

*Сидоренко В.А.*, старший преподаватель  
ФГБОУ ВО «КалмГУ»,  
г. Элиста, Российская Федерация

## **Использование MS Excel в программировании приложений на Delphi**

### **Аннотация**

В статье рассмотрен механизм использования Microsoft Excel в программировании приложений в среде разработки Delphi. Рассмотрены практические примеры такой работы. Данная статья будет полезна для разработчиков приложений, использующих среду программирования Delphi, в том числе и для баз данных.

**Ключевые слова:** Delphi и Microsoft Excel, технология OLE в Delphi, компонентное программирование в Delphi.

В учебном процессе для получения опыта при изучении технологии компонентного программирования в качестве практической задачи полезно рассмотреть технологию OLE на примере включения приложения MS Excel (как EXE-компонента) в проект на Delphi.

OLE (Object Linking and Embedding) – это подход в технологии компонентного программирования, при котором разделение компонентов происходит на уровне самостоятельных приложений. Он был положен корпорацией Microsoft в основу связывания и внедрения объектов. Сегодня OLE – это огромная и сложная технология, которая оказывает определяющее влияние на отдельные стили программирования. В общем случае технология OLE предназначена для объединения инструментальных возможностей разных приложений, независимо от их форматов данных, операционной среды, языка программирования и разработчика.

В Delphi есть возможность работы через OLE с некоторыми приложениями Microsoft, включая Excel. Для этого здесь имеются модули ExcelXP и ComObj. Понятно, что ComObj – для подключения EXE-компонентов, а конкретно ExcelXP – для работы непосредственно с Excel. Не смотря на то, что название ExcelXP намекает на давний MS Office XP, тем не менее, он точно так же работает и с более поздними версиями MS Office.

Практическая польза такой работы программы на предприятии (в офисе) заключается в следующем. Обычно имеется система, в которой информация организована в виде базы данных. Нередко возникает задача о получении каких-то справок, итогов, отчётов – обычно в виде таблиц. Рассматриваемая технология позволяет автоматизировать процесс формирования интересующих итогов в виде таблиц Excel. Есть, конечно, в Delphi и стандартные приёмы получения отчётов, как и в любой среде программирования, имеющей инструменты для разработки баз данных. Но они слишком шаблонны. Excel же

более гибкая и более универсальная система для получения как отчетов, так и других документов. Здесь можно заполнять таблицу Excel как по подготовленному шаблону, так и начиная с пустого листа.

Переходим к конкретному примеру. Допустим, в нашем приложении мы уже получили некоторый результат в виде набора данных, который предстоит передать в таблицу Excel.

№ пп	Ф.И.О.	дата рожд.	нач. леч.	кон. леч.	Сумма
1	Петров Иван Ильич	21.02.1970	11.02.2023	15.02.2023	1 234,50
2	Иванов Пётр Семёнович	13.11.1950	12.02.2023	18.02.2023	3 245,00
3	Воробьев Кузьма Семёнович	31.12.1960	13.03.2023	23.03.2023	5 450,00

Выгрузка

Рис.1. Исходные данные

Наша задача написать процедуру работы с Excel в плане создания файла с таблицей результатов. Для этого сначала в программе подключаем модули ExcelXP, ComObj и объявляем объекты Excel, ExcelWB, ExcelWS:

**uses**

ExcelXP, ComObj;

**var**

Excel: OleVariant; // Excel-приложение

ExcelWB: OleVariant; // Excel-книга

ExcelWS: OleVariant; // Excel-Лист

ExcelOpened: boolean;

И далее – процедура выгрузки данных в Excel-таблицу:

**procedure** TForm1.BitBtn1Click(Sender: TObject);

**var**

FName, ext: string;

ExcRange: OleVariant;

i: integer;

**procedure** Setka(R1,R2:string);

**begin**

ExcRange:= ExcelWS.Range[R1,R2];

ExcRange.Borders[xlEdgeLeft].LineStyle := xlContinuous;

ExcRange.Borders[xlEdgeLeft].Weight := xlMedium;

ExcRange.Borders[xlEdgeTop].LineStyle := xlContinuous;

ExcRange.Borders[xlEdgeTop].Weight := xlMedium;

ExcRange.Borders[xlEdgeBottom].LineStyle := xlContinuous;

ExcRange.Borders[xlEdgeBottom].Weight := xlMedium;

ExcRange.Borders[xlEdgeRight].LineStyle := xlContinuous;

ExcRange.Borders[xlEdgeRight].Weight := xlMedium;

ExcRange.Borders[xlInsideVertical].LineStyle :=

xlContinuous;

```

    try
        ExcRange.Borders[xlInsideHorizontal].LineStyle :=
            xlContinuous;
    except
        // skip
    end;
    ExcRange.Font.Size := 10;
    ExcRange.WrapText := true;
    ExcRange.VerticalAlignment := xlCenter;
end;

begin {BitBtn1Click}
    if not Table1.Active then exit;
    if Table1.RecordCount = 0 then exit;
    if not SaveDialog1.Execute then exit;
    FName:= SaveDialog1.FileName;
    ext:= LowerCase(ExtractFileExt(FName));
    if (ext <> '.xls') and (ext <> '.xlsx') then
        FName:= ChangeFileExt(FName, '.xls');
        // Подключаем Excel
    ExcelOpened:=false;
    try
        Excel:= GetActiveOleObject('Excel.Application');
        ExcelOpened:= true;
    except
        Excel:= CreateOleObject('Excel.Application');
    end;
    Excel.Visible:= false; // Прячем Excel
    ExcelWB:= Excel.Workbooks.Add;
    ExcelWS:= ExcelWB.WorkSheets.Item[1];
    // работаем с Excel-файлом...
    // Заголовок таблицы...
    ExcRange:= ExcelWS.Range['B1','E1'];
    ExcRange.HorizontalAlignment:= xlCenter;
    ExcRange.VerticalAlignment:= xlCenter;
    ExcRange.Merge;
    ExcRange.Font.Name:= 'Arial';
    ExcRange.Font.Size:= 12;
    ExcRange.Font.Bold:= True;
    ExcelWS.Cells[1,2]:= 'Список пролеченных в стационаре';
    ExcelWS.Cells[2,3]:= '2023 год';
    ExcelWS.Cells[2,3].Font.Bold:= true;
    ExcelWS.Cells[2,3].HorizontalAlignment:= xlRight;
    // Названия столбцов...
    ExcelWS.Columns[1].ColumnWidth:= 5.11;
    ExcelWS.Columns[2].ColumnWidth:= 5.89;
    ExcelWS.Columns[3].ColumnWidth:= 27.00;
    ExcelWS.Columns[4].ColumnWidth:= 10.00;
    ExcelWS.Columns[5].ColumnWidth:= 10.00;

```

```

ExcelWS.Columns[6].ColumnWidth:= 10.00;
ExcelWS.Columns[7].ColumnWidth:= 10.56;
ExcelWS.Rows[4].RowHeight:= 26.4;
ExcelWS.Cells[4,1]:= '№ п/п';
ExcelWS.Cells[4,2]:= 'Месяц';
ExcelWS.Cells[4,3]:= 'Ф.И.О.';
ExcelWS.Cells[4,4]:= 'Дата рождения';
ExcelWS.Cells[4,5]:= 'нач. леч.';
ExcelWS.Cells[4,6]:= 'конец леч.';
ExcelWS.Cells[4,7]:= 'Сумма';
Setka('A4','G4');
ExcRange:= ExcelWS.Range['A4','G4'];
ExcRange.HorizontalAlignment:= xlCenter;
ExcRange.VerticalAlignment:= xlCenter;
ExcRange.WrapText:= True;
ExcRange.Font.Name:= 'Arial';
ExcRange.Font.Size:= 10;
ExcelWS.Columns[2].HorizontalAlignment:= xlCenter;
ExcelWS.Columns[7].NumberFormat:= '# ##0,00';
    // заполнение таблицы данными
DataSource1.DataSet.DisableControls; // бездействие связи
i:= 5;
with Table1 do begin
    First;
    while not Eof do begin
        ExcelWS.Cells[i,1]:= Fields[0].AsInteger; // NPP
        ExcelWS.Cells[i,2]:= Fields[1].AsInteger; // Mes
        ExcelWS.Cells[i,3]:= Fields[2].AsString; // FIO
        ExcelWS.Cells[i,4]:= Fields[3].AsString; // DR
        ExcelWS.Cells[i,5]:= Fields[4].AsString; // Dat1
        ExcelWS.Cells[i,6]:= Fields[5].AsString; // Dat2
        ExcelWS.Cells[i,7]:= Fields[6].AsCurrency; // Sum
        inc(i);
    Next
    end;
end;
Setka('A5', 'G' + IntToStr(i-1));
ExcelWS.Cells[i,6]:= 'Итого';
ExcelWS.Cells[i,7].Formula:= '=SUM(G5:G'+IntToStr(i-1)+' )';
ExcelWS.Cells[i,6].Font.Bold:= true;
ExcelWS.Cells[i,7].Font.Bold:= true;

DataSource1.DataSet.EnableControls; // оживляем связь
Excel.ActiveWorkbook.SaveAs (Filename:=FName,
                               FileFormat:= xlNormal);
Excel.Visible:=true; // показываем результат
ShowMessage('Файл готов');

end; {BitBtn1Click}

```

	A	B	C	D	E	F	G
1	<b>Список пролеченных в стационаре</b>						
2	<b>2023 год</b>						
3							
4	<b>№ п/п</b>	<b>Месяц</b>	<b>Ф.И.О.</b>	<b>Дата рождения</b>	<b>нач. леч.</b>	<b>конец леч.</b>	<b>Сумма</b>
5	1	2	Петров Иван Ильич	21.02.1970	11.02.2023	15.02.2023	1 234,50
6	2	2	Иванов Пётр Семёнович	13.11.1950	12.02.2023	18.02.2023	3 245,00
7	3	3	Воробьёв Кузьма Семёнович	31.12.1960	13.03.2023	23.03.2023	5 450,00
8	<b>Итого</b>						<b>9 929,50</b>

Рис.2 Результат работы программы

Здесь мы смогли воспользоваться всеми инструментами Excel, включая форматирование ячеек и формулы. То есть, Delphi-программа выполняла те же действия, как если бы это делал пользователь вручную.

И ещё дополнительное полезное замечание. Как видим, текст работы Delphi с Excel очень похож на правила работы Visual Basic for Applications (VBA) применительно к Excel. Поэтому для удобства определения объектов, методов и функций Excel полезно пользоваться макросами Excel – изучать их реализацию в VBA и буквально брать текст оттуда, но при этом менять синтаксис VBA на Delphi. А названия объектов, методов, функций и констант – те же самые, что и в VBA.

#### Список использованной литературы

1. Н.Елманова, С.Трепалин, А.Тенцер. Delphi и технология COM. СПб: Питер, 2003. 698 с.
2. В.Н. Корняков. Программирование документов и приложений MS Office в Delphi. СПб.: БХВ-Петербург, 2005. 496 с.
3. Осипов Д.Л. Базы данных и Delphi. Теория и практика. СПб: БХВ-Петербург, 2011. 752 с
4. Шейкер Т.Д. Разработка приложений баз данных в системе Delphi: учеб. пособие. Владивосток: Изд-во ДВГТУ, 2009. 138 с.

Сидоренко В.А., 2023