

# Estruturas de dados bi-indexadas

- Declaração em Java de ARRAYS unidimensionais e bidimensionais

- unidimensionais** (Vector)

- `int [ ] vec;    float [ ] med;    char [ ] nom;`

- bidimensionais** (Matrizes ou Tabelas)

- `int [ ][ ] mat;`

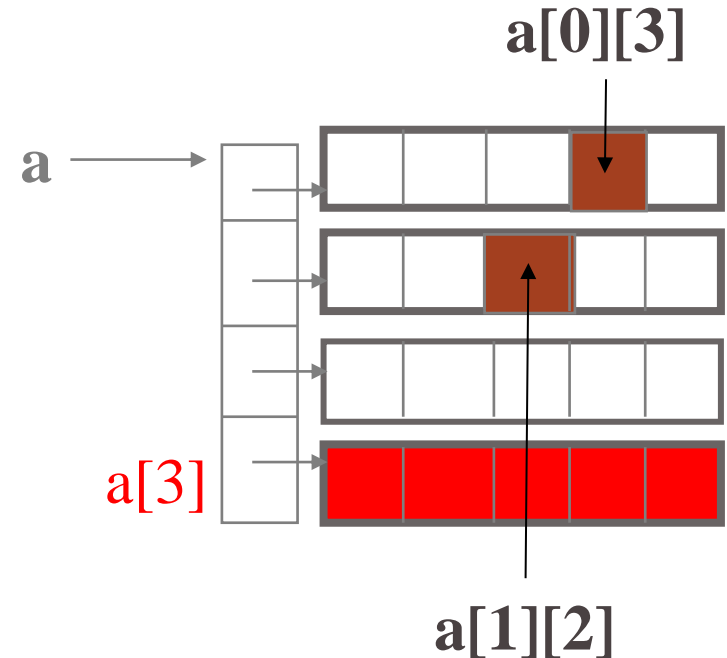
- `double [ ][ ] tab1;`

- Criar um array bidimensional (matriz)

- `int [ ][ ] a = new int[4][5];`

- Em Java

- os arrays multidimensionais são arrays de arrays



# Estruturas de dados bi-indexadas

Criar um array bi-indexado (Matriz)

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

```
//Criar e iniciar array matriz
```

```
int [ ][ ] c={{1,2},{3,4}};
```

Um array multi-indexado pode ser alocado dinamicamente

```
int [ ][ ] a;  
a=new int[2 ][ ];  
a[0]=new int[5];  
a[1]=new int[3];
```

# Manipulação de Matrizes

- Criar matriz
- Inserir elementos
- Atualizar elemento
- Pesquisar elemento
- Listagens várias
- Ordenações várias

# Manipulação de Matrizes - Exemplo

Pretende-se um programa que:

- Crie uma matriz de  $n$  linhas por  $m$  colunas;
- Insira via teclado elementos na matriz;
- Liste esses elementos da matriz.

Desenvolvimento modular do programa.

Módulo 1 - inserirElementos (mat, linhas, cols)

Módulo 2 - listarMatriz (mat, linhas, cols)

# Manipulação de Matrizes - Exemplo

```
public class Matriz {  
  
    static Scanner ler= new Scanner(System.in);  
  
    public static void inserirElementos(  
        int[ ][ ] mat, int li,int col){... }  
  
    public static void listarMatriz(  
        int[ ][ ] mat, int li, int col){... }  
  
    public static void main(String[] args) {...}  
}
```

# Manipulação de Matrizes - Exemplo

```
public static void main(String[] args) {  
  
    int [ ] [ ] mat;  
    int linhas, colunas;  
  
    System.out.println (Qual o número de linhas da matriz ?");  
    linhas=ler.nextInt();  
    System.out.println (Qual o número de colunas da matriz ?");  
    colunas=ler.nextInt();  
  
    mat=new int[linhas] [colunas];  
  
    // Preencher a matriz com nLinhas e nColunas  
    // Se for para preencher todos os elementos não são necessários  
    // os parâmetros nLinhas e nColunas  
    inserirElementos( mat, nlinhas,ncolunas);  
  
    listarMatriz(mat, nLinhas,nColunas);  
}
```

# Manipulação de Matrizes - Exemplo

```
public static void inserirElementos(int[ ][ ] m, int l, int c) {  
  
    for (int i=0; i< l; i++){  
        for (int j=0; j< c; j++){  
            System.out.printf ("%d,%d=",i+1,j+1);  
            m[i][j]=ler.nextInt();  
        }  
    }  
}
```

# Manipulação de Matrizes - Exemplo

```
public static void listarMatriz( int[ ][ ] m, int l, int c) {  
  
    System.out.println("Listagem da matriz");  
  
    for (int i=0; i< l; i++){  
        for (int j=0; j< c; j++){  
            System.out.printf("%d\t",m[i][j]);  
        }  
        System.out.printf("\n");  
    }  
}
```



# Manipulação de Matrizes

- Considere o seguinte excerto de código.  
Qual será a saída?

```
int [][] matriz1=new int[10][20]
```

```
System.out.printf("Nº linhas: %d\n",matriz1.length);
```

10

```
System.out.printf("Nº colunas: %d",matriz1[0].length);
```

20