

Fundamentos de programação

Array - multidimensional

Edson Moreno

edson.moreno@pucrs.br

<http://www.inf.pucrs.br/~emoreno>

Arrays Multidimensionais

- Arrays também podem ser usados para armazenar dados em mais de uma dimensão, como dados de uma tabela
 - Neste caso tem-se uma matriz com linhas e colunas
 - A declaração é feita com dois pares de colchetes:

- Usando new:

```
const int COUNTRIES = 7;  
const int MEDALS = 3;  
int[][] counts = new int[COUNTRIES][MEDALS];
```

- Usando valores inicializados e chaves (neste caso com dois “níveis” de chaves):

```
int[][] counts = {{ 0, 0, 1 }, { 1, 0, 0 }, { 0, 1, 1 },  
                  { 0, 1, 1 }, { 1, 1, 0 }};
```

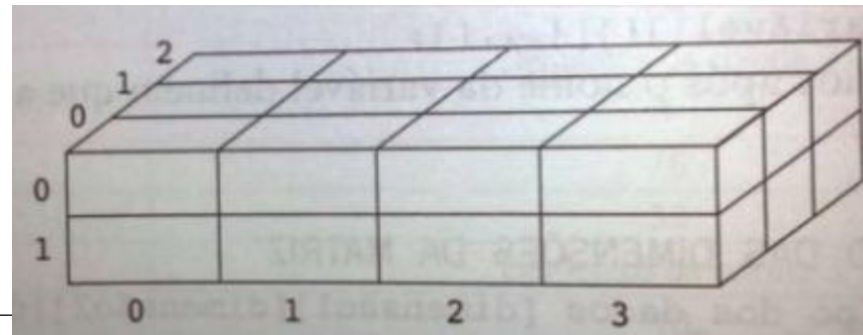
Arrays Multidimensionais

- Exemplo de declaração de array bidimensional
 - `float [][] x;`
 - // Declara uma matriz do tipo float formada por linha e coluna
 - // O primeiro colchete refere-se a linha
 - // O segundo refere-se a coluna
 - // O tamanho da matriz (linha e coluna) não foram definidos ainda
 - `float [][] Y = new float [2][10]`
 - // Definida matriz com nro de linhas e colunas conhecidos

[illegible]

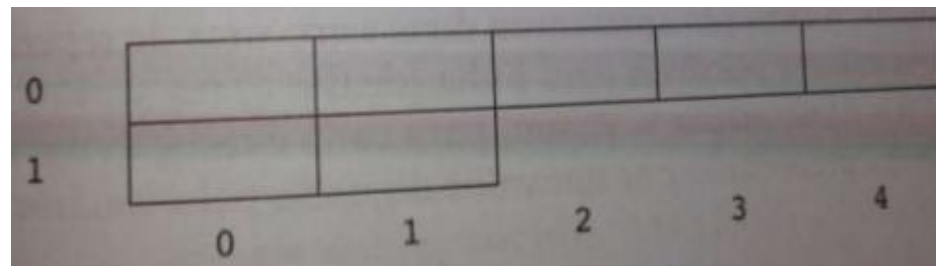
Array Multidimensional

- Exemplo de declaração de array tridimensional
 - `int y[][]`
 - // Declaração de um array de três dimensões de tamanho desconhecido
 - // Trabalha-se aqui com linha, coluna e profundidade
 - Variação aceita de declaração
 - `int [][]y;`
 - `int [][]y;`
 - Inicialização
 - `y=new int[2][3][4]`
 - //2 linha, 3 colunas e 4 profundidades



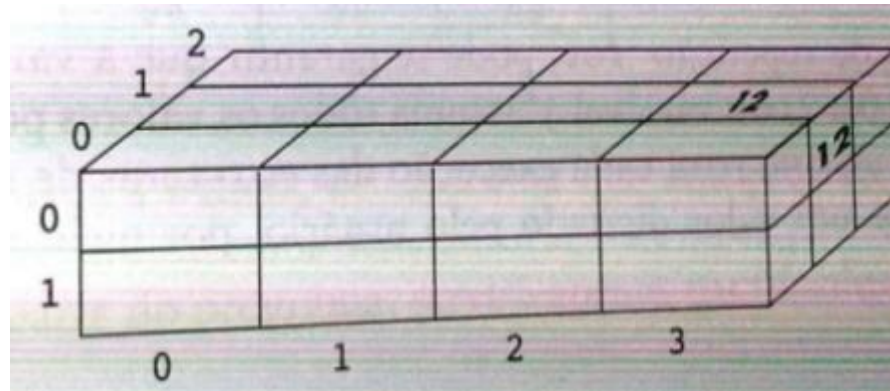
Array Multidimensional

- Declarando arrays de tamanho não uniforme
 - `int x [2][];`
 - Declarada uma matriz de 2 linhas, mas de tamanho de colunas desconhecido
 - `x[0] = new int[5];`
 - Declarada que a primeira linha possuirá 5 colunas
 - `x[1] = new int [2];`
 - Declarada que a segunda linha possuirá 2 colunas



Atribuição de valores

- Equivalente ao processos feito nos arrays unidimensionais
 - Variação depende do número de colchetes utilizados durante a declaração da variável.
- Exemplo de manipulação de uma variável tridimensional
 - $Y[0][3][1] = 12$;



Percorrendo um array

- Pode-se consultar o tamanho de cada dimensão
- Exemplo
 - Dada a seguinte definição
 - `int mat[][];`
 - `mat = new int[3][];`
 - `mat[0] = new int[2];`
 - `mat[1] = new int[5];`
 - `mat[2] = new int[3];`
 - ...
 - `for (int linha=0; linha<mat.length; linha++)`
 - `for (int coluna=0; coluna < mat[linha].length; coluna++)`
 - `Mat[linha][coluna]=linha+coluna;`

Elementos vizinhos

- Alguns programas que trabalham com arrays bidimensionais necessitam localizar elementos que estão em posições adjacentes, o que é muito comum em jogos
- Na posição $[i][j]$, os vizinhos são:

$[i - 1][j - 1]$	$[i - 1][j]$	$[i - 1][j + 1]$
$[i][j - 1]$	$[i][j]$	$[i][j + 1]$
$[i + 1][j - 1]$	$[i + 1][j]$	$[i + 1][j + 1]$

Exercício

- Considere matrizes de inteiros com l linhas e c colunas e implemente métodos para:
 - Ler todos os elementos da matriz
 - Escrever todos os elementos da matriz
 - Trocar a linha i_1 pela linha i_2 da matriz
 - Trocar a coluna c_1 pela coluna c_2 da matriz
 - Inicializar todos os elementos da matriz com valores aleatórios

Exercício

- Considere matrizes quadradas de inteiros e implemente métodos para calcular:
 - Somatório de todos os elementos da matriz
 - Somatório dos elementos da linha i da matriz
 - Somatório dos elementos da coluna j da matriz
 - Somatório dos elementos da diagonal principal da matriz
 - Somatório dos elementos acima da diagonal principal da matriz
 - Somatório dos elementos abaixo da diagonal principal da matriz
 - Somatório dos elementos da diagonal secundária da matriz
 - Somatório dos elementos acima da diagonal secundária da matriz
 - Somatório dos elementos abaixo da diagonal secundária da matriz