

POO usando Java: Herança

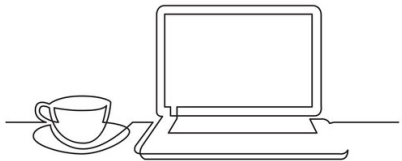
Programação Orientada a Objetos

Profs. Aleffer Rocha

aleffer.rocha@ifpr.edu.br

Prof. William Deus

william.deus@ifpr.edu.br

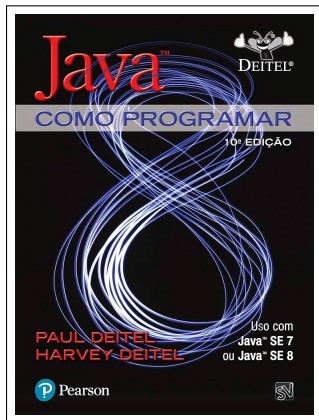


INSTITUTO FEDERAL
Paraná

Campus
Pinhais

Material atualizado em 23 de setembro de 2024

Materiais utilizados



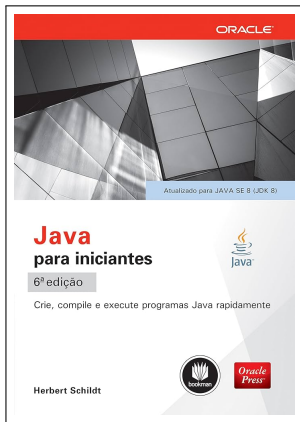
DEITEL, Paul; DEITEL, Harvey. **Java: como programar**. 10. ed. São Paulo, SP: Pearson Education do Brasil, 2017.

Materiais utilizados



HORSTMANN, Cay S. **Conceitos de computação com o essencial de Java.** 3. ed. Porto Alegre, RS: Bookman, 2005.

Materiais utilizados



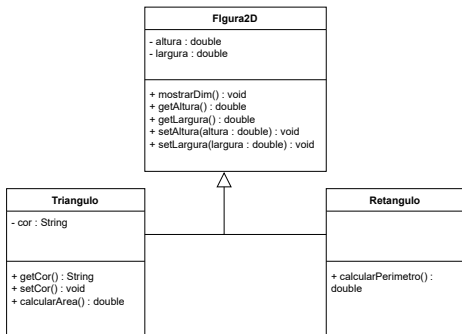
SCHILD, Herbert. **Java para iniciantes:** crie, compile e execute programas Java rapidamente. 6. ed. Porto Alegre, RS: Bookman, 2015.x

A **herança** é a possibilidade de uma classe utilizar os atributos e métodos de uma outra classe como se fossem seus.

Generalização e **especialização** são os nomes entre dois elementos, um mais geral (elemento PAI) e um mais específico (elemento FILHO).

O mais específico é completamente consistente com o mais geral e adiciona informações ao elemento mais geral.

Quando temos herança, a classe que fornece herança (generalizada) também é chamada de **superclasse**, e a classe que recebe a herança (especializada) também é chamada de **subclasse**.



A notação de herança é dada por uma seta vazia que liga as classes. A seta deve, sempre, estar apontando para a superclasse.

Aspectos básicos da herança

Java dá suporte à herança, permitindo que uma classe incorpore outra em sua declaração.

Isso é feito com o uso da palavra-chave `extends`. Portanto, a subclasse traz acréscimo (estende) à superclasse.

Aspectos básicos da herança

Uma vantagem da herança é a **reutilização de código**.

Quando definimos uma subclasse, especificamos campos de instância adicionais, métodos adicionais e métodos alterados ou redefinidos.

Acesso a membros e herança

Ainda que uma classe inclua todos os membros de sua superclasse, não poderá acessar os membros declarados como **private**.

Os membros **protected** de uma superclasse podem ser acessados por membros dessa superclasse, de suas subclasses e outras classes no mesmo pacote.

Superclasse Figura2D

```
1 public class Figura2D {
2     private double largura;
3     private double altura;
4
5     public double getLargura(){ return largura; }
6
7     public double getAltura(){ return altura; }
8
9     public void setLargura(double largura){ this.largura = largura; }
10
11    public void setAltura(double altura){ this.altura = altura; }
12
13    public void mostrarDim(){
14        System.out.println("Largura e altura: " + largura + " e " + altura);
15    }
16 }
```

Subclasse Triangulo

```
1 public class Triangulo extends Figura2D {
2     private String cor;
3
4     public double calcularArea(){
5         return getLargura() * getAltura() / 2;
6     }
7
8     public void mostrarCor(){
9         System.out.println("O triangulo e: " + cor);
10    }
11
12    public void setCor(String cor) {
13        this.cor = cor;
14    }
15 }
```

Criação de objetos Triangulo

```
1 public class Figura {
2     public static void main(String args []){
3         Triangulo t1 = new Triangulo();
4         Triangulo t2 = new Triangulo();
5
6         t1.setLargura(4.0);
7         t1.setAltura(4.0);
8         t1.setCor("Verde");
9
10        t2.setLargura(8.0);
11        t2.setAltura(12.0);
12        t2.setCor("Azul");
13    }
14 }
```

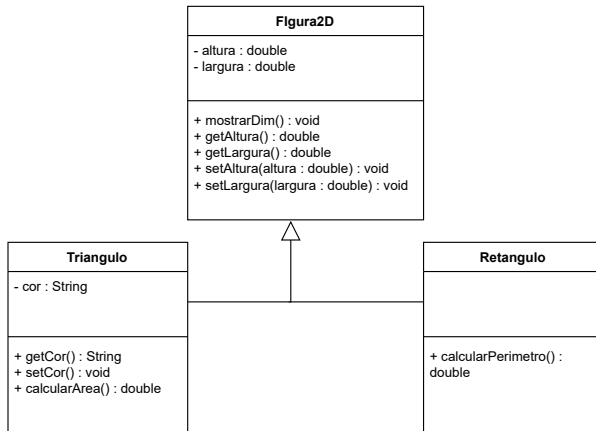
Exercícios para fixação



Vamos colocar em prática o que aprendemos na aula de hoje.

Exercício 1

Faça a implementação da classe Retangulo.



Exercício 2

Classes Pessoa e Aluno

- ▶ Crie uma classe chamada Pessoa com os atributos nome e idade, e seus respectivos métodos *get* e *set*.
- ▶ Crie uma subclasse chamada Aluno, que herda de Pessoa. A classe Aluno deve conter o atributo *matricula*.
- ▶ Implemente o método construtor nas duas classes, de forma que o construtor de Aluno inicialize o atributo *matricula* e os atributos da superclasse.
- ▶ Escreva um programa para instanciar um Aluno e exibir seus dados.