Disciplina: LPOO

Prof. Anderson V. de Araujo

Aula 05: Vetores e Matrizes (*arrays*)

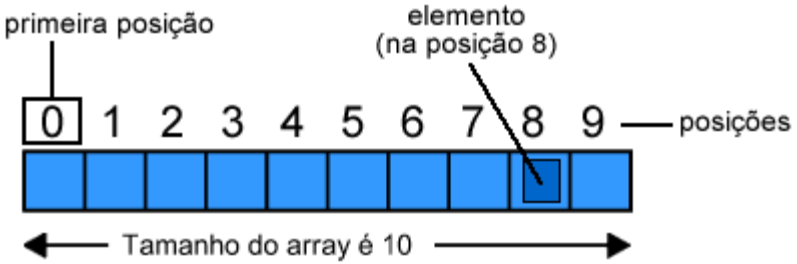
andvicoso@facom.ufms.br
http://prof.facom.ufms.br/~andvicoso/

Arrays

- Em Java, *arrays* ou vetores, são **objetos** que armazenam diversas variáveis do mesmo tipo
  - Variável de referência (objetos?)
- Qualquer tipo permitido em Java pode ser armazenado em um *array*:
  - Tipos primitivos
  - Referências de objetos
  - Outros *arrays*

Arrays

- O tamanho de um *array* precisa ser definido quando este é criado
- Cada item em um *array* é chamado de elemento, e cada elemento é acessado pela posição numérica (índice)
- As posições são numeradas a partir do zero
- O nono elemento, por exemplo, é acessado na posição 8



Utilizando os Arrays

- Para utilizar um array é necessário seguir os três passos abaixo:
  - Declaração
  - Construção
  - Inicialização

Declarando um Array

- A declaração de um array diz ao compilador o **nome do array** e o **tipo de elemento** que será armazenado
- Nenhuma memória é alocada no momento da declaração

Declarando um Array

- Regra:
  - `<tipo_do_array>[] <nome_da_variável>`
    - Ou `<tipo_do_array> <nome_da_variável>[]`
- Exemplos:
  - Declarando um array de tipos primitivos:
    - **`int[] array`** ou `int array[]`
  - Declarando um array que armazena variáveis de referência:
    - `User[] users;`

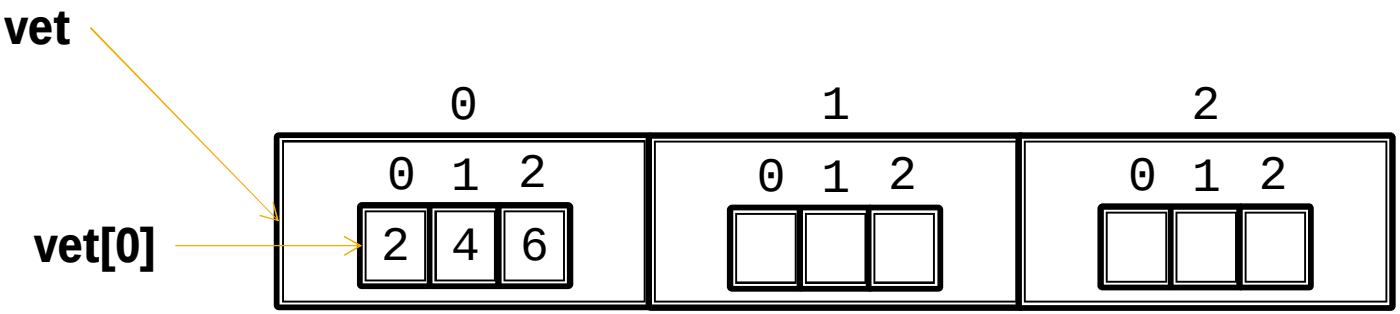
Declarando um Array

- **Não é válido** incluir o tamanho do array na declaração:
  - `int[5] pontuacao;`
- Declarando um array multidimensional (um array com outros arrays):
  - `2d (“matriz”): char[][] caracs;`
  - `3d (“cubo”): String[][][] nomes;`
    - ***Curiosidade:** A JVM pode aceitar até 255 dimensões em um vetor

Matrizes x Vetores

- Na verdade, **matrizes são vetores dentro de vetores**:

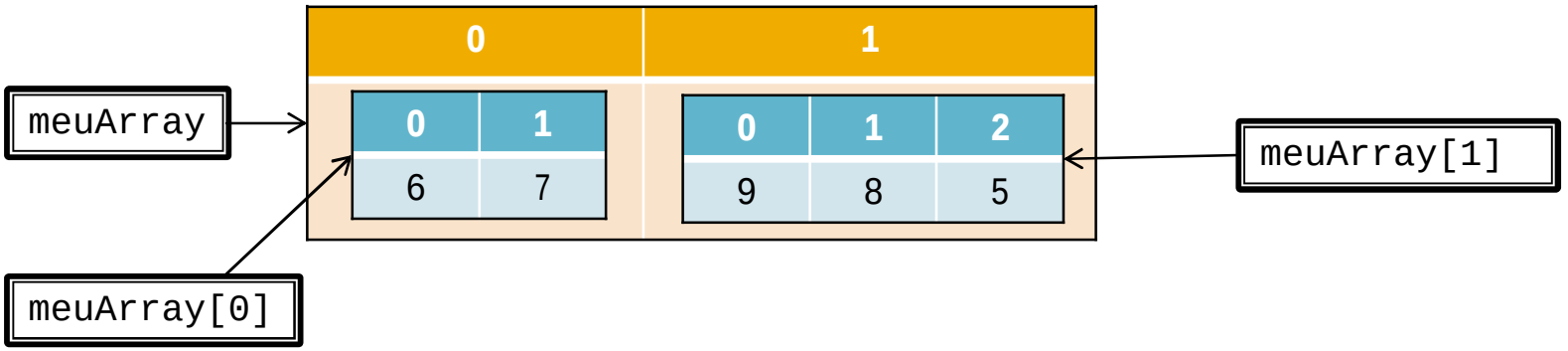
```
int[][] vet = new int[3][3];
vet[0][0]=2;
vet[0][1]=4;
vet[0][2]=6;
```



Uma matriz não é bem uma matriz...

- Declarando, construindo, atribuindo array 2-D

```
int[][] meuArray = new int[2][];// variável de referência do array 2-D
meuArray[0] = new int [2];//o primeiro elemento do array
                          //tem um array de int de 2 elem
meuArray[0] [0] = 6;
meuArray[0] [1] = 7;
meuArray[1] = new int [3];//o segundo elemento do array
                          //tem um array de int de 3 elem
meuArray[1] [0] = 9;
meuArray[1] [1] = 8;
meuArray[1] [2] = 5;
```



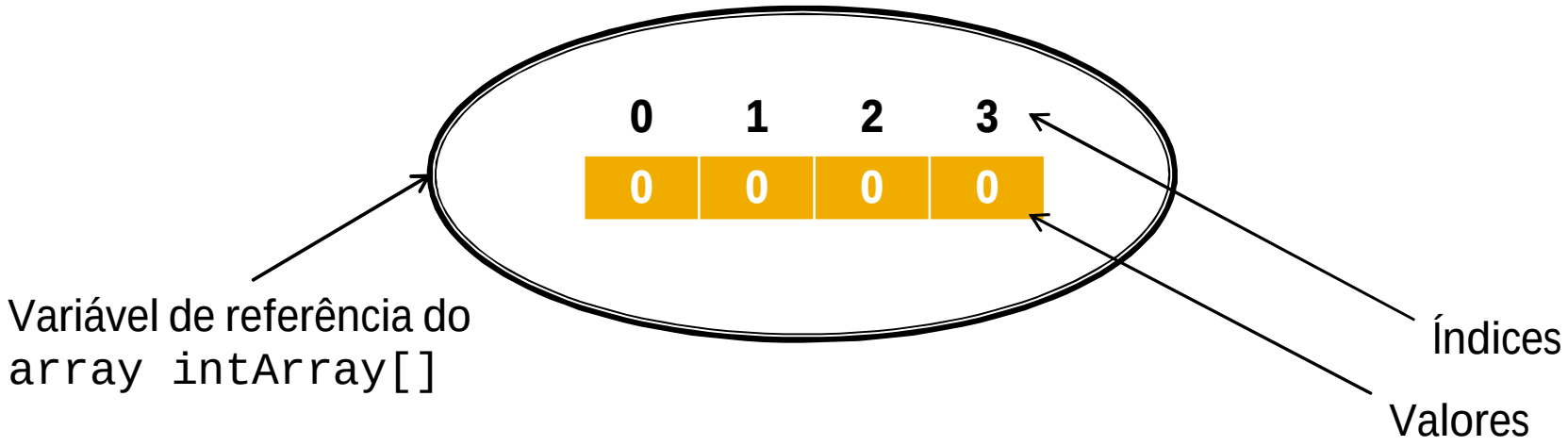
Construindo um Array

- Para criar um array, o Java terá que **saber quanto espaço alocar**
  - É preciso especificar o tamanho do array no momento da construção
- Esse tamanho será igual a **quantidade máxima de elementos** que o array poderá armazenar

Construindo um Array

- Como o array aparecerá na memória:

```
int[] intArray; //Declarando array unidimensional
intArray = new int [4]; //Construindo
```



Construindo um Array

- Uma vez definido o **tamanho** do array este **não pode** mais **ser alterado**
- Quando o array é de referências para objetos, **somente a memória ocupada pela referência é alocada**
- **Nenhum objeto** é criado neste momento

Construindo um Array

- Declarando array unidimensional
  - `int[] intArray;`
- Declarando e construindo um array unidimensional
  - `intArray = new int [4];`
- Declarando e construindo um array bidimensional
  - `String [][] stringMatrix = new String[10][20];`

Construindo um Array

- Quando um array é **construído**, seus valores são **automaticamente** inicializados para **valores padrão**
- Similar aos valores padrão para variáveis de instância

VALORES PADRÃO

Tipo	Valor Inicial
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
boolean	false
Referência/objeto	null

Inicializando um Array

- Inicializar um array significa atribuir (inserir) itens a ele
- O menor índice do array é sempre 0 (zero)
- O maior índice do array é obtido através de:
  - **array.length - 1**

Inicializando um Array

- Declarando, construindo e inicializando um array 1-D

```
int[] umArray; // declarando
umArray = new int[10]; // construindo

for (int i = 0; i < umArray.length; i++) {
     umArray[i] = i; // inicializando (atribuindo)
     System.out.print(umArray[i] + " ");
}
```

Declarando e Inicializando de uma só vez

```
int[] vet = {2,4,6,8,10};
```

É equivalente a:

```
int[] vet = new int[5];
vet[0]=2;
vet[1]=4;
vet[2]=6;
vet[3]=8;
vet[4]=10;
```

Declarando e inicializando de uma só vez

```
Object [] objArray = {"Objeto1", "Objeto2"};

boolean[] answers = {true, false, true, true};

double[][] mat = {{ 1, 2, 3 }, { 1, 2, 3 }};
```

```
class CriandoArray{
    public static void main(String[] args){
            String[] lista = {"Um", "Dois"};

            for(int i=0; i<lista.length; i++){
                    System.out.println(lista[i]);
            }
    }
}
```

Exercício 1

- Criar uma classe Arrays que contém um array de inteiros
- Criar um método “soma” que percorra e retorne a soma de todos os elementos do vetor
- Criar um método main que:
  - Crie uma instância da classe Arrays
  - Leia inteiros do console e armazene os valores no vetor da classe Arrays.
  - Imprima a soma dos números através do método soma

Exercício 2

- Criar uma classe chamada Matrizes
  - Criar um método para imprimir uma matriz recebida por parâmetro
  - Criar um método que tenha como parâmetros duas matrizes e retorne a matriz correspondente a multiplicação delas
  - Declare, construa e inicialize 2 matrizes com números fixos de tamanhos diferentes
- Imprimir a matriz de resultado na tela

Exercício 3

- Criar uma classe Funcionario com as variáveis de instância: nome, cargo e superior (chefe)
- Criar uma classe Empresa que contém um array de funcionários
- Criar um método “listarFuncionarios” que imprima todos os funcionários de uma empresa
- Criar um método main que leia os nomes e cargos dos funcionários do console.
  - Crie objetos Funcionario com cada um deles e armazene no vetor da classe empresa