

# Java: como programar

## Programação Orientada à Objetos

- Douglas Baptista de Godoy

 [/in/douglasbgodoy](https://www.linkedin.com/in/douglasbgodoy)

 [github.com/douglasbgodoy](https://github.com/douglasbgodoy)

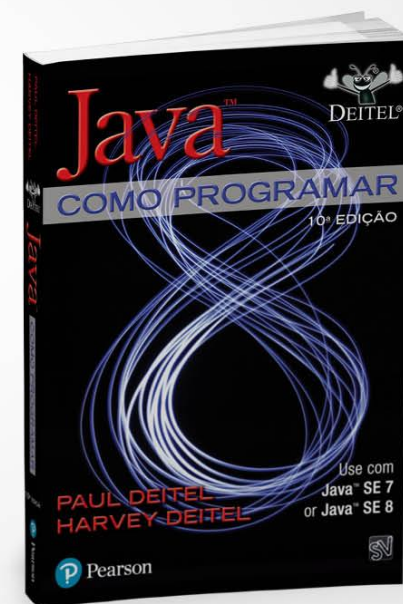
# Informação

Obs: Esta aula é baseada nos livros textos, e as transparências são baseadas nas transparências providenciadas pelos autores.

DEITEL, P. J.; DEITEL, H. M. **Java**: como programar. 10. ed. São Paulo, SP: Pearson, 2017. *E-book*. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 27 fev. 2024.

# Capítulo 9:

## Programação orientada a objetos: herança



# Exercício 1: Hierarquia de Funcionários

- Crie uma hierarquia de classes para representar funcionários de uma empresa. Todos os funcionários devem ter nome, salário base e um método para calcular o salário. Além disso, crie duas subclasses:
  1. Gerente: recebe um bônus fixo adicional ao salário.
  2. Desenvolvedor: recebe um percentual adicional sobre o salário base.
- Implemente um programa que cria objetos dessas classes e exibe seus salários.

# Exercício 2: Hierarquia de Veículos

- Crie uma classe base chamada Veiculo que contenha os atributos marca e modelo e um método `exibirDados()`. Em seguida, crie as subclasses:
  - 1. Carro: possui um atributo adicional `quantidadePortas`.
  - 2. Moto: possui um atributo adicional `cilindradas`.
- Crie um programa que instancie objetos dessas classes e exiba os dados dos veículos.

# Exercício 3: Sistema de Contas Bancárias

- Crie uma classe ContaBancaria com os atributos titular, saldo e um método para sacar() e depositar(). Em seguida, crie as subclasses:
  1. ContaCorrente: cobra uma taxa fixa a cada saque.
  2. ContaPoupanca: possui um método adicional para renderJuros().
- Crie um programa que instancie objetos dessas classes, realize depósitos, saques e exiba os saldos.

# Exercício 4: Sistema de Ingressos

Crie uma classe chamada Ingresso que possua um atributo valor e um método toString que retorne à informação do valor do ingresso.

1. Crie uma classe IngressoVIP, que herda de Ingresso e possui um atributo valor Adicional. O método toString da classe IngressoVIP deve considerar que o valor do ingresso é o valor da superclasse somado ao valor Adicional do IngressoVIP.
2. Crie uma classe para testar os objetos das classes Ingresso e IngressoVIP.

# Referências Bibliográficas

- DEITEL, P. J.; DEITEL, H. M. **Java**: como programar. 10. ed. São Paulo, SP: Pearson, 2017. .