VETORES E MATRIZES

By Eduardo Vieira Machado (Good Guy)

Vetores e matrizes são capazes de armazenar dados de forma organizada. Vetores fazem isso de forma escalar, isto é, com um dado de cada vez em sequência, de forma linear, em série, determinada pelo programador. Matrizes fazem isso por meio de linhas e colunas.

Poderemos ter um vetor com 'n' valores determinado por uma variável, com estes valores totalizados entre parênteses. Estes valores são os índices e cada vetor com índice chama-se elemento.

VETORES I

Ex(1)

```
Private Sub cmdMatrix_Click()

Dim Carro(5) As String  'Ou Dim Carro(0 To 5)
Dim n As Long

Carro(0) = "Gol"
Carro(1) = "Palio"
Carro(2) = "Voyage"
Carro(3) = "Crossfox"
Carro(4) = "Spacefox"

'Looping gerador do escalonamento dos vetores
For n = LBound(Carro) To UBound(Carro)  'LBound é o limite inferior da Matrix e UBound é o limite superior da Matrix (0 e 4 respectivamente)

   MsgBox Carro(n), vbInformation, "Matrix"

Next n

End Sub
```

Agora vamos supor que eu desejasse popular uma caixa de texto do meu formulário com o valor definido pelo usuário do modelo de carro.

Ex(2)

```
Private Sub cmdMatrix_Click()
Dim Carro(5) As String
Dim n As Long

Carro(0) = "Gol"
```

```
Carro(1) = "Palio"
Carro(2) = "Voyage"
Carro(3) = "Crossfox"
Carro(4) = "Spacefox"

n = InputBox("Digite a ordem do carro:", "Matrix")

Tipo = Carro(n)  'Tipo seria um campo não acoplado ou acoplado(a um campo da tabela) presente no formulário

End Sub
```

Ex(3)

```
Private Sub cmdMatrix_Click()

Dim Carro(5) As String 'Ou Dim Carro(0 To 4)
Dim n As Long

Carro(0) = DLookup("Tipo", "tblCarros", "Codigo = 1")
Carro(1) = DLookup("Tipo", "tblCarros", "Codigo = 2")
Carro(2) = DLookup("Tipo", "tblCarros", "Codigo = 3")
Carro(3) = DLookup("Tipo", "tblCarros", "Codigo = 4")
Carro(4) = DLookup("Tipo", "tblCarros", "Codigo = 5")

n = InputBox("Digite a ordem do carro:", "Matrix")

Tipo = Carro(n)

End Sub
```

Ex(4)

Neste exemplo vamos trabalhar com dados aleatórios.

```
Private Sub cmdMatrix_Click()

Dim Carro(5) As String

Carro(0) = "Gol"
Carro(1) = "Palio"
Carro(2) = "Voyage"
Carro(3) = "Crossfox"
Carro(4) = "Spacefox"

Call Randomize

    Tipo = Carro(Int(5 * Rnd) + 1)
```

```
End Sub
```

Testes esses exemplos e veja o que acontece.

Perceba que eu determinei que a minha variável fosse do tipo String. Se eu não soubesse que tipo de variável
meu programa trabalharia, definiria minha variável como do tipo Variant. O compilador procurará um tipo de variável que mais se adapte aos valores armazenados no vetor.

VETORES II*:

Em alguma ocasião, você precisará de um array que precisa se manter aumentando em número e
você não sabe até onde vai parar o limite superior. Por exemplo, suponhamos que você queira incrementar um tocador de MP3 e precisa pedir ao usuário para indicar ou acrescentar nomes de canções. Você poderia tornar isso possível dessa maneira:

```
Private Sub cmdMatrix_Click()

Dim strCancaoNomes() As String    'Array de nomes de canções
Dim blDimensionada As Boolean     'Está o Array Dimensionado
Dim strTexto As String            'Guarda temporariamente nomes de canções
Dim lngPosicao As Long            'Contador

'O array ainda não foi dimensionado
blDimensionada = False

Do

   'Pede o nome de uma canção
   strTexto = InputBox("Entre com um nome de canção:")

   If strTexto <> "" Then

      'O array foi redimensionado?
      If blDimensionada = True Then

         'Sim, extenda o array um elemento maior que o limite superior atual.
         'Sem a instrução "Preserve" abaixo, os elementos anteriores de nosso array seriam apagados
         com o redimensionamento
         ReDim Preserve strCancaoNomes(0 To UBound(strCancaoNomes) + 1) As String

      Else
         ‘Não, então dimensione e assinale quando dimensionado
         ReDim strCancaoNomes(0 To 0) As String
```

```
        blDimensionada = True

    End If

    'Acrescenta o nome da canção ao último elemento do array
    strCancaoNomes(UBound(strCancaoNomes)) = strTexto

  End If

Loop Until strTexto = ""  'Encerra o loop quando você clica em Cancelar

'Exibe os nomes das canções inclusas

For lngPosicao = LBound(strCancaoNomes) To UBound(strCancaoNomes)

  MsgBox strCancaoNomes(lngPosicao)

Next  lngPosicao

'Erase array

Erase strCancaoNomes


End Sub
```

A função Split aplicada em um Array.*

Pode ser que em algum momento, entremos em situações onde vamos querer extrair informação de dentro de uma "string" dada, separá-la em múltiplas strings e, então dispor essas strings em um array. Por exemplo, digamos que tivéssemos este código:

```
Dim cdLista As String
cdLista = "Não se preocupem, OK Pessoal, me importo com o que vocês fazem, Eduardo Machado"
```

Seria legal se pudéssemos facilmente pegar esta string e dispo-la em um array, não seria? Isto pode ser feito ao aplicarmos funções que manipulam strings, assim, ao escrever e atualizar este código poderia ser cansativo e consumiria o nosso tempo. Felizmente para nós, o Visual Basic proporciona uma função específica chamada "Split" que nos permite facilmente analisar a informação de uma string e dispô-la dentro de um array. Tem a seguinte sintaxe:

ArrayName = `split(`*String Input*[, *Delimiter*[, *Length Limit*[, *Compare Mode*]]]`)`

- `String Input(String Digitada)` é a string que você quer analisar

- Delimiter (Delimitador) é um parâmetro opcional que indica que tipo de string separa os elementos em uma string digitada. Por padrão, este parâmetro está ajustado para vazio ("). Isto significa que uma string digitada, por exemplo, "Isto É Um Teste" apresentaria um array de 4(quatro) elementos ("Isto", "É", "Um", "Teste").

- *Length Limit* (Limite de Comprimento) é o tamanho máximo que o seu array resultante pode alcançar.

- *Compare Mode*. Por padrão, o Visual Basic compara strings, caracter por caracter usando seus valores ASCII. No entanto, você pode usar modos diferentes que vão levar o Visual Basic a comparar strings de forma diferente. Por exemplo, vbTextCompare causa comparações de string se tornarem "case insensitive", isto é, aceitar tanto letras maiúsculas quanto minúsculas. Estes parâmetro efetiva como o Delimitador analisa a string digitada.

Private Sub cmdMatrix_Click()

Dim strCDRack() As String
Dim cdLista As String
Dim i As Integer

cdLista = "Não se preocupem, OK Pessoal, me importo com o que vocês fazem, Eduardo Machado"
strCDRack = Split(cdLista, ", ")

For i = LBound(strCDRack) To UBound(strCDRack)

MsgBox strCDRack(i)

Next i

End Sub

A Função Join Aplicada em um Array.*

A função Split nos permitiu fragmentar strings e encaixá-las em arrays, existe uma função que nos permite criar arrays e torná-las em uma grande string? Sim, existe, é chamada Join. Join é uma funçãomuito simples. Tem a seguinte sintaxe:

StringName = **join**(*Array Input*[, *Delimiter*])
- `Array Input` é o array que você quer colocar dentro de uma string.
- *Delimiter* é um parâmetro opcional que indica o que você quer colocar entre os elementos que são adicionado a string. Por padrão, este parâmetro está ajustado para vazio ("").

Utilizando um de nossos exemplos anteriores, aqui está alguns códigos de amostra em como usar **join:**

Private Sub cmdMatrix_Click()

```
Dim strAmigos(0 to 6) As String, lngPosicao as Long

strAmigos(0) = "Bianca"
strAmigos(1) = "Jeana"
strAmigos(2) = "Sam"
strAmigos(3) = "Jenna"
strAmigos(4) = "Erin"
strAmigos(5) = "Carolyn"
strAmigos(6) = "Kate"

Dim meusAmigos As String
```

```
'O resultado da string será: "Bianca, Jeana, Sam, Jenna, Erin, Carolyn, Kate"
meusAmigos = Join(strAmigos, ", ")

MsgBox meusAmigos
```

*Tradução do tutorial Visual Basic Array Tutorial de Adam Wehmann

MATRIZES

Em relação a matrizes, já disporia este valores de forma mais organizada por categoria em linhas e colunas determinado pelo programador por uma variável com x linhas e y colunas.

Exemplo prático de meu aplicativo Matrix:

```
Private Sub cmdMatrix_Click()

    Dim xNome As String
    Dim yNome As String
    Dim pNome As String
    Dim rNome As String
    Dim zNome As String
    Dim cCerta1 As String
    Dim cCerta2 As String
    Dim cCerta3 As String
    Dim cCerta4 As String
    Dim cCerta5 As String
    Dim eErrada1 As String
    Dim eErrada2 As String
    Dim eErrada3 As String
    Dim eErrada4 As String
    Dim eErrada5 As String
    Dim Temp1 As String
    Dim Temp2 As String
    Dim Temp3 As String
    Dim Temp4 As String
    Dim Temp5 As String
    Dim vntArray As Variant
    Dim crtArray As Variant
    Dim erdArray As Variant
    Dim tempArray As Variant
    Dim strBuf As String
    Dim intIndice As Integer

        xNome = DFirst("Nome", "tblSelecao", "Codigo = 1")
        yNome = DFirst("Nome", "tblSelecao", "Codigo = 2")
        pNome = DFirst("Nome", "tblSelecao", "Codigo = 3")
```

```
    rNome = DFirst("Nome", "tblSelecao", "Codigo = 4")
    zNome = DFirst("Nome", "tblSelecao", "Codigo = 5")
    cCerta1 = DFirst("Certas", "tblSelecao", "Codigo = 1")
    cCerta2 = DFirst("Certas", "tblSelecao", "Codigo = 2")
    cCerta3 = DFirst("Certas", "tblSelecao", "Codigo = 3")
    cCerta4 = DFirst("Certas", "tblSelecao", "Codigo = 4")
    cCerta5 = DFirst("Certas", "tblSelecao", "Codigo = 5")
    eErrada1 = DFirst("Erradas", "tblSelecao", "Codigo = 1")
    eErrada2 = DFirst("Erradas", "tblSelecao", "Codigo = 2")
    eErrada3 = DFirst("Erradas", "tblSelecao", "Codigo = 3")
    eErrada4 = DFirst("Erradas", "tblSelecao", "Codigo = 4")
    eErrada5 = DFirst("Erradas", "tblSelecao", "Codigo = 5")
    Temp1 = DFirst("Tempo", "tblSelecao", "Codigo = 1")
    Temp2 = DFirst("Tempo", "tblSelecao", "Codigo = 2")
    Temp3 = DFirst("Tempo", "tblSelecao", "Codigo = 3")
    Temp4 = DFirst("Tempo", "tblSelecao", "Codigo = 4")
    Temp5 = DFirst("Tempo", "tblSelecao", "Codigo = 5")

  vntArray = Array(xNome, yNome, pNome, rNome, zNome)
  crtArray = Array(cCerta1, cCerta2, cCerta3, cCerta4, cCerta5)
  erdArray = Array(eErrada1, eErrada2, eErrada3, eErrada4, eErrada5)
  tempArray = Array(Temp1, Temp2, Temp3, Temp4, Temp5)

'*********************************************************************************
'Looping gerador da distribuição do valores em suas respectivas colunas. Os ::::: são apenas espaçamentos.
For intIndice = LBound(vntArray) To UBound(vntArray)

strBuf = strBuf & "Número: " & intIndice & " = " & Right(vntArray(intIndice), 8) & " ::: " &
crtArray(intIndice) & " ::::: " & erdArray(intIndice) & " :::::::: " & tempArray(intIndice) & vbLf

Next
'*********************************************************************************
'Criando os títulos de colunas
  MsgBox "Desafiantes:    Nome Certas Erradas Tempo Decorrido: " & vbLf & strBuf,
vbInformation, "Matrix - Raciocínio Lógico"

End Sub
```

Fim. Ficamos por aqui. Até mais pessoal !!!