

# Programação Orientada a Objetos

Prof.<sup>a</sup> Ma. Jessica Oliveira

**Aula 04 – 17/03/2025**

# **Relações entre Classes e Composição.**

# Introdução.

- Em um sistema orientado a objetos, as classes representam entidades do mundo real e seus comportamentos.
- No entanto, nenhuma classe atua isoladamente em um *software* bem projetado.
- Em aplicações reais, os objetos interagem entre si para formar sistemas complexos, e essas interações são chamadas de **relações entre classes**.

# Introdução.

- As relações entre classes permitem:
  - Modelar o comportamento real do sistema de forma mais próxima da realidade;
  - Modularizar e organizar o código de maneira eficiente;
  - Facilitar a reutilização de componentes do sistema;
  - Criar um design coeso e de fácil manutenção.

# Associação.



# Definição.

- É a forma mais geral de relacionamento entre classes. Ela representa uma ligação entre objetos, sem necessariamente indicar uma relação de dependência forte entre eles.
- Ocorre quando uma classe usa outra classe sem que haja um vínculo de propriedade entre elas. Um objeto pode se associar a outro para utilizar seus serviços ou interagir de alguma maneira.
- Pode ser unidirecional (quando apenas uma classe conhece a outra) ou bidirecional (quando ambas as classes conhecem uma à outra).

# Exemplo no mundo real...

- Um aluno e uma disciplina têm uma associação: um aluno pode cursar várias disciplinas, e uma disciplina pode ter vários alunos matriculados.
- No entanto, ambos podem existir separadamente.

# Agregação.





# Definição.

- É uma relação do tipo "tem-um", mas com menor dependência entre os objetos.
- É uma associação especial onde um objeto contém outro, mas sem que um precise do outro para existir.
- É usada quando uma classe faz referência a outra como parte do seu conjunto, mas os objetos ainda podem existir de forma independente.

# Exemplo do mundo real...

- Uma biblioteca tem livros, mas um livro pode existir mesmo que a biblioteca seja fechada.

# Composição.



# Definição.

- É uma relação mais forte do tipo "tem-um", onde um objeto depende do outro para existir.
- Na composição, os objetos agregados não podem existir sem o objeto principal. Ou seja, a destruição do objeto principal implica a destruição dos objetos que fazem parte dele.

# Exemplo do mundo real...

- Um pedido possui itens de pedido.
- Se o pedido for cancelado, os itens perdem a razão de existir.

# Microprojeto 04

 **Objetivo:** implementar relacionamentos entre classes, garantindo modularidade e organização.

# Para o pós-aula...

- Estabelecer relações entre as classes utilizando associação, agregação ou composição.
- Criar pelo menos duas novas classes auxiliares e definir suas interações.
- Revisar a organização do código e modularidade.
- Criar um release no GitHub e documentar a evolução do projeto.

# Dúvidas?

[jessica.oliveira@p.ucb.br](mailto:jessica.oliveira@p.ucb.br)