MODELAGEM DE CLASSES

MODELO DE CLASSES DE ANÁLISE

- Como já vimos, a **modelagem de casos de uso** permite ao analista modelar a **visão externa do sistema** .
- Entretanto, além dessa **visão externa**, ***precisamos definir uma visão interna (através de colaborações entre objetos)*** do SSOO, sem considerar características da solução a ser utilizada.
- A definição dessa visão é feita através da **modelagem de classes**.
- O modelo de classes representa a visão **estrutural e estática** dos objetos que compõem um SSOO.

MODELO DE CLASSES DE ANÁLISE

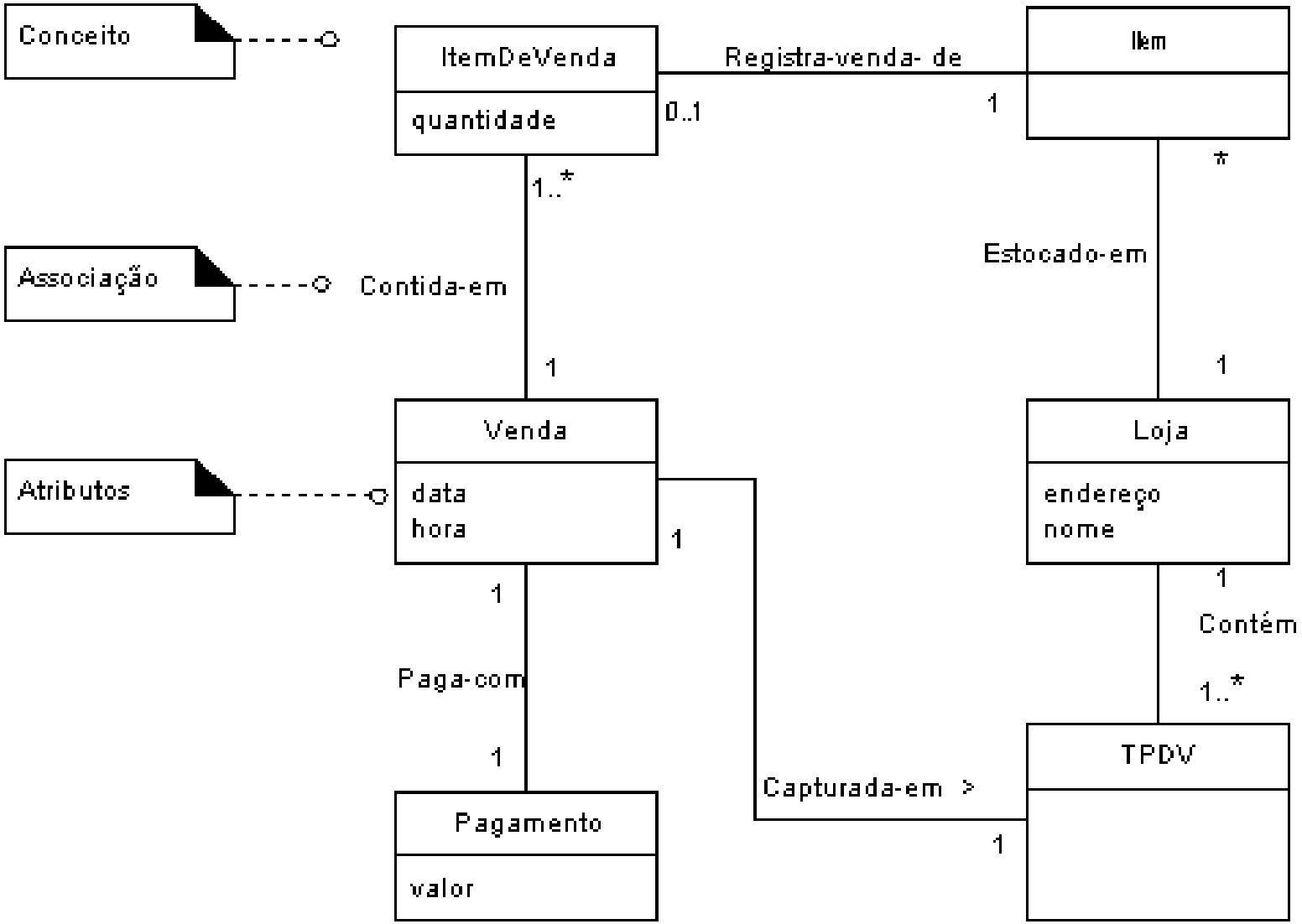
- O modelagem de classes é realizada na fase de análise gera o **modelo de classes de análise** (MCA).
  - Também chamado de **modelo de classes do domínio**.
- O MCA provê respostas para as seguintes perguntas:
  - Quais as classes que representam **entidades do domínio do problema**?
  - Quais são as responsabilidades de cada uma dessas classes e de que forma elas estão relacionadas?

MODELO DE CLASSES DE ANÁLISE

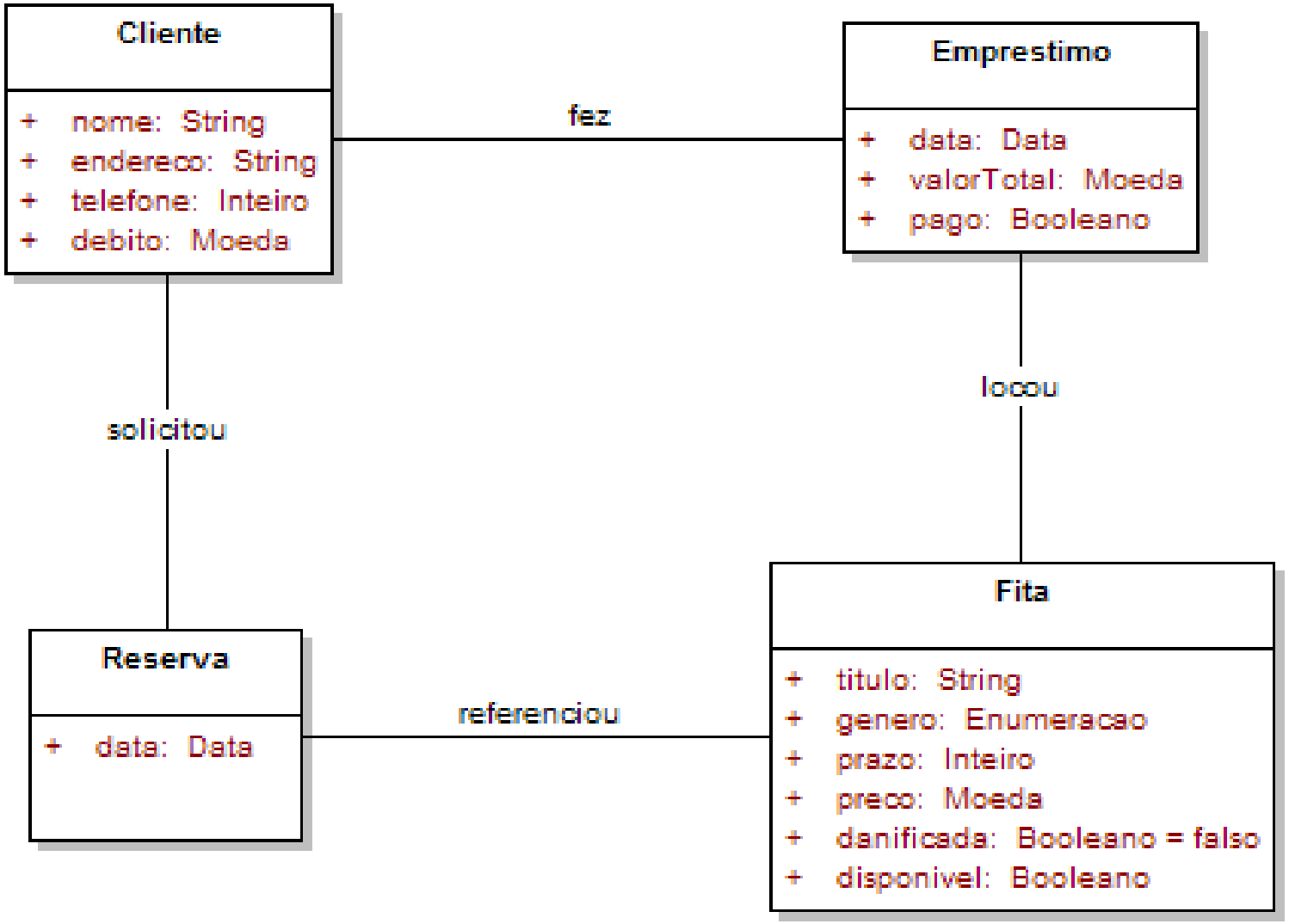
O MCA apresenta dois tipos de classe: **conceituais** e **da aplicação**.

- Classes conceituais
  - Correspondem aos conceitos do domínio do problema.
  - Modelam conceitos relevantes do mundo real.
  - Nomenclatura alternativa: classes conceituais, do domínio, do negócio, etc.
- Classes da aplicação
  - Classes necessárias em uma aplicação específica.
  - Modelam conceitos do mundo virtual.

EXEMPLO CLASSES - CONCEITUAIS



EXEMPLO CLASSES - CONCEITUAIS



NOTAÇÃO UML PARA O MODELO DE CLASSES NA ANÁLISE

MODELO DE CLASSES

- O diagrama da UML utilizado para representar o aspecto estático é o ***diagrama de classes***.
- O ***modelo de classes*** é composto desse diagrama e da descrição textual associada.

DIAGRAMA DE CLASSES

- Esse diagrama mostra a estrutura estática do modelo, em que os elementos representados por classes, com sua estrutura interna e seus relacionamentos.
- Os **diagramas de classe também podem ser organizados em pacotes**, mostrando somente o que relevante em um pacote específico.
- É recomendando a utilização desse diagrama durante a **fase de análise** para a produção de um **modelo conceitual** (neste modelo a preocupação é representar a classes ou atributos e as suas associações).

MODELO DE CLASSES

- 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.
- Três *níveis sucessivos de abstração:*
  - *Domínio*
  - *Especificação*
  - *Implementação*.

MODELO DE CLASSES

- **O *modelo de classes de domínio*** 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.
- **O *modelo de classes de especificação*** é obtido através da adição de detalhes ao modelo anterior conforme a solução de software escolhido.
- **O *modelo de classes de implementação*** corresponde à implementação das classes em alguma linguagem de programação.

CONCEITOS DO MCA

Conceitos normalmente usados durante a construção do modelo de classes de análise (MCA):

- Classes e atributos
- Relacionamentos
  - ✓ Associações (com seus variados adornos)
  - ✓ Composições e agregações
  - ✓ Generalizações
- Classes abstratas
- Associações reflexivas
- Classes de associação

NOTAÇÃO PARA CLASSES NA UML

NOTAÇÃO PARA CLASSES

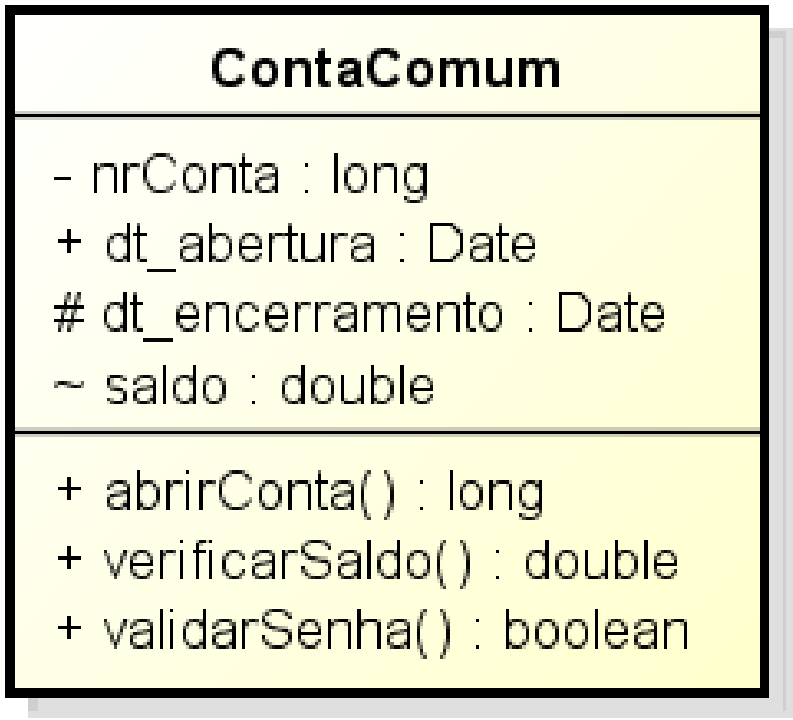
Representada através de uma “caixa” com no máximo três compartimentos exibidos, dependendo do nível de abstração desejado na modelagem.

NOME DA CLASSE	NOME DA CLASSE	NOME DA CLASSE	NOME DA CLASSE
	LISTA DE ATRIBUTOS	LISTA DE OPERAÇÕES	LISTA DE ATRIBUTOS
			LISTA DE OPERAÇÕES
ContaBancaria	ContaBancaria	ContaBancaria	ContaBancaria
	numero saldo dataAbertura	bloquear() creditar() debitar()	- numero: String - saldo: float - dataAbertura: Date
			+ bloquear(): void + cred(float n): void + deb(float n):void

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



Os símbolos de menos (- private), mais (+ public), sustenido (# protected) e til (~ package (default)) na frente dos atributos e métodos representam a visibilidade dos mesmos, o que determina quais objetos de quais classes podem utilizar o atributo ou o método em questão.

VISIBILIDADE

- É utilizada para identificar o nível de acessibilidade de um atributo ou método do sistema, existem quatro modos de visibilidade: público, protegido, privado e pacote
- **Privado (-):** significa que os objetos da classe detentora do atributo ou método poderão enxergá-lo ou utilizá-lo.
- **Protegido (#):** determina que além dos objetos da classe detentora do atributo ou método também os objetos de suas subclasses poderão ter acesso ao mesmo.
- **Pública (+):** determina que o atributo ou método pode ser utilizado por qualquer objeto.
- **Pacote (~):** determina que o atributo ou método é visível por qualquer objeto dentro do pacote.

ASSOCIAÇÕES

Uma associação representa o fato de que objetos podem se relacionar uns com os outros durante a execução do sistema.

- As associações são representadas por linhas ligando as classes envolvidas. Essas linhas podem ter nomes ou títulos.

- Exemplo:

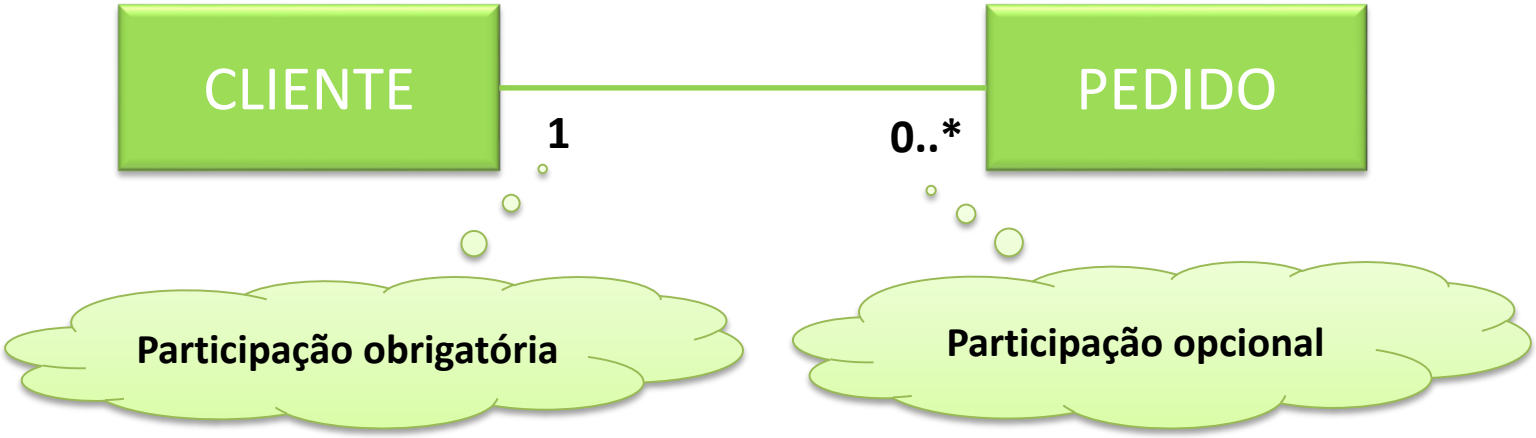
MULTIPLICIDADES DE UMA ASSOCIAÇÃO

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

Nome	Simbologia
Apenas Um	1..1 (ou 1)
Zero ou Muitos	0..* (ou *)
Um ou Muitos	1..*
Zero ou Um	0..1

PARTICIPAÇÃO

A participação de um objeto em uma associação indica a necessidade (ou não) da existência da ligação entre objetos.



ADORNOS PARA ASSOCIAÇÕES

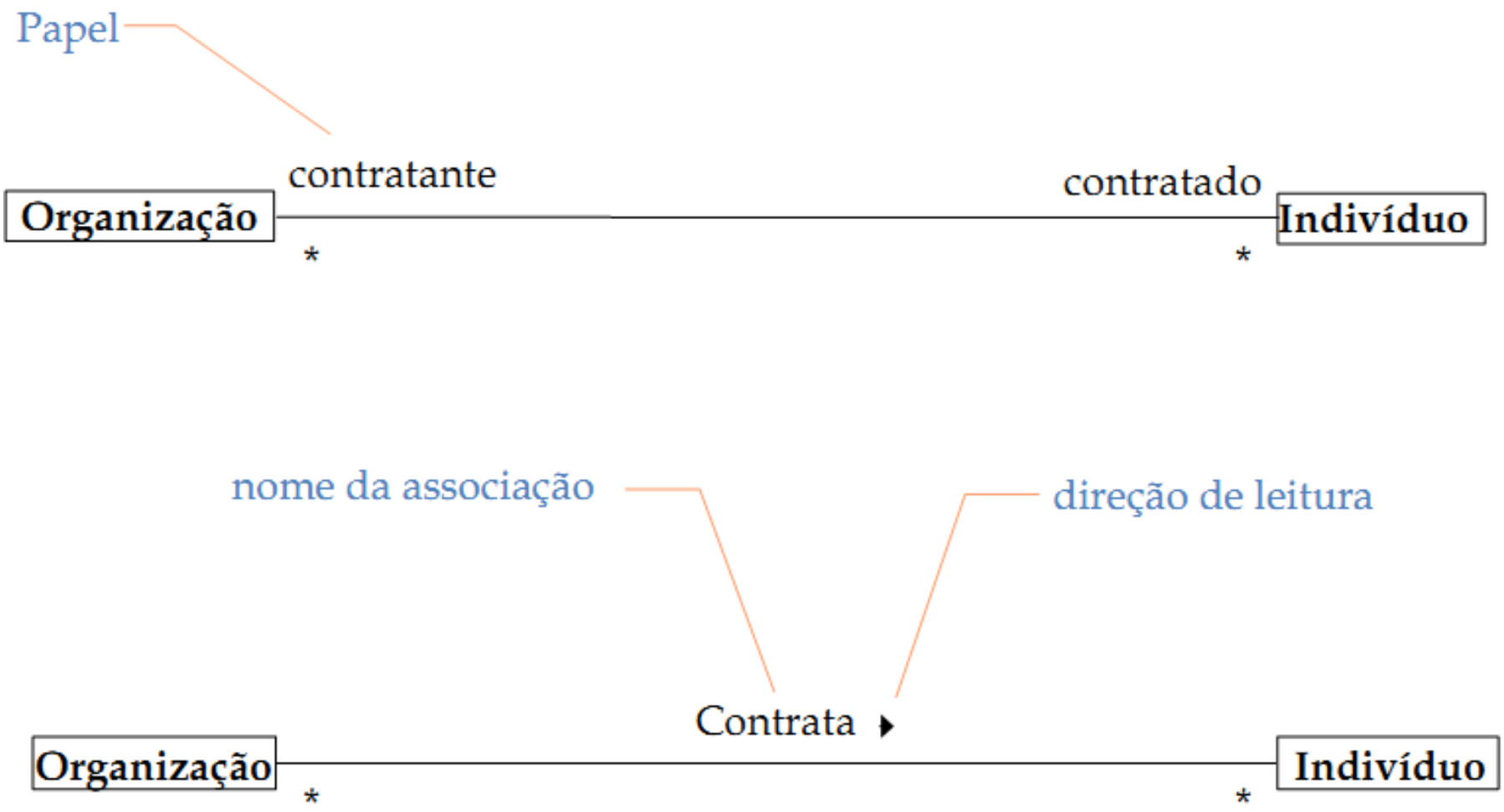
Para adicionar semântica a uma associação, a UML define três recursos de notação (adornos):

– **Nome:** indica o significado de uma associação.

– **Papel:** representa um papel desempenhado por um objeto em uma associação. Serve como substituto do nome da associação.

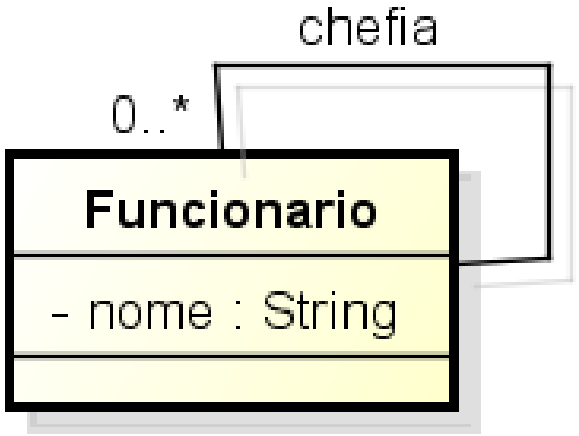
– **Direção de leitura:** indica como a associação deve ser lida. Serve para eliminar alguma ambiguidade na leitura da associação.

ADORNOS PARA ASSOCIAÇÕES



ASSOCIAÇÃO UNÁRIA OU REFLEXIVA

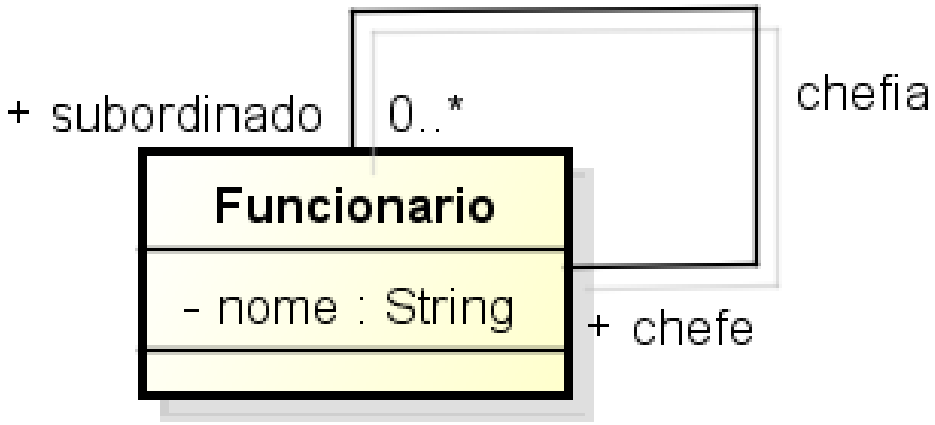
- Esse tipo de associação ocorre quando existe um relacionamento de um objeto de uma classe com objetos da mesma classe.



- Pode-se perceber que só existe multiplicidade em uma das extremidades, quando não existe a multiplicidade explícita, entende-se que a multiplicidade é “1..1” ou “1”.
- **Nesse exemplo um (1) funcionários pode ou não chefiar outros funcionários, mas um funcionário tem um e apenas um funcionário como chefe**.

ASSOCIAÇÃO UNÁRIA OU REFLEXIVA

- Nesse exemplo podemos perceber a função de cada objeto envolvido na associação.
- O objeto da extremidade (1) executa o **papel** de chefe, enquanto os objetos na extremidade de multiplicidade muitos interpretam o **papel** de subordinados.



ASSOCIAÇÕES REFLEXIVAS

- Quando se usa associações reflexivas, a definição de papéis é importante para evitar ambiguidades na leitura da associação.
  - Cada objeto tem um papel distinto na associação.

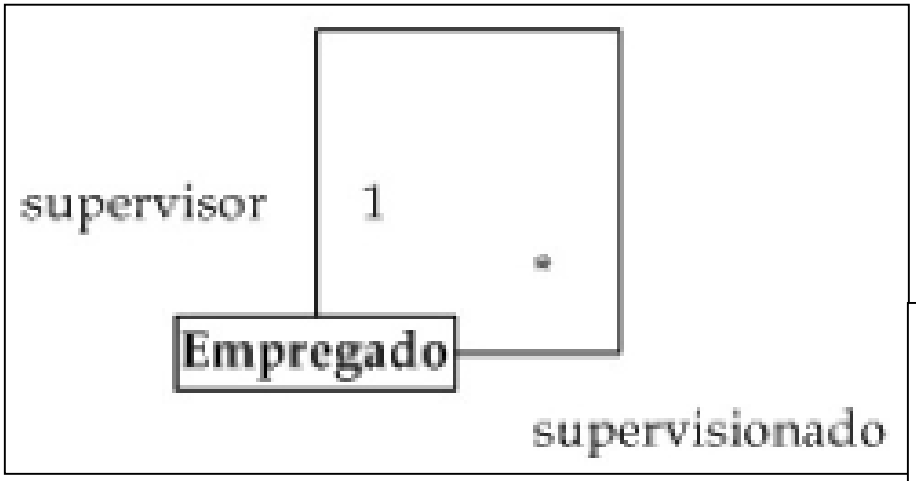
Exemplos:

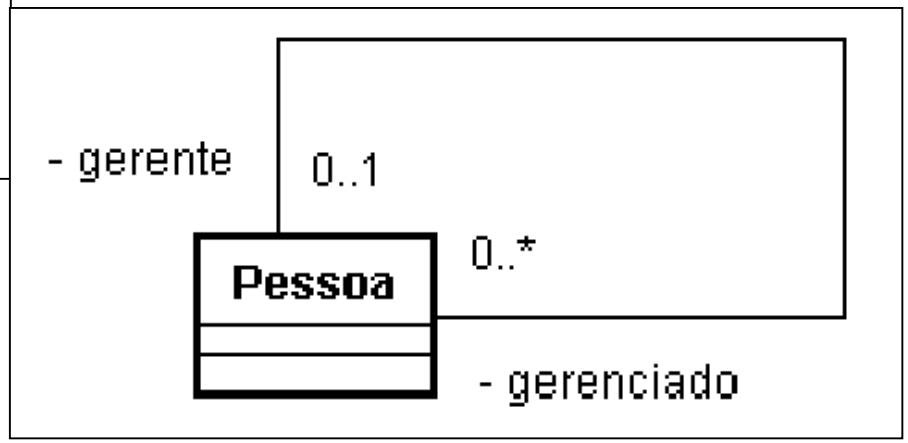
- Um circuito eletrônico é composto de diversos circuitos eletrônicos mais simples.
- Uma peça pode ser composta de diversas outras peças.
- Em um sistema de arquivos, um diretório pode conter outros diretórios.
- Uma cidade é vizinha de diversas outras cidades.
- Uma disciplina é pré-requisito de diversas outras disciplinas.
- Um funcionário supervisiona diversos outros funcionários.
- Um funcionário gerencia diversos outros funcionários.

NOTAÇÃO PARA AUTO-RELACIONAMENTOS

- Representado por uma linha conectando a classe a si mesma.
- Note que isso não significa que um objeto está associado a si próprio!

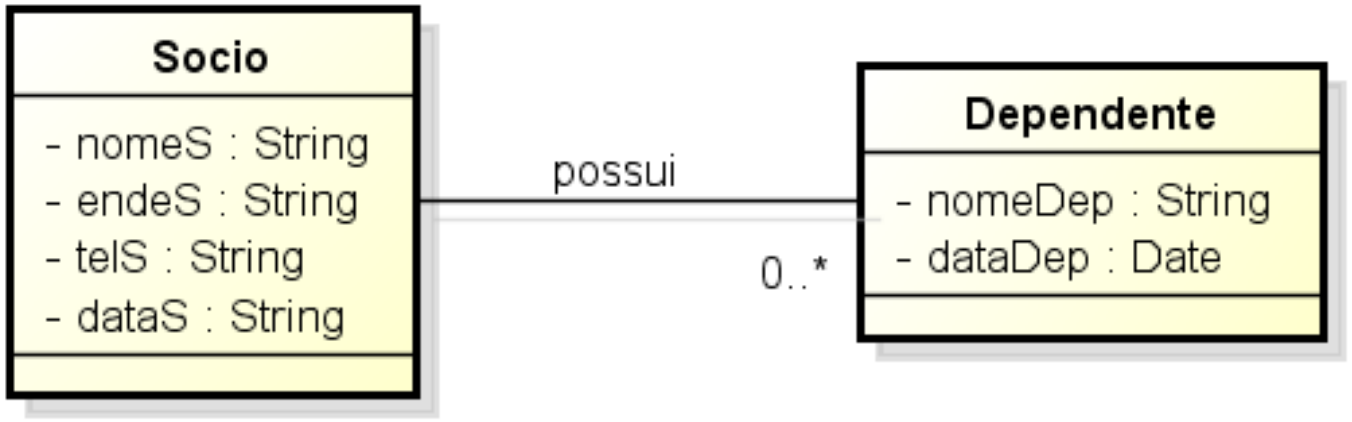
Exemplo:





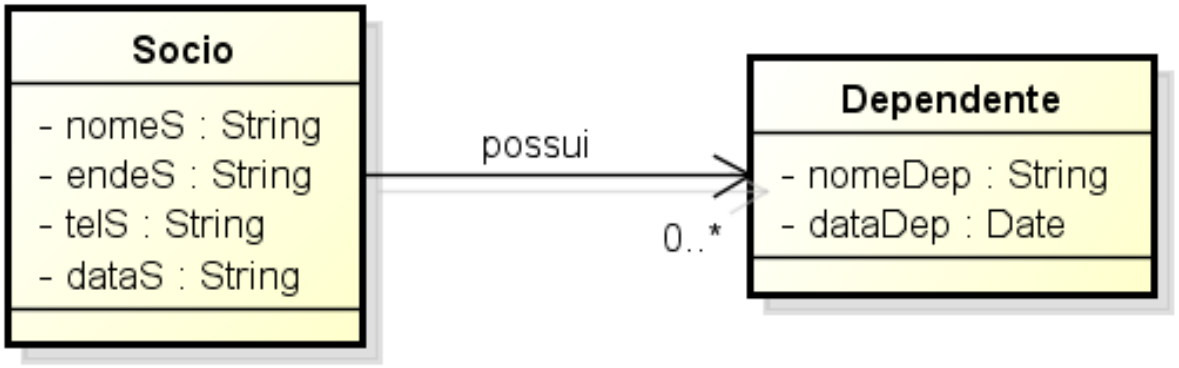
ASSOCIAÇÃO BINÁRIA

- Ocorrem quando são identificados relacionamentos entre objetos de duas classes distintas
- Podemos observar que o objeto da classe Sócio pode relacionar-se ou não com instâncias da classe Dependente



ASSOCIAÇÃO BINÁRIA - NAVEGABILIDADE

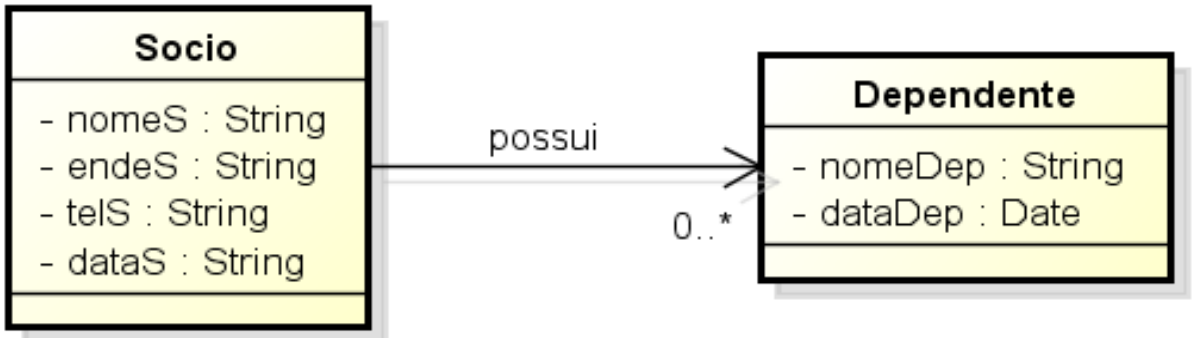
- A navegabilidade é representada por um seta em uma das extremidades da associação, identificando o sentido em que as informações são transmitidas entre os objetos das classes envolvidas. (Sentido em que os métodos poderão ser disparados).
- Um objeto da classe Sócio, poderá disparar métodos em objetos da classe dependente (o que não poderá ocorrer ao contrário).
- Se não houver setas indicando navegabilidade significa que as informações poderão trafegar entre objetos de todas as classes de associação.



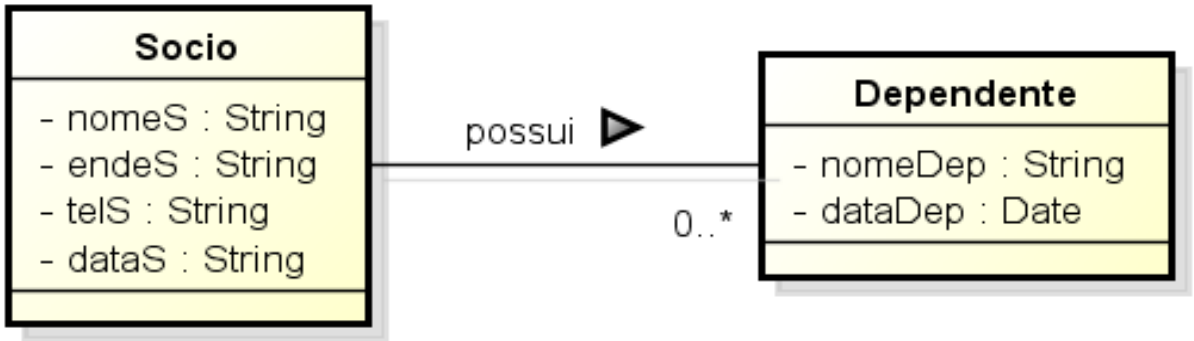
NAVEGABILIDADE X DIREÇÃO DE LEITURA

- A **navegabilidade** só deve ser acrescentada em **fase de projeto**, porém é possível durante a **fase de análise** definir a **direção de leitura.**
- A direção de leitura e a navegabilidade não são a mesma coisa.
  - **Direção de leitura:** facilitar a compreensão da associação.
  - **Navegabilidade**: define o sentido em que os métodos poderão ser disparados.

Navegabilidade



Direção de Leitura



ASSOCIAÇÃO TERNÁRIA OU N' ÁREAS

- Define-se o grau de uma associação como a quantidade de classes envolvidas na mesma.
- Na notação da UML, as linhas de uma associação **n-ária** se interceptam em um losango.
- Na grande maioria dos casos práticos de modelagem, as associações normalmente são **binárias**.
- Quando o grau de uma associação é igual a três, dizemos que a mesma é **ternária**.

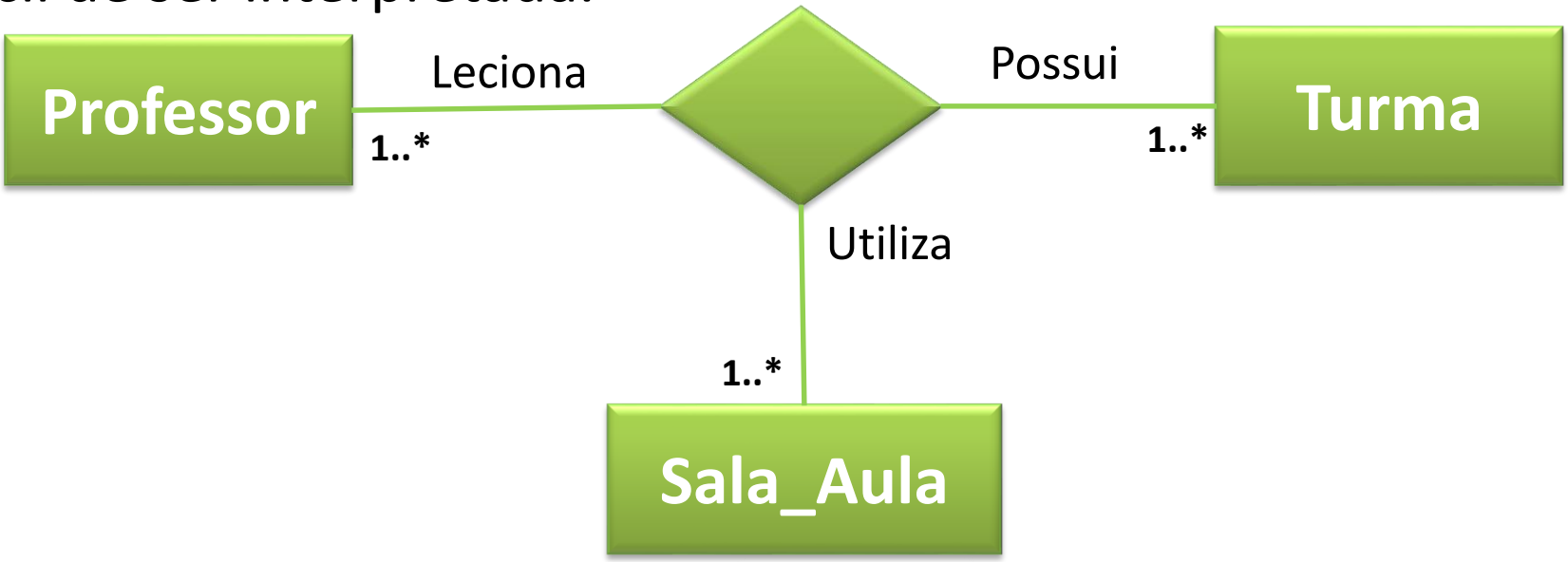
ASSOCIAÇÃO TERNÁRIA OU N' ÁREAS

- Conectam objetos de mais de duas classes. São representadas por um losango para onde convergem todas as ligações da associação.
- Na notação da UML, as linhas de uma associação n-ária se interceptam em um losango nomeado.
  - Notação similar ao do Modelo de Entidades e Relacionamentos

ASSOCIAÇÃO TERNÁRIA

“Um professor leciona para no mínimo uma turma e no máximo para muitas, uma turma tem no mínimo um professor e no máximo muitos, e um professor, ao lecionar para uma determinada turma, utiliza no mínimo uma sala de aula e no máximo muitas.”

As associações ternárias são úteis para demonstrar associações complexas. Deve-se evitar utilizá-las, pois sua leitura é por vezes, difícil de ser interpretada.



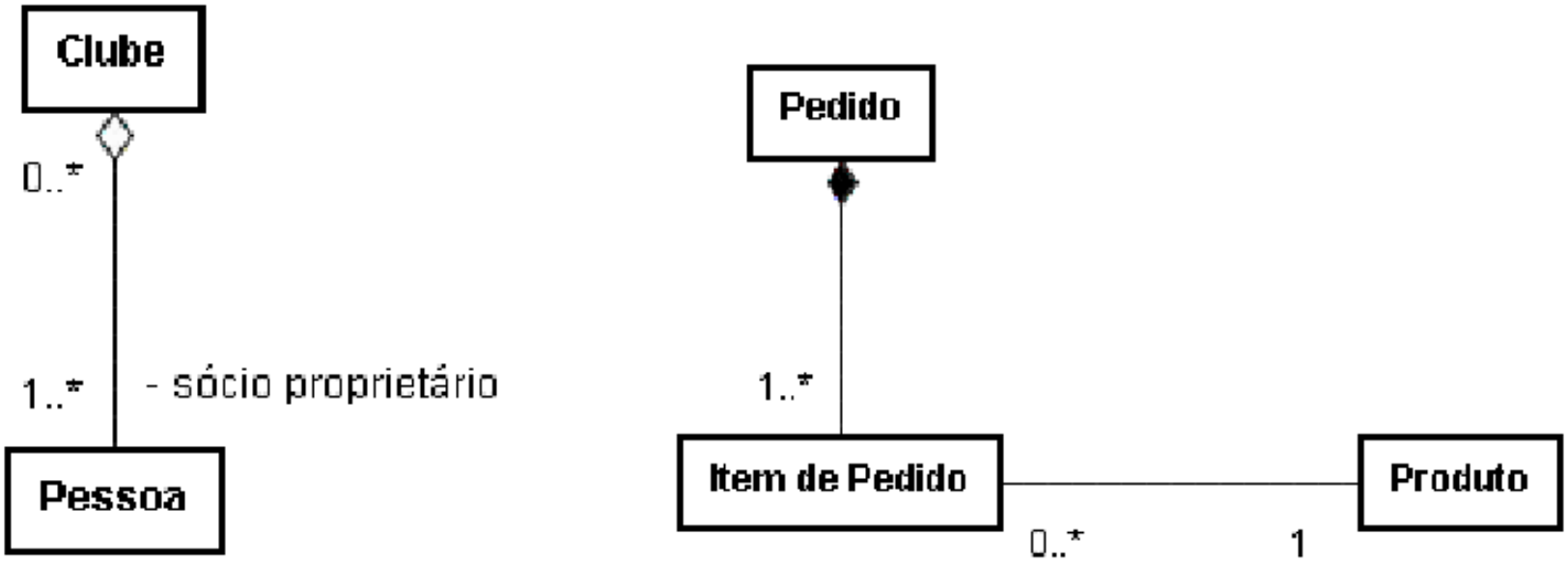
AGREGAÇÕES E COMPOSIÇÕES

- São casos especiais da associação
  - Onde for possível a utilização de uma agregação/composição, uma associação também poderá ser utilizada.
  - Multiplicidades, participações, papéis, etc. podem ser usados igualmente
- Utilizadas para representar conexões que guardam uma **semântica de todo-parte**.

- ✓ Agregações/composições podem se estender por diversos níveis.
  - ✓ O resultado são as hierarquias todo/parte.
  - ✓ O relacionamento todo/parte é transitivo.

AGREGAÇÕES E COMPOSIÇÕES NA UML

- Agregação: linha conectando as classes relacionadas, com um diamante (losango) branco perto da classe que representa o todo.
- Composição: linha conectando as classes relacionadas, com um diamante (losango) preto perto da classe que representa o todo.



AGREGAÇÕES

Na agregação:

– Os objetos que desempenham o papel “parte do todo” são criados e destruídos independentemente do objeto que desempenho o papel de “todo”.

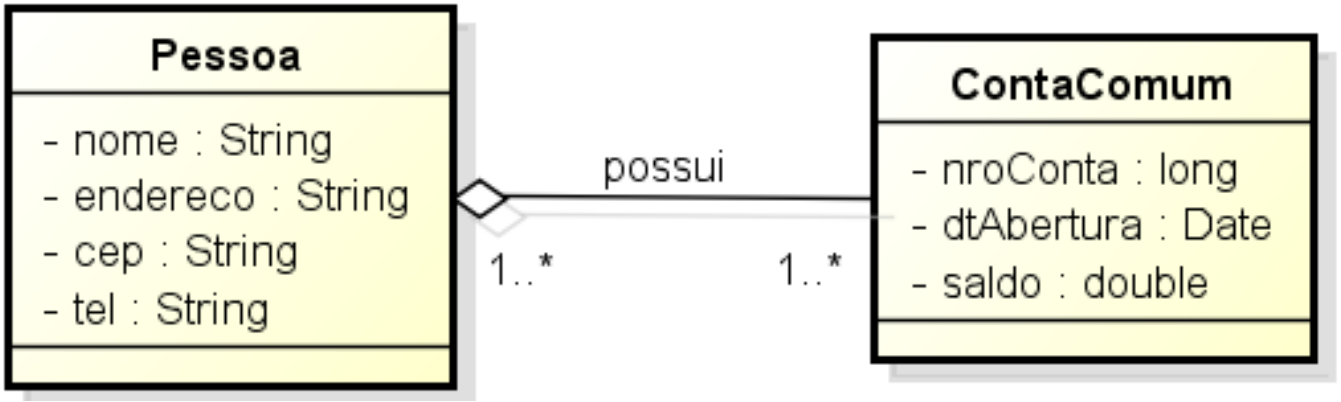
– Um objeto parte pode ser utilizado para compor diversos todos.

– A destruição de um desses objetos todo não implica na destruição do objeto parte (nem vice-versa).

- Em uma agregação, as partes “podem viver” sem o todo.

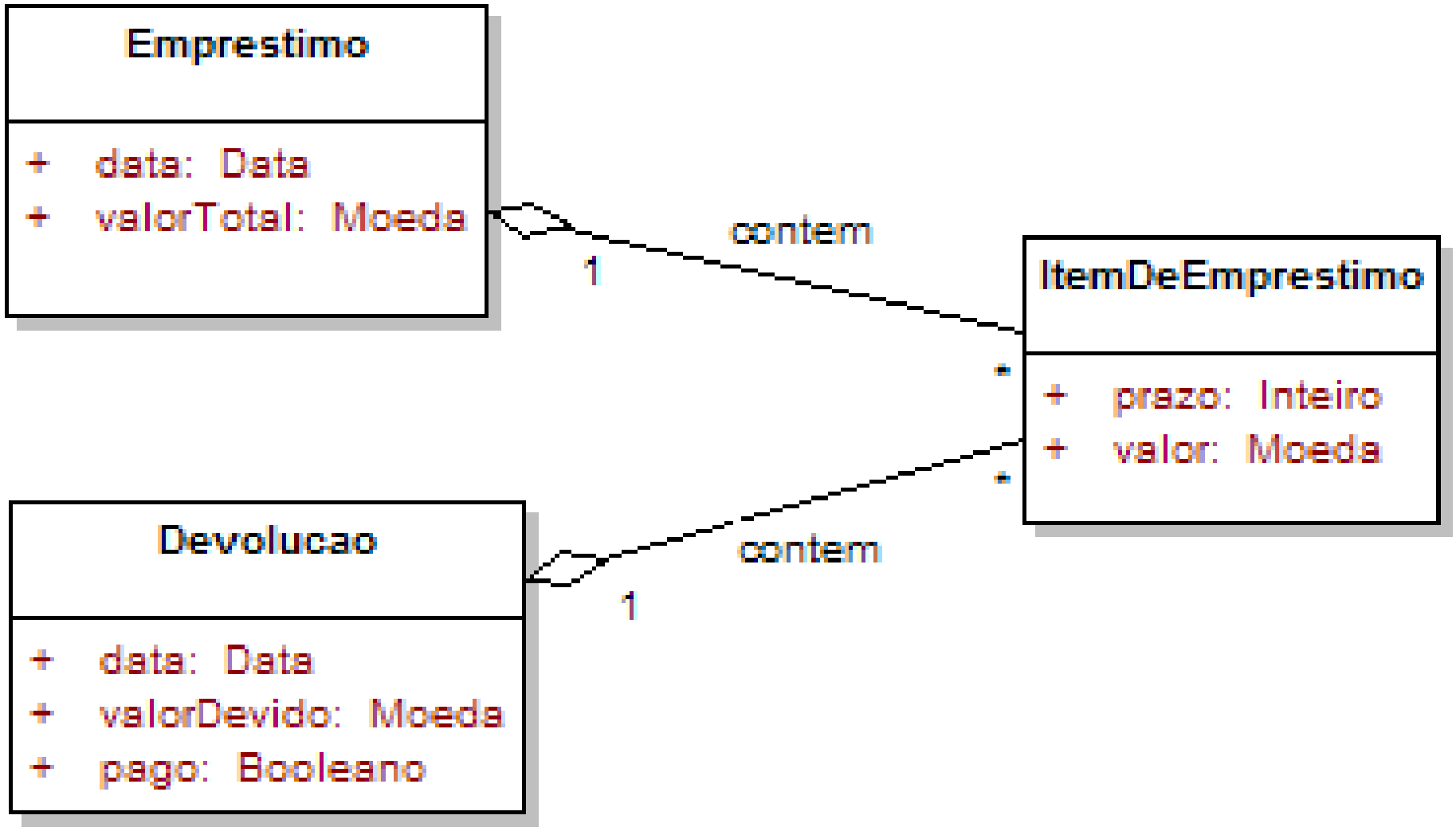
AGREGAÇÕES

- É um tipo especial de associação onde se tenta demonstrar que as informações de um objeto (chamado objeto-todo) precisam ser complementadas pelas informações contidas em um ou mais objetos de outra classe (chamados objetos-parte).
  - Pessoa são objetos-todo que precisam ter suas informações complementadas pelos objetos da classe ContaComum, que nessa associação são objetos-parte



Uma pessoa pode possuir muitas contas, uma conta pode ser possuída por muitas pessoas, como no caso de uma conta conjunta.

AGREGAÇÕES COMPARTILHADA



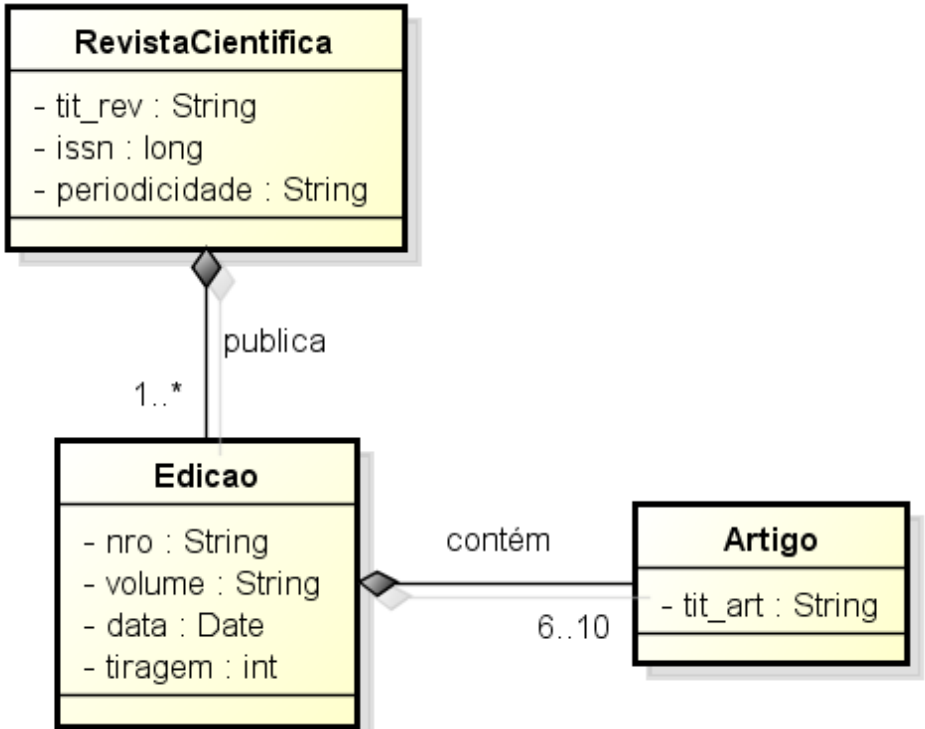
COMPOSIÇÃO

- Constitui-se em uma variação da agregação, onde é apresentado um vínculo mais forte entre os objetos-todo e os objetos-parte, procurando demonstrar que os objetos-parte têm de estar associado a um único objeto-todo.
- Em uma composição os objetos-parte não podem ser destruídos por um objeto diferente do objeto-todo ao qual estão relacionados.

COMPOSIÇÃO

`RevistaCientifica` refere-se a, no mínimo um objeto da classe `Edicao`, podendo se referir a muitos objetos dessa classe, e que cada instância da classe `Edicao` relaciona-se única e exclusivamente a uma instância específica da `RevistaCientifica`, não podendo relacionar-se com nenhuma outra.

A classe `Edicao` deve se relacionar a no mínimo seis objetos da classe `Artigo`, podendo se relacionar com até 10 objetos.

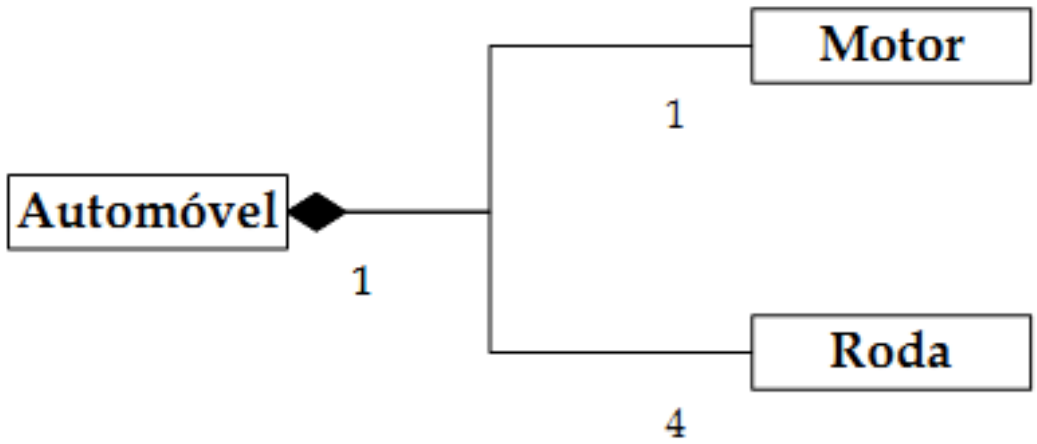


COMPOSIÇÕES

Na composição:

- os objetos parte pertencem a um único todo.
- objetos parte são criados e destruídos pelo objeto todo.

- No objeto composto, são definidas operações para adicionar e remover objetos componentes

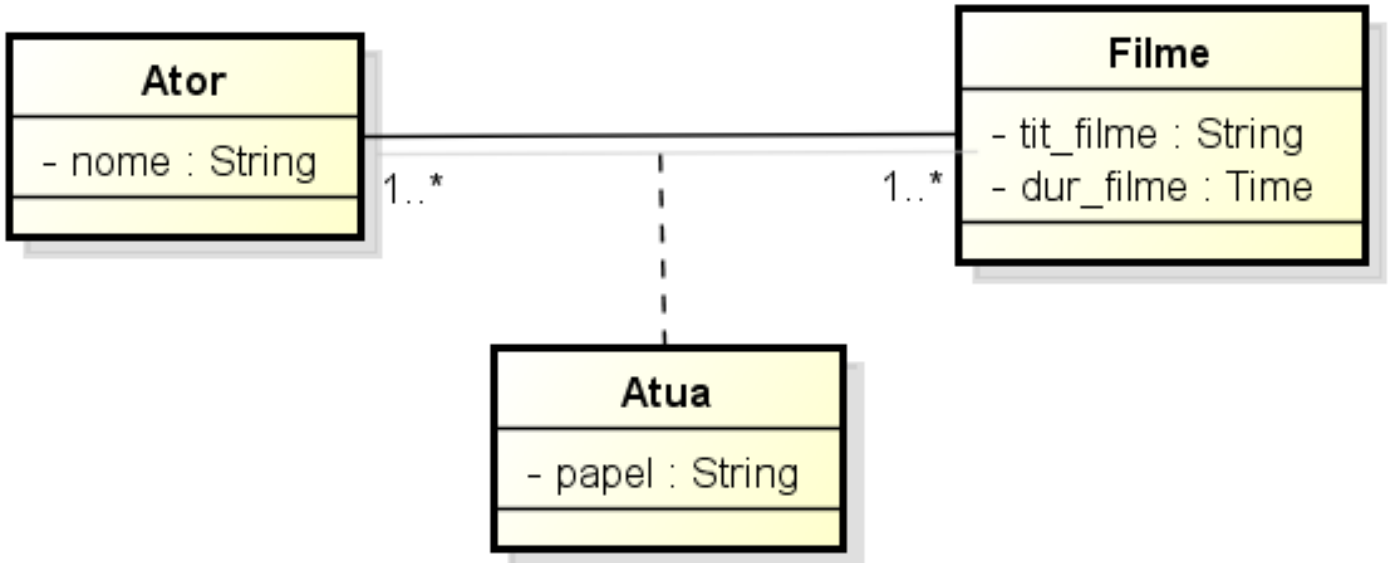


CLASSE ASSOCIATIVA

- São classes produzidas quando da ocorrência de associações que tenham multiplicidade muitos(*) em todas as suas extremidades.
- Essas classes são necessárias nos casos em que existem atributos relacionados à associação que não podem ser armazenados por nenhuma das classes envolvidas.

CLASSE ASSOCIATIVA

- Um instância da classe Ator pode se relacionar com muitas instâncias da classe Filme, e uma instância da classe Filme pode se relacionar com muitas instâncias da classe Ator, ou seja, um ator pode atuar em muitos filmes, e um filme pode ter muitos atores atuando nele.
- Ocorre que existe a necessidade de saber qual o papel interpretado por um ator em um determinado filme, mas onde armazenar a informação?
- Como existe a multiplicidade muitos (*) nas extremidades de ambas as classes de associação, não há como reservar atributos para armazenar as informações decorrentes da associação em quaisquer das classes.



CLASSES DE ASSOCIAÇÃO NA UML

O exemplo abaixo ilustra a notação UML para classes de associação:

O exemplo muito-para-muitos na figura abaixo significa que cada Funcionário associa-se com muitos Projetos, e cada Projeto associa-se com muitos Funcionários. Esse exemplo também ilustra o uso de uma associação de classe. Se uma associação tem atributos, esses são escritos em uma classe que é ligada a associação com uma linha tracejada. A associação de classe chamada **AtribuiçãoTrabalho** na figura abaixo contém duas associações de atributos chamados **atribuiçãoTarefa** e **dataInício**. A associação e a classe juntas formam uma associação de classe.