

*Лабораторная работа № 12*  
**Модели объектов и автоматизация**

Цель работы: Использование возможностей автоматизации (Automation) для работы с приложениями Microsoft Office 2010.

**Задание 1 Использование окна Просмотр объектов**

В этом задании вы используете окно **Object Browser** для изучения различных моделей объектов в Microsoft Office 2010.

➤ Добавление ссылки на объектную библиотеку

1. Откройте редактор Visual Basic в новом документе Microsoft Word.
2. Установите ссылку на Microsoft Excel Object Library (в меню Visual Basic **Сервис (Tools) → Ссылки (References...)**, выбрать Microsoft Excel Object Library и нажать **ОК**)

➤ Просмотр модели объектов

1. Выберите пункт меню **Вид (View) → Просмотр объектов (Object Browser)** или F2.
2. В окне **Object Browser** выберите класс (Classes) **Worksheet**.
3. В «Компонентах» (**Members**) просмотрите три свойства (например, CodeName, Rows, Visible...) и три метода объекта **Worksheet** (например, Activate, Calculate, Move...).

➤ Поиск модели объектов

1. В окне Object Browser найдите все объектные библиотеки и классы, использующие **Range**. Задайте в поле **Поиск текста (Search Text)** – Range.
2. Сколько вариантов **Range** вы видите в панели **Результаты поиска (Search Results)**?

При поиске вы можете найти много вариантов искомого текста. В разных моделях объектов есть свойства, методы, объекты с одинаковыми именами. Вы должны быть уверены, что найденный объект есть тот, что необходим.

3. Сравните свойства и методы класса **Range** в Microsoft Word с классом **Range** в Microsoft Excel. Найдите их в справочной системе для каждого приложения.

**Задание 2 Работа с объектами**

В этом задании вы напишите код для работы со свойствами и методами объектов.

➤ Использование свойств

1. Откройте новую рабочую книгу Microsoft Excel, затем редактор Visual Basic.

2. Вставьте новый модуль для кода.
3. Создайте новую процедуру.
4. Используйте оператор **With** для вывода некоторых свойств объекта **Application** в окне Отладки (Immediate).

Используйте свойства **Caption**, **OrganizationName**, **StartupPath**, **UserName**, и **Version**.

5. Измените текст поля состояния, установив свойство **StatusBar** объекта **Application** равным некоторой строке.

```
Sub ExploreProperties()  
With Application  
' печать некоторых свойств объекта Application  
Debug.Print .Caption  
Debug.Print .OrganizationName  
Debug.Print .StartupPath  
Debug.Print .UserName  
Debug.Print .Version  
.StatusBar = "Hello World!"  
End With  
End Sub
```

6. Протестируйте процедуру и проверьте поля состояния в Microsoft Excel. Для просмотра поля состояния активизируйте рабочий лист Microsoft Excel и посмотрите на текст в левом нижнем углу окна приложения.

➤ Использование методов

1. Создайте новую процедуру.
2. Опишите объектную переменную типа **Worksheet**.
3. Присвойте этой переменной значение, равное третьей странице в этой книге.
4. Используйте метод **Activate** объекта **worksheet** для активизации этой страницы.
5. Используйте свойство **Range** объекта **worksheet** для задания значения ячейке A1.
6. Используйте метод **PrintPreview** для объекта **worksheet**.
7. Используйте метод **Help** объекта **Application** для вывода справки.

```
Sub ExploreMethods()  
Dim wksCurrent As Worksheet  
'  
Set wksCurrent = Worksheets("Лист3")  
wksCurrent.Activate  
wksCurrent.Range("A1") = "Hello World!"  
wksCurrent.PrintPreview  
Application.Help  
End Sub
```

8. Протестируйте процедуру.

### Задание 3 Работа с объектными переменными

В этом задании вы будете работать с объектными переменными.

➤ Описание объектных переменных

1. Создайте новый документ Word.
2. В проект этого документа добавьте новый модуль и процедуру типа **Sub**.
3. Опишите переменную типа **Object**.

Переменная описанная как **Object** обладает поздним связыванием с объектом.

4. Опишите вторую переменную типа **Document**.

Переменная, описанная как объект определенного типа обладает ранним связыванием с объектом.

➤ Использование объектных переменных

1. Введите имя переменной, связанной с документом, за ней поставьте оператор (.). Что вы увидите?

Так как было использовано раннее связывание, вы увидите список доступных методов и свойств.

2. Используйте оператор **Set** для связи переменной с активным документом.
3. Задайте вывод свойства **Name** объекта **Document** в окне Отладки (Immediate).
4. Введите имя переменной, описанной как **Object**, за ней поставьте оператор (.). Что вы увидите? Так как было использовано позднее связывание, вы не увидите никаких методов и свойств.
5. Используйте оператор **Set** для связи переменной с активным документом.
6. Используйте свойство **Name** объектной переменной для вывода его в окно Отладки (Immediate).

```
Sub ObjectVariables()  
Dim objMyObject As Object  
Dim docMyDocument As Document  
Set docMyDocument = Application.ActiveDocument  
Debug.Print docMyDocument.Name  
Set objMyObject = Application.ActiveDocument  
Debug.Print objMyObject.Name  
End Sub
```

➤ Протестируйте ваш код:

Запустите процедуры и посмотрите окно Отладки (Immediate)

### Задание 4 Работа с несколькими объектными библиотеками

В этом задании вы поработаете с несколькими библиотеками объектов.

➤ Описание объектов

1. Создайте новую презентацию Microsoft Power Point.
2. В редакторе Visual Basic, создайте новую процедуру типа **Sub**.
3. Опишите переменную типа **Excel.Worksheet**.
4. Выберите пункт меню **Отладка (Debug) → Компилировать Project (Compile Project)**.

Так как у вас нет ссылки на библиотеку объектов Microsoft Excel, вы получите сообщение об ошибке.

5. Задайте ссылку на библиотеку объектов Microsoft Excel.
6. Откомпилируйте проект снова.

Сообщений об ошибке не будет.

➤ Работа с несколькими объектными библиотеками

1. Задайте ссылку на библиотеку объектов Word.
2. Опишите переменную **rngWord** типа **Range**.

Вы создали переменную типа Range в библиотеке объектов Microsoft Excel.

3. Введите имя **rngWord**, за которым следует оператор точка (.).

Вы увидите свойства и методы для библиотеки Microsoft Excel объект **Range**.

➤ Изменение приоритетов модели объектов

1. Выберите пункт меню Сервис (Tools) → Ссылки (References).
2. Используйте кнопки Приоритет (Priority) для перемещения объектной библиотеки Word выше над объектной библиотекой Microsoft Excel.

Это приведет к тому, что описание типа **Range** будет связано с объектной библиотекой Word.

3. Введите имя переменной **rngWord**, за которым идет оператор точка (.).

Вы увидите свойства и методы **Range** из библиотеки Word.

➤ Описание переменной типа Range с указанием библиотеки

1. Измените описание **rngWord** на тип **Word.Range**.
2. Опишите переменную **rngExcel** типа **Excel.Range**.
3. Введите **rngWord**, за ним оператор точка ( . ).

Вы увидите свойства и методы объекта **Range** из библиотеки Word.

4. Введите **rngExcel**, за ним оператор точка ( . ).

Вы увидите свойства и методы объекта **Range** из библиотеки Excel

#### Задание 5 Использование автоматизации

В этом задании вы используете автоматизацию приложений Office 2010.

➤ Автоматизация Microsoft PowerPoint

1. Создайте новый документ Microsoft Excel или Word.
2. В редакторе Visual Basic, задайте ссылку на библиотеку PowerPoint.
3. Создайте новый модуль и новую процедуру типа **Sub**.
4. Опишите переменную типа **PowerPoint.Application**. Используйте функцию **CreateObject** для связи переменной с новой объектом PowerPoint.
5. Выведите сообщение о запуске PowerPoint.
6. Используйте метод **Quit** объекта **Application** для завершения работы с PowerPoint.

```
Set pptApp = CreateObject("PowerPoint.Application")
Sub AutomatePowerPoint()
Dim pptApp As PowerPoint.Application
MsgBox "Запуск PowerPoint!"
pptApp.Quit
End sub
```

➤ Протестируйте ваш код

1. Запустите вашу процедуру.
2. При появлении окна с сообщением, найдите PowerPoint. Видите ли вы его?

Хотя PowerPoint запущен. Он не видим.

3. Закройте окно сообщения.
  4. Перед кодом с выводом сообщения, установите свойство **Visible** объекта **Application** для PowerPoint равным True (pptApp.Visible = msoTrue)
  5. Запустите процедуру снова. Перед появлением окна сообщения PowerPoint станет видим.
- Добавление нового документа
- В имеющуюся процедуру добавьте операторы для создания и сохранения презентации PowerPoint.

```
Sub AutomatePowerPoint()
Dim pptApp As PowerPoint.Application
Dim pptDoc As PowerPoint.Presentation
' создание экземпляра PowerPoint
Set pptApp = CreateObject("PowerPoint.Application")
' сообщение для пользователя
MsgBox "Запуск PowerPoint!"
pptApp.Visible = msoTrue ' сделать PowerPoint видимым
' создание новой презентации
Set pptDoc = pptApp.Presentations.Add
' сохранение презентации
pptDoc.SaveAs ("C:\NewPPTFile.ppt")
pptApp.Quit ' закрытие PowerPoint
' сообщение для пользователя
MsgBox "PowerPoint закрыт!"
End Sub
```