

УДК 004.912

Внедрение в учебный процесс автоматизации заполнения документов для ведения договоров

*Суханов А.А., студент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Компьютерные системы и сети»*

*Акимов Д.Н., студент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Компьютерные системы и сети»*

*Научный руководитель: Смирнова Е.В. д.т.н, доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Компьютерные системы и сети»
bauman@bmstu.ru*

Кто из нас не сталкивался с долгим, томительным ожиданием, находясь в очереди за тем или иным документом, или же, наоборот, будучи «по другую сторону баррикад», не слышал в свой адрес упреки в медлительности, выполняя заполнение однотипных документов в соответствии с шаблоном?

Сколько времени можно было бы сохранить, если бы была возможность всю эту работу доверить машине, освободив врача/бухгалтера/юриста от столь долгой, трудоемкой, но при этом лишенной абсолютно любой интеллектуальности работы?

К счастью, Microsoft Office позволяет автоматизировать этот процесс, поэтому нам показалось интересным подготовить лабораторную работу для будущих юристов – выпускников МГТУ им. Н.Э. Баумана, чтобы, приобретя навыки разработки приложений в VBA, они смогли максимально упростить себе процесс создания документации.

Подход к автоматизации использования документов используется при создании лингвистических онтологий [3], для мониторинга состояния сетевого обслуживания [4], для автоматизированного анализа активностей пользователей социальных сетей и в многих других областях [5].

Немного упрощённая реализация языка программирования Visual Basic – Visual Basic for Applications (VBA, Visual Basic для приложений), встроенная в линейку продуктов Microsoft Office (включая версии для Mac OS), а также во многие другие программные пакеты, такие как AutoCAD, SolidWorks, CorelDraw, WordPerfect и

ESRI ArcGIS. VBA покрывает и расширяет функциональность ранее использовавшихся специализированных макро-языков, таких как [WordBasic](#).

Этот язык является интерпретируемым языком. Как и следует из его названия, VBA близок к Visual Basic. VBA, будучи языком, построенным на COM, позволяет использовать все доступные в операционной системе COM объекты и компоненты ActiveX. По сути, возможно создание приложения на основе Microsoft Word VBA, использующего только средства CorelDraw.

В будущем Microsoft планирует заменить VBA на Visual Studio Tools for Applications (VSTA) — инструментарий расширения функциональности приложений, основанный на Microsoft .NET.

Достоинства и недостатки среды приведены в таблице 1.

Таблица 1

Достоинства и недостатки VBA

Достоинства	Недостатки
<ul style="list-style-type: none">• Легкость освоения;• Выполнение скрипта в среде офисных приложений.	<ul style="list-style-type: none">• Проблемы совместимости;• Высокая открытость кода.

Макрокоманда, или макрос, — программный алгоритм действий, записанный пользователем. Часто макросы применяют для выполнения рутинных действий.

Во многих офисных продуктах (OpenOffice.org, Microsoft Office и др.), в графических программах (например, CorelDRAW) при обработке макроса автоматически выполняется заданная для каждого макроса последовательность действий — нажатия на клавиши, выбор пунктов меню и т. д. Предоставляется интерфейс для записи новых и перезаписи существующих макросов. Также существуют текстовые редакторы, поддерживающие макросы, например Notepad++.

Используя макросы, можно значительно (в десятки, а то и в сотни раз) увеличить эффективность работы с приложением. Помимо выполнения набора встроенных в приложение команд, макрос позволяет производить обработку внешних файлов, загрузку и передачу файлов и данных через интернет, чтение и изменение настроек операционной системы. В большинстве случаев макрорекордер (режим записи действий пользователя в виде макроса) выдаёт код, нуждающийся в доработке. Но, тем не менее, и такой макрос значительно сокращает время, требующееся на выполнение рутинных операций.

Корректный перевод термина с английского — «макрокоманда», слово же «макрос» получило распространение благодаря использованию в локализованных продуктах американской корпорации Microsoft.

В документе следует вести текст, слова, вместо которых должны быть подставлены данные из Excel выделить фигурными скобками (Рис. 1). В данном случае значения полей {Имя} и {Фамилия} будут подставлены из аналогичных полей Excel.

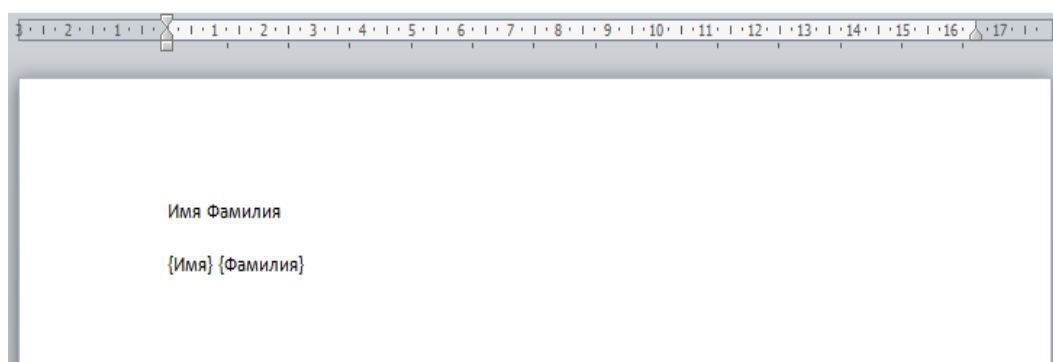


Рис. 1. Пример документа Word

В таблице Excel заполнить первый столбик ключевыми словами (словами, выделенными фигурными скобками) из документа Word. В данном случае это {Имя}, {Фамилия}. Напротив этих ячеек следует вписать данные, которые будут подставлены в документ (Рис. 2).

	A	B	C
1	{Имя}	Иван	
2	{Фамилия}	Иванов	
3			
4			
5			
6			
7			
8			
9			

Рис. 2. Заполнение полей

Далее следует подключить вкладку «Разработчик».

Для этого в Microsoft Office 2010: «Файл» -> «Параметры» -> «Настройка ленты» -> Поставить галочку около «Разработчик». Также можно нажать комбинацию клавиш Alt + F11. После этого появится окно Microsoft Visual Basic for Applications.

В окне Microsoft VBA требуется одключить Модуль к Вашему проекту.

Для этого: «Insert» -> «Module» (Рис. 3).

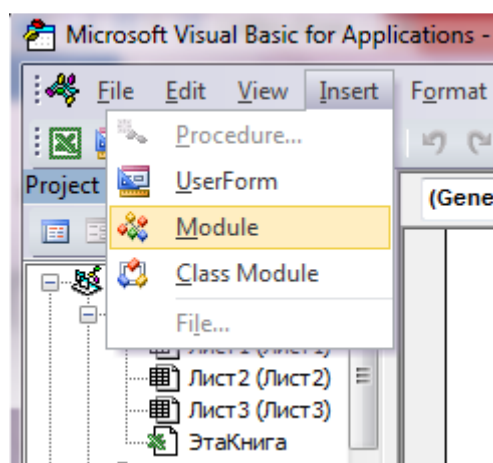


Рис. 3. Подключение модуля

После подключения модуля к проекту появляется пустой лист, в котором пишется текст макроса. Но прежде, чем приступать к написанию непосредственно кода, требуется подключить следующие библиотеки:

- Visual Basic For Applications;
- Microsoft Excel Object Library;
- OLE Automation;
- Microsoft Office Object Library;
- Microsoft Word Object Library;
- Microsoft Forms Object Library.

Объявление функции CountClipboardFormats:

В случае 64-разрядной системы:

```
Public Declare PtrSafe Function CountClipboardFormats Lib "user32" () As LongLong
```

В случае 32-разрядной системы:

```
Public Declare Function CountClipboardFormats Lib "user32" () As Long
```

Объявление глобальных переменных:

```
Dim appWd As Word.Application
```

```
Dim wdFind As Object
```

```
Dim ClipEmpty As New MSForms.DataObject
```

```
Dim ClipT As String
```

Подготовка вспомогательных функции и процедур.

- Функция проверки буфера обмена IsClipboardEmpty()

```
Function IsClipboardEmpty() As Boolean  
IsClipboardEmpty = (CountClipboardFormats() = 0)  
End Function
```

- Процедура проверки буфера обмена CheckClipBrd()

```
Sub CheckClipBrd()  
If IsClipboardEmpty() = True Then  
ClipEmpty.PutInClipboard  
End If  
End Sub
```

- Процедура вставки данных NoFormatPaste()

```
Sub NoFormatPaste()  
wdFind.Replacement.Text = ""  
wdFind.Forward = True  
wdFind.Wrap = wdFindContinue  
wdFind.Execute  
Call CheckClipBrd  
appWd.Selection.PasteSpecial DataType:=wdPasteText  
CutCopyMode = False  
End Sub
```

Подготовка основной процедуры CopyDatatoWord()

- Объявление переменных

```
Dim docWD As Word.Document  
Dim sheet1 As Object
```

- Код процедуры

```
Set appWd = CreateObject("Word.Application")  
appWd.Visible = True  
Set docWD = appWd.Documents.Open("C:\Users\Ñàøà\Desktop\doc.docx")  
‘открыть подготовленный в Word шаблон
```

```
Set sheet1 = Sheets("Лист1") ‘выбор листа, с которого происходит  
копирование
```

```
Set wdFind = appWd.Selection.Find
```

```
ClipT = " "
```

```
ClipEmpty.SetText ClipT
```

```
sheet1.Range("B1").Copy ‘копирование значения ячейки B1, то есть значения  
первого поля
```

```
If appWd.Selection.Find.Execute( _
```

FindText:=CStr(sheet1.Range("A1")), MatchWholeWord:=False) = True Then
'поиск в шаблоне Word элемента совпадающего по написанию со значением
ячейки A1

Call NoFormatPaste 'вызов процедуры вставки данных
End If

sheet1.Range("B2").Copy

If appWd.Selection.Find.Execute(_

FindText:=CStr(sheet1.Range("A2")), MatchWholeWord:=False) = True Then

Call NoFormatPaste

End If

docWD.SaveAs ("C:\Users\Ñàøà\Desktop\doc4.docx")

- Очистка данных

Set appWd = Nothing

Set docWD = Nothing

Set appXL = Nothing

Set wbXL = Nothing

После написания кода остается только запустить макрос

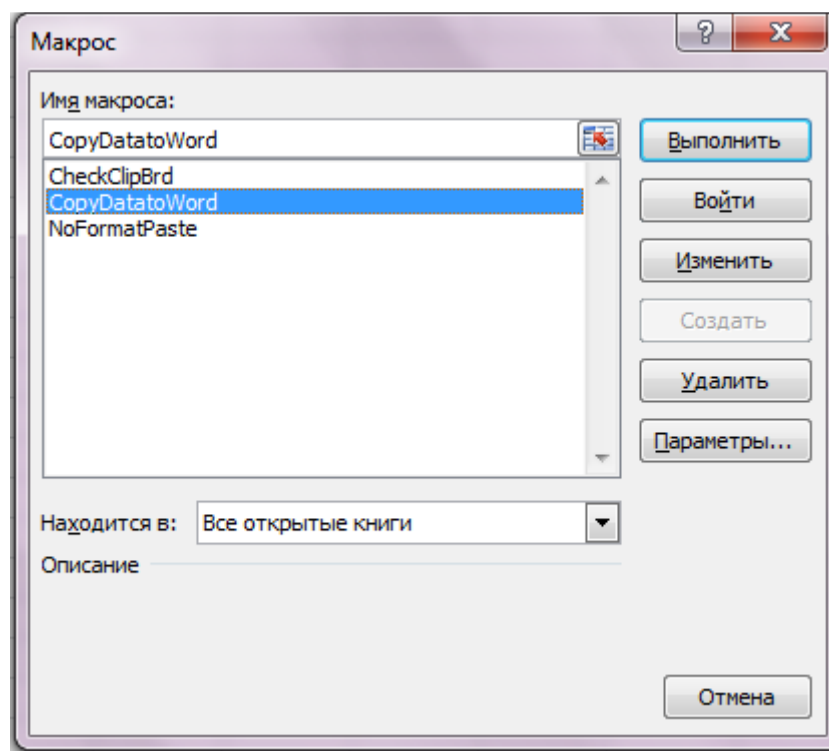


Рис. 4. Запуск макроса

Список литературы

1. Абрамов А.В. Создание лингвистической онтологии образовательной предметной области. Режим доступа <http://scientific-notes.ru/pdf/014-5.pdf> (дата обращения 04.12.2014).
2. Петров А.А. Программа мониторинга состояния сетевого обслуживания. Функциональная архитектура. Режим доступа <http://istcit.svgu.ru/files/letters/petrov.pdf> (дата обращения 04.12.2014).
3. Бурлуцкий В.В., Хасаншин Л. И. Автоматизированный анализ активностей пользователей социальных сетей для выявления общественной реакции // Вестник Югорского Государственного Университета. 2012. Вып. 3 (26). С. 62–65.