

Assunto: **Vetores**

Aula Prática: 06

## Introdução – Vetores

Um vetor também é conhecido como variável composta homogênea unidimensional, ou seja, corresponde a um conjunto de variáveis do mesmo tipo, que possuem o mesmo identificador e são alocadas seqüencialmente na memória. Como as variáveis têm o mesmo nome, o que as distingue é um índice que referencia a sua localização dentro da estrutura.

Em Java são definidos pela existência de colchetes vazios antes ou depois do nome da variável no momento da declaração. Veja o exemplo:

```
int x[] = new int[10];
```

A primeira parte do comando, `int x[]` corresponde à declaração do vetor, incluindo o tipo `int` e o nome do vetor `x[]`. O operador `new` é necessário para criar a instância do vetor, indicando o tipo de dados que está sendo instanciado `int` e o número de posições `[10]`.

O código acima declara um vetor de inteiros com 10 posições, numeradas de 0 a 9, na qual cada posição é uma variável do tipo inteiro. Veja um exemplo de valores atribuídos às posições 0 a 9 do vetor `x[]`:

<code>x[]</code>	10	5	4	8	3	5	2	1	9	-1000
	0	1	2	3	4	5	6	7	8	9

A atribuição de um valor a uma posição da variável `x[]` é realizada conforme segue:

<code>x[0] = 10;</code>	<code>x[5] = 5;</code>
<code>x[1] = 5;</code>	<code>x[6] = 2;</code>
<code>x[2] = 4;</code>	<code>x[7] = 1;</code>
<code>x[3] = 8;</code>	<code>x[8] = 9;</code>
<code>x[4] = 3;</code>	<code>x[9] = -1000;</code>

Uma prática interessante é utilizar uma variável dentro dos colchetes `[]` para percorrer o vetor. Por exemplo, pode-se imprimir as 10 primeiras posições do vetor com:

```
for (int posição = 0; posição < 10; posição++)
    System.out.println( x[posição] );
```

A propriedade `length` retorna o número de posições alocadas:

```
System.out.printf("O tamanho do vetor x é %d", x.length );
```

Uma prática interessante é utilizar uma variável dentro dos colchetes `[]` para percorrer o vetor. Por exemplo, pode-se imprimir as 10 primeiras posições do vetor com:

```
for (int posição = 0; posição < 10; posição++)
```

### Atividade 1

Faça um programa que leia valores para preencher um vetor de `float` com 10 posições. Em seguida, o programa deve imprimir a soma dos valores.

Utilize o `JOptionPane` para leitura dos valores.

### Atividade 2

Faça um programa que leia valores inteiros e preencha um vetor com oito posições denominado `valoresEntrada`. Em seguida, crie dois vetores de mesmo tipo e tamanho, denominados `valoresPositivos` e `valoresNevativos`. O vetor `valoresPositivos` deve conter os números maiores que zero. O vetor `valoresNevativos` deve conter os números menores que zero. Observe que cada vetor resultante deve ter o mesmo tamanho do vetor de leitura, entretanto, nem todas as posições terão valores. Utilize uma variável para manter a contagem de valores inseridos em `valoresPositivos` e outra variável para `valoresNevativos`. Essas variáveis devem ser iniciadas com valor zero e a cada valor inserido nos vetores ela deve ser incrementada. Ao final, imprima os vetores `valoresPositivos` e `valoresNevativos`.

### Atividade 3

Faça um programa que preencha um vetor com dez números inteiros, calcule e mostre um o vetor resultante em ordem crescente de valores.

Exemplo de entrada:

x	10	5	4	8	3	5	2	1	9	-1000
	0	1	2	3	4	5	6	7	8	9

Saída:

x	-1000	1	2	3	4	5	5	6	7	8	9	10
	0	1	2	3	4	5	5	6	7	8	9	10

Para a ordenação, pode ser empregado o seguinte pseudo-código:

```
para ( i = 0 até tamanho do vetor - 1) faça
    para ( j = i até tamanho do vetor) faça
        se ( vetor[i] > vetor[j] ) então
            troque vetor[i] com vetor[j]
        fim-se
    fim-para
fim-para
```