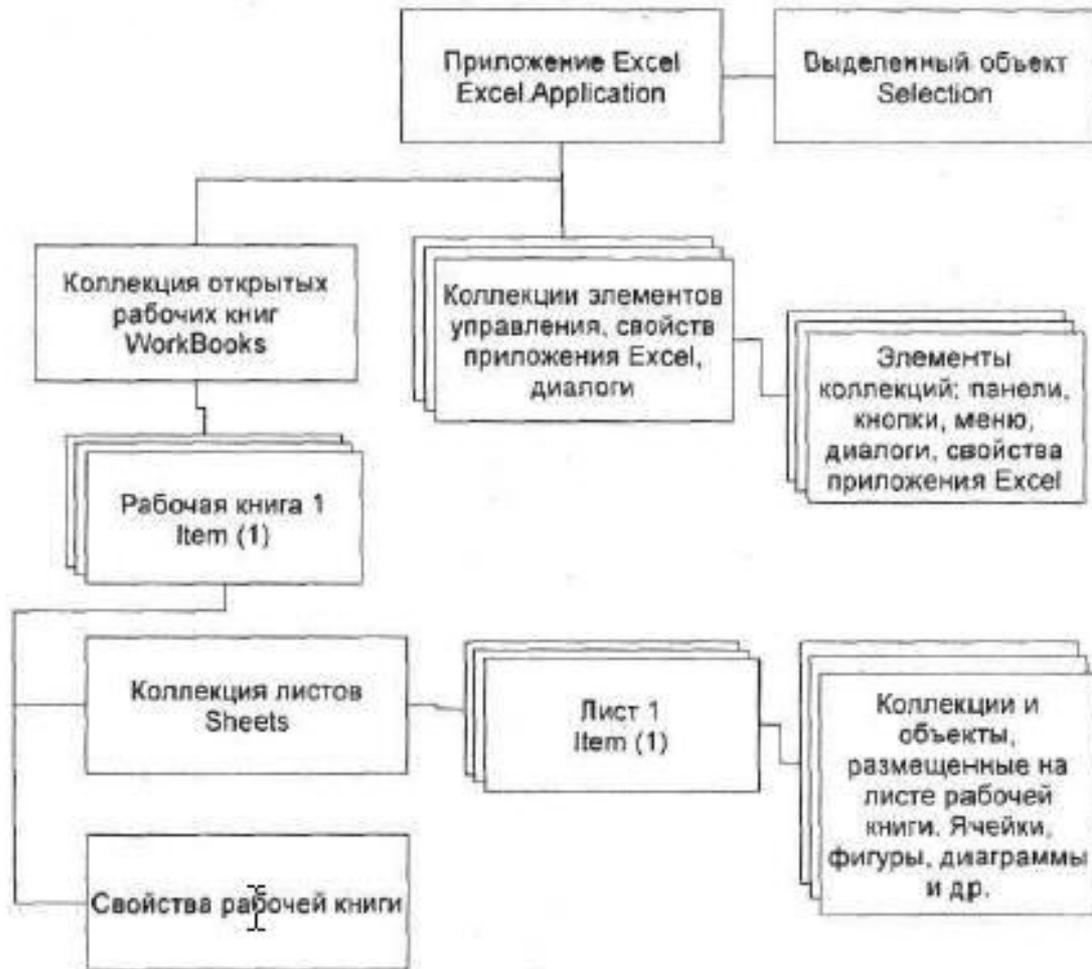


# **Объектная модель Excel**

**Тема №5**

# Структура объектной модели Excel



# Основные объекты Excel

- **Application** - приложение;
- **Workbook** – рабочая книга;
- **Worksheet** – рабочий лист;
- **Range** – диапазон ячеек.

# Основные операции с рабочими книгами

Создание рабочей книги:

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

---

Закрытие всех открытых рабочих книг:

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

---

Открытие существующей рабочей книги:

```
Excel.Workbook wb = ThisApplication.  
Workbooks.Open(@"C:\YourPath\книга1.xls",  
Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, Type.Missing);
```

# Просмотр всех листов рабочей книги

```
private void ListSheets()  
{  
    int i = 0;  
    Excel.Range rng =  
        ThisApplication.get_Range("Sheets",  
        Type.Missing);  
    foreach (Excel.Worksheet sh in  
        ThisWorkbook.Sheets)  
    {  
        rng.get_Offset(i, 0).Value2 =  
            sh.Name;  
        i = i + 1;  
    }  
}
```

# Основные компоненты класса Sheets

## Свойство `Visible`

Значения перечисления `XlSheetVisibility`:

- `XlSheetHidden`;
- `XlSheetVeryHidden`;
- `xlSheetVisible`

Пример:

```
( (Excel.Worksheet) ThisWorkbook.Sheets  
  [1] ).Visible =  
Excel.XlSheetVisibility.  
xlSheetVeryHidden;
```

# Основные компоненты класса Sheets

Добавление нового листа в рабочую книгу:

```
Excel.Sheet sh = ThisWorkbook.  
Sheets.Add(Type.Missing, Type.Missing,  
Type.Missing, Type.Missing);
```

Создание копии листа:

```
Excel.Worksheet) ThisWorkbook.Sheets[1]).  
Copy(Type.Missing, ThisWorkbook.Sheets  
[3]);
```

# Основные компоненты класса Sheets

Удаление заданного листа:

```
( (Excel.Worksheet)  
  ThisWorkbook.Sheets[1] ).Delete ();
```

---

Перенос листа:

```
Excel.Sheets shts = ThisWorkbook.  
  Sheets;  
  ( (Excel.Worksheet) shts[1] ).Move (Typ  
e. Missing, shts[shts.Count] );
```

# Основные компоненты класса Sheets

Печать выбранного объекта:

```
( (Excel.Worksheet) ThisApplication.  
  Sheets[1] ).PrintOut(1, 1, 2, true,  
  Type.Missing, Type.Missing,  
  Type.Missing, Type.Missing);
```

Предварительный просмотр:

```
( (Excel.Worksheet) ThisApplication.  
  Sheets[1] ).PrintPreview(false);
```

Выделение заданного объекта:

```
( (Excel.Worksheet) ThisApplication.  
  Sheets[1] ).Select(Type.Missing);
```

# Методы класса Workbook

Активизация рабочей книги:

```
ThisApplication.Workbooks[1].Activate();
```

---

Закрытие заданной рабочей книги:

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

---

Сохранение всех открытых рабочих книг:

```
foreach (Excel.Workbook wb in  
  ThisApplication.Workbooks)  
  wb.Save();
```

# Программное обращение к диапазонам

Предварительные описания:

```
Excel.Worksheet ws = (Excel.  
Worksheet) ThisWorkbook.Workshee  
ts[1];
```

```
Excel.Range rng, rng1, rng2;
```

Получение активной ячейки:

```
rng = ThisApplication.ActiveCell;
```

# Программное обращение к диапазонам

Непосредственное указание диапазона:

```
rng = ws.get_Range("A1",  
    Type.Missing);  
rng = ws.get_Range("A1:B12",  
    Type.Missing);
```

Указание строки и столбца:

```
rng = (Excel.Range)ws.Cells[1, 1];
```

# Программное обращение к диапазонам

Указание «углов» диапазона:

```
rng = ws.get_Range("A1", "C5");  
    rng = ws.get_Range("A1", "C5").Cells;  
    rng = ws.get_Range("A1", "C5").Rows;  
    rng = ws.get_Range("A1",  
    "C5").Columns;
```

Обращение к именованному диапазону:

```
rng = ThisApplication.Range  
    ("SomeRangeName", Type.Missing);
```

# Программное обращение к диапазонам

Обращение к заданной строке, столбцу или их диапазону:

```
rng = (Excel.Range)ws.Rows[1,  
Type.Missing];
```

```
rng = (Excel.Range)ws.Rows["1:3",  
Type.Missing];
```

```
rng = (Excel.Range)ws.Columns[3,  
Type.Missing];
```

# Программное обращение к диапазонам

Обращение к выделенным ячейкам:

```
rng = ThisApplication.Selection;
```

---

Объединение двух диапазонов:

```
rng = ThisApplication.get_Range  
("A1:D4, F2:G5", Type.Missing);
```

---

Пересечение двух диапазонов:

```
rng = ThisApplication.get_Range  
("A1:D16 B2:F14", Type.Missing);
```

# Программное обращение к диапазонам

Смещение относительно исходного диапазона:

```
rng = (Excel.Range) ws.Cells[1, 1];  
for (int i = 1; i <= 5; i++)  
    rng.get_Offset(i, 0).Value2 =  
    i.ToString();
```

Текущая область диапазона:  
свойство `CurrentRegion`

# Работа с диапазонами

Установка значений ячеек:  
свойство **Value2**

---

Автоматическое заполнение  
диапазонов:  
метод **AutoFill**

---

Сортировка данных в диапазоне:  
метод **Sort**

# Параметры метода Sort

Свойство	Тип	Описание
Key1	Object ( )	Первое поле сортировки: или имя диапазона (String), или объект Range; задает, по каким значениям выполняется сортировка
Order1	XLSortOrder (xlAscending, xlDescending)	Порядок сортировки по полю, заданному параметром Key1
Key2	Object ( )	Второе поле сортировки; этот параметр не используется при сортировке сводной таблицы
Type	Object	Задает, какие элементы участвуют в сортировке сводной таблицы; при сортировке обычного диапазона ни на что не влияет
Order2	XLSortOrder	Порядок сортировки по полю, заданному параметром Key2
Key3	Object ( )	Третье поле сортировки; этот параметр не используется при сортировке сводной таблицы

# Параметры метода Sort

<b>Order3</b>	<b>XlSortOrder</b>	Устанавливает порядок сортировки по полю, заданному параметром <b>Key3</b>
<b>Header</b>	<b>XlYesNoGuess</b> ( <b>xlGuess</b> , <b>xlNo</b> , <b>xlYes</b> )	Задаёт, содержит ли первая строка заголовочную информацию; по умолчанию используется значение <b>xlNo</b> ; если нужно, чтобы Excel определил это самостоятельно, укажите <b>xlGuess</b>
<b>OrderCustom</b>	<b>Integer</b>	Задаёт нестандартный порядок сортировки; указывает индекс с отсчётом от 1 в списке нестандартных порядков сортировки
<b>MatchCase</b>	<b>Boolean</b>	<b>True</b> - сортировка с учётом регистра букв, <b>False</b> - без учёта; для сводных таблиц не используется
<b>Orientation</b>	<b>XlSortOrientation</b> ( <b>xlSortRows</b> , <b>xlSortColumns</b> )	Ориентация сортировки

# Параметры метода Sort

<b>SortMethod</b>	<b>XISortMethod</b> (xlStroke, xlPinYin)	<b>Метод сортировки; применим не ко всем языкам (текущие значения применимы только к сортировке для китайского языка и не используются с другими языками)</b>
<b>DataOption1</b>	<b>XISortDataOption</b> (xlSortTextAsNumbers, xlSortNormal)	<b>Указывает, как сортировать текст в диапазоне, заданном параметром Key1; не применяется при сортировке сводных таблиц</b>
<b>DataOption2</b>	<b>XISortDataOption</b>	<b>Указывает, как сортировать текст в диапазоне, заданном параметром Key2; не применяется при сортировке сводных таблиц</b>
<b>DataOption3</b>	<b>XISortDataOption</b>	<b>Указывает, как сортировать текст в диапазоне, заданном параметром Key3; не применяется при сортировке сводных таблиц</b>

# Пример использования метода Sort

```
private void DemoSort()  
{  
    Excel.Range rng = ThisApplication.  
        get_Range("Fruits", Type.Missing);  
    rng.Sort(rng.Columns[1, Type.Missing],  
        Excel.XlSortOrder.xlAscending,  
        rng.Columns[2, Type.Missing], Type.Missing,  
        Excel.XlSortOrder.xlAscending, Type.Missing,  
Excel.XlSortOrder.xlAscending,  
        Excel.XlYesNoGuess.xlNo, Type.Missing,  
Type.Missing, Excel.XlSortOrientation.xlSortColumns,  
        Excel.XlSortMethod.xlPinYin,  
        Excel.XlSortDataOption.xlSortNormal,  
        Excel.XlSortDataOption.xlSortNormal,  
        Excel.XlSortDataOption.xlSortNormal);  
}
```

# Дополнительные возможности объектной модели Excel

Рисование границы вокруг ячеек:

```
range.BorderAround(Excel.XlLineStyle.  
xlContinuous, Excel.XlBorderWeight.  
xlThin, Excel.XlColorIndex.  
xlColorIndexAutomatic, Type.Missing)  
;
```

Заливка ячеек:

```
range.Interior.ColorIndex = 34;
```

# Дополнительные возможности объектной модели Excel

Рисование только нижней границы:

```
range.Borders [Excel.XlBordersIndex.  
xlEdgeBottom].LineStyle =  
Excel.XlLineStyle.xlContinuous;
```

---

Объединение ячеек:

```
range.Merge (Type.Missing) ;
```

---

Настройка шрифта в ячейках:

```
range.Font.Size = 14 ;  
range.Font.Bold = true ;
```