



Tutorial JAVA

Matrizes

- » Arrays bidimensionais costumam ser utilizados para representar tabelas de valores consistindo em informações organizadas em linhas e colunas.
- » Identificam um elemento particular com dois índices.
- » Por convenção, o primeiro identifica a linha do elemento e o segundo sua coluna.
- » Arrays multidimensionais podem ter mais de duas dimensões.



Matrizes

- » O Java não suporta arrays multidimensionais diretamente.
- » Permite especificar arrays unidimensionais cujos elementos também são arrays unidimensionais, alcançando assim o mesmo efeito.
- » Em geral, um array com m linhas e n colunas é chamado de array m por n .



Matrizes

	Coluna 0	Coluna 1	Coluna 2	Coluna 3
Linha 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Linha 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Linha 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Índice de coluna
Índice de linha
Nome do array

Figura 7.16 | O array bidimensional com três linhas e quatro colunas.



Matrizes

- Arrays multidimensionais podem ser inicializados com inicializadores de array em declarações.
- Um array bidimensional b com duas linhas e duas colunas poderia ser declarado e inicializado com inicializadores de array aninhados como a seguir:
 - `int[][] b = { { 1, 2 }, { 3, 4 } };`
- Os valores iniciais são agrupados por linha entre chaves.
- O número de inicializadores de array aninhados (representado por conjuntos de chaves dentro das chaves externas) determina o número de linhas.
- O número de valores inicializadores no inicializador de array aninhado para uma linha determina o número de colunas nessa linha.
- Linhas podem ter tamanhos diferentes.



Matrizes

- Não é necessário que os comprimentos das linhas em um array bidimensional sejam os mesmos:

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

Cada elemento de `b` é uma referência a um array unidimensional de variáveis `int`.

O `int` array para a linha `0` é um array unidimensional com dois elementos (`1` and `2`).

O array `int` para a linha `1` é um array unidimensional com três elementos (`3`, `4` e `5`).



Matrizes

- Um array multidimensional com o mesmo número de colunas em cada linha pode ser criado com uma expressão de criação de array.

```
int[][] b = new int[ 3 ][ 4 ];
```

3 linhas e 4 colunas.

- Os elementos de um array multidimensional são inicializados quando o objeto array é criado.
- Pode-se criar um array multidimensional em que cada linha tem um número diferente de colunas como mostrado a seguir:

```
int[][] b = new int[ 2 ][ ];    // cria 2  
linhas  
b[ 0 ] = new int[ 5 ]; // cria 5 colunas  
para a linha 0  
b[ 1 ] = new int[ 3 ]; // cria 3 colunas  
para a linha 1
```

Cria um array bidimensional com duas linhas.

A linha 0 tem cinco colunas e linha 1 tem três colunas.



Inicialização de Matrizes

```
1 // Figura 7.17: InitArray.java
2 // Inicializando arrays bidimensionais.
3
4 public class InitArray
5 {
6     // cria e gera saída de arrays bidimensionais
7     public static void main( String[] args )
8     {
9         int[][] array1 = { { 1, 2, 3 }, { 4, 5, 6 } };
10        int[][] array2 = { { 1, 2 }, { 3 }, { 4, 5, 6 } };
11
12        System.out.println( "Values in array1 by row are" );
13        outputArray( array1 ); // exibe array1 por linha
14
15        System.out.println( "\nValues in array2 by row are" );
16        outputArray( array2 ); // exibe array2 por linha
17    } // fim de main
18
```

array1 tem duas linhas,
cada uma com três colunas

array2 tem três linhas,
com duas, uma e três
colunas, respectivamente

Figura 7.17 | Inicializando arrays bidimensionais. (Parte 1 de 2.)



Inicialização de Matrizes

```
19 // gera saída de linhas e colunas de um array bidimensional
20 public static void outputArray(int[][] array )
21 {
22     // faz um loop pelas linhas do array
23     for ( int row = 0; row < array.length; row++ )
24     {
25         // faz um loop pelas colunas da linha atual
26         for ( int column = 0; column < array[ row ].length; column++ )
27             System.out.printf( "%d ", array[ row ][ column ] );
28
29         System.out.println(); // inicia nova linha de saída
30     } // fim do for externo
31 } // fim do método outputArray
32 } // fim da classe InitArray
```

O método pode receber qualquer array bidimensional de valores `int`

O loop externo define o número de linhas no array

O loop interno determina o número de colunas na *linha atual*

Values in array1 by row are

```
1 2 3
4 5 6
```

Values in array2 by row are

```
1 2
3
4 5 6
```

Figura 7.17 | Inicializando arrays bidimensionais. (Parte 2 de 2.)