

UNIVERSIDADE FEDERAL DA FRONTEIRA SUL

Programação I – Trabalho I

Régis Thiago Feyh – <registhiagofeyh@gmail.com>

MANUAL DA LINGUAGEM VISH

Declaração e atribuição de valor a variáveis:

Sintaxe:

Para declarar sem atribuir valor: **dim** <nome_da_variavel>

Para declarar com atribuição de valor: **dim** <nome_da_variavel> = <valor_numerico>

Você pode declarar várias variáveis inicializando os valores delas ou não. Se a declaração não contiver uma atribuição, ela será iniciada com o valor 0. Para realizar uma declaração múltipla de variáveis, basta utilizar a seguinte sintaxe:

dim <nome_var1>,<nome_var2>=**50**,<nome_varN>

Mas essa declaração não é uma regra da linguagem, você pode criar uma nova variável sem utilizar a sintaxe formal acima. O sistema criará uma variável automaticamente se houver uma atribuição a uma variável não existente no programa. Por exemplo:

abc = 20+2

Se a variável **abc** não existir no sistema, ela será criada com o valor 22.

ATENÇÃO: A alocação automática de variáveis não declaradas só é feita nos comandos de atribuição de valores e entrada de dados.

ATENÇÃO²: Só é possível executar operações binárias, ou seja, com dois operandos. Por exemplo: **a+b** ou **c*2** ou **20/4** ou **1%c**. NÃO FAÇA **a+b+c**, ele não vai saber o que é isso, coitado.

TABELA DE OPERADORES ARITMÉTICOS (assuma X e Y como dois valores quaisquer)	
X+Y	Soma de X com Y
X-Y	Subtração de X e Y
X*Y	X multiplicado por Y
X/Y	Divisão de X por Y
X%Y	Resto da divisão de X por Y

Comando para desvio de fluxo:

Sintaxe:

if <condição #1> **then**

<seu código se condição verdadeira>

elseif <condição #2> **then**

<seu código se a condição #2 for verdadeira>

elseif <condição #n> **then**

<seu código se a condição #n for verdadeira>

else

<seu código se nenhuma das condições acima for verdadeira>

end if

Note que você pode aninhar quantos condicionais quiser. No entanto, as condições só aceitam dois operadores. Por exemplo: **a>b** ou **1<=b** ou **2>=1**. Não é possível realizar mais de uma operação nas condições, por exemplo: **a+b>1** ou **a>b&& c>b** como na linguagem C. O

interpretador só entenderá um condicional por vez. Os parênteses na condição são opcionais.

TABELA DE OPERADORES LÓGICOS	
>	Maior que
<	Menor que
==	Igual a
<>	Diferente de
>=	Maior ou igual a
<=	Menor ou igual a

Entrada de dados:

Sintaxe:

scan <nome_da_variavel>

Como já explicado no item anterior, se a variável especificada para receber o valor do teclado não existir, ela será alocada. Só é possível receber valores **numéricos** do teclado que serão convertidos para double pelo interpretador. Se uma string for digitada no momento do scan, o sistema irá procurar alguma variável com o nome no programa. Por exemplo:

```
dim a = 2
scan b
```

Se o usuário informar na entrada o valor **a**, então **b** receberá o valor de **a**, no caso, **2.0**. Se não for encontrada nenhuma variável com o nome da string informada o programa irá mostrar um erro e a interpretação será interrompida.

Impressão de dados na tela:

Sintaxe:

Para imprimir o valor de uma variável: **print** <nome_da_variavel>

Para imprimir uma String: **print** "String qualquer"

Para inserir uma nova linha: **print** \n

Para concatenar qualquer uma das impressões acima basta utilizar o operador &, por

exemplo:

```
dim a = 5.2
print "Valor de a = "&a&\n
```

Irá imprimir na tela:

Valor de a = 5.2

Laço condicional:

Sintaxe:

```
while <condição>
    <seu código enquanto a condição for verdadeira>
break
wend
```

O laço condicional permite o aninhamento de vários laços um dentro do outro e o aninhamento de desvios de fluxo (if) também. Da mesma maneira que os desvios condicionais, os laços condicionais também só identificam operações entre dois operadores, sem a opção de compor

várias operações no condicional. O comando **break** interrompe a execução do **while** independente da condição, sorte que você já sabe programar. Os parênteses na condição são opcionais.

Exemplo de laço condicional:

```
dim a
while a<5
  print a&\n
  a = a+1
  if a > 3 then
    break
  end if
wend
```

Irá imprimir na tela:

0.0
1.0
2.0
3.0

RESUMO DAS FUNCIONALIDADES, entenda o conjunto como um programa qualquer.	
Declaração de variáveis	<pre>dim a,b=2,c,d dim e dim j=a/b</pre>
Condicional	<pre>if (j==2) then print d elseif b>=2 then scan d print "Valor de d*2 = "&d*2 end if</pre>
Laço condicional	<pre>c=0 while c<=5 print "Abacate"&\n c=c+1 if c>3 then break end if wend</pre>
Impressão na tela	<pre>print "Valor de c="&c&\n</pre>
Entrada de dados	<pre>scan abc</pre>
ENTENDEU FERA?	

Observações adicionais e que se você não ler, provavelmente vai fazer me#\$@!:

- Os comandos não são case sensitive. Você pode escrever **if**, **ELSE** ou como um emo **EnD iF** que o interpretador irá corrigir sua falta de padrão e vai entender.
- Se você veio direto pra cá e não leu ali em cima, o interpretador só entende operações com dois operadores, nada de composição de operações. Não venha inventar de colocar $a+b \geq c+2$. Ele não entende isso. Ele só entenderá coisas do tipo $a+b$ ou $20*4$ ou $h>c$ ou $j<>d$.
- Somente um comando por linha, não tem como dividir o mesmo comando em várias linhas ou escrever vários comandos em uma linha só.