



Universidade Tecnológica Federal do Paraná
Câmpus Guarapuava
Curso de Tecnologia em Sistemas para Internet
Professor Eleandro Maschio
Pensamento Computacional e Fundamentos de Programação

Exercícios: Matrizes Bidimensionais

Exemplo Básico

Considere M uma matriz de L linhas e C colunas.

Declaração

```
int m[][] = new int[L][C];
```

Representação Didática da Memória Primária

	coluna 0	coluna 1	coluna 2	...	coluna C-1
linha 0	$m[0][0]$	$m[0][1]$	$m[0][2]$...	$m[0][C-1]$
linha 1	$m[1][0]$	$m[1][1]$	$m[1][2]$...	$m[1][C-1]$
linha 2	$m[2][0]$	$m[2][1]$	$m[2][2]$...	$m[2][C-1]$
...
linha L-1	$m[L-1][0]$	$m[L-1][1]$	$m[L-1][2]$...	$m[L-1][C-1]$

Percorrimento da Matriz em Totalidade

L e C tanto podem ser constantes, quanto podem ser substituídos por inteiros constantes.

```
for (int i = 0; i < L; i++)  
{  
    // Para cada linha, percorre-se todas as colunas  
    for (int j = 0; j < C; j++)  
    {  
        // Operação utilizando  $m[i][j]$   
        // Trata elementos isolados  
    }  
}
```

Exercício 1

Defina a classe `MatrizAleatoria`. Objetos desta classe possuem, como principal atributo, uma matriz inteira de l linha e c colunas, sendo l e c fornecidos ao construtor.

Exercício 2

Faça com que essa matriz seja inicializada, no construtor, com inteiros aleatórios entre 0 e 99. A implementação dos demais métodos é descrita na sequência. Os parâmetros e o retorno precisam ser inferidos.

Exercício 3

`toString()`

Retorna uma cadeia de caracteres com a representação em texto da matriz. O método `toString()` é

bastante importante em Java. Pesquise e entenda como ele funciona.

Exercício 4

somaLinhaColuna()

Altera a matriz de forma que cada elemento seja obtido por meio da soma da sua respectiva linha com a coluna.

Exercício 5

multiplicaPor()

Multiplica a própria matriz por um inteiro n , fornecido como parâmetro.

Exercício 6

somaCom()

Retorna o resultado da operação $M + N$, sendo N fornecida como parâmetro. O retorno deve ser `int[][]`.

Exercício 7

binaria()

Retorna uma matriz binária de mesma ordem, que possui zero nas posições em que a matriz original tinha elementos pares e um nas posições em que havia elementos ímpares. O retorno deve ser `int[][]`.

Exercício 8

diagonalPrincipal()

Retorna os elementos da diagonal principal. Evite passos desnecessários e observe que é fácil incorrer neles. O retorno deve ser `int[]`.

Exercício 9

diagonalSecundaria()

Idem ao anterior, mas para a diagonal secundária.

Exercício 10

trianguloSuperiorPrincipal()

Retorna uma cadeia de caracteres formatada, com os elementos do triângulo superior da diagonal principal, ou seja, aqueles situados acima dela.

Exercício 11

trianguloInferiorPrincipal()

Idem ao anterior, mas considerando o triângulo inferior da diagonal principal, ou seja, os elementos situados abaixo dela.

Exercício 12

trianguloSuperiorSecundaria()

Retorna uma cadeia de caracteres formatada, com os elementos do triângulo superior da diagonal secundária, ou seja, aqueles situados acima dela.

Exercício 13

trianguloInferiorSecundaria()

Idem ao anterior, mas considerando o triângulo inferior da diagonal secundária, ou seja, os

elementos situados abaixo dela.

Exercício 14

`transposta()`

Retorne a matriz transposta, ou seja, (M^T) . O retorno deve ser `int[][]`.

Exercício 15

`multiplicaPor()`

Retorna o resultado da operação $M * N$, sendo N fornecida como parâmetro. Analise a possibilidade de haver outro método homônimo na mesma classe. Proponha uma abordagem para o tratamento de erros, uma vez que nem sempre a multiplicação pode ocorrer. O retorno deve ser `int[][]`.

Como Citar

Todos os exercícios desta lista são autorais.

MASCHIO, Eleandro. **Exercícios: Matrizes Bidimensionais**. Guarapuava: Universidade Tecnológica Federal do Paraná, 2021. 3 p. Material didático da disciplina de Pensamento Computacional e Fundamentos de Programação.