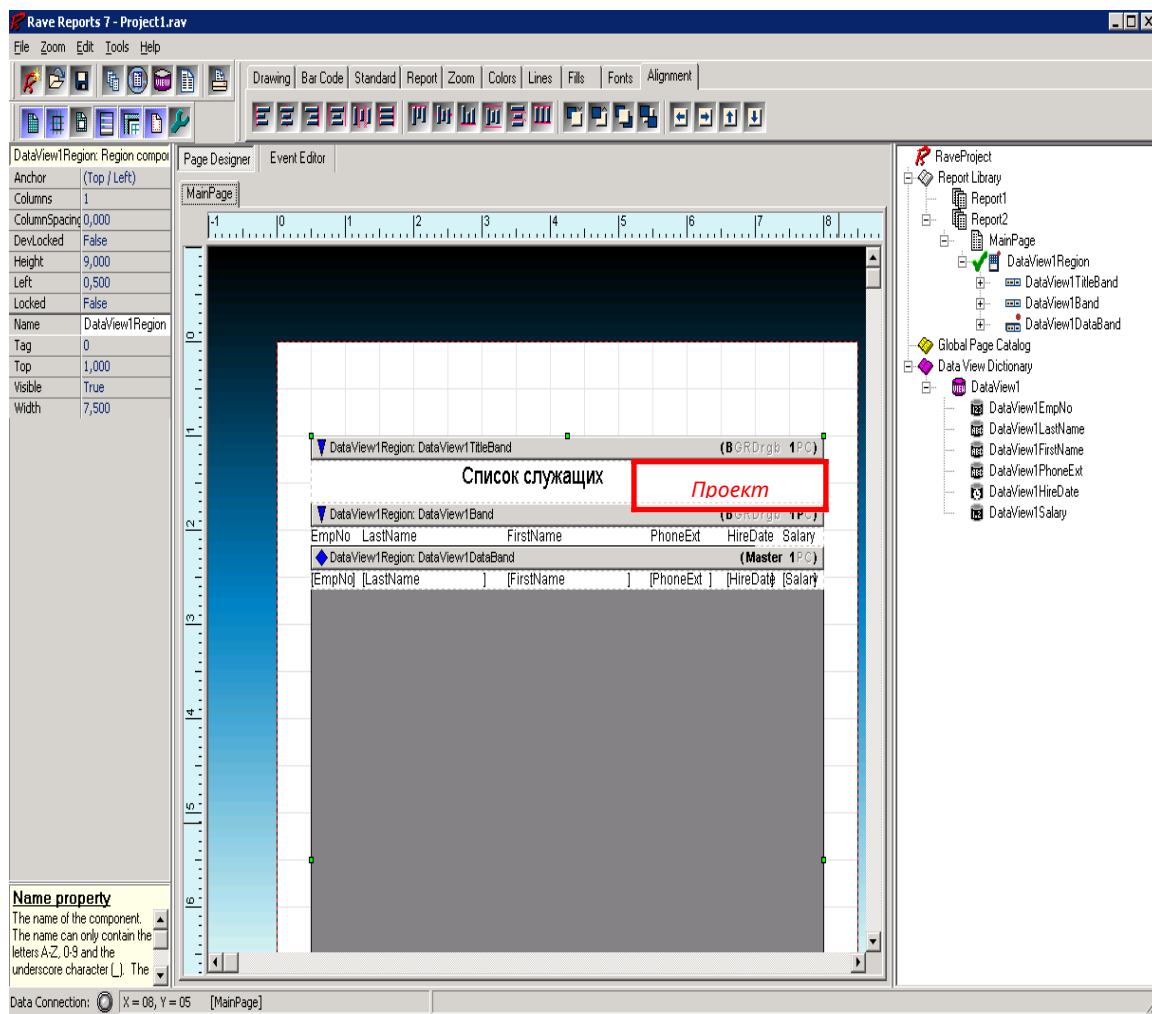
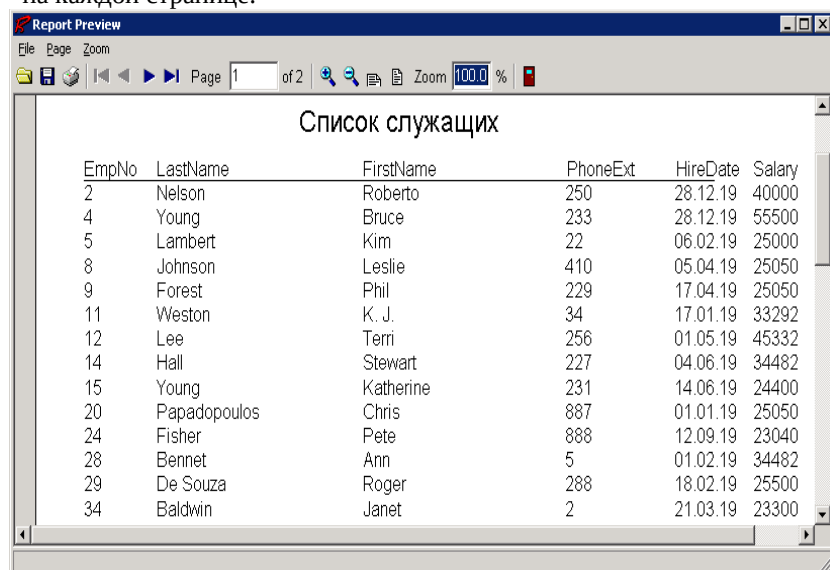


Рис.2. Окно дизайнера отчетов с проектом отчета

В результате работы Мастера создается проект отчета (рис.4), который содержит единственную страницу (компонент MainPage) и

область просмотра данных DataView1Region с тремя полосами DataView1TitleBand, DataView1Band и DataView1DataBand. Общая структура проекта отображается на панели дерева проекта (рис.5).

Две первые полосы (DataView1TitleBand и DataView1Band) являются полосами заголовка отчета и заголовков полей. Они отличаются тем, что первая печатается только в начале отчета, а вторая - на каждой странице.



Report Preview

File Page Zoom

Page 1 of 2 Zoom 100.0 %

Список служащих

EmpNo	LastName	FirstName	PhoneExt	HireDate	Salary
2	Nelson	Roberto	250	28.12.19	40000
4	Young	Bruce	233	28.12.19	55500
5	Lambert	Kim	22	06.02.19	25000
8	Johnson	Leslie	410	05.04.19	25050
9	Forest	Phil	229	17.04.19	25050
11	Weston	K. J.	34	17.01.19	33292
12	Lee	Terri	256	01.05.19	45332
14	Hall	Stewart	227	04.06.19	34482
15	Young	Katherine	231	14.06.19	24400
20	Papadopoulos	Chris	887	01.01.19	25050
24	Fisher	Pete	888	12.09.19	23040
28	Bennet	Ann	5	01.02.19	34482
29	De Souza	Roger	288	18.02.19	25500
34	Baldwin	Janet	2	21.03.19	23300

Рис.3. Окно предварительного просмотра сформированного отчета



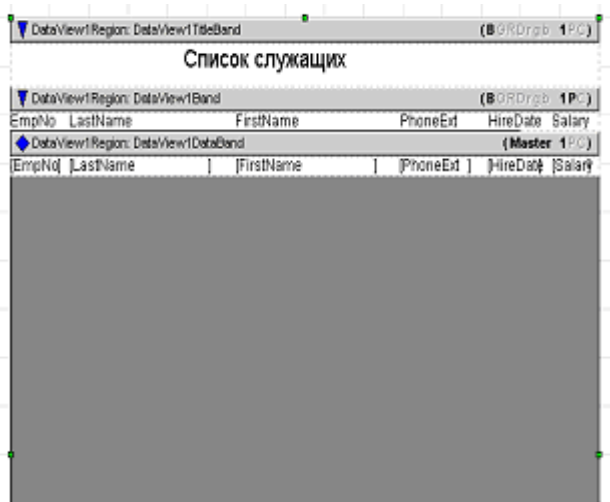


Рис.4. Проект отчета

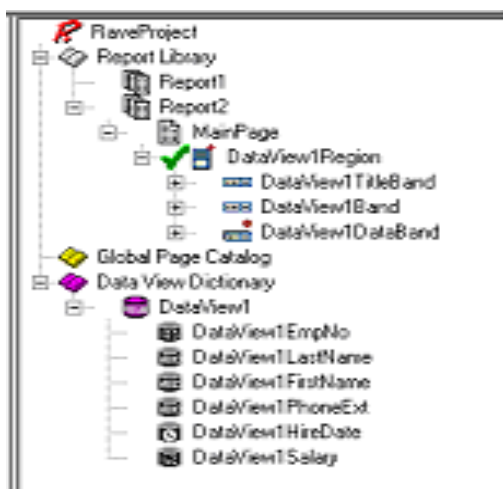



Рис.5. Дерево проекта отчета

Если поочередно выделить каждую из этих полос и на панели свойств выбрать их свойство `BandStyle` и нажать кнопку  , то в появившемся редакторе (рис.6) этого свойства, определяющего положение полосы и ее наличие на странице при печати (генерации) отчета, можно обнаружить для первой установленные флажки `Body Header` и `First`, а для второй еще и флажок `New Page`. Если этот флажок снять, полоса напечатается только на первой странице. На любом из этих полей можно размещать компоненты закладки `Standard` палитры компонентов `Rave Reports Designer`.

Полоса данных `DataView1DataBand` повторяется в формируемом отчете столько раз, сколько записей (строк) содержит соответствующий набор данных. Если после полосы данных расположить еще одну текстовую полосу (например, для нумерации страниц), она будет напечатана только в конце отчета. Чтобы вставить внизу или вверху каждой печатаемой страницы элементы оформления, нужно располагать эти элементы вне области просмотра данных `DataView1Region`. При этом, возможно, понадобится уменьшить вертикальный размер этой области.

Каждая из трех рассмотренных полос содержит компоненты `Text` (закладка `Standard`) и `DataText` (закладка `Report`) для отображения текста заголовков отчета и столбцов и данных из таблицы `Employee`. Эти компоненты можно увидеть в дереве проекта отчета (см. рис.5), если раскрыть узлы, соответствующие полосам `DataView1TitleBand`, `DataView1Band` и `DataView1DataBand`.

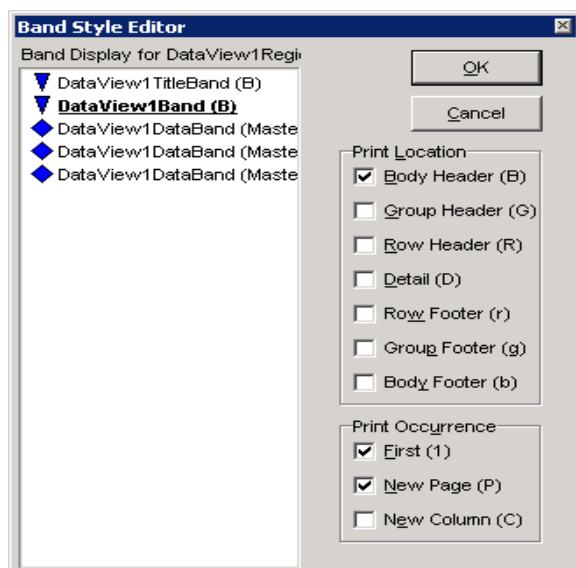


Рис.6. Окно редактора свойства BandStyle

**Изменение созданного отчета.** Созданный Мастером отчет может быть изменен с целью улучшения его оформления, чтобы устранить наложение заголовков, увеличить или уменьшить ширину поля для выводимого значения и т.д.. Например, в отчете, показанном на рис.3, целесообразно предусмотреть расширение поля для даты приема служащего на работу, задать русскоязычные заголовки столбцов, отображение даты генерации, ввести нумерацию страниц, выделить столбцы и строки вертикальными и горизонтальными линиями.

Изменения вносятся в проект отчета, открытый в дизайнера отчетов, и затем сохраняются командой File | Save. Наложение заголовков устраняется выделением одного из них и его перемещением с помощью мыши или одновременным нажатием CTRL и клавиши со стрелкой нужного направления. Изменение размера выделенного заголовка, поля или другого компонента производится перемещением зеленых маркеров, обозначающих границы компонента, или нажатием SHIFT и клавиш со стрелками влево-вправо. Выравнивание взаимного

положения нескольких компонентов осуществляется командами контекстного меню после выделения компонентов при нажатой клавише SHIFT. Размер и другие параметры шрифта задаются свойством Font.

Изменение заголовка столбца или другой надписи осуществляется после их выделения в проекте отчета корректировкой значения свойства Text на панели свойств.

Многие системные параметры отчета, в том числе текущая дата и номер текущей страницы, задаются в компоненте DataText, расположенном в палитре компонентов на закладке Report. Его свойство DataField позволяет не только ссылаться на нужное поле данных, но и использовать системные переменные в выражениях.

Для формирования даты генерации отчета со списком служащих нужно разместить сразу под заголовком отчета компонент DataText (закладка Report) и установить для него следующие свойства: Left=0, Width=7,9, Top=0,6, Font Justify=pjCenter, Font=MS Sans Serif, 12, Bold. В свойство DataField поместить следующий текст: "от "&Report.DateLong, т.е. вместо имени поля в свойство можно помещать произвольное выражение с участием системных переменных (в нашем случае - с переменной Report.DateLong, содержащей текущую дату в полном формате ДД Месяц ГГГГ). Значения этого свойства можно сформировать также с помощью редактора, нажав кнопку [...] в правом углу значения свойства: в поле Data Text окна Data Text Editor введем 'от ' +, раскроем список Report Variables и выберем в нем пункт DateLong, после чего щелкнем на кнопке Insert Report Var.

Чтобы вставить в отчет нумерацию страниц, нужно предусмотреть в проекте отчета пространство для размещения компонента DataText. Мастер создания отчета отводит для заголовков и данных всю страницу. Свободное место появится, если выделить компонент DataView1Region, щелкнув по нему в дереве объектов (предварительно раскрыв узлы ReportLibrary, Report1, MainPage), и установить его свойство Height, равным 7,9. С помощью вертикальной полосы прокрутки следует сместить изображение страницы в окне Page Designer к самому ее концу и вставить под серым пространством компонент DataText и в его свойство DataField поместить такую строку:

```
'Стр. ' + Report.RelativePage+ ' из '+Report.TotalPages
```

Для этого компонента надо установить шрифт MS Sans Serif, 14, Bold.

Чтобы расчертить отчет линиями, необходимо на среднюю полосу (объект `DataViewBand` в дереве проекта) поместить компонент `HLine` (закладка `Drawing`), а также 7 компонентов `VLine` с подходящими размерами и положением. На нижнюю полосу с данными также следует поместить компонент `HLine` и компоненты `VLine`. Затем останется сохранить проект и предварительно просмотреть отчет, чтобы убедиться в правильности внесенных изменений.

#### **Создание простого отчета без использования Мастера.**

Осуществляется в приводимой ниже последовательности.

1. Создать в отчете связанный с данными объект отображения `Direct Data View`, если отчет связан с набором данных (например, таблицей) из приложения `C++Builder`.

2. На пустую страницу поместите надписи, графические изображения и другие элементы оформления (с закладок `Drawing` и `Standard` среды `Rave Reports Designer`), которые должны печататься на каждой странице.

3. На страницу поместить компонент `Region` (закладка `Report`), соответствующий области просмотра данных. Его размеры и положение на странице должны быть такими, чтобы не закрывать оформительские элементы сверху, снизу или по бокам страницы.

4. Поместить в область просмотра данных полосы `Band` (закладка `Report`) для заголовка отчета и названий полей. Каждая помещаемая в область полоса занимает весь ее горизонтальный размер. Для полосы заголовка отчета в редакторе свойства `BandStyle` должны быть установлены флажки `Body Header` и `First`, для полосы названий полей - флажки `Body Header`, `First` и `New Page`.

5. За полосами заголовка поместить полосу `DataBand` (закладка `Report`). Своим свойством `DataView` она связывается с объектом отображения данных.

6. На полосе `DataBand` расположить компоненты для отображения данных из отдельных полей набора данных. Каждый компонент через свойство `DataView` связать с объектом отображения данных, а в свойстве `DataField` указать отображаемое компонентом поле данных или выражение с участием полей и системных переменных.

**Создание отчета с группировкой данных.** Отчет, в котором вся информация разделяется на группы данных, объединенных каким-то общим признаком, называется отчетом с группировкой данных. Например, таким отчетом будет список компаний-покупателей, упорядоченный по штатам, с данными из таблицы `Customer`.

Спецификой отчета с группировкой данных является то, что полосы с данными, относящимися к какой-то группе, должны указывать на объект отображения данных в своем свойстве GroupView и на поле группировки в свойстве GroupKey. Поле группировки - это поле, изменение значения которого означает конец предыдущей группы и начало следующей. Обычно такое поле указывается первым в предложении ORDER BY или GROUP BY оператора на языке SQL.

Рассмотрим пример создания проекта отчета со списком компаний-покупателей с группировкой по штатам (рис.7). Предусмотрим в приложении, создаваемом в системе C++Builder, компонент TADOQuery со следующим текстом запроса в его свойстве SQL:

```
SELECT Company, City, State, FAX, TaxRate, Contact FROM  
Customer ORDER BY State
```

В формируемом отчете полем группировки будет State: изменение значения в нем воспринимается как признак окончания очередной группы данных и начала следующей.

Для связи с компонентом типа TADOQuery следует использовать компонент TRvDataSetConnection.

**Примечание.** В примере для формирования отчета с группировкой данных используется SQL-запрос, обеспечивающий *сортировку* строк таблицы так, чтобы строки с одинаковым обозначением штата оказались в одной группе и в таком порядке отображались в отчете. Аналогичный порядок расположения строк можно получить с помощью SQL-запроса с группировкой SELECT Company, City, State, FAX, TaxRate, Contact FROM Customer GROUP BY State, Company, City, FAX, TaxRate, Contact, но такое решение является нерациональным.

В проекте отчета создается объект отображения данных типа Direct Data View, ссылающийся на связной компонент TRvDataSetConnection в приложении C++Builder. Дальнейший порядок действий описывается ниже.

1. На пустую страницу отчета поместить область просмотра данных Region, установить нужные размеры этой области и разместить над ней компонент DataText (закладка Report) со значением свойства DataField, равным 'Стр. ' + Report.RelativePage + ' из ' + Report.TotalPages, для нумерации страниц отчета.

Report Preview

File Page Zoom

Page 2 of 3 Zoom 100.0 %

Стр. 2 из 3

Штат	FL			
Компания	Город	Факс	Ставка налога	Контактное лицо
Blue Sports Club	Largo	612-897-0348	0	Harry Bathbone
Aquatic Drama	Tampa	613-442-7678	0	Gillian Owen
Divers of Venice	Venice	813-443-9842	0	Simone Green
The Depth Charge	Marathon	800-555-0353	0	Sam Witherspoon
Action Club	Sarasota	813-870-0282	0	Michael Spurling
5 компаний в штате.			Средняя ставка	0

Штат	GA			
Компания	Город	Факс	Ставка налога	Контактное лицо
Island Finders	St Simons	713-423-5676	0	Desmond Ortega
1 компаний в штате.			Средняя ставка	0

Штат	HI			
Компания	Город	Факс	Ставка налога	Контактное лицо
Kauai Dive Shoppe	Kapaa Ka	808-555-0278	8,5	Erica Norman
Ocean Adventures	Maui	776-868-9553	0	Paul Still
Blue Jack Aqua Cente	Waipahu	401-609-9403	0	Ernest Barratt
Ocean Paradise	Kailua-Ko	808-555-8450	0	Paul Gardner
Vashon Ventures	Honolulu	532-099-6654	0	Susan Smith
Makai SCUBA Club	Kailua-Ko	317-649-6787	0	Donna Siaus
6 компаний в штате.			Средняя ставка	1,41666666666667

Рис.7. Пример отчета с группировкой данных

2. В область просмотра данных поместить полосу заголовка отчета (компонент Band).

3. Ниже расположить три полосы для отображения заголовка данных (компонент Band), собственно сгруппированных данных (компонент DataBand) и итога отображения (компонент Band). Каждая из них должна иметь ссылку на объект отображения данных в свойстве GroupDataView и группирующее поле (State) в свойстве GroupKey. В редакторе свойства BandStyle первой полосы установить флажок Group Header, последней полосы - флажок Group Footer. В редакторе свойства BandStyle средней полосы флажки в группе Print Location можно не устанавливать. Заголовочная полоса должна содержать набор компонентов Text (закладка Standard) с

установленными значениями свойства Text, задающими названия группы строк (Штат) и столбцов (Компания, Город, Факс, Ставка налога, Контактное лицо), а также компонент DataText (закладка Report) для отображения значений поля State, установив для него свойства DataView и DataField из ниспадающего списка. Заголовочная и итоговая полосы должны ссылаться на полосу с группой данных в своем свойстве ControllerBand. Чтобы это стало возможным, средняя полоса с данными должна быть компонентом DataBand с установленным свойством DataView. В эту полосу необходимо поместить пять компонентов DataText (закладка Report) для отображения значений полей Company, City, FAX, TaxRate, Contact из таблицы Customer и для каждого компонента выбрать из ниспадающего списка значения свойств DataView и DataField.

Для каждой группы компаний-покупателей целесообразно предусмотреть итоговую строку с количеством компаний в штате и средней ставкой налога. Такого рода сведения получаются как результат обработки группы строк и формируются с помощью так называемых агрегатных функций. В Rave Reports предусмотрены стандартные для языка SQL-92 агрегатные функции, перечисленные в табл.1.

Для получения в отчете результатов работы агрегатной функции предусмотрены два вычислительных компонента закладки Report - CalcText и CalcTotal. Первый вычисляет и показывает значение агрегатной функции, второй лишь вычисляет значение и передает его другому вычислительному компоненту для его дальнейшего преобразования.

Оба компонента имеют свойства DataView, DataField и CalcType. Два первых определяют набор данных и поле, значения которого будут использоваться при вычислении, третье определяет агрегатную функцию. Функция COUNT может быть применена к полю любого типа, остальные функции определены только для числовых полей.

Чтобы определить момент срабатывания вычислительного компонента, нужно ввести в полосу отчета, в которой отображаются значения группы, компонент CalcController (контроллер), отображаемый только в дереве отчета, и ссылку на него поместить в свойство Controller вычислительного компонента. При печати полосы с контроллером он передаст ссылающимся на него вычислительным компонентам сообщение об окончании текущей группы строк, которое приведет к срабатыванию компонента и переходу к очередному циклу вычислений. В результате после завершения печати групповых данных



вычислительный компонент накопит и поместит в отчет значение нужной агрегатной функции.

Для оформления итоговой полосы отчета, показанного на рис.7, на нее помещаются два вычислительных компонента CalcText. В свойствах DataView обоих компонентов указана ссылка на нужный объект отображения данных (DataView1). Первый компонент подсчитывает общее количество значений поля Company, поэтому его свойство CalcType должно иметь значение ctCount. Второй компонент вычисляет среднее значение поля TaxRate, поэтому для него свойство CalcType задается равным значению ctAverage, а свойства DataView и DataField ссылаются на DataView1 и TaxRate. На полосу DataBand1, в которой отображаются данные группы записей, предварительно помещается невидимый компонент CalcController1, и ссылка на него должна быть указана в свойствах Controller вычислительных компонентов. На рис.8 показаны полосы проекта отчета.

**Создание отчета с информацией из главной и подчиненной таблиц с помощью Мастера.** Сформируем отчет, выводящий сведения о поставщиках партий (из таблицы Vendors) и поставленных партиях товаров (из таблицы Parts, которая подчинена главной таблице Vendors). В этом отчете содержатся данные из одной записи главной таблицы и все записи связанной с ней подчиненной таблицы. Для подготовки таких отчетов в дизайнера отчетов предусмотрен мастер, который вызывается командой Tools | Report Wizards | Master/Detail Report.

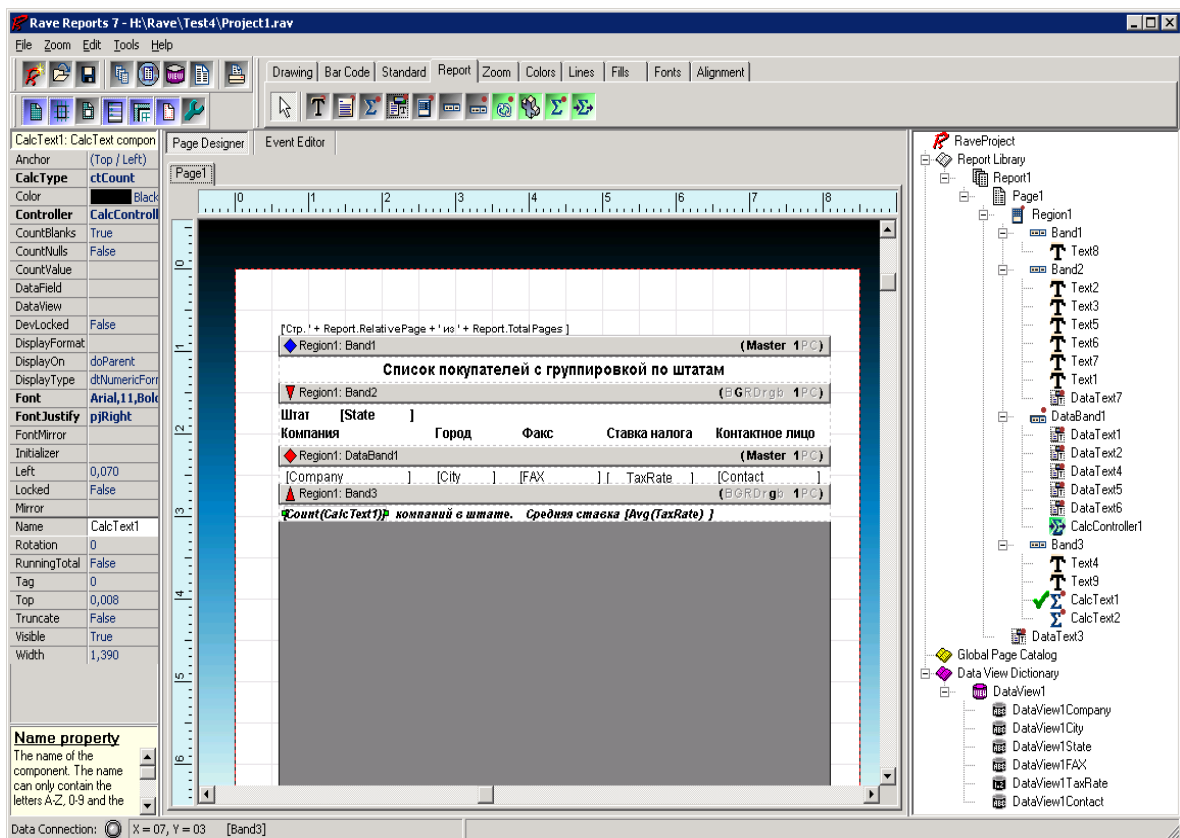


Рис.8. Проект отчета с группировкой данных

Для создания требуемого отчета надо выполнить следующие действия.

1. В форму приложения, создаваемого в системе, поместим два компонента TADOTable и кнопку, настроить их на связь с главной и подчиненной таблицами в базе данных и генерацию отчета. Связывать компоненты между собой не обязательно, так как эта связь будет создана в проекте отчета.

2. Поместить в форму компонент TRvProject и два связанных компонента TRvDataSetConnection и связать их по отдельности с разными компонентами TADOTable.

3. Сохранить проект приложения в отдельной папке (например, Отчет5) командой File | Save Project As.

4. Запустить утилиту Rave Reports Designer, создать новый проект отчета командой File | New и в новом проекте отчета командой File | New Data Object создать два объекта данных Direct Data View, связав их с компонентами RvDataSetConnection1 и RvDataSetConnection2, ранее размещенными в форме сохраненного приложения.

5. Работая в дизайнере отчетов, вызвать Мастер создания отчета (командой Tools | Report Wizards | Master/Detail Table) и с его помощью сгенерировать проект отчета, содержащего все столбцы таблицы Parts и столбцы VendorName, City, State, Country таблицы Vendors (рис.9).

6. Сохранить в папке, содержащей сохраненный проект приложения, файл проекта отчета командой File | Save, оставив предлагаемое по умолчанию имя Project1.rav

7. Осуществить предварительный просмотр сформированного отчета, нажав клавишу F9 или выбрав команду File | Execute Report (рис.10).

8. Вернувшись в систему C++Builder, установить в свойстве ProjectFile компонента RvProject1 ссылку на файл проекта отчета.

9. Запустить приложение на выполнение и сформировать отчет, воспользовавшись кнопкой ОТЧЕТ, имеющейся в форме.

**Создание отчета в свободной форме.** Отчет в свободной форме строится таким же способом, что и простой отчет без использования Мастера, но данные, выводимые в полосу детальной информации, не упорядочиваются по столбцам, а располагаются произвольно.

Отчеты в свободной форме обычно применяются для вывода каких-либо бланков, в которых информация не представлена в виде таблиц. Типичными примерами отчетов в свободной форме являются письма со стандартным текстом и этикетки с почтовыми адресами для конвертов. Такие отчеты содержат только полосу детальной информации, задаваемую компонентом DataBand (закладка Report), который своим свойством DataView связывается с областью просмотра данных, представляемую компонентом Region. Количество выводимых этикеток на странице отчета зависит от размера области просмотра данных.

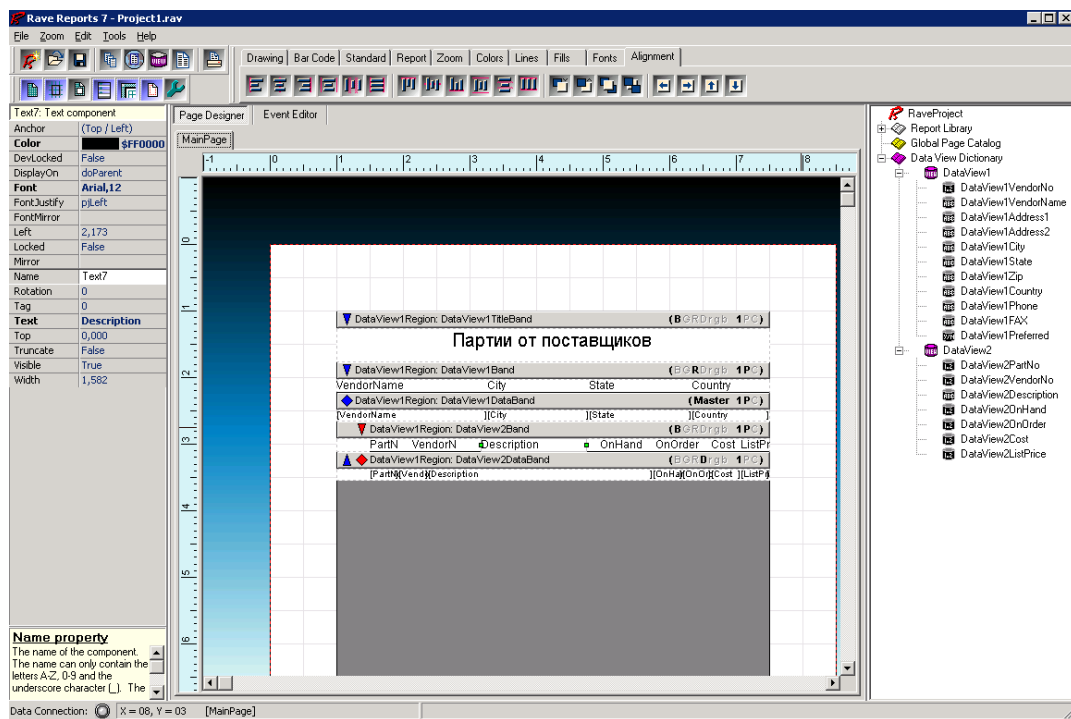


Рис.9. Проект отчета с главной и подчиненной таблицами

**Партии от поставщиков**

VendorName	City	State	Country
Cacor Corporation	Southfield	OH	U.S.A.

PartNo	VendorNo	Description	OnHand	OnOrder	Cost	ListPrice
2612	2014	Direct Sighting Compass	15	12	12,582	34,95
2613	2014	Dive Computer	5	2	76,97	179
2619	2014	Navigation Compass	8	20	9,177	19,95
2630	2014	Wrist Band Thermometer (F)	6	3	7,92	18
2632	2014	Depth/Pressure Gauge (Digital)	12	10	53,64	149
2648	2014	Depth/Pressure Gauge (Analog)	16	15	39,27	119
2657	2014	Wrist Band Thermometer (C)	12	10	6,48	18

Рис.10. Сформированный отчет для главной и подчиненной таблиц

### Лабораторное задание

1. Разработать приложения и проекты отчетов для формирования различных видов отчетов, отображающих информацию из таблиц учебной базы данных DBdemo.
2. Разработать приложения и проекты отчетов для формирования различных видов отчетов, отображающих информацию из таблиц базы данных, созданной для своего варианта в лабораторной работе № 2.

### Порядок выполнения работы

1. При домашней подготовке ознакомиться с назначением компонентов генератора отчетов Rave Reports и технологией создания отчетов. В конспект поместить сведения о компонентах, используемых для формирования отчетов, и настройке этих компонентов, а также эскизы отчетов для выбранных таблиц базы данных, созданной для своего варианта в лабораторной работе № 2.
2. На Терминале 4100 запустить систему C++Builder командой меню Пуск | Embarcadero RAD Studio 2010 | C++Builder 2010. Если в палитре компонентов системы C++Builder 2010 отсутствует группа Rave с компонентами для создания отчетов, то следует добавить эту группу

командой главного меню Component | Install Packages. В появившемся окне (рис.11) следует нажать кнопку Add, а затем указать пакет dclRave.bpl в папке C:\Program Files(x86)\Embarcadero\RAD Studio\7.0 \Rave Reports\Lib\

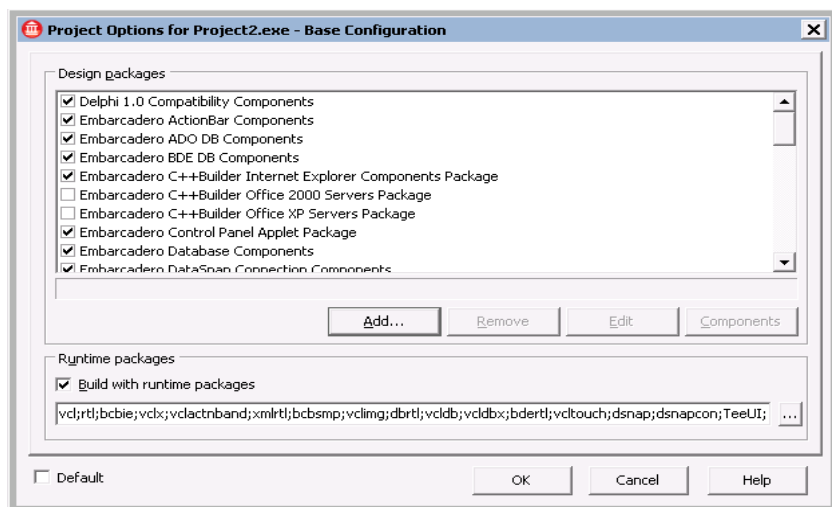


Рис.11. Окно с кнопкой Add для добавления пакета dclRave.bpl

3. Создать с помощью Мастера простой отчет со списком служащих (из таблицы Employee учебной базы данных DBdemo), сохранив приложение и проект отчета в отдельной папке на устройстве С терминального компьютера.

4. Изменить созданный отчет, задав русскоязычные заголовки столбцов, отображение даты генерации, нумерацию страниц и выделив столбцы и строки вертикальными и горизонтальными линиями.

5. Создать без использования Мастера простой отчет со списком компаний-покупателей (из таблицы Customer), сохранив приложение и проект отчета в отдельной папке на устройстве С терминального компьютера.

6. Создать отчет с группировкой и подведением итогов по группам для таблицы Customer, сохранив приложение и проект отчета в отдельной папке на устройстве С терминального компьютера.

7. Создать с помощью Мастера отчет с информацией из таблиц Vendors и Parts, сохранив приложение и проект отчета в отдельной папке на устройстве С терминального компьютера.

8. Создать отчет в свободной форме для печати визитных карточек служащих, сохранив приложение и проект отчета в отдельной папке на устройстве С терминального компьютера.

9. Для одной или двух таблиц базы данных, созданной для своего варианта в лабораторной работе № 2, разработать приложения и проекты отчетов следующих видов:

- а) простой отчет;
- б) отчет с группировкой данных;
- в) отчет с информацией из главной и подчиненной таблиц;
- г) отчет в свободной форме.

Сохранить разработанные приложения и проекты отчетов в отдельных папках на устройстве С терминального компьютера.

10. Оформить и показать преподавателю результаты лабораторной работы и защитить ее.

#### **Требования к отчету**

Отчет должен содержать:

- 1) название и цель работы;
- 2) сведения о компонентах, используемых для формирования отчетов;
- 3) сведения о настройке компонентов, используемых в разработанных приложениях;
- 4) эскизы отчетов для выбранных таблиц базы данных, созданной для своего варианта в лабораторной работе № 2.

#### **Контрольные вопросы**

1. Перечислить параметры отчета, содержащего сведения из БД.
2. На какие группы можно условно разделить отчеты?
3. Что представляет собой проект отчета?
4. Какие компоненты, необходимые для формирования отчета, используются в приложении?
5. Каким оператором приложения запускается процесс формирования отчета?
6. Какие компоненты предусмотрены в визуальном дизайнера для создания проекта отчета?

7. Какое свойство полосы отчета задает ее положение и наличие на странице генерируемого отчета?
8. Почему для отчета с группировкой данных, приведенного в лабораторной работе, следует использовать SQL-запрос с сортировкой вместо SQL-запроса с группировкой?
9. Чем определяется количество строк, выводимых на странице отчета?
10. Для чего предназначены компоненты CalcText и CalcController и какие свойства они имеют?