UNIDADE IV

Linguagem de Programação Portugol

NDDigital

ALGORITMO – PORTUGOL

LAGES
(2014)

Sumário

1 UNIDADE 4 – VETORES E MATRIZES 3
1.1 O que é um vetor? 3
1.2 Exemplo Real 3
1.3 Como declarar e inicializar um vetor? 5
1.3.1 Inserindo valores de um vetor 5
1.3.2 Percorrendo um vetor 6
1.3.3 Acessando os valores de um vetor 6
2 MATRIZ 7
1.2.1 Inserindo valores de uma matriz 7
1.2.2 Percorrendo uma matriz 8
2.13 Exercicios: Fixação 9
Desafio 9

1 UNIDADE 4 – VETORES E MATRIZES

Nessa unidade daremos continuidades aos comandos da linguagem de programação do visualg, portugol. Como vetores, vetores multidimensionais ou matrizes.

1.1 O QUE É UM VETOR?

Considere um programa de computador que realizará cálculos matemáticos com os preços dos produtos de um supermercado. Por exemplo, esse programa deverá calcular a média dos preços ou encontrar o produto mais barato.

Para manipular os preços dos produtos, dentro de um programa, esses valores devem ser armazenados em variáveis. Como a figura abaixo:

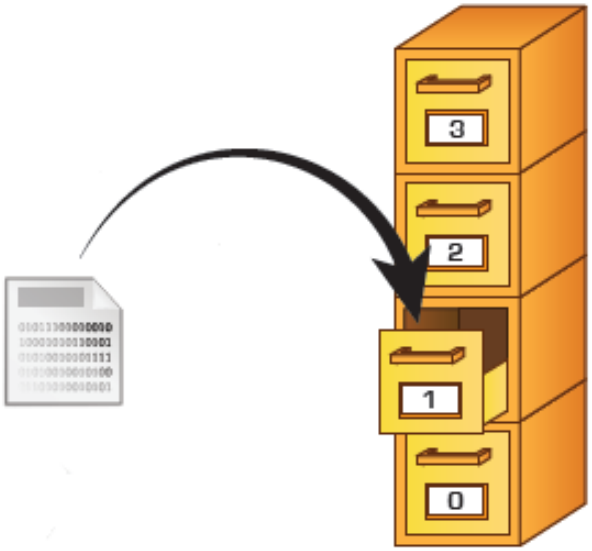
```
var

preco1: real
preco2: real
preco3: real
```

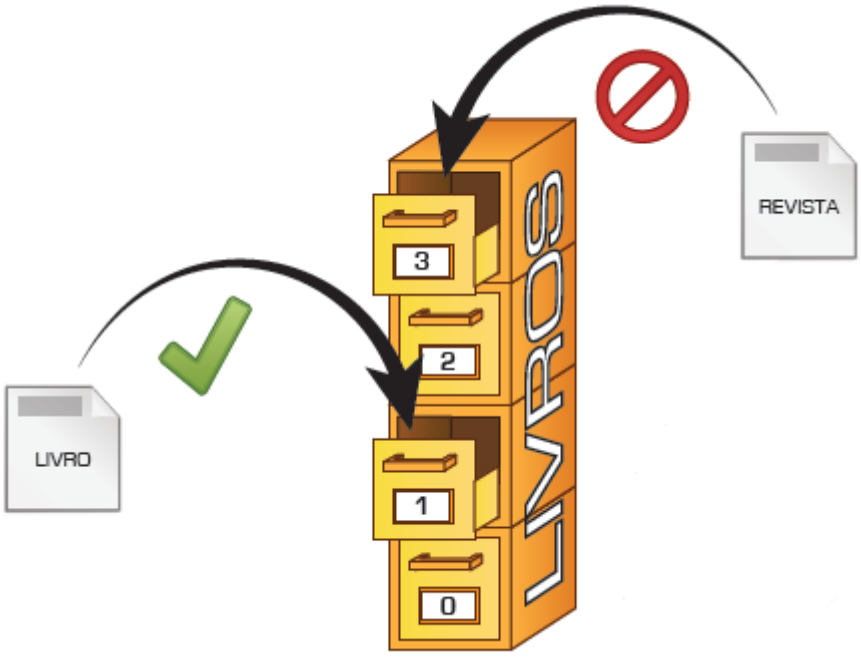
Como uma variável do tipo *real* armazena somente um valor de cada vez, seria necessário uma variável para cada produto. Considerando a existência de uma grande quantidade de produtos, essa abordagem será pouco prática. Nesses casos, podemos utilizar os chamados vetores.

1.2 EXEMPLO REAL

Um vetor é uma estrutura de dados utilizada para armazenar uma coleção de itens. Cada item é identificado através de um índice. Podemos imaginar um vetor como sendo um armário com um determinado número de gavetas e cada gaveta possui um rótulo com um número de identificação. Como a figura abaixo mostra:



Quando criamos um vetor, devemos informar qual o tipo de dado pretendemos armazenar em cada posição. Na analogia com armário, seria como se tivéssemos que definir o que o é permitido guardar em cada gaveta. Por exemplo, se definirmos que um armário deve guardar livros, então somente livros podem ser armazenados nas gavetas desse armário. Não poderemos guardar revistas ou jornais.



1.3 COMO DECLARAR E INICIALIZAR UM VETOR?

Para utilizarmos um vetor, devemos criar uma variável para guardar a referência desse vetor. A declaração dessa variável é semelhante à declaração das variáveis que vimos até agora. Lembre-se que sempre devemos inicializar as variáveis para não ocorrer um erro de compilação. Portanto, vamos inicializar o nosso vetor:

```
var

preco : vetor[1..10] de real

inicio
```

1.3.1 Inserindo valores de um vetor

Esta estrutura repete uma sequência de comandos enquanto uma determinada condição (especificada através de uma expressão lógica) for satisfeita.

Existem diversas formas de inserirmos valores em um vetor. A forma mais comum é a seguinte:

```
 6 var
 7
 8 vet1 : vetor[1..10] de real
 9 vet2 : vetor[1..10] de caractere
10 vet3 : vetor[1..10] de inteiro
11 vet4 : vetor[1..10] de logico
12
13 inicio
14
15 vet1[1] <- 1.5
16 vet2[1] <- "Vetor"
17 vet3[1] <- 2
18 vet4[1] <- VERDADEIRO
19
20 fimalgoritmo
```

Na linha 8,9,10,11 declaramos e inicializamos um vetor de cada tipo real, inteiro, caractere e lógico. Na linha 15, 16, 17, 18 estão sendo inseridos dados respectivamente a cada tipo dos vetores.

A numeração dos índices de um vetor no portugol já é declarado na figura acima podemos ver que os índices vão de 1 a 10. O intervalo dos índices é flexível no portugol, diferente de outras linguagens.

1.3.2 Percorrendo um vetor

Muito importante é poder acessar o vetor por inteiro, e como facilidade, economizando linha. Para isso é fácil, usaremos um comando que já vimos nas outras unidades.

O para faça, com a imagem abaixo ilustra:

```
16 para indice de 1 ate 10 faca
17 leia(vet[indice])
18 fimpara
```

A variável índice está declarada global. Lembrando que para percorrer o vetor por inteiro utilizamos um laço de repetição e o seu índice deve iniciar e terminar conforme o intervalo do índices do vetor.

1.3.3 Acessando os valores de um vetor

Para acessarmos o valor armazenado em uma das posições de um vetor, basta conhecermos o índice de tal posição. Veja o exemplo abaixo:

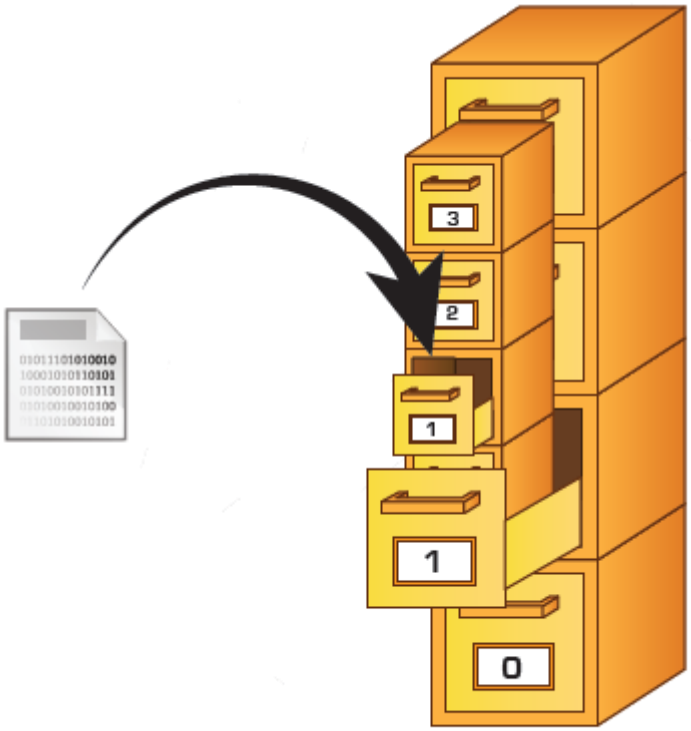
```
16 vet1[1] <- 0.1
17 vet1[2] <- 0.2
18 vet1[3] <- 0.3
19 vet1[4] <- 0.4
20 vet1[5] <- 0.5
21
22 escreva(vet1[1])
23 escreva(vet1[2])
24 escreva(vet1[3])
25 escreva(vet1[4])
26 escreva(vet1[5])
```

Ou percorrer o vetor com um laço de repetição como o exemplo abaixo:

```
29 para indice de 1 ate 10 faca
30 escreva(vet1[indice])
31 fimpara
```

2 MATRIZ

Até agora trabalhamos com vetor de uma dimensão. Porém, tanto em outras linguagens como no portugol, podemos criar vetores com mais de uma dimensão (vetores multidimensionais). Isso nos permite trabalhar com vetores para representar tabelas, matrizes ou até um tabuleiro de batalha naval. Voltando à analogia que fizemos com um armário cheio de gavetas, seria como se pudéssemos guardar dentro da gaveta de um armário um outro armário com gavetas. Veja a figura abaixo:



A declaração de Matriz é muito semelhante à declaração e inicialização de um vetor simples.

```
var

matriz: vetor[1..3,1..3] de real
```

1.2.1 Inserindo valores de uma matriz

Em uma matriz,podemos criar vetores de diferentes tamanhos em cada posição. Assim como nos vetores unidimensionais, para inserir ou acessar valores de um array de

arrays, devemos utilizar os índices de cada posição. Podemos pensar nos índices como um esquema de coordenadas. Por exemplo, se quiséssemos representar um gráfico no sistema cartesiano de eixos x y através de uma matriz, a coordenada de cada ponto do gráfico seria equivalente ao par de índices da nossa matriz (supondo que no gráfico seja permitido apenas coordenadas inteiras). Exemplo abaixo:

```
matriz[1,1] <- 0
matriz[1,2] <- 0
matriz[1,3] <- 0
matriz[2,1] <- 0
matriz[2,2] <- 0
matriz[2,3] <- 0
matriz[3,1] <- 0
matriz[3,2] <- 0
matriz[3,3] <- 0
```

1.2.2 Percorrendo uma matriz

Para percorrer uma matriz, utilizaremos novamente as instruções de repetição enquanto e para faça. Porém, como estamos trabalhando com vetores multidimensionais com mais de uma dimensão, teremos uma ou mais laços encadeados. Como o exemplo ilustra:

```
para linha de 1 ate 3 faca
   para coluna de 1 ate 3 faca
         matriz[linha,coluna]  <- 0
   fimpara
fimpara
```

2.13 EXERCÍCIOS: FIXAÇÃO

1) Faça um programa que exibe o conteúdo de um vetor tamanho 10 de forma invertida.

2) Faça um programa que leia um vetor de 10 posições e crie um segundo vetor substituindo os valores negativos por 1.

3) Faça um programa que leia uma matriz **mat** 3 x 4 de inteiros, substitua seus elementos negativos por 0 e imprima a matriz **mat** original e a modificada.

4) Escreva um algoritmo que leia e mostre um vetor de 20 elementos inteiros. A seguir, conte quantos valores pares existem no vetor.

5) . Faça um algoritmo que leia um código numérico inteiro e um vetor de 50 posições de números reais. Se o código for zero, termine o algoritmo. Se o código for 1, mostre o vetor na ordem direta. Se o código for 2, mostre o vetor na ordem inversa.

6) Faça um espelho de classe. Uma matriz com 14 elementos. Conforme cada lugar dos companheiros de sala.

Desafio

Faça um algoritmo que leia dois vetores (A e B) de 50 posições de números inteiros. O algoritmo deve, então, subtrair o primeiro elemento de A do último de B, acumulando o valor, subtrair o segundo elemento de A do penúltimo de B, acumulando o valor, e assim por diante. Mostre o resultado da soma final.

Bons estudos!