



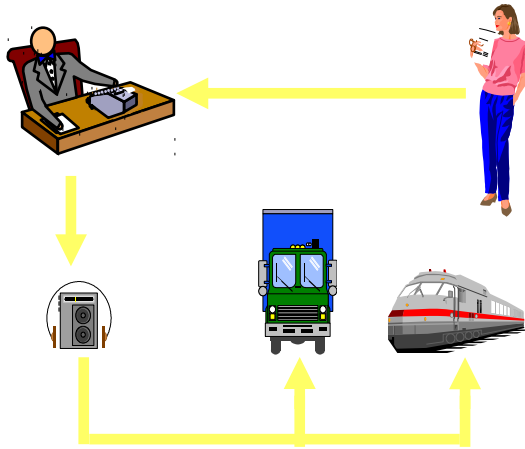
**UML**

**Diagrama de Classes**



# Modelagem

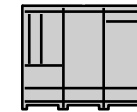
*Modelagem é a utilização das notações gráficas e textuais para representação dos requisitos do sistema*



**Modelagem de sistemas independentes de linguagem de implementação**

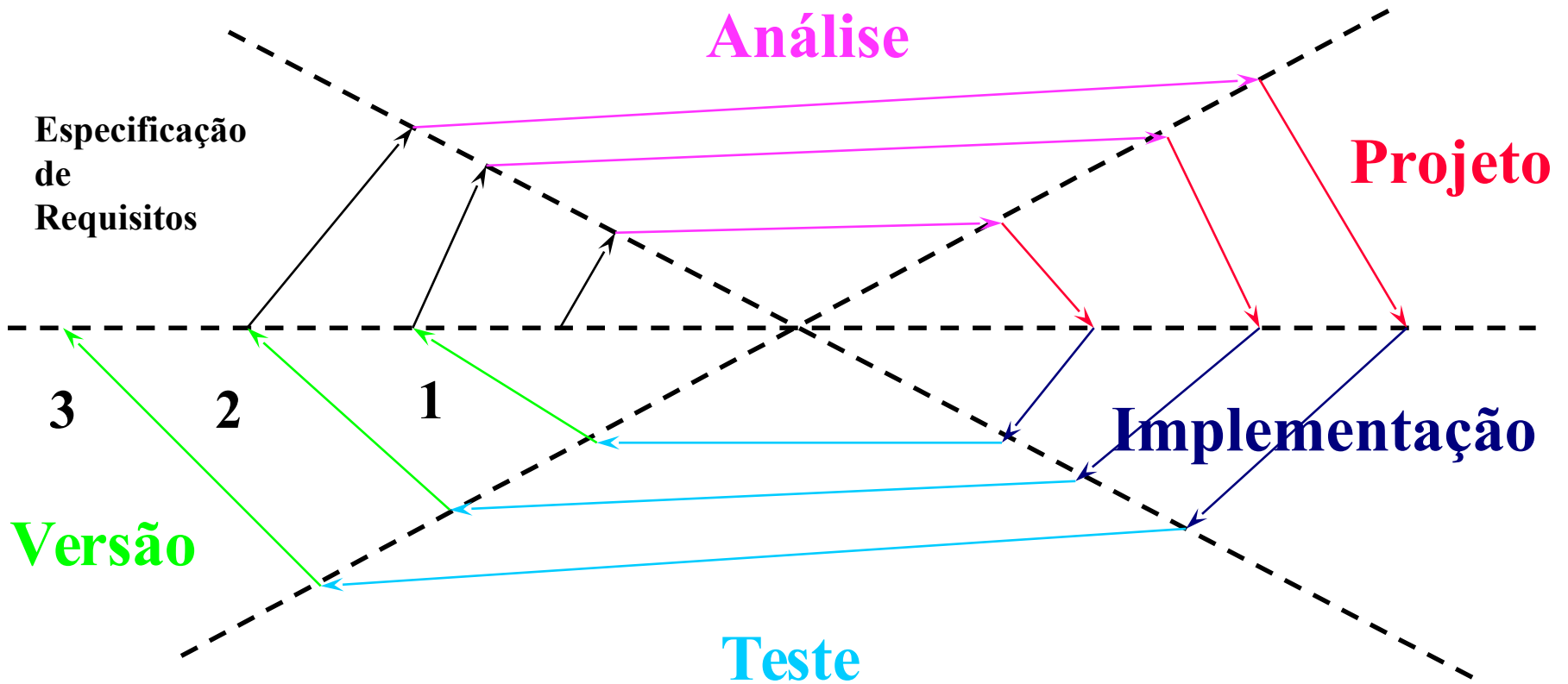


**Análise é a técnica de capturar os processos de negócios das perspectivas dos usuários.**



**Sistemas Computacionais Heterogêneos**

# Processo de Desenvolvimento de Software OO



**Modelo Espiral**

# UML- Unified Modeling Language

- *Visão Lógica*

- ⇒ Diagrama de Classes

- ⇒ Diagrama de estados e Diagrama de atividade

- *Visão de Caso de Uso*

- ⇒ Diagrama de Caso de uso

- ⇒ Diagrama de Sequência

- ⇒ Diagrama de Colaboração

- *Visão de Componentes*

- ⇒ Diagrama de Componentes

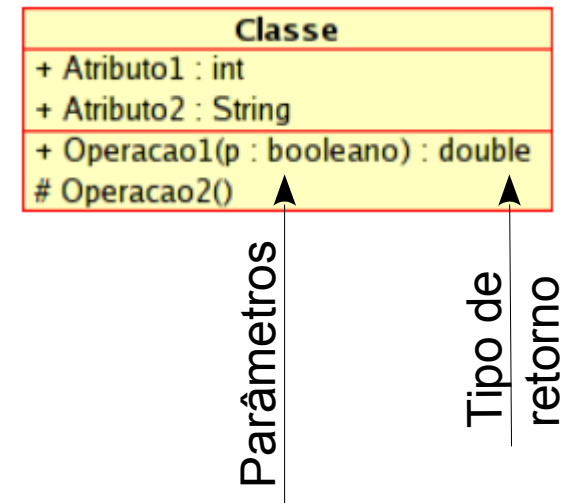
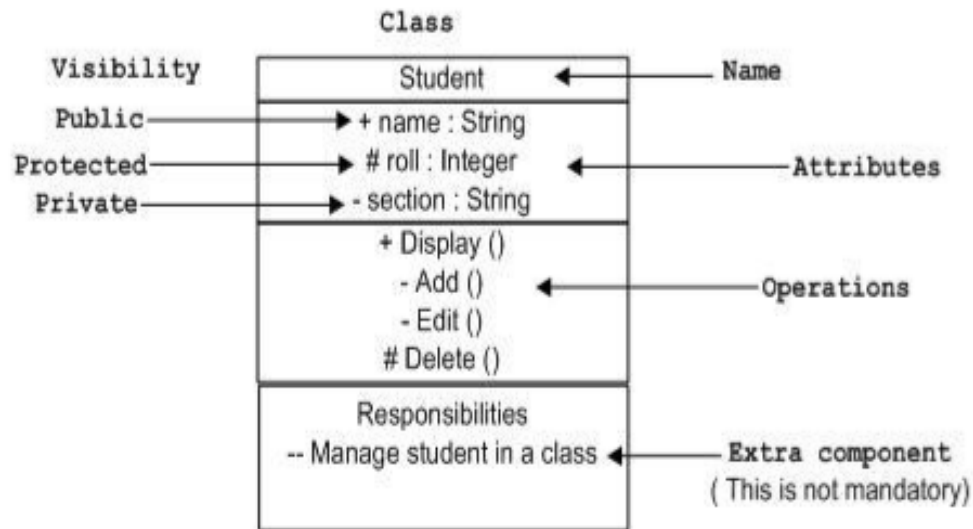
- *Visão de Implementação*

# UML – Diagrama de Classes

## Class Notation:

UML *class* is represented by the diagram shown below. The diagram is divided into four parts.

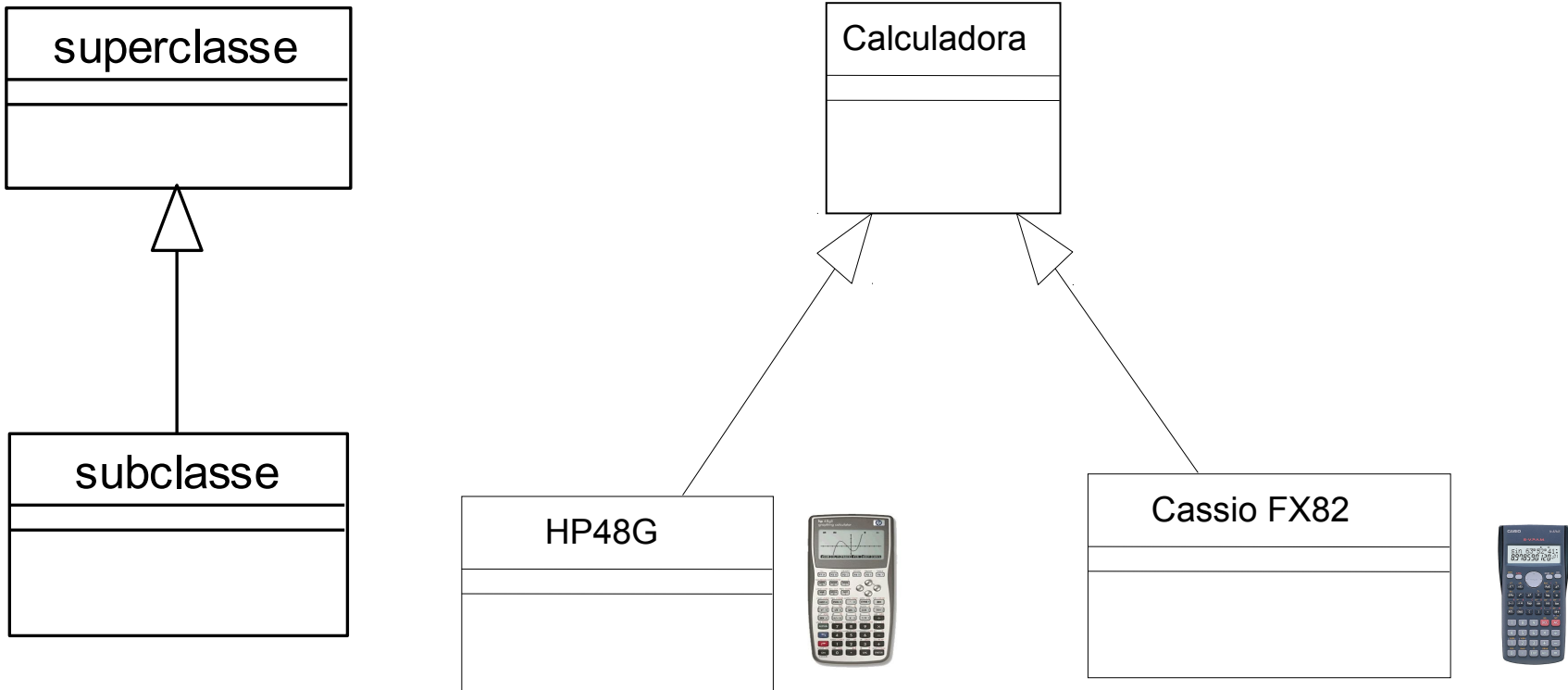
- The top section is used to name the class.
- The second one is used to show the attributes of the class.
- The third section is used to describe the operations performed by the class.
- The fourth section is optional to show any additional components.



Classes are used to represent objects. Objects can be anything having properties and responsibility.

# UML – Diagrama de Classes

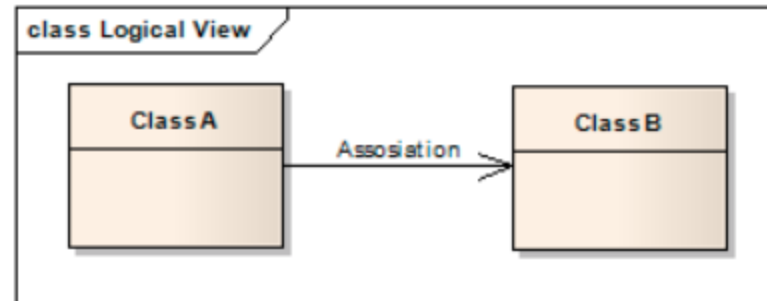
- **Relacionamentos entre classes:**
  - **Herança: generalização e especialização**



# UML – Diagrama de Classes

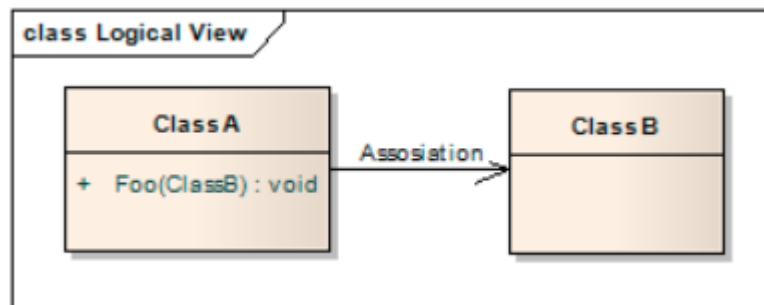
## Associação, agregação e composição

- A maneira mais abstrata disponível para descrever uma relação entre classes é através de uma associação, que simplesmente indica que uma classe depende de outra de alguma forma.



# UML – Associação, agregação e composição

- Associação fraca: A classe A pode estar ligada à classe B de modo que um ou mais métodos incluem objetos do tipo Classe B ou retornam objetos do tipo Classe B.

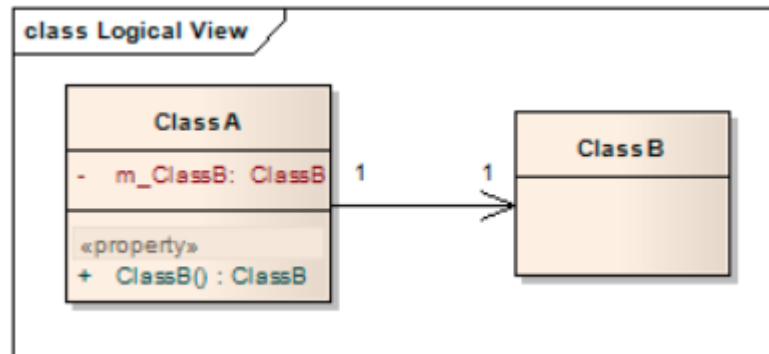




# UML – Associação, agregação e composição

- Associação forte: A Classe A possui um ou mais atributos do tipo Classe B

**Cardinalidade:**



0..1

1

\*

0..\*

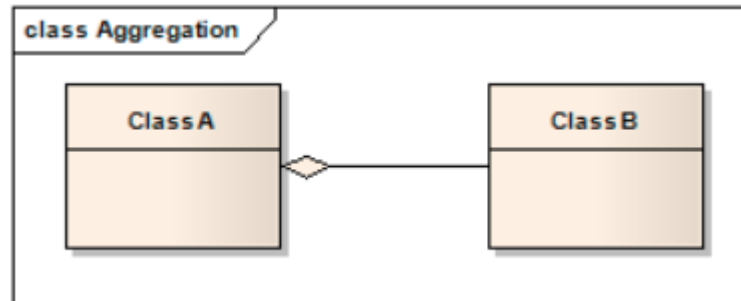
1..\*

1..6

19..\*

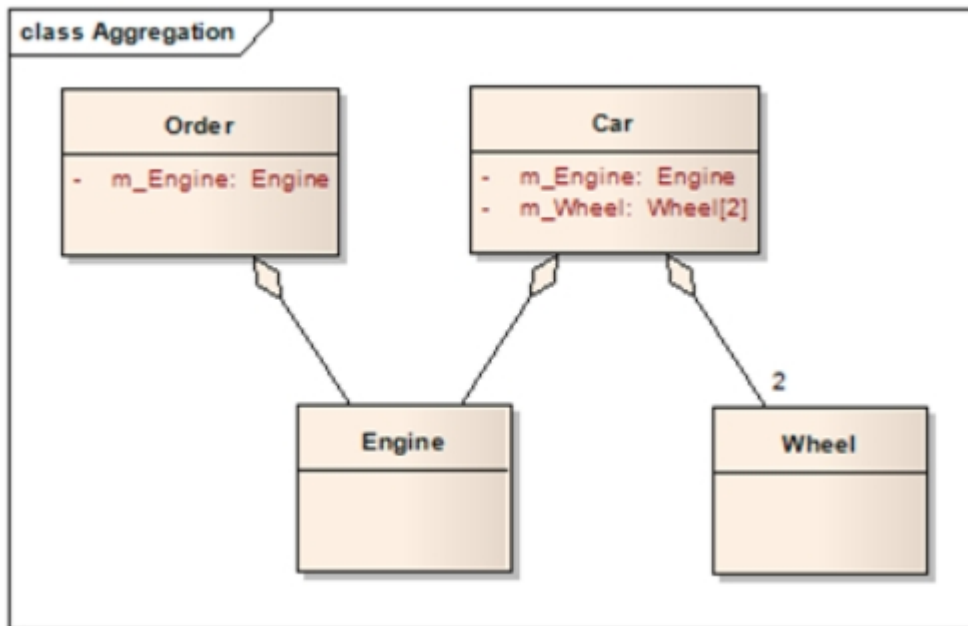
# UML – Associação, agregação e composição

- Agregação: Em casos onde existe um relação de “é parte de” entre a Classe A (todo) e a classe B (parte), é possível utilizar o link mais específico de agregação.



# UML – Associação, agregação e composição

## ■ Agregação. Exemplo:



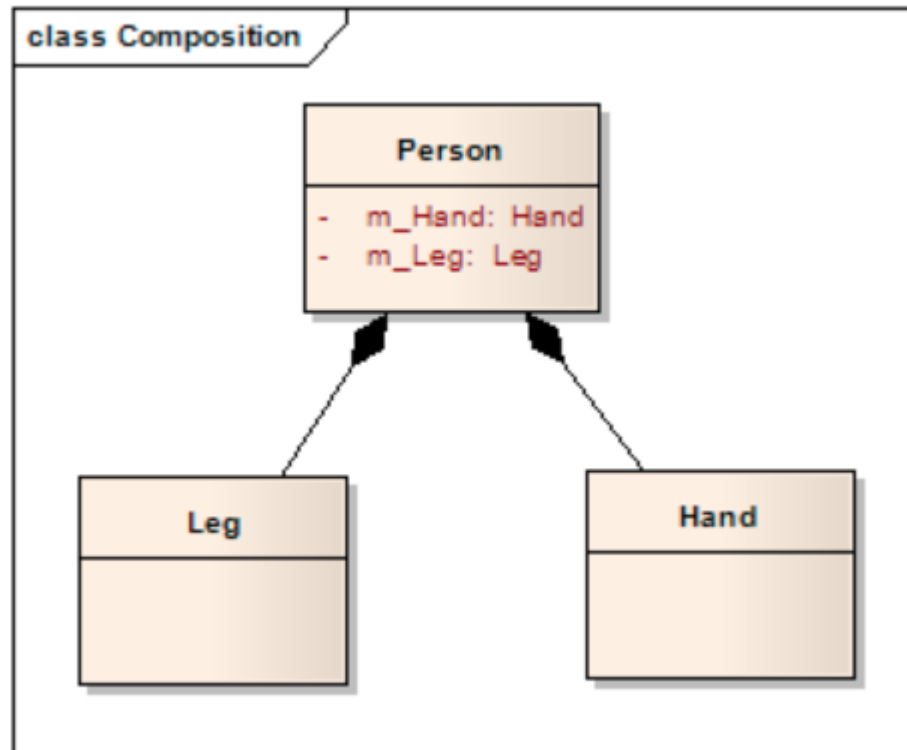
- A agregação é uma de associação forte.
- Uma Classe agregada pode ser utilizada em outra parte do sistema

## UML – Associação, agregação e composição

- Composição: nos casos onde em adição ao conceito de relação “parte de” entre as Classes A e B existir uma forte relação de existência entre as classes, significando que quando a Classe A é apagada a Classe B também deve ser apagada, é possível ser mais específico e utilizar a relação de composição.

# UML – Associação, agregação e composição

## ■ Composição: Exemplo

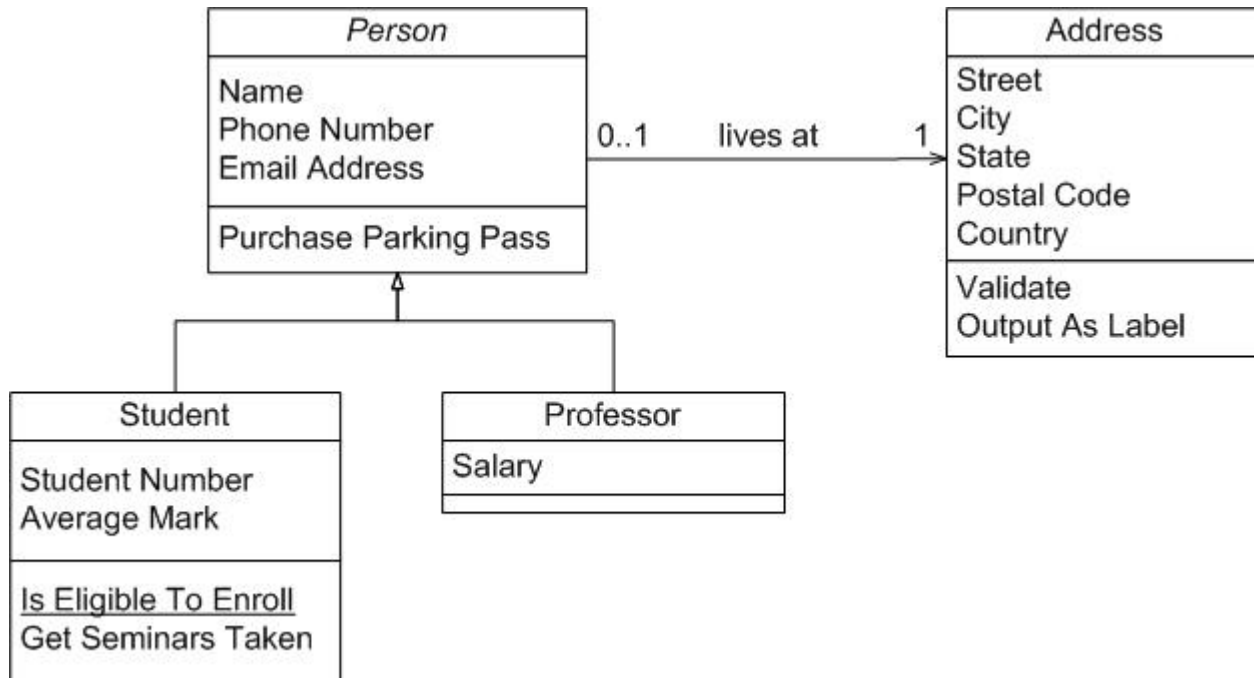


# UML – Associação, agregação e composição

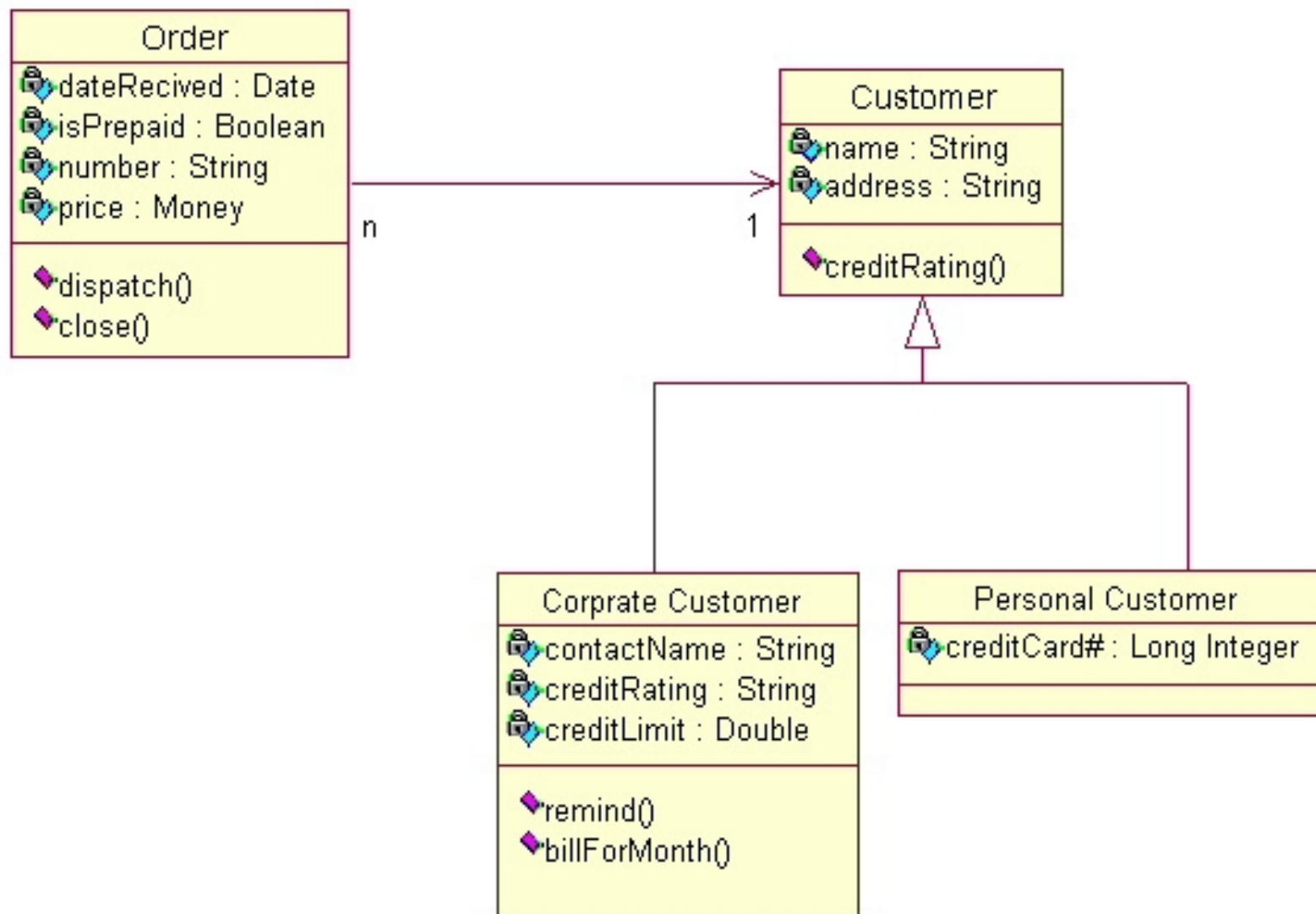
- Composição:
  - A relação de composição indica que o container (todo) possui posse exclusiva da outra classe;
  - Se uma classe compõe outra em uma agregação, ela não pode ser utilizada em outra parte do sistema.

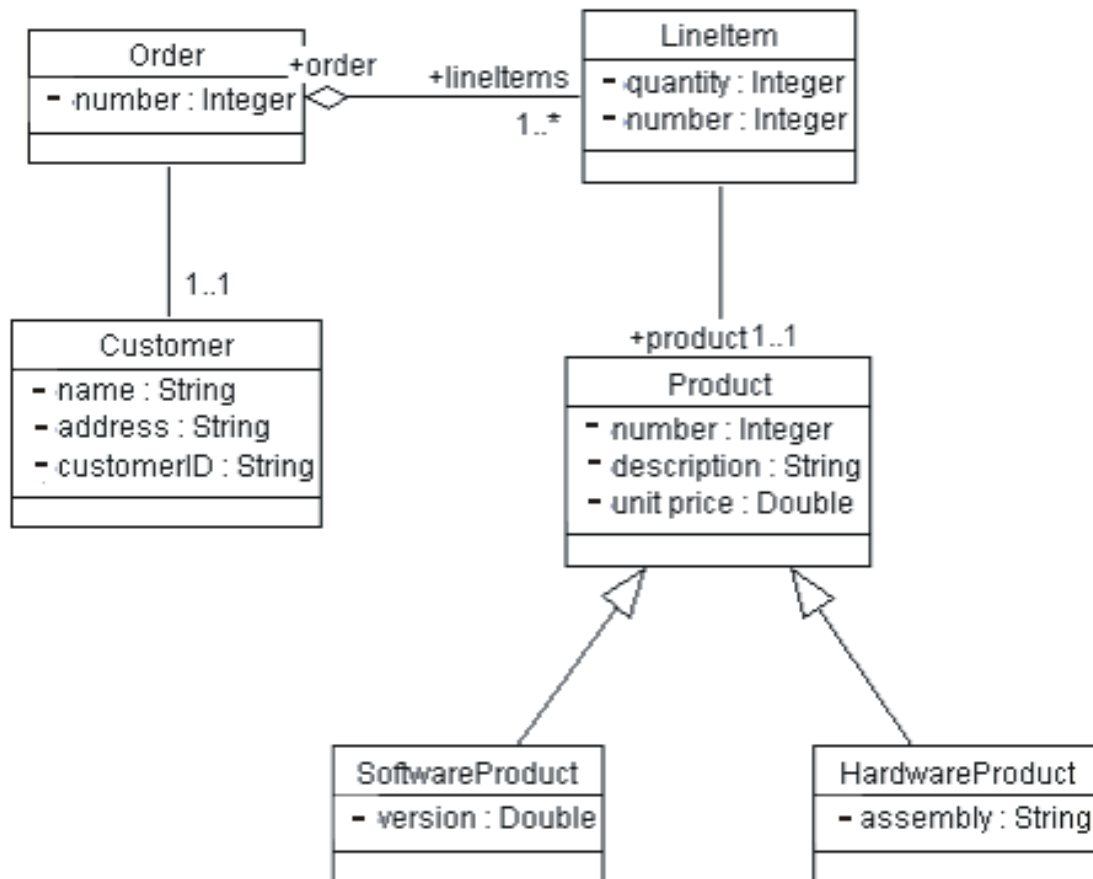
# UML – Associação, agregação e composição

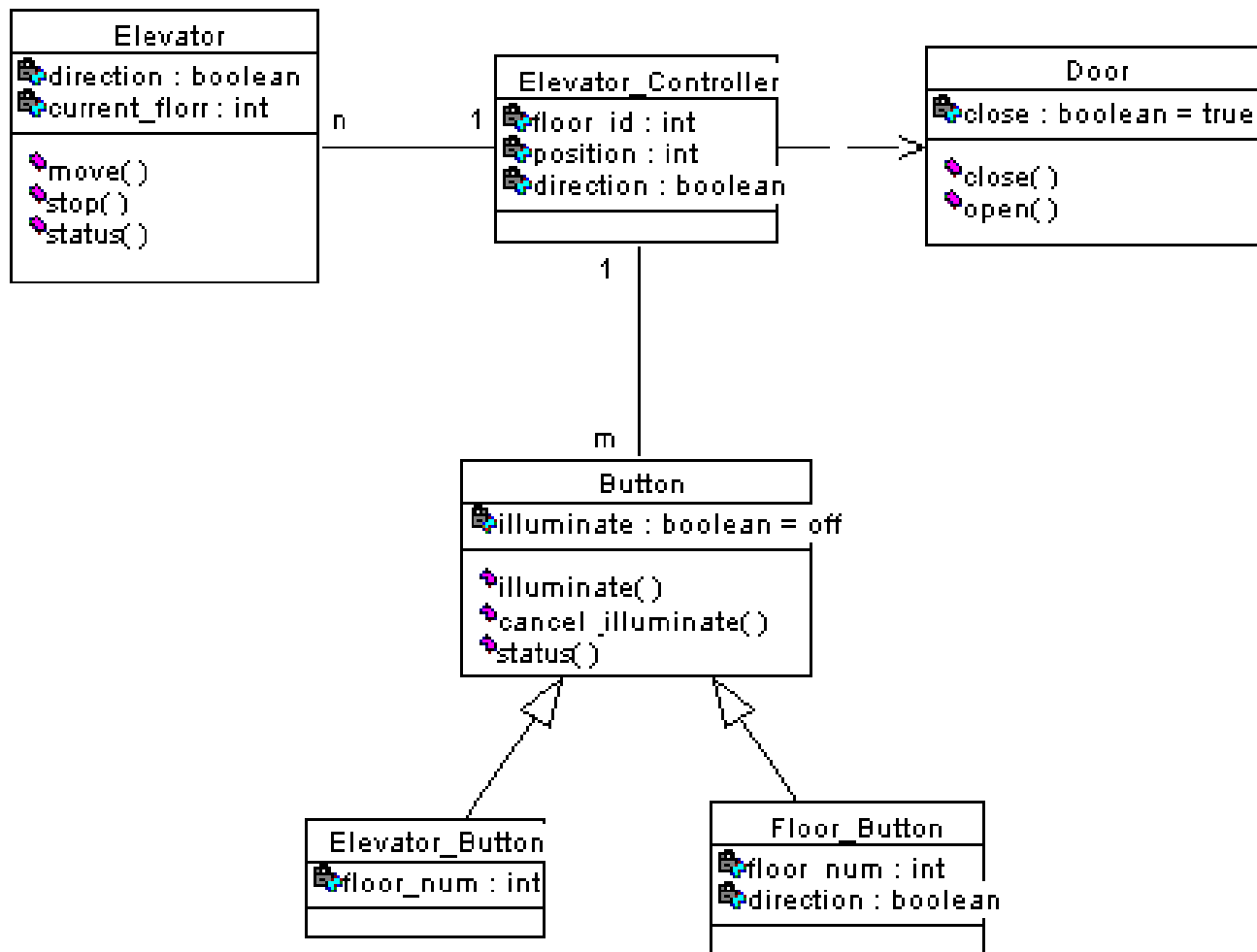
- Composição:
  - A relação de composição indica que o *container* (todo) tem posse exclusiva da outra classe;
  - Se uma classe compõe outra em uma agregação, ela não pode ser utilizada em outra parte do sistema.

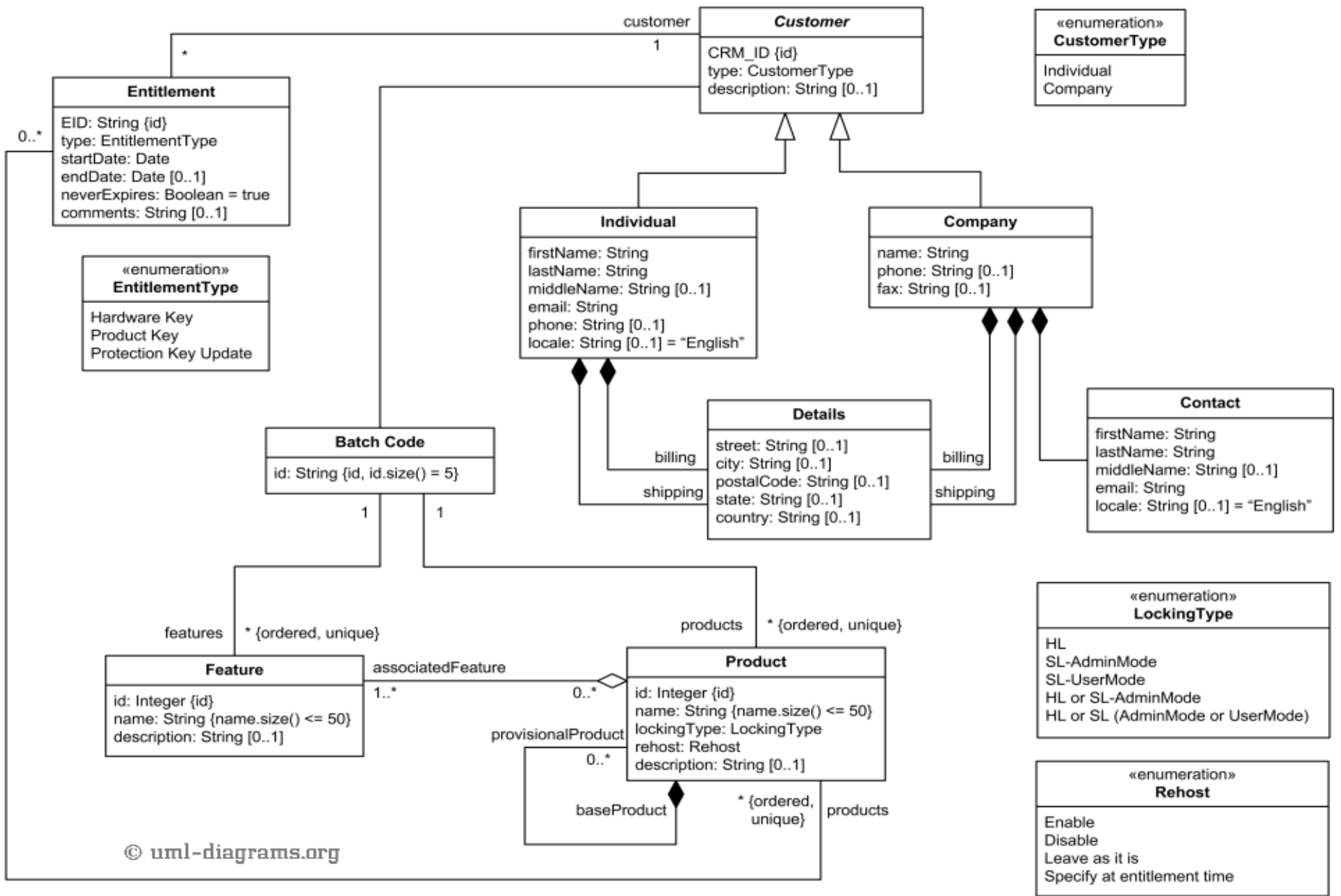


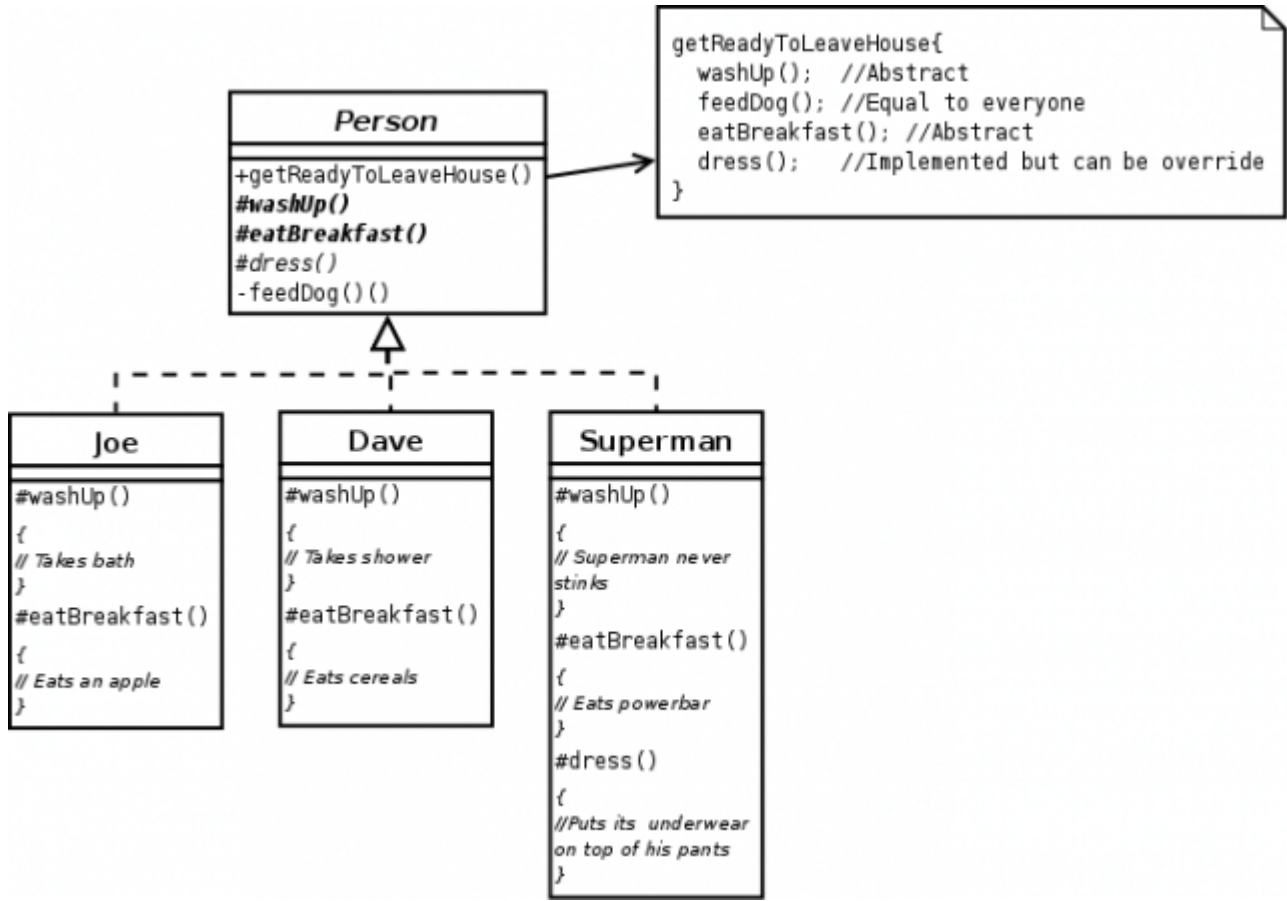












Atenção: isto é uma brincadeira!