

VISUAL BASIC FOR APPLICATIONS

Damiano SOMENZI

SCHOOL OF ECONOMICS AND MANAGEMENT

Advanced Computer Skills

`damiano.somenzi@unibz.it`



Week 7

① IF..THEN..ELSE STATEMENT

② SELECT CASE STATEMENT

VBA CONDITIONAL CONTROL

IF..THEN..ELSE STATEMENT

IF..THEN..ELSE

It executes a group of statements, depending on the value of an expression. The syntax is

```
If condition Then [statements] [Else elsestatements]
```

Or the block form syntax

```
If condition Then [statements]  
[Elseif condition-n Then [elseifstatements] ...  
[Else [elsestatements]]  
End If
```

VBA CONTROL STATEMENT

IF..THEN..ELSE STATEMENT

IF..THEN..ELSE

- `condition` (required) One or more numeric expressions or string expressions that evaluates to **True** or **False**
- *statements* (Optional in block form while required in single-line form that has no Else clause) One or more statements separated by colons; executed if condition is **True**
- `condition-n` (optional), same as `condition`
- *elseifstatements* (optional) One or more statements executed if the associated `condition-n` is **True**
- *elsestatements* (optional) One or more statements executed if no previous `condition` or `condition-n` expression is **True**

IF..THEN STATEMENT

CASE ONE

CASE ONE - EXAMPLE

If Len (Text) > Max_Char **Then** MsgBox ("The text is too long")

Text	Max_Char	Condition	Result
"Advanced Computer Skills"	20	True	The text is too long
"Computer Science"	20	False	–

CASE ONE - EXAMPLE

If Income >= 20000 **Then** Tax = Income * 0.35

Income (€)		Condition	Tax (€)
32450	20000	True	11357.5
14600	20000	False	0

IF..THEN..ELSE STATEMENT

CASE TWO

CASE TWO - EXAMPLE

```
If Len(Text) < Max_Char Then  
    MsgBox ((Max_Char - Len(Text)) & "character(s) could be entered")  
Else  
    MsgBox ("The text is too long")  
End If
```

Text	Max Char	Condition	Result
"Advanced Computer Skills"	20	False	The text is too long
"Computer Science"	20	True	4 character(s) could be entered

IF..THEN..ELSE STATEMENT

CASE TWO

CASE TWO - EXAMPLE

```
If Income >= 20000 Then  
    Tax = Income * 0.35  
Else  
    Tax = Income * 0.20  
End If
```

Income (€)		Condition	Tax (€)
32450	20000	True	11357.5
14600	20000	False	2920

IF..THEN..ELSEIF..THEN..ELSE

CASE THREE

CASE THREE - EXAMPLE

```
If Len(Text) > Max_Char Then  
    MsgBox ("The text is too long")  
Elseif Len(Text) < Min_Char Then  
    MsgBox ("The text is too short")  
Else  
    MsgBox ((Max_Char - Len(Text)) & "character(s) could be entered")  
End If
```

Text	Min	Max	Cond1	Cond2	Result
"Advanced Computer Skills"	10	20	True	-	The text is too long
"Computer Science"	10	20	False	False	4 character(s) ...
"Excel VBA"	10	20	False	True	The string is too short

IF..THEN..ELSEIF..THEN..ELSE

CASE THREE

CASE THREE - EXAMPLE

```
If Income >= 20000 Then  
    Tax = Income * 0.35  
Elseif Income <= 10000 Then  
    Tax = Income * 0.08  
Else  
    Tax = Income * 0.20  
End If
```

Income (€)			Cond1	Cond2	Tax (€)
32450	20000	10000	True	-	11357.5
9750	20000	10000	False	True	780
14600	20000	10000	False	False	2920

IF..THEN..ELSEIF..ELSE..ENDIF

EXAMPLE

EXAMPLE ONE

```
Sub CheckTemperature(t As Double)
  If t > 30 Then
    MsgBox ("t = " & t & " -> HOT")
  ElseIf t > 20 Then
    MsgBox ("t = " & t & " -> WARM")
  ElseIf t > 10 Then
    MsgBox ("t = " & t & " -> FINE")
  Else
    MsgBox ("t = " & t & " -> COLD")
  End If
End Sub

Sub Main()
  Call CheckTemperature(32.5)
  Call CheckTemperature(-3)
  Call CheckTemperature(18.5)
End Sub
```

IF..THEN..ELSEIF..ELSE..ENDIF

EXAMPLE

EXAMPLE TWO

```
Sub FindGroup(st As String)
    If ((st Like "A*") Or (st Like "B*")) Then
        MsgBox ("Student = " & st & " -> Group A")
    ElseIf ((st Like "S*") Or (st Like "T*")) Then
        MsgBox ("Student = " & st & " -> Group B")
    ElseIf (st Like "M*") Then
        MsgBox ("Student = " & st & " -> Group C")
    Else
        MsgBox ("Student = " & st & " -> Group D")
    End If
End Sub

Sub Main()
    Call FindGroup("Smith")
    Call FindGroup("Bauer")
    Call FindGroup("Fisher")
End Sub
```

EXERCISE 1

Write a procedure **Main** which asks the user for an integer number, then it calls the procedure **CheckNumber** and passes to it this number.

CheckNumber checks:

- Whether the number is equal to 2
- If not checks whether it is greater than 2 or less than 2

It displays the result using the *Message Box* function

SOLUTION

```
Sub CheckNumber(n As Integer)
    If n = 2 Then
        MsgBox (n & " is equal to 2")
    ElseIf n > 2 Then
        MsgBox (n & " is greater than 2")
    Else
        MsgBox (n & " is less than 2")
    End If
End Sub

Sub Main()
    Dim num As Integer
    num = Val(InputBox("Enter a number"))
    Call CheckNumber(num)
End Sub
```

EXERCISE 2

Write a procedure **Main** which asks the user for a sentence (string) and then for a number (integer), then it calls the procedure **CheckLengthText** and it passes both to it.

CheckLengthText checks:

- Whether the string's length is equal to the number
- If not checks whether is larger or smaller than the number

It displays the result using the *Message Box* function

SOLUTION

```
Sub CheckLengthText(text As String, l As Integer)
    If Len(text) = l Then
        MsgBox ("The length of (" & _
            text & ") is equal to (" & l & ")")
    ElseIf Len(text) > l Then
        MsgBox ("The length of (" & _
            text & ") is larger than (" & l & ")")
    Else
        MsgBox ("The length of (" & _
            text & ") is not smaller than (" & l & ")")
    End If
End Sub

Sub Main()
    Dim s As String
    Dim n As Integer
    s = InputBox("Enter the sentence")
    n = Val(InputBox("Enter the number"))
    Call CheckLengthText(s, n)
End Sub
```

EXERCISE 3

Given the coefficients a , b , c of the quadratic equation $ax^2 + bx + c = 0$ write a procedure **QuadraticEquation** which computes and displays the solutions. The **Main** procedure asks the user for the coefficients, it calls **QuadraticEquation** and passes to it the coefficients

SOLUTION

```
Sub QuadraticEquation(a As Double, b As Double, c As Double)
    Dim d As Double
    d = b ^ 2 - 4 * a * c
    If d > 0 Then
        MsgBox ("x1 = " & Round((-b + Sqr(d)) / 2 * a, 2) & _
            " x2 = " & Round((-b - Sqr(d)) / 2 * a, 2))
    ElseIf d = 0 Then
        MsgBox ("x = " & Round(-b / 2 * a, 2))
    Else
        MsgBox ("No real solutions")
    End If
End Sub

Sub Main()
    Dim ca As Double
    ca = Val(InputBox("Enter the coefficient a"))
    Dim cb As Double
    cb = Val(InputBox("Enter the coefficient b"))
    Dim cc As Double
    cc = Val(InputBox("Enter the coefficient c"))
    Call QuadraticEquation(ca, cb, cc)
End Sub
```

VBA CONDITIONAL CONTROL

SELECT CASE STATEMENT

SELECT CASE

It executes one of several groups of statements, depending on the value of an expression. The syntax is

```
Select Case testexpression  
[Case expressionlist-n [statements-n]] ...  
[Case Else [elstatements]]  
End Select
```

VBA CONTROL STATEMENT

SELECT CASE STATEMENT

SELECT CASE

- *testexpression* (required) Any numeric expression or string expression
- *expressionlist-n* (required if a Case appears) Delimited list of one or more of the following forms: *expression*, *expression To expression*. The **To** keyword specifies a range of values
- *statements-n* (optional) One or more statements executed if *testexpression* matches any part of *expressionlist-n*
- *elsetatements* (optional) One or more statements executed if *testexpression* doesn't match any of the Case clause

SELECT CASE

EXAMPLE

EXAMPLE

```
Select Case n  
Case 10 To 20  
    MsgBox("n ∈ [10, 20]")  
Case 21 To 30  
    MsgBox("n ∈ [21, 30]")  
Case 31 To 40  
    MsgBox("n ∈ [31, 40]")  
Case Else  
    MsgBox("n ∉ [10, 40]")  
End Select
```

n	Case	Result
25	21 To 30	"n ∈ [21, 30]"
31	31 To 40	"n ∈ [31, 40]"
88	Else	"n ∉ [10, 40]"

SELECT CASE

EXAMPLE

EXAMPLE ONE

```
Sub DaysOfMonth(m As Integer)
    Dim days As Integer
    Select Case m
        Case 4, 6, 9, 11
            days = 30
        Case 1, 3, 5, 7, 8, 10, 12
            days = 31
        Case 2
            days = 28
        Case Else
            days = 0
    End Select
    MsgBox ("month = " & m & " -> days = " & days)
End Sub

Sub Main()
    Dim month As Integer
    month = Val(InputBox("Enter the month"))
    Call DaysOfMonth(month)
End Sub
```