

Lógica de Programação Orientada a Objetos

Aula 03

- Vetores
- Matrizes



Vetores

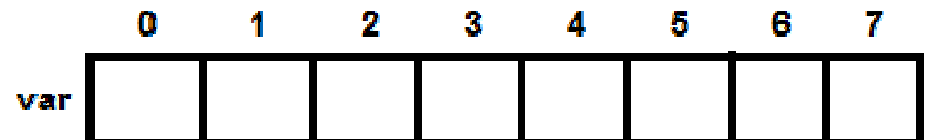
www.3way.com.br



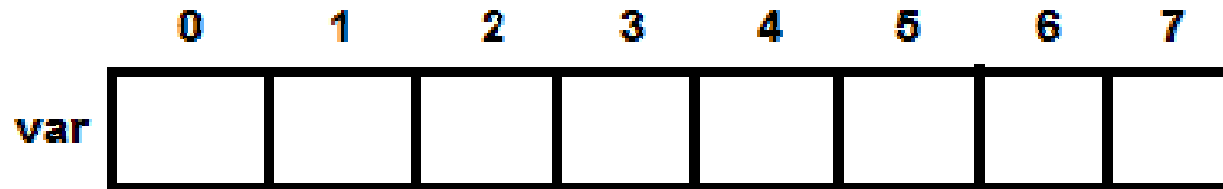
Vetores

- São variáveis que armazenam um conjunto de valores do mesmo tipo de forma seqüencial na memória.
- Coleções homogêneas de dados.
- Seu tamanho é fixo e não pode ser alterado.
- Também conhecidos como arrays.

Um vetor pode ser representado da seguinte forma:



Vetores



- Cada posição do vetor é indicada por um índice.
- O espaço alocado na memória para o vetor, depende do tipo de valor que ele irá armazenar.
- A primeira posição de um vetor **sempre** terá índice igual a 0.



Vetores

Declarando um vetor

- Podemos tratar um vetor, ou um array, como uma matriz unidimensional.
- O tamanho de um vetor é fixo e definido em sua declaração.

Sintaxe:

nome_vetor : **vetor**[posições] **de** tipo_dados_armazenados



Vetores

Declarando um vetor

Exemplos:

arr : vetor[0..99] de inteiro

x : vetor[0..49] de real

results : vetor[0..9] de logico

clientes : vetor[0..999] de caractere



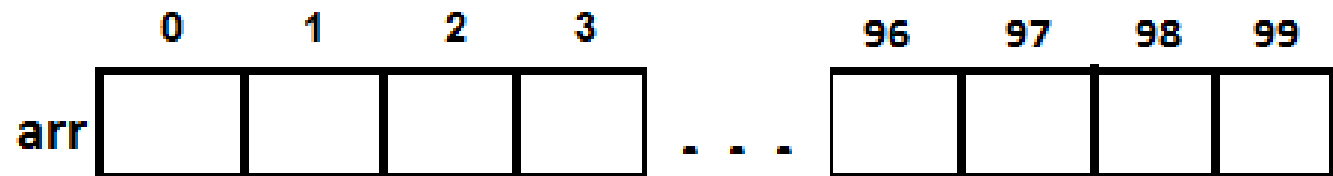
Vetores

Acessando um elemento do vetor

- Para acessarmos uma determinada posição do vetor, devemos indicar entre os colchetes o índice que a representa.
- Um vetor **sempre** terá sua primeira posição no índice **0** e sua última posição no índice **(tamanho – 1)**.

Exemplo:

arr : 100 posições.



Vetores

Acessando um elemento do vetor

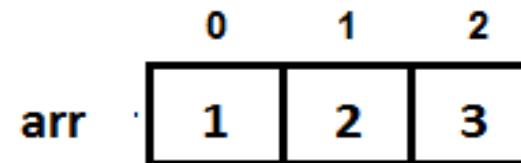
Exemplo:

arr : vetor[0..2] de inteiro

arr[0] \leftarrow 1

arr[1] \leftarrow 2

arr[2] \leftarrow 3



Vetores

Acessando um elemento do vetor

Exemplo1:

algoritmo “PopulaVetorPequeno”

var

vetorInteiros : **vetor[0..4]** de inteiro

inicio

vetorInteiros[0] \leftarrow 1



Vetores

Acessando um elemento do vetor

Exemplo1:

`vetorInteiros[1] ← 2`

`vetorInteiros[2] ← 3`

`vetorInteiros[3] ← 4`

`vetorInteiros[4] ← 5`

[fimalgoritmo](#)



Vetores

Acessando um elemento do vetor

Exemplo2:

algoritmo "PopulaVetorGrande"

var

vetorInteiros : **vetor**[0..*N*] de **inteiro**

indice : **inteiro**

valor : **inteiro**



Vetores

Acessando um elemento do vetor

Exemplo2:

inicio

leia (N)

valor $\leftarrow 1$

para indice **de** 0 **ate** N **faca**

vetorInteiros[indice] \leftarrow valor



Vetores

Acessando um elemento do vetor

Exemplo2:

$valor \leftarrow valor + 1$

fimpara

[fimalgoritmo](#)

Resultado:

	0	1	2	3		N-4	N-3	N-2	N-1
vetorInteiros	1	2	3	4	...	N-3	N-2	N-1	N



Vetores



Run Jeliot - vetores

www.3way.com.br



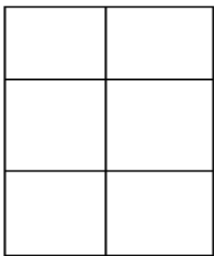
Matrizes

www.3way.com.br



Matrizes

- Matrizes são vetores unidimensionais.
- É um vetor capaz de armazenar outros vetores, ou seja, em cada posição do vetor temos outro vetor armazenado.
- Didaticamente tratamos um vetor de 2 dimensões como uma matriz convencional.



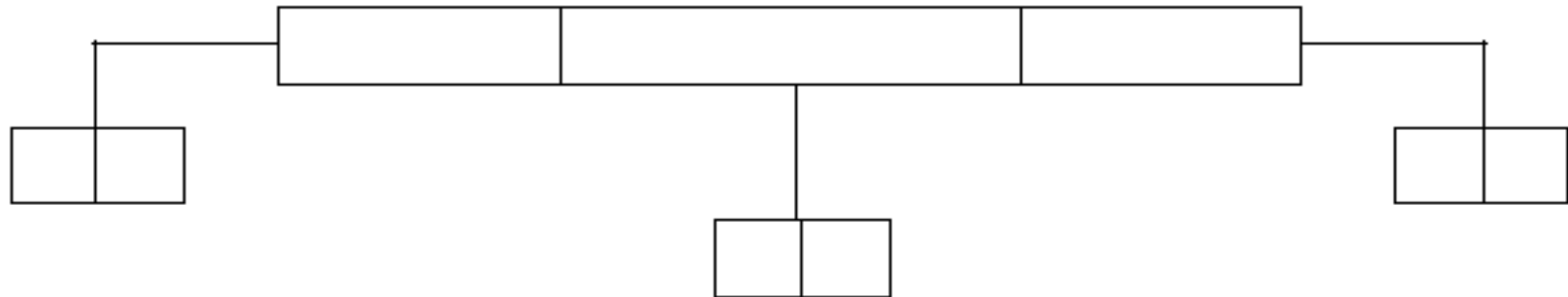
3x2

Mas pela definição,
sua disposição na memória não é essa.



Matrizes

Pela definição, uma matriz **3 x 2** seria algo como:



Assim temos 2 dimensões onde,

A primeira dimensão do vetor, com 3 posições, armazena endereços de memória que estão referenciando outros vetores com 2 posições cada.

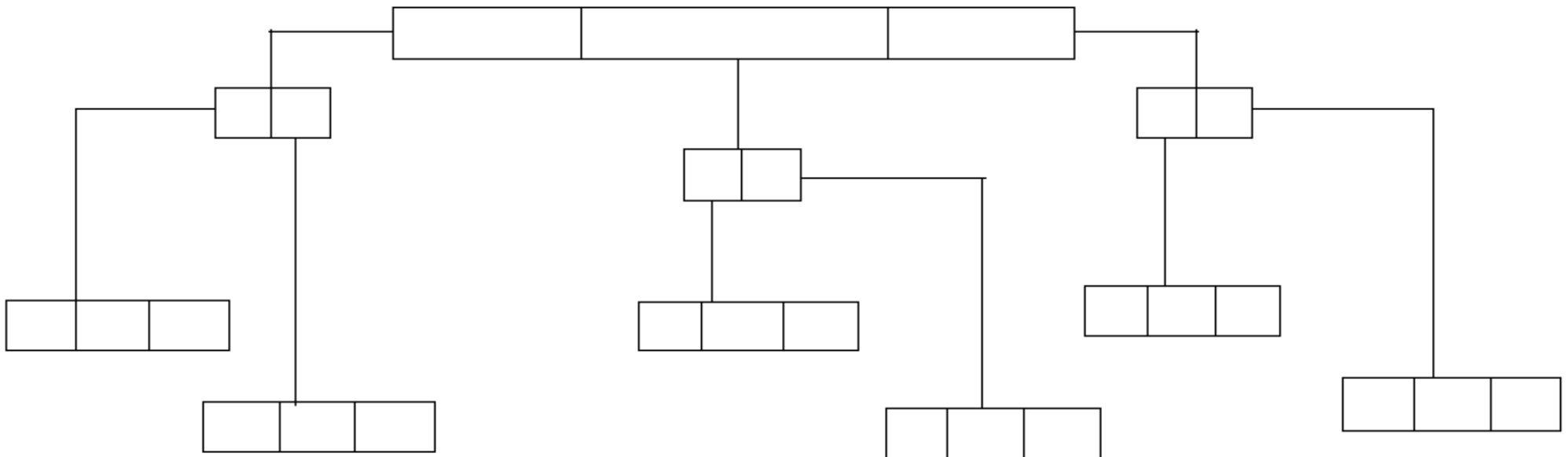
Matrizes



Exemplo:

Os valores de fato, estão armazenados apenas na última dimensão !!

Matriz 3 x 2 x 3 :



www.3way.com.br



Matrizes

Podem ter N dimensões, sendo que estas dimensões lhe dão o nome de N -dimensional.

Exemplos:

Com 2 dimensões : *bi-dimensional*.

Com 3 dimensões : *tri-dimensional*.

...

Com N dimensões : *N -dimensional*

www.3way.com.br



Matrizes

Declarando uma matriz

Declarar uma matriz ou um vetor com **N** dimensões, funciona praticamente da mesma forma que um vetor unidimensional.

Sintaxe:

nome_da_matriz : **vetor** [*tamanhoDimensao1* , *tamanhoDimensao2*] **de** *tipo_da_matriz*



Matrizes

Declarando uma matriz

Exemplos:

arr : vetor[0..2, 0..1] de inteiro

x : vetor[0..4, 0..1] de real

results : vetor[0..3, 0..4] de logico

clientes : vetor[0..5, 0..4] de literal



Matrizes

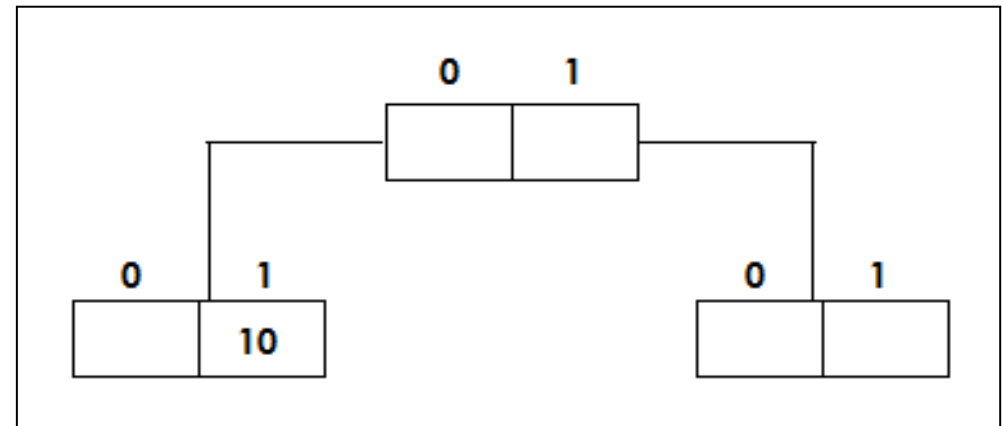
Acessando um elemento da matriz

Para acessarmos uma determinada posição da matriz, devemos indicar entre os colchetes os índices que devemos acessar até a posição desejada.

Exemplo:

matriz : vetor[0..1 , 0..1] de inteiro

matriz [0, 1] \leftarrow 10



Matrizes

Acessando um elemento da matriz

Exemplo:

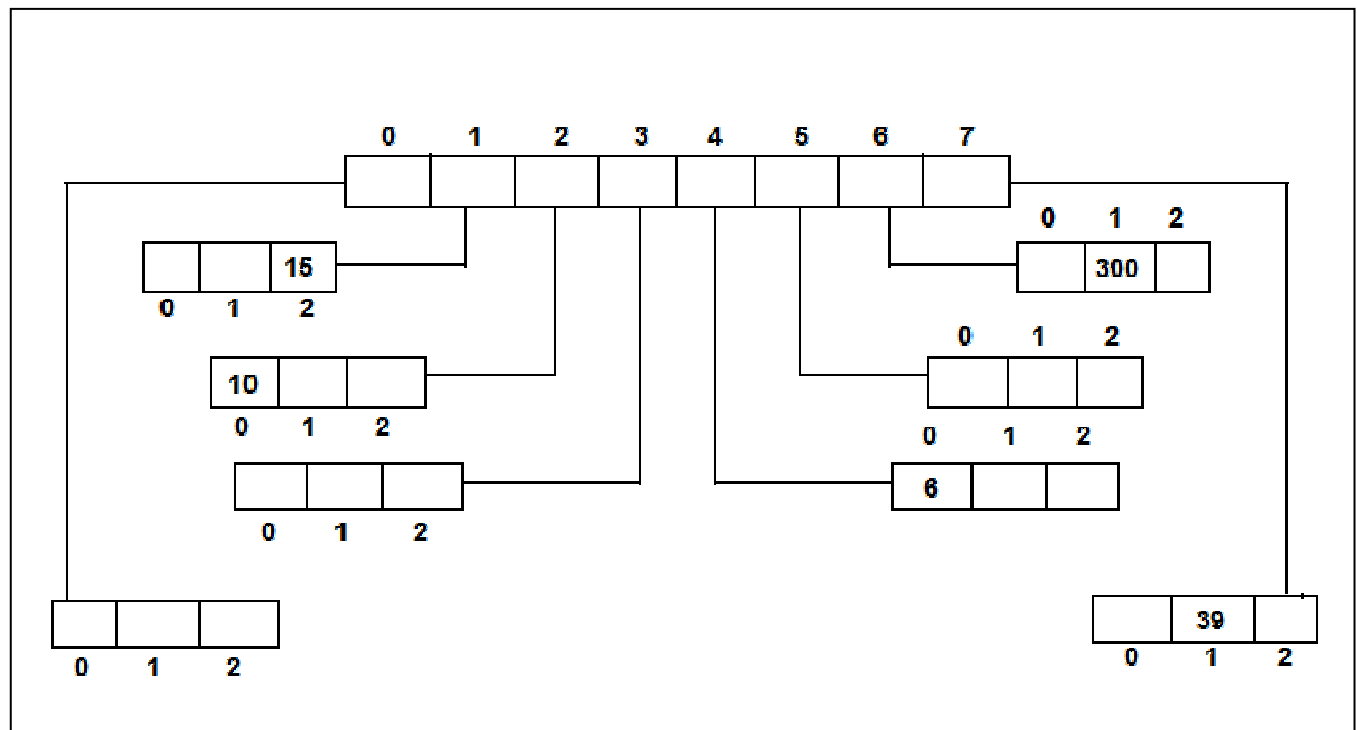
$matriz[1, 2] \leftarrow 15;$

$matriz[2, 0] \leftarrow 10;$

$matriz[4, 0] \leftarrow 6;$

$matriz[6, 1] \leftarrow 300;$

$matriz[7, 1] \leftarrow 39;$



Matrizes

Acessando um elemento da matriz

Exemplo1:

algoritmo "Matriz_Bi-Dimensional"

var

matriz : **v**etor[0..*N*, 0..*M*] de **i**nteiro

linha, *coluna* : **i**nteiro

valor : **i**nteiro



Matrizes

Acessando um elemento da matriz

Exemplo1:

inicio

leia (N)

leia (M)

valor \leftarrow 1

para linha **de** 0 **ate** N **faca**

para coluna **de** 0 **ate** M **faca**



Matrizes



Dessa forma, concluímos que para cada dimensão temos uma estrutura aninhada do tipo PARA-ATÉ-FAÇA, para percorrer suas respectivas posições.

Acessando um elemento da matriz

Exemplo1:

$matriz[linha, coluna] \leftarrow valor$

$valor \leftarrow valor + 1$

fimpara

fimpara

[fimalgoritmo](#)



Matrizes



Run Jeliot - matrizes

www.3way.com.br



Laboratório 03

www.3way.com.br

