Programação Orientada a Objetos

Prof.ª Ma. Jessica Oliveira

Católica
Universidade Católica de Brasília

Aula 05 – 24/03/2025

Herança: reutilização de código e hierarquia de classes.

Católica
Universidade Católica de Brasília

O que é herança?

- É um dos **pilares fundamentais** da POO e tem como principal objetivo promover a **reutilização de código** e o estabelecimento de uma **hierarquia de classes**.
- Trata-se de um mecanismo que permite que uma classe (chamada de subclasse ou classe derivada) herde características (atributos e métodos) de outra classe (conhecida como superclasse ou classe base).
- A principal ideia por trás da herança é que uma subclasse pode ser considerada uma especialização de sua superclasse.
- Isso significa que **a subclasse possui todas as características da superclasse e pode adicionar comportamentos e atributos específicos que a tornam mais especializada**.

Relação de generalização-especialização.

- Na prática, a relação de herança pode ser descrita como uma relação de "é um" (em inglês, "*is-a*"). Por exemplo:
  - Um **Carro** é um **Veículo**.
  - Um **Cachorro** é um **Animal**.
  - Um **Gerente** é um **Funcionário**.
- Essas afirmações indicam que as subclasses (Carro, Cachorro, Gerente) têm características comuns com suas superclasses (Veículo, Animal, Funcionário), mas também podem ter características e comportamentos específicos.

Por que utilizar herança?

- **Reutilização de código:** evita a duplicidade ao aproveitar atributos e métodos já definidos em uma superclasse.
- **Especialização:** permite que a subclasse adicione ou sobrescreva funcionalidades específicas.
- **Facilidade de manutenção:** as atualizações na superclasse se propagam automaticamente para as subclasses, reduzindo a necessidade de alterações manuais.
- **Organização hierárquica:** facilita a modelagem de sistemas complexos com uma estrutura clara e organizada.
- **Polimorfismo:** permite que objetos de subclasses sejam tratados como objetos da superclasse, facilitando a flexibilidade no uso de métodos.

Como funciona a herança em Java?

- É implementada utilizando a palavra-chave **`extends`**.
- Isso significa que uma subclasse "estende" uma superclasse, herdando seus atributos e métodos.

Tipos de herança em Java.

- A linguagem Java admite apenas herança simples, ou seja, uma subclasse pode herdar diretamente de apenas uma superclasse.
- Isso evita problemas de ambiguidade que podem surgir na herança múltipla, presente em outras linguagens como C++.
- Por outro lado, Java permite a herança multinível, na qual uma subclasse herda de uma classe que já é derivada de outra.

Method Overriding.

- Um conceito importante na herança é a sobrescrita de métodos.
- Ela permite que a subclasse forneça uma implementação específica para um método que já existe na superclasse.
- Para sobrescrever um método, basta criar um método na subclasse com a mesma assinatura (nome, tipo de retorno e parâmetros) do método da superclasse.
- A anotação **`@Override`** é usada para indicar que o método está sendo sobrescrito, embora seu uso não seja obrigatório.

Utilização do super.

- Acessar métodos da superclasse que foram sobrescritos pela subclasse.
- Chamar o construtor da superclasse.
- Referenciar atributos da superclasse que foram ocultados pela subclasse.

Microprojeto 05

🎯 **Objetivo:** aplicar o conceito de herança para reutilizar código e estruturar hierarquias.

Para o pós-aula...

- Criar subclasses que herdam de uma classe base.
- Implementar uso de `super` para acessar métodos da superclasse.
- Demonstrar a utilização correta da herança.
- Versionamento: Colaboração e *Pull Requests* no GitHub.
- Relatório técnico abordando os ganhos da reutilização de código com herança.

Dúvidas?

jessica.oliveira@p.ucb.br