



فصل

کاربرد VBA برای ارتباط برنامه‌های ویژوال بیسیک با Microsoft Office

هدف‌های رفتاری: پس از آموزش این فصل هنرجو می‌تواند:

- کاربرد VBA برای ارتباط برنامه‌های ویژوال بیسیک با برنامه‌های آفیس را شرح دهد.
- برنامه‌های ساده‌ای برای ارتباط برنامه‌های ویژوال بیسیک با برنامه‌های آفیس را ایجاد کند.

Visual Basic for Applications (VBA) از زبان ویژوال بیسیک مشتق گرفته شده است که ارتباط زبان ویژوال بیسیک با بسته نرم افزاری Microsoft Office می‌باشد. توابع در VBA از نظر ساختار شبیه به توابع دیگر زبان‌های برنامه نویسی است. اغلب توابع VBA مشابه با توابعی است که در زبان ویژوال بیسیک با آنها آشنا شده‌اید مانند: توابع ریاضی Len() , Abs() , Sin() , ... و توابع رشته‌ای Len() , Mid() , Trim() , ... و غیره.

یک برنامه کاربردی وظایف خودش را از طریق شیء‌ها انجام می‌دهد. برنامه‌های کاربردی مانند Word و Excel دارای صدها شیء هستند که می‌توانند از طریق VBA کار کنند. مانند:

`ActiveCell.Range("A1").Borders.Weight xlThin`

شیء `ActiveCell` و `Range` و `Borders` که به صورت زیر مجموعه و `Weight` خصوصیت می‌باشد؛ همانند شیء فرم که کنترل‌های روی فرم شیء‌های دیگری هستند و هر یک از شیء‌ها خصوصیات مربوط به خود را دارند.


`Form1.Textbox1.text "VBA"`

به دلیل اینکه Office از صدها شیء تشکیل شده است و ارتباط بین این شیء‌ها پیچیده است،

مایکروسافت، ابزاری را فراهم کرده است که بتوان با این شی‌ها به سادگی کار کرد. به کمک این ابزار می‌توان شیء‌ها را مشاهده و در هنگام نیاز از آنها استفاده کرد یا آنها را تغییر داد.

۱-۶- ارتباط Microsoft Office با Visual Basic

همه فایل‌های آفیس در درون خودشان دارای برنامه کاربردی ویژوال بیسیک هستند و می‌توان همانند VB از طریق IDE فایل، دستوراتی را تایپ نمود تا در فایل اجرا شوند. در برنامه‌های کاربردی آفیس ابزاری به نام Macro وجود دارد که به کاربر در انجام بعضی از فعالیت‌های تکراری کمک می‌کند. برای مثال کاربر می‌تواند یک ماکرو برای تولید یک جدول و رنگ متن و چیدمان اطلاعات و... ضبط نموده و آن را بر روی یک فایل دیگر و یا هر جایی که نیاز دارد، اجرا نماید. البته باید به این نکته توجه کرد که عمل ضبط ماکرو همان کدهای VBA است که تولید و در قالب یک Sub در یک Module ذخیره شده است.

 **نکته:** توسط Macro می‌توان دستوراتی را به صورت اتوماتیک تولید و آنها را در قالب یک Sub مشاهده و در صورت نیاز تغییر یا حذف نمود.




ایجاد یک ماکرو ساده: یک ماکرو با نام Macro1 در محیط Excel تولید نمایید (شکل ۱-۶) که فعالیت‌های زیر در آن انجام شود:

- در ردیف اول تولید اعداد اتوماتیک از ۳۰ تا ۳۳
- ردیف ۳ ستون A، فرمول ("Visual") Len
- ردیف ۴ ستون A، فرمول ABS - 2
- ردیف ۵ ستون A، فرمول Sin(A1)

	A	B	C	D	E
۱	۳۰	۳۱	۳۲	۳۳	
۲					
۳	۶				
۴	۲				
۵	-۰.۹۸۸				
۶					

شکل ۱-۶

بعد از اتمام ضبط ماکرو، کدهای تولید شده را مشاهده نمایید.

 **نکته:** مسیر مشاهده محیط IDE و ژوال در آفیس (فشار کلیدهای (Alt + F11) : **OFFICE 2003** از منوی Tools زیر منوی Macro و انتخاب گزینه

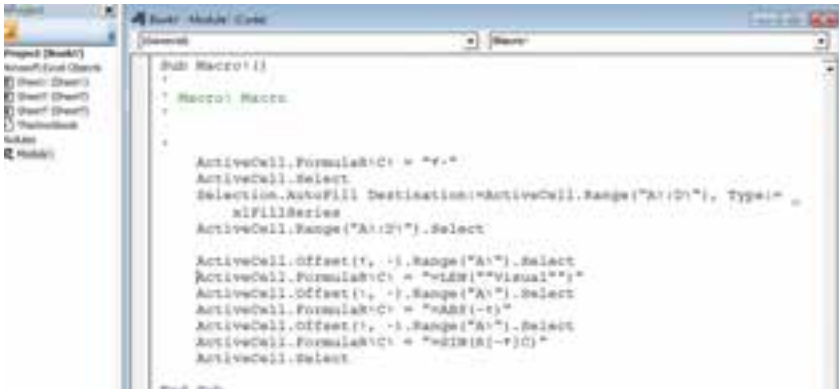
Visual Basic Editor

OFFICE 2007 : از زبانه Developer انتخاب گزینه Visual Basic

(مشاهده Developer با انتخاب گزینه Show Developer tab in the Ribbon

از (Excel Option

شکل ۲-۶ کدهای ماکرو ضبط شده در Moudul را نشان می دهد.



```
Sub Macro1()  
Macro1 Macro  
1  
2  
3  
ActiveCell.FormulaR1C1 = "=PI"  
ActiveCell.Select  
Selection.AutoFill Destination:=ActiveCell.Range("A1:C1"), Type:=xlFillSeries  
ActiveCell.Range("A1:C1").Select  
ActiveCell.Offset(1, -1).Range("A").Select  
ActiveCell.FormulaR1C1 = "=LEN("Visual")"  
ActiveCell.Offset(1, -1).Range("A").Select  
ActiveCell.FormulaR1C1 = "=ABS(-1)"  
ActiveCell.Offset(1, -1).Range("A").Select  
ActiveCell.FormulaR1C1 = "=SIN(PI-C)"  
ActiveCell.Select
```

شکل ۲-۶



ویرایش ماکرو از طریق کد نویسی برای خانه فرمول Sin، کادر ایجاد کرده و رنگ سلول را به

رنگ قرمز تغییر دهید :

بعد از آخرین خط دستور Macro1 کد زیر را بنویسید :

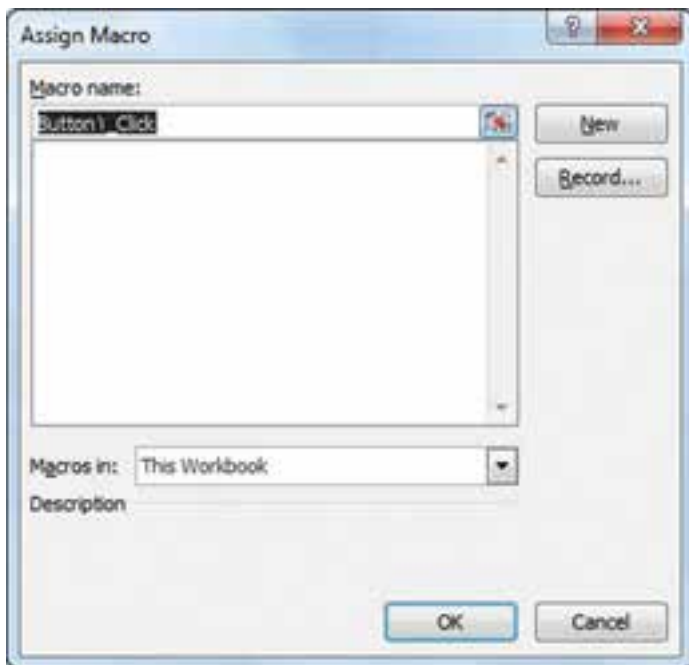
ActiveCell.Range("A1").Borders.Weight xlThin

ActiveCell.Range("A1").Interior.Color vbRed



ایجاد Sheet جدید با نام دلخواه بر روی Work Sheet به وسیله یک Command Button یک Command Button را انتخاب و بر روی Sheet اضافه کنید. روی کنترل اضافه شده

کلیک راست کرده و گزینه Assign Macro و بعد New را انتخاب نمایید (شکل ۶-۳).



شکل ۶-۳

در Modual برای رویداد Button1 Click تکه کد زیر را تایپ نمایید (شکل ۶-۴).

```
Sub Button1_Click()  
Worksheets.Add  
Worksheets(Worksheets.Count).Name = InputBox("نامی را وارد نمایید")  
End Sub
```

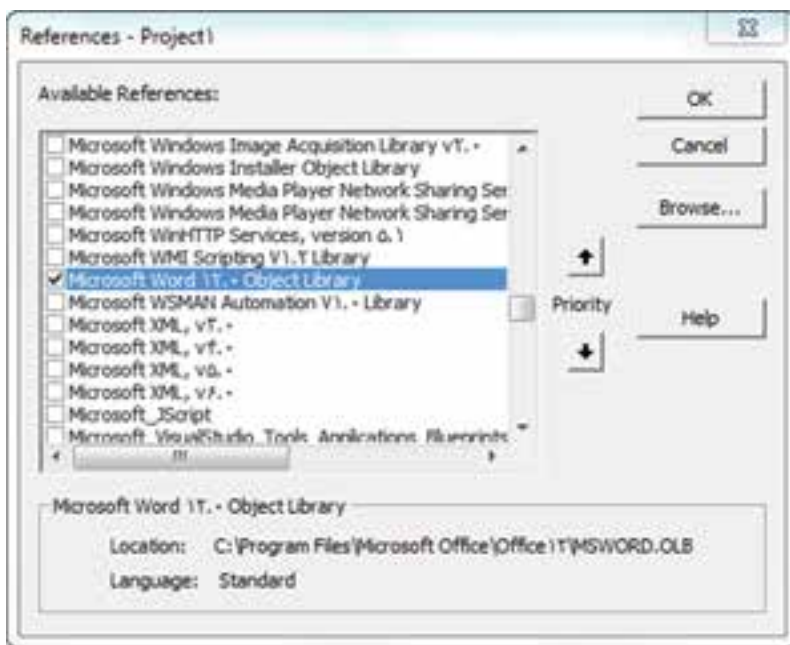
شکل ۶-۴

محیط باز شده را بسته و در Sheet مورد نظر دوباره Command Button را انتخاب کرده و اجرای برنامه را مشاهده نمایید.

۶-۲- ارتباط Visual Basic با Microsoft Office

نرم افزار، Visual Basic می تواند تمام مؤلفه های ActiveX، فایل کتابخانه ای (olb) نرم افزارهای نصب شده در سیستم عامل را مشاهده و در صورت نیاز به پروژه اضافه و از آن استفاده نماید. وقتی یک فایل کتابخانه ای نرم افزاری انتخاب شود می توان از آن به عنوان یک شیء در محیط برنامه نویسی استفاده نمود. که این شیء شامل خصوصیت ها (properties) و متدها (methods) می باشد. یک فایل کتابخانه ای مانند WORD را از گزینه References در منوی Project انتخاب نمایید (شکل ۵-۶).

توجه: حتماً نرم افزار word نصب باشد. و شماره ۱۲ یا ۱۱ یا ۱۰ در این گزینه بستگی به نسخه Office نصب شده دارد.



شکل ۵-۶ - انتخاب فایل کتابخانه ای WORD

در برنامه یک متغیر برای شیء Word ایجاد نمایید، تا بتوانید به دیگر شیء ها و خصوصیات Word دسترسی پیدا کنید.

۳-۶- بعضی از شیء‌های مورد استفاده در نرم‌افزار Word

Dim objWord As New Word Application	● تعریف یک متغیر از شیء کتابخانه‌ای
این متغیر می‌تواند تمام وظایف Word را در قالب یک شیء ارائه نماید	
objWord.Visible objWord.WindowState objWord.Quit	● مشاهده نرم‌افزار مربوطه ● مشاهده وضعیت پنجره نرم‌افزار باز شده ● بستن نرم‌افزار باز شده
objWord.Documents.Add objWord.Documents.Open objWord.Documents.Close objWord.Documents.Save	● با استفاده از شیء Document از word، می‌توان یک صفحه ایجاد، باز، ذخیره و یا سندی بسته شود
objWord.Selection.TypeText objWord.Selection.TypeParagraph	● برای درج متن به سند word ● برای درج یک پاراگراف خالی
objWord.Selection.Paragraphs objWord.Selection.PageSetup	● برای تعیین مشخصات شیء پاراگراف در سند ● برای تعیین مشخصات صفحه سند
objWord.ActiveDocument.SaveAs objWord.ActiveDocument.Tables	● سند Word را می‌تواند تحت نام جدید ذخیره نماید ● برای تعیین مشخصات شیء جدول
Set objWord = Nothing	● پاک کردن متغیر شیء از حافظه

بعضی از شیء‌های نرم‌افزارها پیچیده می‌باشند که می‌توانید برای اطلاعات بیشتر به MSDN مراجعه نمایید.



برنامه‌ای بنویسید که جمله درون کادر متنی را در یک سند Word نمایش دهد و آنرا با اسم Word App.Doc ذخیره نماید.

تکه کد برنامه به صورت زیر می‌باشد :

```
Public objWord As New Word.Application 'General VARIABLE
```

```
Private Sub cmdCreat_Click()
```

```

If txtWord.Text < vbNullString Then
    MsgBox "Insert Text In Textbox", vbInformation, "Warning"
    txtWord.SetFocus
    Exit Sub
End If
objWord.Visible = True

objWord.Documents.Add
objWord.Selection.Paragraphs.Alignment = wdAlignParagraphCenter
objWord.Selection.TypeText txtWord.Text
objWord.Selection.TypeParagraph
objWord.Selection.TypeText txtWord.Text
End Sub

Private Sub cmdQuit_Click()
    On Error Resume Next
    objWord.Visible = False
    objWord.Quit
    Set objWord = Nothing
End Sub

Private Sub cmdSave_Click()
    objWord.ActiveDocument.SaveAs FileName: App.Path & "\WordApp.doc"
    MsgBox "Word File Save in Path " & App.Path & "\WordApp.doc"
End Sub

```

کد را ذخیره و برنامه را اجرا نمایید (شکل ۶-۶).



شکل ۶-۶



در فصل ۱ برنامه دفترچه تلفن را با استفاده از توابع کار با فایل ها نوشتیم. در اینجا می خواهیم از آن فایل در محیط Excel گزارش تهیه کنیم. ابتدا گزینه Excel Microsoft 12.0 Object Library را انتخاب نمایید و در فرم طراحی شده در فصل ۱ دو Command Button با نام های cmdReport , cmdClose را اضافه نمایید و برای رویداد کلیک آنها کدهای زیر را بنویسید.

```
Dim objExcel As New Excel.Application  
Private Sub cmdClose Click()  
    objExcel.Visible = False
```



```
objExcel.Quit
Set objExcel = Nothing
End Sub
```

```
Private Sub cmdReport_Click()
```

```
    Dim iRow As Byte
```

```
    Dim objBook As New Excel.Workbook
```

```
    Dim objSheet As New Excel.Worksheet
```

```
    objExcel.Visible = True
```

```
    Set objBook = objExcel.Workbooks.Add
```

```
    Set objSheet = objBook.Worksheets("sheet1")
```

```
    objSheet.Range("A1:B1").Merge
```

```
    objSheet.Range("A1:B1").Value = "Telephone Book "
```

```
    objSheet.Range("A1:B1").Font.Bold = True
```

```
    objSheet.Range("A1:B1").Font.Color = vbBlue
```

```
    objSheet.Range("A1:B1").VerticalAlignment = xlTop
```

```
    objSheet.Range("A1:B1").HorizontalAlignment = xlCenter
```

```
    iRow = 2
```

```
    With objSheet
```

```
        .Cells(iRow, 1).Value = "Name"
```

```
        .Cells(iRow, 1).BorderAround LineStyle: = 3, Weight: = xlThin
```

```
        .Cells(iRow, 2).Value = "Number"
```

```
        .Cells(iRow, 2).BorderAround LineStyle: = 3, Weight: = xlThin
```

```
    End With
```

```
    For i = 1 To 10
```

```
        iRow = iRow + 1
```

```

With objSheet
    . Cells(iRow, 1).Value = recperson.Name
    . Cells(iRow, 1).BorderAround LineStyle: 1, Weight: xlThin
    . Cells(iRow, 2).Value = recperson.Tel
    . Cells(iRow, 2).BorderAround LineStyle: 1, Weight: xlThin
End With

Next
objSheet.Columns.AutoFit
objSheet.Name = "Telephone"
If Dir(App.Path & "\tel.xls") = "tel.xls" Then
    Kill App.Path & "\tel.xls"
    objBook.SaveAs App.Path & "\tel.xls"
Else
    objBook.SaveAs App.Path & "\tel.xls"
End If
Set objSheet = Nothing
Set objBook = Nothing
End Sub

```

برنامه را ذخیره و اجرا نمایید.

تحقیق: چگونه می‌توان به یک فایل اکسل موجود اطلاعات دیگری اضافه نمود.

خودآزمایی

پروژه‌ای را ایجاد کنید که عمل بررسی املاي کلمات را در سندهاي word انجام دهد (spelling checker)