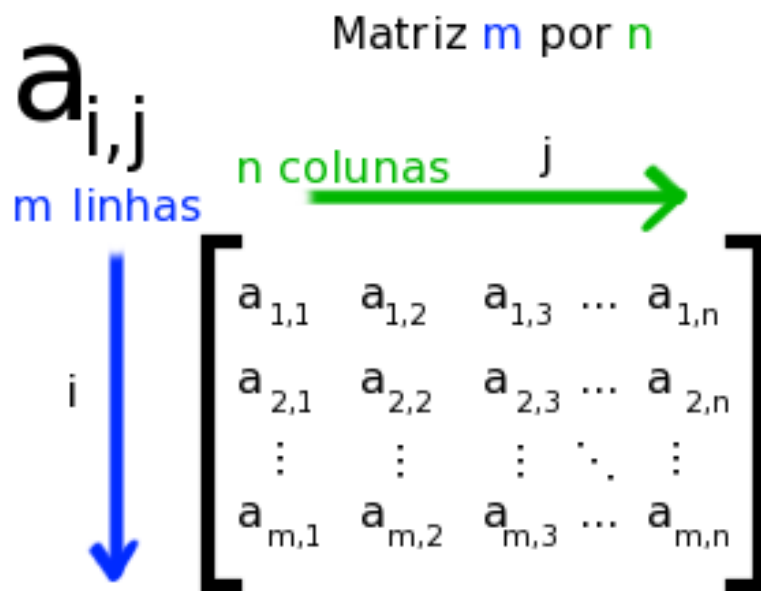


Matrizes

Fabio Lubacheski
fabio.aglubacheski@sp.senac.br

Matrizes

Em matemática uma **matriz** é um conjunto retangular de números, símbolos ou expressões, organizados em linhas e colunas. Cada um dos itens de uma matriz é chamado de **elemento**.



Matrizes

Para representar uma matriz em uma linguagem de programação utilizamos o mesmo conceito de vetores, com a diferença que um vetor armazena informações com indexação em uma única dimensão (índice), também chamado de **array unidimensional**.

Uma **matriz** estende o conceito de vetor, pois uma informação é referenciada por dois índices, um para cada dimensão (**array bidimensional**)

Matriz	0	1	2
0			
1			
2			

Matrizes

A declaração de matrizes em Java é bastante simples:

```
tipo nomeMatriz[ ][ ] = new tipo [linhas][colunas];
```

Exemplos:

```
int A[ ][ ] = new int[2][3];
```

```
char M[ ][ ] = new char[3][2];
```

Matrizes também podem ser declaradas e inicializadas junto com a declaração. Neste contexto, o tamanho da matriz é definido pela inicialização. Para os dois exemplos acima teríamos:

```
int A[ ][ ]={{0,4,5},{-5,6,8}};
```

```
char M[ ][ ]={{'A','B'},{'C','D'},{'E','F'}};
```

Matrizes

O acesso aos elementos de uma matriz é feito através de dois índices: um para a linha e, outro, para a coluna. Por exemplo, considerando-se a matriz A , o acesso $A[0][1]$ irá recuperar o valor 4 (linha 0, coluna 1).

De forma similar, podemos ter *arrays* de qualquer dimensão:

$A[\][\][\], A[\][\][\][\], \dots$. Para cada dimensão, utilizamos um índice de acesso.

Matrizes

Considere a declaração abaixo:

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

Para sabermos o numero de linhas da matriz basta utilizar a propriedade `length` da matriz:

`A.length` temos o número de linhas da matriz

`A[0].length` temos o número de colunas da primeira linha, lembrando que todas as linhas temos o mesmo número de colunas

Matrizes

Assim. se quisermos inicializar todas as posições da matriz com zero deveríamos fazer o seguinte:

```
for( int i=0;i<A.length;i++) //anda na linha
    for( int j=0;j<A[0].length;j++) // anda na coluna
        A[i][j]=0;
```

E para imprimir os elementos da matriz teríamos?

```
for( int i=0;i<A.length;i++) //anda na linha
    for( int j=0;j<A[0].length;j++) //anda na coluna
        System.out.println("A["+i+"]["+j+"]:"+A[i][j]);
```

Exercícios

- 1) Dado a matriz $A_{n \times m}$, faça uma função que recebe a matriz $A_{n \times m}$ por parâmetro, em seguida a função aloca e devolve sua transposta A^t , onde $A[i][j] = A^t[j][i]$ para qualquer i e j .

Exemplo, se $A_{3 \times 2}$

$$\begin{bmatrix} 0 & 6 \\ -1 & 2 \\ 5 & 0 \end{bmatrix}$$

a matriz transposta representada por $A^t_{2 \times 3}$, será

$$\begin{bmatrix} 0 & -1 & 5 \\ 6 & 2 & 0 \end{bmatrix}$$

- 2) Escreva uma função que receba uma matriz $n \times m$ de números inteiros e devolva o maior valor presente nesta matriz;
- 3) O traço de uma matriz é a soma dos elementos de sua **diagonal principal**. Implemente uma função que receba uma matriz quadrada (número de linhas = número de colunas) e devolva o seu traço.

Exercícios

- 4) Dizemos que uma matriz quadrada A é simétrica se e somente se $A[i][j] = A[j][i]$.

Implemente uma função para verificar se uma matriz de números inteiros é simétrica, se a matriz for simétrica sua função retorna true e false caso contrário.

- 5) Escreva uma função que recebe por parâmetros duas matrizes, A e B , com n linhas e m colunas.

Sua função deve calcular a soma de $A + B$ e armazena na matriz $C_{n \times m}$ e ao final retornar a matriz C .

- 6) Dadas a matriz $A_{n \times n}$ e o vetor B com n elementos, calcule a multiplicação de A por B , e armazene na matriz $C_{n \times 1}$.

- 7) Dadas duas matrizes $A_{m \times n}$ e $B_{n \times p}$. Obter a matriz $C_{m \times p}$ onde $C = A \times B$.

Fim