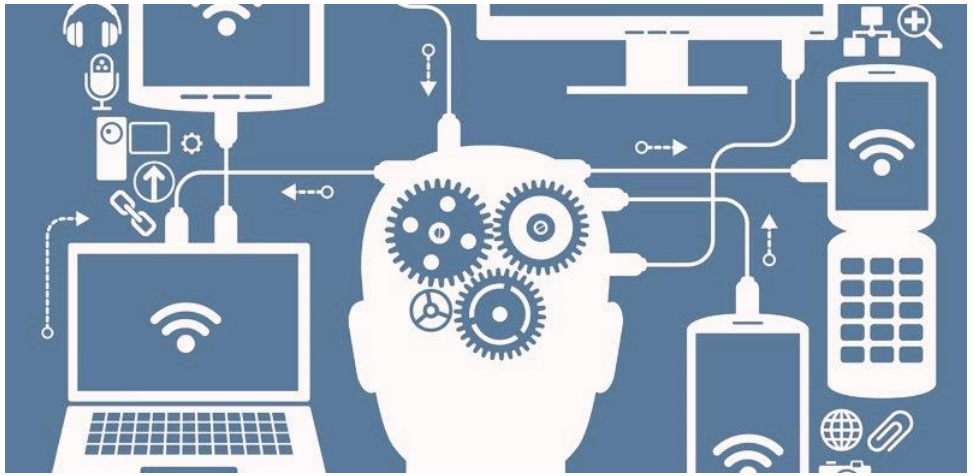
UNIVERSIDADE
DO ESTADO DE MINAS GERAIS | UEMG
UNIDADE PASSOS

Engenharia de Software 2



Prof. Esp. João Paulo S. Araújo

É sempre bom lembrar...

Modelagem de Sistema:

- *Processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo representa uma visão ou perspectiva, diferente do sistema.*

- *Normalmente representa o sistema com algum tipo de notação gráfica, quase sempre baseada em notações* ***UML (Unified Modeling Language)***

- *Modelos são usados durante os processos de Engenharia de Requisitos para ajudar a extrair os requisitos do sistema.* *(sistemas novos e sistemas já existentes)*

(SOMMERVILLE, 2011)

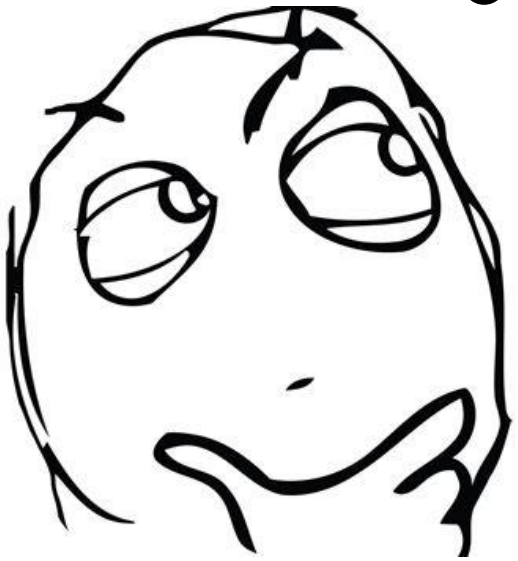
Modelagem UML

Diagrama de Classes

Diagrama de Classes

- Diagrama mais utilizado da UML
- Serve de apoio para a maioria dos outros diagramas
- Reflete a estrutura do código
  - *É o diagrama mais próximo da implementação.*

Opa opa peraí!
Mas o que é uma
Classe mesmo?



Revisando alguns conceitos...

- Classe:
  - *É a especificação que o objeto (do mundo real) vai conter;*
  - *Define as características e operações dos objetos.*

- Objeto:
  - *É a representação de algo do mundo real;*
  - *É uma referência a uma especificação (classe);*
  - *É a instância de uma classe.*

Diagrama de Classes

- Composto por
  - **Classes** – Cada classe com seus *atributos* e *métodos*
  - **Associações** – Relacionamento entre as classes

- "Ideia" do diagrama Entidade-Relacionamento (ER)
  - *Ou Modelo Entidade Relacionamento (MER)*

Diagrama de Classes – *Principais objetivos*

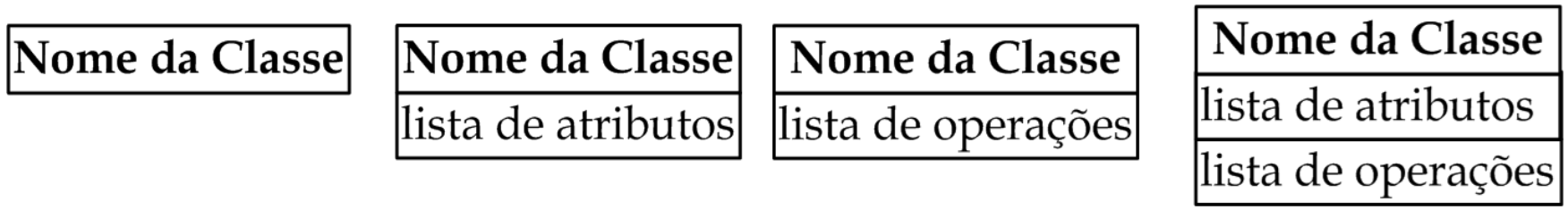
- Modelar os elementos de um programa orientado a objetos em tempo de desenvolvimento
  - ***Classes*** com seus *atributos* e *métodos*

- Modelar os relacionamentos entre classes, de forma mais explícita que aquela do código
  - *Associação*
  - *Agregação (e composição)*
  - *Herança*
  - *Dependência*

Notação de uma Classe

- Representada através de uma "caixa" com no máximo três divisões.

- Notação utilizada depende do nível de abstração desejado.

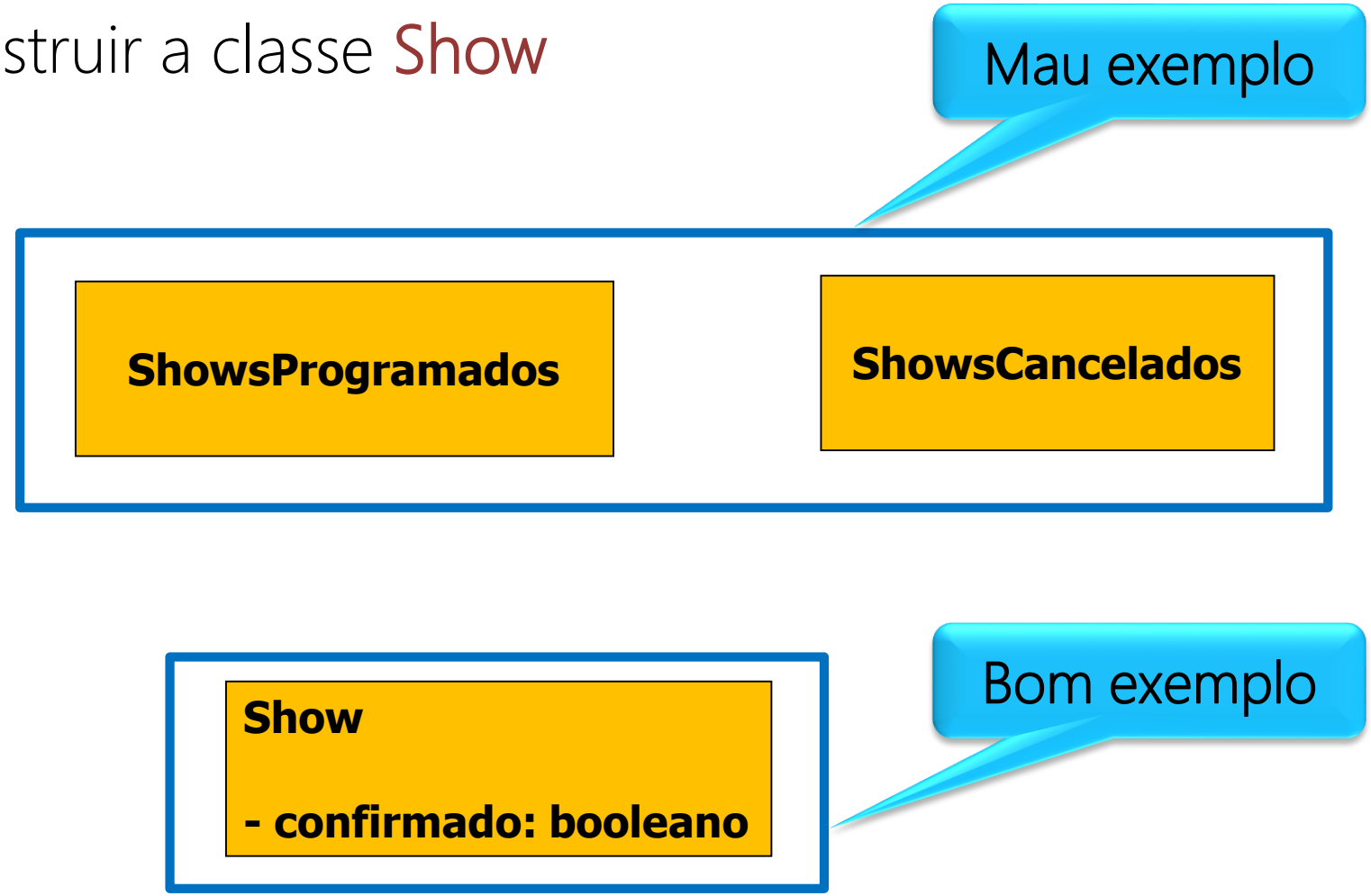


Notação de uma Classe - *Exemplo*

- Classe: ContaBancaria

Bons e maus exemplos do uso de Classes

- Construir a classe Show



Classes – *Atributos*

- São as características / propriedades da classe.
  - *Properties*
- Na UML, ***atributos*** são mostrados com pelo menos seu ***nome***
- Podem também mostrar seu *tipo*, *valor inicial* e outras propriedades.

Classes – *Métodos*

- São comportamentos (operações) de uma classe.
  - *Similares as Functions/Procedures*
- ***Métodos*** são exibidos com pelo menos seu ***nome***
- Podem também mostrar seus ***parâmetros*** e ***valores de retorno***.

Classes – *Acesso e visibilidade*

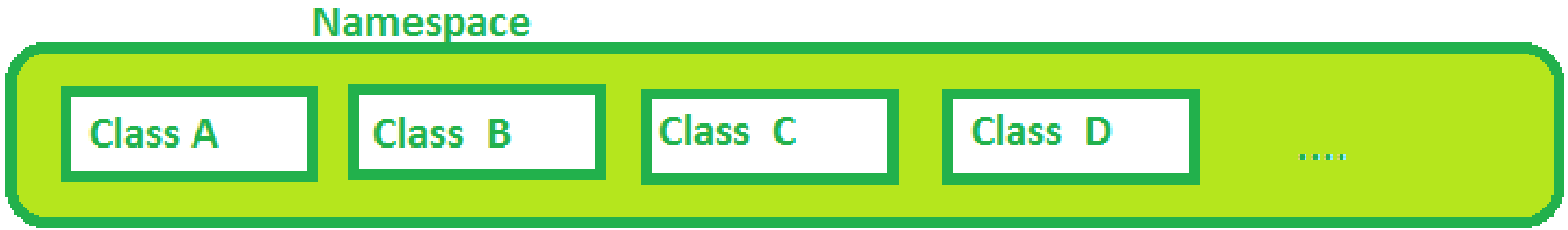
- Agrupamentos Lógicos e Organização
  - *Package* – *"Pacote" (Em Java)*
    ou *Namespace* – *"Espaço de Nomes" (Em .NET)*

- Modificadores de Acesso
  - *Public* (+) – Público
  - *Protected* (#) – Protegido
  - *Private* (-) – Privado

Classes – *Agrupamento e Organização*

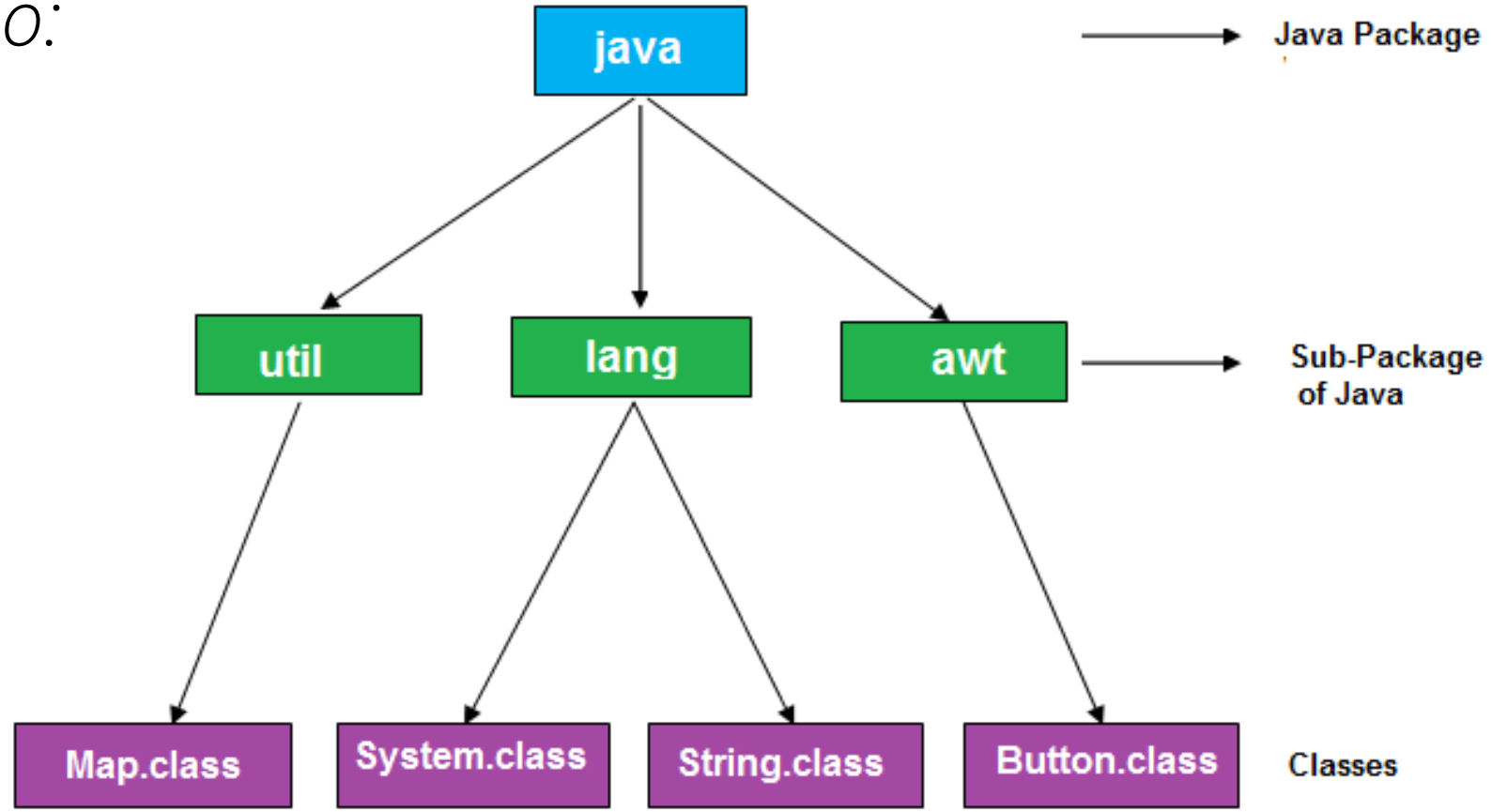
- *Na medida em que os sistemas crescem passamos a ter diversas classes e pode surgir problemas em relação a organização de todas as classes;*
- *Para isto existe o conceito de agrupamento lógico de classes*
  - *Package* (ou *Namespace) – Agrupar e organizar classes que tem alguma relação*
- *Trata-se de um agrupamento lógico de classes com escopos e características similares. (Agrupamento lógico de "tipos")*

Classes – *Agrupamento e Organização*



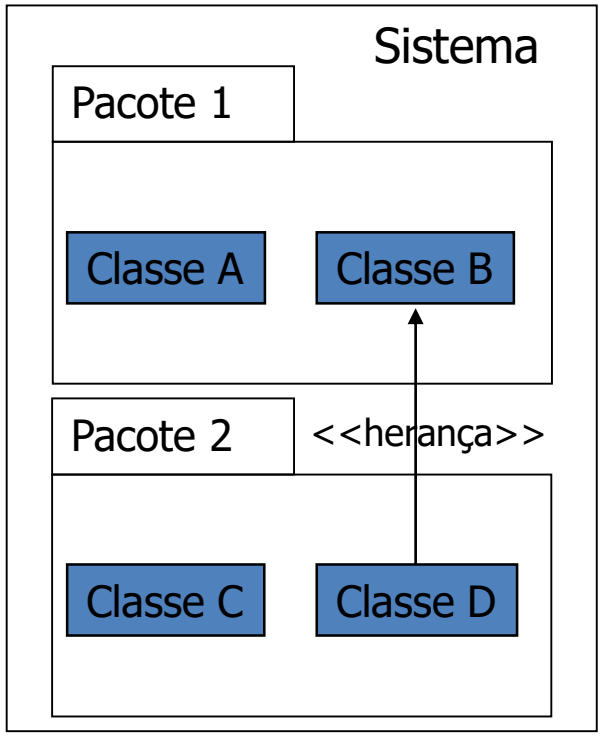
Classes – *Agrupamento e Organização*

Exemplo:



Classes – *Formas de Acesso*

- *Public* (+)
  - Acessível para qualquer elemento que possa ver a classe.

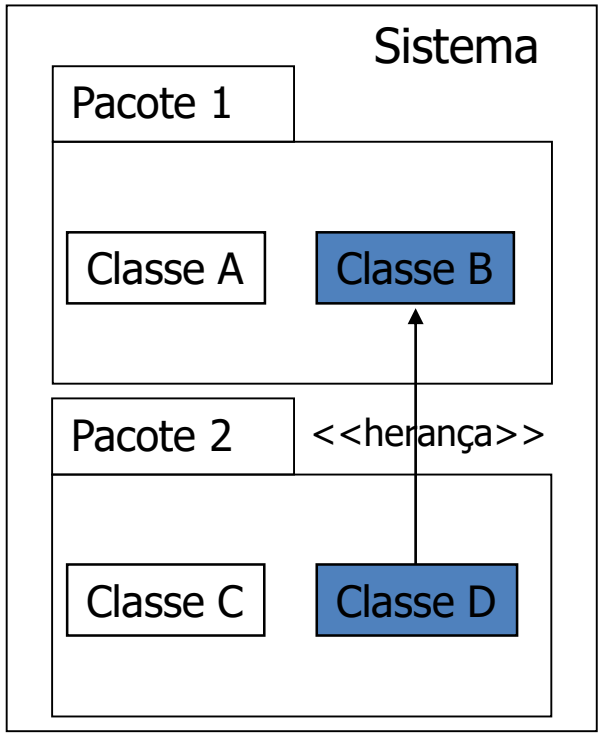


Atributos e Métodos Públicos:

- Podem ser acessados por qualquer outra classe do sistema.

Classes – *Formas de Acesso*

- *Protected* (#)
  - Acessível a outros elementos dentro da classe e de subclasses (herança).

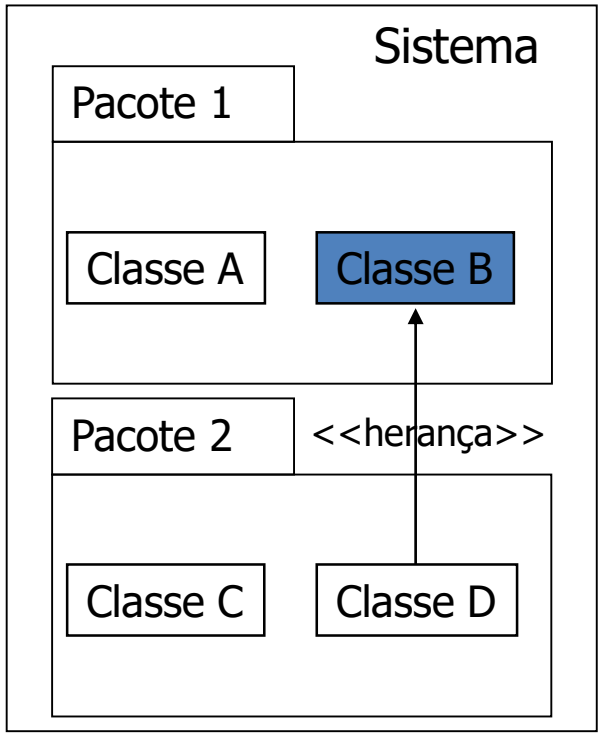


Atributos e Métodos Protegidos:

- Só podem ser acessados por uma classe que tenha uma relação de *generalização/especialização* independente do pacote;

Classes – *Formas de Acesso*

- *Private* (-)
  - Acessível somente para outros elementos que dentro da própria classe.

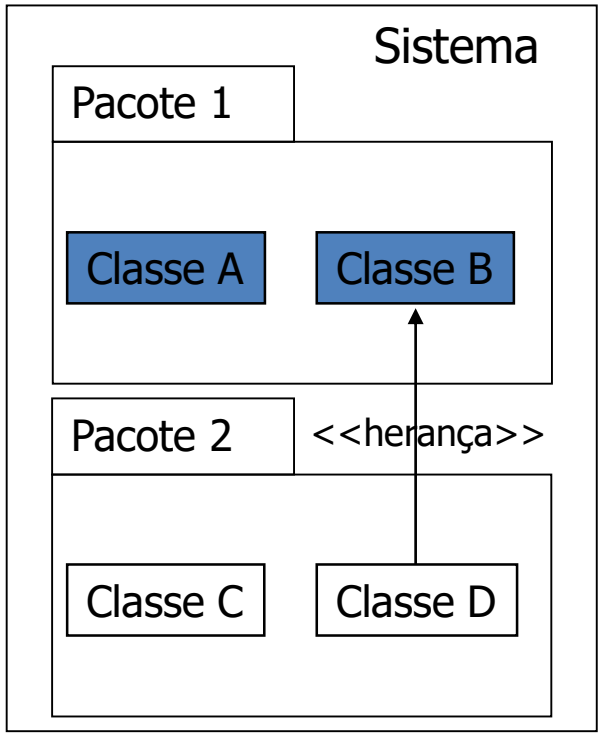


Atributos e Métodos Privados:

- *Só são acessíveis dentro do mesmo objeto.*
- *Normalmente este é o estado inicial dos atributos, depois se necessário, pode ser modificado.*

Classes – *Formas de Acesso*

- *Package* (~) *(ou namespace)*
  - Visível a elementos do mesmo pacote.



Atributos e Métodos Privados:

- *Só são acessíveis dentro do mesmo objeto.*
- *Normalmente este é o estado inicial dos atributos, depois se necessário, pode ser modificado.*

Recomendação para estabelecimento das formas acesso e visibilidade

- Atributos → *Privados* ou *Protegidos*
  - Princípio da ocultação de informação do paradigma de orientação a objetos
  - Possibilita que atributos herdados ou definidos na classe sejam tratados de maneira uniforme.
- Evitar atributos *Públicos*

Recomendação para estabelecimento das formas acesso e visibilidade

- Métodos → *Públicos*
  - O meio (objeto) externo acessa uma classe através de seus métodos.

- Necessidades específicas podem justificar a modificação das formas de acesso.

Diagrama de Classes - *Relacionamentos*

- Relacionamento entre classes
  - *É a maneira na qual as classes conectam entre si com o intuito de compartilhar informações e colaborarem umas com as outras para permitir a execução de processos.*

- Em POO há três tipos de relacionamentos:
  - *Especializações/Generalizações (Heranças)*
  - *Associações*
  - *Dependências*

Diagrama de Classes - *Relacionamentos*

- Generalização / Especialização
  - Relacionamento entre **classes mais gerais** a outras **classes mais específicas** onde o elemento mais específico **herda** as propriedades e métodos do elemento mais geral.
  - Conhecido como relacionamento de **superclasse/subclasse**
    - **Subclasse** (classe derivada ou filha): consiste na classe que herda todos os métodos e atributos de uma classe existente;
    - **Superclasse** (classe base ou classe mãe): é a classe existente que é herdada por uma outra classe.
  - É implementada como herança em *POO*.

Diagrama de Classes - *Relacionamentos*

- Generalização / Especialização – Exemplo

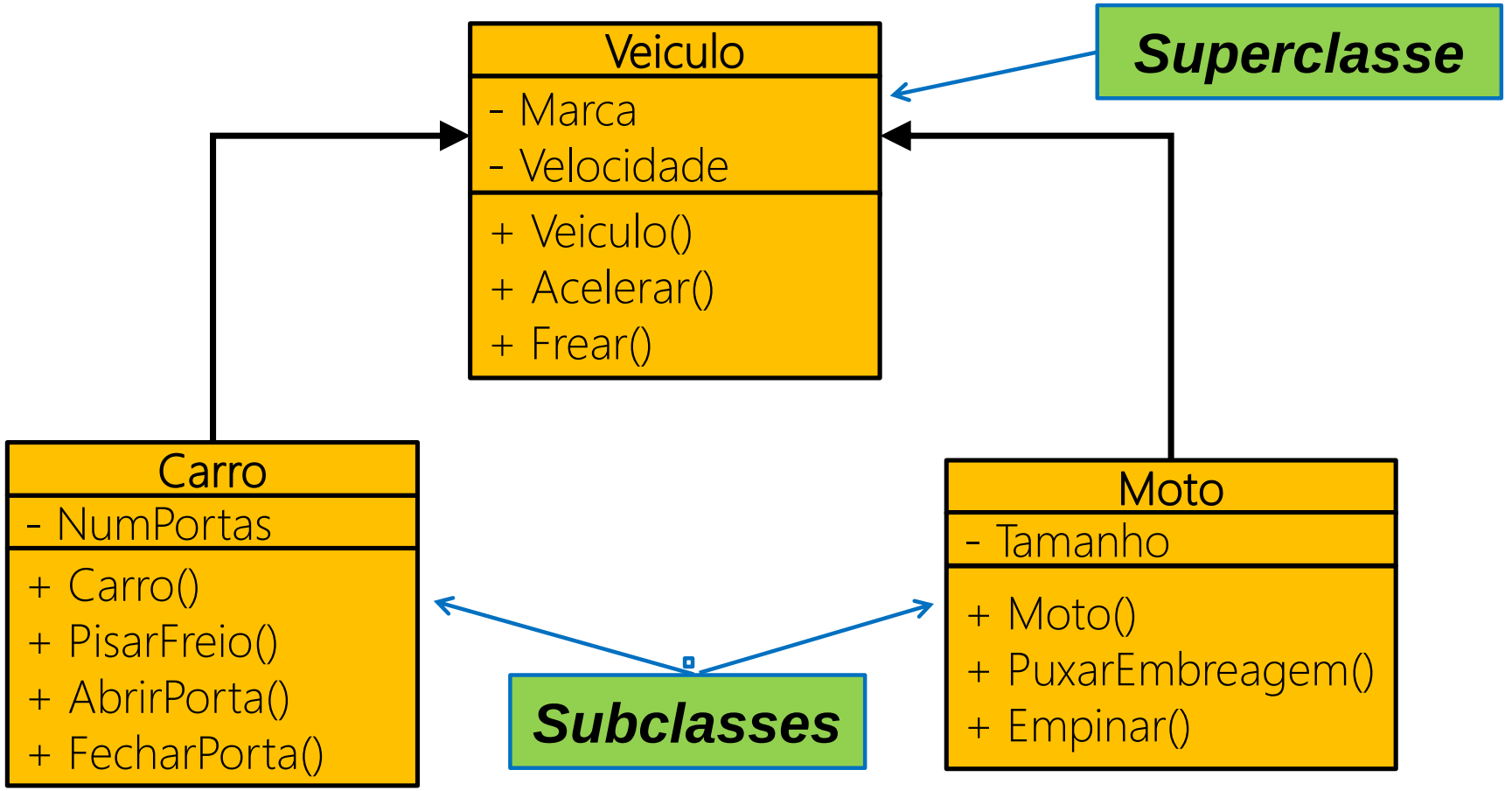


Diagrama de Classes - *Relacionamentos*

- Generalização / Especialização – Exemplo

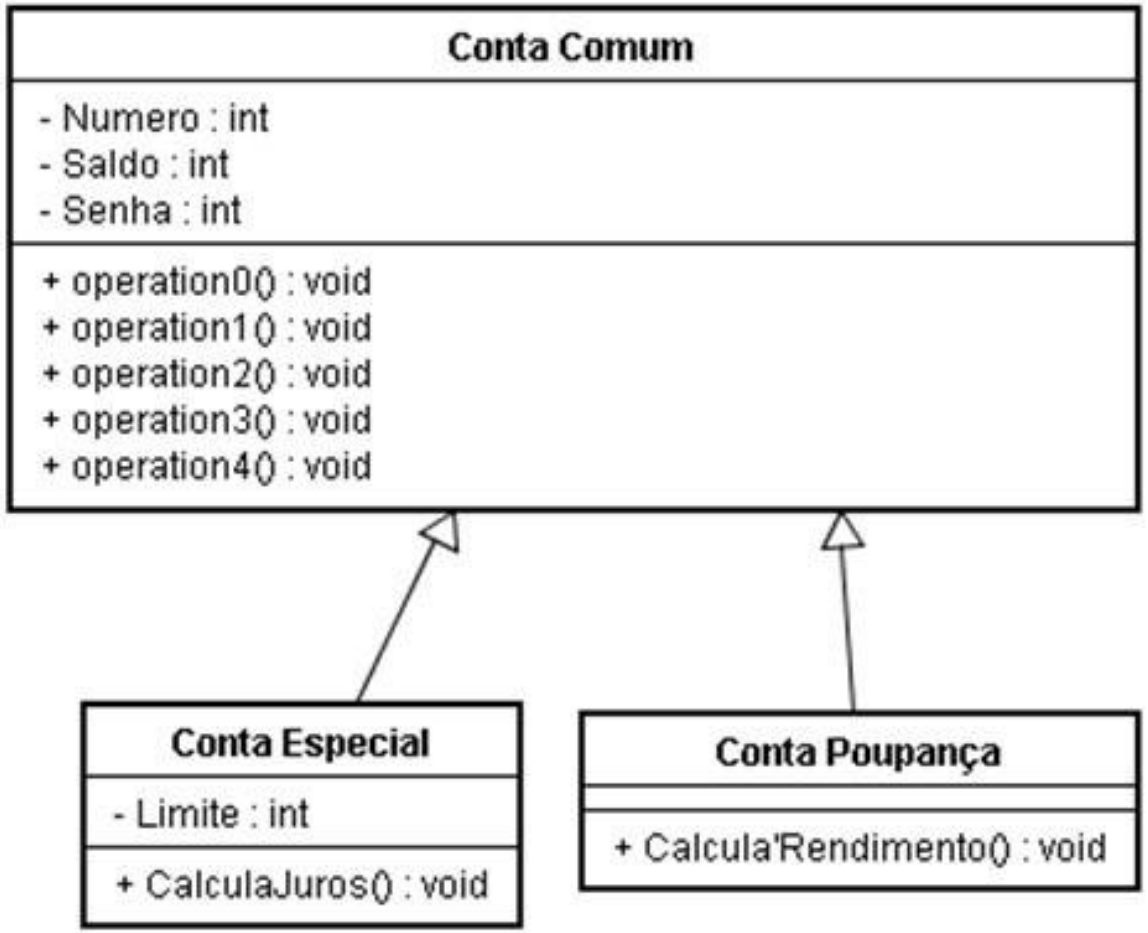


Diagrama de Classes - *Relacionamentos*

- Associação
  - É uma conexão entre classes.
  - Relacionamentos estruturais entre instâncias e especificam que *objetos de uma classe* estão ligados *a objetos de outras classes*, podendo haver troca de informações e compartilhamento de métodos.
  - Ocorre normalmente entre duas classes (binária), entre uma classe com ela mesma (unária) e entre várias classes (ternária/N-ária).
  - "Equivale" aos relacionamentos E-R.

Diagrama de Classes - *Relacionamentos*

- Associação – Exemplos

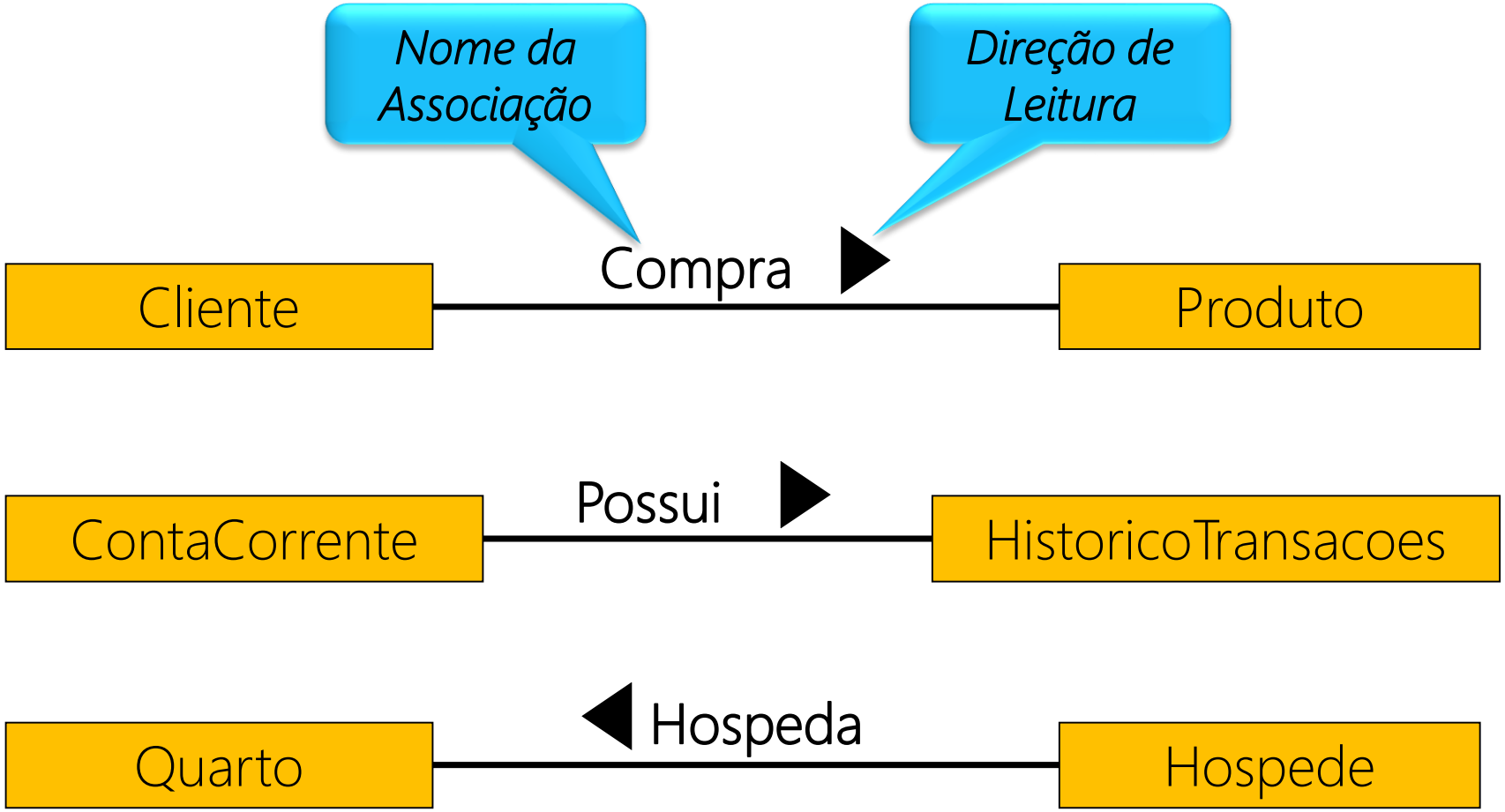


Diagrama de Classes - *Relacionamentos*

- Associação
  - **Participação**: Consiste no **papel** *(pode ser uma função ou cargo)* de uma determinada classe em um relacionamento com outra classe.

  - Exemplo:

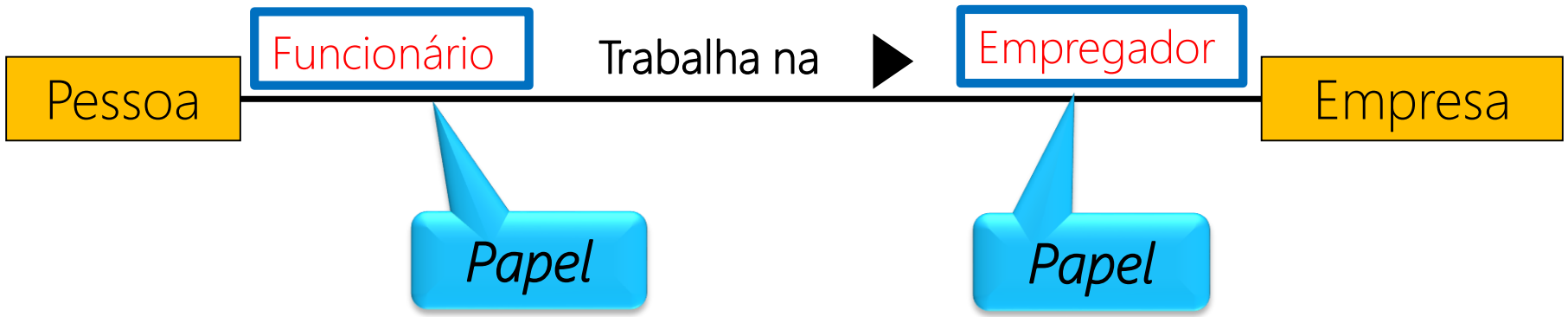


Diagrama de Classes - *Relacionamentos*

- Associação
  - **Multiplicidade**: é a quantidade mínima e máxima de objetos que podem ser conectados pela instancia de uma associação.
  - Exemplo:

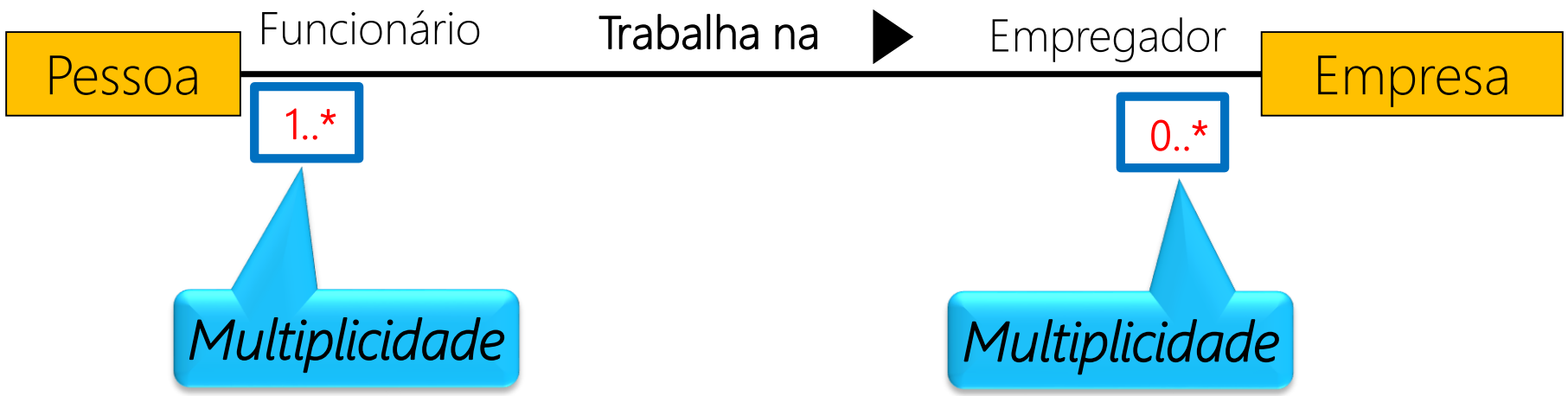


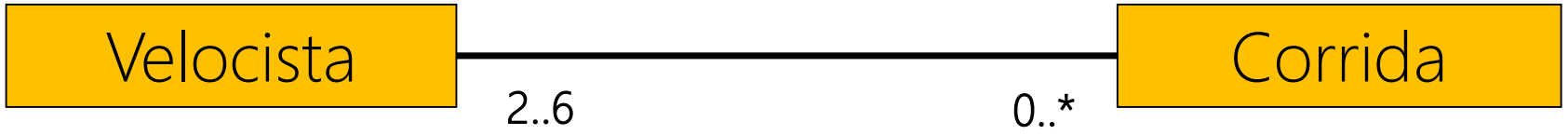
Diagrama de Classes - *Multiplicidade*

- Simbologia e Significado

Nome	Simbologia
Apenas um	**1**
Zero ou muitos	**0..***
Um ou muitos	**1..***
Zero, um ou muitos	*****
Zero ou um	**0..1**
Intervalo específico	**ex.: 2..8**

Diagrama de Classes - *Multiplicidade*

- Exemplos de Multiplicidades

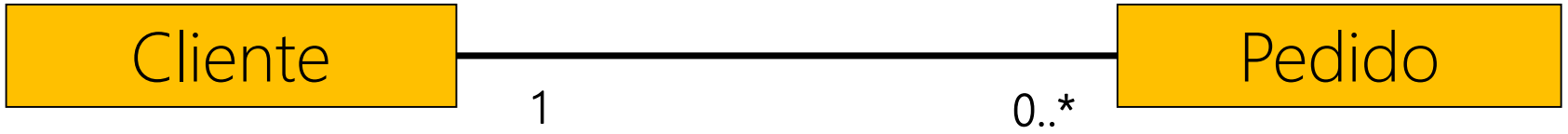


  - **Uma corrida** está associada a, no mínimo, **dois velocistas**
  - **Uma corrida** está associada a, no máximo, **seis velocistas**.

  - **Um velocista** pode estar associado a **nenhuma corrida**.
  - **Um velocista** pode estar associado a **várias corridas**.

Diagrama de Classes - *Multiplicidade*

- Exemplos de Multiplicidades



  - **Um cliente** pode estar associado a **nenhum pedido**.
  - **Um cliente** pode estar associado a **vários pedidos**.

  - **Um pedido** está associado a **um**, **e somente um**, **cliente**.

Diagrama de Classes - *Conectividade*

- Corresponde ao tipo de associação entre classes:
  - "muitos para muitos", "um para muitos" e "um para um".

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..***

Diagrama de Classes - *Conectividade*

- Exemplos de Conectividades

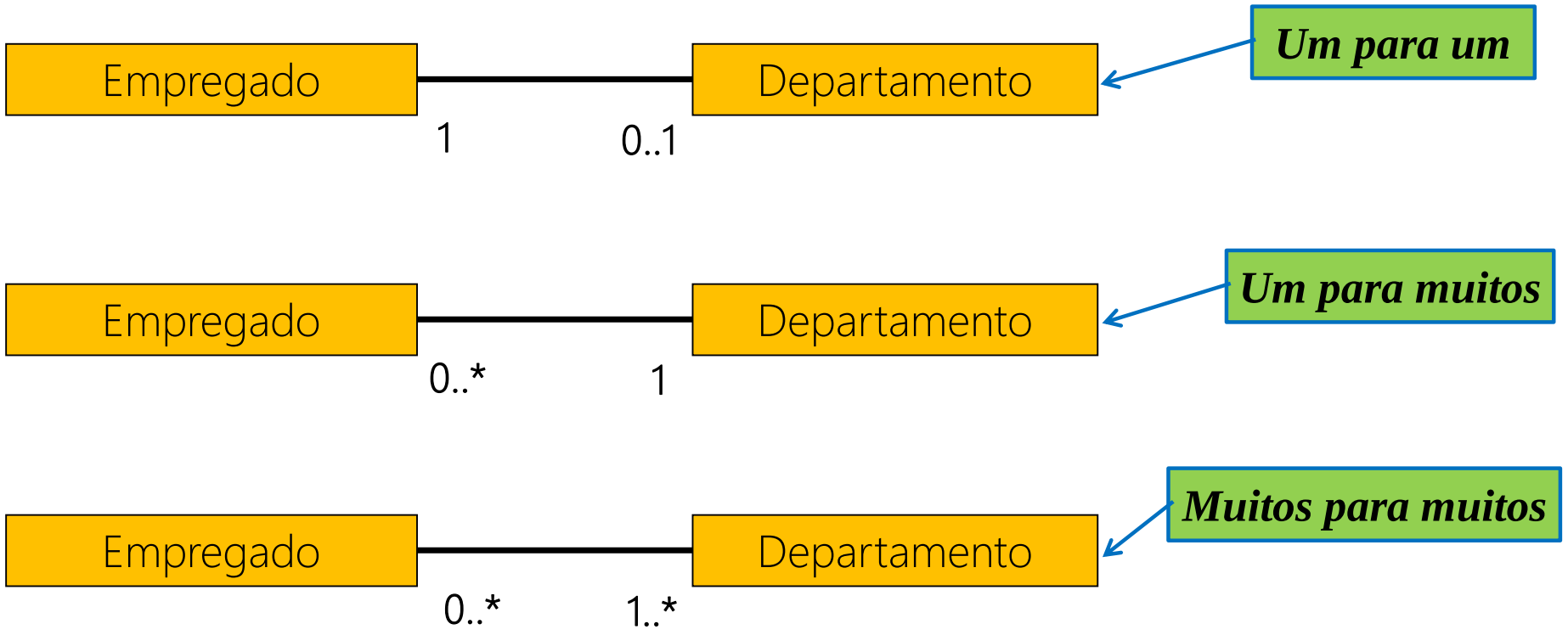


Diagrama de Classes - *Relacionamentos*

- Associação Binária
  - Associações entre duas classes
  - Mais comum
  - Exemplo:

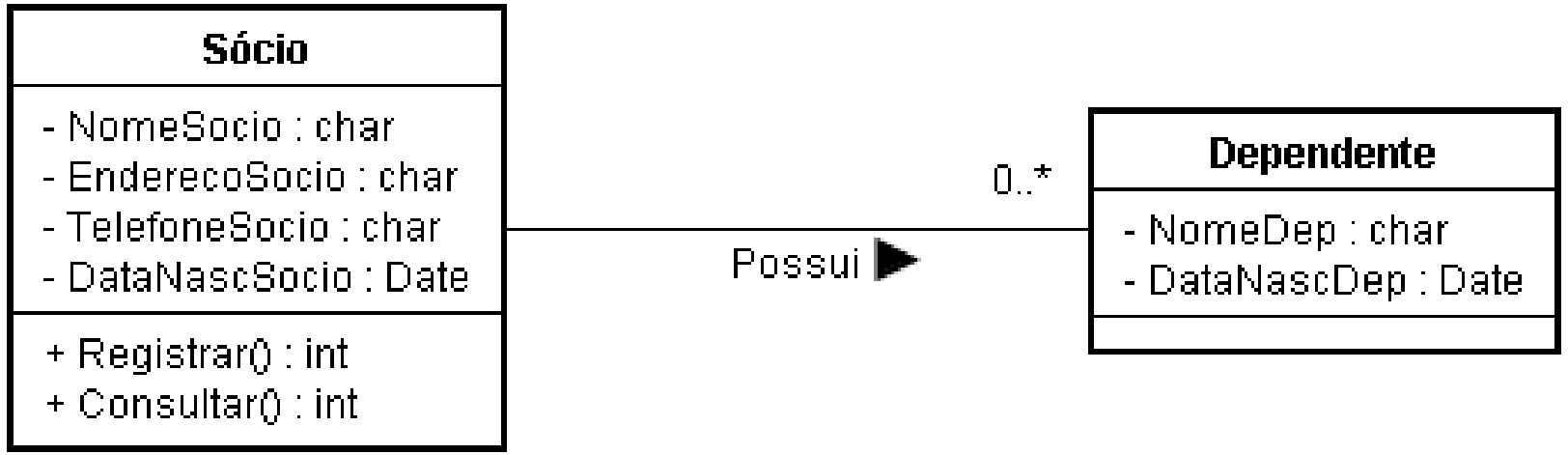


Diagrama de Classes - *Relacionamentos*

- Associação Unária (ou Reflexiva)
  - Ocorre quando uma classe relaciona consigo mesma.
  - Exemplo:

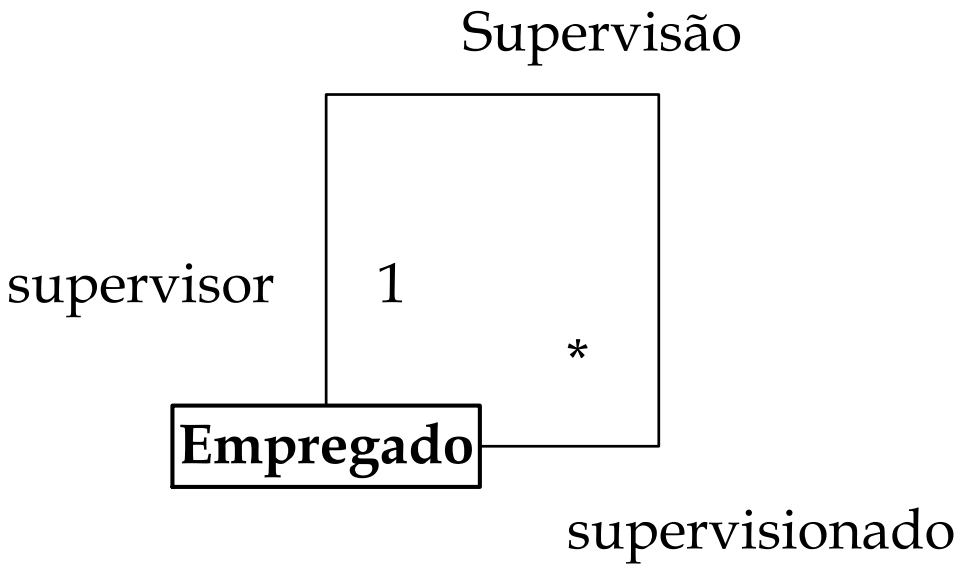


Diagrama de Classes - *Relacionamentos*

- Associação Ternária ou N-ária
  - Associações que conectam mais de duas classes
  - Úteis para demonstrar associações complexas
  - Devem ser evitadas por serem mais difíceis de interpretar
  - São representadas por um **losango** para onde convergem todas as ligações de associação

Diagrama de Classes - *Relacionamentos*

- Associação Ternária ou N-ária
  - Exemplo:

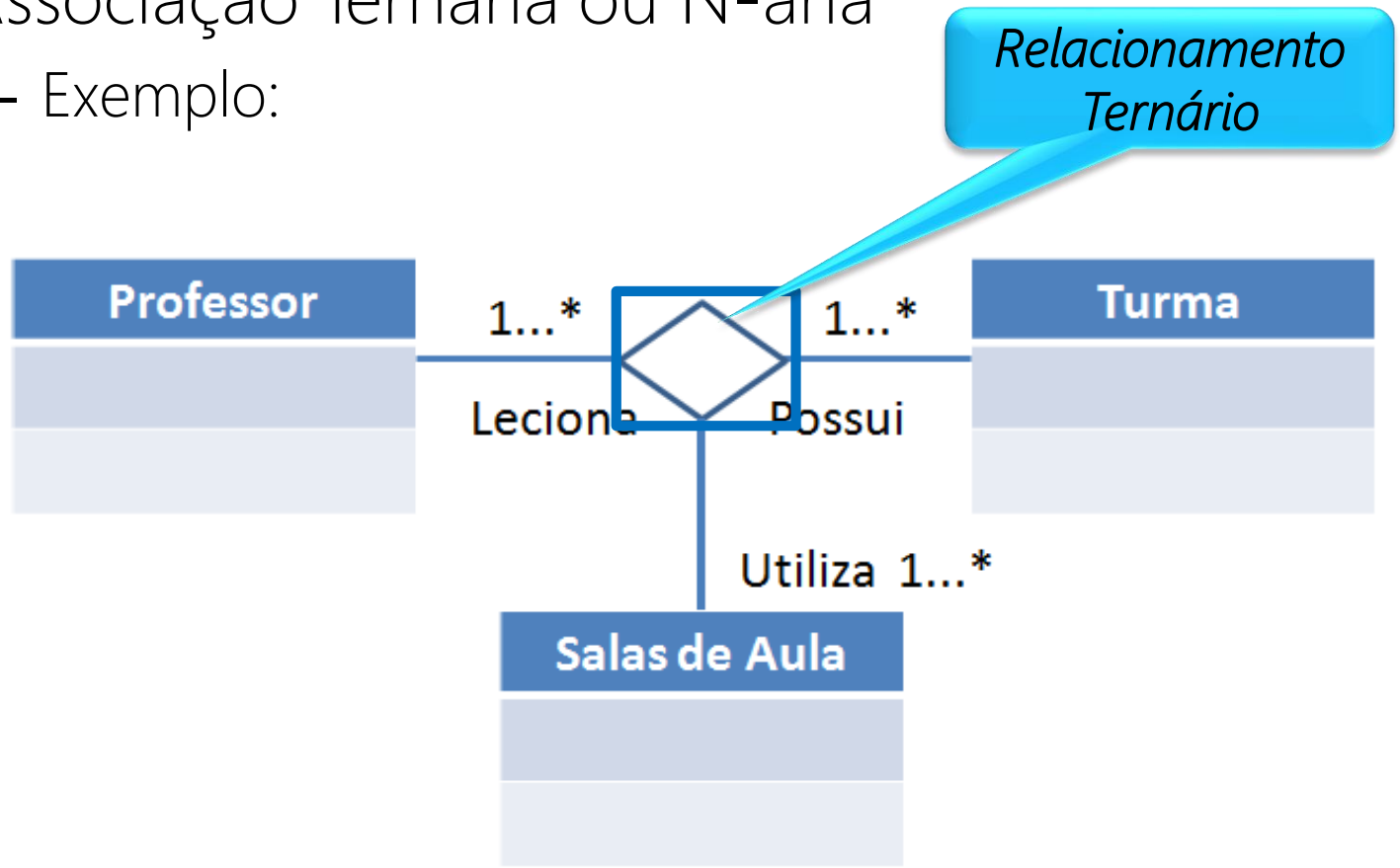


Diagrama de Classes - *Relacionamentos*

- Associação Ternária ou N-ária
  - Exemplo:

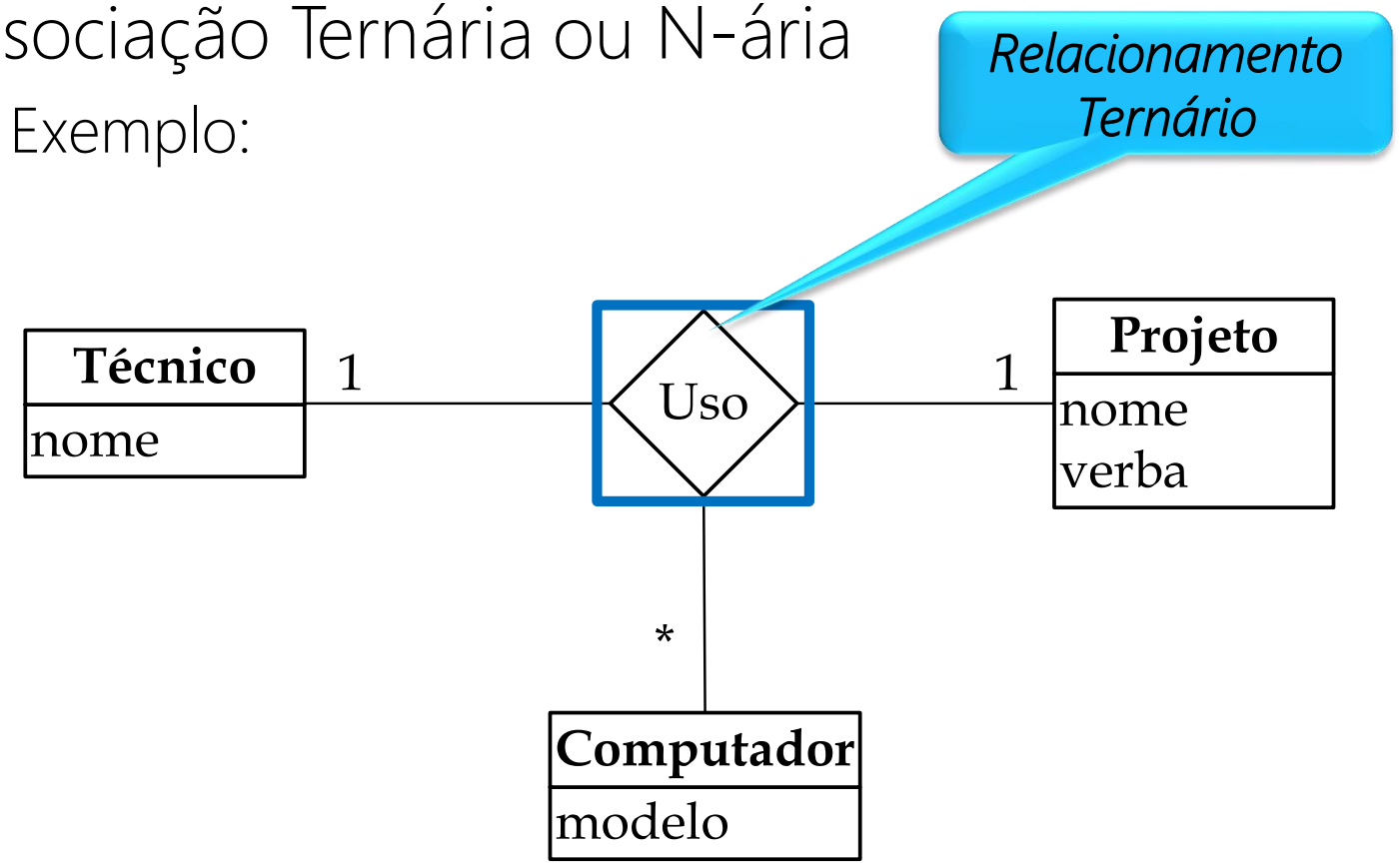


Diagrama de Classes - *Relacionamentos*

- Agregação
  - Qual o significado da palavra Agregação?
    - *Reunião de partes homogêneas formando um todo;*
  - Sinônimos: junção, acumulação, coesão, aglomeração, anexo, conjunto...
  - Ex.: Em física significa *"Porção de moléculas agrupadas"*.

Diagrama de Classes - *Relacionamentos*

- Agregação
  - *É um tipo especial de associação onde tenta-se demonstrar que as informações de um objeto **(objeto-todo)** precisam ser complementadas pelas informações contidas em um ou mais objetos de outra classe **(objeto-parte)***
  - *Em uma agregação, um objeto **está contido** no outro*
  - *Temos a Relação Todo-Parte*

Diagrama de Classes - *Relacionamentos*

- Quando utilizar a Agregação?
  - *Deseja-se modelar um relacionamento **"todo/parte"**, em que uma classe representa uma entidade completa (todo), composta de outras entidades (partes).*

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

Diagrama de Classes - *Relacionamentos*

- **Agregação** – Exemplo

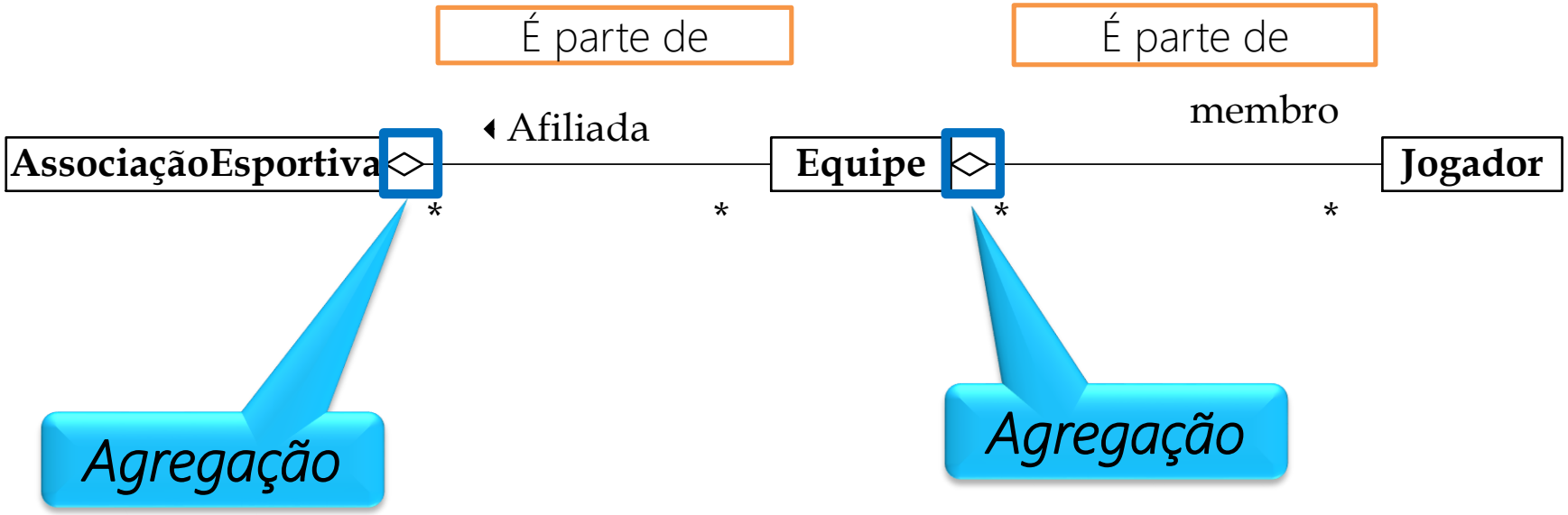


Diagrama de Classes - *Relacionamentos*

- **Agregação** – Exemplo

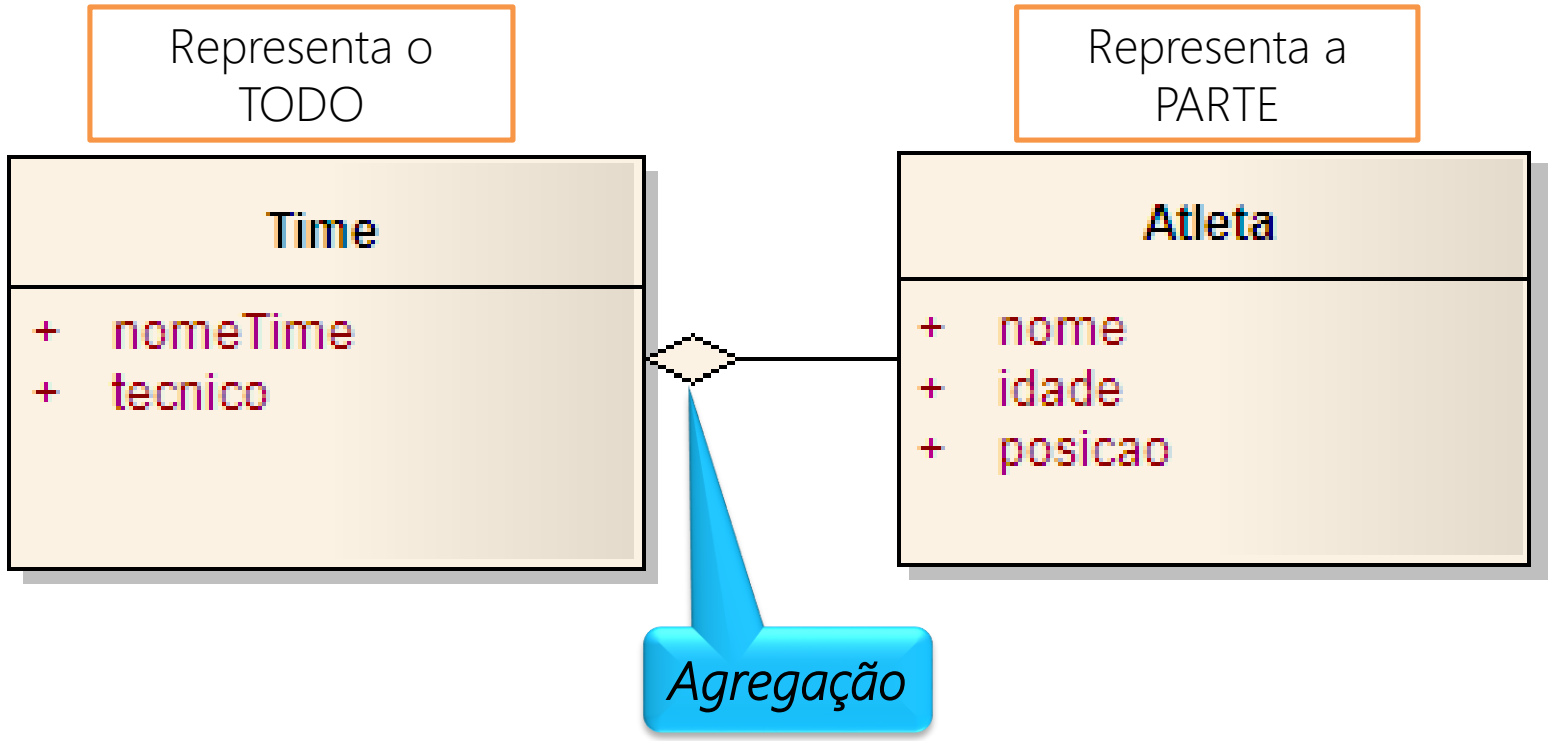


Diagrama de Classes - *Relacionamentos*

- Composição
  - Qual o significado da palavra Composição?
    - *O que constitui ou forma algo. Ação de compor um todo juntando as partes.*
  - Sinônimos: Arranjo, disposição, associação, combinação, constituição, organização, estrutura...
  - Ex.: Composição do sangue, da orquestra, de uma palavra; Exercício de redação escolar.

Diagrama de Classes - *Relacionamentos*

- Composição
  - É uma **forma de agregação** onde há:
    - Vínculo **mais forte** entre Objetos-Todo e Objetos-Parte
    - Objetos-Parte têm de pertencer **exclusivamente** a um Objeto-Todo
    - As partes **não podem existir** sem o todo
  - Ou seja, o relacionamento entre um elemento (o todo) e outros elementos (as partes), onde as partes só podem pertencer ao todo e são criadas e destruídas com ele.
  - Representada como uma linha conectando as classes relacionadas, com um **losango preto** perto da classe que representa o todo.

Diagrama de Classes - *Relacionamentos*

- **Composição** – Exemplo

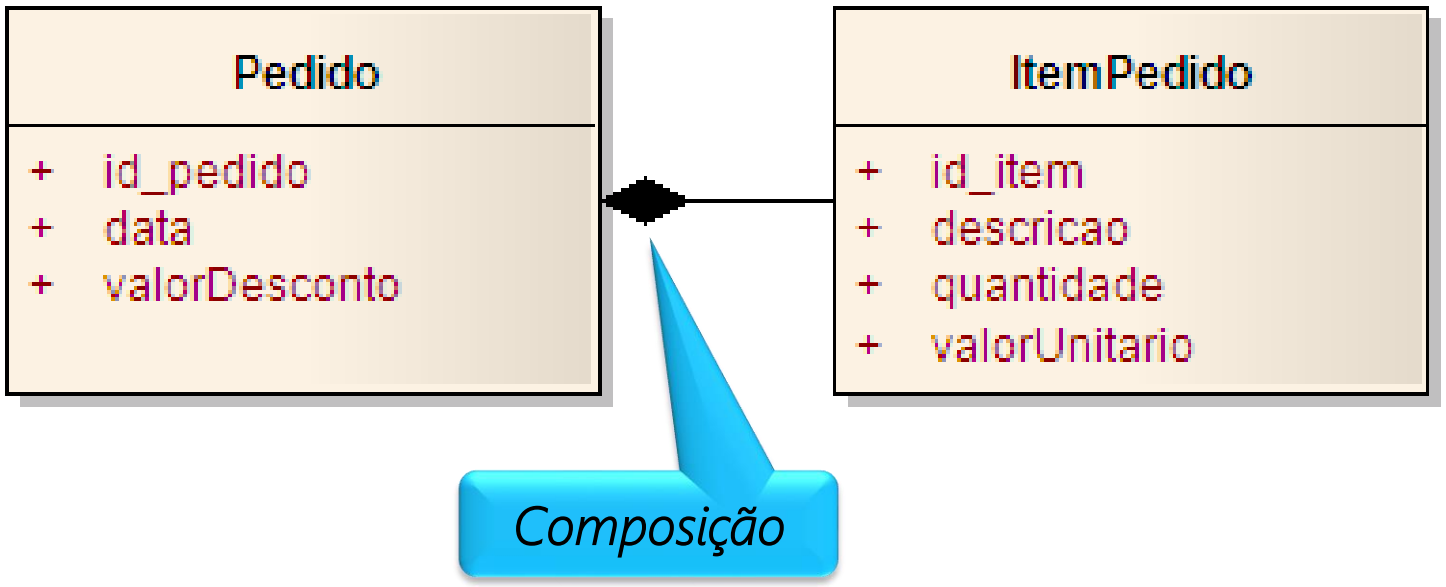


Diagrama de Classes - *Relacionamentos*

- **Composição** – Exemplo

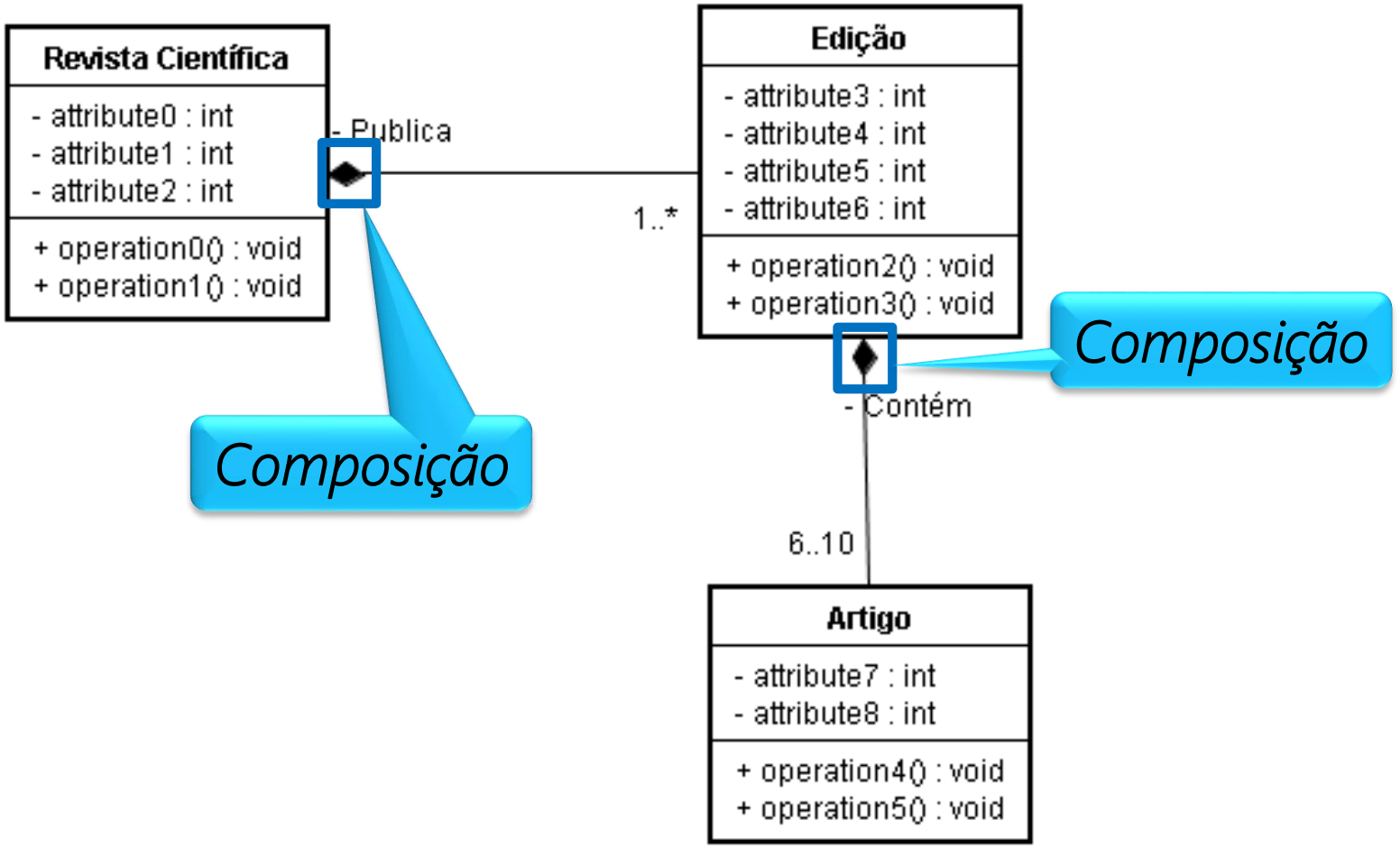


Diagrama de Classes - *Relacionamentos*

- Associação com propriedades
  - Em uma associação entre classes, a própria associação pode ter propriedades.
  - O conjunto destas propriedades formam uma **Classe Associativa.**

Diagrama de Classes - *Relacionamentos*

- Associação com propriedades
  - Em uma associação entre classes, a própria associação pode ter propriedades.
  - O conjunto destas propriedades formam uma **Classe Associativa.**

- Classe Associativa
  - Produzida quando da ocorrência de associações que possuem multiplicidade muitos (*) em todas as suas extremidades.
  - Classe para armazenar os atributos transmitidos pela associação.

Diagrama de Classes - *Relacionamentos*

- **Classe Associativa** – Exemplo

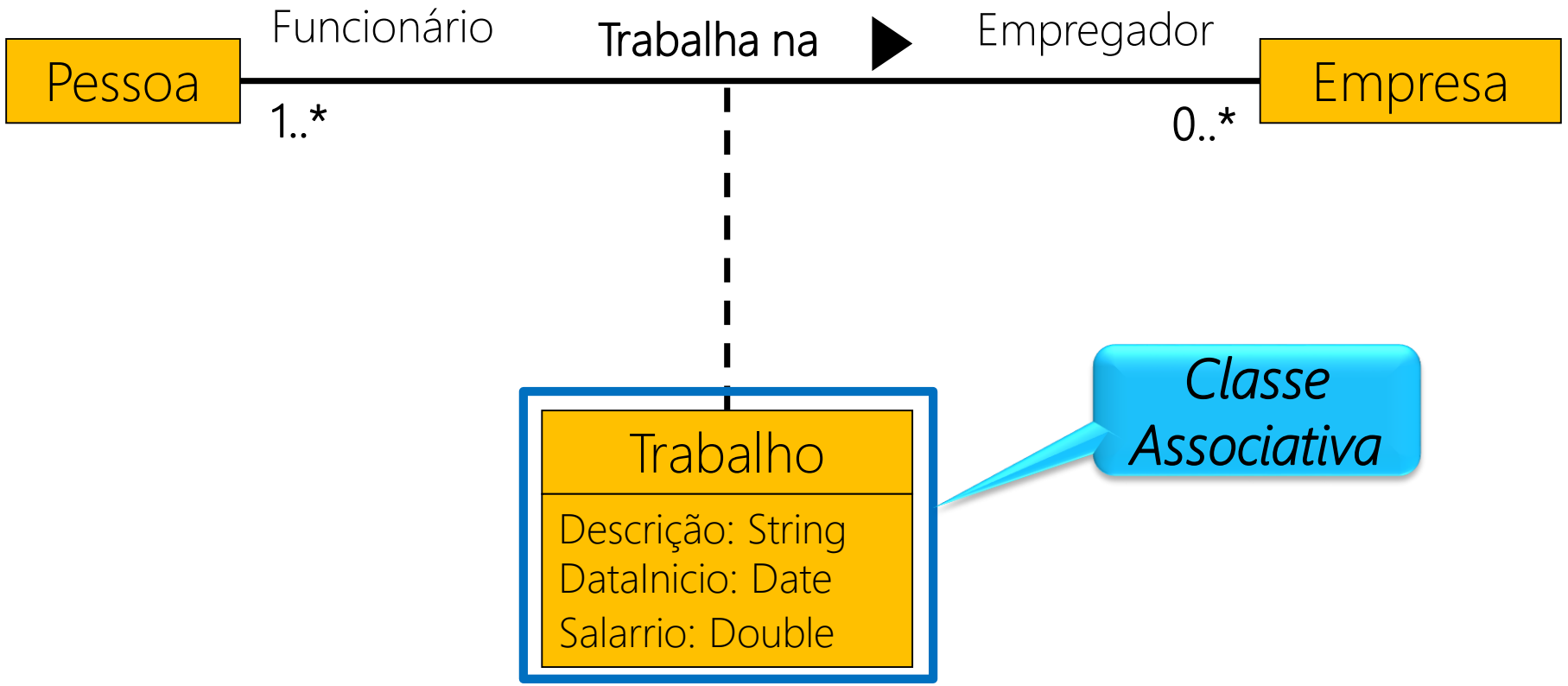


Diagrama de Classes - *Interfaces*

- Uma **Interface** é formada por declarações de métodos desprovidos de implementação.
    - Ou seja, apenas assinatura e sem o corpo dos métodos.

- Uma **Interface** pode ser vista como um contrato estabelecido entre a classe e a interface.

- Interfaces contêm apenas métodos públicos abstratos.

Diagrama de Classes - *Relacionamentos*

- Interfaces – notação

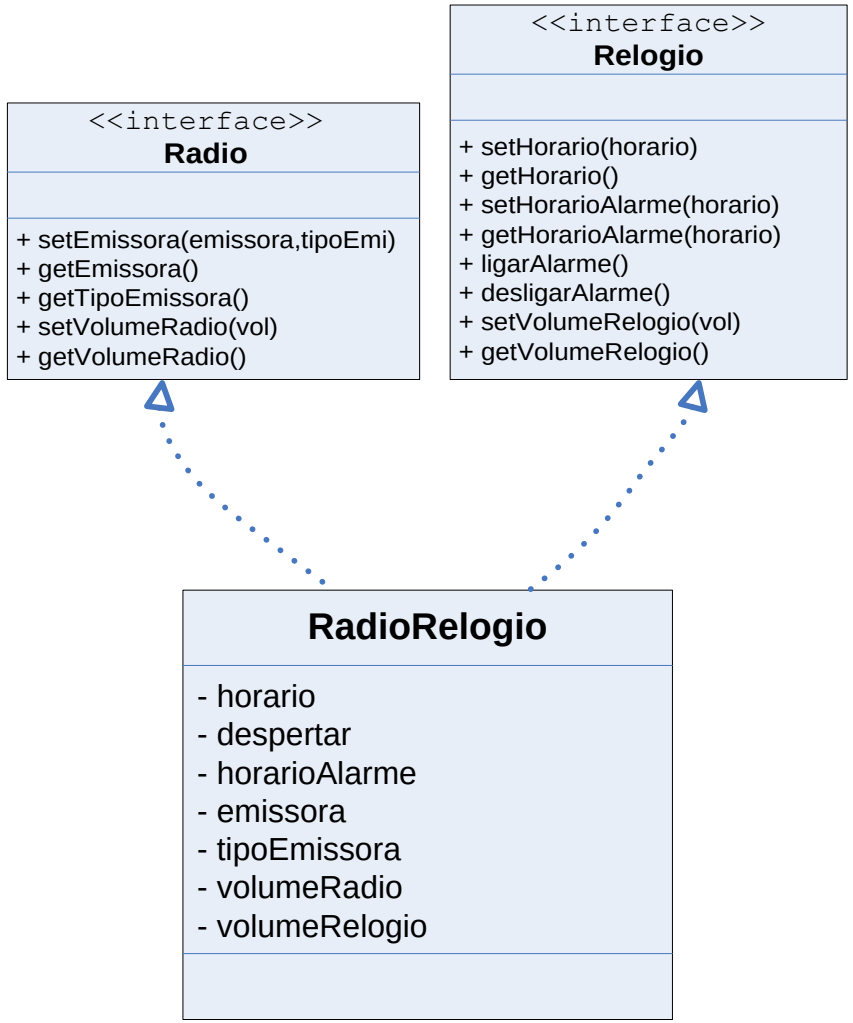


Diagrama de Classes - *Relacionamentos*

- Interfaces – notação

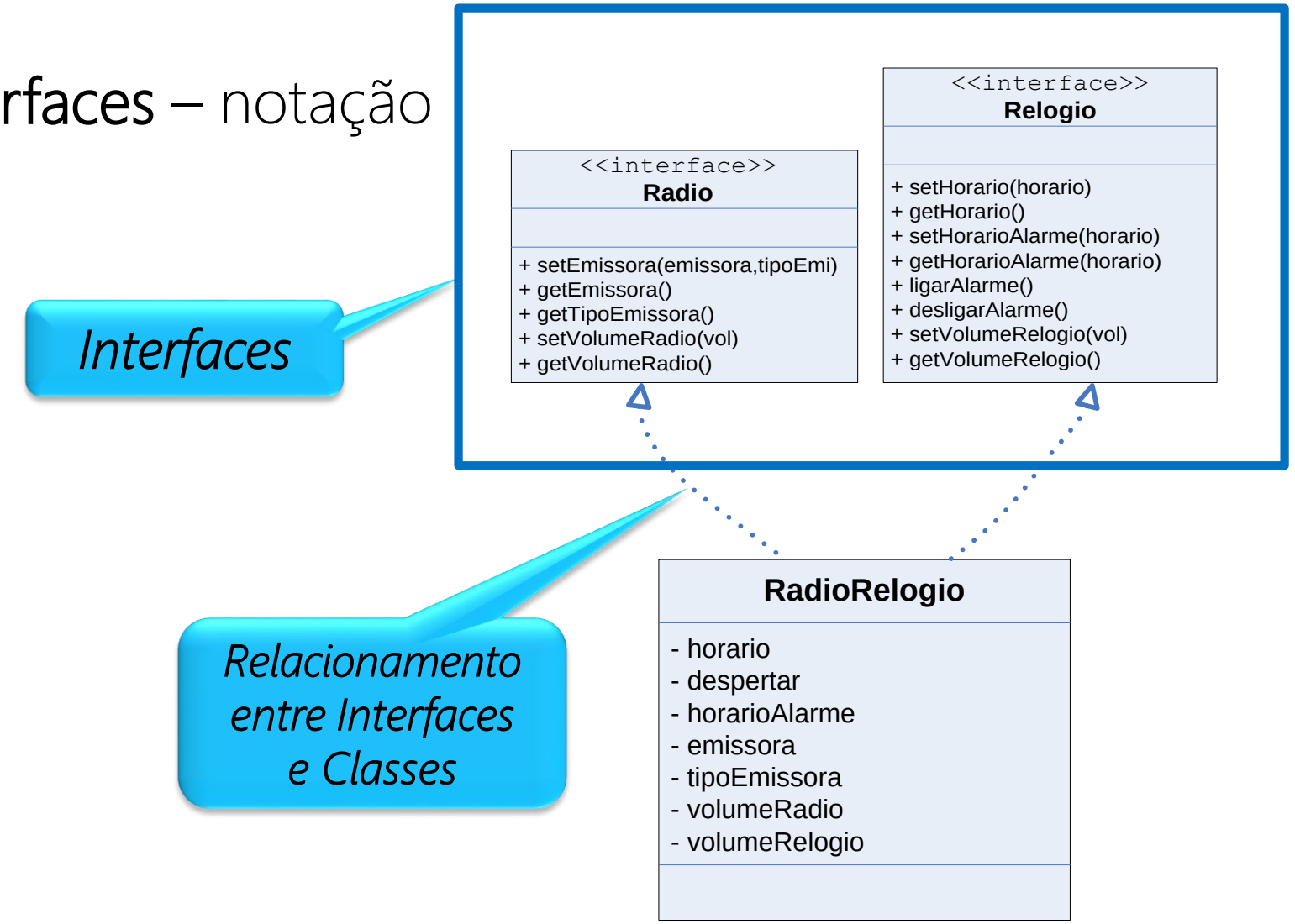
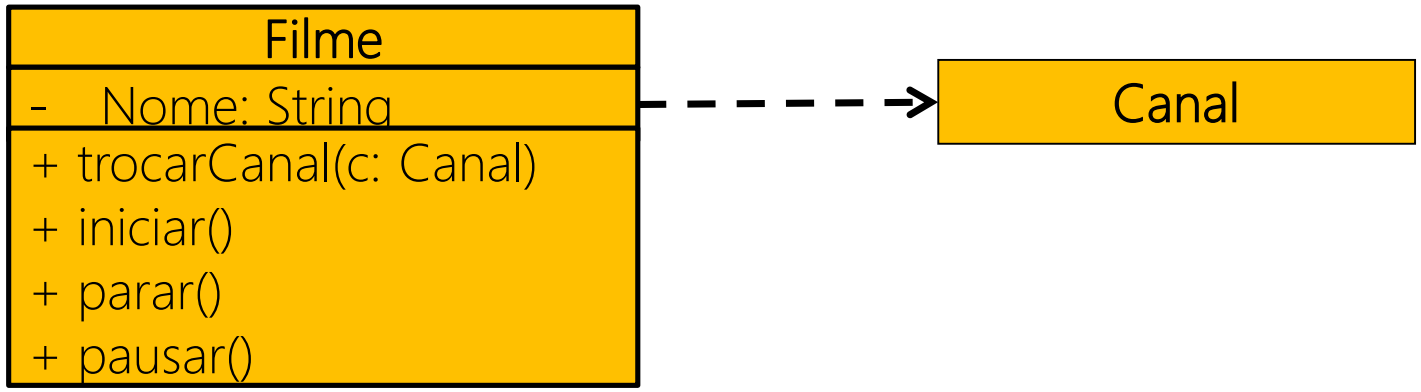


Diagrama de Classes - *Relacionamentos*

- Dependência
  - Demonstra certo grau de dependência de uma classe a outra
  - A dependência entre classes indica que os objetos de uma classe **usam operações**, **atributos**, **variáveis** ou **argumentos** dos objetos de outra classe.
  - Mudança numa classe deverá refletir na outra.

Diagrama de Classes - *Relacionamentos*

- Dependência
  - Por exemplo: a classe **Filme** é uma classe que depende da classe **Canal** para executar suas operações. Mudanças na classe **Canal** pode afetar a classe **Filme**. Assim, dizemos que a classe **Filme** é dependente da classe **Canal**.



Exercício: Analise e elabore as classes relacionadas ao exemplo abaixo

- Uma determinada loja de roupas solicitou que fosse feito um sistema para gerenciar seus produtos. Algumas informações foram previamente fornecidas:
  - *"Preciso registrar informações dos meus clientes, como nome e telefone";*
  - *"Preciso registrar meus produtos para consultar posteriormente. Eles também precisam estar organizados em categorias considerando que faço vendas de camisas, camisetas, bermudas e calças";*
  - *"Preciso registrar os pedidos dos meus clientes e saber o valor total em produtos em cada pedido";*