

# PADRÕES DE PROJETO

PROF. DEMÉTRIUS DE CASTRO

PROF2303@UNIESP.EDU.BR

83 9 87730383

WWW.DEMETRIUSDECASTRO.COM.BR

# REVISÃO JAVA

## Parte Final

# REVISÃO JAVA

## ARRAY

Os arrays ou matrizes, como são conhecidos pelo Java, fazem parte do pacote `java.util` na coleção da API do Java. São objetos de recipientes que contém um número fixo de valores de um único tipo.

O comprimento de um array é estabelecido quando criado, sendo que após a criação o seu comprimento fica fixo.

Cada item em um array é chamado de elemento, e cada elemento é acessado pelo número, o índice.

# REVISÃO JAVA

## ARRAY

3	41	28	79	15	10
0	1	2	3	4	5

# REVISÃO JAVA

## DECLARANDO ARRAY

Na declaração de um array, cada elemento recebe um valor padrão, sendo 0 (**zero**) para números de tipo primitivo, falso (**false**) para elementos booleanos e nulo (**null**) para referências.

Ex.:

```
int[] a = new int[6]; ou,
```

```
int[] b;
```

```
b = new int[6]; ou,
```

```
int[] c = {2, 5, 14, 8, 9, 10};
```

# REVISÃO JAVA

## DECLARANDO ARRAY

Declare um array que receba 10 valores do tipo inteiro, e os imprima em ordem crescente.

# REVISÃO JAVA

## MODIFICADORES DE ACESSO

**public** : É o menos restritivo de todos. Atributos e métodos declarados como public em uma classe podem ser acessados pelos métodos da própria classe, por classes derivadas desta e por qualquer outra classe em qualquer outro pacote (veremos mais sobre pacotes e classes derivadas posteriormente).

**protected** : Atributos e métodos definidos como protected são acessíveis pelos métodos da própria classe e pelas classes derivadas.

**private** : É o mais restritivo. Atributos e métodos declarados como private só podem ser acessados pelos métodos da própria classe.

# REVISÃO JAVA

## MÉTODOS

É uma sub-rotina que é executada por um objeto ao receber uma mensagem. Os métodos determinam o comportamento dos objetos de uma classe e são análogos a funções ou procedimentos da programação estruturada. O envio de mensagens (chamada de métodos) pode alterar o estado de um objeto



# REVISÃO JAVA

## MÉTODOS

### Declaração

```
public void saudacao(String nome) {  
    System.out.println("Olá " + nome + "!");  
}
```

### Chamada

```
saudacao("Mundo");
```

# REVISÃO JAVA

## MÉTODOS

Escreva um método que receba 2 valores inteiro, faça a sua soma e retorne o seu valor para o usuário.

# REVISÃO JAVA

## CLASSES

Classe é uma estrutura fundamental que define o comportamento e as propriedades de objetos. Ela contém métodos (funções) e campos (variáveis) que representam o comportamento e o estado dos objetos que serão criados a partir dessa classe.

# REVISÃO JAVA

## CLASSES E OBJETOS

### Declaração

```
class Carro {  
    String modelo;  
    int ano;  
  
    void ligar() {  
        system.out.println("O carro ligou");  
    }  
}
```

# REVISÃO JAVA

## CLASSES E OBJETOS

Criando objeto para a classe carro

```
Carro meuCarro = new Carro();  
meuCarro.modelo = "Cronos";  
meuCarro.ano = 2013;  
meuCarro.ligar();  
}
```

# REVISÃO JAVA

## CLASSES E OBJETOS

Escreva uma classe chamada `compras`, que contenha as variáveis `descricao`, `valor` e `quantidade`. Crie um objeto que passe esses parâmetros e retorne o valor total da compra de acordo com a quantidade do produto.

# REVISÃO JAVA

## HERANÇA

A herança permite que uma classe herde características de outra, promovendo reutilização de código.

```
public class Animal {  
    public void emitirSom() {  
        System.out.println("Som do animal");  
    }  
}
```

```
public class Cachorro extends Animal {  
    // Cachorro herda o método emitirSom da classe Animal  
}
```

# REVISÃO JAVA

## HERANÇA

Crie uma classe chamada “pessoa”, que tenha as variáveis “nome”, “cpf” e “data\_nascimento”, e o construtor para preencher estes atributos.

Logo após, crie as classes “aluno”, “funcionário” e “professor” que devem herdar os atributos da classe “pessoa”.

A classe “aluno” deve ter um atributo chamado “matricula”.

A classe “professor” deve ter os atributos “salario” e “disciplina”.

A classe “funcionário” deve ter os atributos “salario”, “data\_admissão” e “cargo”.



# REVISÃO JAVA

## POLIMORFISMO

É o termo definido em linguagens orientadas a objeto, como por exemplo Java, C# e C++, que permite ao desenvolvedor usar o mesmo elemento de formas diferentes. Polimorfismo denota uma situação na qual um objeto pode se comportar de maneiras diferentes ao receber uma mensagem.

Temos 2 tipos de polimorfismo, o Estático (sobrecarga) e o Dinâmico (Sobreposição).

# REVISÃO JAVA

## POLIMORFISMO

O Polimorfismo Estático se dá quando temos a mesma operação implementada várias vezes na mesma classe. A escolha de qual operação será chamada depende da assinatura dos métodos sobrecarregados.

O Polimorfismo Dinâmico acontece na herança, quando a subclasse sobrepõe o método original. Agora o método escolhido se dá em tempo de execução e não mais em tempo de compilação. A escolha de qual método será chamado depende do tipo do objeto que recebe a mensagem.

# REVISÃO JAVA

## POLIMORFISMO

```
class Animal {  
    void fazerSom() {  
        System.out.println("Som do animal");  
    }  
}  
  
class Cachorro extends Animal {  
    @Override  
    void fazerSom() {  
        System.out.println("Latir");  
    }  
}  
  
class Gato extends Animal {  
    @Override  
    void fazerSom() {  
        System.out.println("Miar");  
    }  
}
```