

# Aula 9 — Matrizes e Vetores Multidimensionais

Processamento da Informação

Universidade Federal do ABC

# MATRIZES E VETORES MULTIDIMENSIONAIS

- ▶ Matrizes e Vetores Multidimensionais são generalizações de vetores simples vistos anteriormente.
- ▶ Suponha por exemplo que devemos armazenar as notas de cada aluno em cada uma das listas de exercício de PI
- ▶ Podemos alocar 7 vetores (um para cada lista) de tamanho 30 (tamanho da turma), onde cada vetor representa as notas de uma lista específica.
- ▶ Matrizes e Vetores Multidimensionais permitem fazer a mesma coisa mas com todas as informações sendo acessadas por um nome comum (em vez de 7 nomes distintos).

# DECLARAÇÃO

```
TIPO[]...[] identificador = new TIPO[tamanho_1]...[tamanho_n]
```

Exemplo:

```
float[][] notasListas = new float[7][30];
```

- ▶ O acesso aos elementos deve ser feito utilizando `[i][j]`..., onde `i` e `j` são os índices da posição desejada.
- ▶ O uso do valor `v[i][j]` pode ser feito **exatamente como uma variável**.
- ▶ **ATENÇÃO** Assim como os vetores unidimensionais, os elementos são **indexados a partir de 0**.

```
float[][] notasListas = new float[7][30];  
notasListas[0][0] = 9.2; //Primeira nota do primeiro aluno  
notasListas[5][2] += 1;
```

# MATRIZES - DECLARAÇÃO

- ▶ É muito comum criarmos matrizes usando a seguinte sintaxe:

```
float[][] notasListas = new float[7][30];
```

Que é equivalente a:

```
float[][] notasListas = new float[7][];  
notasListas[0] = new float[30];  
notasListas[1] = new float[30];  
...  
notasListas[6] = new float[30];
```

## VETORES MULTIDIMENSIONAIS - DECLARAÇÃO

- ▶ Contudo, imagine que os alunos vão desistindo da disciplina. Não é necessário um vetor de 30 elementos para cada uma das listas.
- ▶ Pode-se então declarar sua estrutura de dados assim

```
float[][] notasListas = new float[7][];  
notasListas[0] = new float[30];  
notasListas[1] = new float[25];  
notasListas[2] = new float[20];  
...  
notasListas[6] = new float[15];
```

## DECLARAÇÃO E INICIALIZAÇÃO SIMULTÂNEA

- ▶ É possível declarar e já inicializar os valores da seguinte maneira

```
int notas[][]={{50,60,55},{62,65,70},{72,66,77}};
```

- ▶ Que é equivalente a

```
int notas[][]= new int[3][3];  
notas[0][0] = 50;  
notas[0][1] = 60;  
notas[0][2] = 55;  
notas[1][0] = 62;  
notas[1][1] = 65;  
notas[1][2] = 70;  
notas[2][0] = 72;  
notas[2][1] = 66;  
notas[2][2] = 77;
```

# ESTRUTURAS N-DIMENSIONAIS

- ▶ Imagine que deseja-se fazer um controle para uma biblioteca. Especificamente deseja-se saber quantos livros estão em cada uma das prateleiras de cada uma das estantes.
- ▶ Considere que:
  - ▶ A biblioteca tem 2 prédios
  - ▶ O primeiro prédio tem 10 estantes e o segundo 20
  - ▶ As estantes são todas iguais e todas têm 5 prateleiras

## ESTRUTURAS N-DIMENSIONAIS

- ▶ Considere que:
  - ▶ A biblioteca tem 2 prédios
  - ▶ O primeiro prédio tem 10 estantes e o segundo 20
  - ▶ As estantes são todas iguais e todas têm 5 prateleiras

```
int [][][]num_livros = new int[2][][]; //2 prédios
num_livros[0] = new int[10][5]; //10 estantes, 5 prateleiras
num_livros[1] = new int[20][5]; //20 estantes, 5 prateleiras
//1ª prateleira da 3ª estante do 2º prédio tem 17 livros
num_livros[1][2][0] = 17;

...
System.out.printf(
    "A 1ª prateleira, 10ª estante, 1º prédio tem %d livros\n",
    num_livros[0][9][0]);
```

## EX. 1 - SOMA E SUBTRAÇÃO DE MATRIZES

- ▶ Crie uma função que receba como parâmetro 2 matrizes quadradas de inteiros,  $A$  e  $B$ , e que devolva  $A + B$
- ▶ Crie uma função que receba como parâmetro 2 matrizes quadradas de inteiros,  $A$  e  $B$ , e que devolva  $A - B$
- ▶ Defina qual a assinatura sua função deve possuir. Quais informações ela necessita como parâmetro?
- ▶ Crie o seu próprio programa com diferentes exemplos de entrada para testar as suas funções.
- ▶ Lembre-se de testar diferentes tamanhos de matrizes

## EX. 2 - MATRIZ TRANSPOSTA

- ▶ Crie uma função que recebe uma matriz  $A$  e que devolve a sua transposta
- ▶ **Atenção:**  $A$  pode não ser quadrada, mas você pode assumir que todas as linhas tem o mesmo número de colunas.
- ▶ Assim como no primeiro exercício, os testes ficam por sua conta.

## EX. 3 - MULTIPLICAÇÃO DE MATRIZES

- ▶ Crie uma função que recebe uma matriz  $A$ , com  $M$  linhas e  $N$  colunas e uma matriz  $B$ ,  $N$  linhas e  $P$  colunas e que devolve como resultado uma matriz  $C$ , com  $M$  linhas e  $P$  colunas tal que  $C = A \times B$ .
- ▶ Lembre-se que a posição  $(i, j)$  da matriz de resultado é o produto interno (feito na aula passada!) do vetor linha  $i$  de  $A$  e do vetor coluna  $j$  de  $B$ :

$$C[i, j] = \sum_{k=0}^{N-1} A[i, k] \cdot B[k, j]$$

# LISTA DE EXERCÍCIOS (ENTREGA VIA URI)

- ▶ 1181 (Linha na Matriz)
- ▶ 1182 (Coluna na Matriz)
- ▶ 1183 (Acima da Diagonal Principal)
- ▶ 1185 (Acima da Diagonal Secundária)
- ▶ 1190 (Área Direita)
- ▶ 1435 (Matriz Quadrada I)
- ▶ 1478 (Matriz Quadrada II)
- ▶ 1534 (Matriz Quadrada 123)
- ▶ 2710 (Jogo da Matriz)
- ▶ 2163 (Despertar da Força)
- ▶ 2168 (Crepusculo em Portland)