

Тема: Создание отчетов из Delphi средствами MSWord.

Цель: Приобретение навыков создания отчетов инструментами, предоставляемых средой Delphi. В случае, если для отчета необходимо использовать данные из двух или более таблиц, необходимо создать соответствующий запрос на выборку данных.

Контрольный пример:

Для связи проектов Delphi с приложением Microsoft Word существует множество подходов. Один из них – технология DDE (Dinamic Data Exchange Динамический обмен данными). Эта технология построена по принципу клиент-сервер. В нашем случае клиентом будет приложение Delphi, а сервером – MSWord.

Основной компонент для создания клиента – это TDDEClientConv, расположенный на вкладке System. Этот компонент позволяет отправить набор команд приложению MSWord в виде текстовой строки. Отправленные команды автоматически транслируются в вызовы методов объекта WordBasic. Синтаксис методов WordBasic проще, чем язык Visula Basic. Вцелом, объект WordBasic содержит около 900 кооманд, что вполне достаточно для управления приложением Word.

Создадим два отчета – «Прайс-лист» и «Счет-фактуру». Для этого откорректируем меню, добавив в пункт «Отчеты» два подпункта: «Прайс-лист» и «Счет-фактура».

Создание отчета «Прайс-лист»

1. Создадим документ Word, который будет шаблоном для отчета «Прайс-лист» как показано на рисунке 1.

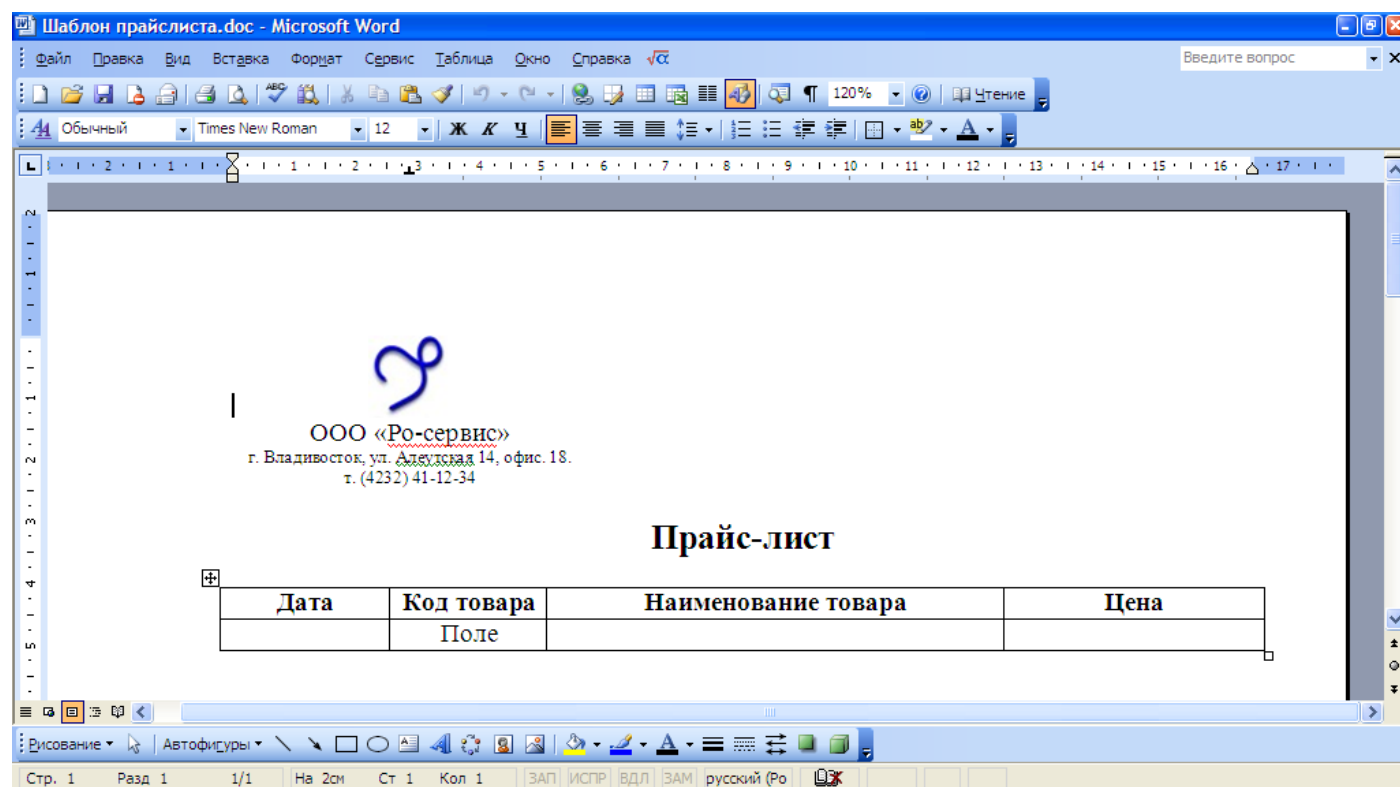
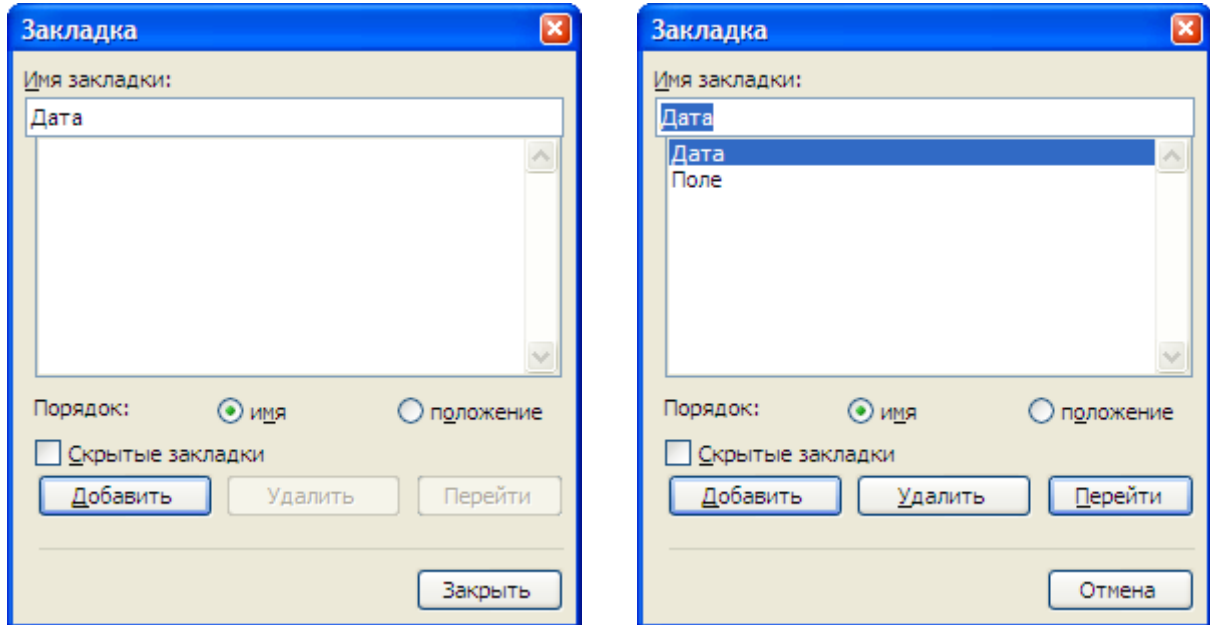


Рис. 1. Шаблон отчета «Прайс-лист»

Для того, чтобы вставить программно значения на место слов «Дата» и «Поле», необходимо в документе Word создать закладки на их месте.

2. Выделяем слово «Дата».
3. Вставка → Закладка...
4. В появившемся окне набираем имя закладки (Рис. 2а)



а

б

Рис. 2. Окно «Закладка».

5. Точно также делаем для слова «Поле».

Имена закладок можно задавать на любом языке. В имени закладки можно использовать только буквы и цифры, и символ нижнего подчеркивания. Имя закладки допускается **начинать только с буквы. Пробелы не допускаются.**

В результате должен получиться список закладок, текст на которых позже можно заменить из программы Delphi (Рис. 2б).

6. Переходим в проект Delphi. Добавляем на форму компонент TDDECLienConv и задаем следующие свойства (в этом же порядке):

Name = DDE

ServiceApplication = c:\Program Files\Microsoft Office\OFFICE11\WINWORD.EXE

ConnectMode = ddeManual

DdeService = WinWord

Для Word2007:

ServiceApplication = c:\Program Files\Microsoft Office\OFFICE12\WINWORD.EXE

7. Открываем описание метода пункта меню «Прайс-лист». В секции var добавляем переменные:

```
var
  Command, // строка команды
  NameTemplate, // путь к файлу шаблона
  NameResult: string; // путь до отчета
```

```
Macros: PChar; // строка команды на выполнение в Word
i, j: integer; // переменные цикла
```

8. В описании метода дописываем следующий программный код:

```
// задаем путь до файла шаблона
NameTemplate := GetCurrentDir + '\Шаблон прайслиста.doc';

TimeSeparator := '_'; // Задаем разделитель времени (по умолчанию «:»)
NameResult := GetCurrentDir + '\Прайс ' + DateToStr(Now) + ' '
+ TimeToStr(Now) + '.doc';

// Для того, чтобы не было совпадений сохраняемых имен файлов
// Прайс-листов, будем хранить в формате «Прайс Дата Время.doc»:
// Прайс 01.11.10 13_38_13.doc

// Далее в переменную Command добавляем команды, разделяя их []
// открываем файл шаблона и переходим в начало документа
Command := '[FileOpen "' + NameTemplate + '"][StartOfDocument]';
// сохраняем документ под новым именем
Command := Command + '[FileSaveAs "' + NameResult + '"]';

// переходим на закладку «Дата»
Command := Command + '[EditBookmark.name = "Дата", .Goto]';
// Вставляем текущую дату
Command := Command + '[Insert "' + DateToStr(Now) + '"]';

// переходим на закладку «Поле»
Command := Command + '[EditBookmark.name = "Поле", .Goto]';
// вставляем пустой текст
Command := Command + '[Insert " " ]';

// Заполняем основную таблицу. Данные берем из таблицы «Товары»
// Чтобы не писать много раз DM и ADOTableTovar используем оператор with
with DM, ADOTableTovar do
begin
  First; // переходим на первую запись в таблице «Товары»
  // переходим на две ячейки назад в документе в MSWord
  // Это нужно, чтобы после перехода в последнюю ячейку не вызвать
  // еще один переход, который добавит пустую строку в конце таблицы
  Command := Command + '[PrevCell][PrevCell]';
  for I := 0 to RecordCount - 1 do
  begin
    // так как в прайс-листе не используется кол-во на складе,
    // то полей на 1 меньше
    Command := Command + '[NextCell]';
    for j := 0 to FieldCount - 2 do
    begin
      // переходим на следующую ячейку в документе MSWord
      Command := Command + '[NextCell]';
      // Добавляем в таблицу значение поля j
      Command := Command + '[Insert "' + Fields[j].AsString + '"]'
    end;
    Next; // переходим к следующей записи в таблице «Товары»
  end;
end;

Command := Command + '[EndOfDocument]'; // переходим в конец документа
Command := Command + '[FileSave]'; // сохраняем файл

Macros := PChar(Command); // Преобразуем тип String в PChar
```

DDE.OpenLink; // Отключаем соединение с Word
DDE.ExecuteMacro(Macros, false); // Отправляем команды на выполнение

Создание отчета «Счет-фактура» по текущему заказу

Для выполнения отчета «Счет-фактура» необходимо использовать несколько таблиц. По этому предварительно создаем *Query* на *DataModule*.

1. Для Query настроим следующие свойства:

1.1. *Connection* – ADOConnection1

1.2. *SQL*:

```
select z_f.id_zakaz, tovar.tovar, z_t.kol, tovar.price, firma.firma  
from z_t  
left join tovar on tovar.id_tovar=z_t.id_tovar  
left join z_f on z_f.id_zakaz=z_t.id_zakaz  
left join firma on firma.id_firma=z_f.id_firma
```

1.3. *Active* – True

2. Создаем отчет (Рис. 3) так же, как и отчет «Прайс-лист». Список закладок представлен на (Рис. 4)

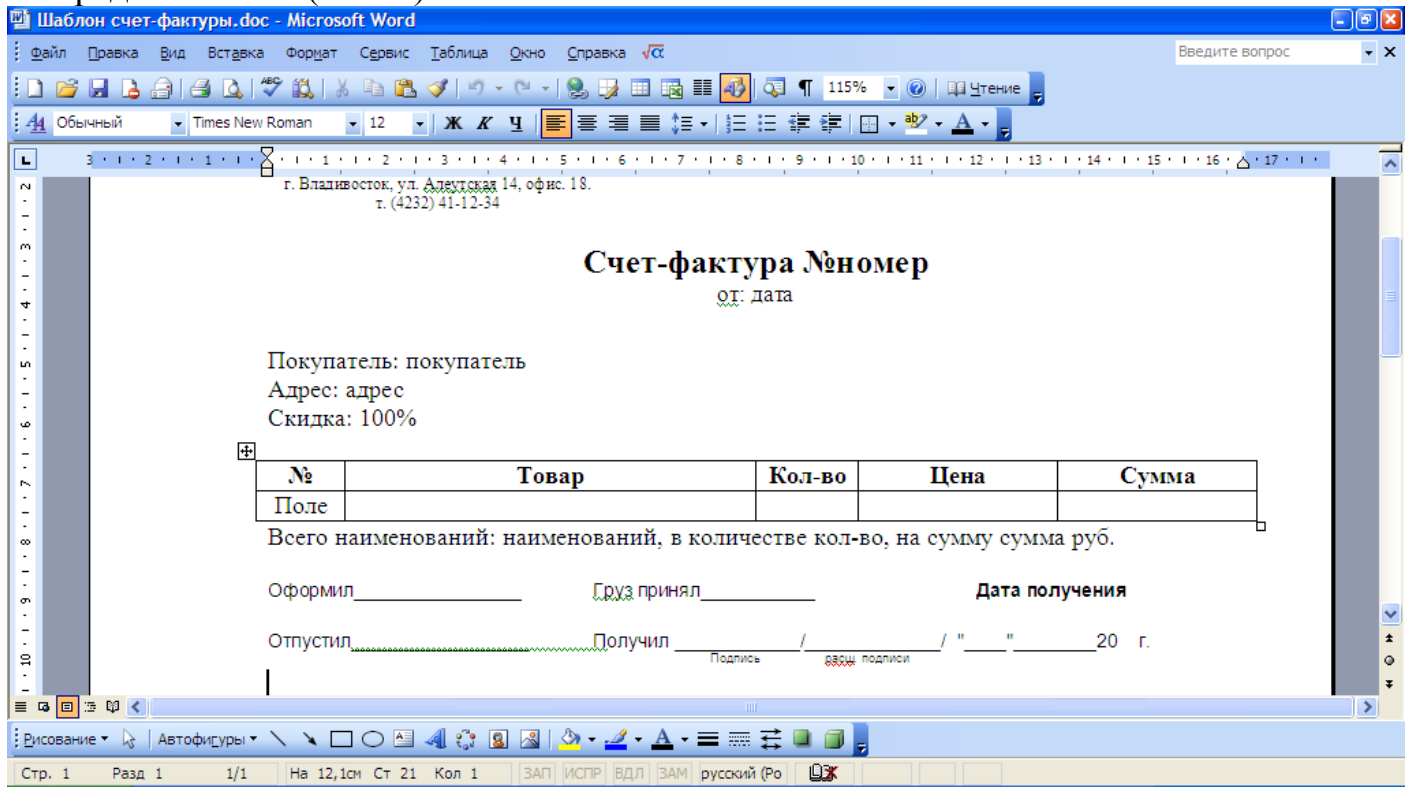


Рис.3 Макет отчета «Счет-фактура»

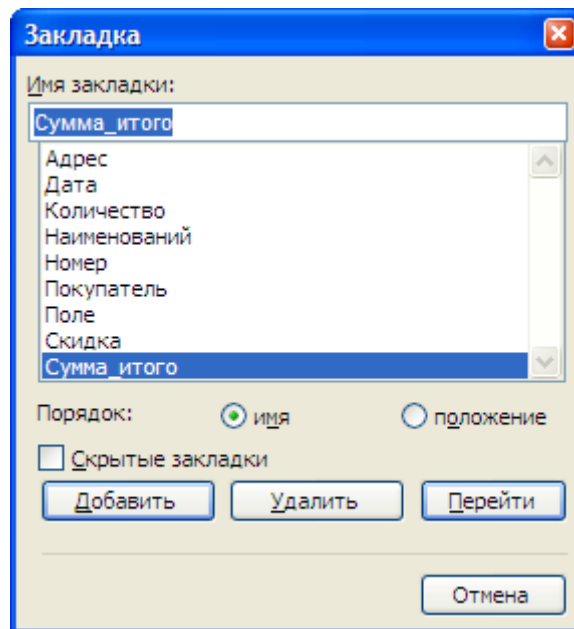


Рис. 4. Список закладок шаблона счет-фактуры.

3. Открываем описание метода пункта меню «Прайс-лист». В секции var то же, что и в случае с прайс-листом и еще одну переменную: ResDis: real для хранения значения, позволяющего применить скидку.
4. В описании метода дописываем следующий программный код:

```

TimeSeparator := '_'; // Задаем разделитель времени (по умолчанию «:»)
DecimalSeparator := '.'; // Задаем разделитель действит. чисел (по умолчанию «,»)

// задаем путь до файла шаблона счет-фактуры
NameTemplate := GetCurrentDir + '\Шаблон счет-фактуры.doc';

// файл будем хранить в формате «Фактура Дата Время.doc»
NameResult := GetCurrentDir + '\Фактура ' + DateToStr(Now) + ' '
  + TimeToStr(Now) + '.doc';

Command := Command + '[FileOpen "' + NameTemplate + '"][StartOfDocument]';
Command := Command + '[FileSaveAs "' + NameResult + '"]';

// Для всех запросов берем текущий номер заказа из таблицы z_f
// заполняем поля номер заказа, адрес, покупатель, дата
with DM, ADOQueryBase do
begin
  // создаем запрос к БД через компонент Query
  // Для текущего заказа вычисляем: Компанию, дату, адрес
  SQL.Clear;
  SQL.Add('SELECT id_zakaz, firma, date, address');
  SQL.Add('FROM z_f INNER JOIN firma');
  SQL.Add('ON z_f.id_firma = firma.id_firma');
  SQL.Add('WHERE id_zakaz = ' +
    ADOTableZakazFirma.FieldName('id_zakaz').AsString);

  // запускаем запрос безопасно, чтобы отловить ошибки
  try
    ExecSQL;
    Active := true;
  
```

except

ShowMessage('Ошибка 1');

end;

// заменяем соответствующие закладки на значения из БД

Command := Command + '[EditBookmark.name = "Номер", .Goto]';

Command := Command + '[Insert "' + FieldByName('id_zakaz').AsString + '"]';

Command := Command + '[EditBookmark.name = "Дата", .Goto]';

Command := Command + '[Insert "' + FieldByName('Date').AsString + '"]';

Command := Command + '[EditBookmark.name = "Адрес", .Goto]';

Command := Command + '[Insert "' + FieldByName('Address').AsString + '"]';

Command := Command + '[EditBookmark.name = "Покупатель", .Goto]';

Command := Command + '[Insert "' + FieldByName('firma').AsString + '"]';

end;

with DM, ADOQueryBase do

begin

// вычисляем скидку

// Создаем запрос вычисляющий скидку для фирмы по номеру заказа

SQL.Clear;

SQL.Add('SELECT discount');

SQL.Add('FROM z_f INNER JOIN firma');

SQL.Add('ON z_f.id_firma=firma.id_firma');

SQL.Add('WHERE id_zakaz = ' +

ADOTableZakazFirma.FieldByName('id_zakaz').AsString);

try

ExecSQL;

Active := true;

except

ShowMessage('Ошибка 2-1');

end;

*// так как поле в запросе одно и одна запись, то можно обратиться
// к полю по номеру. Первое поле имеет номер 0.*

// заносим значение скидки в счет-фактуру

Command := Command + '[EditBookmark.name = "Скидка", .Goto]';

Command := Command + '[Insert "' + Fields[0].AsString + '"]';

*// вычисляем величину ResDis, на которую нужно умножить цену, чтобы
// она уменьшилась на величину скидки*

resdis := (100 - Fields[0].AsInteger) / 100;

end;

// заполняем основную таблицу

Command := Command + '[EditBookmark.name = "Поле", .Goto]';

Command := Command + '[Insert " "]';

with DM, ADOQueryBase do

begin

// выполняем запрос к БД, который выводит для текущего заказа

// имя товара, количество, цену и сумму(с учетом скидки)

SQL.Clear;

SQL.Add('SELECT z_t.id_zakaz, tovar.tovar, z_t.kol,');

SQL.Add('tovar.price, tovar.price*z_t.kol*' +

```

FloatToStr(resdis) + ' AS summa');
SQL.Add('FROM z_t INNER JOIN tovar');
SQL.Add('ON z_t.id_tovar = tovar.id_tovar');
SQL.Add('WHERE id_zakaz = ' +
  ADOTableZakazFirma.FieldName('id_zakaz').AsString);
try
  ExecSQL;
  Active := true;
except
  ShowMessage('Ошибка 2-2');
end;

First; // переходим на первую запись в запросе
Command := Command + '[PrevCell]'; // переходим на ячейку назад.
for i := 0 to RecordCount - 1 do
begin
  for j := 0 to FieldCount - 1 do
  begin
    Command := Command + '[NextCell]';
    // если первое поле в БД, то вставляем номер записи
    // если нет, то вставляем значение из запроса.
    if j = 0 then
      Command := Command + '[Insert "' + IntToStr(i + 1) + '"]'
    else
      Command := Command + '[Insert "' + Fields[j].AsString + '"]';
  end;
  Next;
end;
end;

// заполняем поле "количество наименований"
with DM, ADOQueryBase do
begin
  // Создаем запрос, вычисляющий количество наименований
  // для текущего заказа
  SQL.Clear;
  SQL.Add('SELECT COUNT(id_zakaz) AS kol_naimen');
  SQL.Add('FROM z_t');
  SQL.Add('WHERE id_zakaz = ');
  SQL.Add(ADOTableZakazFirma.FieldName('id_zakaz').AsString);
  SQL.Add('GROUP BY id_zakaz');
  try
    ExecSQL;
    Active := true;
  except
    ShowMessage('Ошибка 3');
  end;
  // заполняем соответствующую закладку
  Command := Command + '[EditBookmark.name = "Наименований", .Goto]';
  Command := Command + '[Insert "' + Fields[0].AsString + '"]';
end;

// заполняем поле "количество единиц товара итого"
with DM, ADOQueryBase do
begin

```

```

// Создаем запрос, вычисляющий количество единиц товара всего
// для текущего заказа
{самостоятельно}
// заполняем соответствующую закладку «Количество»
end;

// заполняем поле "сумму итого"
with DM, ADOQueryBase do
begin
// Создаем запрос, выводящий итоговую сумму с учетом скидки
// для текущего заказа
{Самостоятельно}
// заполняем соответствующую закладку «Сумма_итого»
end;

Command := Command + '[EndOfDocument]';
Command := Command + '[FileSave]';

Macros := PChar(Command); // преобразуем строку команды в PChar
DDE.OpenLink; // открываем соединение с MSWord
DDE.ExecuteMacro(Macros, false); // выполняем команды

```

Индивидуальное задание (самостоятельно):

1. Для всех вариантов заполнить поля отчета «Сумма_итого» и «Количество».
2. Для нечетных вариантов создать отчет, выводящий информацию о товарах, заказанных определенной фирмой. Написать команду автоматического вывода документа на принтер, назначенный по-умолчанию.
3. Для четных вариантов требуется создать отчет обо всех заказах, сделанных в определенный день. Написать команду автоматического закрытия текущего документа и завершения приложения Word.