

Объектная модель Excel

- **Объект Application**
- **Класс Workbook**
- **Класс Worksheet**
- **Объект Range**
- **Импорт (экспорт) данных в (из) Excel**

Объектная модель

Excel

Microsoft Excel (Microsoft Office Excel) — программа для работы с электронными таблицами, созданная корпорацией Microsoft для Microsoft Windows, Windows NT и Mac OS.

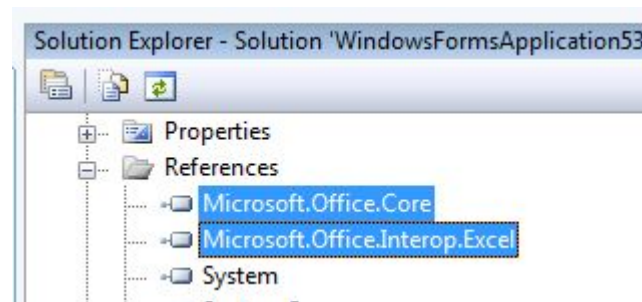
Основные объекты Excel:

- **Application** – приложение Excel
- **Workbook** – рабочая книга
- **Worksheet** – рабочий лист
- **Range** – ячейка или группа ячеек рабочего листа
- Cells

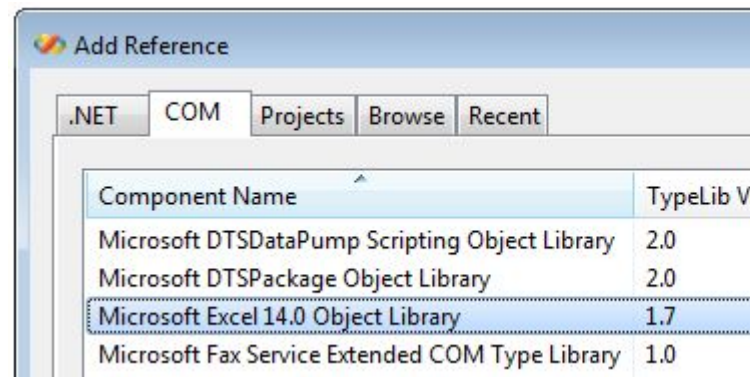
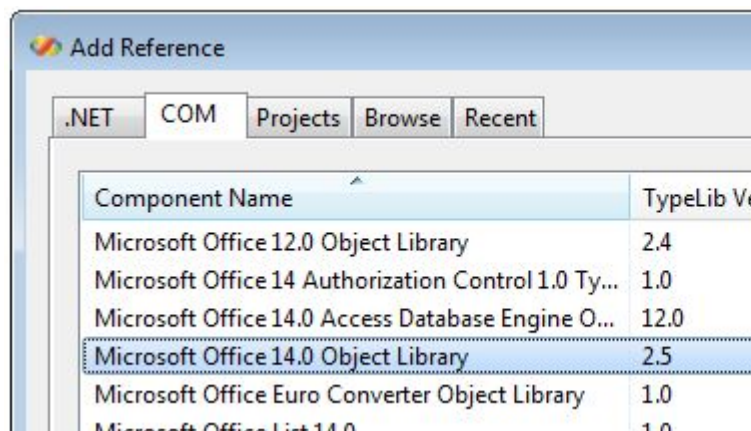
Объектная модель

Excel

using Microsoft.Office.Interop.Excel;



Для работы с объектами Excel, необходимо добавить соответствующие библиотеки:



Объект

Application

Объект **Application** приложения Excel представляет само приложение Excel.

Свойства Application, управляющие отображением:

DisplayAlerts – показывать предупреждающие сообщения

DisplayFormulaBar – показывать стандартную строку формулы для редактирования ячеек

DisplayFullScreen – выполнять в полноэкранном режиме

Объект

Application

Свойства Application, возвращающие объекты:

ActiveCell – возвращает ссылку на текущую активную ячейку активного окна.

ActiveChart – возвращает ссылку на текущую активную диаграмму.

ActiveSheet – возвращает ссылку на активный лист активной рабочей книги

ActiveWindow – возвращает ссылку на активное окно

Charts – возвращает набор объектов Sheet, содержащий ссылки на каждую из диаграмм активной рабочей книги

Selection – возвращает объект, выбранный в приложении.

Sheets – возвращает набор объектов Sheet, содержащий ссылки на каждый из листов активной рабочей книги

Workbooks – возвращает набор объектов Workbook, содержащий ссылки на каждую открытую рабочую книгу

Набор

Workbooks

Набор **Workbooks** позволяет работать со всеми открытыми рабочими книгами, создавать рабочую книгу и импортировать данные в новую рабочую книгу.

```
Excel.Application ObjExcel = new Excel.Application();
```

Создание рабочей книги:

```
Excel.Workbook wb =  
ThisApplication.Workbooks.Add(Type.Missing);
```

Заккрытие всех открытых рабочих книг:

```
ThisApplication.Workbooks.Close();
```

Набор

Workbooks

Открытие существующей рабочей книги:

```
Excel.Workbook wb = ThisApplication.Workbooks.Open(  
"C:\\YourPath\\Yourworkbook.xls",  
Type.Missing, Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, Type.Missing);
```

Type.Missing – параметры по умолчанию

Набор

Workbooks

Открытие текстового файла, базы данных или XML-файла как рабочей книги:

- OpenText
- OpenDatabase
- OpenXml

```
Excel.Workbook wb =  
ThisApplication.Workbooks.OpenText("C:\\Test.txt",  
Type.Missing, 3, Excel.XlTextParsingType.xlDelimited,  
Excel.XlTextQualifier.xlTextQualifierDoubleQuote,  
Type.Missing, Type.Missing, Type.Missing, True,  
Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, Type.Missing, Type.Missing);
```


Набор

Workbooks

Обращение к отдельным рабочим книгам:

```
Excel.Workbook wb = ThisApplication.Workbooks[1];
```

```
// До сохранения Book1:
```

```
wb = ThisApplication.Workbooks["Book1"];
```

```
// После сохранения Book1:
```

```
wb = ThisApplication.Workbooks["Book1.xls"];
```

Набор

Workbooks

Методы класса Workbook:

Activate – активизирует раб. книгу, выбирая лист рабочей КНИГИ:

```
ThisApplication.Workbooks[n].Activate;
```

Close – закрывает заданную рабочую книгу

```
ThisApplication.Workbooks[1].Close(false, Type.Missing,  
Type.Missing);
```

Protect (Unprotect) – устанавливают (снимают) защиту раб. КНИГИ

Save (SaveAs) – сохраняет рабочую книгу.

Набор

Workbooks

Вычисляет заданный диапазон:

```
ThisApplication.get_Range("A1", "B12").Calculate();
```

Evaluate – этот метод позволяет создать ссылку в виде строки, а потом при необходимости преобразовать ее в настоящую ссылку на объект или вычислить значение выражения.

Текст адреса ячейки поместить в ячейку с этим адресом:

```
Excel.Range rng =
```

```
    ThisApplication.get_Range("Evaluate", Type.Missing);
```

```
Excel.Range rngNew =
```

```
    (Excel.Range)ThisApplication.Evaluate(rng.get_Offset(0,  
1).Value2);
```

```
rngNew.Value2 = "Hello, World!";
```

Набор Workbooks

Программно выйти из Excel:

ThisApplication.**Quit()**;

Отменяет последнюю операцию:

ThisApplication.**Undo()**;

Объект

Range

Range – представляет ячейку, строку, столбец, выборку ячеек, содержащую один или несколько блоков ячеек (которые не обязательно должны быть смежными) и даже группу ячеек из разных листов.

Выделяем диапазон ячеек и очищаем:

```
ThisApplication.ActiveCell.CurrentRegion.ClearContents();
```

Объект

Range

Способы указания диапазона ячеек:

```
rng = ws.get_Range("A1", Type.Missing);  
rng = ws.get_Range("A1:B12", Type.Missing);  
rng = ws.get_Range("A1", "C5");  
rng = (Excel.Range)ws.Cells[1, 1];  
rng = ThisApplication.Range("SomeRangeName", Type.Missing);  
rng = (Excel.Range)ws.Rows["1:3", Type.Missing];  
rng = (Excel.Range)ws.Columns[3, Type.Missing];  
  
rng = (Excel.Range)ws.Cells[1, 1];  
for (int i = 1; i <= 5; i++) rng.get_Offset(i, 0).Value2 = i.ToString();  
  
ThisApplication.get_Range  
("C3", Type.Missing).CurrentRegion.Font.Bold = True;
```

Свойства

Application

RecentFiles – возвращает набор строк, содержащий имена всех файлов, которые показываются в меню File в списке недавно использовавшихся файлов.

```
private void ListRecentFiles()
{
    Excel.Range rng = (Excel.Range)ThisApplication.
        get_Range("RecentFiles", Type.Missing).Cells[1, 1];
    for (int i = 1; i <= ThisApplication.RecentFiles.Count; i++)
    {
        rng.get_Offset(i - 1, 0).Value2 =
            ThisApplication.RecentFiles[i].Name;
    }
}
```

Считывание данных из

Excel:

//Создаем экземпляр приложения Excel:

```
Microsoft.Office.Interop.Excel.Application ObjExcel = new  
Microsoft.Office.Interop.Excel.Application();
```

//Создаем книгу Excel:

```
Workbook ObjWorkBook = ObjExcel.Workbooks.Open(Файл,  
Type.Missing, Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, Type.Missing);
```

//Создаем лист Excel:

```
Worksheet ObjWorkSheet = ObjExcel.ActiveSheet as Worksheet;
```


Считывание данных из

Excel:
//Экземпляр ячейки:

```
Range rg = ObjWorkSheet.get_Range(«a6», «e7»);
```

```
foreach (Range item in rg)  
    MessageBox.Show(item.Text.ToString());
```

```
//Заккрытие приложения Excel
```

```
ObjExcel.Quit();
```

```
//Обнуляем созданные объекты
```

```
ObjWorkBook = null;
```

```
ObjWorkSheet = null;
```

```
ObjExcel = null;
```

```
//Вызываем сборщик мусора для освобождения памяти
```

```
GC.Collect();
```

Запись данных в

Excel:

```
Microsoft.Office.Interop.Excel.Application ObjExcel = new  
Microsoft.Office.Interop.Excel.Application();  
Workbook ObjWorkBook =  
ObjExcel.Workbooks.Add(System.Reflection.Missing.Value);  
Worksheet ObjWorkSheet =  
(Microsoft.Office.Interop.Excel.Worksheet)ObjWorkBook.Sheets[2];  
  
ObjWorkSheet.Activate(); //активируем второй лист  
ObjExcel.Cells[3, 2] = «Привет мир!»;  
ObjWorkBook.SaveAs(«C:\\111\\excelFile.xlsx»,  
Type.Missing, Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, XlSaveAsAccessMode.xlExclusive, Type.Missing,  
Type.Missing, Type.Missing, Type.Missing, Type.Missing);
```

Запись данных в

Excel:

//Заккрытие приложения Excel

ObjExcel.Quit();

//Обнуляем созданные объекты

ObjWorkBook = null;

ObjWorkSheet = null;

ObjExcel = null;

//Вызываем сборщик мусора для освобождения памяти

GC.Collect();

Лабораторная работа 9

Написать программное приложение, которое создает следующий Excel-файл:

	A	B	C	D	E
1			Почта		
2					
3		Номер	Наименование	Дата отправки	
4		1290	посылка	12.10.2015	
5		764	бандероль	04.11.2012	
6		6526	письмо	05.10.2012	
7					
8					