



Algoritmos e Programação II

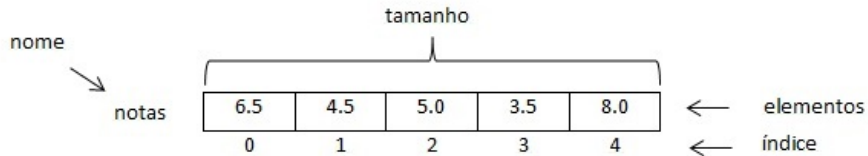
Vetor

Prof. Alexandre Mignon

- Uma tarefa comum em programação é a manutenção de uma coleção numerada de valores relacionados
- Por exemplo, deseja-se manter a relação das notas de cinco alunos
- Em vez de usar cinco variáveis diferentes para esta tarefa, prefere-se usar um único nome para a coleção e usar índices numéricos para referenciar as notas dentro da coleção

- Vetor é uma coleção numerada de variáveis do mesmo tipo que são referenciadas por um único nome
- Um vetor possui um tamanho, definido na sua inicialização, e que nunca se altera. Esse tamanho indica quantos elementos o vetor pode armazenar
- Cada variável ou **elemento** em um vetor tem um **índice** que referencia o valor armazenado no elemento de forma única
- Os elementos de um vetor são numerados por um número natural 0, 1, 2 e assim por diante
- O primeiro elemento do vetor tem índice zero e o último elemento tem índice *tamanho* - 1

Vetor



Declaração de Vetor

- Como uma variável simples, um vetor deve ser declarado antes de sua utilização
- A declaração de um vetor é feita indicando o tipo de dado que o vetor irá armazenar e o seu nome

Declaração de Vetor

```
tipo [] nome;
```

- *tipo* é um tipo qualquer de dados;
- *nome* é o nome pelo qual o vetor será referenciado;

Exemplo

```
float [] notas;
```

Criação de Vetor

- Após a declaração de um vetor é necessário criá-lo, isto é, criar o conjunto de elementos na memória, determinando seu tamanho
- A criação é feita utilizando-se o operador **new**

Criação de Vetor

```
nome = new tipo[tamanho];
```

- *nome* é o nome pelo qual o vetor foi declarado;
- *tipo* é o tipo pelo qual o vetor foi declarado;
- *tamanho* é a quantidade de elementos que o vetor pode conter.

Exemplo

```
notas = new float [5];
```

Declaração e Criação de Vetor

- Em Java, é possível declarar e criar um vetor em uma única linha

Declaração e Criação de Vetor

```
tipo [] nome = new tipo[tamanho];
```

Exemplo

```
float [] notas = new float [5];
```

Inicialização de Vetor

- Um vetor criado da forma apresentada inicializa seus elementos com o valor zero se o tipo do vetor for um tipo numérico
- É possível inicializar um vetor com diferentes valores

Inicialização de Vetor

tipo [] nome = {lista de valores separados por vírgula};

- A quantidade de elementos na lista determina o tamanho do vetor

Exemplo

`float [] notas = {6.5, 7.0, 5.5, 9.0, 2.5};`

Referenciando um Elemento do Vetor

- Os elementos de um vetor devem ser referenciados de forma individualizada
- Não é possível referenciar todos os elementos ao mesmo tempo
- Um elemento é referenciado pelo nome do vetor seguido do índice entre colchetes onde o elemento está armazenado

Referencia o terceiro elemento do vetor notas

```
notas[2];
```

- Podemos atribuir ou acessar um elemento do vetor

Exemplos

```
notas[2] = 7.5;  
float nota = notas[0];
```

- O primeiro exemplo atribui 7.5 ao terceiro elemento do vetor
- O segundo exemplo acessa o valor do primeiro elemento do vetor e atribui a variável nota

Lendo Dados para um Vetor

- Imagine que temos um vetor denominado *notas* que pode armazenar 5 elementos do tipo *float*. Queremos solicitar ao usuário que informe os dados para preencher esse vetor

```
Scanner entrada = new Scanner (System.in);  
float [] notas = new float[5];  
for (int i = 0; i < notas.length; i++) {  
    System.out.print ("Nota: ");  
    notas[i] = entrada.nextFloat();  
}
```

- Em Java é possível acessar o tamanho de um vetor através do campo *length* associado ao nome do vetor.
- Com isso, não é necessário ter que saber o tamanho de um vetor

Acessando o Tamanho de um Vetor

nome_do_vetor.length;

Acessando Dados de um Vetor

- Após o preenchimento do vetor queremos manipular os seus elementos
- O trecho de código abaixo calcula a média das notas do vetor preenchido anteriormente e apresenta os elementos do vetor na tela

```
float soma = 0.0;
for (int i = 0; i < notas.length; i++) {
    soma += notas[i];
    System.out.println ("Nota " + i + ": " + notas[i]);
}
float media = soma / notas.length;
System.out.println ("Media = " + media);
```

Exemplo

```
import java.util.Scanner;

public class ExemploVetor {
    public static void main (String [] args) {
        Scanner entrada = new Scanner(System.in);
        // declara e inicializa o vetor
        float [] notas = new float[5];
        // le os dados para o vetor
        for (int i = 0; i < notas.length; i++) {
            System.out.print ("Nota: ");
            notas[i] = entrada.nextFloat();
        }
        // calcula a media das notas e as apresenta na tela
        float soma = 0.0f;
        for (int i = 0; i < notas.length; i++) {
            soma += notas[i];
            System.out.println ("Nota " + i + ": " + notas[i]);
        }
        float media = soma / notas.length;
        System.out.println ("Media = " + media);
    }
}
```

Vetor e Função

- Um vetor pode ser passado como parâmetro para uma função
- Para indicarmos que um parâmetro de uma função é um vetor, basta colocarmos colchetes antes do nome do parâmetro
- Na especificação do parâmetro não é necessário colocar o tamanho do vetor
- O tipo de passagem de parâmetro de um vetor para uma função é sempre por *referência*

Declaração de um Vetor como Parâmetro

```
tipoFuncao nomeFuncao (tipoVetor [] nomeVetor) {  
    instruções;  
}
```

- Quando chamamos uma função que têm como parâmetro um vetor, basta passar apenas o nome do vetor para a função

```
public static void main (String [] args) {  
    // declara e inicializa um vetor  
    int [] vet = {10, 20, 30, 40, 50, 60};  
    // chama a funcao apresentarVetor passando vet como parametro  
    apresentarVetor (vet);  
}  
  
static void apresentarVetor (int [] vetor) {  
    instruções;  
}
```

Exemplo

```
public class ExemploVetorFuncao {  
    public static void main(String [] args) {  
        // declaracao e inicializacao do vetor  
        int [] vet = {10, 20, 30, 40, 50, 60};  
        // chama a funcao apresentarVetor  
        apresentarVetor(vet);  
        // chama a funcao somarVetor  
        int soma = somarVetor(vet);  
        System.out.println ("\nSoma = " + soma);  
    }  
}
```

Exemplo (cont.)

```
static void apresentarVetor(int [] vetor) {  
    for (int i = 0; i < vetor.length; i++) {  
        System.out.print(vetor[i] + " ");  
    }  
}
```

```
static int somarVetor(int [] vetor) {  
    int soma = 0;  
    for (int i = 0; i < vetor.length; i++) {  
        soma += vetor[i];  
    }  
    return soma;  
}
```