



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE

AULA:

# Vetores e Matrizes em Java (Arrays)

Programação Orientada a Objetos

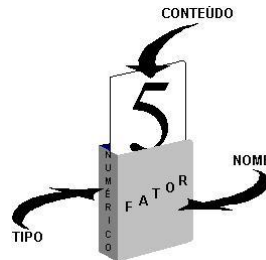
**Alba Lopes, Profa.**

<http://docentes.ifrn.edu.br/albalopes>  
[alba.lopes@ifrn.edu.br](mailto:alba.lopes@ifrn.edu.br)

## Introdução

### ► Variável

- Analogia: uma caixa, na qual você pode dar o nome que lhe achar conveniente, e guardar o conteúdo que desejar



- Possui um tipo (String, boolean, int, double, ou qualquer classe...)
- O valor dentro da “caixa” que pode ser alterado de acordo com a execução do programa



## Introdução

- ▶ Agora imagine como ficaria na declaração de variáveis, declarando uma a uma, as 50 variáveis para o nome, depois as variáveis para as médias de cada aluno...
- ▶ Em casos como esse que é útil a utilização da estrutura de dados conhecida como vetor
- ▶ Um vetor é uma espécie de caixa com várias divisórias para armazenar coisas (dados)
  - ▶ É uma variável que pode armazenar vários valores



# Introdução

meuVetor



medias



nomes



alunos



## Arrays

- ▶ A forma mais eficiente de trabalhar com coleções de elementos em Java é através da construção de vetores (**arrays**).
- ▶ Em Java, **arrays são objetos que armazenam múltiplas variáveis do mesmo tipo**
  - ▶ **Uma dimensão = vetor**
  - ▶ **Duas dimensões = matriz**
- ▶ Uma vez criado, um **array** não pode ter seu tamanho alterado.

## Arrays

- ▶ Os vetores são definidos pelo **tipo de dados** que eles devem armazenar e a **quantidade de posições**
- ▶ **Exemplo:**
  - ▶ Vetor de 8 posições para armazenar números reais
  - ▶ Vetor de 40 posições para armazenar objetos do tipo Aluno
- ▶ Os vetores são estruturas **homogêneas**.
  - ▶ Ex: um vetor de inteiros só armazena dados do tipo inteiro  
um vetor de alunos só armazena dados do tipo Aluno (*depois veremos o conceito de polimorfismo!*)

## Criação de Arrays

### ▶ Criando Arrays - Vetores

#### ▶ Sintaxe:

▶ <tipo> [] vetor = new <tipo>[tamanho];

#### ▶ Exemplos:

▶ `int [] meuVetor = new int[5];`

▶ `boolean [] resultados = new boolean[30];`

▶ `String [] nomes = new String[8];`

▶ `Automovel [] meusCarros = new Automovel[10];`

▶ Automaticamente, todos os elementos do vetor recebem valores default (0, false ou null).



## Criação de Arrays (unidimensionais)

### ► Declarando e informando quantidade de elementos

```
<tipo> vetor[] = new <tipo>[tamanho];
```



```
int vetor1[] = new int[3];
```

```
<tipo>[] vetor = new <tipo>[tamanho];
```



```
int[] vetor2 = new int[3];
```

### ► Declarando e inicializando elementos

```
<tipo>[] vetor = new <tipo>[ ] {val0, ..., valN-1};
```



```
int[] vetor3 = new int[] {1,2,3};
```

### ► Declarando e inicializando elementos (forma compacta)

```
<tipo> vetor[] = {val0, val1, ..., valN-1};
```



```
int vetor4[] = {1,2,3};
```

```
<tipo>[] vetor = {val0, val1, ..., valN-1};
```



```
int[] vetor5 = {1,2,3};
```



## Criação de Arrays (multidimensionais)

### ► Formas de criação de arrays (matrizes) em Java:

#### ► Sintaxe/Exemplo:

```
<tipo> matriz[][] = new <tipo>[linhas][colunas];
```



```
int m1[][] = new int[2][2];
```

```
<tipo>[][] matriz = new <tipo>[linhas][colunas];
```



```
int[][] m2 = new int[2][2];
```

```
<tipo>[][] matriz[] = {{val00, val01}, {val10, val11}};
```



```
int[][] m3 = {{1,2},{3,4}};
```

## Acessando os elementos do array

- ▶ Os elementos são acessados através do operador de indexação []
- ▶ O índice dos elementos inicia em zero
- ▶ A exceção `IndexOutOfRangeException` é levantada se um índice inválido é usado
  - ▶ Recuperando o valor de um elemento

```
int[] vetor = new int[4] { 1, 2, 3, 4 };  
int total = vetor[0] + vetor[1] + vetor[2] + vetor[3];
```

- ▶ Atribuindo valor a um elemento

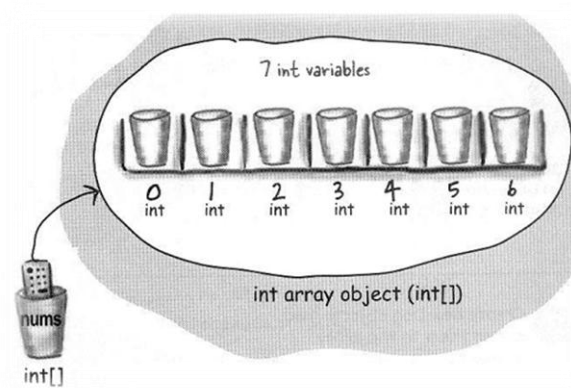
```
vetor[0] = vetor[1] = vetor[2] = vetor[3] = 0;
```

## Exemplo de Declaração, Inicialização e Uso

- ▶ Exemplo de declaração, inicialização e acesso de vetor

```
int [] nums;  
nums = new int [7];
```

```
nums [0] = 6;  
nums [1] = 19;  
nums [2] = 46;  
nums [3] = 35;  
nums [4] = 8;  
nums [5] = 34;  
nums [6] = 12;
```



## Array de referências

- ▶ Declarando variáveis de arrays

- ▶ É necessário definir o tipo do elemento, seguido de um par de chaves e de uma variável para referenciar o vetor

- ▶ Criando instâncias

- ▶ O operador new é usado para alocar o vetor na memória

```
int[] vetorVazio;
```

```
int[] vetor = new int[4];
```

## Exemplo de vetores de referências

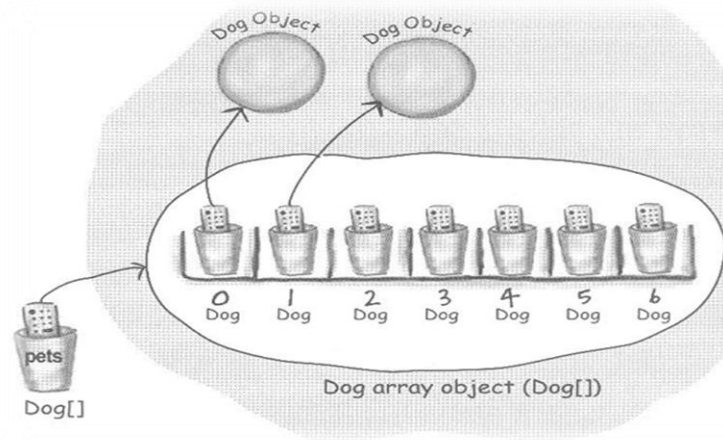
- ▶ Exemplo de atribuição de referências entre vetores

```
Dog[] d = new Dog[7];
```

```
d[0] = new Dog();  
d[1] = new Dog();
```

```
Dog fido = new Dog();  
fido.name = "Fido";  
fido.bark();  
fido.chaseCat();
```

```
d[0] = fido;  
d[1] = new Dog();  
d[1].name = "Rex";  
d[1].bark();
```



## Exemplo de vetores de referências

- ▶ Criando um array de objetos:

```
Automovel [] carros = new Automovel[5];
```

- ▶ Esse array tem capacidade para armazenar 5 objetos do tipo Automovel, entretanto, os objetos ainda não foram criados
  - ▶ Cada posição do array contem o valor NULL
- ▶ Inicializando posições do array manualmente:

```
carros[0] = new Automovel("Fiat", "Palio", "Branco");  
carros[1] = new Automovel("Fiat", "Uno", "Preto")
```

## Tamanho de arrays

- ▶ Para conhecer o tamanho total de um array basta você acessar o atributo `length`
- ▶ Este atributo retorna um valor inteiro (`int`) que indica qual a capacidade máxima de armazenamento deste array

### ▶ Lembretes:

- ▶ Primeira posição (índice): de qualquer array é sempre 0
- ▶ Última posição (índice): é sempre o seu tamanho - 1 (`length - 1`)

### ▶ Exemplos:

```
int[] vetor = new int[10];  
vetor.length;
```

```
int[][] matriz = new int[2][3];  
matriz.length;  
matriz[0].length;
```

## Iterando por um array

- ▶ Iterar um array é percorrer todos os seus elementos
- ▶ Utiliza-se alguma estrutura de repetição para realizar a iteração
- ▶ Utilizando for

```
int[] pins = { 9, 3, 7, 2 };  
for (int i = 0; i < pins.length; i++)  
{  
    int pin = pins[i];  
    System.out.println(pin);  
}
```



length é um atributo e não um método, razão pelo qual não é necessário usar chaves para chamá-lo





## Iterando por um array

- ▶ Iterar um array é percorrer todos os seus elementos
- ▶ Utiliza-se alguma estrutura de repetição para realizar a iteração
  - ▶ Utilizando foreach

```
int[] pins = { 9, 3, 7, 2 };  
for (int pin : pins)  
{  
    System.out.println(pin);  
}
```

## Iterando por um array de objetos

### ► Utilizando for

```
Automovel [] carros = new Automovel[3];
carros[0] = new Automovel("Palio", "Branco", 50); //construtor que inicializa atributos modelo, cor e vel
carros[1] = new Automovel("Uno", "Preto", 100);
carros[2] = new Automovel("Gol", "Vermelho", 20);

for (int i = 0; i < carros.length; i++)
{
    System.out.println(a.getModelo());
}
```

## Iterando por um array de objetos

### ► Utilizando foreach

```
Automovel [] carros = new Automovel[3];
carros[0] = new Automovel("Palio", "Branco", 50); //construtor que inicializa atributos modelo, cor e
vel
carros[1] = new Automovel("Uno", "Preto", 100);
carros[2] = new Automovel("Gol", "Vermelho", 20);

for (Automovel a : carros)
{
    Automovel a = carros[i];
    System.out.println(a.getModelo());
}
```

## Exemplos

- ▶ Exemplo com vetor
- ▶ Preenchendo um vetor de inteiros com dados do usuário

```
import java.util.Scanner;
public class ExemploVetor {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int [] vetor = new int[5];

        //Leitura de dados
        for (int i = 0; i < 5; i++) {
            System.out.println("Digite:");
            vetor[i] = sc.nextInt();
        }
        //Escrita de dados
        for (int i = 0; i < 5; i++) {
            System.out.println(vetor[i]);
        }
    }
}
```



## Exemplos

### ► Seja a classe Contato

Contato
- nome: String - telefone: String
+ Contato(String, String) + getNome(): String + setNome(String): void + getTelefone(): String + setTelefone(String): void + toString(): String

```
public class Contato {
    private String nome;
    private String telefone;

    public Contato(String nome, String telefone) {
        this.nome = nome;
        this.telefone = telefone;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getTelefone() {
        return telefone;
    }

    public void setTelefone(String telefone) {
        this.telefone = telefone;
    }
}
```

### ► Criar um array chamado agenda para armazenar os contatos telefônicos. Preencha o array com dados e, em seguida, escreva os dados na tela:

```
public class TesteArrayContatos {
    public static void main(String[] args) {
        Contato [] agenda = new Contato[5];
        agenda[0] = new Contato("Maria", "123456");
        agenda[1] = new Contato("Pedro", "654321");
        agenda[2] = new Contato("Marta", "951753");
        agenda[3] = new Contato("João", "357159");
        agenda[4] = new Contato("Mirim", "456258");
        System.out.println(" *** Minha agenda de contatos ***");
        for (int i=0; i<agenda.length; i++){
            System.out.print("Nome: "+agenda[i].getNome() + " / ");
            System.out.println("Telefone: "+agenda[i].getTelefone());
        }
    }
}
```



## Exemplos

- ▶ Alterando o exemplo anterior para receber os dados do usuário através do console:

```
import java.util.Scanner;
```

```
public class TesteArrayContatos {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        Contato [] agenda = new Contato[5];  
  
        for (int i=0; i<agenda.length; i++){  
            System.out.println("Dados do contato #" + (i+1) + " :");  
            System.out.print("Nome:");  
            String nome = sc.nextLine();  
            System.out.print("Telefone:");  
            String telefone = sc.nextLine();  
            agenda[i] = new Contato(nome, telefone);  
        }  
  
        System.out.println(" *** Minha agenda de contatos ***");  
        for (int i=0; i<agenda.length; i++){  
            System.out.print("Nome: " + agenda[i].getNome() + " / ");  
            System.out.println("Telefone: " + agenda[i].getTelefone());  
        }  
    }  
}
```

### Contato

- nome: String
- telefone: String

- + Contato(String, String)
- + getNome(): String
- + setNome(String): void
- + getTelefone(): String
- + setTelefone(String): void
- + toString(): String



## Exercício #1

- ▶ Crie a classe aluno tal como apresentada no diagrama ao lado.
  - ▶ Crie o construtor, que recebe por parâmetro o nome e as duas notas do aluno
  - ▶ Crie os métodos get e set para cada um dos atributos
  - ▶ O método calcularMedia retorna um valor do tipo double representando a média aritmética das duas notas do aluno
- ▶ Crie uma classe TestarArrayAluno que possua um array de objetos, chamado turma, do tipo Aluno, com capacidade para 5 objetos.
  - ▶ Crie os 5 objetos do tipo aluno. Pelo construtor, atribua o nome e as notas de cada aluno.
  - ▶ Itere sobre o array e apresente uma lista contendo o nome e a média de cada um dos alunos

Aluno
- nome: String - nota1: double - nota2: double
+ Aluno(String, int, int) + getNome(): String + setNome(String): void + getNota1(): double + setNota1(double): void + getNota2(): double + setNota2(double): void + calcularMedia():double



## Exercício #1

- ▶ Crie a classe aluno tal como apresentada no diagrama ao lado.
  - ▶ Crie o construtor, que recebe por parâmetro o nome e as duas notas do aluno
  - ▶ Crie os métodos get e set para cada um dos atributos
  - ▶ O método calcularMedia retorna um valor do tipo double representando a média aritmética das duas notas do aluno
- ▶ Crie uma classe TestarArrayAluno que possua um array de objetos, chamado turma, do tipo Aluno, com capacidade para 5 objetos.
  - ▶ Crie os 5 objetos do tipo aluno. Pelo construtor, atribua o nome e as notas de cada aluno.
  - ▶ Itere sobre o array e apresente uma lista contendo o nome e a média de cada um dos alunos

Aluno
- nome: String - nota1: double - nota2: double
+ Aluno(String, int, int) + getName(): String + setName(String): void + getNota1(): double + setNota1(double): void + getNota2(): double + setNota2(double): void + calcularMedia():double



## Iteração: buscando elemento no array

- ▶ Verificar se determinado elemento se encontra no array

```
Scanner sc = new Scanner(System.in);
int[] pins = { 9, 3, 7, 2 };

System.out.println("Digite o número que deseja buscar: ");
int procurar = sc.nextInt();
boolean achou = false;
for (int i = 0; i < pins.length; i++)
{
    if (pins[i] == procurar){
        achou = true;
        break;
    }
}

if (achou){
    System.out.println("O número que você procura está no array!");
}else{
    System.out.println("O número que você procura não está no array!");
}
```

## Iteração: buscando elemento no array

- Verificar se determinado elemento se encontra no array (buscando objetos)

```
public class TesteArrayContatos {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Contato [] agenda = new Contato[5];

        agenda[0] = new Contato("Maria", "123456");
        agenda[1] = new Contato("Pedro", "654321");
        agenda[2] = new Contato("Marta", "951753");
        agenda[3] = new Contato("João", "357159");
        agenda[4] = new Contato("Mirim", "456258");

        System.out.println("Digite o nome contato que deseja buscar: ");
        String buscar = sc.nextLine();
        Contato contatoEncontrado = null;
        for (int i=0; i<agenda.length; i++){
            if (agenda[i].getNome().equals(buscar)){
                contatoEncontrado = agenda[i];
                break;
            }
        }

        if (contatoEncontrado != null){
            System.out.println("O telefone de "+contatoEncontrado.getNome() + " é: ");
            System.out.println(contatoEncontrado.getTelefone());
        }else{
            System.out.println("Contato inexistente na agenda");
        }
    }
}
```

## Exercício #2

- ▶ Para o mesmo array de alunos construído no Exercício #1:
  - ▶ Verifique se um determinado aluno se encontra no array turma. Caso o aluno esteja no array, exiba o seu nome e a sua média.
  - ▶ Verifique se existe algum aluno abaixo da média (considere que a média é 60).
  - ▶ Conte quantos alunos estão acima da média.

Aluno
- nome: String - nota1: double - nota2: double
+ Aluno(String, int, int) + getNome(): String + setNome(String): void + getNota1(): double + setNota1(double): void + getNota2(): double + setNota2(double): void + calcularMedia(): double



## Referências

- ▶ <http://www.hardware.com.br/artigos/programacao-orientada-objetos/>
- ▶ <http://www.fontes.pro.br/educacional/materialpaginas/java/arquivos/jdbc/jdbc.php>
- ▶ <http://www.dm.ufscar.br/~waldeck/curso/java>
- ▶ Campos, Edmilson. Arrays: Vetores e Matrizes em Java. Notas de aula. Disponível em: [www.edmilsoncampos.net](http://www.edmilsoncampos.net)