IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

ENGENHARIA DE SOFTWARE APLICADA
LICENCIATURA EM ENGENHARIA INFORMÁTICA
3º ANO

Diagrama de Classes

Agenda

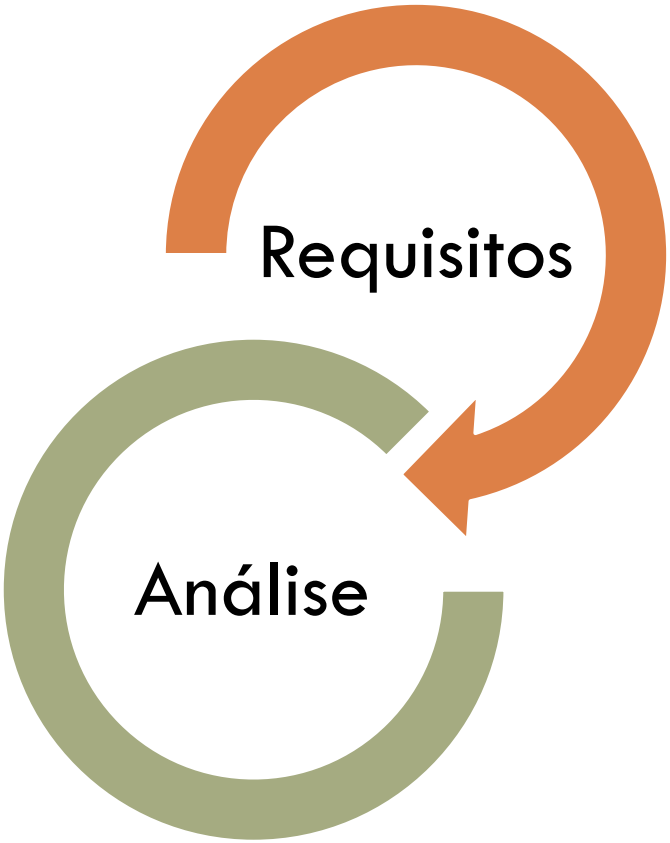
- Diagrama de classes em UML
- Relacionamentos
  - Associação
    - Agregação e composição
  - Generalização/Especialização

Atividades de um processo de desenvolvimento de SI

- ☑ Análise dos processos organizacionais

- ☑ Requisitos
- ☐ **Análise (Especificação funcional e não funcional do sistema)**
- ☐ Desenho (Arquitetura do software, modelos de conceção)
- ☐ Implementação (código)
- ☐ Validação, integração e instalação
- ☐ Manutenção

Enquadramento da Análise



- As fases de Análise e de Requisitos estão fortemente interligadas e muitas vezes parcialmente sobrepostas.
- Os termos utilizados de ambas as fases são do domínio do problema.
- É na fase de Análise que muitas vezes se descobrem falhas no levantamento de requisitos (omissão e distorção).

Objetivos da Análise

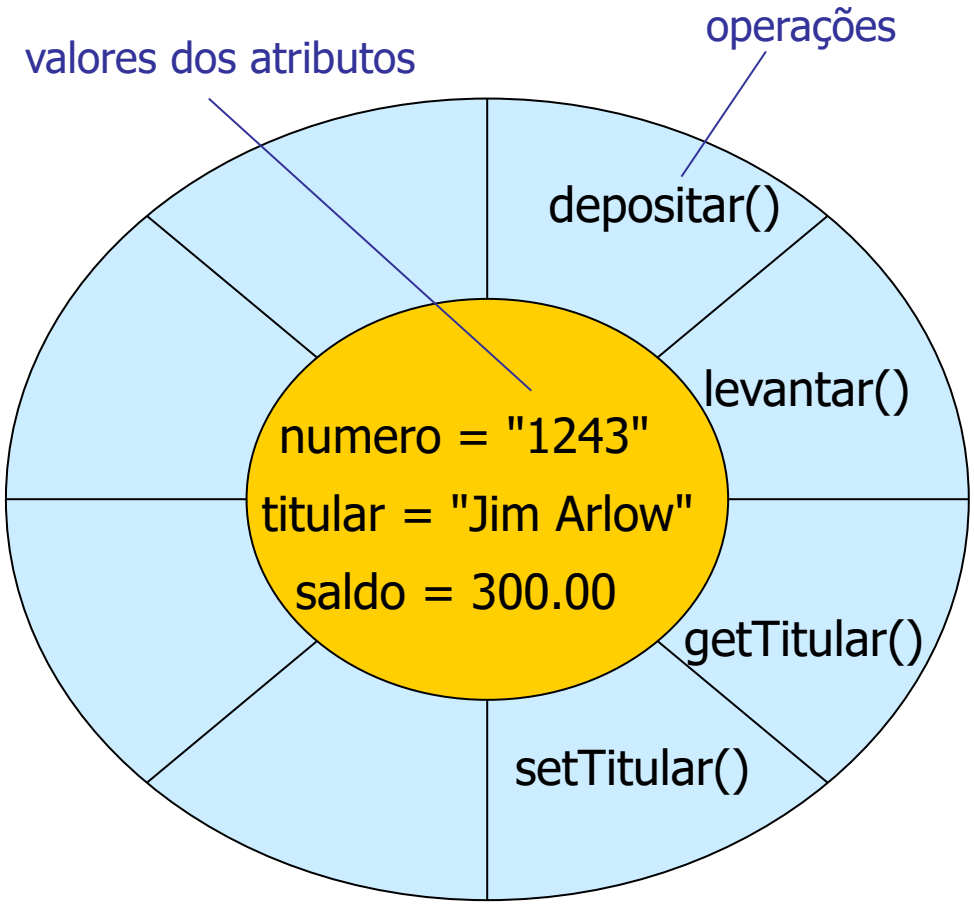
- Capturar uma visão geral do problema.
- Identificar e especificar as entidades informacionais (classes) do domínio do problema.
- Identificar como se relacionam as diversas entidades informacionais.
- Demonstrar que é possível realizar as funcionalidades especificadas na fase de levantamento de requisitos utilizando as entidades informacionais especificadas – *realização dos use cases*.

Entidades informacionais

- Numa abordagem orientada a objetos, as principais entidades informacionais são representadas através de classes.
- Os objetos representam instancias dessas mesmas classes e caracterizam-se por:
  - **Identidade** – Cada objeto pode ser acedido individualmente.
  - **Estado** – caracterizado pelos valores dos dados armazenados no objeto num determinado instante de tempo
  - **Comportamento** – conjunto de operações que o objeto pode efetuar.

Objetos: Encapsulamento de dados

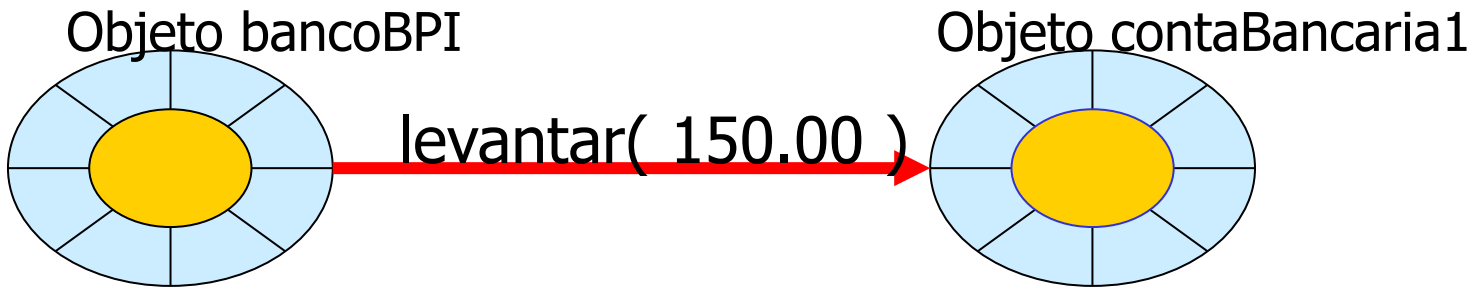
- Os dados são escondidos dentro do objeto – a única forma de aceder aos dados é através de uma operação.
- O encapsulamento permite criar software mais robusto e permite a reutilização de código.



Objeto contaBancaria1

Objeto: Comunicação entre objetos

- Nos sistemas Orientados a Objetos (OO) os objetos comunicam uns com os outros enviando mensagens.
- Estas mensagens fazem com que o objeto execute uma operação.
- Os objetos só conseguem comunicar se conhecerem o objeto destinatário.



O objeto bancoBPI envia uma mensagem levantar 150 euros ao objeto contaBancaria1.

O objeto contaBancaria1 responde à mensagem executando a operação levantar. A operacao decrementa o saldo do objeto contaBancaria1 em 150 euros.

Classes de análise

- As classes de análise representam abstrações claras do domínio do problema.
  - Estas classes são independentes da linguagem de programação a utilizar na implementação.
- As classes de análise têm:
  - Um conjunto de atributos de alto nível
  - Operações que são especificações de alto nível do conjunto de serviços fundamentais que a classe terá que disponibilizar
- As classes de análise devem ser mapeadas dos conceitos do negócio do mundo real.

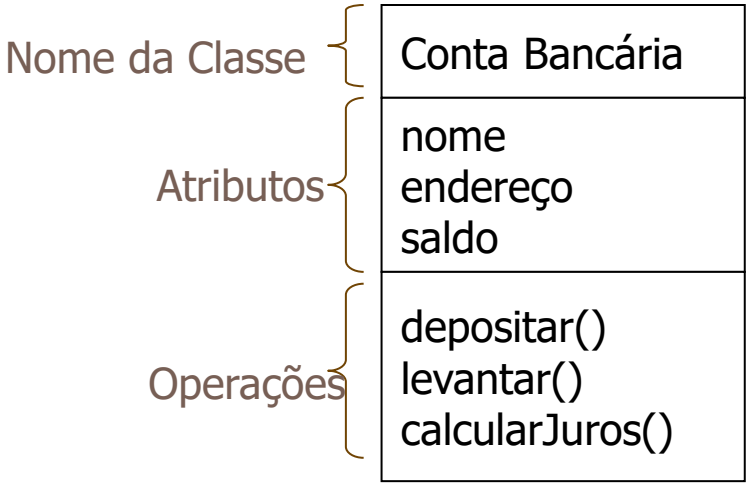
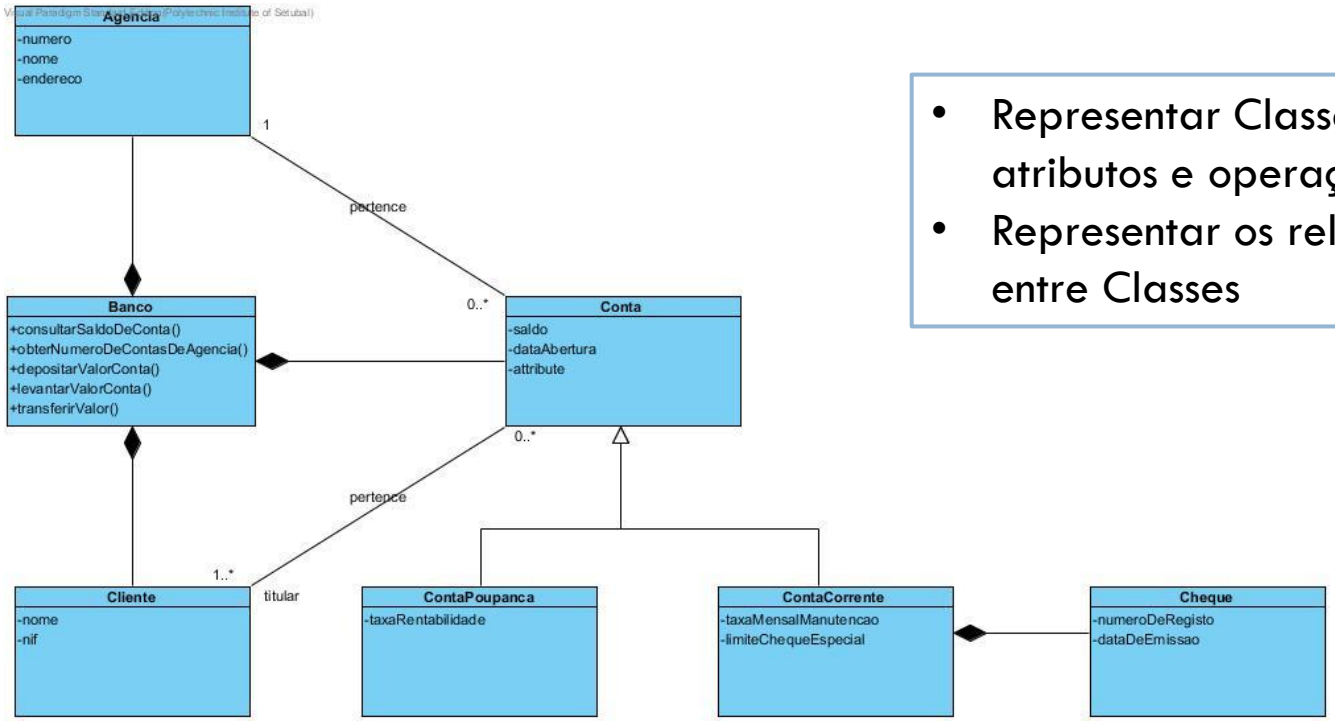


Diagrama de Classes em UML

- **Como especificar as classes e relacionamentos entre classes ?** Usar Diagrama de classes em UML.



- Representar Classes, com os atributos e operações
- Representar os relacionamentos entre Classes

O que é um relacionamento?

- Relacionamentos são ligações semânticas (com significado) entre elementos de modelação
- Já vimos alguns tipos de relacionamentos:
  - entre atores e Use Cases (associação)
  - entre Use Cases e Use Cases («include», «extend»)
  - entre atores e atores (generalização)

Relacionamentos entre Classes

Existem três tipos base de relacionamentos:

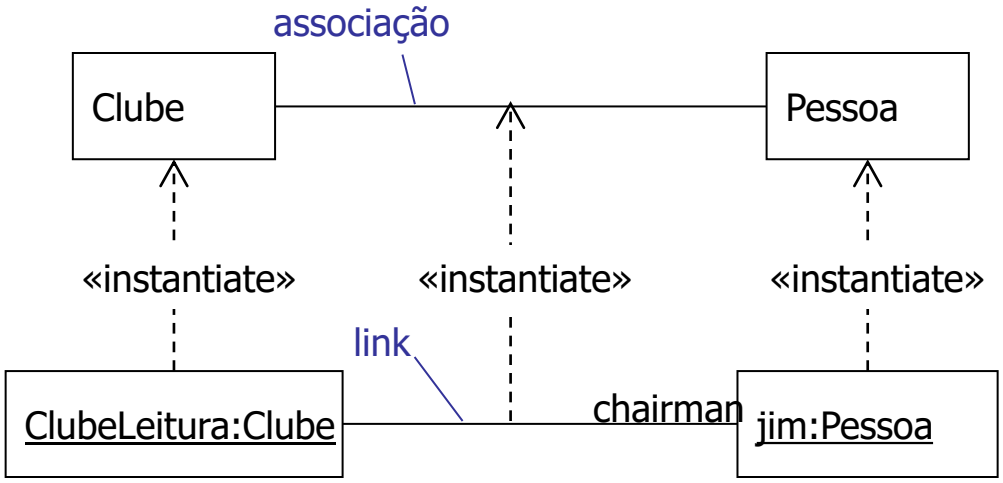
1. **Associação** – indica se existe algum tipo de ligação entre objetos das duas classes.

2. **Generalizacão/Especializacão**: relação entre classe mais geral e classe mais especifica.

3. **Dependência**: indica se uma classe depende de outra (a abordar na próxima aula)

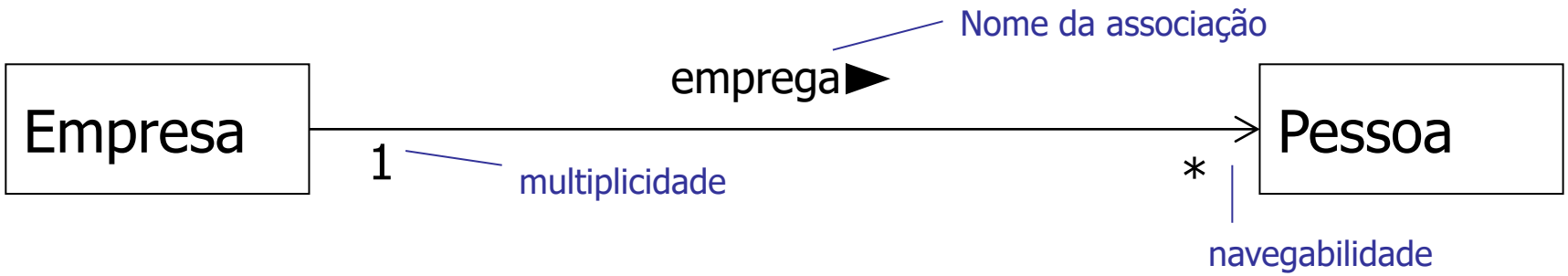
Relacionamento de Associação

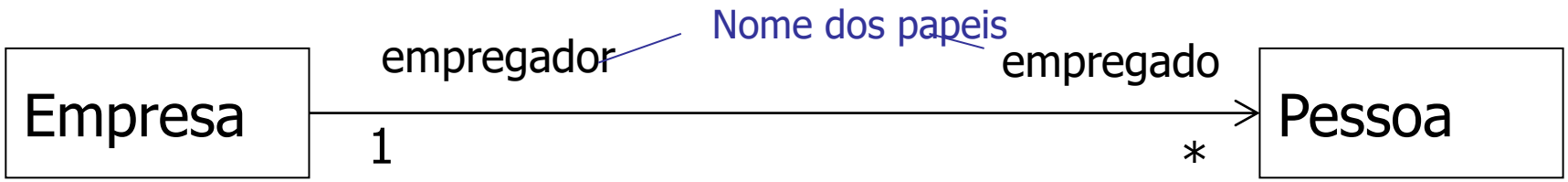
- Associações são ligações entre classes
- Os objetos são instâncias das classes e os links são instâncias das associações



Sintaxe das associações

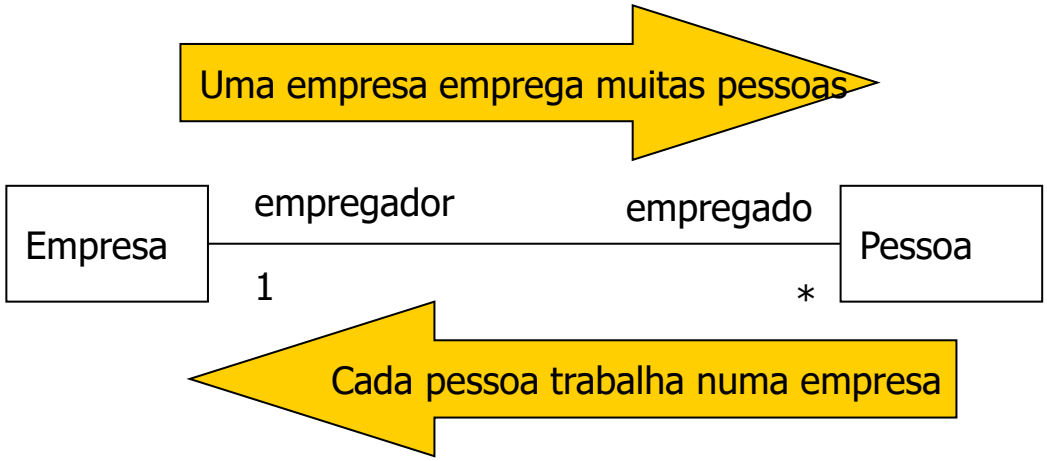
- As associações podem ter: nome da associação (verbos), papéis (substantivo), multiplicidade, navegabilidade





Multiplicidade

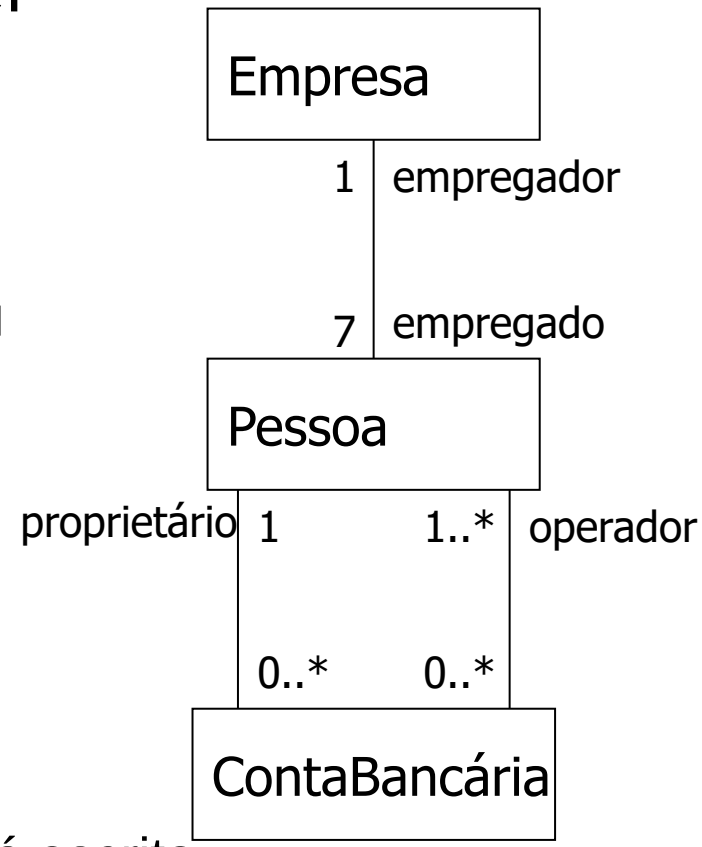
- A multiplicidade é uma restrição que especifica o número de objetos que podem participar num relacionamento num determinado instante de tempo
- Não existe multiplicidade por defeito – se não estiver expressa no modelo significa que ainda não está decidido o seu valor



multiplicity syntax: minimum..maximum	
0..1	zero or 1
1	exactly 1
0..*	zero or more
*	zero or more
1..*	1 or more
1..6	1 to 6

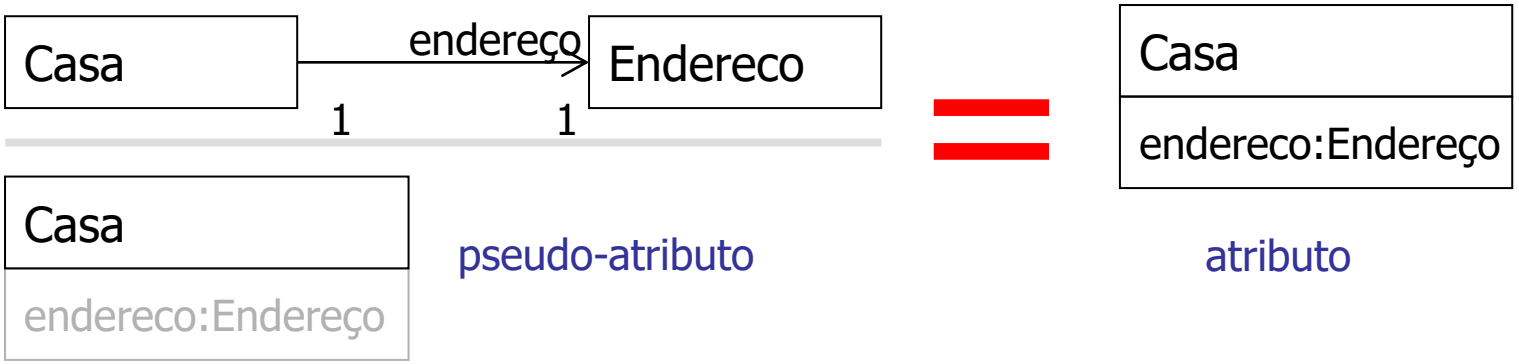
Exercício 1

1. Quantos empregados pode ter uma empresa?
2. Quantos empregadores poderá ter uma pessoa?
3. Quantos proprietários poderá ter uma conta bancária?
4. Quantas contas bancárias poderá ter uma pessoa?
5. Quantas contas bancárias pode uma pessoa operar



Nota: Ler o modelo exatamente como está escrito

Associações e atributos



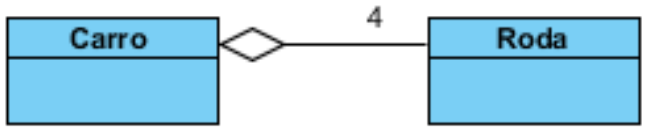
- Se um relacionamento navegável tiver um papel, é como se a classe fonte tivesse um pseudo-atributo cujo nome é o nome do papel e cujo tipo é a classe destino ("target")
- Os objetos da classe fonte referem-se aos objetos da classe "target" utilizando este pseudo-atributo.

Associações e atributos

- Utilizar associações quando:
  - A classe "Target" é uma parte importante do modelo
  - A classe "Target" é uma classe criada por si e que tem que ser mostrada no modelo
- Utilizar atributos quando:
  - A classe "Target" não é uma parte importante do modelo ex. tipo primitivo como number, string, etc.
  - A classe "Target" é apenas um detalhe de implementação tal como um ”library component" (ex: Java util vector) ou um componente adquirido.

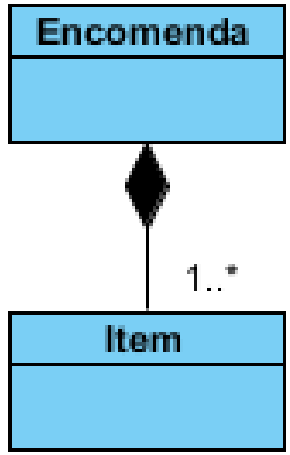
Agregação (has-a)

- Agregações são um tipo especial de associação no qual as duas classes participantes não possuem um nível igual, mas fazem um relacionamento “todo-parte”.
- Uma Agregação descreve como a classe que representa o todo, é composta (tem) de outra classe, que representa as partes.
- Para Agregações, a classe que age como o todo tem uma multiplicidade de um.



Composição (has-a)

- A composição é um tipo de agregação em que a relação do todo para as partes é mais forte do que na agregação simples.
  - **O tempo de vida do todo é igual ao tempo de vida das partes, no entanto é possivel em qualquer altura do ciclo de vida, adicionar partes ao todo, ou retirar partes ao todo.**
    - Ex. É possivel adicionar e eliminar items à encomenda.
  - **A destruição do todo implica na destruição das partes.**
    - Ex. Se encomenda for distruida (cancelada) todos os seus items o serão também.
  - **O todo é responsável pela criação das partes**
    - Ex. A encomenda é responsável pela criação dos vários items (linhas de ncomenda).
  - **Os objetos parte pertencem a um único todo**
    - *Ex: Um item de produto só pode pertencer a uma única encomenda.*



Exercício 2

- Das situações abaixo indique quais representam relacionamentos de composição e quais são de agregação.

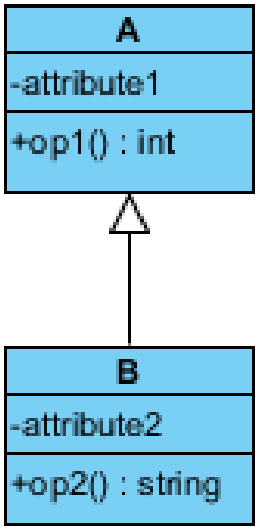
1. Banco e Conta Bancaria.
2. Aluno e Turma (tipo escola do 1° ciclo).
3. Carro e Motor
4. Conta Bancaria e Cartão de Débito.

Agregação vs Associação

- Usa-se uma associação entre duas classes quando estas estão conceptualmente no mesmo nivel.
  - Uma pessoa trabalha numa empresa.
- Usa-se uma agregação quando uma classe representa o todo e outra a parte desse todo.
  - O Carro tem rodas
  - A Equipa de Futebol tem Jogadores

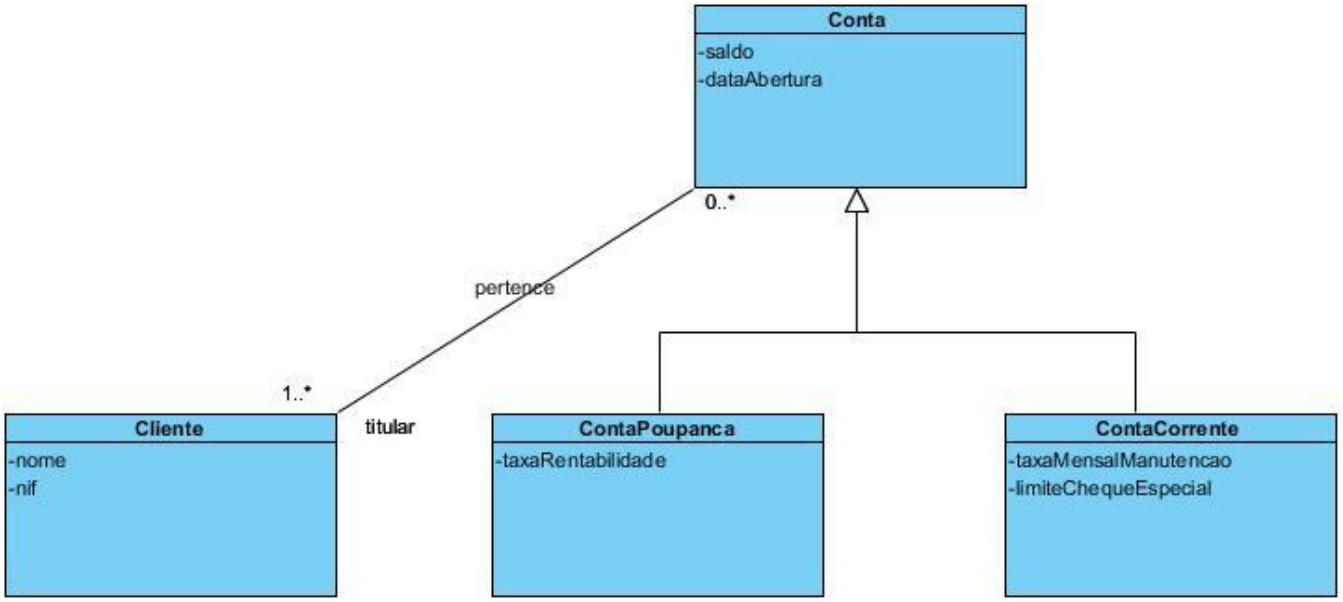
Generalização/Especialização (is-a)

- Herança é uma relação entre classes em que umas herdam o estado e o comportamento das outras.



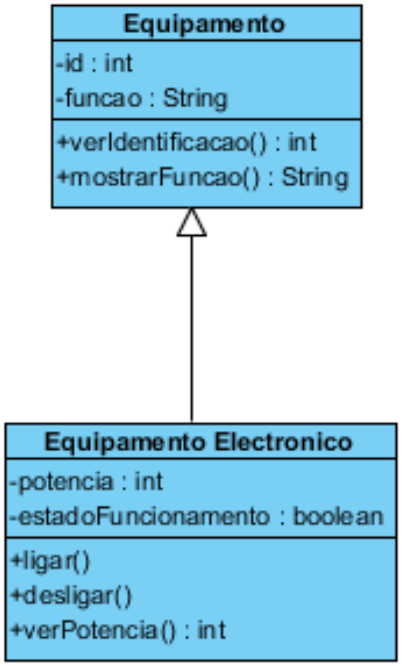
Generalização/Especialização (is-a)

- Uma conta bancaria é uma generalização de uma conta corrente e de uma conta poupança.
- A **conta** agrupa o que é comum ás classes conta poupança e conta corrente:
  - a estrutura de dados (atributos) e/ou
  - o comportamento (métodos)
  - relacionamentos



Generalização/Especialização (is-a)

- Um Equipamento Eletrónico é uma especialização de um Equipamento. O Equipamento Eletrónico é uma extensão, refinamento ou **especialização** desta, acrescentando-lhe:
  - mais estrutura de dados (atributos) e/ou
  - mais comportamento (métodos)



A reter

- Um Diagrama de classes indica os diversos **tipos** de objetos (classes) que existem no sistema e como estes se **relacionam** entre si!
- Existem distintos tipos de relacionamentos entre classes:
  1. **Associação**: indica se existe algum tipo de ligação entre objetos das duas classes.
     - Relacionamentos todo-parte :Agregação e Composição .
  2. **Generalizacão/Especializacão**: relação entre classe mais geral e classe mais especifica.

Exercício

- Usar classes para definir o glossário do sistema "Jogo de Futebol" descrito de seguida:

O jogo de futebol é realizado por duas equipas de jogadores. Cada equipa é composta por 11 jogadores, com diferentes funções:

Guarda-redes, Defesas, Médios, Atacantes e Pontas de Lança. O ponta de lança é um atacante especial por ter características especiais de goleador.

O jogo é realizado num campo com medidas regulamentares (em comprimento e largura), tem duas balizas, cada qual em extremos opostos do campo. Ganha o jogo a equipa que marcar mais golos (i.e., colocar a bola) na baliza do adversário. No jogo apenas existe um única bola, que apresenta características (peso, diâmetro, …) regulamentares... O jogo de futebol é mediado por uma equipa de 3 árbitros, em que um é o árbitro principal, e os outros dois árbitros auxiliares…

- Tendo em conta o sistema "Jogo de Futebol" descrito no exercício anterior e as classes identificadas, estabeleça agora as suas relações de forma a descrever o modelo de classes correspondente.