

Matrizes



Objetivos

- Conceituação de Matrizes Bidimensionais
- Manipulação de Matrizes Bidimensionais
- Entender como manipular entrada, saída e índices de matrizes bidimensionais
- Resolver problemas que requeiram o uso de matrizes bidimensionais



Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia
Processamento da Informação

Relembrando

Vetores em Java

- Permite a declaração de **variáveis do tipo CONJUNTO**
- Uma variável do tipo CONJUNTO pode armazenar **mais de um valor**.
- No ato da declaração da variável **pode ou não** informar o seu **tamanho**.
- Para o problema de armazenar **10 notas**, pode-se definir uma variável de **tamanho 10** do tipo double.
 - Declaração:

double nota[] = new double[10]

nota										
conteúdo	5.5	6.5	8.0	3.0	7.5	2.5	7.5	6.0	4.5	10.0
índice	0	1	2	3	4	5	6	7	8	9

Vetores em Java

- Declaração:

double **nota**[] = new double[10]

- A declaração acima cria em memória uma variável chamada **NOTA** com **10 posições** do tipo REAL
- As 10 posições são numeradas de **0 a 9 (índice)**

nota										
conteúdo	5.5	6.5	8.0	3.0	7.5	2.5	7.5	6.0	4.5	10.0
índice	0	1	2	3	4	5	6	7	8	9

- Acesso a variável:

Para acessar cada posição deve-se usar o nome da variável e a sua posição **ou índice** (entre colchetes)

A instrução abaixo imprime O TERCEIRO ELEMENTO da variável NOTA.

- escreva(“ A TERCEIRA NOTA = ” , **nota [2]**)

O índice pode ser uma variável:

Ex. **nota [x]**



Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia

Processamento da Informação

PROBLEMA

Problema

- **Escreva um programa para armazenar as notas de 20 alunos em 5 disciplinas.**

Escreva um programa para armazenar as notas de 20 alunos em 5 disciplinas.

ALUNOS	DISC01	DISC02	DISC03	DISC04	DISC05
1	5,5	8,3	6,5	10,0	9,5
2	8,0	7,5	8,5	6,5	10,0
3
4
5
6
7	7,5	8,0	9,0	8,0	8,5
...	6,5	3,5	6,5	4,5	9,0
18
19
20	8,0	9,0	8,0	10,0	9,0

Escreva um programa para armazenar as notas de 20 alunos em 5 disciplinas.

ALUNOS	DISC01	DISC02	DISC03	DISC04	DISC05
1	5,5	8,3	6,5	10,0	9,5
2	8,0	7,5	8,5	6,5	10,0
3
4
5
6
7	7,5	8,0	9,0	8,0	8,5
...	6,5	3,5	6,5	4,5	9,0
18
19
20	8,0	9,0	8,0	10,0	9,0

Poderíamos enxergar essa estrutura como:

- 5 vetores de 20 posições
- 20 vetores de 5 posições

Ou

1 matriz de 20 x 5



Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia
Processamento da Informação

MATRIZES

MATRIZES

**Matrizes são estruturas
MULTIDIMENSIONAIS (mais de uma
dimensão) capazes de armazenar dados**

**A figura abaixo representa uma matriz
BIDIMENSIONAL de números inteiros**

10	5	33	41
53	20	-10	0
29	17	30	8

MATRIZES

		Colunas (j)				
		0	1	2	3	4
Linhas (i)	0					
	1					
	2					
	3					
	4					
	5					
	6					
	7					

No exemplo ao lado
temos uma matriz de
8 linhas por 5 colunas

- Sintaxe em java:

```
tipo nome[][] = new tipo[#linhas][#colunas];
```

- Acesso a elementos da matriz**

	0	1	2	3
0	10	5	33	41
1	53	20	-10	0
2	29	17	30	8

- `System.out.print(A [2] [1]);` ← imprime o valor 17

- `i = 0, j = 2;`
- `System.out.print (A [i] [j]);` ← imprime o valor 33
- `System.out.print (A [j] [i]);` ← imprime o valor 29

Matrizes Bidimensionais

- Para acessar TODOS os elementos da matriz é necessário que os índices assumam todas as combinações possíveis

- (i, j)
- $(0,0), (0,1), (0,2), (0,3) \leftarrow i = 0 \text{ e } j = 0, 1, 2, 3$
- $(1,0), (1,1), (1,2), (1,3) \leftarrow i = 1 \text{ e } j = 0, 1, 2, 3$
- $(2,0), (2,1), (2,2), (2,3) \leftarrow i = 2 \text{ e } j = 0, 1, 2, 3$

A

	0	1	2	3
0	10	5	33	41
1	53	20	-10	0
2	29	17	30	8

Problema 1

- **Escreva um programa para LER (via teclado) uma matriz de 3 x 4 e EXIBIR o seu conteúdo no console.**

```
//package aulaspi;  
public class Main {  
  
    static java.util.Scanner scanner = new java.util.Scanner(System.in);  
  
    public static void main(String[] args) {  
  
        System.out.print("Digite o numero de linhas: ");  
        int L = scanner.nextInt();  
        System.out.print("Digite o numero de colunas: ");  
        int C = scanner.nextInt();  
  
        int m[][] = new int[L][C]; // matriz m de inteiros  
  
        // ler matriz  
        System.out.print("Digite os elementos da Matriz: ");  
        for (int i = 0; i < m.length; i++) { // linhas i  
            for (int j = 0; j < m[0].length; j++) { // coluna j  
                m[i][j] = scanner.nextInt(); // lê elemento i,j  
            }  
        }  
  
        // imprimir matriz (cada linha de m em uma linha)  
        System.out.println("Matriz: ");  
        for (int i = 0; i < m.length; i++) { // linhas i  
            for (int j = 0; j < m[0].length; j++) { // colunas j  
                System.out.print(m[i][j] + "\t");  
            }  
            System.out.println();  
        }  
    }  
}
```

Degustar este código em
<https://repl.it/languages/java>

```
run:  
Digite o numero de linhas: 3  
Digite o numero de colunas: 4  
Digite os elementos da Matriz: 1 2 3 4 5 6 7 8 9 10 11 12  
Matriz:  
1      2      3      4  
5      6      7      8  
9      10     11     12  
CONSTRUÍDO COM SUCESSO (tempo total: 11 segundos)
```



```
public class Main {
    static java.util.Scanner scanner = new java.util.Scanner(System.in);
    public static void main(String[] args) {
        System.out.print("Digite o numero de linhas: ");
        int L = scanner.nextInt();
        System.out.print("Digite o numero de colunas: ");
        int C = scanner.nextInt();

        int m[][];

        m = lerMatriz(L, C); // Método de entrada
        imprimirMatriz(m); // Método de saída
    }

    static int[][] lerMatriz(int L, int C) {
        int m[][] = new int[L][C];
        for (int i = 0; i < m.length; i++) { // linhas i
            for (int j = 0; j < m[0].length; j++) { // coluna j
                m[i][j] = scanner.nextInt(); // lê elemento i,j
            }
        }
        return m;
    }

    static void imprimirMatriz(int m[][]) {
        for (int i = 0; i < m.length; i++) { // linhas i
            for (int j = 0; j < m[0].length; j++) { // colunas j
                System.out.print(m[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Agora com
métodos, que
podem ser
usados em
vários
problemas
semelhantes!

Degustar este código em
<https://repl.it/languages/java>

```
run:
Digite o numero de linhas: 3
Digite o numero de colunas: 4
Digite os elementos da Matriz: 1 2 3 4 5 6 7 8 9 10 11 12
Matriz:
1      2      3      4
5      6      7      8
9      10     11     12
CONSTRUÍDO COM SUCESSO (tempo total: 11 segundos)
```

```
public class Main {
    static java.util.Scanner scanner = new java.util.Scanner(System.in);
    public static void main(String[] args) {
        System.out.print("Digite o numero de linhas: ");
        int L = scanner.nextInt();
        System.out.print("Digite o numero de colunas: ");
        int C = scanner.nextInt();

        int m[][];

        m = lerMatriz(L, C); // Método de entrada
        imprimirMatriz(m); // Método de saída
    }

    static int[][] lerMatriz(int L, int C) {
        int m[][] = new int[L][C];
        for (int i = 0; i < m.length; i++) { // linhas i
            for (int j = 0; j < m[0].length; j++) { // coluna j
                m[i][j] = scanner.nextInt(); // lê elemento i,j
            }
        }
        return m;
    }

    static void imprimirMatriz(int m[][]) {
        for (int i = 0; i < m.length; i++) { // linhas i
            for (int j = 0; j < m[0].length; j++) { // colunas j
                System.out.print(m[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Agora com métodos, que poderão se usados em vários problemas semelhantes!

Problema: Usuário digita “no escuro”!
Vamos melhorar o método *lerMatriz*?

```
run:
Digite o numero de linhas: 3
Digite o numero de colunas: 4
Digite os elementos da Matriz: 1 2 3 4 5 6 7 8 9 10 11 12
Matriz:
1      2      3      4
5      6      7      8
9      10     11     12
CONSTRUÍDO COM SUCESSO (tempo total: 11 segundos)
```

Problema 2

- Escreva um programa para **exibir a soma dos elementos da linha K** de uma matriz 3 x 4.

A		0	1	2	3
	0	10	5	33	41
	1	53	20	-10	0
	2	29	17	30	8

Problema 3

Escreva um programa que leia uma matriz $N \times N$ e mostre a soma da diagonal principal.

	0	1	2
0	10	5	33
1	53	20	-10
2	29	17	30

Problema 4

Construa um método que receba, por parâmetro, uma matriz (preenchida) e suas respectivas dimensões e retorne V (verdadeiro) se existir alguma linha com todos os elementos zerados e F (falso) caso contrário.

$$\begin{pmatrix} 1 & 0 & 2 & 3 \\ 4 & 0 & 5 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Problema 5

Escreva um programa para ler duas matrizes A e B 5×3 . Construir uma matriz C de mesma dimensão, sendo que C é formada pela soma dos elementos da matriz A com os elementos da matriz B. Apresentar os elementos da matriz C.