

Организация и инструментальные средства создания интерактивного учебного пособия по VBA.

Алексеев А.Ю., Васина М.В., Колобов А.А.

В последнее время наблюдается устойчивый рост интереса к изучению Visual Basic For Application (VBA) – встроенного языка программирования приложений Microsoft Office, используемого и в приложениях многих других разработчиков. Изучение VBA оказывается равно востребовано как при подготовке профессиональных разработчиков программного обеспечения, для которых VBA может служить средством интеграции разнородных приложений, так и в начальном курсе программирования, где VBA предоставляет исключительные возможности наглядного освоения объектных моделей и методов объектно-ориентированного программирования [1,2].

Изучение любого языка программирования (VBA – не исключение) предполагает разбор обучаемым большого количества фрагментов кода на изучаемом языке. Включение таких фрагментов в учебный курс позволяет проиллюстрировать правила применения синтаксических конструкций, приучает к принятому стилю записи программ, знакомит с типовыми приемами, используемыми в изучаемом языке. Более полное понимание подобных учебных фрагментов достигается, когда они изучаются в контексте некоторой законченной программы. Также изучению конструкций языка программирования способствует их пошаговое выполнение и анализ поведения программы при разных наборах исходных данных. При этом необходима рациональная организация учебного материала, такая, чтобы, с одной стороны, изучение иллюстративного материала было максимально приближено к анализу текста реальной программы, а, с другой, доступ к иллюстрациям и примерам не должен надолго отвлекать от основного текста пособия.

В традиционном – печатном – пособии это противоречие трудно разрешимо. В лучшем случае, развернутые примеры помещают в приложения или на компакт-диск, прикладываемый к пособию. К сожалению, обычно такие диски не имеют даже элементарной программы-оболочки, позволяющей быстро найти фрагменты, относящиеся к изучаемой в данный момент теме (см., напр., [3]). Сказанное заставляет обратиться к электронным учебным пособиям, которые имеют преимущества перед традиционными еще и с дидактической точки зрения. Например, в них можно интегрировать процесс усвоения знаний с получением умений и навыком, что положительным образом сказывается на приобретении обучаемым необходимых компетенций.

Электронные учебные пособия могут создаваться на основе Интернет-технологий, технологий СУБД, сочетания этих технологий, а также могут разрабатываться в специализированных обучающих средах, таких как MOODLE [4]. В рамках любого из указанных подходов достаточно просто реализуются нелинейные маршруты обучения, учащийся легко переходит из основного текста к примерам, требующим разбора. Однако при этом программы-иллюстрации оказываются оторванными от среды разработки и исполне-

ния, поэтому полноценное изучение иллюстрирующих примеров требует выполнения многочисленных действий, прямо не связанных с изучаемой темой. Так, для выполнения примера требуется скопировать или загрузить пример и поместить его текст в редактор VBA того приложения, для которого был создан пример. Кроме того, для программ, использующих объектные модели нескольких приложений, необходимо подключить соответствующие библиотеки.

При создании электронного учебного пособия по VBA некоторые из перечисленных проблем удастся разрешить, используя возможности среды разработки самого языка программирования. Свойства и методы интегрированной среды разработки и отладки (IDE) инкапсулированы в объекте VBE и доступны при выполнении приложения VBA. Использование объектной модели VBE позволяет организовать интерактивное учебное пособие по VBA следующим образом.

Учебное пособие представлено совокупностью документов Word, связанных гиперссылками. В основной корпус текстов пособия включаются лишь краткие, от одной до десятка строк, фрагменты кода. Законченные программы, иллюстрирующие рассматриваемые в некотором разделе пособия конструкции или приемы, могут быть внедрены в тот же документ пособия или в самостоятельный документ. Для перехода к просмотру и, при необходимости, выполнению примера достаточно нажать кнопку — управляющий элемент ActiveX. Кнопки размещаются в тех точках текста пособия, которые авторам пособия представляются наиболее естественными для перехода к рассмотрению развернутых примеров.

Обработчик события — нажатия — для любой из кнопок представляет собой обращение к макросу открытия в редакторе IDE заданной процедуры из заданного модуля в проекте заданного документа. Пример такого обработчика показан в листинге 1. В примере вызывается макрос *Module4.eOpenSubInProject*, открывающий в IDE процедуру *Пример2*, размещенную в модуле *Module1* проекта, содержащегося в рабочей книге Excel *K_работe14.xls*.

Листинг 1. Обработчик события, вызывающий макрос открытия

```
Private Sub CommandButton2_Click()  
Module4.eOpenSubInProject "Пример2", "Module1", _  
    "F:\ТЕКСТЫ\ТГПУ\Офисное пр-е Лекции\K_работe14.xls"  
End Sub
```

Макрос, открывающий проект в рабочей книге Excel и выделяющий в нем заданную процедуру, приведен в листинге 2. Макрос основан на процедурах из [6]. В макросе последовательно проверяется существование файла рабочей книги, отрывается заданная рабочая книга, открывается редактор VBA, проверяется существование заданного модуля и процедуры, заданная процедура размещается в верхней части окна редактора.

Листинг 2. Открыть заданную процедуру в книге Excel

```
Public Sub eOpenSubInProject(ByVal ProcName As String, _
                            ByVal ModuleName As String, _
                            Optional ByVal FullDocName As String = "omit")
    Dim aExc As New Excel.Application
    aExc.Visible = True
    Dim wb As Workbook
    If FullDocName = "omit" Then
        Set wb = ThisWorkbook
    Else
        ' проверка существования и открытие документа
        Dim fso As FileSystemObject
        Set fso = CreateObject("Scripting.FileSystemObject")
        ' Если не указан путь к документу, то полагаем,
        ' что он в текущей папке.
        If InStr(FullDocName, "\") = 0 Then _
            FullDocName = ThisWorkbook.Path & "\" & FullDocName
        If fso.FileExists(FullDocName) Then
            Set wb = Workbooks.Open(FullDocName)
        Else
            MsgBox "Файл " & FullDocName & " не существует"
            aExc.Quit : Set aExc = Nothing : Exit Sub
        End If
    End If
    ShowVisualBasicEditor = True
    Dim cm As CodeModule
    Dim cp As CodePane
    ' проверить, существует ли модуль
    If Not eModuleExist(ModuleName, wb) Then
        MsgBox "Модуль " & ModuleName & " не существует"
        aExc.Quit : Set aExc = Nothing : Exit Sub
    End If
    Set cm = wb.VBProject.VBComponents(ModuleName).CodeModule
    Set cp = cm.CodePane
    cp.Show
    If eProcExist(ProcName, ModuleName, wb) Then
        firstline = cm.ProcBodyLine(ProcName, vbext_pk_Proc)
        cp.TopLine = firstline 'первая строка просмотра
        cp.SetSelection firstline, 1, firstline, 80
    Else
        MsgBox "Процедура " & ProcName & " не существует"
        aExc.Quit : Set aExc = Nothing
    End If
End Sub

Public Function eModuleExist(ModuleName As String, _
                             wb As Workbook) As Boolean
    ' см. [6] стр. 72
    On Error Resume Next
    ModuleExist=Len(wb.VBProject.VBComponents(ModuleName).Name)<> 0
End Function

Function eProcExist(ProcName As String, ModuleName As String, _
                    wb As Workbook) As Boolean
    ' см. [6] стр. 73
```

```

On Error Resume Next
eProcExist = wb.VBProject.VBComponents(ModuleName). _
CodeModule.ProcStartLine(ProcName, vbext_pk_Proc) <> 0
End Function

```

Макросы, аналогичные приведенному, созданы для приложений Word, Access, PowerPoint. Таким образом, необходимые программы-примеры для любого приложения оказываются «на расстоянии одного клика» от основного текста пособия. Для выполнения примера достаточно второго клика — по кнопке **Run Macro** панели инструментов IDE. Если необходимые для учебного пособия примеры разработаны и отлажены, то включение их в электронное пособие сводится к внедрению в нужное место документа Word управляющего элемента — кнопки, и создания для нее обработчика события из единственной строки.

Объектная модель VBA позволяет не только программно переходить к просмотру интересующих процедур проекта, но и программно изменять проекты VBA. Это позволяет автоматизировать включение в текст пособия кнопок для просмотра и выполнения иллюстрационных примеров. Созданное для этого приложение реализовано на VBA. Внедрение кнопки в документ с помощью этого приложения сводится к размещению указателя в той строке пособия, где нужно внедрить кнопку, и заполнению формы (рис.1) с указанием параметров примера.

Рисунок 1. Форма для ввода параметров внедряемой кнопки

Приложение проверяет существование в документе пособия требуемых процедур открытия примеров и, при необходимости, производит их экспорт. Затем проверяется существование описанного в полях формы примера. Наконец, в документ внедряется управляющий элемент **CommandButton** и в проект лекции добавляется обработчик события **Click** этого элемента.

Предложенная технология и разработанные макросы позволяют создать электронное учебное пособие, представляющее совокупность связанных документов Word, в которые внедрены элементы управления ActeveX, обеспечивающие открытие редактора VBA, поиск модуля и процедуры, иллюстрирующей излагаемый учебный материал. На основе описанного подхода можно также организовать контроль маршрута обучения, анализировать затраты времени на изучение примеров, решать другие задачи управления обучением.

Библиографический список

1. Алексеев А.Ю., Сорокина Н.В. «Офисное программирование» как инструмент подготовки математиков-программистов специальности 351500 «Математическое обеспечение и администрирование информационных систем». — «Университет XXI века: научное измерение», Тула, ТГПУ им. Л.Н.Толстого, 2010. Т.1., стр.67-70
2. Колобов А.А. Изучение VBA, в качестве первого языка программирования // Актуальные проблемы гуманитарных и естественных наук. Москва. – 2011. – № 6. – С. 142 -144.
3. Туркин О.В. VBA. Практическое программирование. – М.: СОЛОН-ПРЕСС, 2007 – 128 с.
4. Живенков А.Н., Иванова О.Г. Реализация информационной адаптивной системы обучения на базе LMS MOODLE- «Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии», Воронеж, ВГУ, 2010. № 2. С. 88-92.
5. Михеев Р.Н. VBA и программирование в MS Office для пользователей – Спб.:БХВ-Петербург, 2006 – 384 с.
6. Гарнаев А.Ю. Самоучитель VBA – Спб.:БХВ-Петербург, 2004 – 560 с.