

Técnico em Desenvolvimento de Sistemas

Lógica de Programação II

Matrizes e Arrays Multidimensionais

Alex Helder Cordeiro do Rosário de Oliveira

Instituto Federal de Brasília - *Campus* Brasília

Arrays Bidimensionais

- Em Java, podemos ter arrays de qualquer tipo;
- Incluindo arrays de arrays.
- Quando fazemos um array de arrays, estamos fazendo um array de duas dimensões.
- O array bidimensional é tradicionalmente chamado de matriz.

3	2	5	4
7	1	4	9
1	0	6	2
8	3	9	4

Arrays Bidimensionais

- As matrizes são úteis para armazenar qualquer tipo de informação que seja melhor representada em formato de tabela.
- Usamos uma dimensão para representar as linhas e outra para representar as colunas.
- Declaramos as matrizes da seguinte forma:

```
int[] [] matriz = new int[4][4];
```

- O acesso a um termo específico na linha x e coluna y se faz da seguinte forma:

```
int a = matriz[x][y];  
matriz[x][y] = 55;
```

Arrays Bidimensionais

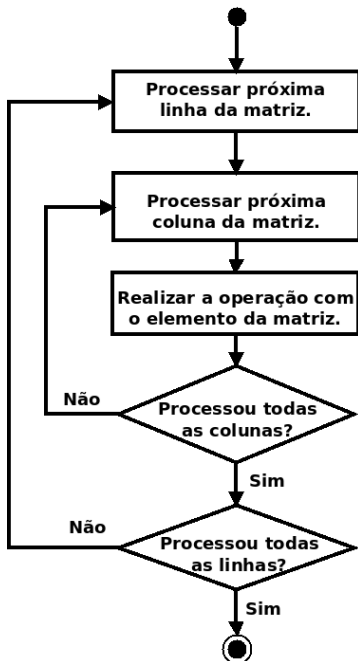
- Assim como os índices de um array comum, os índices de um array bidimensional (tanto o da linha como o da coluna) começam em 0 e terminam no *tamanho - 1* (quantidade de linhas - 1; e quantidade de colunas - 1).
- Toda linha terá o mesmo tamanho.
- Toda coluna terá o mesmo tamanho.
- É importante observar que a utilização de um único colchetes com índice para referenciar um elemento de uma matriz na realidade estará referenciando uma linha inteira.

Arrays Bidimensionais

- Para processar cada linha da matriz, é necessário o uso de um for;
- Para processar cada coluna da matriz, é necessário o uso de um for;
- Portanto, para processar cada elemento da matriz, é necessário que se use um for para vasculhar a linha e outro for para vasculhar a coluna.

```
for(int i = 0; i < quantLinhas;i++) {  
    for(int j = 0; j < quantColunas; j++) {  
        processa(matriz[i][j]);  
    }  
}
```

- Para cada dimensão (consequentemente, para cada for), é necessário o uso de uma variável diferente para o índice.



Arrays Bidimensionais

- Como solicitar ao usuário os elementos da matriz:

```
for(int i = 0; i < quantLinhas;i++) {  
    for(int j = 0; j < quantColunas; j++) {  
        String texto = JOptionPane.showInputDialog("Qual o  
        matriz[i][j] = Integer.parseInt(texto);  
    }  
}
```

- Para mostrar a matriz em forma de tabela:

```
for(int i = 0; i < quantLinhas;i++) {  
    for(int j = 0; j < quantColunas; j++) {  
        System.out.print(matriz[i][j]+" ");  
    }  
    System.out.println("");  
}
```

Arrays Multidimensionais

- Podemos criar também arrays de 3 ou mais dimensões;
- Para cada nova dimensão, temos mais um par de colchetes com o tamanho nesta dimensão.
- A utilização de arrays com muitas dimensões não é algo tão comum, exigindo um código mais sofisticado, e uma apresentação visual mais elaborada.
- Pode-se usar quantas dimensões forem desejadas, mas recomenda-se que só se use o necessário.

Arrays Multidimensionais

- Cada dimensão é limitada a 2.147.483.647 posições, pois é uma quantidade inteira.
- Não existe limite de dimensões em tempo de compilação.

```
Object[] [] [] [] arrayGigante =  
    new Object[512][7345][851][12035][8654];
```

- Acesso aos elementos semelhante aos arrays monodimensional;
 arrayGigante[2][3][1][6][4] = new Object();
- Mas existe o limite de memória da JVM.

Sua vez:

- 17 Escreva um programa que gere uma matriz (array bidimensional) de tamanho 3×3 com valores aleatórios entre 0 e 100 (inclusive). O programa deve mostrar a matriz na tela e depois mostrar os valores das somas dos elementos de cada coluna e das somas dos elementos de cada linha.

Ex.:

10	55	65
72	34	48
100	9	86

Soma da primeira linha: 130

Soma da segunda linha: 154

Soma da terceira linha: 195

Soma da primeira coluna: 182

Soma da segunda coluna: 98

Soma da terceira coluna: 199