

Vetores



PROFESSOR: DIEGO RICARDO KROHL
diego.krohl@ifc-videira.edu.br

Estruturas de dados homogênea



- Até agora, trabalhamos variáveis com tipos de dados real, inteiro, lógico e caractere.
- Essas variáveis são chamadas de tipo simples, pois armazenam apenas um valor.
- Variáveis do tipo simples ocupam apenas um lugar na memória e não podem armazenar mais que um valor.

```
nota: inteiro
```

```
nota<-10
```

```
nota<-9
```

```
nota<-8
```

Estruturas de dados homogênea



- Como fazíamos para calcular a média de um aluno com cinco notas?

```
var
    n1, n2, n3, n4, n5, media: real
inicio
    leia(n1)
    leia(n2)
    ...
    media<- (n1+n2+n3+n4+n5) / 5
    ...
fimalgoritmo
```

Vetor



- Então, o vetor é um arranjo de elementos (dados) armazenados na memória do computador.
- Os dados de um vetor são organizados de forma eficiente um após o outro, usando um mesmo identificador, cujas regras são as mesmas para criar um identificador de uma variável simples, que possui capacidade de guardar somente um valor.

Características de um vetor



- Contém vários valores (número definido).
- Todos os valores são do mesmo tipo de dado (homogêneo).
- Possui um único nome (identificador da variável).
- Cada valor do conjunto é acessível de forma independente, de acordo com seu índice.
- Todos os seus elementos, ou dados armazenados, são igualmente acessíveis a qualquer momento do processamento (acesso aleatório).

Características de um vetor



- Quando declaramos o vetor nota, a variável teve um espaço de memória reservado para ela:

Nota: vetor [1..5] de real				
N1	N2	N3	N4	N5

Declaração de um vetor



- Para declarar uma variável vetorial, usamos a seguinte estrutura no Visualg:

```
<nome_variavel>: vetor [<valor_inicial>..
```

- Os vetores, que podem armazenar vários elementos, normalmente tem sua utilização em conjunto com uma estrutura de repetição.

Exemplo



- Vamos criar um algoritmo para solicitar e armazenar o nome de 4 alunos, e em seguida, exibir os nomes na sequência inversa de que foram digitados.

Exemplo



algoritmo "Nomes"

var

//Declaração do vetor

nomes: vetor [0..3] de caractere

cont: inteiro

inicio

//Estrutura de repetição para solicitar os nomes

para cont de 0 ate 3 faça

 escreva ("Informe o nome", cont+1, ": ")

 leia (nomes[cont])

fimpara

//Estrutura de repetição para exibir os nomes na ordem inversa

para cont de 0 ate 3 faça

 escreval ("Nome", (4-cont), ": ", nomes[3-cont])

fimpara

fimalgoritmo

Matrizes



- Vetores também podem ser considerados uma matriz, só que apenas de uma dimensão, linear ou vertical.
- A matriz possui duas dimensões, com linhas e colunas, como aprendemos em matemática.

Matrizes



- Uma matriz, representada por uma tabela, tem as seguintes posições:

Matriz M: 3x3		
M[0,0]	M[0,1]	M[0,2]
M[1,0]	M[1,1]	M[1,2]
M[2,0]	M[2,1]	M[2,2]

- A primeira referência entre os colchetes é a linha, a segunda referência é a coluna.

Declaração de uma matriz



- Para declarar uma variável matriz, usamos a seguinte estrutura no Visualg:

```
<nome_vari>: vetor [<lin_ini>..
```

- As matrizes normalmente tem sua utilização em conjunto com duas estruturas de repetição, uma para as linhas e outra para as colunas.
- Pode-se utilizar um vetor no lugar de uma matriz, mas como veremos no exemplo a seguir, os dados poderiam ficar desestruturados.

Declaração de uma matriz



- Vamos criar um algoritmo para solicitar o peso em kg e a altura em metros de três pessoas, calcular o IMC de cada uma e em seguida, vamos apresentar os dados organizados de forma que em cada linha apareça as informações de peso, altura e IMC de cada pessoa. O cálculo do IMC é dado por: $(\text{peso}/(\text{altura}*\text{altura}))$.

Resolução

algoritmo "IMCs"

var

 // Declaração do vetor

 dados: vetor [0..2, 0..2] de real

 linha, coluna: inteiro

 imc: real

inicio

 // Estrutura de repetição para cada linha

 para linha de 0 ate 2 faca

 // Solicita peso

 escreva("Informe o peso em kg da pessoa ", (linha+1), ": ")

 leia(dados[linha,0])

 // Solicita altura

 escreva("Informe altura em metros da pessoa ", (linha+1), ": ")

 leia(dados[linha,1])

 // Calcula o IMC

$imc \leftarrow (dados[linha,0]) / (dados[linha,1]^2)$

 // Armazena IMC na terceira coluna da linha

 dados[linha,2] ← imc

 fimpara

 // Exibe os dados estruturados

 escreval("-----")

 // repetição para cada linha

 para linha de 0 ate 2 faca

 escreva("Pessoa ", (linha+1), ": ")

 // repetição para coluna

 para coluna de 0 ate 2 faca

 escreva(dados[linha,coluna])

 escreva(" | ")

 fimpara

 escreval("")

 escreval("-----")

 fimpara

fimalgoritmo