METODE AVANSATE DE GESTIUNE A DOCUMENTELOR ȘI A SISTEMELOR DE CALCUL
- CURS 7 -

Asist. Diana – Florina Șotropa

www.cs.ubbcluj.ro/~diana.sotropa

INTRODUCERE ÎN VBA

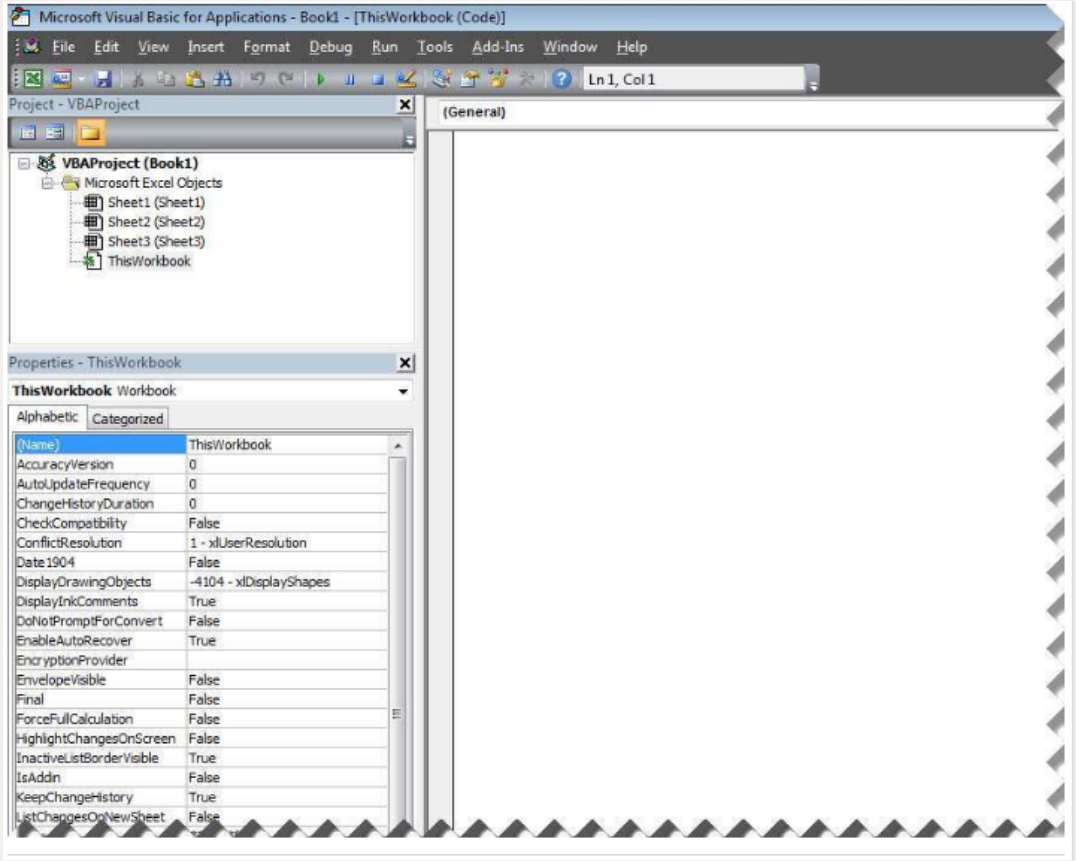
(**V**isual **B**asic for **A**pplications)

Visual Basic for Applications

- Limbaj de programare orientat pe evenimente
- Utilizat de aplicațiile Microsoft Office (Word, Excel, Access)

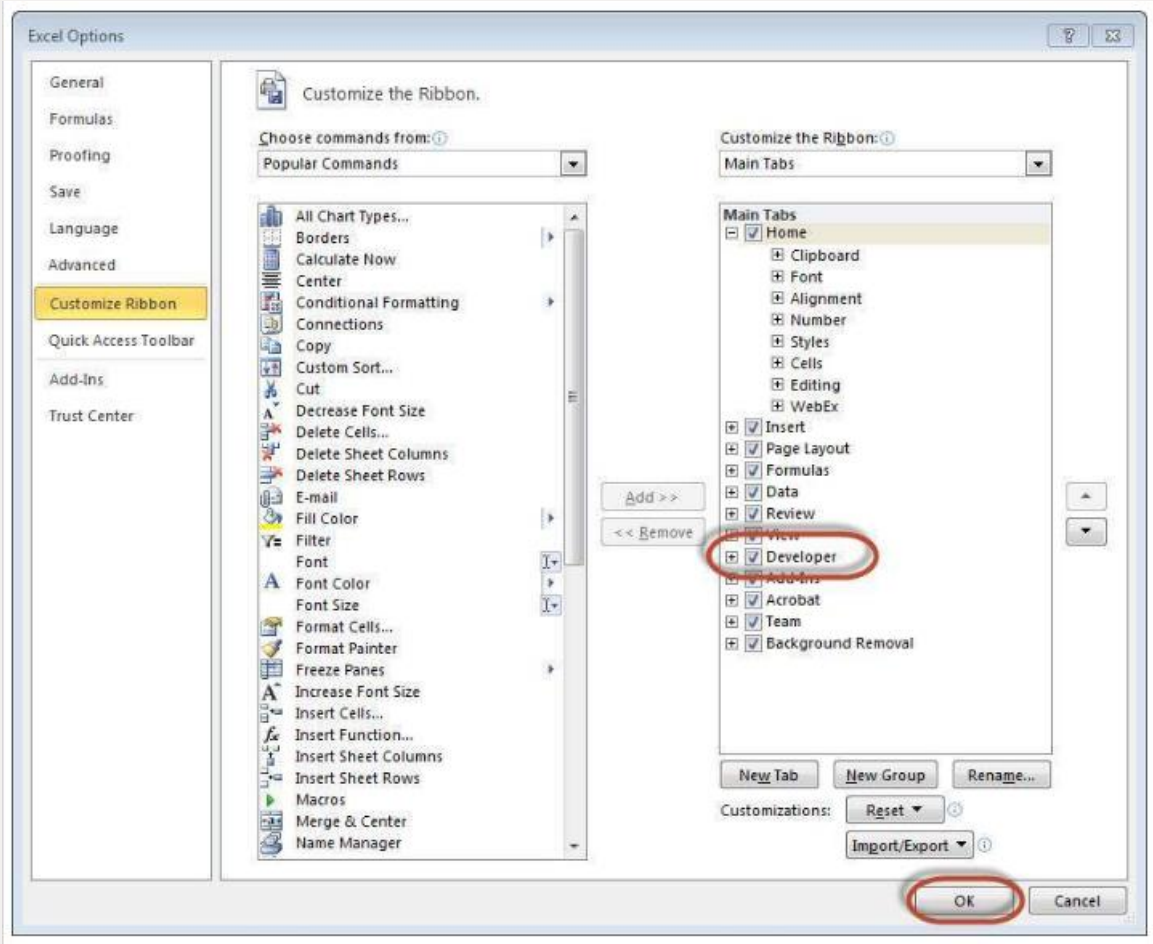
Visual Basic for Applications

- ALT + F11 – deschiderea ferestrei VBA
- F8 – rulare pas cu pas; F5 - rulare



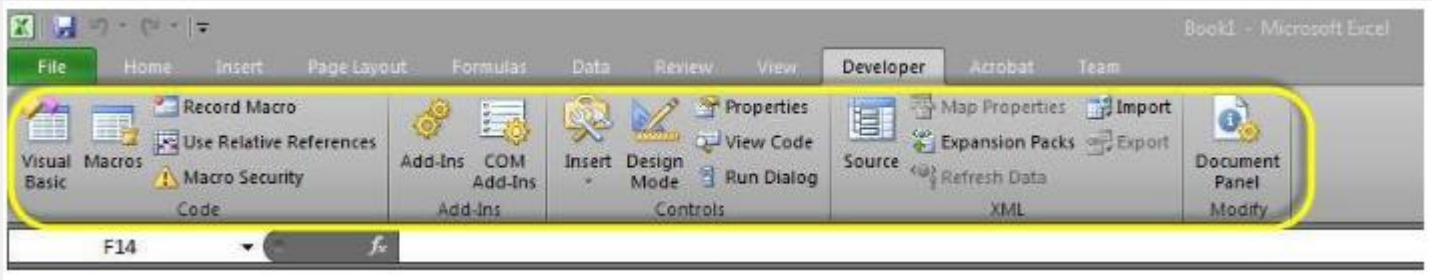
Visual Basic for Applications

- Click pe meniul CUSTOMIZE RIBBON > TAB Developer din FILE > OPTIONS

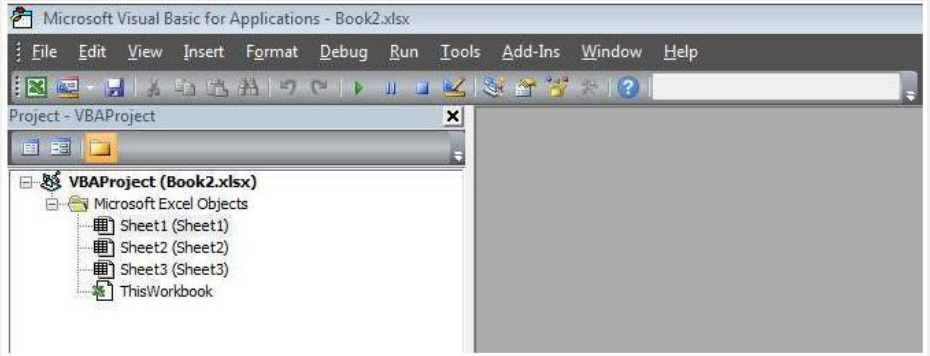


Visual Basic for Applications

- Tab-ul DEVELOPER apare în Ribbon



- Click pe VISUAL BASIC pentru a deschide editorul VBA



Visual Basic for Applications

- Folosirea celulelor, adreselor celulelor si a intervalelor de adrese
  - Range(Cell1, [Cell2])

Exemple
(se pot scrie in View > Immediate Window sau CTRL + G)

- **Range("B2").Select** - selecteaza celula B2
- **Range("B2:H2").Select** - selecteaza intervalul de celule B2:H2
- **Range("H14").Select** - selecteaza celula H14
- **Range(Selection, "B2").Select** – selecteaza intervalul de celule B2:H14 (H14 este selectia anterioara)
- **?Selection.Count** – calculeaza numarul de cellule din selectia anterioara

Visual Basic for Applications

- Folosirea celulelor, adreselor celulelor si a intervalelor de adrese
  - Cells

Exemple
(se pot scrie in View > Immediate Window sau CTRL + G)

- **Cells.Select** – selecteaza toate celulele
- **Cells.Item(5).Select** – selecteaza a 5a celula din prima linie
- **Cells.Item(16385).Select** – selecteaza prima celula de pe linia 2 – Excel permite existenta a 16384 (2^14) celule intr-o singura linie
- **Cells.Item(3,2).Select** – selecteaza celula de pe linia 3 coloana 2
- **Cells.Item(1048576,16384).Select** – selecteaza celula de pe ultima linie si ultima coloana – Excel permite existenta a 1048576 (2^20) cellule
- **Cells(1).Select ⇔ Cells.Item(1).Select ⇔ Range("A1").**

Visual Basic for Applications

- Folosirea celulelor, adreselor celulelor si a intervalelor de adrese
  - **Columns**

Exemple
(se pot scrie in View > Immediate Window sau CTRL + G)

- **Columns.Select**
- **Columns(3).Select**
- **Columns(“D”).Select**
- **Columns(“B:H”).Select**

Visual Basic for Applications

- Folosirea celulelor, adreselor celulelor si a intervalelor de adrese
  - **Rows**

Exemple
(se pot scrie in View > Immediate Window sau CTRL + G)

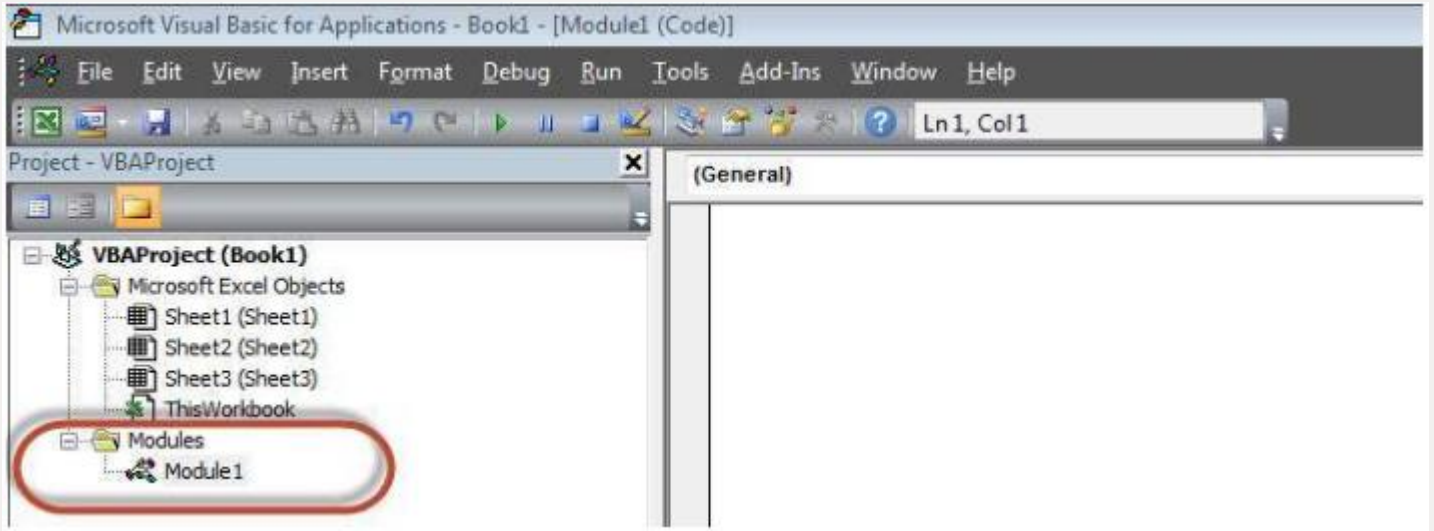
- **Rows.Select**
- **Rows(3).Select**
- **Rows(“3:14”).Select**

SUBPROGRAME

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications Terminologii

- MODULE:
  - Zona de cod
  - INSERT > MODULE
- În module se scrie cod VBA în cadrul subprogramelor
- SUBPROGRAM:
  - Serie de instrucțiuni VBA care sunt executate ca un întreg
  - Prin intermediul lor se specifică Excel-ului cum trebuie efectuate anumite sarcini
  - Există două tipuri de subprograme: FUNCTION și SUB PROCEDURES



Visual Basic for Applications Terminologii - FUNCTION

- Grup de cod reutilizabil care poate fi apelat în orice zona a programului
- Pot să returnez o valoare

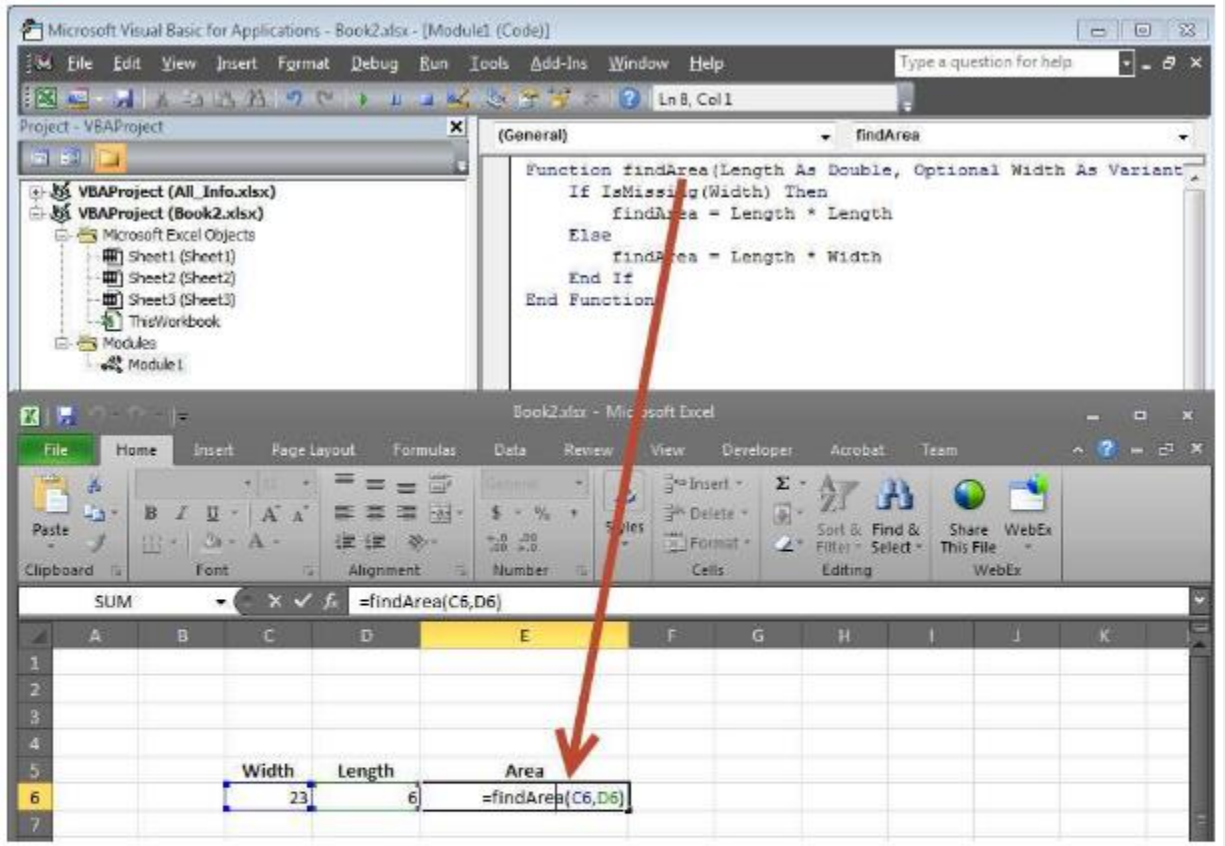
FUNCTION

...

END FUNCTION

Visual Basic for Applications Funcții

- Apelul unei funcții



The Output of the area would be displayed to the user.

Width	Length	Area
23	6	138

Visual Basic for Applications Terminologii – SUB PROCEDURES

- Sunt similare funcțiilor
- NU returnează valoare

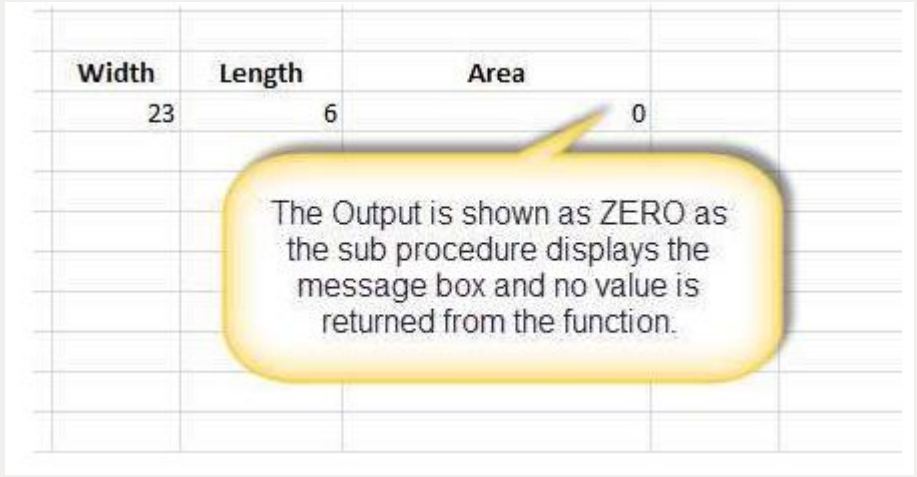
SUB

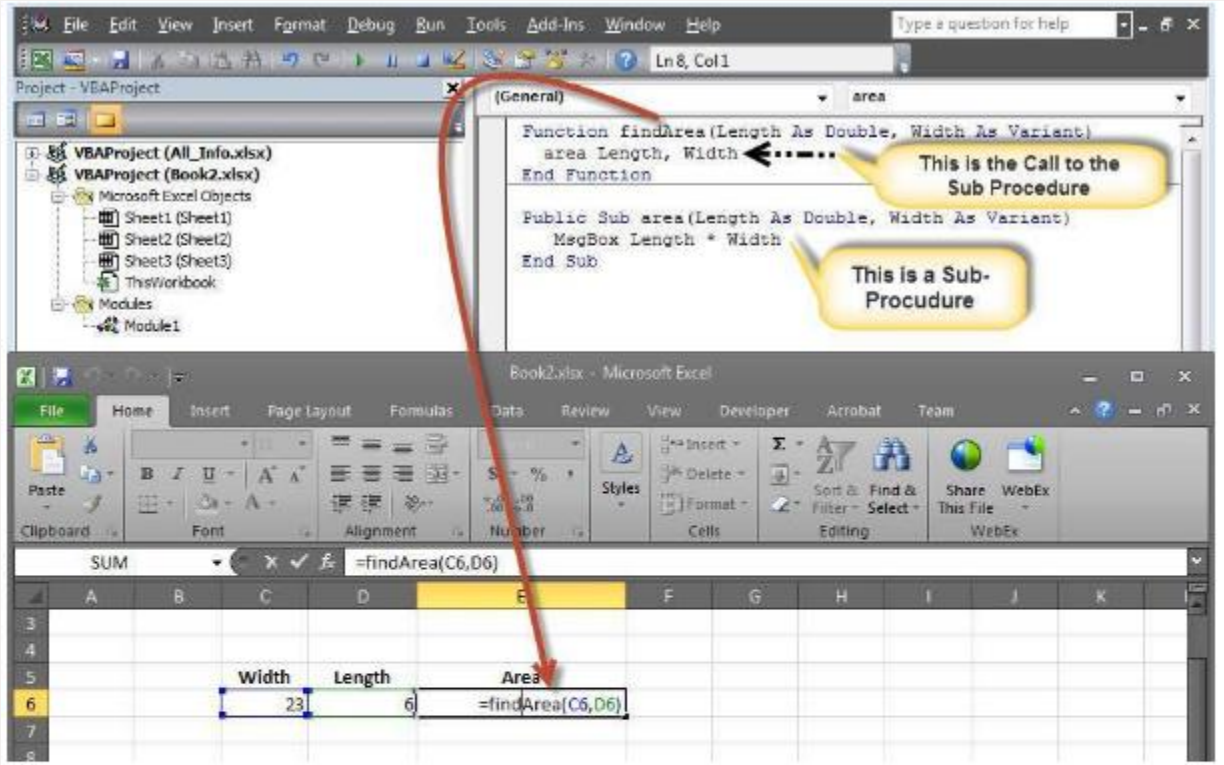
...

END SUB

Visual Basic for Applications Sub proceduri

- Apelul unei proceduri





2. The Area is calculated and shown only in Message box.

Visual Basic for Applications

```
Sub LabelMonths()
   Range("B1").Select
   ActiveCell.FormulaR1C1 = "January"
   Range("C1").Select
   ActiveCell.FormulaR1C1 = "February"
   Range("D1").Select
   ActiveCell.FormulaR1C1 = "March"
   Range("D2").Select
End Sub
```

---- sau ----

```
Sub LabelMonths()
   Range("B1").FormulaR1C1 = "January"
   Range("C1").FormulaR1C1 = "February"
   Range("D1").FormulaR1C1 = "March"
 End Sub
```

Visual Basic for Applications

```
Sub MakeBoldItalic()
    Range("B1").Select
    Selection.Font.Bold = True
    Selection.Font.Italic = True
End Sub
---- sau ----
Sub MakeBoldItalic()
    Dim myRange As Range
    Set myRange = Range("B1")
    myRange.Font.Bold = True
    myRange.Font.Italic = True
End Sub
---- sau ----
With Range("B1")
 .Font.Bold = True
 .Font.Italic = True
End With
```

Visual Basic for Applications Comentarii

- Orice linie care începe cu apostrof (‘) este tratată ca un comentariu

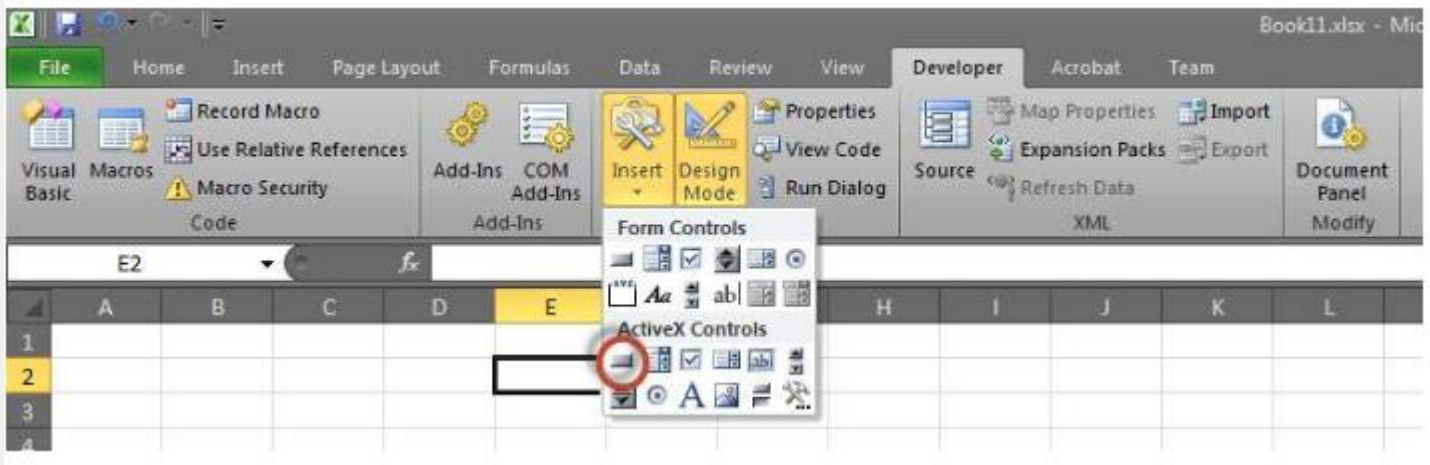
```
' This Script is invoked after successful login
```

- Orice linie care începe cu REM este tratată ca un comentariu

```
REM This Script is written to Validate the Entered Input
```

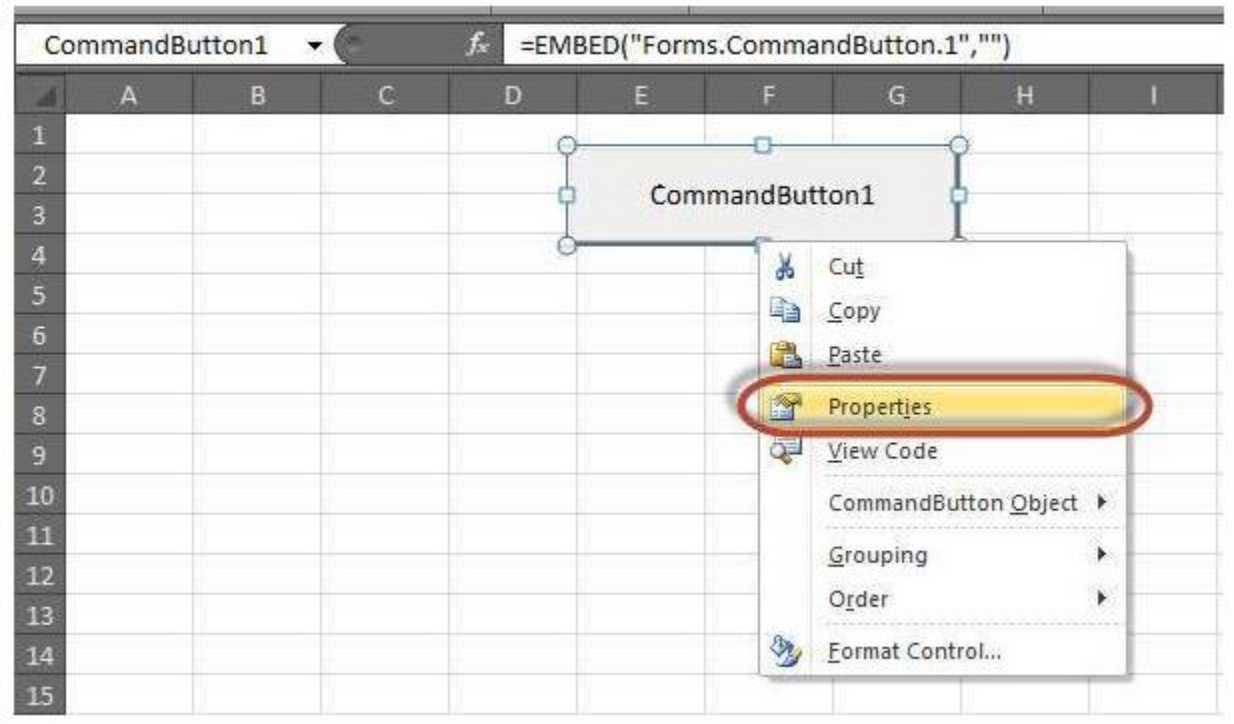
Visual Basic for Applications Adăugarea unui buton

- INSERT > BUTTON



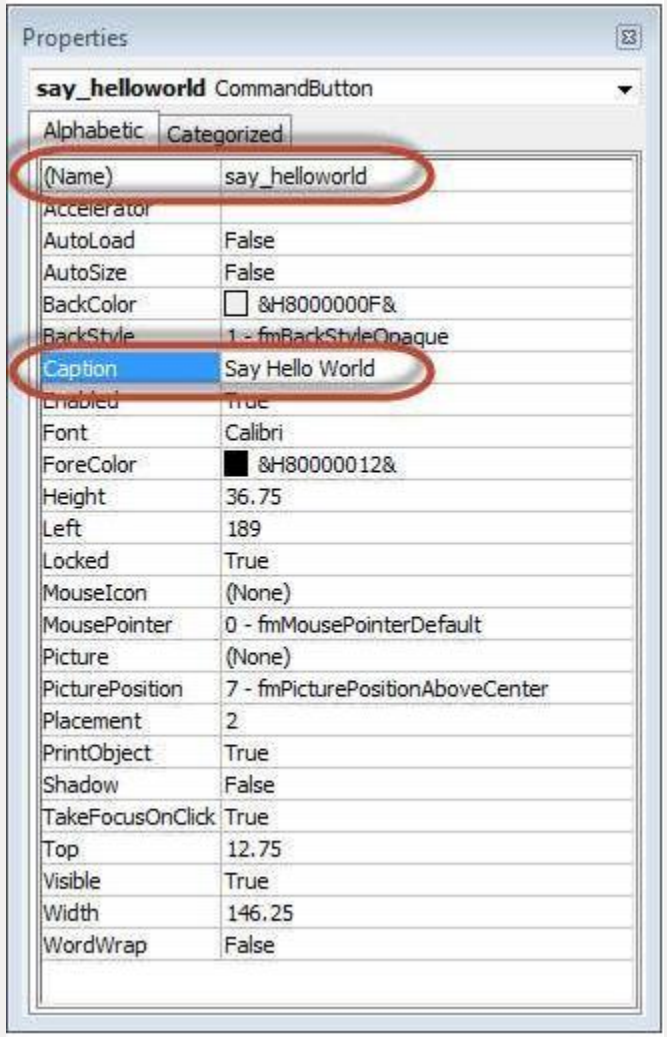
Visual Basic for Applications Adăugarea unui buton

- Click de dreapta pe PROPERTIES



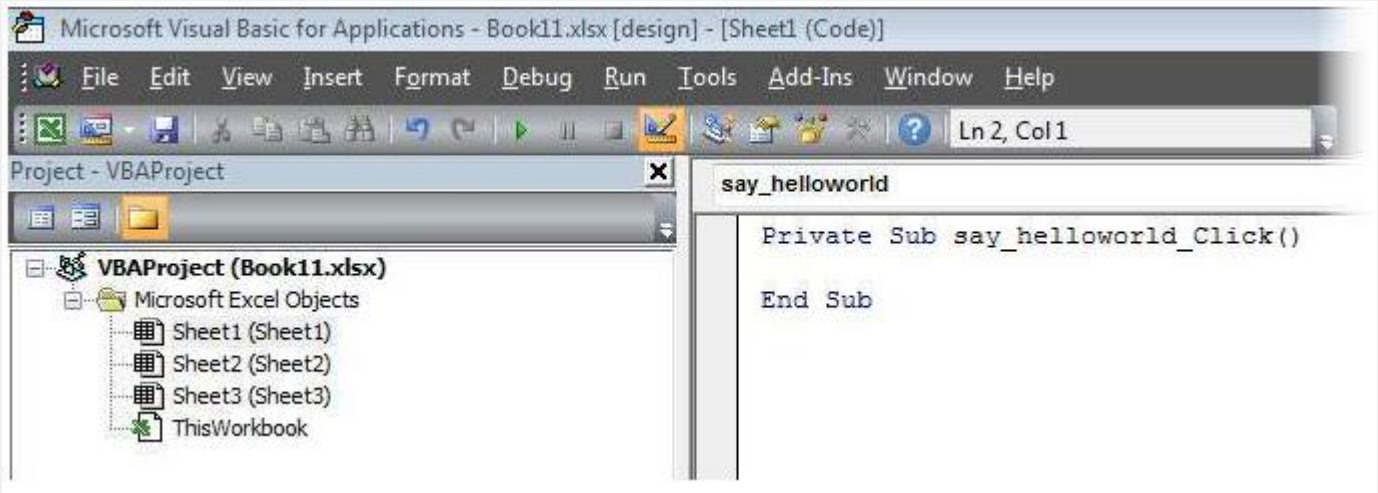
Visual Basic for Applications Adăugarea unui buton

- Editați CAPTION, NAME, ...



Visual Basic for Applications Adăugarea unui buton

- Dublu click pe buton => se deschide zona de editare de cod pentru a scrie procedura care se va executa la click



Visual Basic for Applications Adăugarea unui buton

- Exemplu: Adăugarea unui mesaj la click pe buton

```
Private Sub say_helloworld_Click()
    MsgBox "Hi"
End Sub
```

- Click pe buton

MESSAGEBOX

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications Message Box

- Funcția **MsgBox** afișează într-o căsuță de dialog un mesaj și așteaptă ca utilizatorul să dea click pe un buton pentru a acționa conform acțiunii utilizatorului
- SINTAXĂ

```
MsgBox(prompt[,buttons][,title])
```

  - **Prompt (*)**: șirul de caractere care se afișează în căsuța de dialog (maxim 1024 caractere);
  - **Buttons**: expresie numerică ce specifică tipul butoanelor afișate, a stilului utilizat, identitatea butoanelor și tipul de căsuță de dialog;
  - **Title**: șirul de caractere specificat în bara de titlu a căsuței de dialog;

Visual Basic for Applications
*Message Box - **Buttons***

BUTOANE

- 0 - vbOKOnly
- 1 - vbOKCancel
- 2 - vbAbortRetryIgnore
- 3 - vbYesNoCancel
- 4 - vbYesNo
- 5 - vbRetryCancel

STIL

- 16 - vbCritical
- 32 - vbQuestion
- 48 - vbExclamation

CARE DIN BUTOANE E DEFAULT

- 0 - vbDefaultButton1
- 256 - vbDefaultButton2
- 512 - vbDefaultButton3
- 768 - vbDefaultButton4

TIPUL DE MESSAGE BOX

- 0 - vbApplicationModal – aplicația curentă nu continuă până când utilizatorul răspunde la mesajul din căsuța de dialog
- 4096 - vbSystemModal – toate aplicațiile nu vor continua până când utilizatorul răspunde la mesajul din căsuța de dialog

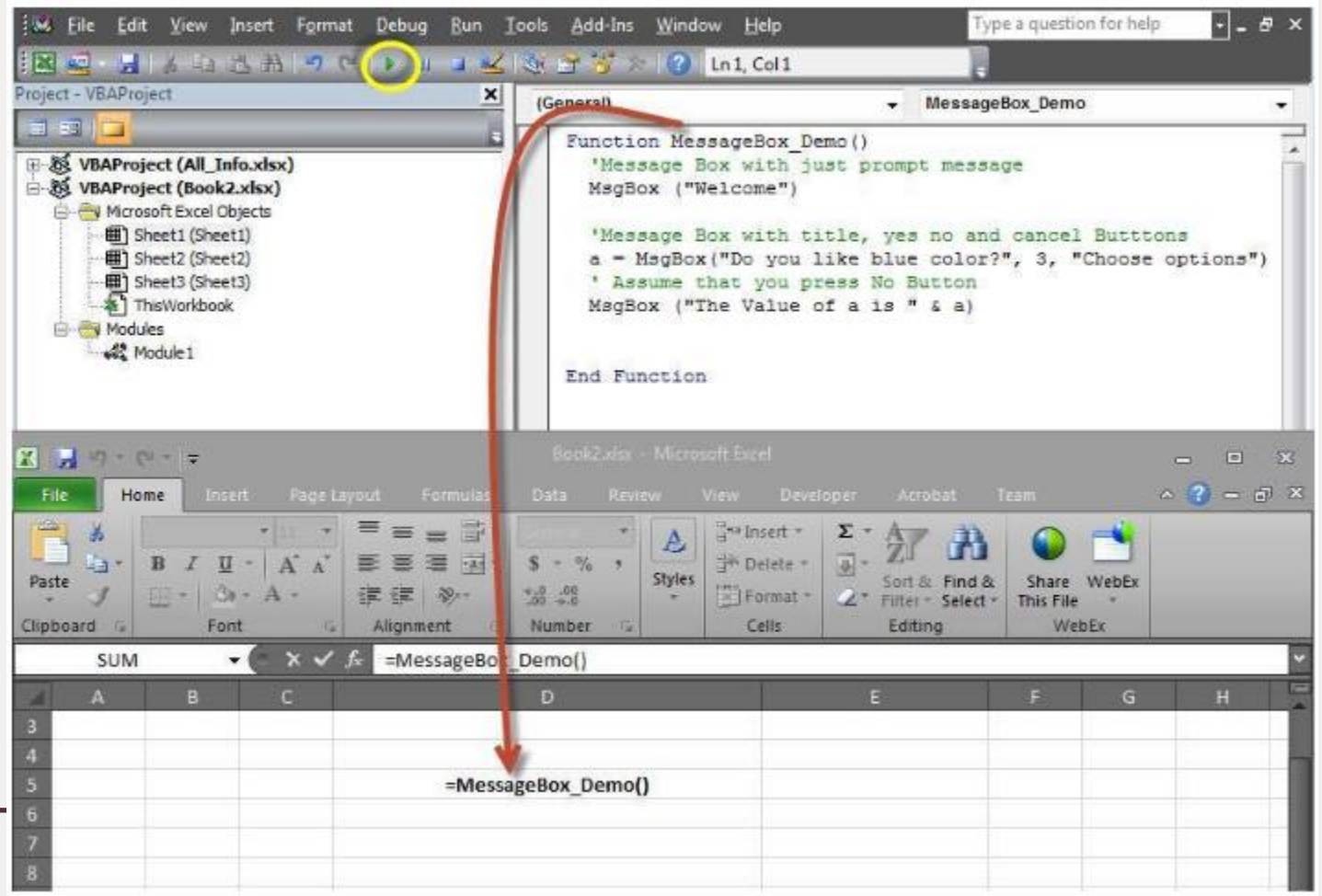
Visual Basic for Applications Message Box – ***Valori returnate***

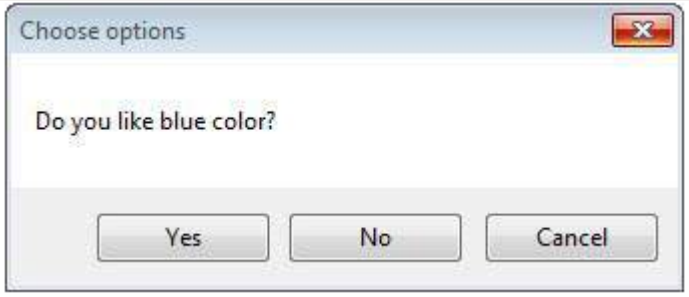
- 1 – vbOK
- 2 – vbCancel
- 3 – vbAbort
- 4 – vbIgnore
- 6 – vbYes
- 7 – vbNo

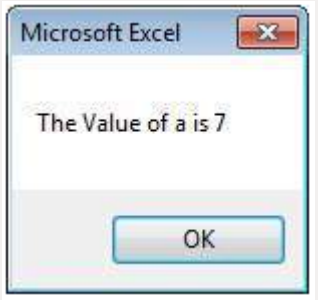
Visual Basic for Applications Exemplu

```
Function MessageBox_Demo()
'Message Box with just prompt message
MsgBox("Welcome")

'Message Box with title, yes no and cancel
a = MsgBox("Do you like blue color?",3,"Choose options")

'Assume that you press No
MsgBox ("The Value of a is " & a)
End Function
```

Visual Basic for Applications







INPUTBOX

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications
Input Box

- Funcția **InputBox** se folosește atunci când se dorește utilizarea datelor de intrare furnizate de utilizatori
- SINTAXĂ

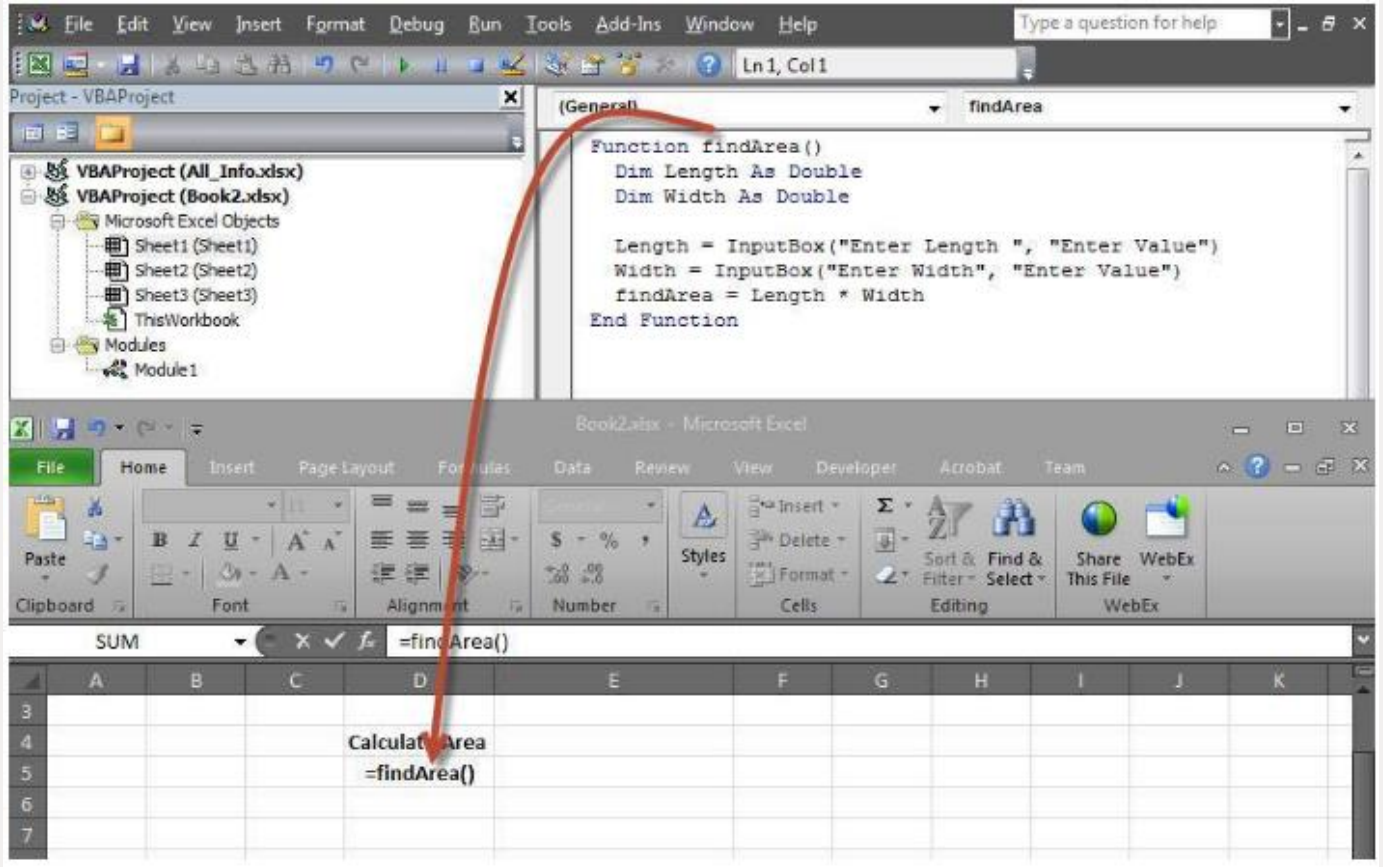
```
InputBox(prompt[,title][,default][,xpos][,ypos])
```

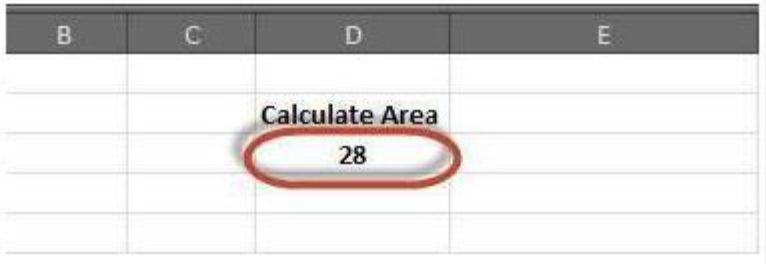
  - **Prompt (*)**: șirul de caractere care se afișează în căsuța de dialog (maxim 1024 caractere);
  - **Title**: șirul de caractere specificat în bara de titlu a căsuței de dialog;
  - **Default**: textul implicit din textbox pe care utilizatorul dorește să îl seteze implicit;
  - **xPos**: poziția axei OX ce reprezintă distanța prompt-ului față de partea stângă a ecranului; dacă nu se completează input box-ul se va centra orizontal;
  - **yPos**: poziția axei OX ce reprezintă distanța prompt-ului față de partea de sus a ecranului; dacă nu se completează input box-ul se va centra orizontal;

Visual Basic for Applications Exemplu

```
Function findArea()
  Dim Length As Double
  Dim Width As Double

  Length = InputBox("Enter Length ", "Enter a Number")
  Width = InputBox("Enter Width", "Enter a Number")
  findArea = Length * Width
End Function
```

Visual Basic for Applications
Rulare Exemplu





VARIABILE

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications
Variabile

- Primul caracter: literă
- Caractere care nu sunt permise: . (punct), (spatiu), !, @, &, $, #
- Lungime maximă: 255 caractere
- SINTAXĂ

```
Dim <<variable_name>> As <<variable_type>>
```

- **Data types**:
  - Byte, Integer, Long, Single, Double, Currency, Decimal
  - String, Date, Boolean, Object, Variant (orice tip de date nedeclarat explicit)

Visual Basic for Applications
Exemplu

```
Private Sub Variables_demo_Click()
 Dim password As String
 password = "Admin#1"

 Dim num As Integer
 num = 1234

 Dim BirthDay As Date
 BirthDay = DateValue("Jun 19, 2010")

 MsgBox "Password is " & password & Chr(10) & "Value of num is " & num & Chr(10)
& "Value of Birthday is " & BirthDay
End Sub
```

CONSTANTE

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications
Constante

- Primul caracter: literă
- Caractere care nu sunt permise: . (punct), (spatiu), !, @, &, $, #
- Lungime maximă: 255 caractere
- SINTAXĂ

```
Const <<constant_name>> As <<constant_type>> = <<constant_value>>
```

- Dacă se schimba valoarea constantei se va returna eroare

Visual Basic for Applications Exemplu


```
Private Sub Constant_demo_Click()
  Const MyInteger As Integer = 42
  Const myDate As Date = #2/2/2020#
  Const myDay As String = "Sunday"

  MsgBox "Integer is " & MyInteger & Chr(10) & "myDate is " & myDate & Chr(10) &
"myDay is " & myDay
End Sub
```

OPERATORI

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications Operatori

```
4 + 5
Operanzi: 4, 5
Operator: +
```

- Aritmetici
  - +, -, *, /, %, ^
- De comparare
  - ==, <>, >, <, >=, <=
- Logici (Relaționali)
  - AND, OR, NOT, XOR
- De concatenare
  - +, &

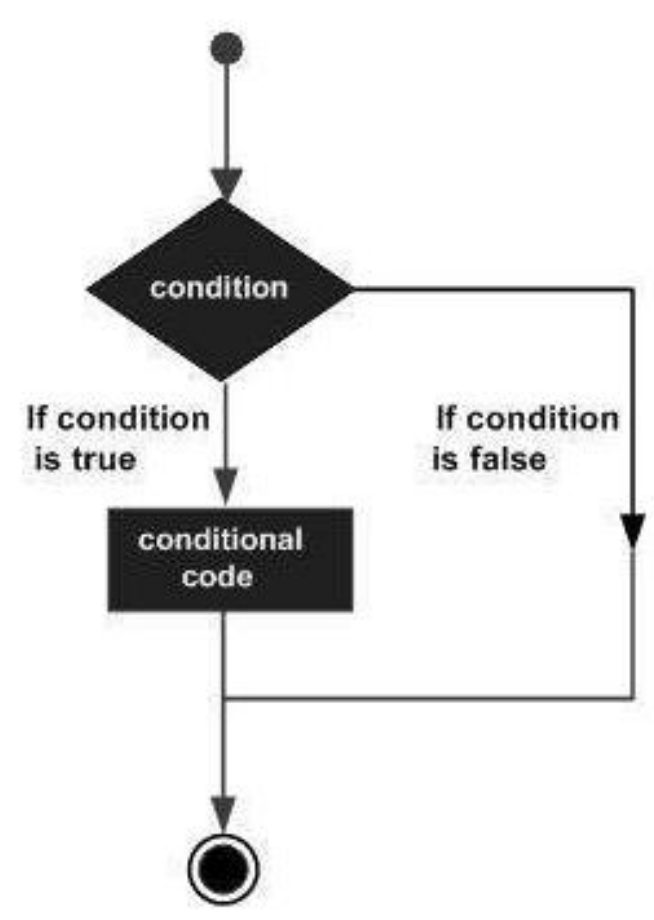
EXEMPLE:

- Dacă A=5 și B=10
  - *A+B=15*
  - *A&B=510*
- Dacă A=“abc” și B=“def”
  - *A+B=“abcdef”*
  - *A&B=“abcdef”*

IF

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications
IF




```
If(boolean_expression) Then
   Statement 1
       .....
       .....
   Statement n
End If
```

```
Private Sub if_demo_Click()
    Dim x As Integer
    Dim y As Integer
    x = 234
    y = 32
    If x > y Then
       MsgBox "X is Greater than Y"
    End If
End Sub
```

Visual Basic for Applications

```
If(boolean_expression) Then
   Statement 1
       .....
       .....
   Statement n
Else
   Statement 1
       .....
       ....
   Statement n
End If
```

```
Private Sub if_demo_Click()
    Dim x As Integer
    Dim y As Integer
    x = 234
    y = 324
    If x > y Then
       MsgBox "X is Greater than Y"
       Else
          Msgbox "Y is Greater than X"
    End If
End Sub
```

Visual Basic for Applications

```
If(boolean_expression) Then
   Statement 1
       .....
   Statement n
ElseIf (boolean_expression) Then
   Statement 1
       .....
   Statement n
ElseIf (boolean_expression) Then
   Statement 1
       .....
   Statement n
Else
   Statement 1
       .....
   Statement n
End If
```

```
Private Sub if_demo_Click()
    Dim x As Integer
    Dim y As Integer
    x = 234
    y = 234
    If x > y Then
       MsgBox "X is Greater than Y"
       ElseIf y > x Then
          Msgbox "Y is Greater than X"
       Else
          Msgbox "X and Y are EQUAL"
    End If
End Sub
```

Visual Basic for Applications Nested IFs

```
If(boolean_expression) Then
   Statement 1
       .....
   Statement n
   If(boolean_expression) Then
      Statement 1
             .....
      Statement n
   ElseIf (boolean_expression) Then
      Statement 1
             ....
      Statement n
   Else
      Statement 1
             ....
      Statement n
   End If
Else
   Statement 1
       ....
   Statement n
End If
```

Visual Basic for Applications Nested IFs

```
Private Sub nested_if_demo_Click()
  Dim a As Integer
  a = 23
  If a > 0 Then
     MsgBox "The Number is a POSITIVE Number"
     If a = 1 Then
        MsgBox "The Number is Neither Prime NOR Composite"
     ElseIf a = 2 Then
        MsgBox "The Number is the Only Even Prime Number"
     ElseIf a = 3 Then
        MsgBox "The Number is the Least Odd Prime Number"
     Else
        MsgBox "The Number is NOT 0,1,2 or 3"
     End If
  ElseIf a < 0 Then
     MsgBox "The Number is a NEGATIVE Number"
  Else
     MsgBox "The Number is ZERO"
  End If
End Sub
```

SWITCH

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications
SWITCH

```
Select Case expression
   Case expressionlist1
      statement1
      ....
      statement1n
   Case expressionlist2
      statement1
      ....
   Case expressionlistn
      statement1
      ....
  Case Else
      elsestatement1
      ....
End Select
```

```
Private Sub switch_demo_Click()
  Dim MyVar As Integer
  MyVar = 1
  Select Case MyVar
     Case 1
       MsgBox "The Number is the Least Composite Number"
     Case 2
       MsgBox "The Number is the only Even Prime Number"
     Case 3
       MsgBox "The Number is the Least Odd Prime Number"
     Case Else
       MsgBox "Unknown Number"
  End Select
End Sub
```

FOR

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications
FOR

```
For counter = start To end [Step stepcount]
  [statement 1]
  [statement 2]
  ....
  [statement n]
  [Exit For]
  [statement 11]
  [statement 22]
  ....
  [statement n]
Next
```

```
Private Sub Constant_demo_Click()
   Dim a As Integer
   a = 10
   For i = 0 To a Step 2
     MsgBox "The value is i is : " & i
   Next
End Sub
```

Visual Basic for Applications
FOR EACH

```
For Each element In Group
   [statement 1]
   [statement 2]
   ....
   [statement n]
   [Exit For]
   [statement 11]
   [statement 22]
Next
```

```
Private Sub Constant_demo_Click()
   'fruits is an array
   fruits = Array("apple", "orange", "cherries")
   Dim fruitnames As Variant


   'iterating using For each loop.
   For Each Item In fruits
      fruitnames = fruitnames & Item & Chr(10)
   Next

   MsgBox fruitnames
End Sub
```

VBA Exemplu

```
Sub ProtectSheets()
     Dim mySheet As Worksheet
     For Each mySheet In Worksheets
     mySheet.Select
     mySheet.Protect "Password", True, True, True
     Next mySheet
End Sub

Sub UnprotectSheets()
     Dim mySheet As Worksheet
     For Each mySheet In Worksheets
     mySheet.Select ' This statement is optional.
     mySheet.Unprotect "Password"
     Next mySheet
End Sub
```

VBA Exemplu

```
Sub CompareCells()
    Dim i As Integer
    Calculate
    For i = 1 To Range("E3:G6").Cells.Count
        If Range("E3:G6").Cells(i) > Range("A3:C6").Cells(i) Then
            Range("E3:G6").Cells(i).Interior.Color = rgbLightGreen
        Else
            Range("A3:C6").Cells(i).Interior.Color = rgbLightSteelBlue
        End If
    Next i
End Sub
```

	A	B	C	D	E	F	G
1							
2	Old				New		
3	143	116	110		146	110	106
4	133	136	114		146	116	137
5	123	113	120		123	143	119
6	103	148	129		112	134	108

WHILE

*(**V**isual **B**asic for **A**pplications)*

```
While condition(s)
   [statements 1]
   [statements 2]
   ...
   [statements n]
Wend
```

Visual Basic for Applications
WHILE

```
Private Sub Constant_demo_Click()
  Dim Counter :  Counter = 10
  While Counter < 15          ' Test value of Counter.
    Counter = Counter + 1     ' Increment Counter.
    msgbox "The Current Value of the Counter is : " & Counter
  Wend    ' While loop exits if Counter Value becomes 15.
End Sub
```

DO WHILE

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications
DO - WHILE

```
Do While condition
   [statement 1]
   [statement 2]
   ...
   [statement n]
   [Exit Do]
   [statement 1]
   [statement 2]
   ...
   [statement n]
Loop
```

```
Private Sub Constant_demo_Click()
  Do While i < 5
    i = i + 1
    msgbox "The value of i is : " & i
  Loop
End Sub
```

Visual Basic for Applications
DO – WHILE – Sintaxă alternativă

```
Do
    [statement 1]
    [statement 2]
    ...
    [statement n]
    [Exit Do]
    [statement 1]
    [statement 2]
    ...
    [statement n]
Loop While condition
```

```
Private Sub Constant_demo_Click()
 i = 10
 Do
  i = i + 1
  MsgBox "The value of i is : " & i
 Loop While i < 3
 'Condition is false.Hence loop is executed once.
End Sub
```

DO UNTIL

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications
DO - UNTIL

```
Do Until condition
    [statement 1]
    [statement 2]
    ...
    [statement n]
    [Exit Do]
    [statement 1]
    [statement 2]
    ...
    [statement n]
Loop
```

```
Private Sub Constant_demo_Click()
  i=10
  Do Until i>15  'Condition is False.Hence loop will be
executed
     i = i + 1
     msgbox ("The value of i is : " & i)
  Loop
End Sub
```

Visual Basic for Applications
DO – UNTIL - Sintaxă alternativă

```
Do
    [statement 1]
    [statement 2]
    ...
    [statement n]
    [Exit Do]
    [statement 1]
    [statement 2]
    ...
    [statement n]
Loop Until condition
```

```
Private Sub Constant_demo_Click()
  i=10
  Do
    i = i + 1
    msgbox "The value of i is : " & i
  Loop Until i<15 'Condition is True.Hence loop is
executed once.
End Sub
```

VBA Exemplu

```
Sub ListFiles()
    Dim myRow As Integer
    Dim myFile As String

    Cells.Clear
    myRow = 1
    myFile = Dir("*.xls*")
    Do Until myFile = ""
        Cells(myRow, 1) = myFile
        myRow = myRow + 1
        myFile = Dir
    Loop
End Sub
```

EXIT

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications
EXIT FOR

Exit For

```
Private Sub Constant_demo_Click()
  Dim a As Integer
  a = 10
  For i = 0 To a Step 2 'i is the counter variable and it is incremented by 2
    MsgBox ("The value is i is : " & i)
    If i = 4 Then
      i = i * 10 'This is executed only if i=4
      MsgBox ("The value is i is : " & i)
      Exit For 'Exited when i=4
    End If
  Next
End Sub
```

Visual Basic for Applications
EXIT DO

```
Exit Do
```

```
Private Sub Constant_demo_Click()

i = 0
Do While i <= 100
  If i > 10 Then
    Exit Do   ' Loop Exits if i>10
  End If
  MsgBox ("The Value of i is : " & i)
  i = i + 2
Loop

End Sub
```

SIRURI DE CARACTERE

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications
Șiruri de caractere

```
variablename = "string"
```

- str1 = "string" ' Only Alphabets
- str2 = "132.45" ' Only Numbers
- str3 = "!@#$;*" ' Only Special Characters
- str4 = "Asc23@#" ' Has all the above

Visual Basic for Applications
Șiruri de caractere – InStr:
returnează prima apariție a unui string în alt string

```
InStr([start,]string1,string2[,compare])
```

- **Start:** poziția inițială de căutare
- **String1 (*)**: string-ul în care se caută
- **String2 (*)**: string-ul care se caută
- **Compare**: tipul de comparare utilizat (binar sau text)

Visual Basic for Applications
Șiruri de caractere – InStr:
returnează prima apariție a unui string în alt string

```
InStr([start,]string1,string2[,compare])
```

```
Private Sub Constant_demo_Click()
  Dim Var As Variant
  Var = "Microsoft VBScript"
  MsgBox ("Line 1 : " & InStr(1, Var, "s"))
  MsgBox ("Line 2 : " & InStr(7, Var, "s"))
  MsgBox ("Line 3 : " & InStr(1, Var, "f", 1))
  MsgBox ("Line 4 : " & InStr(1, Var, "t", 0))
  MsgBox ("Line 5 : " & InStr(1, Var, "i"))
  MsgBox ("Line 6 : " & InStr(7, Var, "i"))
  MsgBox ("Line 7 : " & InStr(Var, "VB"))
End Sub
```

```
Line 1 : 6
Line 2 : 0
Line 3 : 8
Line 4 : 9
Line 5 : 2
Line 6 : 16
Line 7 : 11
```

Visual Basic for Applications
Șiruri de caractere – InStrRev:
returnează prima apariție a unui string în alt string de la dreapta la stânga

```
InStrRev(string1,string2[,start,[compare]])
```

- **String1 (*)**: string-ul care se caută
- **String2 (*)**: string-ul în care se caută
- **Start:** poziția inițială de căutare
- **Compare**: tipul de comparare utilizat (binar sau text)

Visual Basic for Applications
Șiruri de caractere – InStrRev:

returnează prima apariție a unui string în alt string de la dreapta la stânga

InStrRev(string1,string2[,start,[compare]])

```
Private Sub Constant_demo_Click()
  var="Microsoft VBScript"
  msgbox("Line 1 : " & InStrRev(var,"s",10))
  msgbox("Line 2 : " & InStrRev(var,"s",7))
  msgbox("Line 3 : " & InStrRev(var,"f",-1,1))
  msgbox("Line 4 : " & InStrRev(var,"t",5))
  msgbox("Line 5 : " & InStrRev(var,"i",7))
  msgbox("Line 6 : " & InStrRev(var,"i",7))
  msgbox("Line 7 : " & InStrRev(var,"VB",1))
End Sub
```

```
Line 1 : 6
Line 2 : 6
Line 3 : 8
Line 4 : 0
Line 5 : 2
Line 6 : 2
Line 7 : 0
```

Visual Basic for Applications
Șiruri de caractere – LCase:

returnează șirul de caractere transformat în litere mici

LCase(String)

```
Private Sub Constant_demo_Click()
  var="Microsoft VBScript"
  msgbox("Line 1 : " & LCase(var))
  var="MS VBSCRIPT"
  msgbox("Line 2 : " & LCase(var))
  var="microsoft"
  msgbox("Line 3 : " & LCase(var))
End Sub
```

```
Line 1 : microsoft vbscript

Line 2 : ms vbscript

Line 3 : Microsoft
```

Visual Basic for Applications
Șiruri de caractere – UCase:

returnează șirul de caractere transformat în litere mari

UCase(String)

```
Private Sub Constant_demo_Click()
  var="Microsoft VBScript"
  msgbox("Line 1 : " & UCase(var))
  var="MS VBSCRIPT"
  msgbox("Line 2 : " & UCase(var))
  var="microsoft"
  msgbox("Line 3 : " & UCase(var))
End Sub
```

```
Line 1 : MICROSOFT VBSCRIPT

Line 2 : MS VBSCRIPT

Line 3 : MICROSOFT
```

Visual Basic for Applications
Șiruri de caractere – Left:

returnează un număr specificat de caractere (de la stânga spre dreapta)

Left(String, Length)

```
Private Sub Constant_demo_Click()
  Dim var as Variant
  var="Microsoft VBScript"
  msgbox("Line 1 : " & Left(var,2))
  var="MS VBSCRIPT"
  msgbox("Line 2 : " & Left(var,5))
  var="microsoft"
  msgbox("Line 3 : " & Left(var,9))
End Sub
```

```
Line 1 : Mi

Line 2 : MS VB

Line 3 : Microsoft
```

Visual Basic for Applications
Șiruri de caractere – Right:

returnează un număr specificat de caractere (de la dreapta spre stânga)

Right(String, Length)

```
Private Sub Constant_demo_Click()
  var="Microsoft VBScript"
  msgbox("Line 1 : " & Right(var,2))
  var="MS VBSCRIPT"
  msgbox("Line 2 : " & Right(var,5))
  var="microsoft"
  msgbox("Line 3 : " & Right(var,9))
End Sub
```

```
Line 1 : pt

Line 2 : CRIPT

Line 3 : Microsoft
```

Visual Basic for Applications
Șiruri de caractere – Mid:
returnează un număr specificat de caractere

Mid(String,start[,Length])

```
Private Sub Constant_demo_Click()
  Dim var as Variant
  var="Microsoft VBScript"
  msgbox("Line 1 : " & Mid(var,2))
  msgbox("Line 2 : " & Mid(var,2,5))
  msgbox("Line 3 : " & Mid(var,5,7))
End Sub
```

```
Line 1 : icrosoft VBScript
Line 2 : icros
Line 3 : osoft V
```

Visual Basic for Applications
Șiruri de caractere – LTrim:
șterge spațiile din stânga șirului de caractere

LTrim(String)	

Private Sub Constant_demo_Click()
  Dim var as Variant
  var =" Microsoft VBScript"
  msgbox "After Ltrim : " & LTrim(var)
End Sub
	After Ltrim : Microsoft VBScript

Visual Basic for Applications
Șiruri de caractere – Trim:
șterge spațiile din stânga și dreapta șirului de caractere

Trim(String)	

Private Sub Constant_demo_Click()
  var="Microsoft VBScript"
var =" Microsoft VBScript "
  msgbox ("After Trim : " & Trim(var))
End Sub
	After trim : Microsoft VBScript

Visual Basic for Applications
Șiruri de caractere – Len:
returnează lungimea șirului de caractere

Len(String)

```
Private Sub Constant_demo_Click()
  Dim var1 as Variant
  Dim var2 as Variant
  var1 ="Microsoft VBScript"
  msgbox("Length of var1 is : " &
Len(var1))
  var2 ="    Microsoft VBScript        "
  msgbox ("Length of var2 is : " &
Len(var2))
End Sub
```

```
Length of var1 is : 18

Length of var2 is : 30
```

Visual Basic for Applications
Șiruri de caractere – Replace:

înlocuirea unui șir de caractere cu un alt șir de caractere

Replace(string,find,replacewith[,start[,count[,compare]]])

```
Private Sub Constant_demo_Click()
  Dim var as Variant
  var="This is VBScript Programming"
  msgbox("Line 1: " & Replace(var,"VBScript","MS
VBScript"))
  msgbox("Line 2: " & Replace(var,"VB","vb"))
  msgbox("Line 3: " & Replace(var,"is","##"))
  msgbox("Line 4: " & Replace(var,"is","##",5))
  msgbox("Line 5: " & Replace(var,"s","##",1,2))
  msgbox("Line 6: " & Replace(var,"r","##",1,-1,1))
  msgbox("Line 7: " & Replace(var,"t","##",1,-1,0))
End Sub
```

```
1: This is MS VBScript Programming

2: This is vbScript Programming
3: Th## ## VBScript Programming
4: ## VBScript Programming
5: Thi## i## VBScript Programming
6: This is VBSc##ipt P##og##amming
7: This is VBScrip## Programming
```

Visual Basic for Applications
Șiruri de caractere – Space:

concatenează șirului de caractere un anumit număr de spații

Space(number)	

`Private Sub Constant_demo_Click()`
`  Dim var1 as Variant`
`  var1="Microsoft"`
`  Dim var2 as Variant`
`  var2="VBScript"`
`  msgbox(var1 & Space(2)& var2)`
`End Sub`
	`Microsoft VBScript`

Visual Basic for Applications
Șiruri de caractere – StrComp:
compară două șiruri de caractere și returnează un înreg (-1, 0, 1)

StrComp(string1,string2[,compare])

```
Private Sub Constant_demo_Click()
  Dim var1 as Variant
  msgbox("Line 1 :" &
StrComp("Microsoft","Microsoft"))
  msgbox("Line 2 :"
&StrComp("Microsoft","MICROSOFT"))
  msgbox("Line 3 :"
&StrComp("Microsoft","MiCrOsOfT"))
  msgbox("Line 4 :"
&StrComp("Microsoft","MiCrOsOfT",1))
  msgbox("Line 5 :"
&StrComp("Microsoft","MiCrOsOfT",0))
End Sub
```

```
Line 1 :0

Line 2 :1

Line 3 :1

Line 4 :0

Line 5 :1
```

Visual Basic for Applications
Șiruri de caractere – String:

concatenează șirului de caractere un anumit caracter de un anumit număr de ori

String(number,character)

```
Private Sub Constant_demo_Click()
  msgbox("Line 1 :" & String(3,"$"))
  msgbox("Line 2 :" & String(4,"*"))
  msgbox("Line 3 :" & String(5,100))
  msgbox("Line 4 :" & String(6,"ABCDE"))
End Sub
```

```
Line 1 :$$$
Line 2 :****
Line 3 :ddddd
Line 4 :AAAAAA
```

Visual Basic for Applications
Șiruri de caractere – StrReverse:
inversează șirul de caractere

StrReverse(string)

```
Private Sub Constant_demo_Click()
  msgbox("Line 1 : " & StrReverse("VBSCRIPT"))
  msgbox("Line 2 : " & StrReverse("My First
VBScript"))
  msgbox("Line 3 : " & StrReverse("123.45"))
End Sub
```

```
Line 1 : TPIRCSBV
Line 2 : tpircSBV tsriF yM

Line 3 : 54.321
```

DATA

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications

Dată - Date

returnează data sistem

Date()	
Private Sub Constant_demo_Click() Dim a as Variant a = date() msgbox "The Value of a : " & a End Sub	The Value of a : 19/01/2017

Visual Basic for Applications
Dată - Cdate

convertește o expresie care conține o dată validă în tipul de date Data

Cdate()	
Private Sub Constant_demo_Click() Dim a as Variant Dim b as Variant a = cdate("Jan 01 2020") msgbox("The Value of a : " & a) b = cdate("31 Dec 2050") msgbox("The Value of b : " & b) End Sub	The Value of a : 1/01/2020 The Value of b : 31/12/2050

Visual Basic for Applications
Dată - DateAdd

returnează data după adăugarea unui interval de timp

- **Interval(*):**
  - d – ziua
  - m – luna
  - y – ziua din an
  - yyyy – an
  - w – zi din săptămâna
  - ww – săptămâna
  - q – trimestru
  - h – ora
  - n – minutul
  - s – secunda
- **Number(*)**
- **Date (*)**

```
DateAdd(interval,number,date)
```

Pentru a adauga un numar de zile la o data se pate folosi Day of Year ("y"), Day ("d"), or Weekday ("w").

Visual Basic for Applications
Dată - DateAdd

returnează data după adăugarea unui interval de timp

```
Private Sub Constant_demo_Click()
  ' Positive Interval
  date1="01-Jan-2013"
  msgbox("Line 1 : " &DateAdd("yyyy",1,date1))
  msgbox("Line 2 : " &DateAdd("q",1,date1))
  msgbox("Line 3 : " &DateAdd("m",1,date1))
  msgbox("Line 4 : " &DateAdd("y",1,date1))
  msgbox("Line 5 : " &DateAdd("d",1,date1))
  msgbox("Line 6 : " &DateAdd("w",1,date1))
  msgbox("Line 7 : " &DateAdd("ww",1,date1))
  msgbox("Line 8 : " &DateAdd("h",1,"01-Jan-2013 12:00:00"))
  msgbox("Line 9 : " &DateAdd("n",1,"01-Jan-2013 12:00:00"))
  msgbox("Line 10 : "&DateAdd("s",1,"01-Jan-2013 12:00:00"))
End Sub
```

```
Line 1 : 01/01/2014
Line 2 : 01/04/2013
Line 3 : 01/02/2013
Line 4 : 02/01/2013
Line 5 : 02/01/2013
Line 6 : 02/01/2013
Line 7 : 08/01/2013
Line 8 : 1/01/2013 1:00:00 PM
Line 9 : 1/01/2013 12:01:00 PM
Line 10 : 1/01/2013 12:00:01 PM
```

Visual Basic for Applications
Dată - DateAdd

returnează data după adăugarea unui interval de timp

```
Private Sub Constant_demo_Click()
  ' Negative Interval
  date1="01-Jan-2013"
  msgbox("Line 11 : " &DateAdd("yyyy",-1,date1))
  msgbox("Line 12 : " &DateAdd("q",-1,date1))
  msgbox("Line 13 : " &DateAdd("m",-1,date1))
  msgbox("Line 14 : " &DateAdd("y",-1,date1))
  msgbox("Line 15 : " &DateAdd("d",-1,date1))
  msgbox("Line 16 : " &DateAdd("w",-1,date1))
  msgbox("Line 17 : " &DateAdd("ww",-1,date1))
  msgbox("Line 18 : " &DateAdd("h",-1,"01-Jan-2013 12:00:00"))
  msgbox("Line 19 : " &DateAdd("n",-1,"01-Jan-2013 12:00:00"))
  msgbox("Line 20 : " &DateAdd("s",-1,"01-Jan-2013 12:00:00"))
End Sub
```

```
Line 11 : 01/01/2012
Line 12 : 01/10/2012
Line 13 : 01/12/2012
Line 14 : 31/12/2012
Line 15 : 31/12/2012
Line 16 : 31/12/2012
Line 17 : 25/12/2012
Line 18 : 1/01/2013 11:00:00 AM
Line 19 : 1/01/2013 11:59:00 AM
Line 20 : 1/01/2013 11:59:59 AM
```

Visual Basic for Applications
Dată - DateDiff
returnează diferența dintre două intervale de timp

```
DateDiff(interval, date1, date2 [,firstdayofweek[, firstweekofyear]])
```

- **Interval(*):**
  - d – ziua
  - m – luna
  - y – ziua din an
  - yyyy – an
  - w – ziua din săptămâna
  - ww – săptămâna
  - q – trimestru
  - h – ora
  - n – minutul
  - s – secunda
- **Date1(*)**
- **Date2 (*)**

- **firstdayofweek**
  - *0 – implicit, 1 –Duminică, ... , 7 – Sâmbătă*
- **firstdayofyear**
  - *0 – implicit, 1 – săptămâna în care e 1 ianuarie, 2 – săptămâna care are cel puțin 4 zile în noul an, 3 – săptămâna care are 7 zile în noul an*

Visual Basic for Applications

Dată – DateDiff - *returnează diferența dintre două intervale de timp*

```
Private Sub Constant_demo_Click()
  Dim fromDate as Variant
  fromDate="01-Jan-09 00:00:00"
  Dim toDate as Variant
  toDate="01-Jan-10 23:59:00"
  msgbox("Line 1 : " &DateDiff("yyyy",fromDate,toDate))
  msgbox("Line 2 : " &DateDiff("q",fromDate,toDate))
  msgbox("Line 3 : " &DateDiff("m",fromDate,toDate))
  msgbox("Line 4 : " &DateDiff("y",fromDate,toDate))
  msgbox("Line 5 : " &DateDiff("d",fromDate,toDate))
  msgbox("Line 6 : " &DateDiff("w",fromDate,toDate))
  msgbox("Line 7 : " &DateDiff("ww",fromDate,toDate))
  msgbox("Line 8 : " &DateDiff("h",fromDate,toDate))
  msgbox("Line 9 : " &DateDiff("n",fromDate,toDate))
  msgbox("Line 10 : "&DateDiff("s",fromDate,toDate))
End Sub
```

```
Line 1 : 1
Line 2 : 4
Line 3 : 12
Line 4 : 365
Line 5 : 365
Line 6 : 52
Line 7 : 52
Line 8 : 8783
Line 9 : 527039
Line 10 : 31622340
```

Visual Basic for Applications
Dată - DatePart
returnează o parte specifică a datei

DatePart(interval,date[,firstdayofweek[,firstweekofyear]])

```
Private Sub Constant_demo_Click()
  Dim Quarter as Variant
  Dim DayOfYear as Variant
  Dim WeekOfYear as Variant
  Date1 = "2013-01-15"
  Quarter    = DatePart("q", Date1)
  msgbox("Line 1 : " & Quarter)
  DayOfYear  = DatePart(„d", Date1)
  msgbox("Line 2 : " & DayOfYear)
  WeekOfYear = DatePart("ww", Date1)
  msgbox("Line 3 : " & WeekOfYear)
  msgbox("Line 4 : " &
DatePart("m",Date1))
End Sub
```

```
Line 1 : 1

Line 2 : 15

Line 3 : 3

Line 4 : 1
```

Visual Basic for Applications
Dată – DateSerial

returnează data din parametrii lună, zi, an

DateSerial(year,month,day)	
Private Sub Constant_demo_Click() msgbox(DateSerial(2013,5,10)) End Sub	10/05/2014

Visual Basic for Applications
Dată – FormatDateTime

formatează și returnează o dată validă

FormatDateTime(date,format)

```
Private Sub Constant_demo_Click()
        d=("2013-08-15 20:25")
 msgbox("Line 1 : " & FormatDateTime(d))
msgbox("Line 2 : " & FormatDateTime(d,1))
msgbox("Line 3 : " & FormatDateTime(d,2))
msgbox("Line 4 : " & FormatDateTime(d,3))
msgbox("Line 5 : " & FormatDateTime(d,4))
                End Sub
```

```
Line 1 : 15/08/2013 8:25:00 PM
Line 2 : Thursday, 15 August 2013
       Line 3 : 15/08/2013
       Line 4 : 8:25:00 PM
          Line 5 : 20:25
```

- Date(*)
- Format
  - 0 – implicit, 1 – long date, 2 – short date, 3 – long time, 4 – short time

Visual Basic for Applications
Dată – Day
returnează un număr între 1 și 31

Day(date)	
Private Sub Constant_demo_Click() msgbox(Day("2013-06-30")) End Sub	30

Visual Basic for Applications
Dată – Month
returnează un număr între 1 și 12

Month(date)	
Private Sub Constant_demo_Click() msgbox(Month("2013-06-30")) End Sub	6

Visual Basic for Applications
Dată – MonthName

returnează numele lunii

MonthName(month[,toabbreviate])

```
Private Sub Constant_demo_Click()
msgbox("Line 1 : " & MonthName(01,True))
msgbox("Line 2 : " & MonthName(01,false))
msgbox("Line 3 : " & MonthName(07,True))
msgbox("Line 4 : " & MonthName(07,false))
End Sub
```

```
Line 1 : Jan
Line 2 : January
Line 3 : Jul
Line 4 : July
```

Visual Basic for Applications
Dată – Year
returnează un număr întreg

Year(date)	
Private Sub Constant_demo_Click() msgbox(Year("2013-06-30")) End sub	2013

Visual Basic for Applications
Dată – WeekDay
returnează un număr între 1 și 7

Weekday(date[,firstdayofweek])

```
Private Sub Constant_demo_Click()
msgbox("Line 1: " & Weekday("2013-05-16",1))
msgbox("Line 2: " & Weekday("2013-05-16",2))
msgbox("Line 3: " & Weekday("2013-05-16",2))
 msgbox("Line 4: " & Weekday("2010-02-16"))
 msgbox("Line 5: " & Weekday("2010-02-17"))
 msgbox("Line 6: " & Weekday("2010-02-18"))
                   End Sub
```

```
Line 1: 5
Line 2: 4
Line 3: 4
Line 4: 3
Line 5: 4
Line 6: 5
```

Visual Basic for Applications
Dată – WeekdayName

returnează numele zilei din săptămână

WeekdayName(weekday[,abbreviate[,firstdayofweek]])

```
Private Sub Constant_demo_Click()
   msgbox("Line 1 : " &WeekdayName(3))
msgbox("Line 2 : " &WeekdayName(2,True))
msgbox("Line 3 : " &WeekdayName(1,False))
msgbox("Line 4 : " &WeekdayName(2,True,0))
msgbox("Line 5 : " &WeekdayName(1,False,1))
                   End Sub
```

```
Line 1 : Tuesday
Line 2 : Mon
Line 3 : Sunday
Line 4 : Tue
Line 5 : Sunday
```

TIMP

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications
Timp – Now

returnează ora și data sistemului

Now()	
Private Sub Constant_demo_Click() Dim a as Variant a = Now() msgbox("The Value of a : " & a) End Sub	The Value of a : 19/07/2013 3:04:09 PM

Visual Basic for Applications
Timp – Hour

returnează un număr între 0 și 23

Hour(time)

```
Private Sub Constant_demo_Click()
msgbox("Line 1: " & Hour("3:13:45 PM"))
msgbox("Line 2: " & Hour("23:13:45"))
msgbox("Line 3: " & Hour("2:20 PM"))
End Sub
```

```
Line 1: 15
Line 2: 23
Line 3: 14
```

Visual Basic for Applications
Timp – Minute

returnează un număr între 0 și 59

Minute(time)

```
Private Sub Constant_demo_Click()
msgbox("Line 1: " & Minute("3:13:45 PM"))
msgbox("Line 2: " & Minute("23:13:45"))
msgbox("Line 3: " & Minute("2:20 PM"))
End Sub
```

```
Line 1: 13
Line 2: 43
Line 3: 20
```

Visual Basic for Applications
Timp – Second
returnează un număr între 0 și 59

Second(time)

```
Private Sub Constant_demo_Click()
msgbox("Line 1: " & Second("3:13:45 PM"))
msgbox("Line 2: " & Second("23:13:45"))
msgbox("Line 3: " & Second("2:20 PM"))
End Sub
```

```
Line 1: 25
Line 2: 45
Line 3: 0
```

Visual Basic for Applications
Timp – Time
returnează ora sistem

Time()	
Private Sub Constant_demo_Click() msgbox("Line 1: " & Time()) End Sub	Line 1: 3:29:15 PM

Visual Basic for Applications
Timp – TimeSerial

returnează timpul din ora, minutul si secunda specificată

TimeSerial(hour,minute,second)

```
Private Sub Constant_demo_Click()
    msgbox(TimeSerial(20,1,2))
    msgbox(TimeSerial(0,59,59))
 msgbox(TimeSerial(7*2,60/3,15+3))
End Sub
```

```
8:01:02 PM
12:59:59 AM
2:20:18 PM
```

Visual Basic for Applications
Timp – TimeValue

convertește datele de intrare în tipul de dată Timp

TimeValue(StringTime)

```
Private Sub Constant_demo_Click()
    msgbox(TimeValue("20:30"))
    msgbox(TimeValue("5:15"))
    msgbox(TimeValue("2:30:58"))
End Sub
```

```
8:30:00 PM
5:15:00 AM
2:30:58 AM
```

ARRAY

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications
Array

```
'Method 1 : Using Dim
Dim arr1()    'Without Size

'Method 2 : Mentioning the Size
Dim arr2(5)   'Declared with size of 5

'Method 3 : using 'Array' Parameter
Dim arr3
arr3 = Array("apple","Orange","Grapes")
```

Visual Basic for Applications
Array

```
Private Sub Constant_demo_Click()
  Dim arr(5)
  arr(0) = "1"              'Number as String
  arr(1) = "VBScript"       'String
  arr(2) = 100              'Number
  arr(3) = 2.45             'Decimal Number
  arr(4) = #10/07/2013#     'Date
  arr(5) = #12.45 PM#       'Time

  msgbox("in Array index 0 : " & arr(0))
  msgbox("in Array index 1 : " & arr(1))
  msgbox("in Array index 2 : " & arr(2))
  msgbox("in Array index 3 : " & arr(3))
  msgbox("in Array index 4 : " & arr(4))
  msgbox("in Array index 5 : " & arr(5))
End Sub
```

```
in Array index 0 : 1
in Array index 1 : VBScript
in Array index 2 : 100
in Array index 3 : 2.45
in Array index 4 : 7/10/2013
in Array index 5 : 12:45:00 PM
```

Visual Basic for Applications
Array multidimensional

```
Private Sub Constant_demo_Click()
  Dim arr(2,2) as Variant
' Which has 3 rows and 3 columns
  arr(0,0) = "Apple"
  arr(0,1) = "Orange"
  arr(0,2) = "Grapes
  arr(1,0) = "cucumber"
  arr(1,1) = "beans"
  arr(1,2) = "carrot"
  arr(2,0) = "potato"
  arr(2,1) = "sandwitch"
  arr(2,2) = "coffee"
  msgbox("in Array index 0,1 : " &  arr(0,1))
  msgbox("in Array index 2,2 : " &  arr(2,2))

End Sub
```

```
in Array index : 0 , 1 : Orange
in Array index : 2 , 2 : coffee
```

Visual Basic for Applications
Array

ReDim [Preserve] varname(subscripts) [, varname(subscripts)]

```
Private Sub Constant_demo_Click()
        Dim a() as variant
              i=0
           redim a(5)
           a(0)="XYZ"
           a(1)=41.25
            a(2)=22
         REDIM PRESERVE a(7)
            For i=3 to 7
              a(i)= i
               Next
        'to Fetch the output
        For i=0 to ubound(a)
            Msgbox a(i)
              Next
            End Sub
```

```
XYZ
41.25
22
3
4
5
6
7
```

Visual Basic for Applications
Array - LBound
indicele minim al șirului

LBound(ArrayName[,dimension])

```
Private Sub Constant_demo_Click()
  Dim arr(5) as Variant
  arr(0) = "1"            'Number as String
  arr(1) = "VBScript      'String
  arr(2) = 100            'Number
  arr(3) = 2.45           'Decimal Number
  arr(4) = #10/07/2013#   'Date
  arr(5) = #12.45 PM#     'Time
  msgbox(„arr is : " & LBound(arr))
  ' For MultiDimension Arrays :
  Dim arr2(3,2) as Variant
  msgbox(„arr2-1 is : " & LBound(arr2,1))
  msgbox("arr2-2 is : " & LBound(arr2,2))
End Sub
```

```
Arr is : 0


Arr2-1 is : 0
Arr2-2 is : 0
```

Visual Basic for Applications
Array - UBound
indicele maxim al șirului

UBound(ArrayName[,dimension])

```
Private Sub Constant_demo_Click()
  Dim arr(5) as Variant
  arr(0) = "1"            'Number as String
  arr(1) = "VBScript      'String
  arr(2) = 100            'Number
  arr(3) = 2.45           'Decimal Number
  arr(4) = #10/07/2013#   'Date
  arr(5) = #12.45 PM#     'Time
  msgbox(„arr is : " & UBound(arr))
  ' For MultiDimension Arrays :
  Dim arr2(3,2) as Variant
  msgbox("arr2-1 is : " & UBound(arr2,1))
  msgbox("arr2-2 is : " & UBound(arr2,2))
End Sub
```

```
Arr is : 5


Arr2-1 is : 3
Arr2-2 is : 2
```

Visual Basic for Applications
Array - Split
returnează un șir

Split(expression[,delimiter[,count[,compare]]])

```
Private Sub Constant_demo_Click()
 ' Splitting based on delimiter comma '$'
 Dim a as Variant
 Dim b as Variant
 a=Split("Red $ Blue $ Yellow","$")
 b=ubound(a)
 For i=0 to b
    msgbox("The value of array in " & i & "
is :"  & a(i))
 Next
End Sub
```

```
The value of array in 0 is :Red
The value of array in 1 is : Blue
The value of array in 2 is : Yellow
```

Visual Basic for Applications
Array - Join

returnează un șir de caractere

Join(List[,delimiter])

```
Private Sub Constant_demo_Click()
  ' Join using spaces
  a = array("Red","Blue","Yellow")
  b = join(a)
  msgbox("The value of b " & " is :"  & b)

  ' Join using $
  b = join(a,"$")
  msgbox("The Join result after using
delimiter is : " & b)
End Sub
```

```
The value of b is :Red Blue Yellow
   The Join result after using
  delimiter is : Red$Blue$Yellow
```

Visual Basic for Applications

Array - Filter

returnează un șir

Filter(inputstrings,value[,include[,compare]])

```
Private Sub Constant_demo_Click()
  Dim a,b,c,d as Variant
  a = array("Red","Blue","Yellow")
  b = Filter(a,"B")
  c = Filter(a,"e")
  d = Filter(a,"Y")
  For each x in b
    msgbox("The Filter result 1: " & x)
  Next
  For each y in c
    msgbox("The Filter result 2: " & y)
  Next
  For each z in d
    msgbox("The Filter result 3: " & z)
  Next
End Sub
```

```
The Filter result 1: Blue
The Filter result 2: Red
The Filter result 2: Blue
The Filter result 2: Yellow
The Filter result 3: Yellow
```

Visual Basic for Applications
Array - isArray
returnează True / False

IsArray(variablename)

```
Private Sub Constant_demo_Click()
  Dim a,b as Variant
  a = array("Red","Blue","Yellow")
  b = "12345"

  msgbox("The IsArray result 1 : " &
IsArray(a))
  msgbox("The IsArray result 2 : " &
IsArray(b))
End Sub
```

```
The IsArray result 1 : True

The IsArray result 2 : False
```

Visual Basic for Applications
Array – Erase

Erase ArrayName

```
Private Sub Constant_demo_Click()
  Dim NumArray(3)
  NumArray(0) = "VBScript"
  NumArray(1) = 1.05
  NumArray(2) = 25
  NumArray(3) = #23/04/2013#
  Dim DynamicArray()
  ReDim DynamicArray(9)    ' Allocate storage space.
  Erase NumArray           ' Each element is reinitialized.
  Erase DynamicArray       ' Free memory used by array.
  ' All values would be erased.
  msgbox("Zeroth index of NumArray is " & NumArray(0))
  msgbox("First index of NumArray is " & NumArray(1))
  msgbox("Second index of NumArray is " & NumArray(2))
  msgbox("Third index of NumArray is " & NumArray(3))
End Sub
```

```
Zeroth index of NumArray is
First index of NumArray is
Second index of NumArray is
Third index of NumArray is
```

EVENIMENTE

*(**V**isual **B**asic for **A**pplications)*

Visual Basic for Applications Evenimente

- Clasificare
  - Evenimente care apar la schimbări în Worksheet
  - Evenimente care apar la schimbări în Workbook

Visual Basic for Applications

Evenimente care apar la schimbări în Worksheet

- Creare:
  - Click de dreapta pe tab-ul corespunzător foii de calcul + VIEW CODE
  - Selectare Worksheet + Selectare eveniment

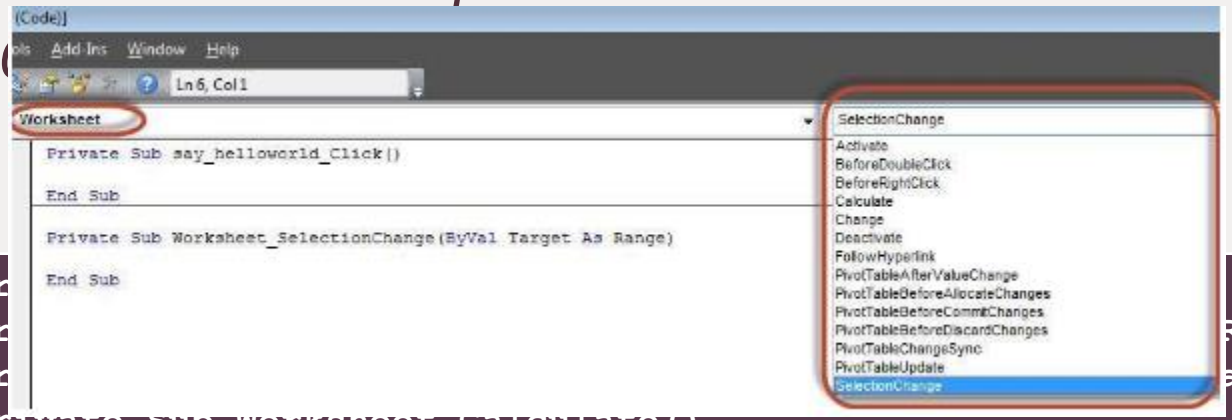
```
Private Sub Worksheet_Activate()
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
Private Sub Worksheet_BeforeRightClick(ByVal Target As Range, Cancel As Boolean)
Private Sub Worksheet_Calculate()
Private Sub Worksheet_Change(ByVal Target As Range)
Private Sub Worksheet_Deactivate()
Private Sub Worksheet_FollowHyperlink(ByVal Target As Hyperlink)
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```

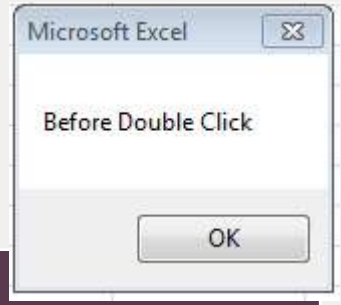
```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
    MsgBox "Before Double Click"
End Sub
```

Visual Basic for Applications

Evenimente care apar la s(

- Creare:
  - Click de dreapta pe tab-ul corespunzător foii de calcul + VIEW CODE
  - Selectare Worksheet + Selectare eveniment





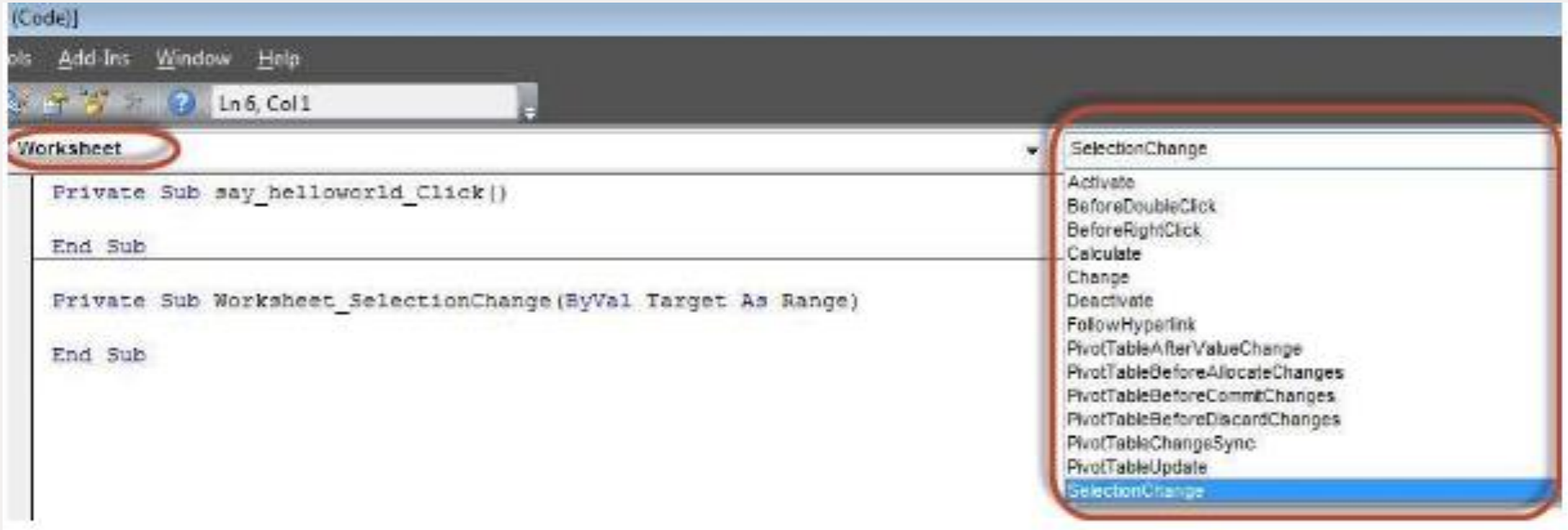

```
Pr
Pr                                                    Range, Cancel As Boolean)
Pr                                                    ange, Cancel As Boolean)
Private Sub Worksheet_Calculate()
Private Sub Worksheet_Change(ByVal Target As Range)
Private Sub Worksheet_Deactivate()
Private Sub Worksheet_FollowHyperlink(ByVal Target As Hyperlink)
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
    MsgBox "Before Double Click"
End Sub
```

Visual Basic for Applications

Evenimente care apar la schimbări în Worksheet

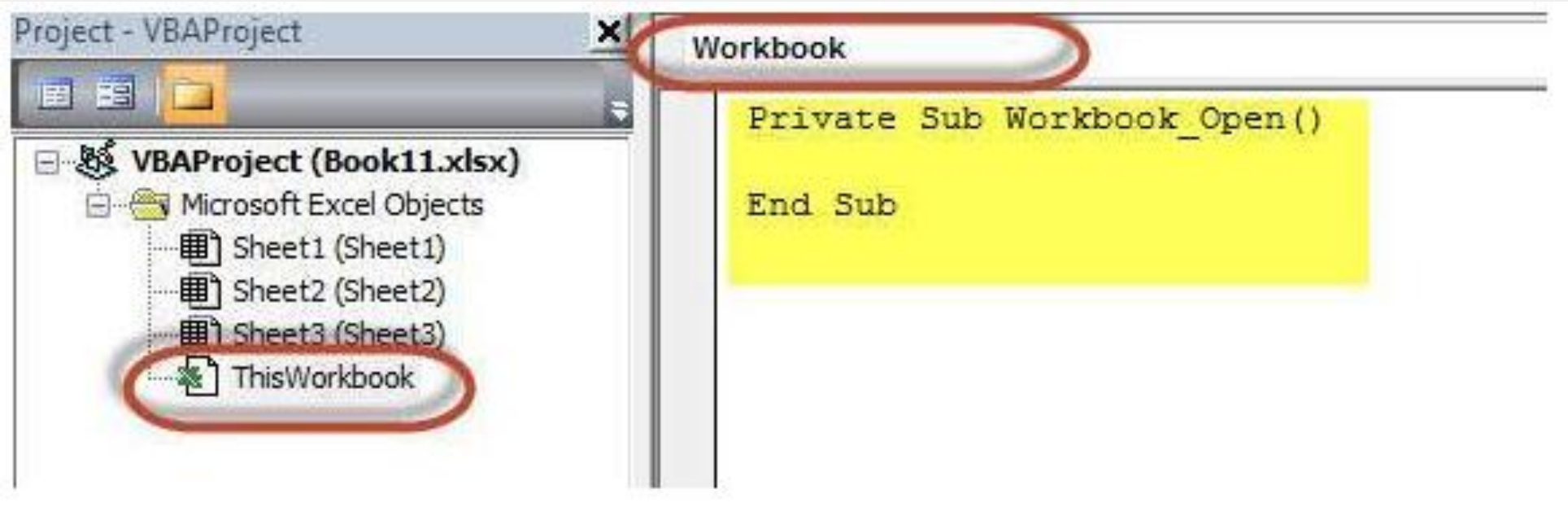
- Creare:
  - Click de dreapta pe tab-ul corespunzător foii de calcul + VIEW CODE
  - Selectare Worksheet + Selectare eveniment



Visual Basic for Applications

Evenimente care apar la schimbări în Workbook

- Creare:
  - Click de dreapta pe tab-ul corespunzător foii de calcul + VIEW CODE
  - Selectare This Workbook + Selectare eveniment



Visual Basic for Applications

Evenimente care apar la schimbări în Workbook

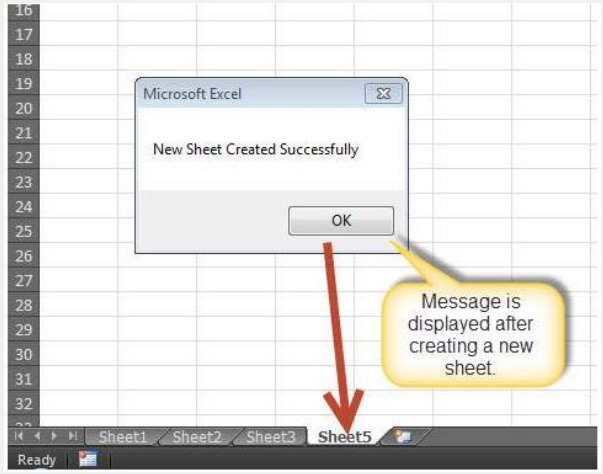
- Creare:
  - Click de dreapta pe tab-ul corespunzător foii de calcul + VIEW CODE
  - Selectare This Workbook + Selectare eveniment

```
Private Sub Workbook_AddinUninstall()
Private Sub Workbook_BeforeClose(Cancel As Boolean)
Private Sub Workbook_BeforePrint(Cancel As Boolean)
Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)
Private Sub Workbook_Deactivate()
Private Sub Workbook_NewSheet(ByVal Sh As Object)
Private Sub Workbook_Open()
Private Sub Workbook_SheetActivate(ByVal Sh As Object)
Private Sub Workbook_SheetBeforeDoubleClick(ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean)
```

Visual Basic for Applications

Evenimente care apar la schimbări în Workbook

- Creare:
  - Click de dreapta pe tab-ul corespunzător foii de calcul + VIEW CODE
  - Selectare This Workbook + Selectare eveniment

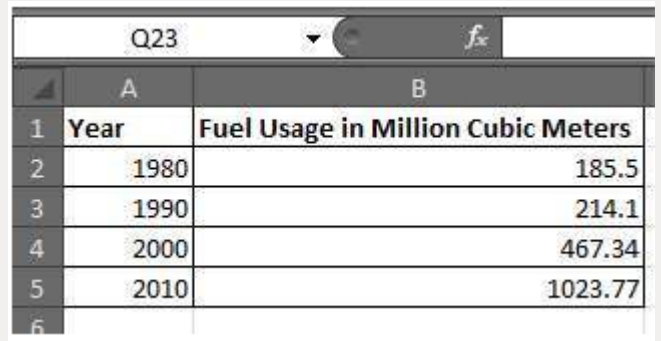



```
Private Sub Workbook_SheetBeforeRightClick(ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean)
Private Sub Workbook_SheetCalculate(ByVal Sh As Object)
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
Private Sub Workbook_SheetDeactivate(ByVal Sh As Object)
Private Sub Workbook_SheetFollowHyperlink(ByVal Sh As Object, ByVal Target As Hyperlink)
Private Sub Workbook_SheetSelectionChange(ByVal Sh As Object, ByVal Target As Range)
Private Sub Workbook_WindowActivate(ByVal Wn As Window)
Private Sub Workbook_WindowDeactivate(ByVal Wn As Window)
Private Sub Workbook_WindowResize(ByVal Wn As Window)
```

```
Private Sub Workbook_NewSheet(ByVal Sh As Object)
    MsgBox "New Sheet Created Successfully"
End Sub
```

Visual Basic for Applications

```
' Procedure to Generate Pie Chart
Private Sub fn_generate_pie_graph_Click()
    Dim cht As ChartObject
    For Each cht In Worksheets(1).ChartObjects
        cht.Chart.Type = xlPie
    Next cht
End Sub
' Procedure to Generate Bar Graph
Private Sub fn_Generate_Bar_Graph_Click()
    Dim cht As ChartObject
    For Each cht In Worksheets(1).ChartObjects
        cht.Chart.Type = xlBar
    Next cht
End Sub
' Procedure to Generate Column Graph
Private Sub fn_generate_column_graph_Click()
    Dim cht As ChartObject
    For Each cht In Worksheets(1).ChartObjects
        cht.Chart.Type = xlColumn
    Next cht
End Sub
```



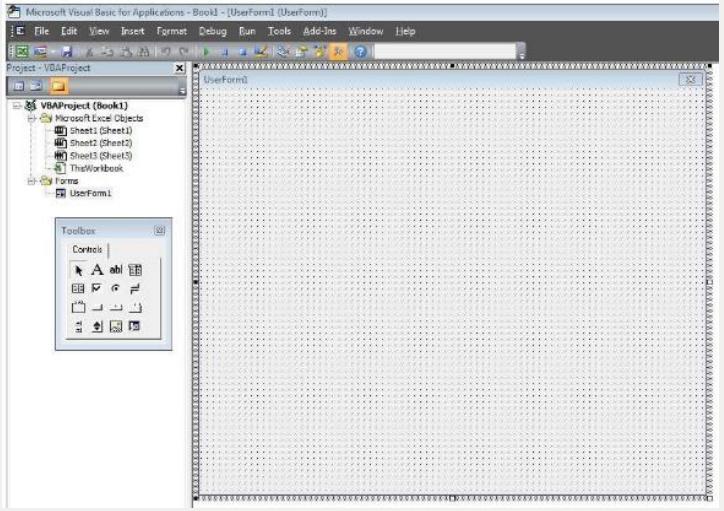
Q23

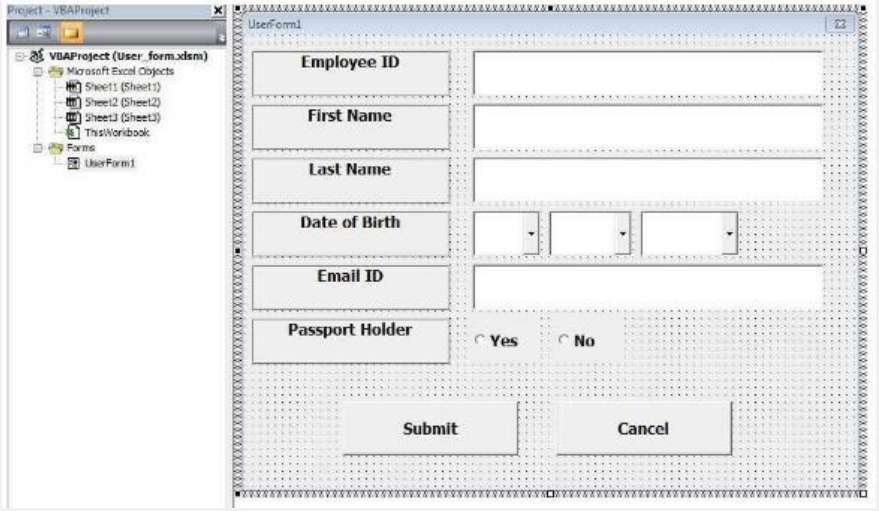
	A	B
1	Year	Fuel Usage in Million Cubic Meters
2	1980	185.5
3	1990	214.1
4	2000	467.34
5	2010	1023.77
6		

Visual Basic for Applications

Formulare

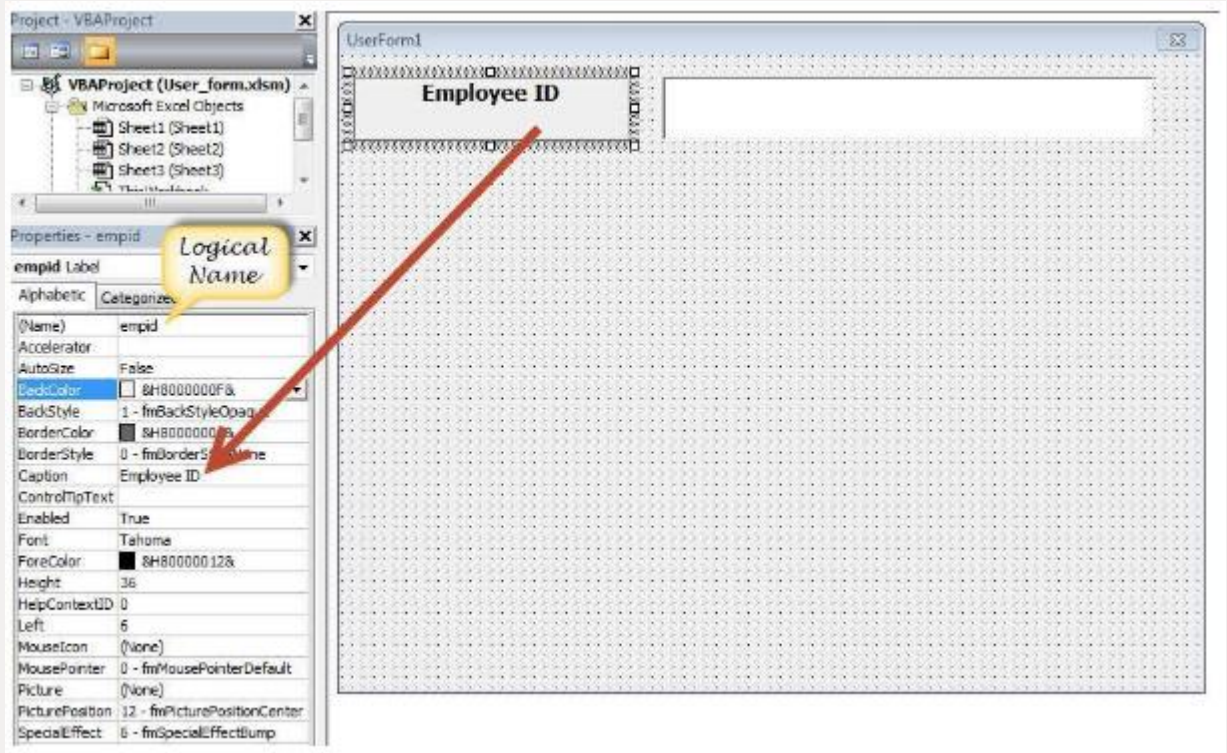
- ALT + F11
- Insert > User Form





Visual Basic for Applications

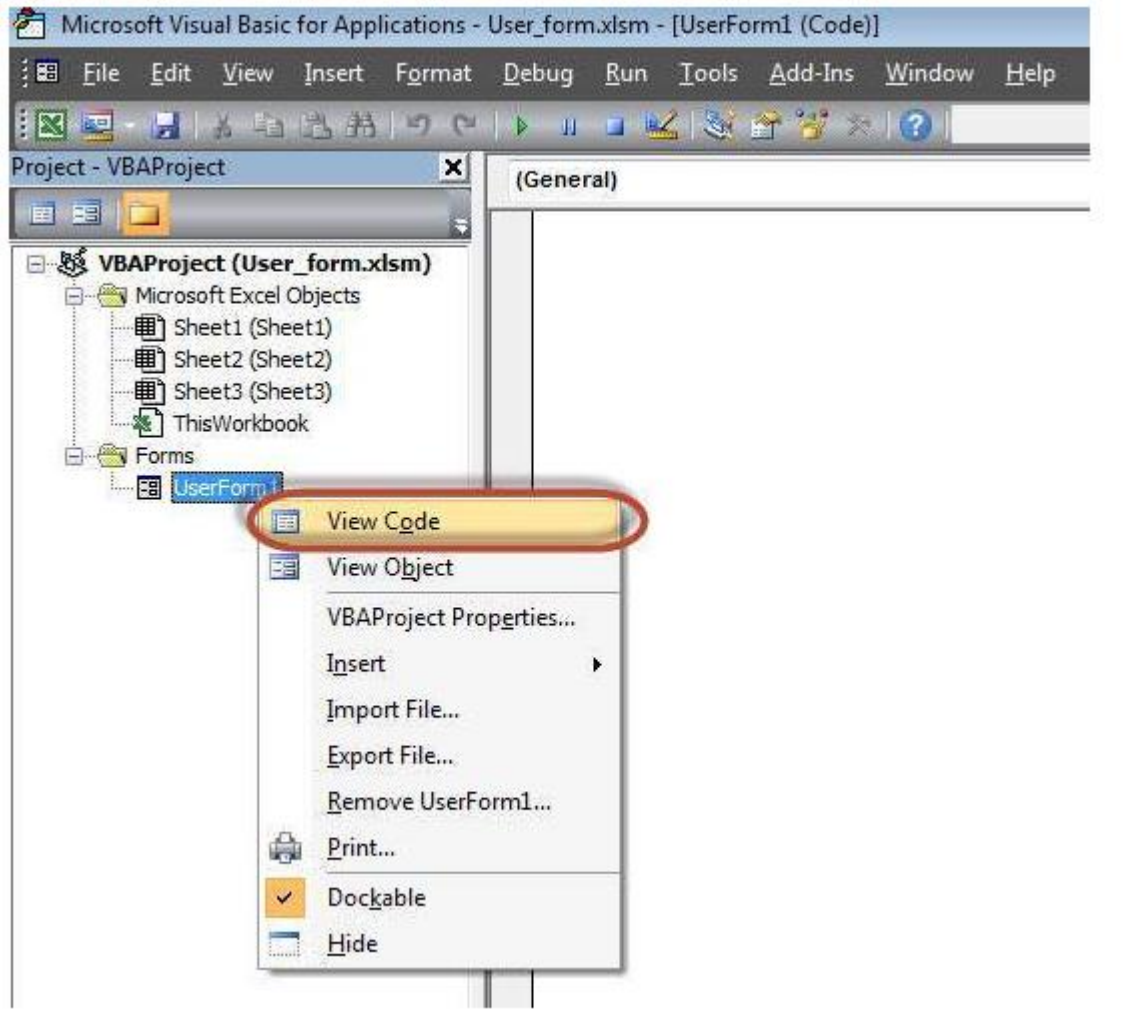
Formulare



Control	Logical Name	Caption
From	frmempform	Employee Form
Employee ID Label Box	empid	Employee ID
firstname Label Box	firstname	First Name
lastname Label Box	lastname	Last Name
dob Label Box	dob	Date of Birth
mailid Label Box	mailid	Email ID
Passportholder Label Box	Passportholder	Passport Holder
Emp ID Text Box	txtempid	NOT Applicable
First Name Text Box	txtfirstname	NOT Applicable
Last Name Text Box	txtlastname	NOT Applicable
Email ID Text Box	txtemailid	NOT Applicable
Date Combo Box	cmbdate	NOT Applicable
Month Combo Box	cmbmonth	NOT Applicable
Year Combo Box	cmbyear	NOT Applicable
Yes Radio Button	radioyes	Yes
No Radio Button	radiono	No
Submit Button	btnsubmit	Submit
Cancel Button	btncancel	Cancel

Visual Basic for Applications

Formulare



Visual Basic for Applications

Formulare

```
Private Sub UserForm_Initialize()
   txtempid.Value = ""
   txtempid.SetFocus

   'Empty all other text box fields
   txtfirstname.Value = ""
   txtlastname.Value = ""
   txtemailid.Value = ""

   'Clear All Date of Birth Related Fields
   cmbdate.Clear
   cmbmonth.Clear
   cmbyear.Clear

   'Fill Date Drop Down box - Takes 1 to
31
   With cmbdate
   .AddItem "1"
   .AddItem "2"
   .AddItem "3"
   .AddItem "4"
   .AddItem "5"
   .AddItem "6"
   .AddItem "7"
   .AddItem "8"
   .AddItem "9"
   .AddItem "10"
   .AddItem "11"
   .AddItem "12"
   .AddItem "13"
   .AddItem "14"
   .AddItem "15"
   .AddItem "16"
   .AddItem "17"
   .AddItem "18"
   .AddItem "19"
   .AddItem "20"
   .AddItem "21"
   .AddItem "22"
   .AddItem "23"
   .AddItem "24"
   .AddItem "25"
   .AddItem "26"
   .AddItem "27"
   .AddItem "28"
   .AddItem "29"
   .AddItem "30"
   .AddItem "31"
   End With

   'Fill Month Drop Down box - Takes
Jan to Dec
   With cmbmonth
   .AddItem "JAN"
   .AddItem "FEB"
   .AddItem "MAR"
   .AddItem "APR"
   .AddItem "MAY"
   .AddItem "JUN"
   .AddItem "JUL"
   .AddItem "AUG"
   .AddItem "SEP"
   .AddItem "OCT"
   .AddItem "NOV"
   .AddItem "DEC"
   End With

   'Fill Year Drop Down box - Takes
1980 to 2014
   With cmbyear
   .AddItem "1980"
   .AddItem "1981"
   .AddItem "1982"
   .AddItem "1983"
   .AddItem "1984"
   .AddItem "1985"
   .AddItem "1986"
   .AddItem "1987"
   .AddItem "1988"
   .AddItem "1989"
   .AddItem "1990"
   .AddItem "1991"
   .AddItem "1992"
   .AddItem "1993"
   .AddItem "1994"
   .AddItem "1995"
   .AddItem "1996"
   .AddItem "1997"
   .AddItem "1998"
   .AddItem "1999"
   .AddItem "2000"
   .AddItem "2001"
   .AddItem "2002"
   .AddItem "2003"
   .AddItem "2004"
   .AddItem "2005"
   .AddItem "2006"
   .AddItem "2007"
   .AddItem "2008"
   .AddItem "2009"
   .AddItem "2010"
   .AddItem "2011"
   .AddItem "2012"
   .AddItem "2013"
   .AddItem "2014"
   End With

   'Reset Radio Button. Set it to False
when form loads.
   radioyes.Value = False
   radiono.Value = False

End Sub
```

Visual Basic for Applications

Formulare

```
Private Sub btnsubmit_Click()
  Dim emptyRow As Long

  'Make Sheet1 active
  Sheet1.Activate

  'Determine emptyRow
  emptyRow = WorksheetFunction.CountA(Range("A:A")) + 1

  'Transfer information
  Cells(emptyRow, 1).Value = txtempid.Value
  Cells(emptyRow, 2).Value = txtfirstname.Value
  Cells(emptyRow, 3).Value = txtlastname.Value
  Cells(emptyRow, 4).Value = cmbdate.Value & "/" & cmbmonth.Value & "/" & cmbyear.Value
  Cells(emptyRow, 5).Value = txtemailid.Value

  If radioyes.Value = True Then
     Cells(emptyRow, 6).Value = "Yes"
  Else
     Cells(emptyRow, 6).Value = "No"
  End If
End Sub
```

```
Private Sub btncancel_Click()
    Unload Me
End Sub
```