



Técnicas de Programação e Algoritmos - TPA

- Matriz



Introdução

Vetores e Matrizes em geral são caracterizadas por se tratarem de uma única variável de um determinado tamanho que armazena várias informações do mesmo tipo.

Essas informações são gravadas na memória sequencialmente e são referenciadas através de índices.

Vetores – Unidimensionais

Matrizes – Multidimensionais

São muito utilizados por exemplo, em desenvolvimento de jogos, análise de conjunto de dados, desenhos, cálculos matemáticos entre outros.

Matriz

É uma variável que possui mais de uma dimensão (linha).

A declaração de matrizes bidimensionais em java deve obedecer a seguinte sintaxe:

```
Tipo[ ] [ ] nome_variável = new Tipo[qtde_linhas][qtde_colunas];
```

O Tipo deve ser especificado de acordo com o tipo de informação que será armazenada na matriz (ex. int float, char, String...).

E a *qtde_linha* representa a quantidade máxima de linhas da matriz e *qtde_colunas* representa a quantidade máxima de colunas da matriz.

É importante lembrar que as matrizes começam pelo índice 0 (zero) tanto na linha quanto na coluna, para indicar a posição do primeiro elemento na matriz.

Matriz – Exemplo de Declaração e Inicialização

Também é possível inicializar a matriz no momento de sua declaração. Para isso veja a sintaxe abaixo:

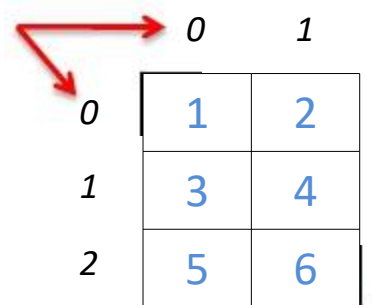
Tipo [][] nome_matriz = { { valor1, valor2}, {valor3, valor4}, {valor5, valor6} };

Sendo que o 1º conjunto de chaves representa a quantidade de linhas da matriz e o 2º conjunto de chaves representa a quantidade de colunas, portanto neste exemplo, a matriz é do tamanho 3 (linhas) x 2 (colunas).

Declaração da variável **matriz_exemplo** com os valores atribuídos.

`int [][] matriz_exemplo = { { 1, 2 }, { 3, 4 }, {5, 6} };`

Índices/Posições



The diagram shows a 3x2 matrix. To the left of the matrix, the row indices 0, 1, and 2 are listed. Above the matrix, the column indices 0 and 1 are listed. Red arrows point from the text 'Índices/Posições' to these indices. The matrix cells contain the values 1, 2, 3, 4, 5, and 6 in blue text. A red arrow points from the bottom right of the matrix to the text 'Conteúdo da posição linha 2 e coluna 1'.

	0	1
0	1	2
1	3	4
2	5	6

Conteúdo da posição linha 2 e coluna 1

Exemplo 7 – Cadastro e Exibição de dados usando Matriz

Neste exemplo, o propósito é cadastrar 9 números Inteiros em uma matriz e depois exibi-los na tela.

Observem que foi necessária a declaração de apenas 1 variável para armazenar os 9 números inteiros e outras 2 variáveis do tipo ***int*** para referenciar as posições Linha e coluna na matriz.

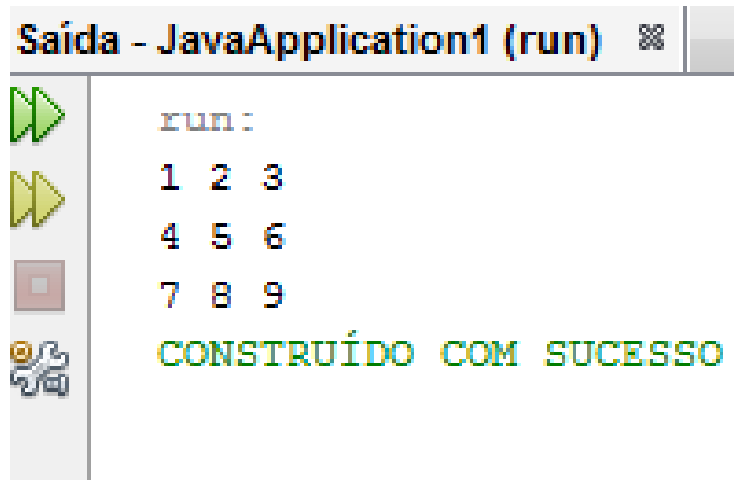
- Verifique o código no próximo slide:

```
1 import javax.swing.JOptionPane;
2
3 public class Exemplo_7 {
4
5     public static void main(String[] args) {
6         final int N_Lin = 3, N_Col = 3; // final para definir constante
7         int[][] mat = new int[N_Lin][N_Col];
8         int l,c;
9
10        JOptionPane.showMessageDialog(null, "***** Cadastrando dados na matriz *****\n Entre com o valor em cada posição da matriz");
11
12        //atribuição dos valores na matriz
13        for (l = 0; l < N_Lin; l++)
14        {
15            for (c = 0; c < N_Col; c++)
16            {
17                mat[l][c]=Integer.parseInt(JOptionPane.showInputDialog("Valor linha "+(l + 1)+" coluna "+(c + 1)+" ::"));
18            } // fim do for da coluna
19        } // fim do for da linha
20
21        JOptionPane.showMessageDialog(null, "***** Exibindo os dados da matriz *****");
22
23        //exibição dos valores na matriz
24        for (l = 0; l < N_Lin; l++)
25        {
26            for (c = 0; c < N_Col; c++)
27            {
28                System.out.print(mat[l][c]+" ");
29            } // fim do for da coluna
30            System.out.print("\n"); // para organizar a apresentação em linhas diferentes
31        } // fim do for da linha
32    }
33
34 }
```

Exemplo 7 – Cadastro e Exibição de dados usando Matriz - Resultado

Foram digitados 9 números inteiros e na sequência todos os números foram exibidos na tela, de acordo com o conteúdo de cada posição da matriz.

Notamos que para exibir no formato de matriz, logo após o término de cada iteração do segundo *for*, foi usando um **System.out.print("\n");** ou seja, após percorrer todas as colunas de cada linha, é necessário pular uma linha.



```
Saída - JavaApplication1 (run) ✖
run:
1 2 3
4 5 6
7 8 9
CONSTRUÍDO COM SUCESSO
```

The screenshot shows a Java IDE output window titled "Saída - JavaApplication1 (run)". It contains the output of a program that prints a 3x3 matrix of integers. The output is displayed in a monospaced font with line wrapping. The first line is "run:", followed by three lines of the matrix: "1 2 3", "4 5 6", and "7 8 9". The final line is "CONSTRUÍDO COM SUCESSO" in all caps. The output window has a standard toolbar with icons for running, stepping through code, and other debugging actions.

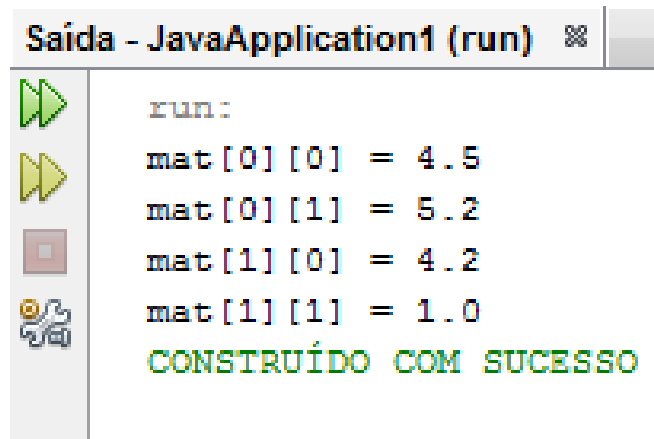
Exemplo 8 – Matriz com valores pré-definidos

Neste exemplo, a matriz foi declarada e já inicializada com valores definidos em tempo de programação. E na sequência os dados são exibidos na tela.

```
1  import javax.swing.JOptionPane;
2
3  public class Exemplo_8 {
4      public static void main(String[] args) {
5          double[][] matriz = {{4.5,5.2},{4.2,1}};
6          int x,y;
7          JOptionPane.showMessageDialog(null, "***** Imprimindo a matriz *****");
8          //exibição dos valores na matriz
9          for (x = 0; x < 2; x++)
10             {
11                 for (y = 0; y < 2; y++)
12                     {
13                         System.out.println("mat["+x+"]["+y+"] = "+matriz[x][y]+" ");
14                     }// fim do for da coluna (y)
15             }// fim do for da linha (x)
16     }
17 }
```


Exemplo 8 – Matriz com valores pré-definidos - Resultado

O resultado apresentado, mostra os valores reais sendo exibidos com apenas uma casa decimal após a vírgula.



```
Saída - JavaApplication1 (run) ✖  
  
run:  
mat[0][0] = 4.5  
mat[0][1] = 5.2  
mat[1][0] = 4.2  
mat[1][1] = 1.0  
CONSTRUÍDO COM SUCESSO
```

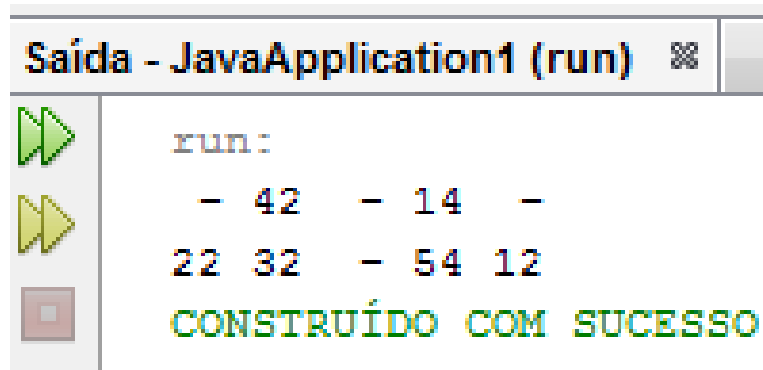
Exemplo 9 – Testando elementos da Matriz

Neste exemplo, a matriz também foi declarada e já inicializada com valores definidos em tempo de programação. E na sequência os dados são exibidos na tela.

```
1  import javax.swing.JOptionPane;
2
3  public class Exemplo_9 {
4
5      public static void main(String[] args) {
6          int[][] mat = {{1,42,23,14,51},{22,32,55,54,12}};
7          int x,y;
8          JOptionPane.showMessageDialog(null, "*****  Exibindo apenas os elementos pares da matriz *****");
9          //exibição dos valores na matriz
10         for (x = 0; x < 2; x++)
11         {
12             for (y = 0; y < 5; y++)
13             {
14                 if(mat[x][y] % 2 == 0)
15                 {
16                     System.out.print(mat[x][y]+ " ");
17                 }
18                 else
19                 {
20                     System.out.print(" - ");
21                 }
22             }
23             System.out.print("\n");
24         }
25     }
26 }
27
28
29
```

Exemplo 9 – Testando elementos da Matriz - Resultado

O resultado apresentado, mostra apenas os valores pares que estão armazenados na matriz.



```
Saída - JavaApplication1 (run) ✖
run:
- 42 - 14 -
22 32 - 54 12
CONSTRUÍDO COM SUCESSO
```

The screenshot shows a Java IDE output window titled "Saída - JavaApplication1 (run)". The output text is as follows:

```
run:
- 42 - 14 -
22 32 - 54 12
CONSTRUÍDO COM SUCESSO
```

The output displays the even elements of a matrix. The first line shows the even elements of the first row: -42, -14, and -2. The second line shows the even elements of the second row: 22, 32, -54, and 12. The third line indicates that the program was built successfully.

Exercício

Escreva um programa que leia uma **matriz** 4 x 4 e ao final exiba:

- a soma dos números pares digitados,
- a média dos números pares digitados.

Bom trabalho!!