

Visual Basic for Applications

Aula 02

Raul Ikeda

2º semestre de 2017

Sub – Subrotinas

- “Function” que não retorna valor.
- Também pode ter argumentos de entrada.
- Tanto Sub quanto Function podem ter argumentos:
 - ByVal – variável passada por valor, uma cópia da variável original.
 - ByRef – variável passada por referência, ou seja, a própria variável é passada.
- É possível realizar uma chamada de uma Sub dentro de uma Sub ou uma Function dentro de uma Sub.
- Faz sentido fazer uma chamada de Sub dentro de uma Function?

Sub – Subrotinas

- Exemplo:

```
Sub TestarArgumento()  
  
    Dim x As Integer  
    Dim y As Integer  
  
    x = 3  
    y = 3  
  
    Call Calcular(x, y)  
    'Ou:  
    Calcular x, y 'Atenção aos parênteses!  
  
End Sub  
  
Sub Calcular(ByVal x As Integer, ByRef y As Integer)  
  
    x = x + 1  
    y = y + 1  
  
End Sub
```

Sub – Input/Output

- Inputbox – Abre uma tela com caixa de entrada para o usuário.
- MsgBox – Abre uma tela de confirmação para o usuário. Pode retornar um valor que indica qual botão foi apertado.
- Exemplo (Perceba o uso do MsgBox como Function e como Sub):

```
Sub Dobrar()  
  
    Dim x As Integer, y As Integer  
  
    If MsgBox("Deseja calcular o dobro de um número?", vbYesNo, "Dobro") = vbYes Then  
        x = CInt(InputBox("Digite um número inteiro"))  
        y = 2 * x  
        MsgBox "O dobro do número é: " & y, vbOKOnly, "Resultado"  
    End If  
  
End Sub
```

- Pergunta: Faz sentido usar Inputbox/MsgBox em uma Function?

Sub – Input/Output

- Cells: É possível ler e escrever em células.
- Exemplo:

```
Dim x As Double
x = CDBl(Workbooks("Pastal").Worksheets("Plan1").Cells(2, 3).Value)
Workbooks("Pastal").Worksheets("Plan1").Cells(2, 4).Value = x ^ 2
```

- Utilizando *With* (apenas estético):

```
Sub Dobrar()

    Dim f As Variant
    With Workbooks("Pastal")
        With .Worksheets("Plan2")
            f = CStr(.Cells(2, 3).Interior.Color)
            .Cells(2, 4).Interior.Color = f
            .Cells(2, 3).Value = 2
            .Cells(2, 4).Formula = "=C2^2"
        End With
    End With

End Sub
```

- Note que existem outras propriedades de um objeto Range (Cells).
Aperte F2 para ver as definições dos objetos.

Sub – Exercício 01

- 15 min. – Fazer uma Sub que descobre qual a primeira linha vazia em uma planilha.

```
Sub DescobrirLinhaVazia()  
  
    Dim n As Integer 'Quantas linhas preenchidas  
  
    n = 0  
    While Cells(n + 1, 1).Value <> vbEmpty  
        n = n + 1  
    Wend  
  
    'n + 1 é a primeira linha vazia!  
  
    'O que acontece quando não tem o prefixo "Workbooks..."?  
    'Teste em planilhas diferentes  
  
End Sub
```

Sub – Eventos

- Capturam uma ação assíncrona.
- Exemplos:

```
'Dentro do espaço do Plan1
Private Sub Worksheet_SelectionChange(ByVal Target As Range)

    'Selecione uma célula diferente
    MsgBox Target.Row & " - " & Target.Column

End Sub
```

```
'Dentro do espaço do Workbook
Private Sub Workbook_Open()

    MsgBox "Arquivo Aberto!"

End Sub
```

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)

    Cancel = True
    MsgBox "Cancela o fechamento do arquivo!"

End Sub
```

Sub – Vetores e Matrizes

- Funcionam como uma lista n-dimensional porém com quantidade fixa de posições. Contém tipo único e fixo.
- Declaração:

```
Dim m As Integer, n As Integer
Dim v() As Double, A() As Single
m = CInt(InputBox("Quantas linhas?"))
n = CInt(InputBox("Quantas colunas?"))

ReDim v(1 To m) As Double
ReDim A(1 To m, 1 To n) As Double
'Redim C(1 to m, 1 to n, 1 to p, ...) As ...
```

- Descobrendo o tamanho:
 - LBound() – qual o índice inferior da matriz
 - UBound() – qual o índice superior da matriz

Sub – Exercício 02

- 15 min. – Fazer uma função que entra uma matriz e retorna a sua transposta.
- Lembrando que dado C_{mn} :

$$c_{ij}^T = c_{ji}$$

```
Function TransporMatriz(C As Range) As Variant()  
  
    Dim m As Integer, n As Integer, i As Integer, j As Integer  
    Dim ct() As Variant  
  
    m = C.Rows.Count  
    n = C.Columns.Count  
  
    ReDim ct(1 To m, 1 To n) As Variant  
  
    For i = 1 To m  
        For j = 1 To n  
            ct(i, j) = CDb1(C(j, i).Value)  
        Next j  
    Next i  
  
    TransporMatriz = ct  
  
End Function
```

Sub – Exercício 02

- 20 min. – Fazer uma função que entra duas matrizes e retorna a multiplicação das duas matrizes.
- Lembrando que dado $C_{mp} = A_{mp}B_{np}$:

$$c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}$$

Sub – Exercício 02

```
Function MultiplicarMatriz(A As Range, B As Range) As Variant()  
  
    Dim i As Integer, j As Integer, k As Integer  
    Dim m As Integer, n As Integer, p As Integer  
    Dim C() As Variant  
  
    m = A.Rows.Count  
    n = A.Columns.Count  
    p = B.Columns.Count  
  
    ReDim C(1 To m, 1 To p) As Variant  
  
    For i = 1 To m  
        For j = 1 To p  
            C(i, j) = 0  
            For k = 1 To n  
                C(i, j) = C(i, j) + CDbl(A(i, k).Value) * CDbl(B(k, j).Value)  
            Next k  
        Next j  
    Next i  
  
    MultiplicarMatriz = C  
  
End Function
```

Sub – Exercício 03

- Fazer uma Sub que entra uma matriz A_{mp} e fazer a decomposição LU de A:

$$A_{mn} = L_{mn}U_{nn}$$

- Onde L é uma matriz triangular inferior e U uma matriz triangular superior.
- Algoritmo:
 - Para $k = 1, \dots, m$
 - Para $i = 1, \dots, n$
 - $u_{ki} = a_{ki} - \sum_{j=1}^{k-1} l_{kj}u_{ji}$
 - $l_{kk} = 1$
 - Para $i = k + 1, \dots, m$
 - $l_{ik} = \frac{a_{ik} - \sum_{j=1}^{k-1} l_{ij}u_{jk}}{u_{kk}}$