Análise e Programação Orientada a Objetos

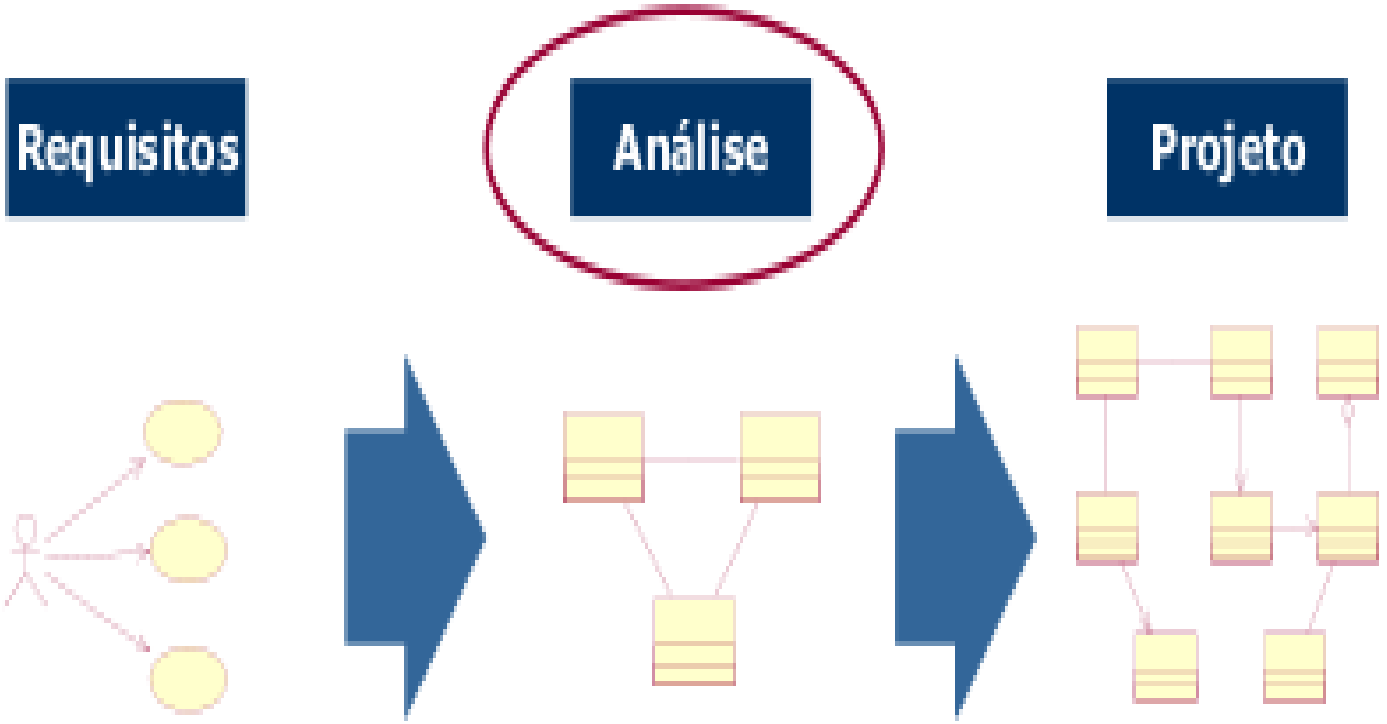
Sistemas de Informação

Unidade II – Parte III

Modelo de Classes de Domínio

Prof. Marciel de Liz Santos

Introdução



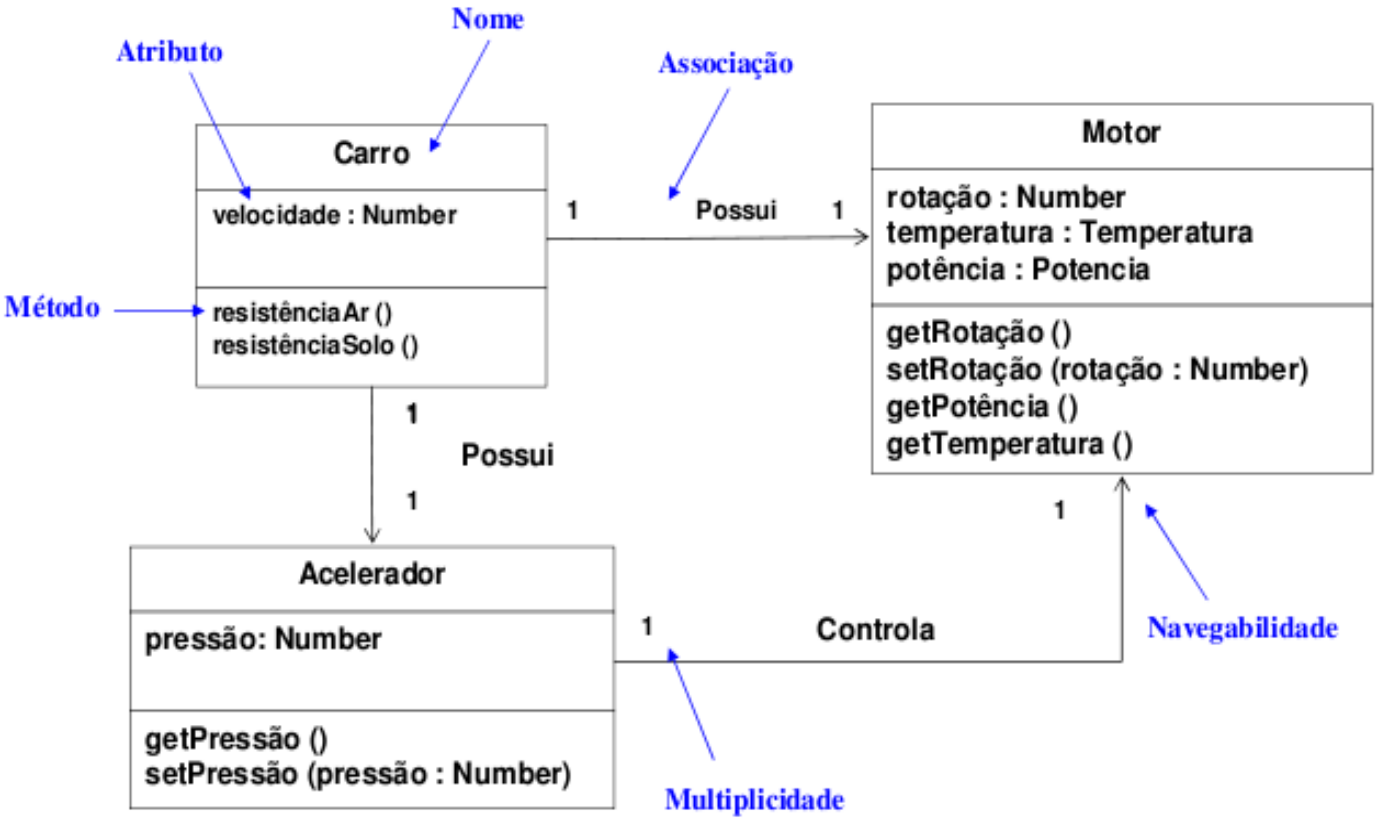
Introdução

- O modelo de **casos de uso** fornece uma perspectiva do sistema a partir de um ponto de vista **externo**
- Os desenvolvedores precisam prosseguir no desenvolvimento do sistema
- A funcionalidade externa de um sistema O.O é fornecida através **de colaborações entre objetos**
  - Externamente, os atores visualizam resultados de cálculos, relatórios produzidos, etc.
  - Internamente, os objetos colaboram uns com os outros para produzir os resultados
- Essa colaboração pode ser vista sob o **aspecto dinâmico** e sob o **aspecto estrutural estático**

Modelo de Classes

- O diagrama da UML utilizado para representar o aspecto estático é o diagrama de classes.
  - Exibe um conjunto de classes, interfaces e seus relacionamentos
  - As classes especificam tanto a estrutura como o comportamento dos objetos
- O modelo de classes é composto desse diagrama e da descrição textual associada

Diagrama de Classes



Modelo de Classes

- Na fase de análise, tendo em mãos o **diagrama de use-case**, podemos definir o diagrama de classes do sistema.
- O modelo de classes evolui durante o desenvolvimento do sistema.
  - À medida que o **sistema é desenvolvido**, o modelo de classes é **incrementado** com novos detalhes.
- Possui três **níveis sucessivos** de abstração:
  - Domínio
  - Especificação
  - Implementação

Modelo de Classes

- O **modelo de domínio (conceitual)** representa as classes no domínio do negócio em questão.
  - Não leva em consideração restrições inerentes à **tecnologia** a ser utilizada na **solução** de um problema.
  - Descreve o **problema** a ser desenvolvido, sem considerar características da **solução** a ser utilizada.
  - Modela **classes do domínio** do problema;
  - **Métodos e atributos** de acesso a banco de dados, **estrutura de mensagens** entre objetos, não aparecem no diagrama, apenas os **atributos básicos**

Modelo de Classes

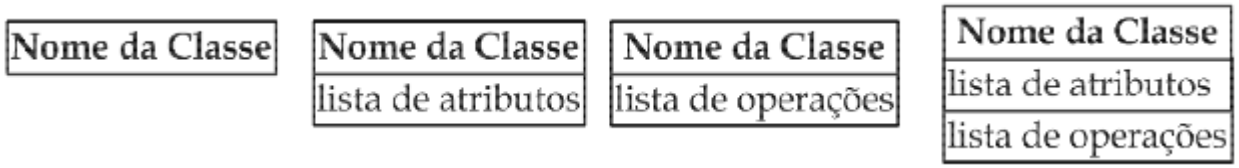
- **O modelo de classes de especificação**
  - Obtido através da adição de detalhes ao modelo anterior conforme a solução de software escolhida
  - Procura-se **tipos** sem pensar em implementação.
  - Pensa-se em **interfaces** e não na implementação.
    - Concentra-se no comportamento das coisas e não **o que** elas são;
- **O modelo de classes de implementação**
  - Corresponde à **implementação das classes** em alguma **linguagem de programação**.

Classes

- Uma classe representa um grupo de objetos semelhantes
- Uma classe descreve esses objetos através de **atributos** e **operações**
- Os **atributos** correspondem às informações que um objeto armazena
- As **operações** correspondem às ações que um objeto sabe realizar

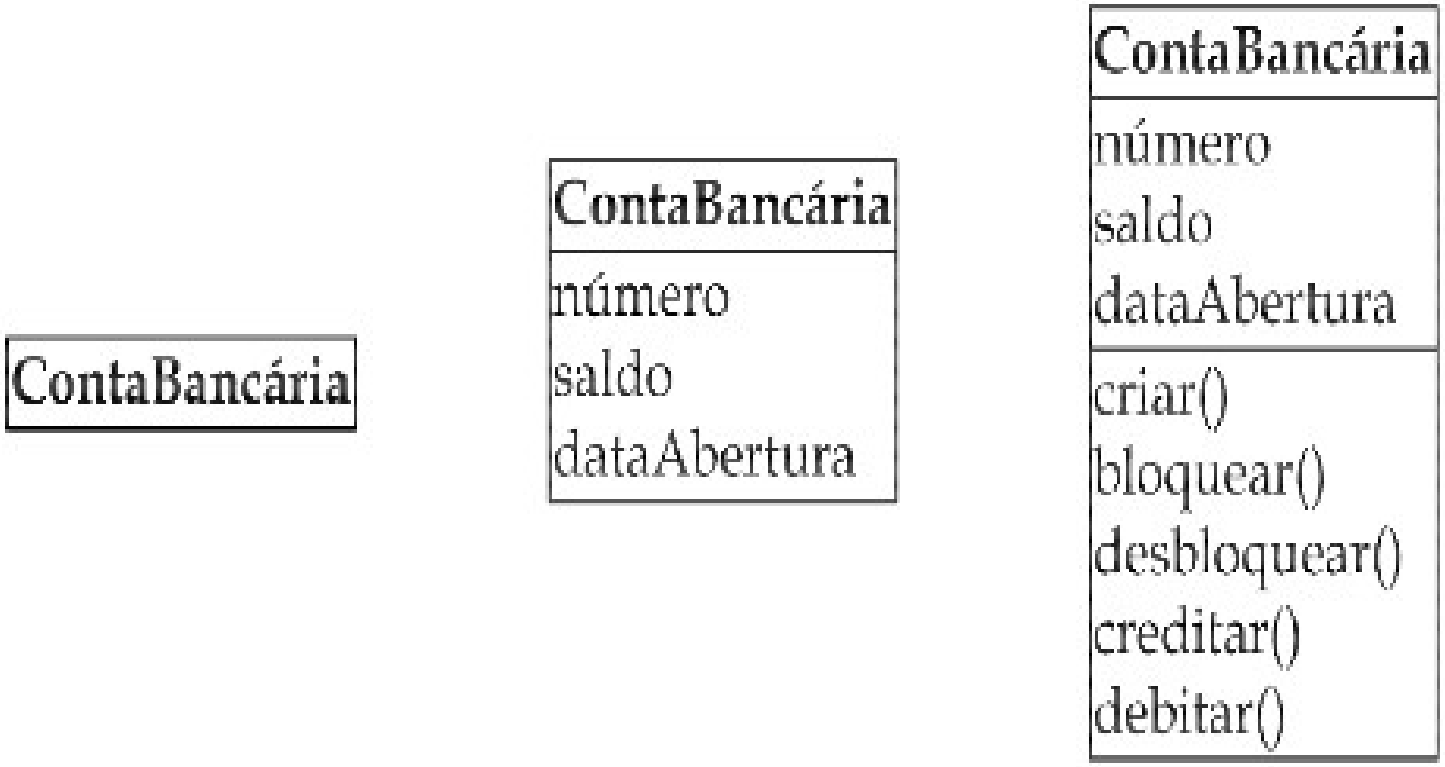
Classes

- Representada através de uma “caixa” com no máximo três compartimentos exibidos.
- Notação utilizada depende do nível de abstração desejado.



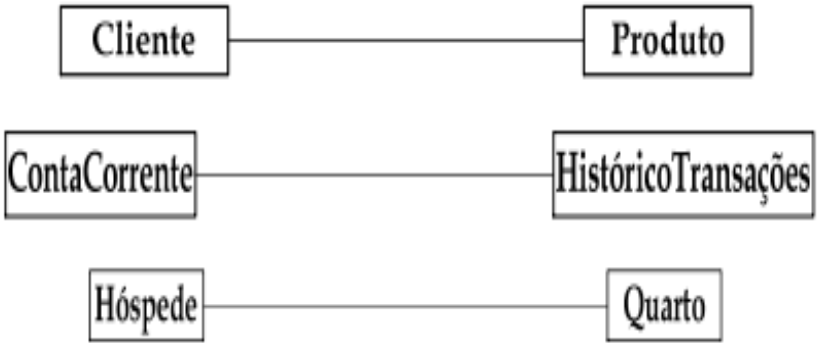
- Nomenclatura
  - Nome de classes e relacionamentos
    - Cliente, ItemPedido, ContaBancaria
  - Nome de atributos e operações
    - nome, dataNascimento, obterTotal
  - Nome de operações - verbo (creditar, debitar,sacar)

Exemplo (classe ContaBancária)



Associações

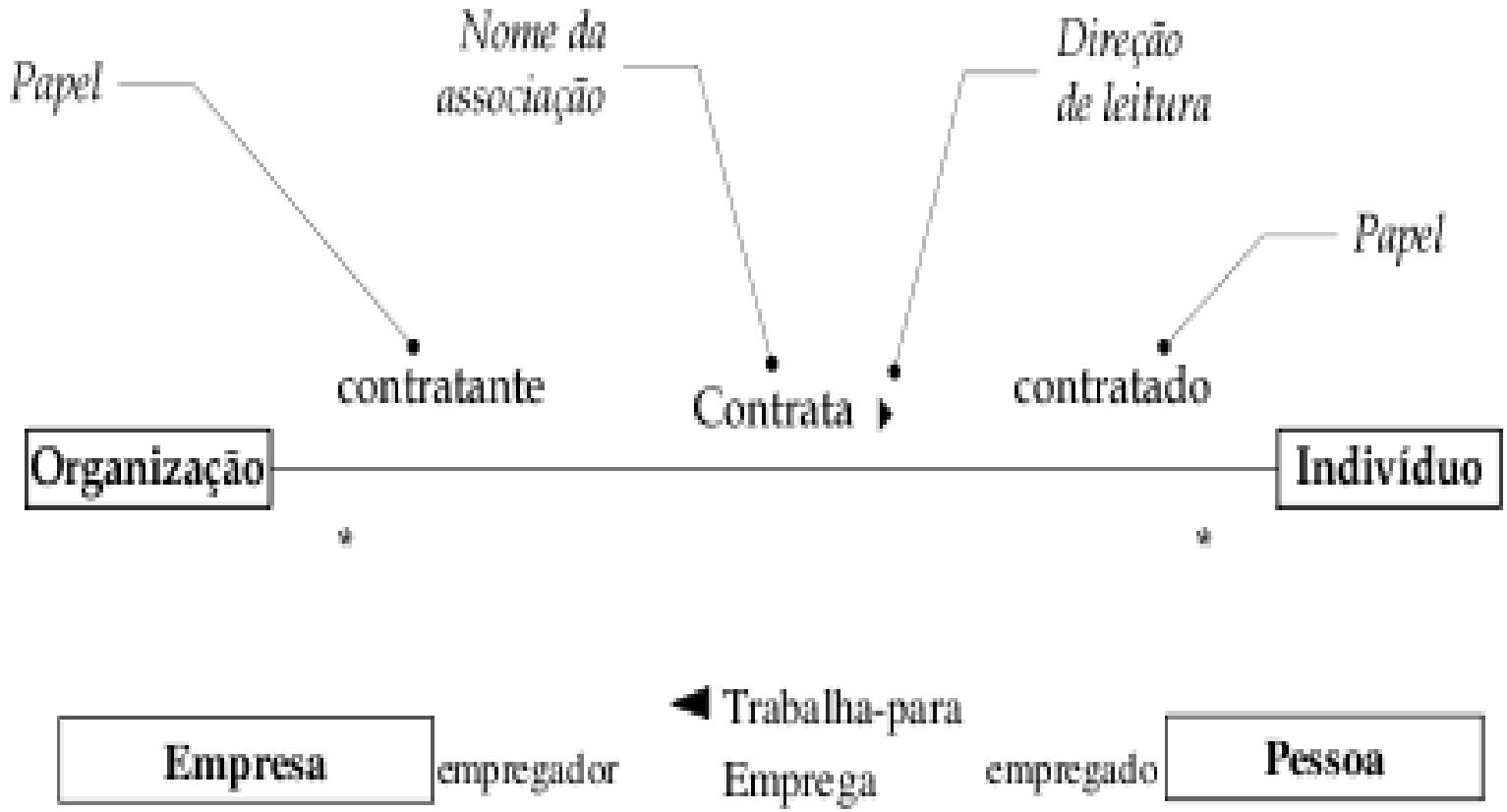
- Para representar o fato de que objetos podem se relacionar uns com os outros, utiliza-se a associação.
- Representa relacionamentos (ligações) que são formados entre objetos durante a execução do sistema, significando que elas "conhecem uma a outra”
- Representada através de um segmento de reta ligando as classes cujos objetos se relacionam.



Nome da associação, direção de leitura e papéis

- Para melhor esclarecer o significado de uma associação no diagrama de classes, a **UML** define três recursos de **notação**:
  - **Nome da associação**: fornece algum significado semântico a mesma (é um verbo ou uma frase verbal)
  - **Direção de leitura**: indica como a associação deve ser lida
  - **Papel**: para representar um papel específico em uma associação (rótulos na ponta da associação)
    - Interpretação da classe (objeto) para o qual ele está apontando

Nome da associação, direção de leitura e papéis (Exemplo)



Multiplicidade

- 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**.

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

Exemplo Multiplicidade

- Pode haver um cliente que esteja associado a **vários** pedidos.
- Pode haver um cliente que não esteja associado a pedido **algum**.
- Um pedido está associado a **um, e somente um, cliente**.

Exemplo Multiplicidade

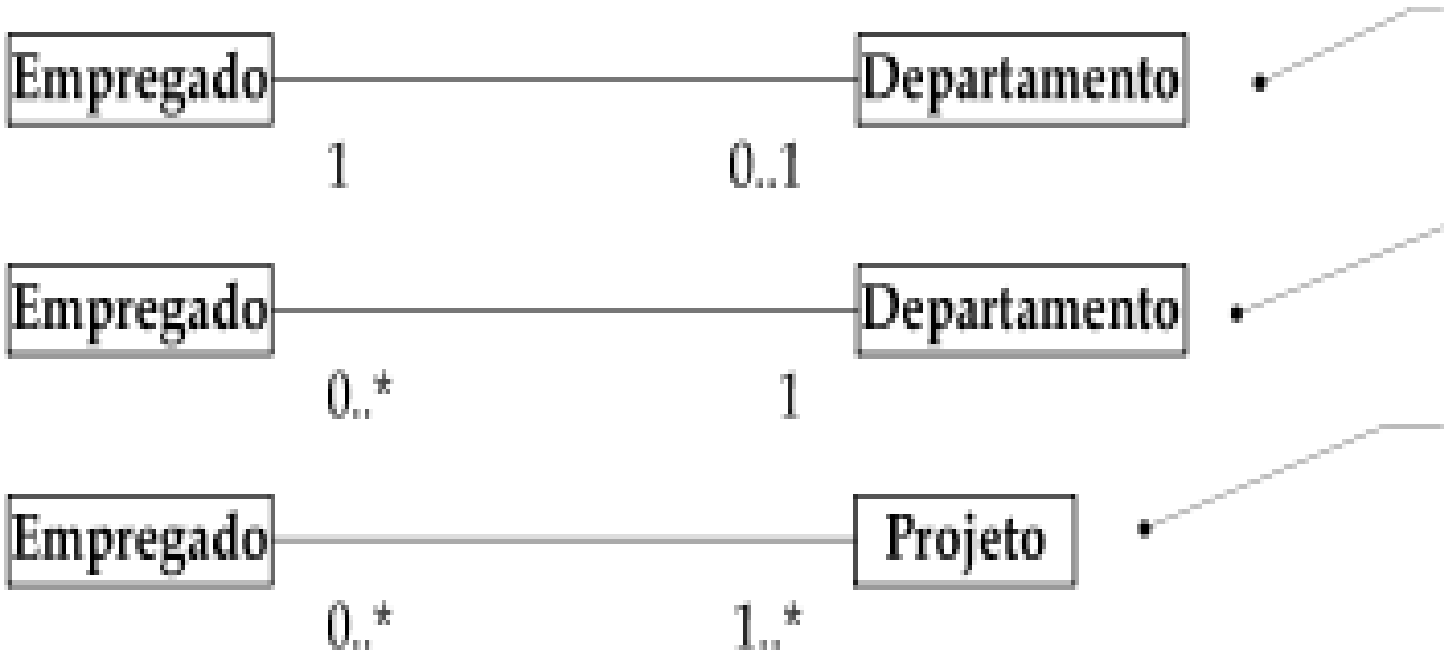
- Uma corrida está associada a, **no mínimo**, dois velocistas
- Uma corrida está associada a, **no máximo**, seis velocistas.
- Um velocista **pode** estar associado a várias corridas.

Conectividade

- A **conectividade** corresponde ao tipo de associação: "*muitos para muitos*", "*um para muitos*" e "*um para um*"
- Depende dos símbolos de multiplicidade que são utilizados na associação

Conectividade	Em um extremo	No outro extremo
Um para um	0..1 1	0..1 1
Um para muitos	0..1 1	* 1..* 0..*
Muitos para muitos	* 1..* 0..*	* 1..* 0..*

Conectividade (Exemplo)

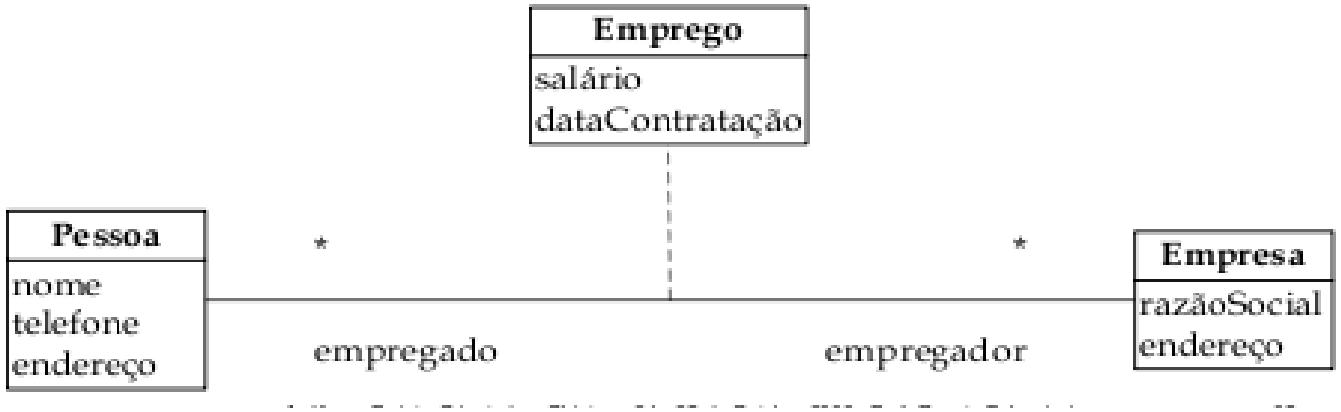


Participação

- Uma característica de uma associação que indica a necessidade (ou não) da existência desta associação entre objetos.
- A participação pode ser **obrigatória** ou **opcional**.
  - Se o valor mínimo da multiplicidade de uma associação é igual a 1, significa que a participação é obrigatória
  - Caso contrário, a participação é opcional.

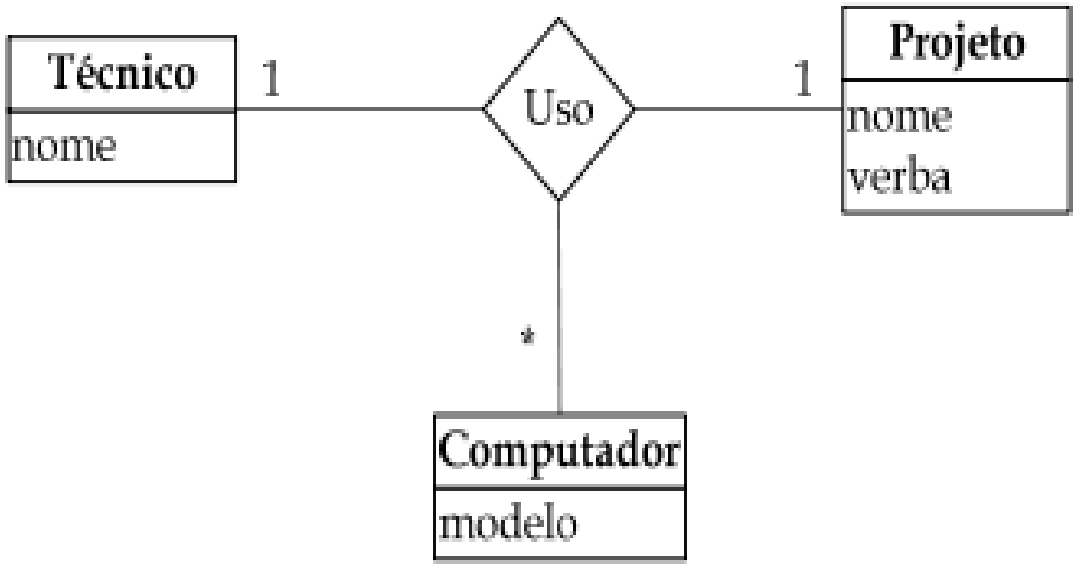
Classe Associativa

- Classe que está ligada a uma associação, ao invés de estar ligada a outras classes.
- É necessária quando duas ou mais classes estão associadas, e é necessário manter informações sobre esta associação.
- Pode estar ligada a associações de qualquer tipo de conectividade.



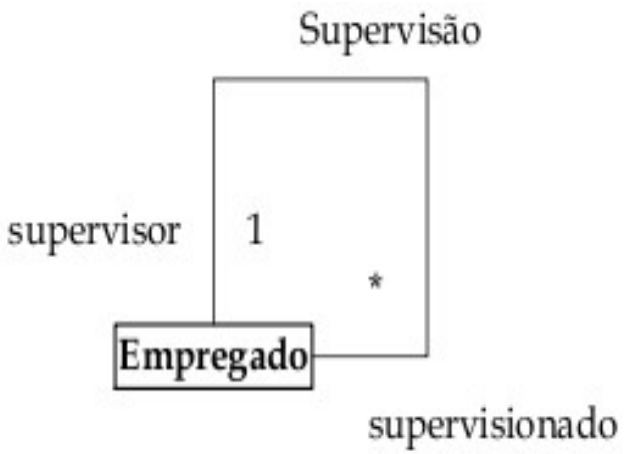
Associações n-árias

- Representar a associação existente entre objetos de n classes.
- Uma **associação ternária** (n = 3).
- Na notação da UML, as linhas de associação se interceptam em um losango.



Associações Reflexivas

- Associa objetos da mesma classe.
  - Cada objeto tem um papel distinto na associação.
- Uma associação reflexiva não indica que um objeto se associa com ele próprio.
  - Ao contrário, indica que objetos de uma mesma classe se associam

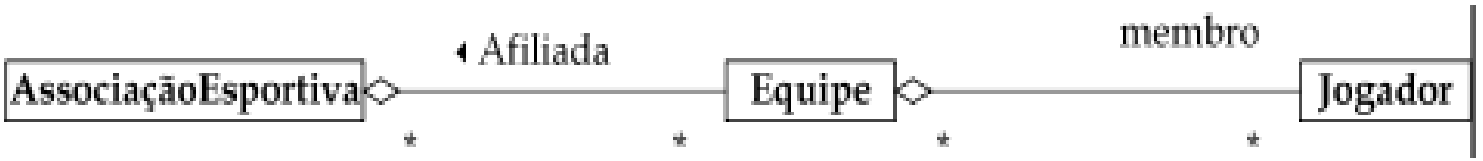


Agregação

- É um caso especial da associação
  - multiplicidades, participações, papéis, etc. podem ser usados igualmente
- 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.
- São assimétricas: se um objeto **A é parte de um objeto B**, B não pode ser parte de A

Agregação

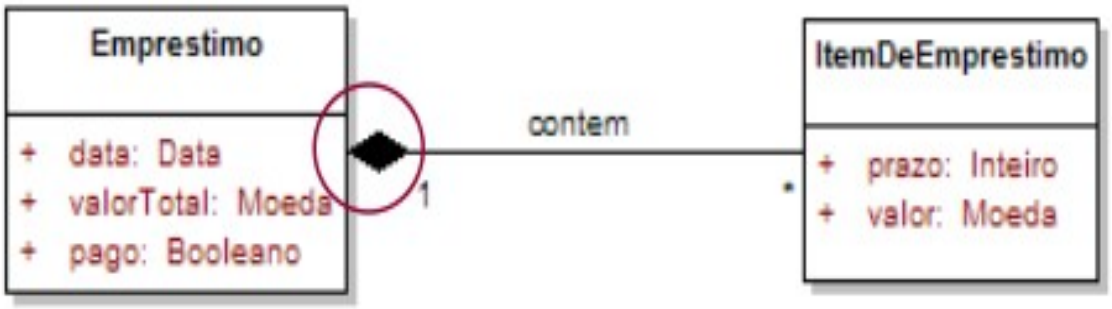
- Representada como uma linha conectando as classes relacionadas, com **losango** branco perto da classe que representa o todo.



- Como saber se é uma agregação ou associação???
- Sejam duas classes associadas, X e Y. Se uma das perguntas a seguir for respondida com um sim, provavelmente há uma agregação onde X é todo e Y é parte.
  - X tem um ou mais Y?
  - Y é parte de X?

Composição

Agregações fortes – se existir **exclusividade** nessa associação, ou seja, se um item não puder fazer parte de nenhum outro conceito



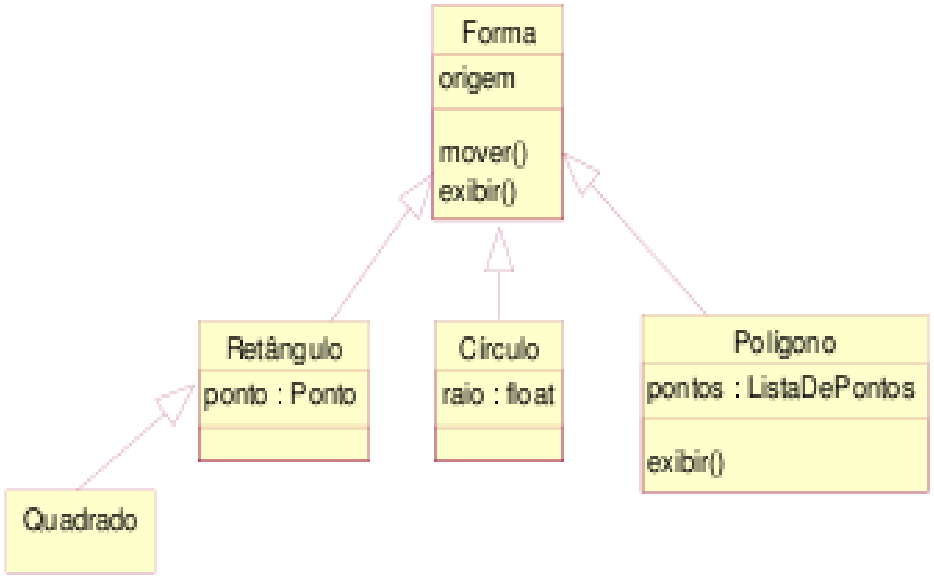
Quando a exclusividade não é exigida ...

Agregação X Composição

- Como saber se é agregação ou composição???
  - As diferenças não são bem definidas
- Na agregação
  - A destruição do **objeto todo** **NÃO** implica necessariamente a destruição do **objeto parte**.
  - Ex: se uma equipe de futebol for extinta, este jogador ainda poderá continuar jogando em outras equipes.
- Na composição
  - Os **objetos parte** necessitam do **objeto todo** para terem razão de existir
  - Ex: item pedido / pedido; rodas e motor / carro

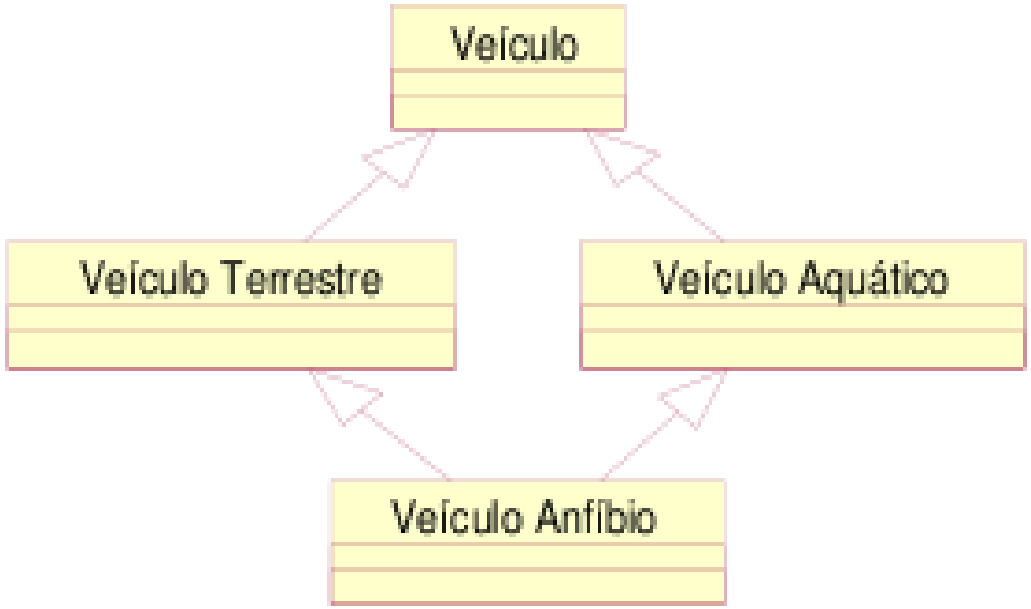
Generalização

- Relacionamento entre um elemento mais geral (chamado de superclasse ou pai) e um mais específico (chamado de subclasse ou filho)



Herança Múltipla

- Quando uma classe tem múltiplas superclasses



Notação para Relacionamentos