

A thick black L-shaped frame is positioned on the left and bottom edges of the slide, framing the central text.

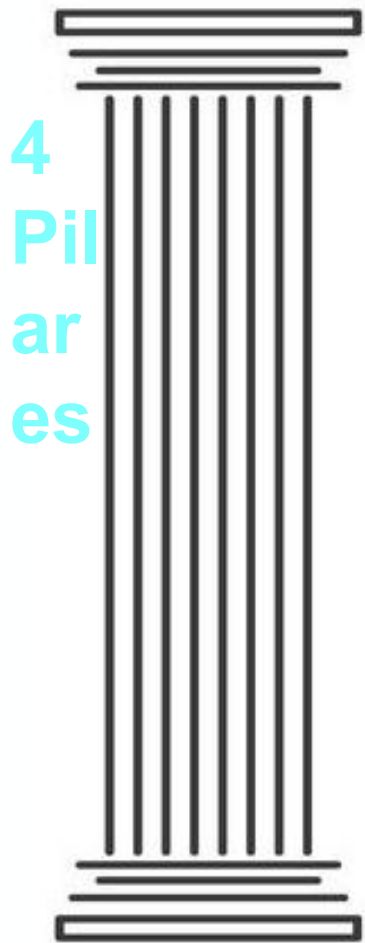
ALGORITMOS E PROGRAMAÇÃO II

Professor: Jânio E. Teixeira Júnior



POLIMORFISMO

Orientação a Objetos



1. Abstração
2. Encapsulamento
3. Herança
4. Polimorfismo

Polimorfismo

- A palavra **polimorfismo** advém do grego ***polimorfos*** e significa diversas formas. Em POO o conceito segue os mesmos princípios da palavra.
- Polimorfismo nada mais é que a definição de métodos com o mesmo nome na própria classe ou na superclasse.
- Pode ser aplicado de duas formas:
 - *Sobrecarga (overload)*;
 - *Sobreposição (override)*.

Sobrecarga - *Overload*

- A sobrecarga consiste na possibilidade de ter em uma mesma classe vários métodos com o mesmo nome, porém com **assinaturas diferentes**;
 - *A assinatura de um método consiste em: tipo de retorno, nome do método e lista de parâmetros.*
- ❖ **Em Java a sobrecarga de métodos é definida apenas pela lista de parâmetros.**

Sobrecarga - Overload

```
public class Soma {  
    public int somar(int v1, int v2) {  
    public double somar(double v1, double v2) {  
    public double somar(int v1, double v2) {  
    public double somar(double v1, int v2) {  
}
```

Sobrecarga – Utilizando o Cast

```
public class Soma {  
    public int somarInteiro(int v1, int v2) {  
        return v1 + v2;  
    }  
    public int somarInteiro(double v1, double v2) {  
        return (int)v1 + (int)v2;  
    }  
    public int somarInteiro(int v1, double v2) {  
        return v1 + (int)v2;  
    }  
    public int somarInteiro(double v1, int v2) {  
        return (int)v1 + v2;  
    }  
    public static void main(String[] args) {  
        Soma soma = new Soma();  
        System.out.println(soma.somarInteiro(22.5, 5));  
    }  
}
```

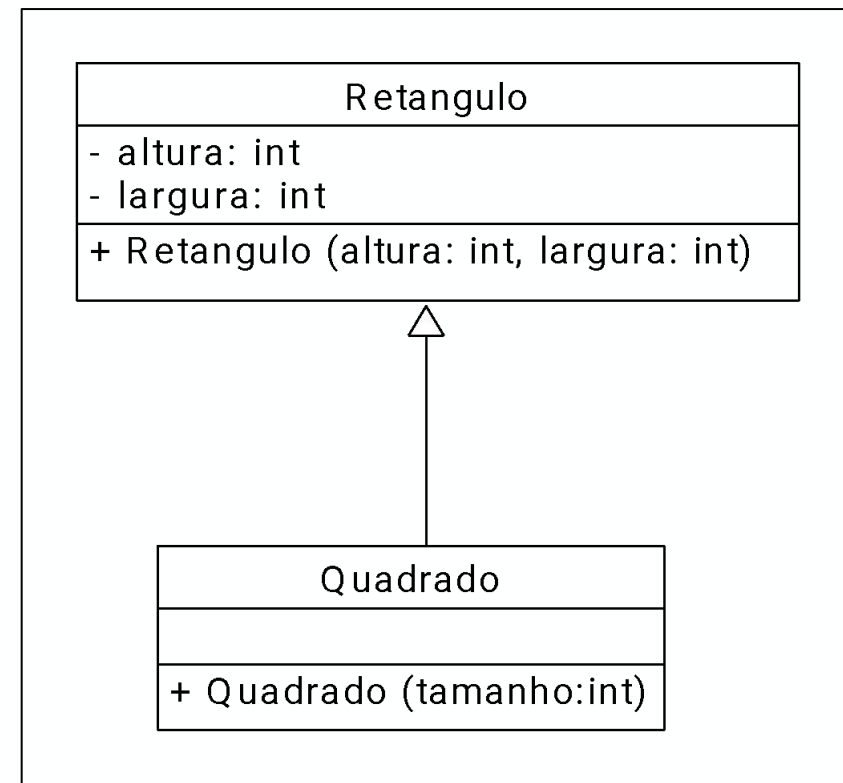
Cast

- O **cast** é a forma mais comum de conversão de tipos;
- Pode ser realizado de duas formas: Implícita e Explícita.

```
public static void main(String[] args) {  
    double i = 5 / 9;  
    double i2 = (double) (5 / 9);  
    double i3 = (double) 5 / 9;  
    System.out.println(i);  
    System.out.println(i2);  
    System.out.println(i3);  
}
```


Comando instanceof

- *Instanceof* – “Instância de”;
- Comando utilizado para verificar a instância de um determinado objeto.



```
public static void imprimir(Retangulo retangulo) {
    if (retangulo instanceof Quadrado)
        System.out.print("Quadrado :");
    else if (retangulo instanceof Retangulo)
        System.out.print("Retangulo:");

    System.out.println(" altura="+retangulo.getAltura()
        + " largura:"+retangulo.getLargura());
}
```

Sobrecarga de Construtores

```
public class Pessoa {  
    public Pessoa() { // Construtor default  
    public Pessoa(int id) {  
    public Pessoa(int id, String nome) {  
    public Pessoa(String nome, String cpf) {  
}
```

Sobrescrita - *Override*

- Sobrescrita, Sobreposição ou Redefinição de métodos;
- Sua aplicação é realizada sobre métodos existentes nas superclasses. Como o nome já diz, esse conceito tem como objetivo redefinir o funcionamento do método da classe Pai.

Sobrescrita - Override

```
public class Quadrado extends Retangulo {
```

```
    public Quadrado(int tamanho) {
```

```
        @Override
```

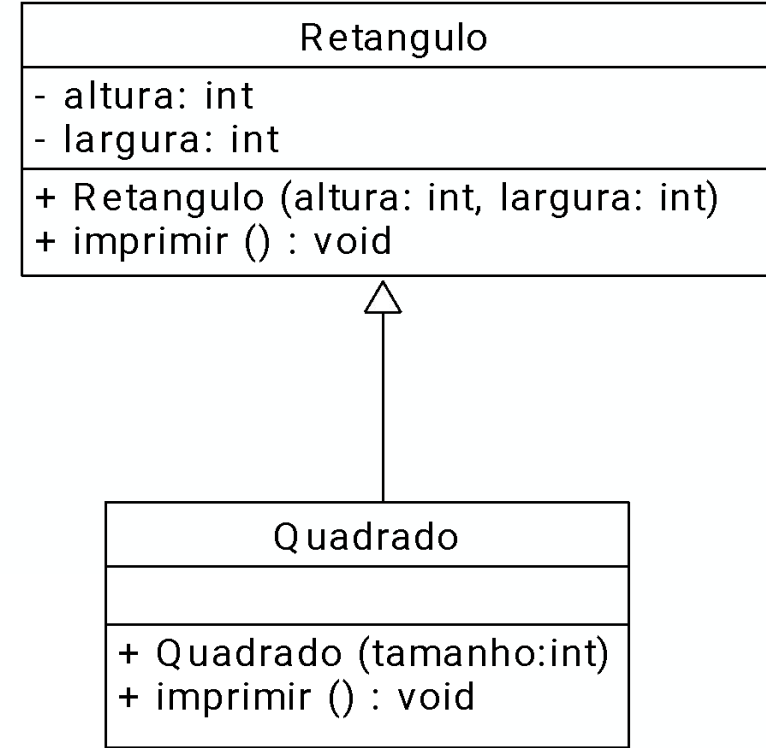
```
        public void imprimir() {
```

```
            System.out.print("Quadrado:");
```

```
            System.out.println("  altura=" + getAltura()  
                               + "  largura=" + getLargura());
```

```
        }
```

```
    }
```



Referências

- Livro: Coelho, Alex. Java com Orientação a Objetos. Editora Ciência Moderna Ltda., 2012.

A thick black L-shaped frame is positioned on the left and bottom edges of the slide, framing the central text.

ALGORITMOS E PROGRAMAÇÃO II

Obrigado!