

Algoritmos I

Variáveis Compostas Homogêneas

Prof.^a DSc. Vanessa de Oliveira Campos

Variáveis Compostas Homogêneas

- Imagine, por exemplo, como faríamos para construir um algoritmo, que lesse os nomes de 500 pessoas e imprimisse um relatório destes mesmos nomes, mas ordenados alfabeticamente.
- Não seria uma tarefa simples, pois teríamos que definir 500 variáveis do tipo literal, como é mostrado a seguir.



Variáveis Compostas Homogêneas

Algoritmo *"Inviável"*

var

nome1, nome2, nome3, nome4, nome5, nome6, ■ ■ ■,
nome497, nome498, nome499, nome500: caracter

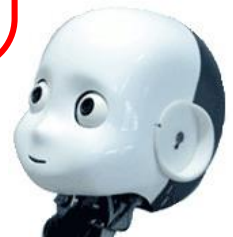
Início

leia (nome1, nome2, nome3, ..., nome500)

■ ■ ■

Fimalgoritmo

Considere o tamanho do algoritmo, e o trabalho braçal necessário para construí-lo.



Variáveis Compostas Homogêneas

- Para resolver problemas como este, e outros, existem as **variáveis indexadas**.
- A declaração de uma variável indexada corresponde à declaração de *várias variáveis* cujo *identificador* difere apenas por um **índice**.
- O índice corresponde a um *valor numérico*, normalmente começando por 1.



Variáveis Compostas Homogêneas

- Cada variável indexada pode receber valores no decorrer do algoritmo como se fosse uma variável comum.
- Uma variável indexada é constituída por dados do **mesmo tipo** (*homogêneos*) e são agrupados continuamente na memória e acessados por sua **posição** (*índice* - número inteiro).



Variável Indexada Unidimensional: Vetor

- **Vetor** é uma variável composta homogênea **unidimensional** formada por uma *sequência de variáveis*, todas do *mesmo tipo*, com o mesmo identificador e alocado sequencialmente na memória.
- A distinção entre as variáveis é feita usando índice.



Variável Indexada Unidimensional: Vetor

A **sintaxe** para declaração é:

< identificador > : vetor [**<intervalo>**] de *<tipo-de-dado>*

onde:

intervalo => [VI..VF]

VI: Valor inicial do índice

VF: Valor final do índice



Variável Indexada Unidimensional: Vetor

Exemplos:

var

idade: vetor [1..5] de inteiro

nome: vetor [1..5] de caracter

- A declaração acima corresponde à declaração de 10 variáveis:

nome[1], nome[2], nome[3], nome[4], nome[5]

idade[1], idade[2], idade[3], idade[4], idade[5]

Note que isto substitui uma declaração do tipo:

var

idade1, idade2, idade3, idade4, idade5: inteiro

nome1, nome2, nome3, nome4, nome5: caracter



Variável Indexada Unidimensional: Vetor

Para se atribuir um valor a um elemento do vetor devemos utilizar o seguinte padrão:

< identificador>[<posição>] ← <valor>

Exemplos:

nome[1] ← “João da Silva”

idade[1] ← 35

nome[3] ← “Maria Aparecida”

idade[3] ← idade[1]

i ← 5

idade[i] ← 45

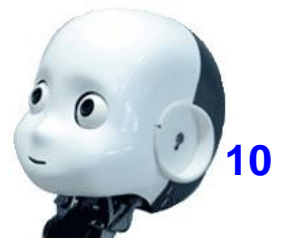


Variável Indexada Unidimensional: Vetor

Algoritmo exemplo:

Leia um vetor numérico de 6 posições.

Em seguida, conte quantos valores são negativos e mostre esta informação.



Variável Indexada Unidimensional: Vetor

Algoritmo "vetores"

var

NUMERO: vetor [1..6] de real

i, conta_neg: inteiro

Inicio

conta_neg \leftarrow 0

para i de 1 ate 6 faca

leia (NUMERO[i])

se NUMERO[i] < 0 entao

conta_neg \leftarrow conta_neg + 1

fimse

fimpara

escreva ("Total de números negativos: ", conta_neg)

Fimalgoritmo



Vetor em Pascal

Em Pascal, um vetor é declarado seguindo o formato:

identificador : array [**limite_inferior** . . **limite_superior**] of **tipo**;

- **identificador** é nome do vetor.
- **limite_inferior** é o índice do primeiro elemento do vetor. Em geral será o valor 1, mas pode assumir qualquer valor positivo ou negativo.
- **limite_superior** é o índice do último elemento do vetor. Deve ser maior ou igual que o valor do **limite_inferior**.
- **tipo** é o tipo de todos os elementos do vetor (integer, real, char, string, boolean).



Vetor em Pascal

```
var  
notas: array [1..6] of real;  
quantidade : array [1..200] of integer;  
gabarito : array [1..80] of char;  
questoesVF : array [1..10] of boolean;  
Fx : array [-5..5] of real;
```

índices

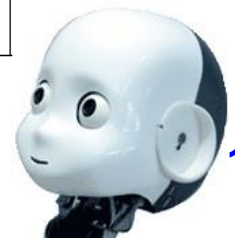
1	2	3	4	5	6

Notas

índices

-5	-4	-3	-2	-1	0	1	2	3	4	5

Fx



Vetor em Pascal

```
var
Nota: array [1..10] of real;
ind : integer;
begin
Nota[2] := 7.4;
Nota[4] := 5.6;
Nota[7] := Nota[4] + 0.5;
ind := 5;
Nota[ind] := 9.0;
```

	índices									
	1	2	3	4	5	6	7	8	9	10
Nota										



Preenchendo valores em um vetor

- Preencher um vetor significa atribuir valores a todas as suas posições. Assim, deve-se implementar um mecanismo que controle o valor do índice. Exemplo:

```
var  
Nota: array [1..10] of real;  
ind : integer;  
for ind := 1 to 10 do  
    readln(Nota[ind])
```

- O acesso de um vetor em uma posição específica tem o mesmo comportamento que uma variável simples.



Mostrando os conteúdos de um vetor

- Mostrar todos os valores contido em um vetor exige o uso de um índice para percorrer cada elemento.
- Assim, deve-se implementar um mecanismo que controle o valor do índice

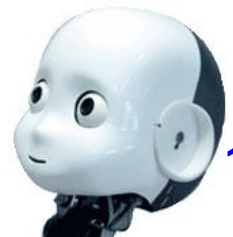
```
var  
Nota: array [1..10] of real;  
ind : integer;  
for ind := 1 to 10 do  
    writeln(Nota[ind]);
```

- O conteúdo elemento do vetor também pode ser atribuído a outra variável de um mesmo tipo.



Exercícios

1. Faça um programa que leia a nota de 6 alunos, calcule a média dessa turma de alunos e mostre para o usuário a quantidade de alunos que estão acima da média da turma e a média dessa turma.



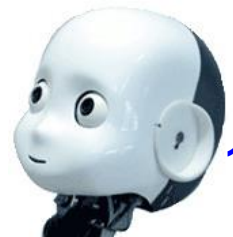
Exercícios

2. Escreva um algoritmo que leia e mostre um vetor de 20 números. Depois mostre todos os valores ímpares contidos no vetor. Em seguida, mostre os valores pares existentes no vetor.



Exercícios

3. Fazer um algoritmo para ler uma lista de números positivos e escrevê-los na ordem inversa da entrada. A entrada de um número negativo é utilizada como **flag**, que marca o fim da entrada dos números válidos. A quantidade máxima de números válidos a serem digitados é 500.



Exercícios

4. Fazer um algoritmo para ler uma lista de 50 números e dizer qual valor existente na lista que mais se aproxima do valor médio (a média) dos elementos da própria lista.

