

---

# Programação Orientada a Objetos

## Arrays em Java

Prof. Leandro Rodrigues Pinto  
<leandrorodp@gmail.com>

---

# O que é um Array

Quando desejamos armazenar uma grande quantidade de valores de um determinado tipo, podemos utilizar arrays.

Um array é um objeto que pode armazenar muitos valores de um determinado tipo.

Um **array** é um **grupo de variáveis** (chamados elementos ou componentes) que contém valores, todos do mesmo tipo. Os elementos de um array podem ser tipos **primitivos** ou tipos por **referências** (DEITEL; DEITEL, 2010, p.190).

# O que é um Array

Um array é uma estrutura de dados capaz de armazenar uma **coleção de variáveis**.

Todo array possui uma capacidade. Essa **capacidade** é a **quantidade** de variáveis que o array armazena.

**As variáveis contidas em um array não possuem nome.**

# O que é um Array

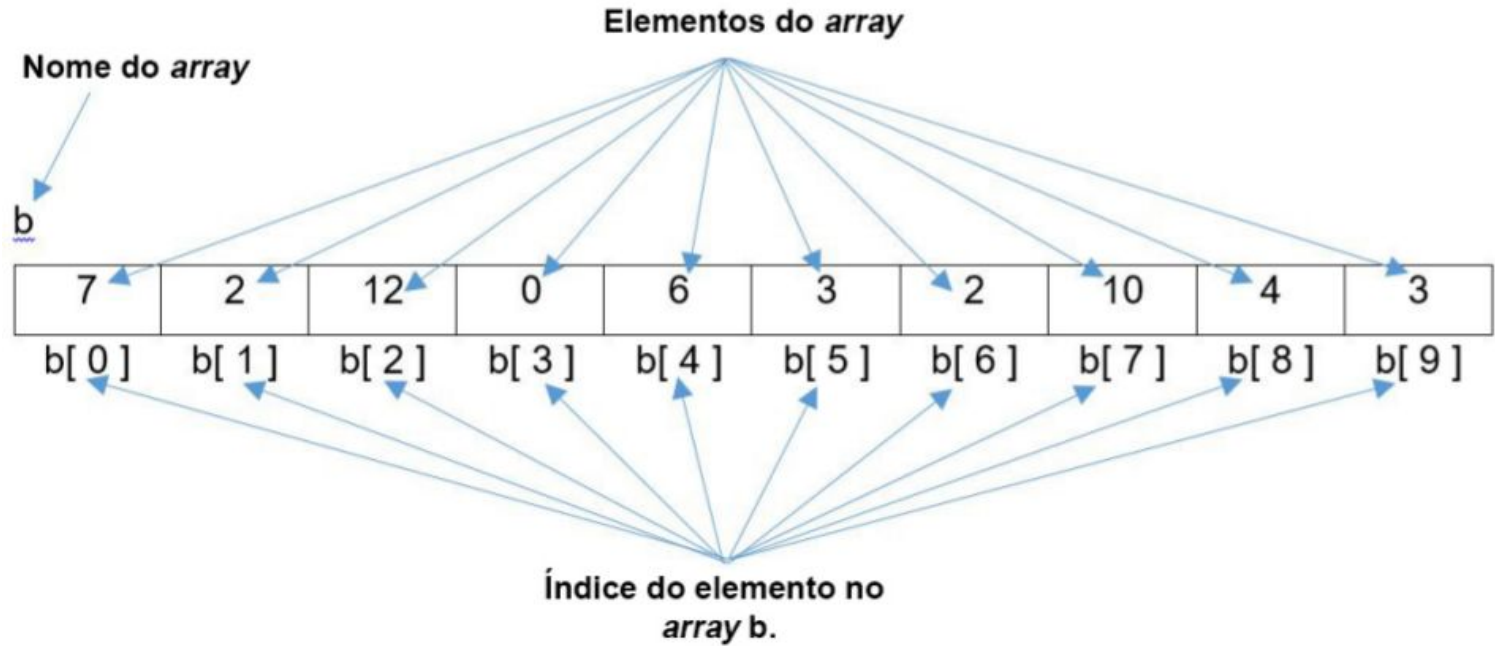
Para identificá-las, elas são numeradas de **0** até a **capacidade - 1**.

Dessa forma, o índice da primeira variável é 0, o índice da segunda variável é 1, o índice da terceira variável é 2 e assim sucessivamente.

Como as variáveis dentro de um array são organizadas de forma sequencial, é comum utilizar o termo “posição” para se referir a essas variáveis.

Por exemplo, utilizaremos “posição 10” ou invés de “variável 10”.

# O que é um Array



# Tipos de Array

**Array Unidimensional : Vetores**

**Array Multidimensional : Matrizes**

# Criando um array

Em Java, os arrays são criados através do comando new

```
int[] numeros = new int[100];
```

TIPO DE DADO QUE SERÁ  
ARMAZENADO PELO ARRAY



int[]



INFORMA QUE A VARIÁVEL  
SERÁ UM ARRAY

IDENTIFICADOR DA VARIÁVEL



nomeDoArray

=

INFORMA O TIPO DO  
NOVO OBJETO



new int[10];



INSTANCIA UM  
NOVO OBJETO



INFORMA A QUANTIDADE  
DE POSIÇÕES DO ARRAY

Declaração e inicialização de um array

# Criando um array

Em Java, os arrays são criados através do comando new .

```
Int [ ] numeros; // declaração  
numeros = new int [10]; // inicialização
```

```
Int [ ] numeros = new int [10]; // declaração e inicialização
```



# Criando um array

```
Int [ ] numeros = new int [10]; // as posições são inicializadas com 0
```

```
Boolean [ ] aprovados = new boolean [10]; // as posições são inicializadas com false
```

```
String [ ] nomes = new String [10]; // as posições são inicializadas com null
```

# Array Multidimensional

Até agora, utilizamos apenas arrays unidimensionais. Contudo, os arrays podem ser multidimensionais, ou seja, podemos criar arrays com duas ou mais dimensões.

Por exemplo, um array bidimensional pode ser utilizado para representar uma tabela, uma matriz ou até um tabuleiro de batalha naval.

# Array Multidimensional

Declaração:

```
int [ ][ ] numeros; // declaração
```

```
numeros = new int [10][10]; // inicialização
```

```
int [ ][ ] num = new int[5][5]; // declaração e inicialização
```

```
int [ ][ ] x ; // bidimensional
```

```
int [ ][ ] y [ ]; // tridimensional
```

```
int [ ] z [ ][ ][ ]; // quadrimensional
```

```
int[ ][ ][ ] numeros = new int[10][20][30];
```

# Atividade prática 1

Criar uma classe que realiza a soma dos valores armazenados em dois vetores e exibir o resultado, que também deverá ser armazenado em outro vetor.

Podem utilizar a classe do próximo slide para geração randômica.

# Atividade prática - Gerador de Números Aleatórios

```
1+ import java.util.Random;
5
6 public class GeradorNrAleatorios {
7-     public static void main(String[] args){
8
9         Set<Integer> numeros = new TreeSet<Integer>();
10        Random rand = new Random();
11        while(numeros.size() < 10) {
12            numeros.add(rand.nextInt(1000));
13        }
14        System.out.println("Números: " + numeros);
15
16    }
17 }
```

## Atividade prática 2

Criar uma classe que realiza a ordenação de um vetor de **n** elementos, utilizando o método *quick sort*.

Podem utilizar a classe do slide anterior para geração randômica dos elementos do vetor.

# Bibliografia

Cadenhead, Rogers; LEMAY, Laura, Aprenda em 21 dias Java 2. 4. ed. São Paulo: Campus, 2005.

DEITEL, Harvey H .; DEITEL, Paul J. Java: como Programar. 8. Ed. São Paulo: Pearson / Prentice-Hall, 2010.

# Perguntas?





**Obrigado!**