

Linguagem de Programação

VETORES

Prof. Me. Humberto Zanetti



Vetores

- Operações com vetores
- Imprimindo os valores
- Copiando vetor
- Busca em vetor

Tamanho de vetor

- Atributo **length**
 - Todo vetor possui o atributo **length**, que pode ser acessado como:
`nome_vetor.length`
- Esse atributo retorna um valor inteiro, referente à quantidade de elementos de um vetor, ou como chamamos, o **tamanho do vetor**.
- Geralmente atribuímos esse valor a uma variável, ou usamos em um laço de repetição para percorrer o vetor.

Tamanho do vetor

```
public static void main(String[] args) {  
    int[] numeros = {1,2,3,4,5,6,7,8,9,10};  
    int tam;  
    for(int i = 0; i < numeros.length; i++ ){  
        System.out.println(numeros[i]);  
    }  
    tam = numeros.length;  
    System.out.println("Tamanho do vetor: "+ tam);  
}
```

Imprimindo um vetor

- A estratégia mais comum é imprimir o vetor com cada elemento por vez
 - Utilizado para diferenciar cada elemento e sua posição

```
public static void main(String[] args) {  
    int[] num = {1,2,3};  
    for(int i = 0; i < num.length; i++) {  
        System.out.println("num["+i+"]:"+num[i]);  
    }  
}
```

Saída:
num[0]:1
num[1]:2
num[2]:3

Imprimindo um vetor

- Em Java, podemos imprimir **TODOS** os elementos de um vetor de uma só vez!
 - Usado muito para visualizar rapidamente os valores.
- Usamos a função `Arrays.toString(vetor)` ;
 - Devemos importar a biblioteca `java.util.Arrays` ;

```
public static void main(String[] args) {  
    int[] numeros = {1,2,3,4,5,6,7,8,9,10};  
    System.out.println(Arrays.toString(numeros)) ;  
}
```

Saída: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Copiando um vetor para outro

Estratégia 1: uso de laço de repetição (posição por posição)

```
public static void main(String[] args) {  
    int[] num1 = {1,2,3,4,5,6,7,8,9,10};  
    int[] num2 = new int[num1.length];  
    for(int i = 0; i < num1.length; i++) {  
        num2[i] = num1[i];  
    }  
    System.out.println(Arrays.toString(num2));  
}
```

Saída: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Copiando um vetor para outro

Estratégia 2: usando a função `System.arraycopy()`

`arraycopy(vetor_original, posicao, vetor_destino, posicao, tamanho)`

```
public static void main(String[] args) {  
    int[] num1 = {1,2,3,4,5,6,7,8,9,10};  
    int[] num2 = new int[num1.length];  
    System.arraycopy(num1, 0, num2, 0, num1.length);  
    System.out.println(Arrays.toString(num2));  
}
```

Saída: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Copiando um vetor para outro

- É possível copiar conteúdos de uma vetor para o outro, mesmo sendo de tamanhos diferentes:
 - Um vetor de tamanho menor, para um de tamanho maior;
 - Copiar apenas uma porção de um vetor para outro vetor;
 - Copiar a partir de um índice específico

Copiando parcialmente

Estratégia 1: usando o laço for()

```
public static void main(String[] args) {  
    int[] num1 = {1,2,3,4,5};  
    int[] num2 = new int[10];  
    for(int i=0; i<num1.length; i++){  
        num2[i] = num1[i];  
    }  
    System.out.println(Arrays.toString(num2));  
}
```

Saída: [1, 2, 3, 4, 5, 0, 0, 0, 0, 0]

Copiando parcialmente

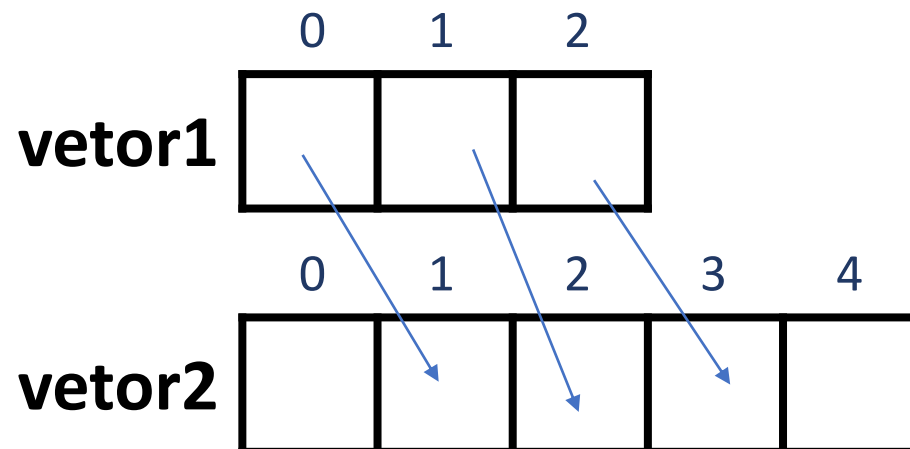
Estratégia 2: usando a função `System.arraycopy()`

```
public static void main(String[] args) {  
    int[] num1 = {1,2,3,4,5};  
    int[] num2 = new int[10];  
    System.arraycopy(num1, 0, num2, 0, num1.length);  
    System.out.println(Arrays.toString(num2));  
}
```

Saída: [1, 2, 3, 4, 5, 0, 0, 0, 0, 0]

Copiando com índices específicos

- **Exercício prático:** como podemos copiar parcialmente um vetor, sem partir do primeiro índice (**índice 0**)?
- Suponha que o **vetor1** tenha 3 valores inteiros e o **vetor2** seja tenha 5 posições livres. Como podemos gravar os valores do vetor1, a partir da segunda posição (índice 1) do vetor2? Faça um programa usando o laço `for()` e depois outro com `arraycopy()`.



Buscando valor no vetor

- Busca de dados em um vetor é umas das operações mais comuns que temos.
 - Vetores armazenam dados que podem ser operados ou comparados.
- Dicas:
 - Não confundir **índice** com **elemento**
 - Não acessar ou gravar algum dado em uma posição inexistente, como exceder o limite do vetor.
 - Erro **java.lang.IndexOutOfBoundsException**

Buscando valor no vetor

```
public static void main(String[] args) {  
    int[] numeros = {5,7,6,4,8,3,6,7,9,2};  
    int x =7; //valor a ser buscado  
    for(int i=0; i<numeros.length; i++){  
        if(numeros[i] == x){  
            System.out.println("Valor x na posição:"+i);  
        }  
    }  
}
```

Saída:

Valor x na posição:1

Valor x na posição:7

Buscando valor no vetor

Mostrar de o valor se encontra na primeira metade ou na segunda metade do vetor, supondo um vetor de 10 posições.

```
public static void main(String[] args) {  
    int[] numeros = {5,7,6,4,8,3,6,7,9,2};  
    int x =7; //valor a ser buscado  
    for(int i=0; i < 5; i++){  
        if(numeros[i] == x){  
            System.out.println("Na 1.a metade, posição "+i);  
        }  
    }  
    for(int i=5; i <= 9; i++){  
        if(numeros[i] == x){  
            System.out.println("Na 2.a metade, posição "+i);  
        }  
    }  
}
```

Saída:

Na 1.a metade, posição 1

Na 2.a metade, posição 7

Exercícios de fixação

1. Faça um programa para que o usuário preencha o vetor de 10 valores inteiros e escolha um número, o qual deverá ser buscado dentro desse vetor e mostre em qual(is) posição(ões) em que ele se encontra. Deverá ser informado ao usuário o caso do número não existir no vetor.
2. Faça um programa em que o usuário preenche um vetor de 20 posições e informe se deseja imprimir a primeira ou a segunda metade do vetor.

Dúvidas!?

humberto.zanetti@fatec.sp.gov.br