

Visual Basic for Applications Programming

Damiano SOMENZI

School of Economics and Management

Advanced Computer Skills

`damiano.somenzi@unibz.it`

Week 5



Outline

- 1 Do ... Loop
 - Do While .. Loop
 - Do .. Loop While
- 2 Examples
- 3 Exercises

Outline

- 1 Do ... Loop
 - Do While .. Loop
 - Do .. Loop While
- 2 Examples
- 3 Exercises

Do While .. Loop Statement

Do While .. Loop Statement

Repeats a block of statements while a condition is **True**.
We look at the following syntax as reference

Do While *condition*
 [statements]
Loop

```
x = 0.1 (set)  
Do While Log(x) < 1 (condition)  
    x = x + 0.1 (statement)  
Loop
```

Do While...Loop Statement

Do While...Loop Statement

- *condition*, is a numeric expression or string expression that is **True** or **False**
 - .. check the condition before entering the loop ..
 - .. when the expression is **False** exit **Do** statement and transfer control to the statement immediately following the **Loop** ..
- *statements*, one or more statements that are repeated while condition is **True**
 - .. tasks to be accomplished an indefinite number of times ..

Outline

- 1 Do ... Loop
 - Do While .. Loop
 - Do .. Loop While
- 2 Examples
- 3 Exercises

Do .. Loop While Statement

Do .. Loop While Statement

Repeats a block of statements while a condition is **True**.
We look at the following syntax as reference

Do

[statements]

Loop While *condition*

```
x = 0 (set)
```

```
Do
```

```
    x = x + 0.1 (statement)
```

```
Loop While Log(x) < 1 (condition)
```

Do ...Loop While Statement

Do ... Loop While Statement

- *statements*, one or more statements that are repeated while condition is True
 - .. tasks to be accomplished an indefinite number of times ..
- *condition*, is a numeric expression or string expression that is **True** or **False**
 - .. check the condition **after the loop has run at least once** ..
 - .. when the expression is **False** exit **Do** statement and transfer control to the statement immediately following the **Loop While** ..

Do...Loop Statement

Set of Examples

```
Function fraction(ByVal epsilon As Single) As Long

    ' the function determines the smaller n such that in
    ' the infinite sequence 1/1, 1/2, 1/3, .. 1/n, .. (n > 0)
    ' 1/n value is smaller than or equal to epsilon
    ' epsilon > 0

    Dim n As Long
    If epsilon > 0 Then
        n = 1
        Do While 1 / n > epsilon
            n = n + 1
        Loop
        fraction = n
    Else
        fraction = -1
    End If
End Function
```

Exercises

one

In computer science we typically manage integer numbers in binary representation.

A simple but useful function gets the decimal representation of a positive integer number and determines the corresponding binary representation.

We are aware that binary numbers can be represented only as a sequence of characters 0/1.

(i.e. $207 \rightarrow 11001111$)

\ Operator

- Used to divide two numbers and return an integer result
- **Syntax:** `result = number1 \ number2`

Exercises

```
Function decimalTobinary(ByVal n As Integer) As String

    ' n is a positive integer number in decimal representation
    ' the function determines the corresponding
    ' binary representation (sequence of the characters 0/1

    Dim b As String
    b = ""
    Do While n > 0
        b = (n Mod 2) & b
        n = n \ 2
    Loop
    decimalTobinary = b
End Function
```

Exercises

two

We have a function “mirror” that gets a sequence of characters and returns the same sequence, but in the reverse order:

- `"University" -> "ytisrevinU"`

A function “broken mirror” in a similar way gets a sequence of character plus a “special” character occurring in it, and returns the same sequence, but the characters from the beginning to the previous first occurrence of the “special” character in the reverse order:

- `"University", "e" -> "vinUersity"`
- `"Computer Science", " " -> "retupmoC Science"`

Exercises

```
Function brokenMirror(ByVal text As String, ByVal b As String) As String

    ' the function gets a text and a special character occurring in it
    ' the function returns the same sequence but the characters from
    ' the beginning to the previous special character in the reverse order

    Dim i As Integer
    Dim m As String
    i = 1
    m = ""
    Do While Mid(text, i, 1) <> b
        m = Mid(text, i, 1) & m
        i = i + 1
    Loop
    m = m & Mid(text, i) 'Mid returns all characters from i to the end
    brokenMirror = m
End Function
```

Exercises

three

Assume that a quantity, for example a social or an economics indicator, grows every month as $10 + 1.5^t$.

($t_1 = 10 + 1.5^1 = 11.5$, $t_2 = 10 + 1.5^2 = 12.25$, ...).

A useful function should estimate the number of months after that the quantity exceeds a target value

Exercises

```
Function growth(ByVal target As Double) As Integer

    Rem the function estimates the number of months after that
    Rem the quantity exceeds the target value

    Dim t As Integer
    t = 0
    Do
        t = t + 1
    Loop While (10 + 1.5 ^ t) < target
    growth = t
End Function
```