

Algoritmos II

Ricardo Ribeiro Assink
ricardo.assink@unisul.br
ricardo@equipedigital.com

<http://www.ricardoassink.com.br/>

Matrizes

Uma variável indexada Bidimensional também conhecida como “Matriz”, como o próprio nome já indica, possui duas dimensões (linha e coluna), sendo possível definir variáveis com quaisquer tipo de dados.

Algoritmos II

Vetores e Matrizes

Matrizes

Exemplo de Matriz

Abaixo temos um exemplo de uma matriz chamada A de 3 linhas por 4 colunas ou simplesmente A 3 x 4.

	1	2	3	4
1	A[1,1]	A[1,2]	A[1,3]	A[1,4]
2	A[2,1]	A[2,2]	A[2,3]	A[2,4]
3	A[3,1]	A[3,2]	A[3,3]	A[3,4]

Dentro dos algoritmos:

	0	1	2	3
0	A[0][0]	A[0][1]	A[0][2]	A[0][3]
1	A[1][0]	A[1][1]	A[1][2]	A[1][3]
2	A[2][0]	A[2][1]	A[2][2]	A[2][3]

Algoritmos II

Vetores e Matrizes

Matrizes - Utilização

Cidade	Florianópolis	Tubarão	Itajaí	Joinville
Florianópolis	0	137	115	156
Tubarão	137	0	252	294
Itajaí	115	252	0	41
Joinville	156	294	41	0

Despesa/Mês	Janeiro	Fevereiro	Março	Abril	Maior	Junho
Faculdade						
Passagem						
Alimentação						
Aluguel						

Algoritmos II

Vetores e Matrizes

Matrizes

Declaração

```
public class Define
{
    public static void main(String args[])
    {
        <tipo de dado> <nome>[][] = new <tipo de dado>[i1][i2];

        <comandos>
    }
}
```

Obs.:

- a) O valor "i1" corresponde ao número de linhas da matriz.
- b) O valor "i2" corresponde ao número de colunas da matriz.
- c) Uma variável indexada pode ser apenas de um tipo de dado

Matrizes

Exemplo: Definir uma variável indexada bidimensional como sendo do tipo Real, sendo que a matriz deverá ter 3 linhas e 4 colunas. Esta matriz deverá corresponder a 12 posições de memória.

```
public class Exemplo_Declaracao {  
    public static void main(String args[]) {  
        double A[ ][ ] = new double[3][4];  
        <comandos>  
    }  
}
```


Algoritmos II

Vetores e Matrizes

Após a definição da variável A, a memória estará como mostrada no esquema abaixo:

	0	1	2	3
0				
A = 1				
2				

Os Valores numéricos apresentados acima(coluna) e a esquerda(linha) correspondem aos índices da variável indexada bidimensional. Não esqueça que o índice que identifica a primeira linha e a primeira coluna no JAVA têm sempre valor 0. Portanto se tivermos uma matriz com 10 linhas, a primeira linha terá como índice o valor 0 e a última linha terá como índice o valor 9.

Atribuição

Exemplo:

```
public class Atribui {  
    public static void main(String args[ ]) {  
        double A[ ][ ] = new double[3][4];  
        A[0][0] = 10;  
        A[1][1] = 20;  
        A[2][1] = 30;  
        A[2][3] = A[0][0] + A[2][1];  
    }  
}
```


Algoritmos II

Vetores e Matrizes

Atribuição

	0	1	2	3
0	10			
A = 1		20		
2		30		40

Exemplo:

```
public class Atribui {  
    public static void main(String args[ ]) {  
        double A[ ][ ] = new double[3][4];  
        A[0][0] = 10;  
        A[1][1] = 20;  
        A[2][1] = 30;  
        A[2][3] = A[0][0] + A[2][1];  
    }  
}
```

Algoritmos II

Leitura

Vetores e Matrizes

```
import javax.swing.*;  
public class Exemplo6 {  
    public static void main(String args[]){  
  
        double A[ ][ ] = new double[3][4];  
        int i, j;  
  
        for(i=0; i<3; i++){  
            for (j=0; j<4; j++){  
                A[i][j] = Double.parseDouble( JOptionPane.showInputDialog("Digite  
o valor da linha "+i+" e coluna "+j+": "));  
            }  
        }  
    }  
}
```

Algoritmos II

Leitura

Vetores e Matrizes

Memória		Tela
i	j	
0	0	Digite o valor na linha 0 e coluna 0 : 13
0	1	Digite o valor na linha 0 e coluna 1 : 15
0	2	Digite o valor na linha 0 e coluna 2 : 12
0	3	Digite o valor na linha 0 e coluna 3 : 11
1	0	Digite o valor na linha 1 e coluna 0 : 27
1	1	Digite o valor na linha 1 e coluna 1 : 20
1	2	Digite o valor na linha 1 e coluna 2 : 29
1	3	Digite o valor na linha 1 e coluna 3 : 21
2	0	Digite o valor na linha 2 e coluna 0 : 34
2	1	Digite o valor na linha 2 e coluna 1 : 27
2	2	Digite o valor na linha 2 e coluna 2 : 32
2	3	Digite o valor na linha 2 e coluna 3 : 33

Algoritmos II

Leitura

Vetores e Matrizes

A memória ficará assim após a execução do programa de leitura:

		0	1	2	3
A =	0	13	15	12	11
	1	27	20	29	21
	2	34	27	32	33

Algoritmos II

Vetores e Matrizes

Escrita

```
import javax.swing.*;
public class Exemplo7 {
    public static void main(String args[ ]) {
        double A[ ][ ] = new double[3][4];
        int i, j;

        // leitura dos valores
        for(i=0; i<3; i++) {
            for (j=0; j<4; j++){
                A[i][j] = Double.parseDouble(JOptionPane.showInputDialog("Digite o valor da
linha "+i+" e coluna "+j+": "));
            }
        }

        // escrita dos valores
        for(i=0; i<3; i++){
            for (j=0; j<4; j++){
                JOptionPane.showMessageDialog(null, "O valor contido na linha " +i+" e coluna
"+j+ " eh " + A[i][j]);
            }
        }
    }
}
```

Algoritmos II

EXEMPLO 8

Vetores e Matrizes

	Florianópolis	Blumenau	Lages
Florianópolis	0	200	300
Blumenau	200	0	500
Lages	300	500	0


```

import javax.swing.*;
public class Exemplo8{
    public static void main(String
args[]){
        double A[][] = new double[3][3];
        int i, j;

        // cria as legendas da matriz
        String linhas_A[] = new String [3];
        linhas_A[0] = "Florianópolis";
        linhas_A[1] = "Blumenau";
        linhas_A[2] = "Lages";

        String cols_A[] = new String [3];
        cols_A[0] = "Florianópolis";
        cols_A[1] = "Blumenau";
        cols_A[2] = "Lages";

```

```

// distâncias partindo de florianópolis

```

```

A[0][0] = 0; // floriipa -> floriipa
A[0][1] = 200; // floriipa -> Blumenau
A[0][2] = 300; // floriipa -> Lages

```

```

// distâncias partindo de blumenau

```

```

A[1][0] = 200; // blumenau -> floriipa
A[1][1] = 0; // blumenau -> Blumenau
A[1][2] = 500; // blumenau -> Lages

```

```

// distâncias partindo de lages

```

```

A[2][0] = 300; // lages -> floriipa
A[2][1] = 500; // lages -> Blumenau
A[2][2] = 0; // lages -> Lages

```

```

// mostra a distância entre todas as cidades

```

```

for(i=0; i<3; i++){
    for (j=0; j<3; j++){
        JOptionPane.showMessageDialog(null, "A
distância entre: " + linhas_A[i] + " e " + cols_A[j] + "
é: " + A[i][j]);
    }}}

```

Faça um algoritmo que leia uma matriz de ordem 3x3 de números inteiros. Após a leitura faça:

- a) Calcule e mostre a soma dos elementos da primeira coluna;
- b) Calcule e mostre a soma dos elementos de cada coluna;
- c) Calcule e mostre o produto dos elementos da primeira linha;
- d) Calcule e mostre a soma de todos os elementos da matriz;
- e) Calcule e mostre a média dos elementos da matriz;
- f) Calcule e mostre os elementos que são maiores que a média;
- g) Calcule e mostre o maior elemento da matriz e sua posição;
- h) Calcule e mostre o menor elemento da matriz e sua posição;
- i) Mostre os elementos da diagonal principal;
- j) Mostre os elementos do triângulo superior principal;
- k) Mostre os elementos do triângulo inferior principal;
- l) Calcule e mostre a soma do diagonal principal;
- m) Mostre os elementos da diagonal secundária;
- n) Mostre os elementos do triângulo superior secundário;
- o) Mostre os elementos do triângulo inferior secundário;
- p) Calcule e mostre a Soma da diagonal secundária;

Algoritmos II

FIM

Material Original: Osmar de Oliveira Braz Junior