Univ. Tecnológica Federal do Paraná
Campus Medianeira
Bacharelado em Ciência da Computação

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

UML: MODELO DE CLASSES

Prof. Dr. Alan Gavioli

alan@utfpr.edu.br

VISÕES INTERNA E EXTERNA DE UM SISTEMA O. O.

- **A funcionalidade externa de um sistema orientado a objetos é fornecida através de colaborações entre objetos.**
  - **Externamente**, os atores visualizam resultados de cálculos, relatórios produzidos, confirmações de requisições realizadas, etc (**diagrama de casos de uso** mostra isso).

MODELO DE CLASSES

- **Internamente**, os objetos colaboram uns com os outros para produzir os resultados (isto o **diagrama de casos de uso** não mostra). Essa colaboração pode ser vista sob o **aspecto dinâmico** e sob o **aspecto estrutural estático**.

- O diagrama da UML utilizado para representar os **aspectos estáticos** das colaborações entre objetos é o **diagrama de classes.**

- O modelo de classes evolui (é incrementado) durante o desenvolvimento do sistema.

ASSOCIAÇÕES

- Para representar o fato de que objetos podem se relacionar uns com os outros, utiliza-se o relacionamento de **associação**.

- Uma associação representa relacionamentos (ligações) que são formados entre objetos durante a execução do sistema.

MULTIPLICIDADES

- Representam a informação dos limites inferior e superior da quantidade de objetos aos quais um outro objeto pode estar associado.

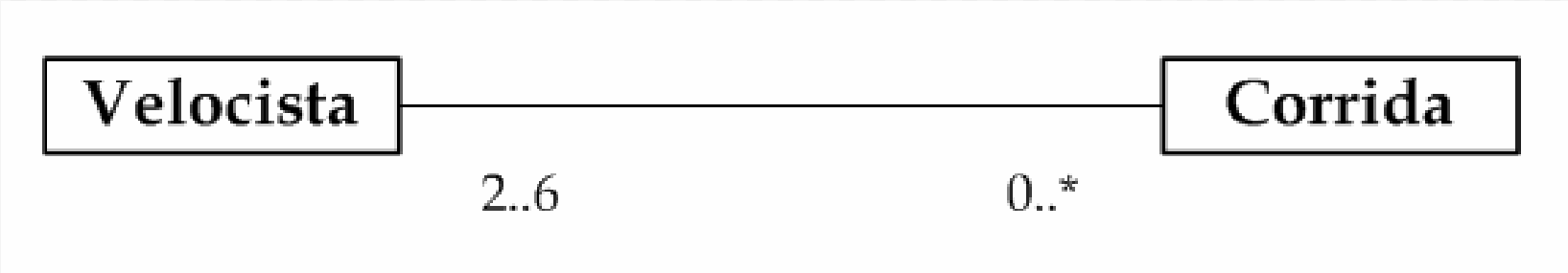
- Cada associação em um diagrama de classes possui duas multiplicidades: uma em cada extremo da linha de associação.

MULTIPLICIDADES

Nome	Simbologia
Apenas Um	1..1 (ou 1)
Zero ou Muitos	0..* (ou *)
Um ou Muitos	1..*
Zero ou Um	0..1
Intervalo Específico	$l_i..l_s$

MULTIPLICIDADES

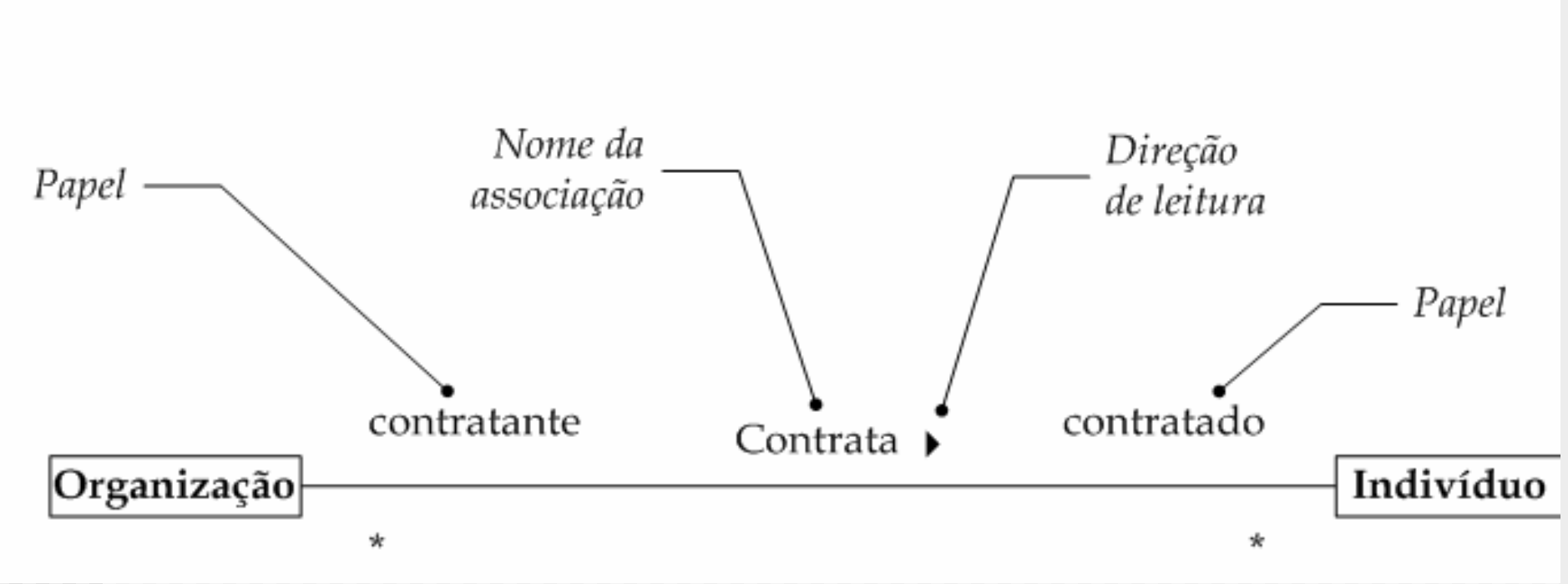
- Uma corrida está associada a, no mínimo, dois velocistas.
- Uma corrida está associada a, no máximo, seis velocistas.
- Um velocista **pode** estar associado a várias corridas.



COMPLEMENTOS PARA O RELACIONAMENTO DE ASSOCIAÇÃO

- **A UML define três recursos para complementar a notação da associação:**
  - **Nome da associação (rótulo)**: fornece algum significado semântico à associação.
  - **Direção de leitura**: indica como a associação deve ser lida.
  - **Papel**: para representar um papel específico em uma associação.

COMPLEMENTOS PARA A NOTAÇÃO DE UMA ASSOCIAÇÃO



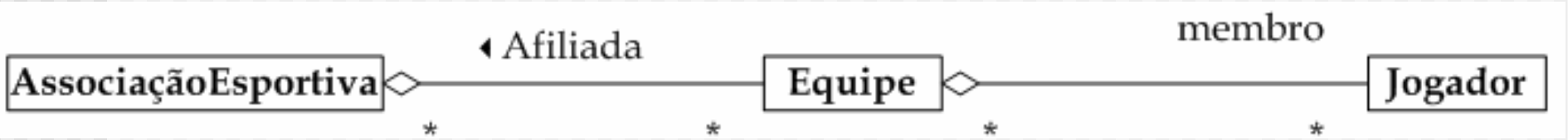
AGREGAÇÃO

- É um caso especial da associação.
- Utilizada para representar conexões que guardam uma relação **todo-parte** entre si.
- Em uma agregação, um objeto está contido no outro, ao contrário de uma associação.

AGREGAÇÃO

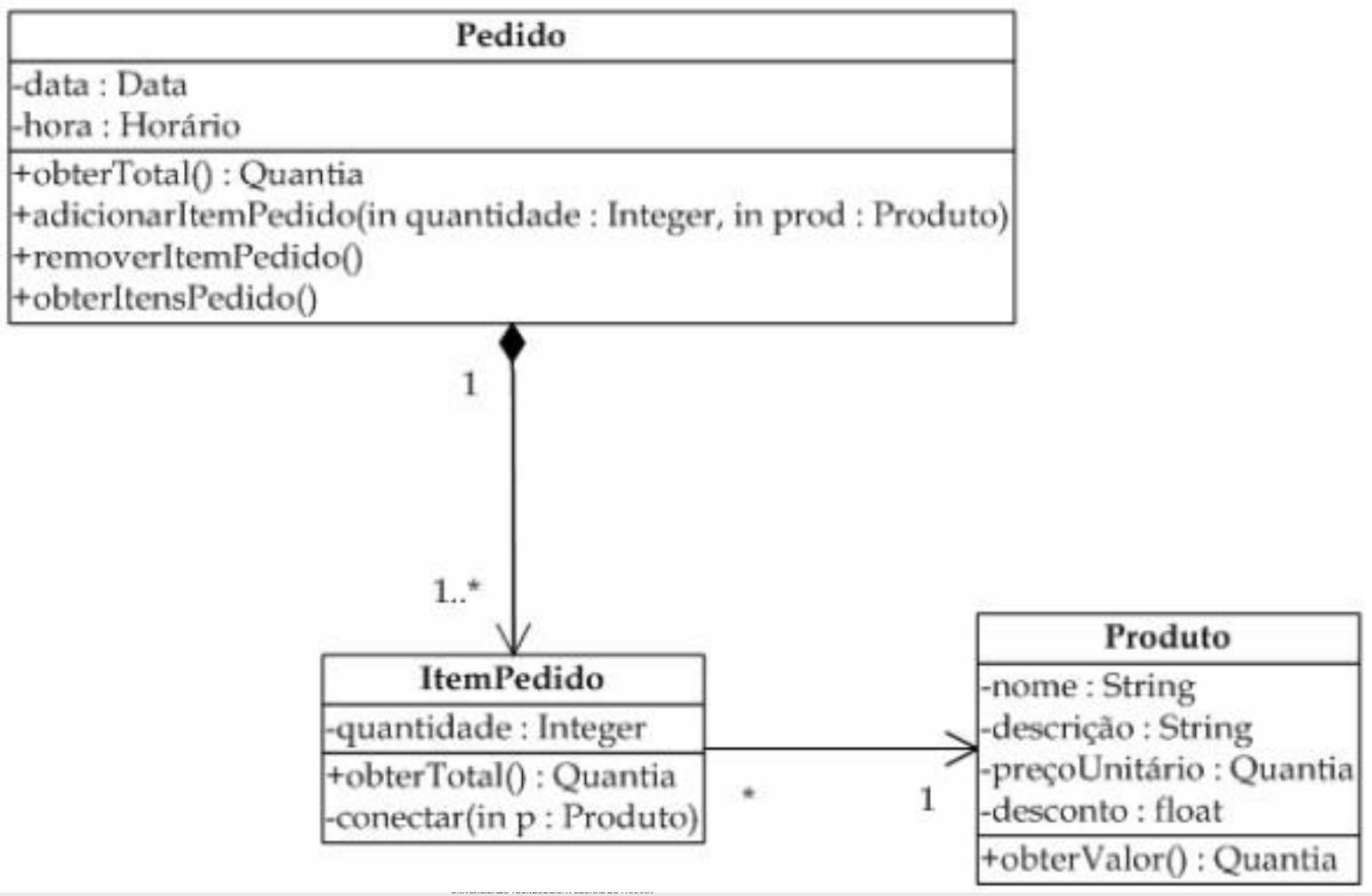
- Sejam duas classes associadas, X e Y. Se uma das perguntas a seguir for respondida com "sim", provavelmente há uma agregação onde X é todo e Y é parte:
  - ***X contém um ou mais Y?***
  - ***Y é parte de X?***

- Exemplo:



COMPOSIÇÃO

- A composição (ou agregação composta) é considerada uma forma mais forte de agregação.
- Ex:



AGREGAÇÃO x COMPOSIÇÃO

- **Na agregação simples:**
  - Os objetos que fazem parte do todo são criados e destruídos independentemente deste último.
  - Um objeto parte pode ser utilizado para compor diversos objetos todos.
  - A destruição de um desses objetos todo não implica na destruição do objeto parte.

- **Na composição (agregação composta):**
  - Os objetos parte pertencem a um único todo.
  - Objetos parte são sempre criados e destruídos pelo objeto todo.
  - No objeto todo, são definidas operações para adicionar e remover objetos componentes.

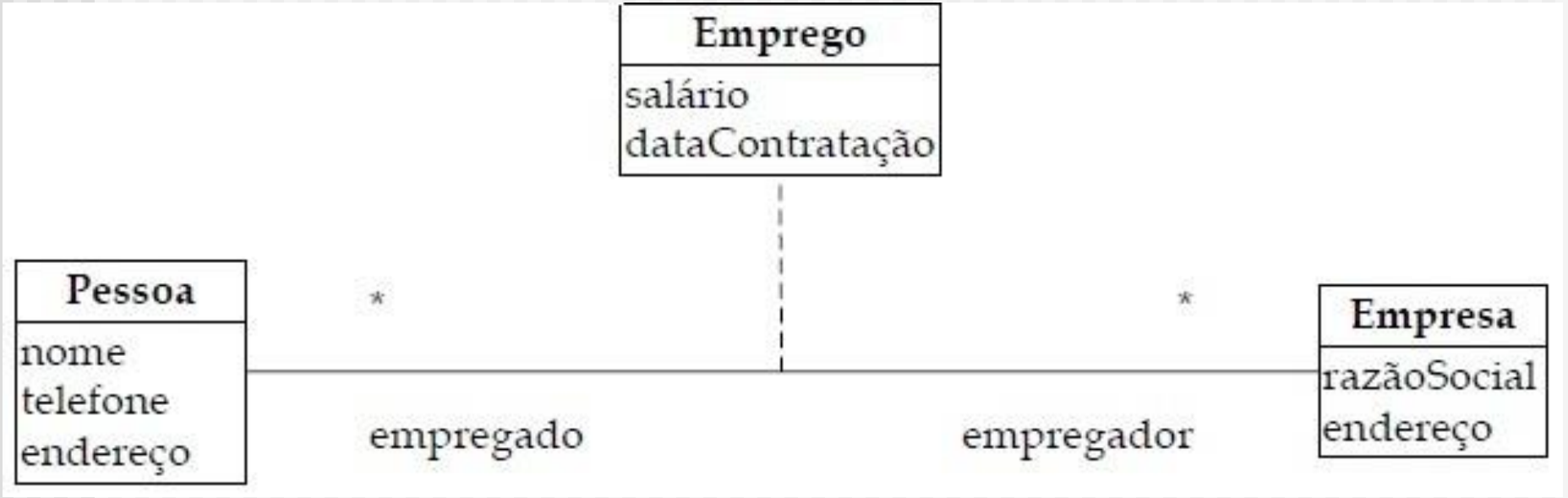
CLASSE ASSOCIATIVA

- É uma classe que está ligada a uma associação, ao invés de estar ligada a outras classes.

- É normalmente necessária quando duas ou mais classes estão associadas e é necessário manter informações sobre esta associação.

CLASSE ASSOCIATIVA

- **Exemplo:**

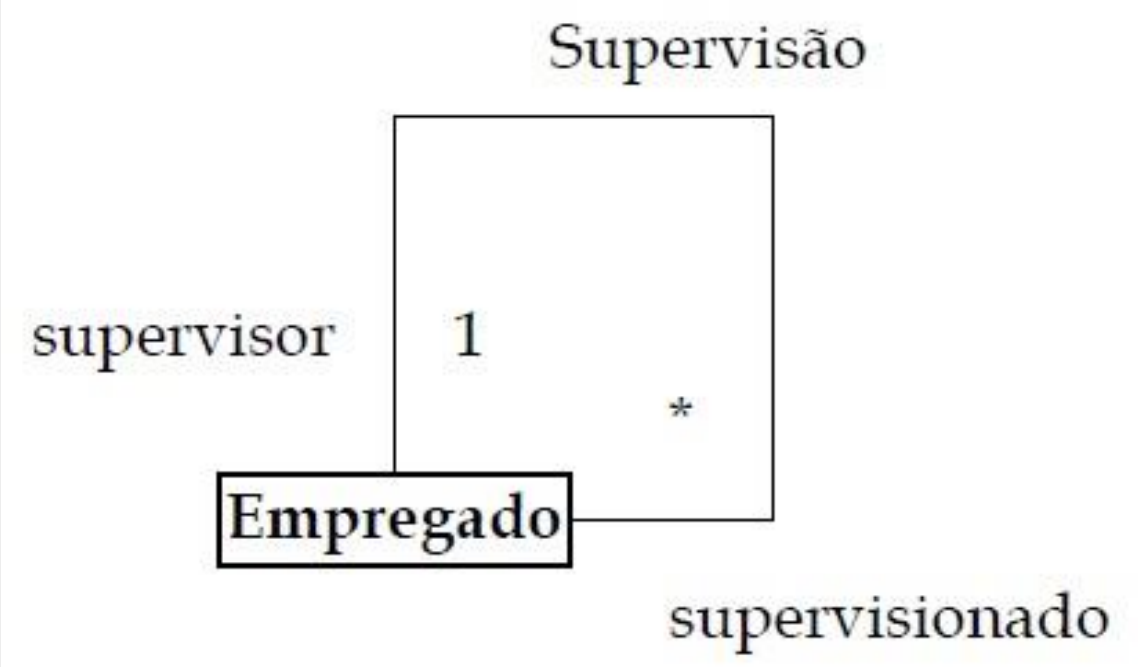


ASSOCIAÇÃO REFLEXIVA

- Associa objetos da mesma classe.
  - Cada objeto tem um papel distinto na associação.
- A utilização de papéis é bastante importante para evitar ambiguidades na leitura da associação.
- Uma associação reflexiva não indica que um objeto se associa com ele próprio.

ASSOCIAÇÃO REFLEXIVA

- **Exemplo:**



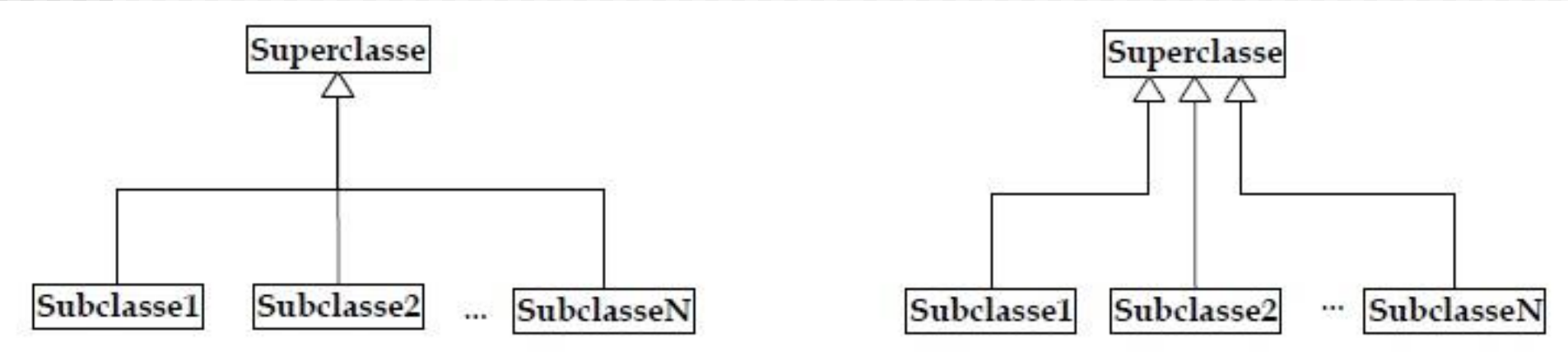
GENERALIZAÇÃO

- O **relacionamento de generalização** também costuma ser identificado pelos nomes **"herança"** ou **"especialização"**.

- **Terminologia:**
  - Superclasse x subclasse.
  - Classe base x classe herdeira.
  - Classe de generalização x classe de especialização.
  - Ancestral e descendente (generalização em vários níveis).

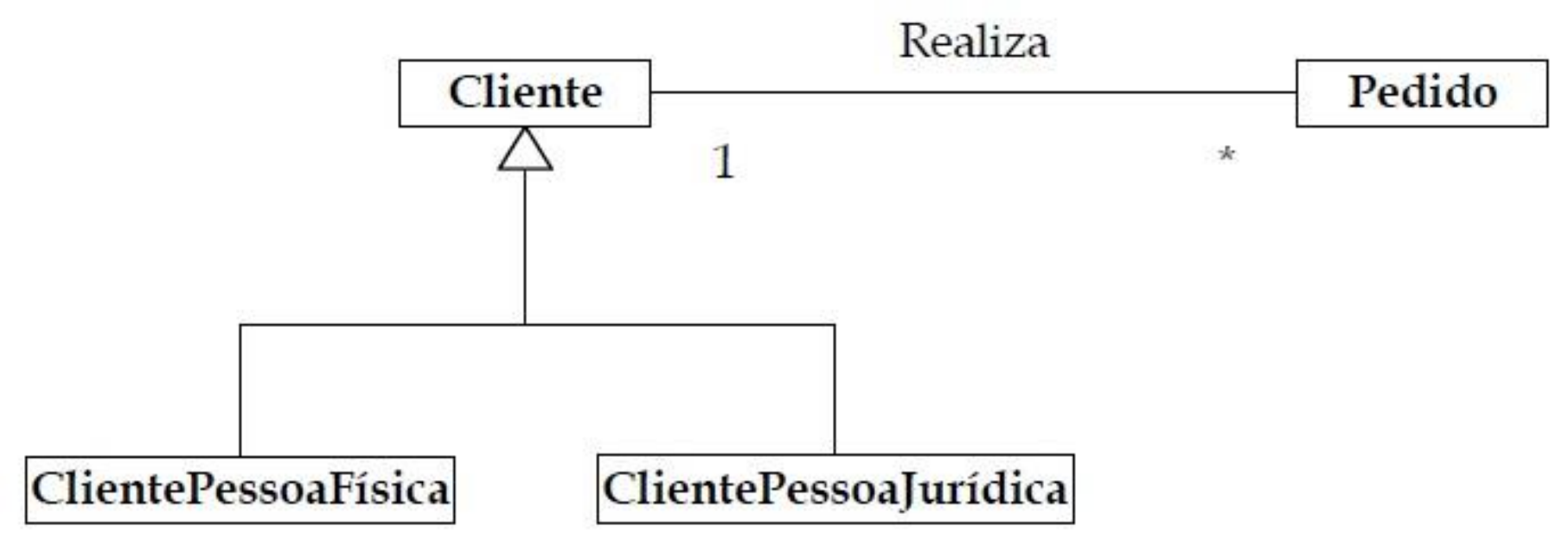
GENERALIZAÇÃO

- **Exemplos:**



HERANÇA DE ASSOCIAÇÕES

- Atributos, operações e **associações** são herdados pelas subclasses.

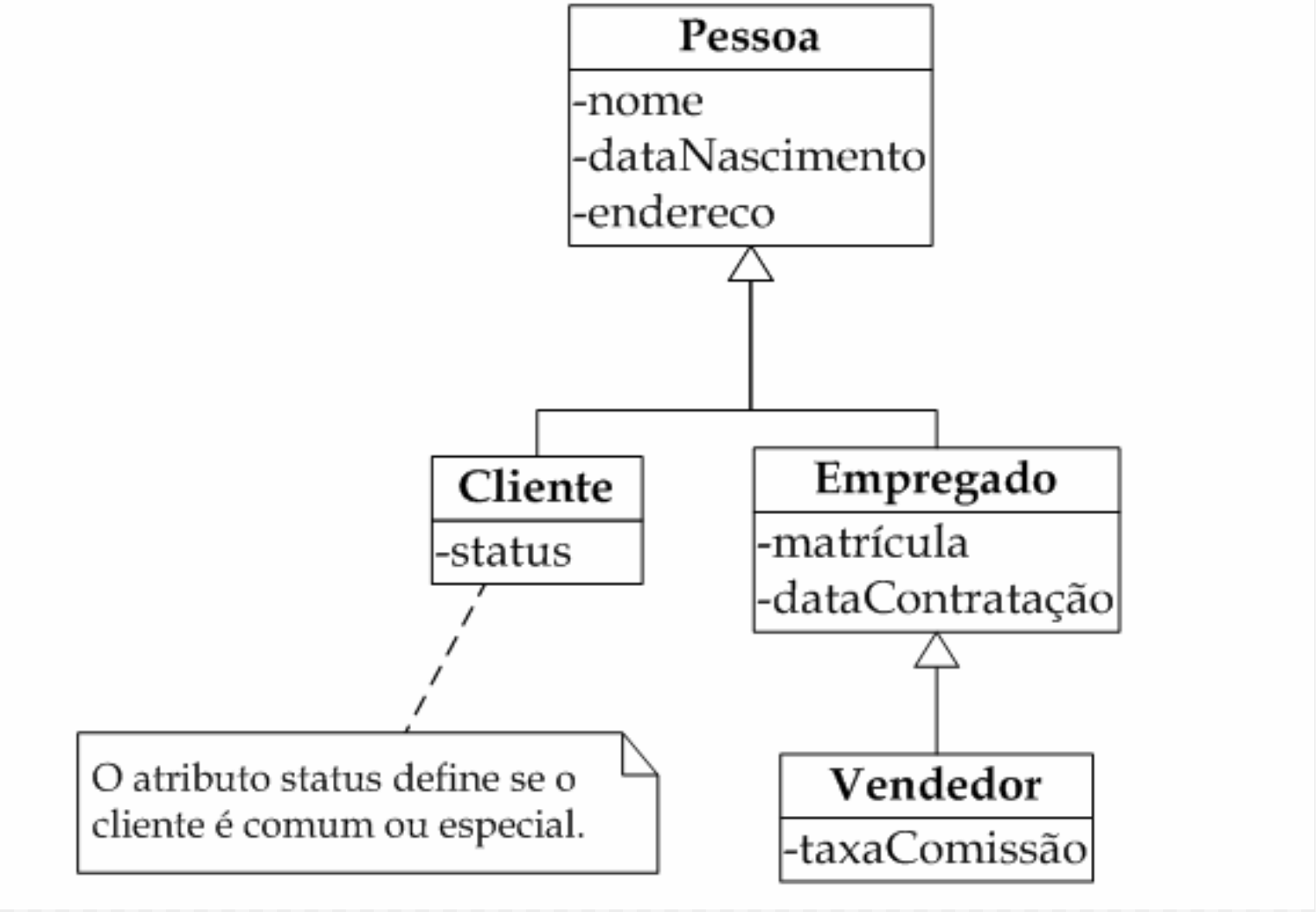


HIERARQUIAS DE GENERALIZAÇÃO

- A generalização pode ser aplicada em vários níveis (**hierarquia de generalização**).

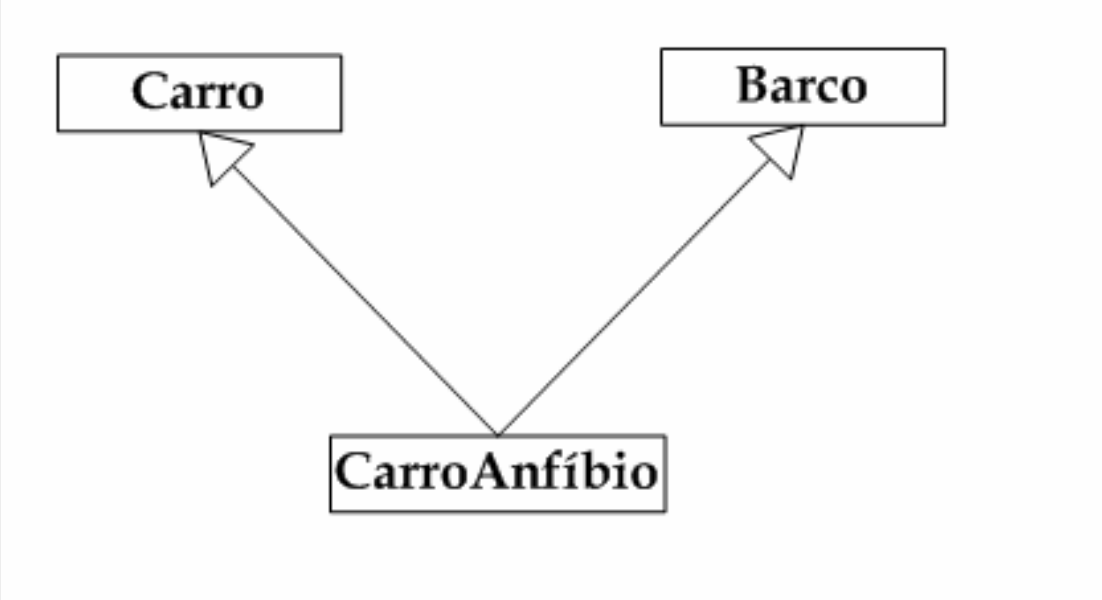
- **Característica importante:**
  - **Transitividade:** uma classe em uma hierarquia herda propriedades e relacionamentos de todos os seus ancestrais.

HIERARQUIAS DE GENERALIZAÇÃO



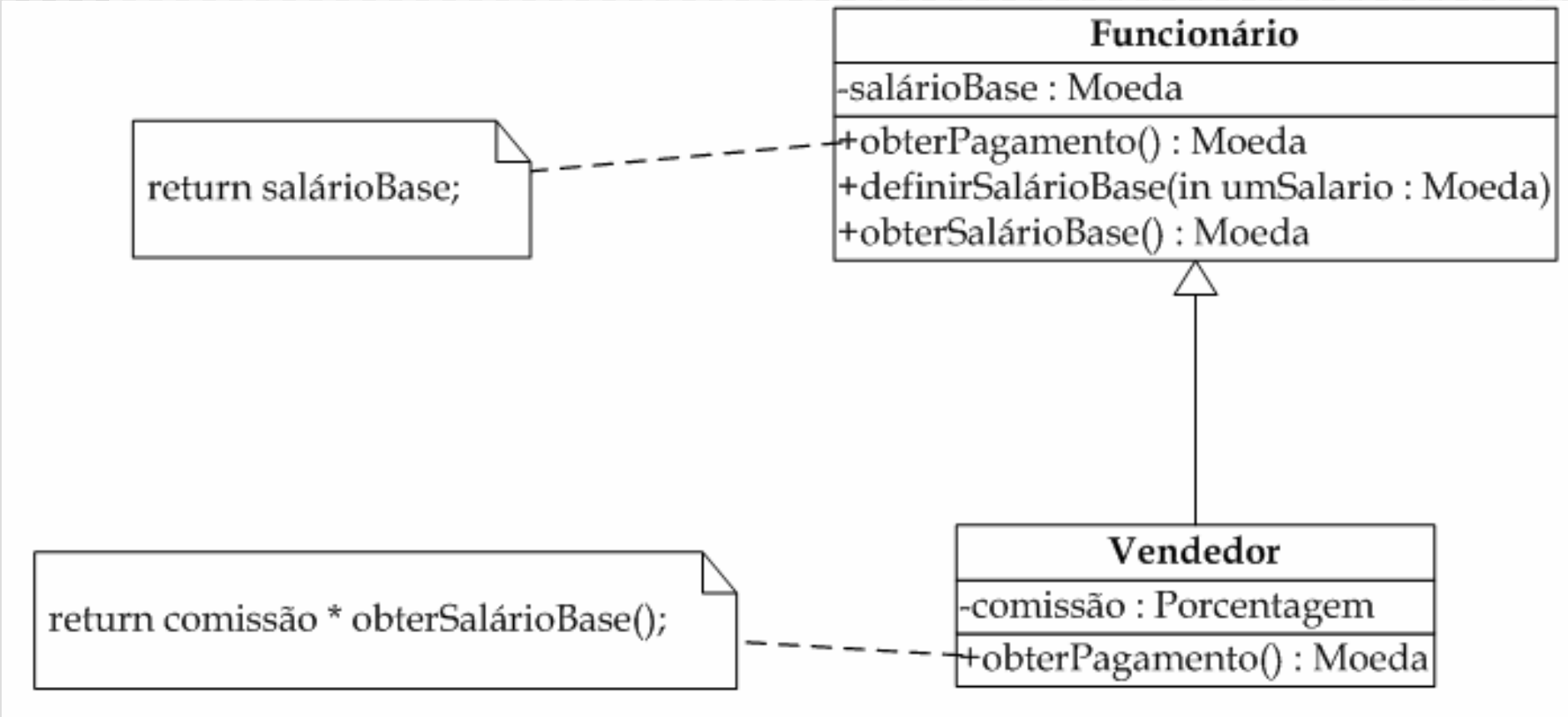
HERANÇA MÚLTIPLA

- Relacionamento que ocorre quando **uma classe tem mais de uma superclasse**.
  - Tal classe herda de todas as suas superclasses.



OPERAÇÕES POLIMÓRFICAS

- **Exemplo:**



CLASSES ABSTRATAS

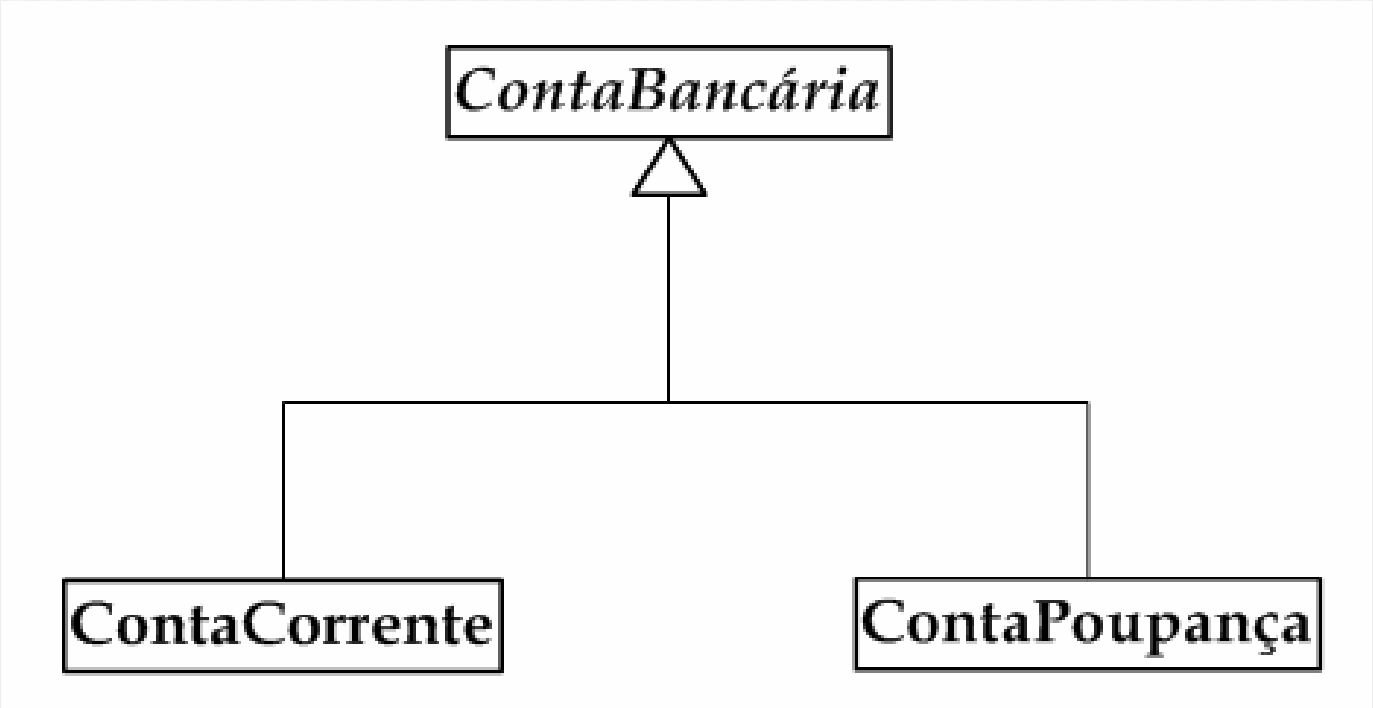
- Podem existir classes que não geram instâncias diretas: **classes abstratas**.

- Utilizadas para organizar e simplificar uma hierarquia de generalização: **propriedades comuns a diversas classes podem ser definidas em uma classe abstrata, a partir da qual as primeiras herdam.**

- **Subclasses** de uma classe abstrata também podem ser abstratas, mas a hierarquia deve terminar em uma ou mais classes concretas.

CLASSES ABSTRATAS

- Na UML, uma classe abstrata é representada com o seu nome em *itálico.*

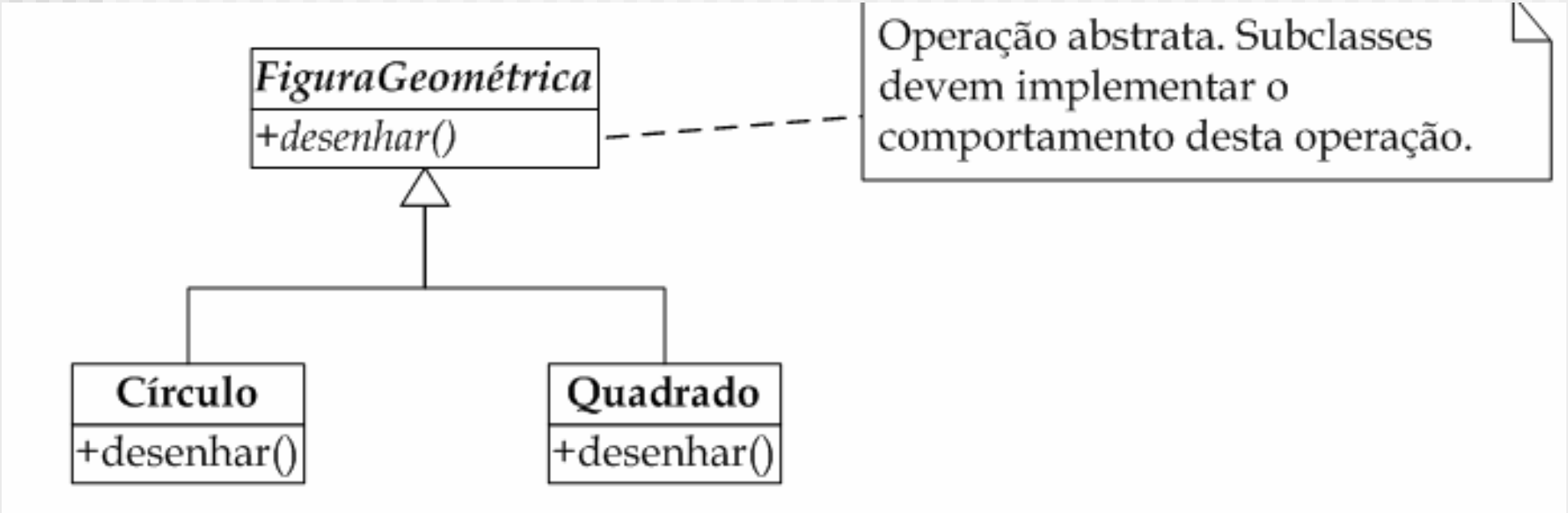


OPERAÇÕES ABSTRATAS E POLIMORFISMO

- **Em termos de operações**, uma **classe é abstrata** quando ela possui pelo menos uma **operação abstrata**.
- Uma operação abstrata **não possui implementação**.
- Uma classe pode possuir tanto operações abstratas quanto operações concretas.

OPERAÇÕES ABSTRATAS E POLIMORFISMO

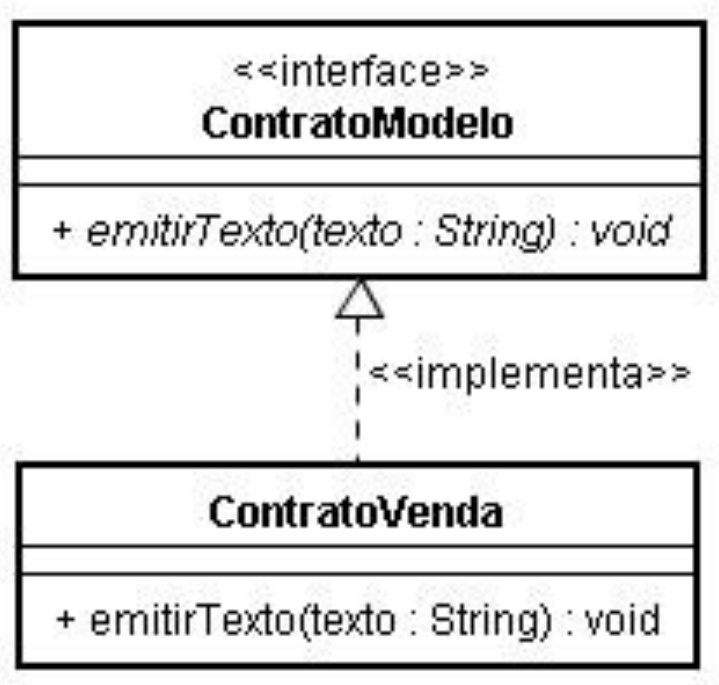
- **As classes Círculo e Quadrado são concretas**, pois fornecem implementação para a operação abstrata herdada.



REALIZAÇÃO

- Relacionamento usado para identificar **classes responsáveis por executar funções para classes que representam interfaces**.

- Herda o **comportamento** de uma classe, mas não sua estrutura.



ALGUMA DÚVIDA?

MOMENTO DE PRATICAR!

BIBLIOGRAFIA UTILIZADA

- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: Guia do Usuário. 2 ed. Rio de Janeiro: Campus, 2005.
- LARMAN, C. Utilizando UML e Padrões: uma introdução à Análise e ao Projeto Orientados a Objetos e ao Desenvolvimento Iterativo. 3 ed. Porto Alegre: Bookman, 2007.
- PRESSMAN, R. S.; MAXIM, B. R. Engenharia de Software: Uma Abordagem Profissional. 8 ed. Porto Alegre: AMGH Editora Ltda, 2016.
- SOMMERVILLE, I. Engenharia de Software. 10 ed. São Paulo: Pearson, 2019.