

Expressões Lógicas e Comandos de Decisão

Alexandre Agustini

Adaptado do material das Prof^{as}
Sílvia Moraes
Prof^a Katia Saikoski



Expressões Lógicas



São expressões que resultam em valores lógicos : **verdadeiro** ou **falso**.

Em **VB**,

- a constante **True** representa verdadeiro e
- e a constante **False** o valor falso.

Expressões Lógicas



Para compor as expressões lógicas utiliza-se os operadores:

- **relacionais**: comparam dois valores.
>, <, >=, <=, =, <>
like (para comparação de strings)
- **lógicos**: combinam expressões lógicas.
and, or, not, xor
(e) (ou) (não) (ou exclusivo)

Expressões Lógicas



Operador relacional - Like

- Exemplos:

Dim achou As Boolean

achou = "Ana Julia" like "Ana*" ' true

...

achou = "Ana Maria" like "Ana*" ' true

...

achou = "Katia" like "Kat?a" ' true

Expressões Lógicas



Operadores Lógicos

- **not A** : inverte o valor lógico de A
- **A and B**: para resultar **true** exige que as expressões A e B sejam verdadeiras ao mesmo tempo.
- **A or B** : para resultar **false** exige que as expressões A e B sejam falsas ao mesmo tempo.
- **A xor B**: para resultar **true** uma das expressões deve ser verdadeira, mas não as duas

Expressões Lógicas



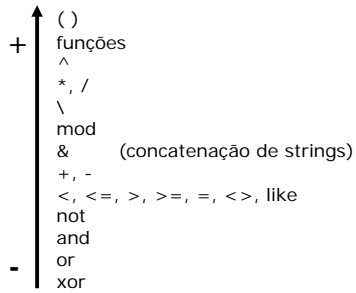
Considerando que A e B são duas expressões lógicas quaisquer.

A	B	not A	A and B	A or B	A xor B
F	F	T	F	F	F
F	T	T	F	T	T
T	F	F	F	T	T
T	T	F	T	T	F

? Qual a **tabela verdade** de:

- not (A and B)? not (A or B)?

Prioridade dos Operadores - VB



Expressões Lógicas

As expressões abaixo são válidas?

Double num, val As integer
 Double op As string

num = 3
 val = 5
 op = "c"

- num < val + 1 and num <= val ^ 2 - 7
- op < "E" or op <= "e" and op != "c"
- num > 1 and num < sqrt(64) and not(num < > 2)
- op == "C" or op == "c"

Comandos de decisão

- Permite criar alternativas para o fluxo de execução de um programa
- VB prevê 2 tipos de comandos de decisão:
 - **If ... Then ... [Else ...] End If**
 - **Select Case...**

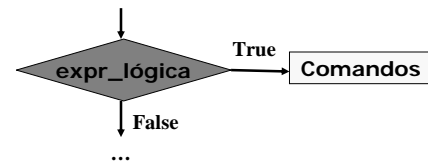
Comando if

A forma do comando **if** é a seguinte:

```
If expr_lógica Then  

    comandos  

End If
```



Comando if-else

Outra forma de usar o **if** é com o **else**. Esse tipo de construção permite a definição de ações alternativas.

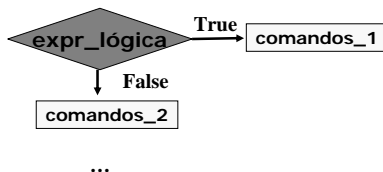
```
If expr_lógica Then  

    comandos_1  

Else  

    comandos_2  

End If
```



Exemplo Comando if

...

```
nota1 = Cdbl(Text1.Text)  

nota2 = Cdbl(Text2.Text)
```

```
If nota1 = -1 or nota2 = -1 Then `parentesis???  

    Label5.Caption = "Aluno pode realizar PS"  

End If
```

Exemplo Comando if

```
...  
  
delta = b^2 - 4 * a * c  
  
If delta < 0 Then  
    Label4.Caption = "equação não tem raízes reais"  
Else  
    ' calcular e apresentar as raízes  
    ' ...  
End If
```

Exemplo Comando if "encadeado"

```
...  
delta = b^2 - 4 * a * c  
  
If delta < 0 Then  
    Label4.Caption = "equação não tem raízes reais"  
Else  
    If (delta = 0 ) Then  
        Label4.Caption = "Uma única raiz!"  
        ' calcular e apresentar a raiz  
    Else  
        Label4.Caption = "Duas raízes reais:"  
        ' calcular e apresentar as raízes  
    End If  
End If
```

Exercícios:

- 1) Faça um algoritmo que encontra o maior de 3 valores.
- 2) Altere o algoritmo anterior para que ele encontre o maior de 4 valores.
- 3) Como seriam os algoritmos anteriores se tivéssemos disponível uma função $max(a,b)$?
- 4) Escreva um algoritmo que a partir da idade de um nadador classifica-o em uma das seguintes categorias:

```
5 a 7 anos: Infantil A  
8 a 10 anos : Infantil B  
11 a 13 anos : Juvenil A  
14 a 17 anos: Juvenil B  
maiores de 17 anos: Adulto
```

Comando *Select Case*

- Este é um comando alternativo ao (encadeamento de) comando(s) "**If Then Else**", que possibilita tornar o código mais simples (legível). Exemplo:

```
Select Case estadoCivil  
Case 1  
    Label1.Caption = "Solteiro"  
Case 2  
    Label1.Caption = "Casado"  
Case Else  
    Label2.Caption = "Estado civil não informado"  
End Select
```

Como fica este código com comandos If?

Comando *Select Case* *Formato*

```
Select Case expressão  
Case listaDeValores_1  
    comandos_1  
[ Case listaDeValores_2  
    comandos_2 ]  
...  
[ Case Else  
    "comandos_else" ]  
End Select
```

Exercício:

- 1) (Re)escreva um algoritmo (utilizando a instrução **Select Case**) que a partir da idade de um nadador classifica-o em uma das seguintes categorias:

```
5 a 7 anos: Infantil A  
8 a 10 anos : Infantil B  
11 a 13 anos : Juvenil A  
14 a 17 anos: Juvenil B  
maiores de 17 anos: Adulto
```