

Tutorial JAVA

- » 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.

- » 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.

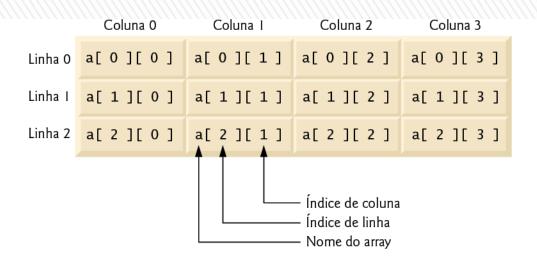


Figura 7.16 | O array bidimensional com três linhas e quatro colunas.



- 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.

• 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).

 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

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

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



Inicialização de Matrizes

```
// gera saída de linhas e colunas de um array bidimensional
19
                                                                            O método pode receber qualquer
        public static void outputArray(int[][] array ) ←
20
                                                                            array bidimensional de valors int
21
           // faz um loop pelas linhas do array
22
                                                                                 O loop externo define o
           23
                                                                                 número de linhas no array
24
25
              // faz um loop pelas colunas da linha atual
                                                                                 O loop interno determina
              for ( int column = 0; column < array[ row ].length; column++ ) ←</pre>
26
                 System.out.printf( "%d ", array[ row ][ column ] );
                                                                                 o número de colunas na
27
                                                                                 linha atual
28
29
              System.out.println(); // inicia nova linha de saída
            } // fim do for externo
30
31
        } // fim do método outputArray
     } // fim da classe InitArray
32
Values in array1 by row are
Values in array2 by row are
1 2
4 5 6
```

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