



Prof. Dr. Fernando Almeida  
[proffernando.almeida@fiap.com.br](mailto:proffernando.almeida@fiap.com.br)





# DDD (Domain Driven Design)

## Arrays unidimensionais - vetores



# O QUE VAMOS APRENDER HOJE?

1

Vetores - Definição

2

Declaração de vetores

3

Inicialização de vetores

4

Propriedade Length

5

Vetores com métodos



# O que são vetores?

## Definição



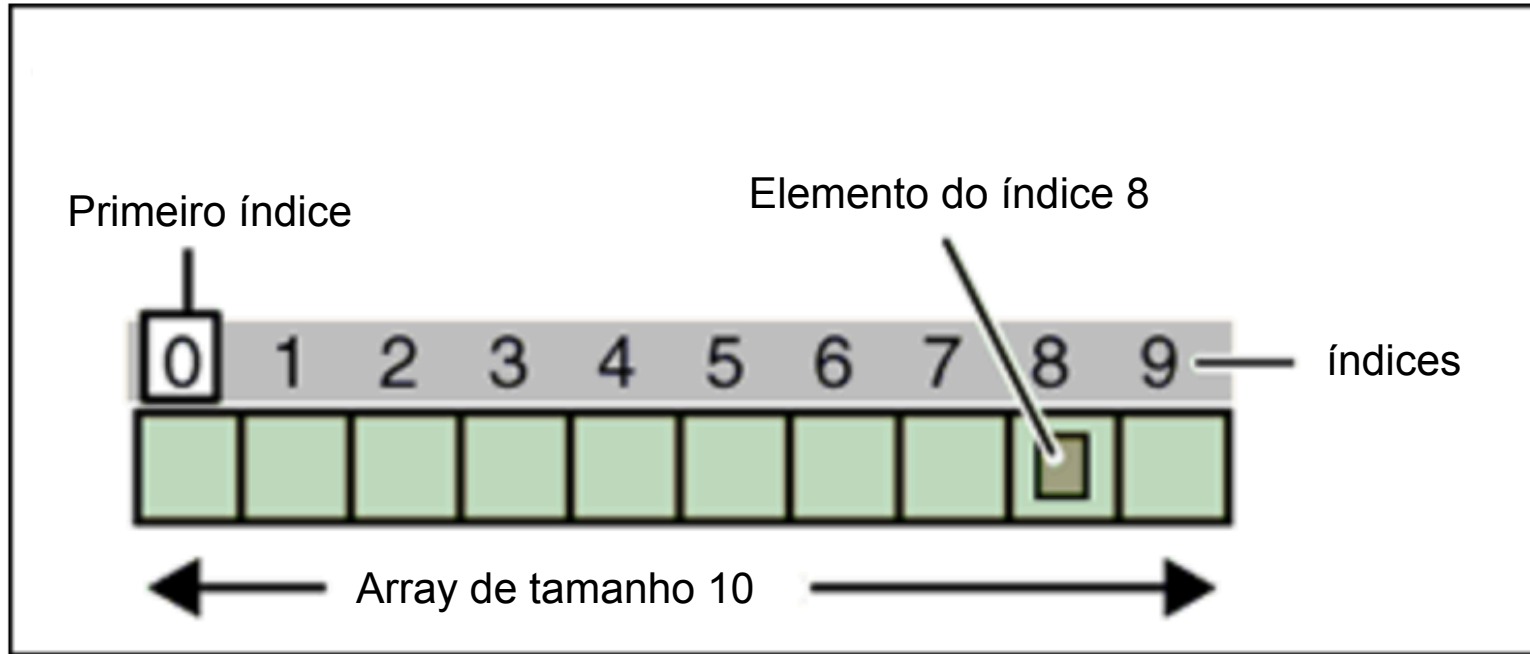
“Vetores são estruturas homogêneas (mesmo tipo) de dados que armazenam uma **quantidade fixa** de elementos de um certo tipo



# Vetores

- Internamente, um **vetor** armazena diversos valores, cada um associado a um número que se refere à **posição** de armazenamento
- Os **vetores** são **estruturas indexadas**, em que **cada valor** que pode ser armazenado em uma **posição** (índice) é chamado de **elemento** do **vetor**
- Cada elemento do **vetor** pode ser utilizado **individualmente** de forma **direta**, tornando-o uma estrutura de dados de **acesso aleatório**
- O número de **posições** de um vetor corresponde ao **tamanho** que ele tem
- Cada posição de um vetor é unicamente identificada por um valor inteiro positivo, linear e sequencialmente numerado, iniciado em ZERO

# Representação de um vetor



- Vetores permanecem do mesmo tamanho após sua criação


# Declaração de um vetor

- Semelhante a declaração de variáveis
- Sintaticamente diferenciados pelo uso de **colchetes**

```
1 | int vetor[]; // declaração do vetor
```

Embora declarados, o vetor não está pronto para o uso, sendo necessário reservar espaço para seus elementos (**alocação de memória**)

```
1 | vetor = new int[10]; // alocação de espaço para vetor
```



operador **new** (palavra reservada da linguagem) para alocar memória, neste exemplo, aloca memória para 10 elementos do tipo **int**



# Exemplos de declarações de variáveis do tipo Vetor

Na **Linguagem Java** um **vetor** é uma **classe** e deve-se utilizar o operador **new**, para instanciar o objeto, acionando o método **construtor** correspondente (**int**, **double**, **float**, **char**, **String**, entre outros)

- Exemplos: `int nro[] = new int[10];`

```
1 | int nro[] = new int[10];
```

 declarando um vetor para armazenar 10 números inteiros

```
double salario[] = new double[10];
```

```
1 | double salario[] = new double[10];
```

 declarando um vetor para armazenar 10 números do tipo real

```
1 | String mes[] = new String[12];
```

 declarando um vetor para armazenar o nome dos 12 meses do ano

# Representação de um vetor em memória

```
int c[ ] = new int[12];
```

declaração

array com 12 elementos

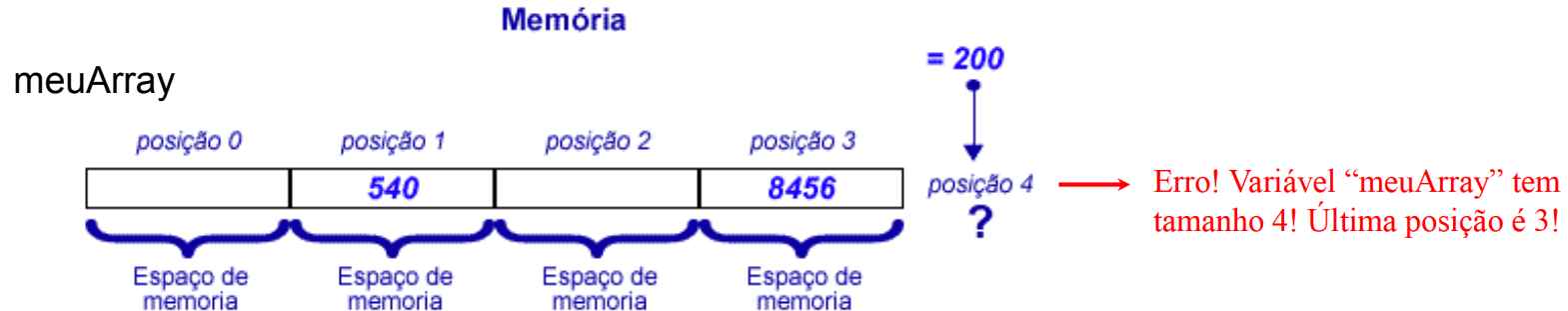
Nome do array (c)

Índice (ou subscripto)  
do elemento no array c

c[ 0 ]	-45	c[0] = -45;
c[ 1 ]	6	
c[ 2 ]	0	
c[ 3 ]	72	
c[ 4 ]	1543	
c[ 5 ]	-89	
c[ 6 ]	0	
c[ 7 ]	62	
c[ 8 ]	-3	
c[ 9 ]	1	
c[ 10 ]	6453	
c[ 11 ]	78	System.out.println(c[11]);

# Vetores

- O tamanho (limite) do vetor é sempre o seu tamanho, menos 1
- Em um vetor de tamanho 4, o último elemento estará na posição 3 (tamanho-1)
- Atribuir um valor na posição 4 ocasionará um **erro**
- **Exemplo: meuArray[4] = 200;**





# Inicialização de vetores

## Direta e indireta



# Inicialização indireta

## vetores e estruturas de repetição

```
1 int n = 10; // tamanho do vetor
2 int v[] = new int[n]; // declaração e alocação de espaço para o vetor "v"
3 int i; // índice ou posição
4
5 // processando os "n" elementos do vetor "v"
6 for (i=0; i<n; i++) {
7     v[i] = i; // na i-ésima posição do vetor "v" armazena o valor da variável "i"
8 }
```

Representação interna:

v[0]	v[1]	v[2]	v[3]	v[4]	v[5]	v[6]	v[7]	v[8]	v[9]
0	1	2	3	4	5	6	7	8	9

## Inicialização direta

```
1 | String nome[] = {"Juca Bala", "Maria da Silva", "Marcos Paqueta"};
```

nome[0]

nome[1]

nome[2]

Nota-se que **não** é necessário indicar o **tamanho** do vetor e fazer a alocação com o operador **new**

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

# Propriedade **Length**

- O tamanho do vetor será conhecido através da propriedade **length**

```
public class ExemploLength {  
    public static void main(String[] args) {  
        String nome[] = {"Juca Bala", "Maria da Silva", "Marcos Paqueta"};  
  
        int n = nome.length;  
  
        for(int i=0; i<n; i++){  
            System.out.printf("%do. nome = %s \n", (i+1), nome[i]);  
        }  
    }  
}
```

- • Propriedade **Length**

- • Outro exemplo

```
public class ExemploLength2 {  
    public static void main(String[] args) {  
  
        String meses[] = {"Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho",  
                           "Julho", "Agosto", "Setembro", "Outubro", "Novembro", "Dezembro"};  
  
        int diaMes[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};  
  
        for(int i=0; i<meses.length; i++){  
            System.out.printf("%s, tem %d dias. \n", meses[i], diaMes[i]);  
        }  
    }  
}
```



# Exercícios

1. Escreva um programa que crie um vetor de tamanho n e imprima apenas os elementos positivos. Adicione um contador para imprimir a quantidade de elementos positivos.
2. Escreva um programa que some os elementos de dois vetores (de mesmo tamanho) e armazene o resultado em um terceiro vetor. Imprima os elementos do terceiro vetor.
3. Escreva um programa para unir dois vetores (de mesmo tamanho) em um terceiro vetor. O programa deve imprimir os elementos do terceiro vetor.



# Aplicação Java com Vetores (sem métodos)

```
1 import java.util.Scanner;
2
3 public class Exemplo1 {
4
5     public static void main(String[] args) {
6         Scanner ler = new Scanner(System.in);
7
8         int n = 10; // tamanho do vetor
9         int v[] = new int[n]; // declaração do vetor "v"
10        int i; // índice ou posição
11
12        // Entrada de Dados
13        for (i=0; i<n; i++) {
14            System.out.printf("Informe %2do. valor de %d: ", (i+1), n);
15            v[i] = ler.nextInt();
16        }
17
```

```
18 // Processamento: somar todos os valores, definir o maior e o menor valor
19     int soma = 0;
20     int menor = v[0]; // v[0] = 1o. valor armazenador no vetor "v"
21     int maior = v[0];
22     for (i=0; i<n; i++) {
23         soma = soma + v[i];
24
25         if (v[i] < menor)
26             menor = v[i];
27
28         if (v[i] > maior)
29             maior = v[i];
30     }
31
```

```
32 // Saída (resultados)
33 System.out.printf("\n");
34 for (i=0; i<n; i++) {
35     if (v[i] == menor)
36         System.out.printf("v[%d] = %2d <--- menor valor\n", i, v[i]);
37     else if (v[i] == maior)
38         System.out.printf("v[%d] = %2d <--- maior valor\n", i, v[i]);
39     else System.out.printf("v[%d] = %2d\n", i, v[i]);
40 }
41
42 System.out.printf("\nSoma = %d\n", soma);
43 }
44
45 }
```



# Vetores com Métodos

## Exemplo



# Exemplos

```
public int tamanhoVetor(){
    System.out.println("*-- Tamanho do Vetor --*");
    int tamanho;
    System.out.println("Tamanho do Vetor: ");
    tamanho = input.nextInt();
    return tamanho;
}
```

Exemplo 1: Métodos (review)

```
public int[] criaVetor(int tamanho){
    System.out.println("*-- Criação do Vetor --*");
    int[] vetor = new int[tamanho]; //cria um vetor com elementos ZEROS
    return vetor;
}
```

Exemplo 2: Método que retorna um vetor

```
public int[] populaVetor(int[] vetor){
    System.out.println("*-- População do Vetor --*");
    for(int i = 0; i < vetor.length; i++){
        System.out.printf("vetor[%d] = ", i);
        System.out.println("");
        vetor[i] = input.nextInt();
    }
    return vetor;
}
```

Exemplo 3: Método que recebe um vetor por parâmetro



# Vamos testar no Eclipse?

## Exemplo Prático





# Exercício

Modifique o código do Exemplo1 (slides 19, 20 e 21) separando-o em **Métodos**.



# OBRIGADO

## FIAP

Copyright © 2021 | Prof. Dr. Fernando Luiz de Almeida

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.



