

# Desenvolvimento de Sistemas

- Douglas Baptista de Godoy

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

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

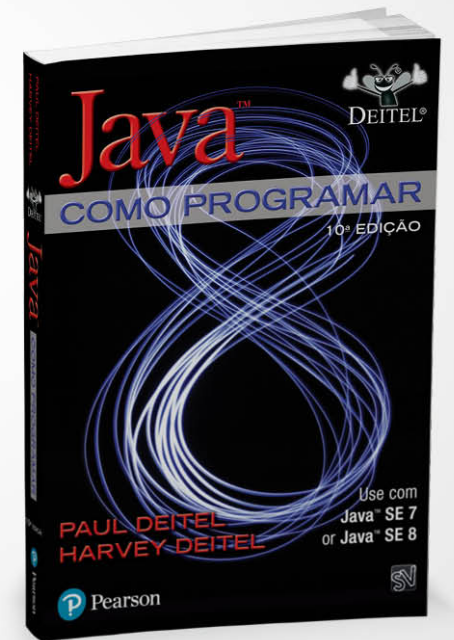
# Informação

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 1:

## Introdução a computadores, internet e Java



# Introdução

- **Java**: uma das linguagens de programação mais utilizadas no mundo.
- Instruções fazem com que os computadores realizem tarefas.
- O software (isto é, as instruções que você escreve) controla o hardware (isto é, os computadores).

# Introdução

- ***Programação orientada a objetos*** — atualmente a metodologia-chave de programação.
- Ele também é amplamente utilizado para implementar aplicativos e softwares baseados na internet para dispositivos que se comunicam por uma rede.

# Introdução

- De acordo com um estudo realizado pela Gartner, os dispositivos móveis continuarão a ultrapassar os PCs como os dispositivos de computação principais dos usuários.
- Isso está criando oportunidades profissionais significativas para pessoas que programam aplicativos móveis, muitos dos quais são programados em Java.

## Hardware e Software

- Os computadores processam dados sob o controle de conjuntos de instruções chamados **programas de computador**.
- Esses programas de software orientam o computador por meio de ações ordenadas especificadas por pessoas chamadas **programadores de computador**.
- **Programação orientada a objetos**: melhora a produtividade do programador, reduzindo os custos de desenvolvimento de softwares.

# Introdução à tecnologia de objetos

- Para realizar uma tarefa em um programa é necessário um **método**. O método armazena as declarações do programa que, na verdade, executam as tarefas; além disso, ele oculta essas declarações do usuário.
- No Java, criamos uma unidade de programa chamada **classe** para armazenar o *conjunto de métodos* que executam as tarefas dela.



# Introdução à tecnologia de objetos

- Assim como alguém tem de *fabricar um carro* a partir dos desenhos de engenharia antes que possa dirigi-lo, você deve *construir um objeto* de uma classe antes que um programa possa executar as tarefas que os métodos da classe definem.
- O processo para fazer isso é chamado *instanciação*. Um objeto é então referido como uma **instância** da sua classe.

# Introdução à tecnologia de objetos

- Ao dirigir um carro, o ato de pressionar o acelerador envia uma *mensagem* para o veículo realizar uma tarefa — isto é, ir mais rápido. Da mesma forma, você *envia mensagens para um objeto*.
- Cada mensagem é implementada como uma **chamada de método** que informa a um método do objeto a maneira de realizar sua tarefa.

# Introdução à tecnologia de objetos

- Um objeto tem atributos que ele incorpora à medida que é usado em um programa. Esses atributos são especificados como parte da classe do objeto.
- Por exemplo, um objeto conta bancária tem um *atributo saldo* que representa a quantidade de dinheiro disponível.
- Os atributos são especificados pelas **variáveis de instância** da classe.

# Introdução à tecnologia de objetos

- Classes (e seus objetos) **encapsulam**, isto é, contêm seus atributos e métodos.
- Os atributos e métodos de uma classe (e de seu objeto) estão intimamente relacionados. Os objetos podem se comunicar entre si, mas eles em geral não sabem como outros objetos são implementados — os detalhes de implementação permanecem *ocultos* dentro dos próprios objetos.
- Esse **ocultamento de informações** é crucial à boa engenharia de software.

# Introdução à tecnologia de objetos

- Uma nova classe de objetos pode ser criada por meio de **herança** — ela (chamada **subclasse**) começa com as características de uma classe existente (chamada **superclasse**), possivelmente personalizando-as e adicionando aspectos próprios.
- O Java também suporta **interfaces** — coleções de métodos relacionados que normalmente permitem informar aos objetos *o que* fazer, mas não *como* fazer.

# Introdução à tecnologia de objetos

- Se esse processo envolve analisar e projetar o sistema de um ponto de vista orientado a objetos, ele é chamado de **processo de análise e projeto orientados a objetos** (*object-oriented analysis and design* — OOAD).
- Linguagens como Java são orientadas a objetos. A programação nessa linguagem, chamada **programação orientada a objetos** (*object-oriented programming* — OOP), permite-lhe implementar um projeto orientado a objetos como um sistema funcional.

# Java

- Linguagem de programação orientada a objetos.
- “Escreva uma vez, execute em qualquer lugar”.
- Utilizado para desenvolver aplicativos corporativos de grande porte, aprimorar a funcionalidade de servidores da web, fornecer aplicativos para dispositivos voltados ao consumo popular (por exemplo, telefones celulares, smartphones, televisão, *set-up boxes* etc.) e para muitos outros propósitos.
- Linguagem-chave para desenvolvimento de aplicativos Android adequados a smartphones e tablets.

# Java

- Você pode criar cada classe e método de que precisa para formar seus programas Java. Porém, a maioria dos programadores Java tira proveito das ricas coleções de classes existentes e métodos nas **bibliotecas de classe Java**, também conhecidas como **Java APIs (*application programming interfaces*)**.



# Referências Bibliográficas

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