



Processamento da Informação

Matrizes – Parte 2

Prof. Jesús P. Mena-Chalco
CMCC/UFABC

Q2/2018

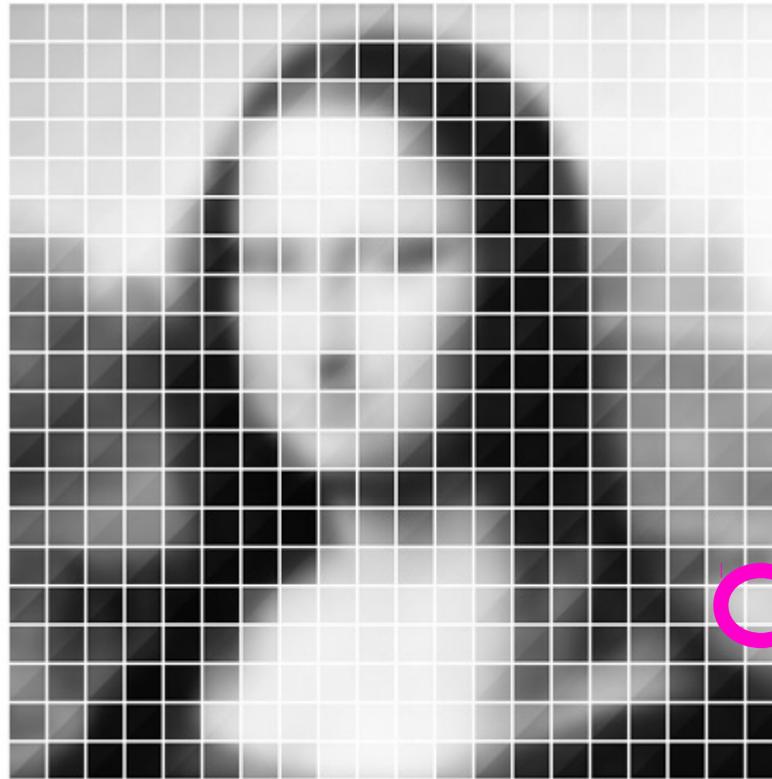
Matriz bidimensional em Java

Declaração de uma variável que representa uma **matriz bidimensional de inteiros** (10 linhas e 20 colunas)

```
int matriz[][];  
matriz = new int[10][20];
```

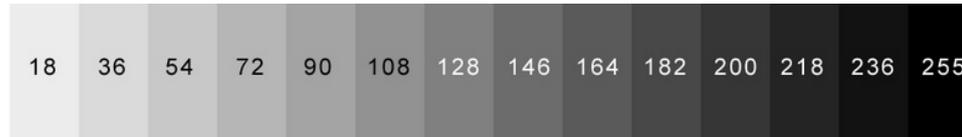
```
int matriz[][] = new int[10][20];
```

Matriz bidimensional (imagem em níveis de cinza)

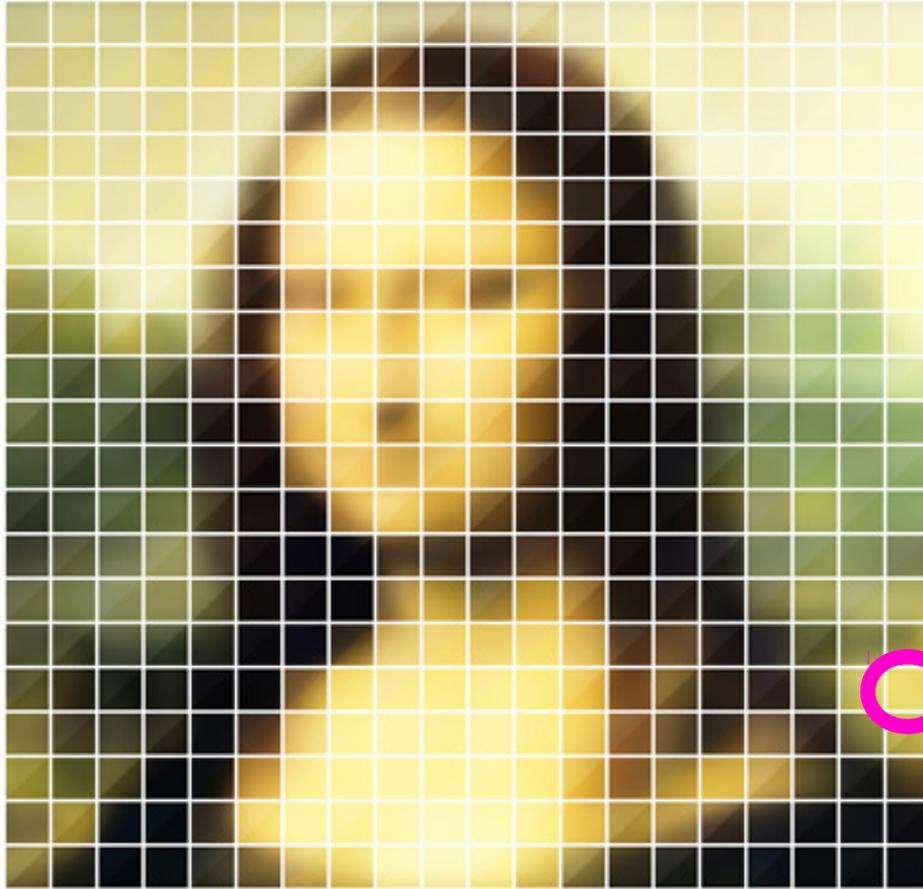


Nível=18

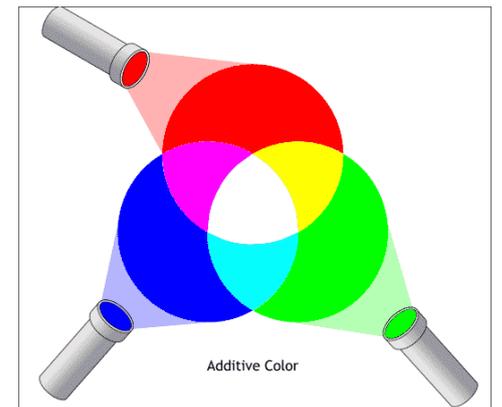
0 18 36 54 72 90 108 128 146 164 182 200 218 236 255



Matriz tridimensional (imagem em RGB)

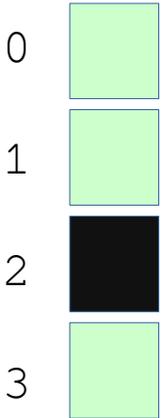


→ {Red, Green, Blue}



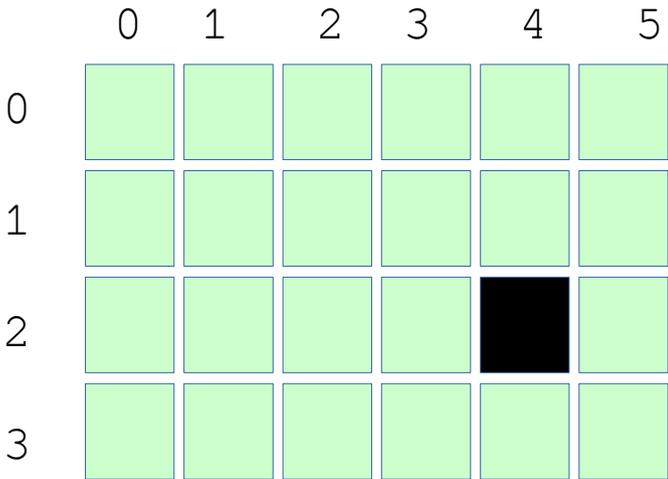
Matrizes

Matriz unidimensional (vetor/Array)



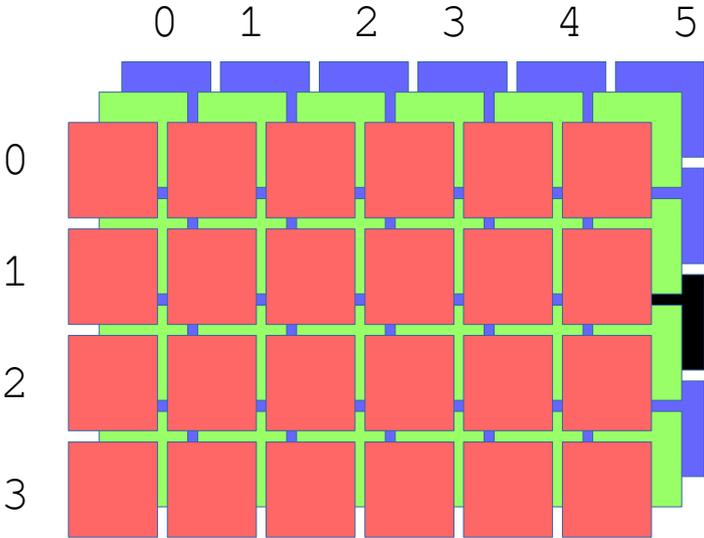
`M[2]`

Matriz bidimensional (2D)



`M[2][4]`

Matriz tridimensional (3D)



`M[2][5][2]`

Exemplo: Matriz diagonal

Vamos desenvolver um método que permita **verificar** se uma matriz é diagonal.

Uma matriz diagonal é uma matriz quadrada onde os elementos que não pertencem à diagonal principal **são obrigatoriamente** iguais a zero.

Assinatura:

```
static boolean ehDiagonal( int M[] [] )
```

$$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 11 \end{pmatrix}$$

Pode ser também `[][]M`

Exemplo: Matriz diagonal

```
static boolean ehDiagonal( int [][]M ) {
    int i, j;
    int n = M.length;
    int m = M[0].length;

    if (n!=m) {
        return false;
    }
    else {
        for (i=0; i<n; i=i+1) {
            for (j=0; j<n; j=j+1) {
                if (i!=j && M[i][j]!=0) {
                    return false;
                }
            }
        }
        return true;
    }
}
```

Exemplo: Matriz transposta

Vamos desenvolver um método que permita calcular a matriz diagonal de uma matriz dada como entrada.

Assinatura:

```
static int [] [] transposta ( int [] [] M )
```

$$\begin{bmatrix} 6 & 4 & 24 \\ 1 & -9 & 8 \end{bmatrix}^T = \begin{bmatrix} 6 & 1 \\ 4 & -9 \\ 24 & 8 \end{bmatrix}$$

Exemplo: Matriz transposta

```
static int[][] transposta( int [][]M ) {
    int i, j;
    int n = M.length;
    int m = M[0].length;
    int [][]R = new int[m][n];

    for (i=0; i<n; i=i+1) {
        for (j=0; j<m; j=j+1) {
            R[j][i] = M[i][j];
        }
    }

    return R;
}
```

Exercício

Escreva uma função que, recebendo uma Matriz bidimensional de números inteiros, verifica se existem ao menos duas linhas idênticas.

Assinatura:

```
static boolean linhasIdenticas ( int M[] [] )
```

Exemplos:

1	1	1
2	2	2
3	3	3

false

1	2	3
1	2	3
1	2	3

true



Atividade em aula

Questão 1 – a

```
static boolean elemento8 ( int [][]M ) {
    int i, j;
    int n = M.length;
    int m = M[0].length;

    for (i=0; i<n; i=i+1) {
        for (j=0; j<m; j=j+1) {
            if (M[i][j]==8) {
                return true;
            }
        }
    }

    return false;
}
```

Questão 1 – b

```
static boolean elemento8elegante ( int [][]M ) {
    int i, j;
    int n = M.length;
    int m = M[0].length;
    boolean resposta = false;

    for (i=0; i<n && resposta==false; i=i+1) {
        for (j=0; j<m && resposta==false; j=j+1) {
            if (M[i][j]==8) {
                resposta = true;
            }
        }
    }

    return resposta;
}
```

Questão 1 – c

```
static int somaDiagonais ( int [][]M ) {  
    int i, j, soma=0;  
    int n = M.length;  
  
    for (i=0; i<n; i=i+1) {  
        soma = soma + M[i][i] + M[i][n-i-1];  
    }  
  
    return soma;  
}
```

Wally



Questão 2

```
static void wally ( int [][]M ) {
    int i, j, wi=-1, wj=-1;
    int n = M.length;
    int m = M[0].length;

    for (i=1; i<n-1; i=i+1) {
        for (j=1; j<m-1; j=j+1) {
            if (M[i][j]==1111 && M[i][j-1]==0 && M[i][j+1]==0 && M[i-1][j]==4 && M[i+1][j]==8 ) {
                wi=i;
                wj=j;
            }
        }
    }
    if (wi!=-1){
        System.out.println(wi+" "+wj);
    }
    else {
        System.out.println("Wally nao esta na matriz");
    }
}
```