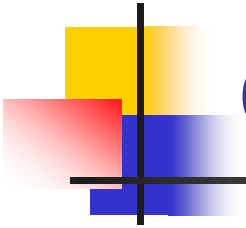
Matriz: vetores multidimensional

Comparando vetor x matriz

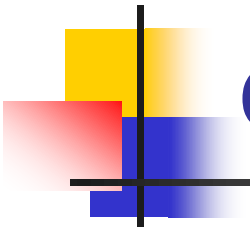
LPA

Integrado em Informática -IFSP - Hortolândia



Objetivo

- Ao final desta aula você deve responder: Sobre Matriz
  - Compreender que matriz nada mais é que um vetor multidimensional (vetor onde cada elemento é um vetor).
  - Qual a sua importância como **estrutura de dados** para armazenar informações.
  - Como **declarar**, **inicializar** e **iterar**(percorrer) matriz.
  - Como utilizar a instrução **for** encadeada (for com outro for para iterar sobre a matriz. E **for aprimorado** encadeado.
  - Saber realizar as operações matemáticas de Matriz de modo computacional.



Conteúdo

- Matriz: declaração, inicialização e manipulação de matriz
- Matriz com vetor multidimensional

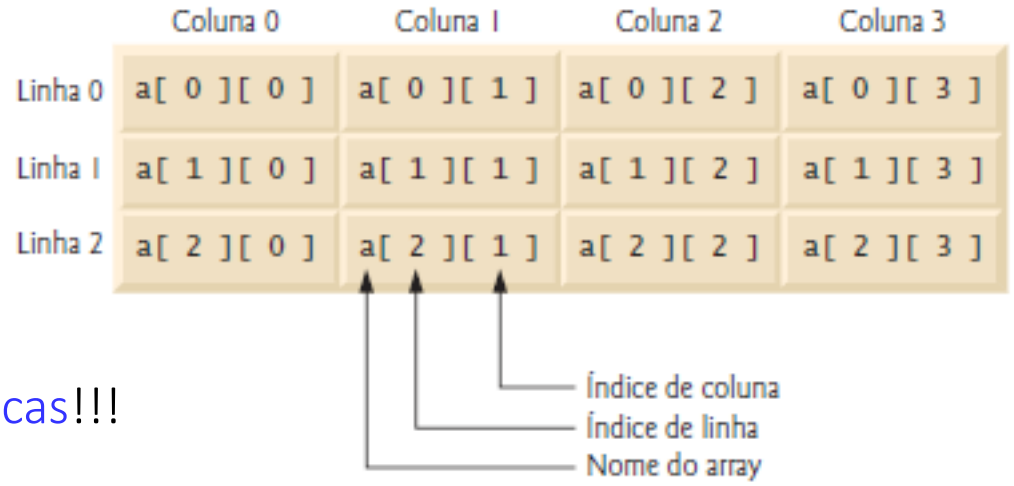
Cap. 7 Array e ArrayLists – Java Como Programar (Deitel & Deitel)

Matriz: conceito

- A matriz ou vetores multidimensionais com duas dimensões costuma ser usados para representar tabela de valores consistindo em informações organizadas em linhas e colunas.
- *Para identificar um elemento da* tabela, devemos especificar dois índices. Por convenção o primeiro identifica **linha** do elemento e o segundo sua **coluna**.
- Os vetores com dois índice, são chamados de bidimensional (vetor multidimensional podem ter mais de duas dimensões).

- Vetor bidimensional

chamado a **com 3 linhas e 4 colunas**



- Matriz é uma caso especial de

vetor, mas tem as mesma características!!!

Declaração

vetor(unidimensional) x matriz(bidimensional)

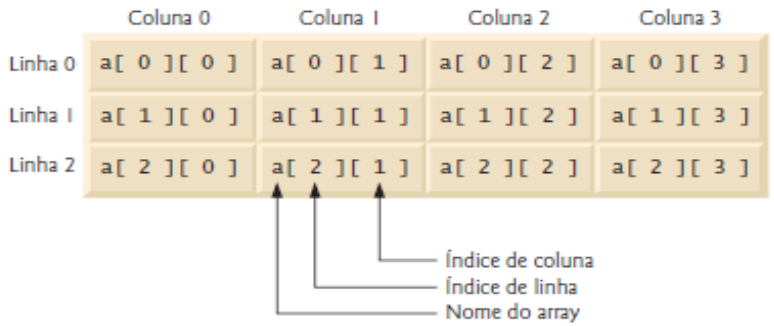
- Declaração **vetor**:

  int [] c ;

c[0]
c[1]
c[2]
c[3]
c[4]
c[5]
c[6]
c[7]

- Declaração **matriz** :

  int [] [] a ;



- O *vetor como matriz é* um objeto.
- Os elementos dos vetores e matrizes podem ser de tipos primitivos ou *tipos por referência.*
- O índice precisa ser um valor int ou um valor de tipo que possa ser promovido a int (byte, short, char). ***E o primeiro índice é 0.***

Declaração e instanciação (criação)

vetor(unidimensional) x matriz(bidimensional)

- Declara a variável **c** como referência para um vetor de inteiro.

  int [ ] **c**;

- Cria/instância o vetor e atribui a variável **c** como referência para um vetor de inteiro.

  c = **new** int [8];

- DECLARA e INSTÂNCIA o vetor `

  int [ ] **c** = new int [8];

c[0]
c[1]
c[2]
c[3]
c[4]
c[5]
c[6]
c[7]

- Declara a variável **a** como referência para uma matriz de inteiro.

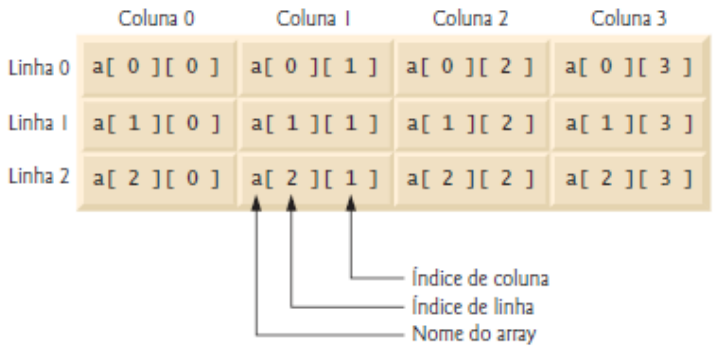
  int [ ] [ ] **a**;

- Cria/instância a matriz e atribui a variável **a** como referência para a matriz de inteiro 3 (linha) x 4 (coluna) .

  a= **new** int [3] [4];

- DECLARA e INSTÂNCIA o vetor

  int [ ] [ ] **a** = new int [3][4];



Criação: valores inicializados

vetor(unidimensional) x matriz(bidimensional)

- **Lembrando**: o vetor e matriz são objetos, portanto na sua criação os seus valores são inicializados pela JVM (**tempo de execução**) com valores de acordo com o seu tipo.

```
int [ ] c = new int[8] ;
```

c [0]	??
c [1]	??
c [2]	??
c [3]	??
c [4]	??
c [5]	??
c [6]	??
c [7]	??

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

m [0] [0] ??	m [0] [1] ??	m [0] [2] ??	m [0] [3] ??
m [1] [0] ??	m [1] [1] ??	m [1] [2] ??	m [1] [3] ??
m [2] [0] ??	m [2] [1] ??	m [2] [2] ??	m [2] [3] ??

Qual o valores no vetor c e matriz m, após a execução da linha anterior?

E seu o tipo ao invés de int fosse: float, char, objeto, boolean?

Manipulação dos elementos

vetor(unidimensional) x matriz(bidimensional)

- Os elementos do vetor/matriz são **acessados ou alterados** por meio dos índices que indicam a posição do elemento que deseja manipular.

```
int [ ] c = new int[8];
```

- Para deixar o vetor conforme a fig.

```
c [ 0 ] = -4;
c [ 1 ] = 6;
.....
c [ 7 ] = 3;
```

c [0]	-4
c [1]	6
c [2]	0
c [3]	72
c [4]	64
c [5]	53
c [6]	78
c [7]	3

Como faz para mostrar o elemento que esta na 3 posição? Qual o seu índice?

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

- Para deixar o vetor conforme a fig.

```
m [ 0 ] [ 0 ] = 2;
m [ 0 ] [ 1 ] = 4;
.....
m [ 2 ] [ 3 ]= 24;
```

m [0] [0] 2	m [0] [1] 4	m [0] [2] 6	m [0] [3] 8
m [1] [0] 10	m [1] [1] 12	m [1] [2] 14	m [1] [3] 16
m [2] [0] 18	m [2] [1] 20	m [2] [2] 22	m [2] [3] 24

Como faz para mostrar o elemento que esta na segunda linha e terceira coluna? Qual os seus índices?

Inicialização

vetor(unidimensional) x matriz(bidimensional)

- Pode-se criar um vetor/matriz e inicializar seus elementos com uma declaração de inicialização:

int c [ ] = { -4 , 6 , 0 , 72 , 64, 53 , 78, 3 };

c [0]	-4
c [1]	6
c [2]	0
c [3]	72
c [4]	64
c [5]	53
c [6]	78
c [7]	3

int m[ ][ ] = { { 2,4,6,8 } , { 10,12,14,16 } , { 18,20,22,24 } };

m [0] [0] 2	m [0] [1] 4	m [0] [2] 6	m [0] [3] 8
m [1] [0] 10	m [1] [1] 12	m [1] [2] 14	m [1] [3] 16
m [2] [0] 18	m [2] [1] 20	m [2] [2] 22	m [2] [3] 24

- A operação apropriada do new ocorre nos *"bastidores"* pela JVM.

Manipulação

vetor(unidimensional) x matriz(bidimensional)

- Para iterar sobre o vetor/matriz utilizamos a estrutura: **for.**
- Para VETOR: for simples e MATRIZ: **for aninhado** (for dentro de outro).

```
for ( int i = 0; i < c.length ; i++)
    System.out.printf("%d %d", i , c[i]);

// somar todos os elementos do vetor
int soma = 0;
for ( int i = 0; i < c.length ; i++)
   soma = soma + c[ i ]); // soma += c [ i ]
```

c[0]	-4
c[1]	6
c[2]	0
c[3]	72
c[4]	64
c[5]	53
c[6]	78
c[7]	3

```
for ( int linha= 0; linha < 3 ; linha ++)   // linhas
    for ( int col = 0; col < 4 ; col++)  // coluna
        System.out.printf( "%d ",  m [ linha ] [ col ]);

    System.out.println(); // nova linha da matriz
}

for ( int linha= 0; linha < 3 ; linha ++)   // linhas
    for ( int col = 0; col < 4 ; col++)  // coluna
      soma = soma +  m[ linha ] [col] );
```

2	4	6	8
10	12	14	16
18	20	22	24

Passando como parâmetro

vetor(unidimensional) x matriz(bidimensional)

Índice	valor
c [0]	7
c [1]	6
c [2]	0
c [3]	7
c [4]	6
c [5]	5

c

d

```
private int c [ ] = { 7,6,0,7,6,5};

public void manipulaVetor() {
   ManipulaVetor  v  = new ManipulaVetor();
    .....
   v. mostra( c ) ;
}
```

c

```
public void mostra(int [ ]  d ){
   .....

}
```

m [0] [0] 2	m [0] [1] 4	m [0] [2] 6	m [0] [3] 8
m [1] [0] 10	m [1] [1] 12	m [1] [2] 14	m [1] [3] 16
m [2] [0] 18	m [2] [1] 20	m [2] [2] 22	m [2] [3] 24

m

n

```
private int m [ ] [ ] = { { 2,4,6,8 } , { 10,12,14,16 } , { 18,20,22,24 } };

public void manipulaVetor() {
   ManipulaVetor  v  = new ManipulaVetor();
   .....
   v. mostra( m ) ;
}
```

m

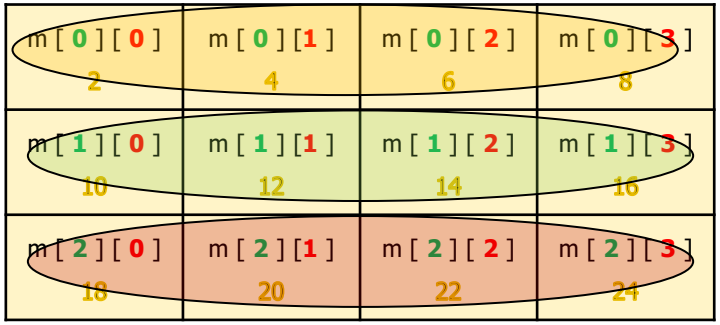
```
public void mostra(int [ ] [ ] n ){
   .....


}
```

Java e a representação de vetor multidimensional

- O Java não suporta vetores multidimensionais diretamente, mas permite especificar vetores unidimensionais cujos elementos também são vetores unidimensionais, alcançando assim o mesmo efeito.
- int m[ ] [ ] = new int [3] [4]

int m[ ][ ] = { { 2,4,6,8 } , { 10,12,14,16 } , { 18,20,22,24 } };

m = [v1 v2 v3]



- Java interpreta: m é um vetor de três elementos (linhas), onde cada elemento é um novo vetor com quatro elementos(colunas).

Manipulando Matriz de modo Genérico

- O método abaixo, serve apenas para matriz[3][4]

2	4	6	8
10	12	14	16
18	20	22	24

```
public void mostra( int m [ ] [ ] ) {

    for ( int linha= 0; linha < 3 ; linha ++) {  // linhas

        for ( int col = 0; col < 4 ; col++)  // coluna
            System.out.printf( "%d ",  m [ linha ] [ col ]);

        System.out.println(); // nova linha da matriz
    } // fim for externo
} // fim metodo
```

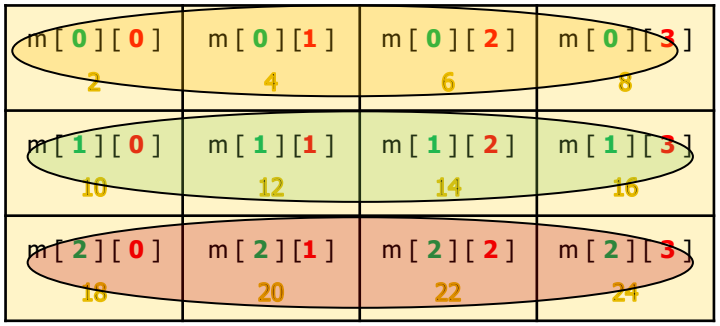
Manipulando Matriz de modo Genérico

m.length

row 0 **m**[0].length

row 1 **m**[1].length

row 2 **m**[2].length



2	4	6	8
10	12	14	16
18	20	22	24

```
// Mostra todos os elementos de uma matriz
// linhas loop pelas linhas do vetor
for ( int linha = 0; linha < m.length ; linha++)  // linhas
{
    // linhas loop pelas colunas da linha atual
    for (int coluna = 0; coluna <  m[linha].length ; coluna++)  // coluna
         System.out.printf( "%d ", m[linha][coluna] );

    System.out.println();
}
```

Manipulando Matriz de modo Genérico

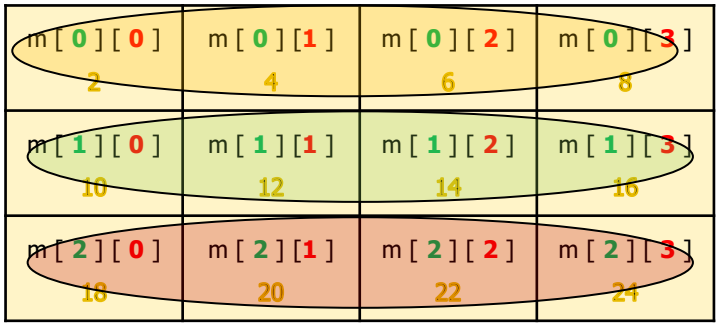
for aprimorado alinhado

array.length

row 0 m[0].length

row 1 m[1].length

row 2 m[2].length




```
//assume que o primeiro elemento da matriz é a menor valor
int menorValor = m[0][0];

for ( int[] valoresLinha: m)  // valoreslinhas recebe os valores das linha
{
     for ( int valor: valoresLinha)  // coluna
     {
          // se valor for menor que menorValor atribui valor para menorValor
          if( valor < menor )
               menor = valor;
     }
}
return menor;
```

Matrizes (vetores bidimensionais) com linhas de diferentes tamanhos.

```
int[][] b = {{1, 2}, {3, 4, 5}};
```

Cria vetor de inteiros `b` com dois elementos (determinados pelo número de inicializadores do vetor aninhados) que representam as linhas do vetor bidimensional. Cada elemento de `b` é uma *referência* a um vetor unidimensional de variáveis `int`. O vetor `int` da linha `0` é um vetor unidimensional com *dois* elementos (`1` e `2`), e o vetor `int` da linha `1` é um vetor unidimensional com *três* elementos (`3`, `4` e `5`).

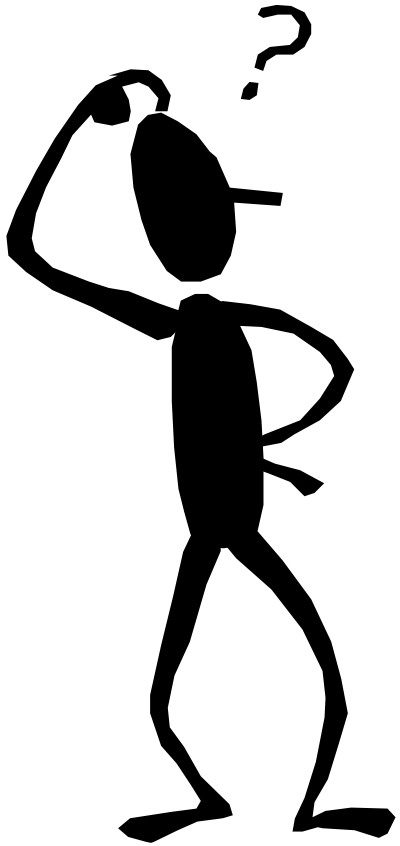
```
int[][] b = new int[2][];   // cria 2 linhas
b[0] = new int[5]; // cria 5 colunas para a linha 0
b[1] = new int[3]; // cria 3 colunas para a linha 1
```

Pode-se criar um vetor multidimensional em que cada linha tem um número *diferente* de colunas. As instruções anteriores criam um vetor bidimensional com duas linhas. A linha 0 tem *cinco* colunas, e a linha 1, *três* colunas.

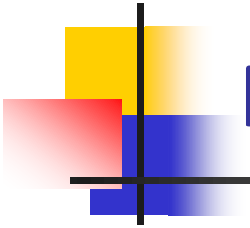
Gráfico de barra (deitado) para frequência

```
Distribuição das Notas
Nota  Quantidade
   0:     0 -
   1:     0 -
   2:     1 -*
   3:     2 -**
   4:     3 -***
   5:     4 -****
```

```
System.out.println( "Distribuicao das Notas" ); // cabecalho
System.out.println( "Nota  Quantidade" ); // cabecalho

for ( int contador = 0; contador < freq.length ; contador++) {

     System.out.printf( "\n%4d:%6d -", contador, freq[ contador ] );

     // mostra a qtd de estrela de acordo com a qtd existente em freq
     for ( int estrela = 0; estrela < freq[contador] ; estrela++)
         System.out.print("*");
}
```



FIM!!!!



Próximas aulas

- Revisão de métodos e atributos static.
- Properties para tradução.