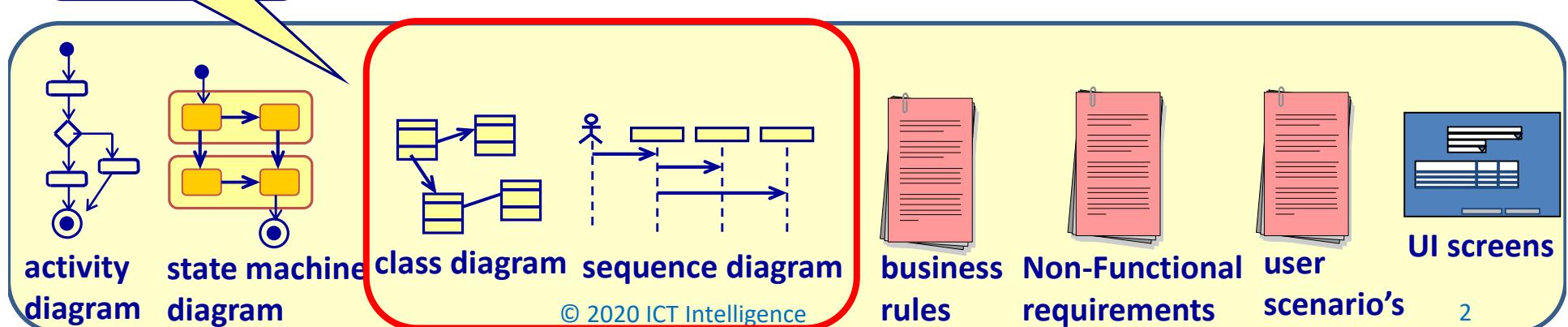
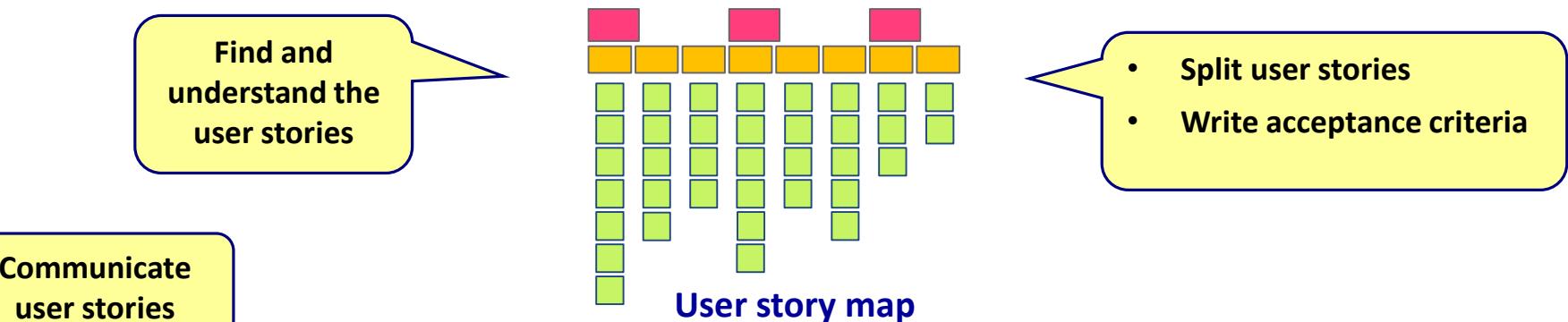
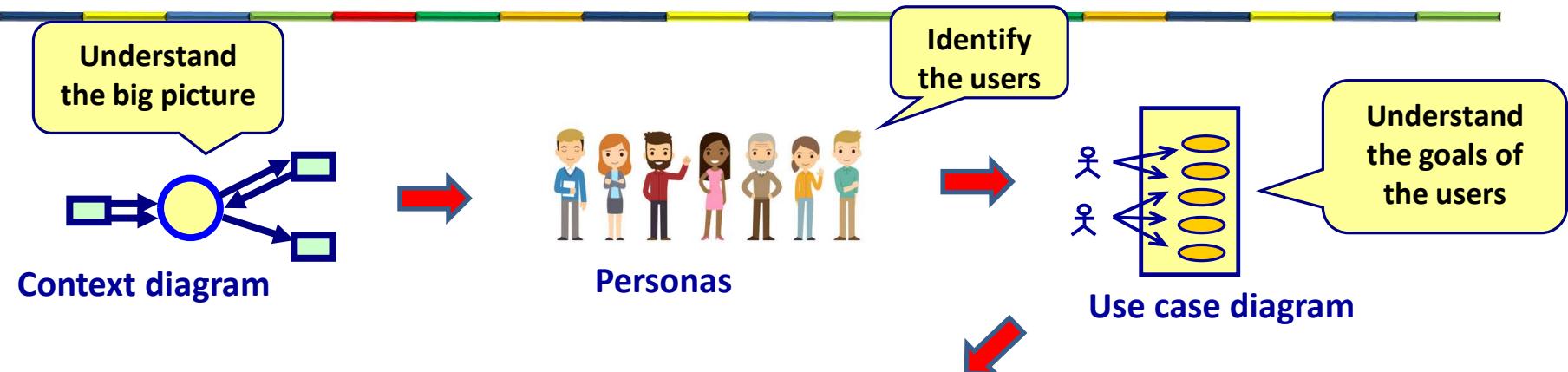
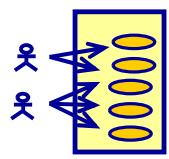
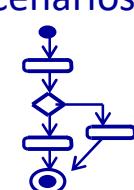
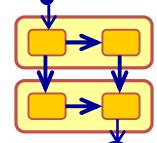
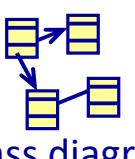
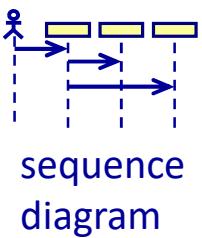
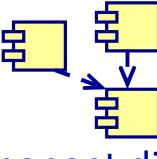
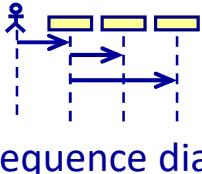
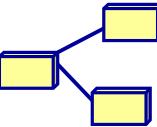
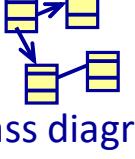
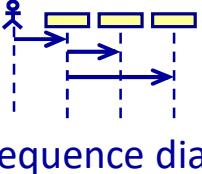


# Lesson 6

## Analysis

# Agile requirements



Requirements	Analysis	Architecture	Design
 <b>use case diagram</b>  <b>scenarios</b>  <b>activity diagram</b>  <b>state diagram</b> <div style="border: 1px solid blue; padding: 5px; border-radius: 10px;">           Black box/ grey box         </div>	 <b>class diagram</b>  <b>sequence diagram</b> <div style="border: 1px solid blue; padding: 5px; border-radius: 10px;">           White box Not technical         </div>	 <b>component diagram</b>  <b>sequence diagram</b>  <b>deployment diagram</b> <div style="border: 1px solid blue; padding: 5px; border-radius: 10px;">           White box Technical         </div>	 <b>class diagram</b>  <b>sequence diagram</b> <div style="border: 1px solid blue; padding: 5px; border-radius: 10px;">           White box Technical         </div>

# Analysis

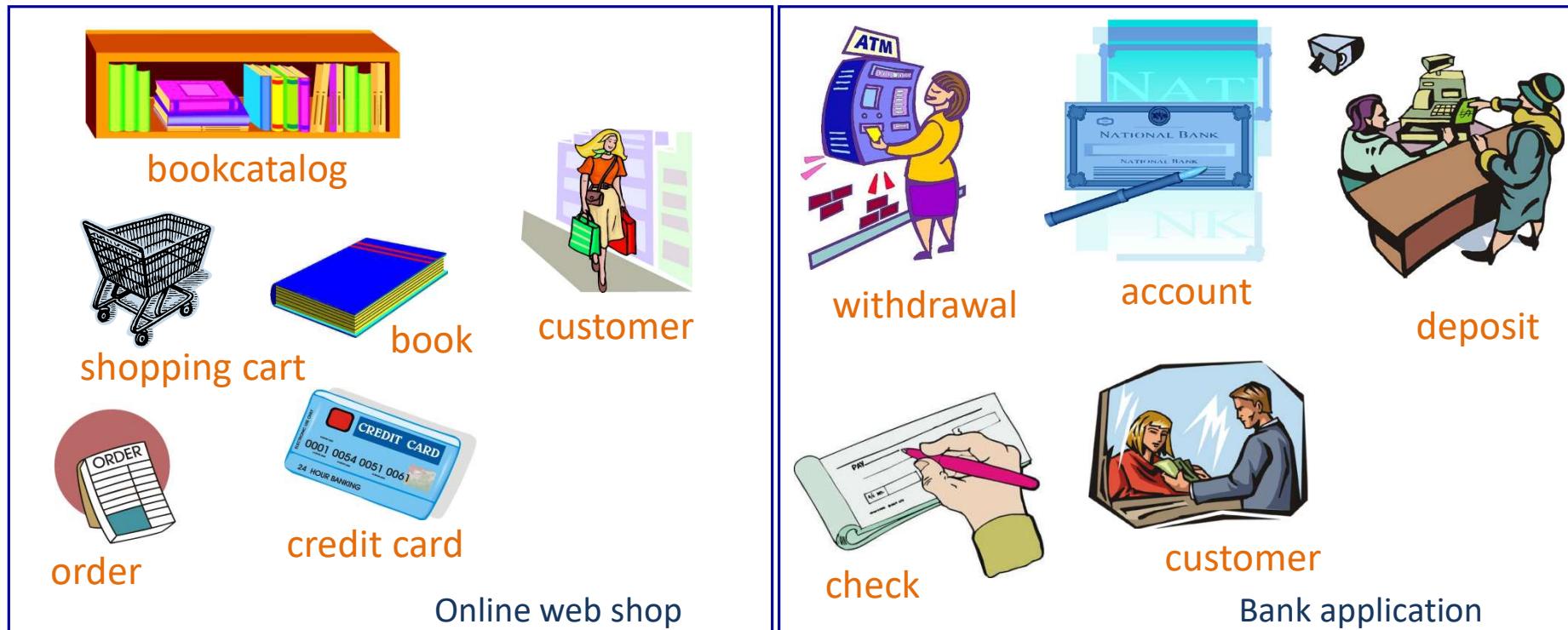
# Analysis model

---

- Structure of the domain model
  - Objects: class diagram
- Behavior of the domain model
  - Sequence diagram
  - Communication diagram (not much used)

# Entity objects in the problem domain

---



# OO concepts

---

- Object
- Behavior, state and identity
- Relations
- Class
- Inheritance
- Polymorphism

# Object

---

- An object is a concept or a thing in the problem domain



Customer



Destination



Date



Luggage



Creditcard



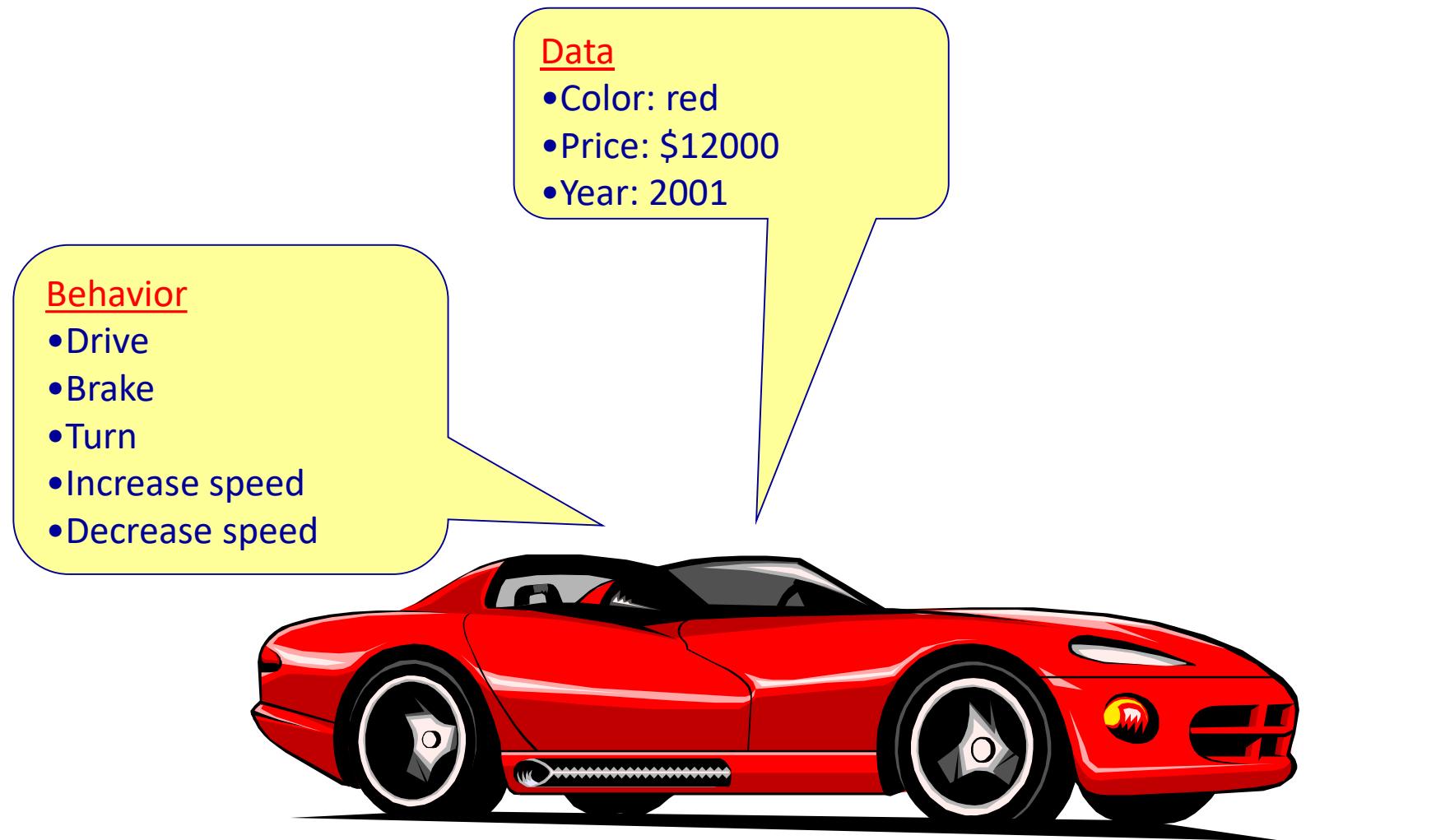
Flight



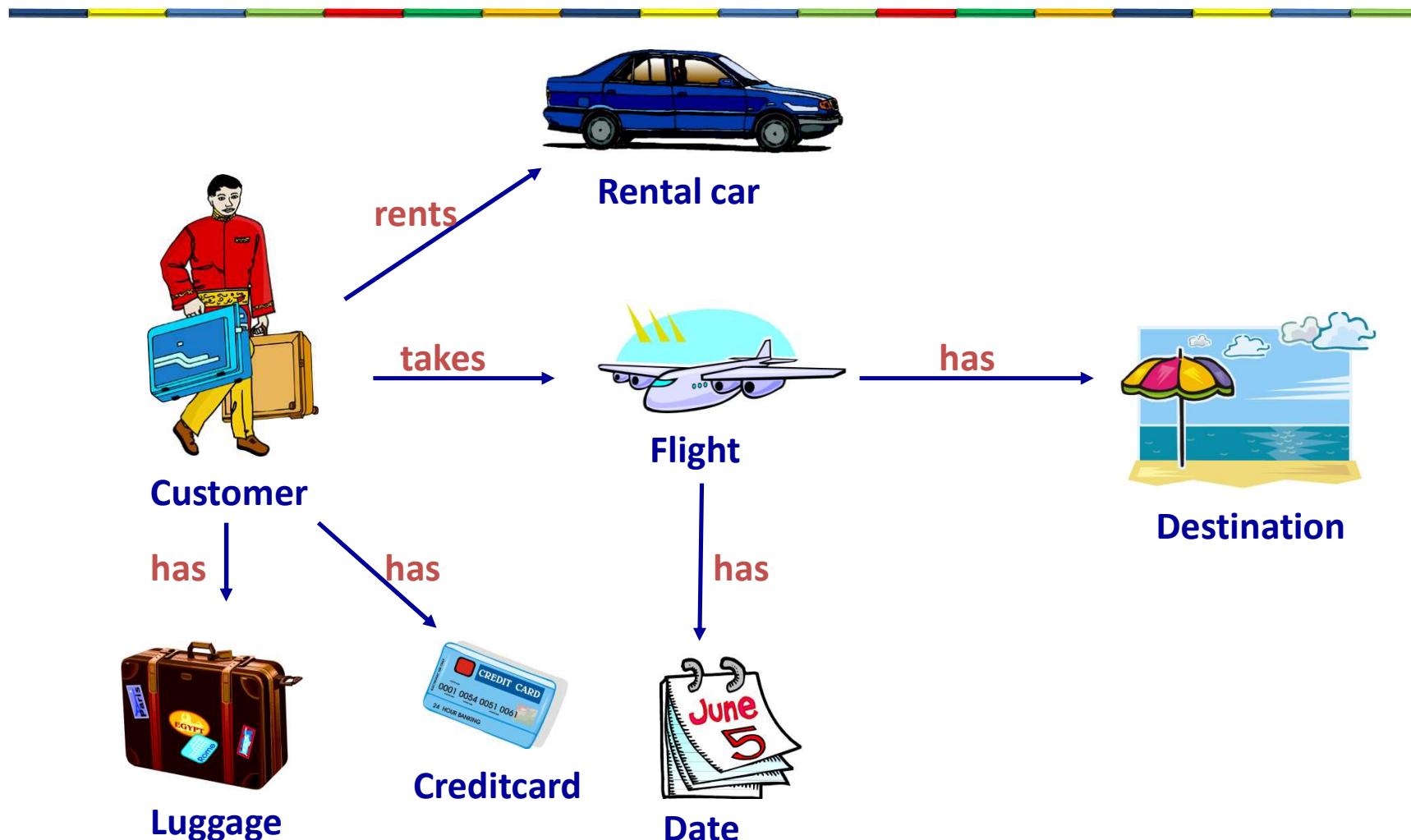
Rental car

# An object has

---

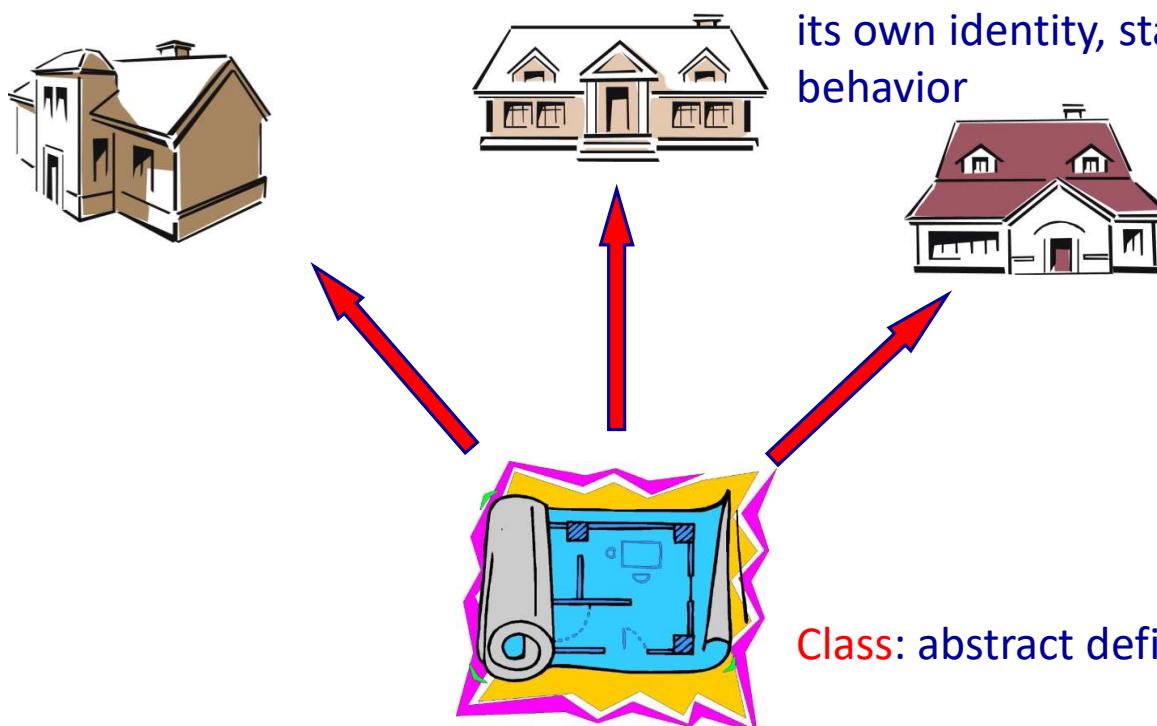


# Object relations



# Class

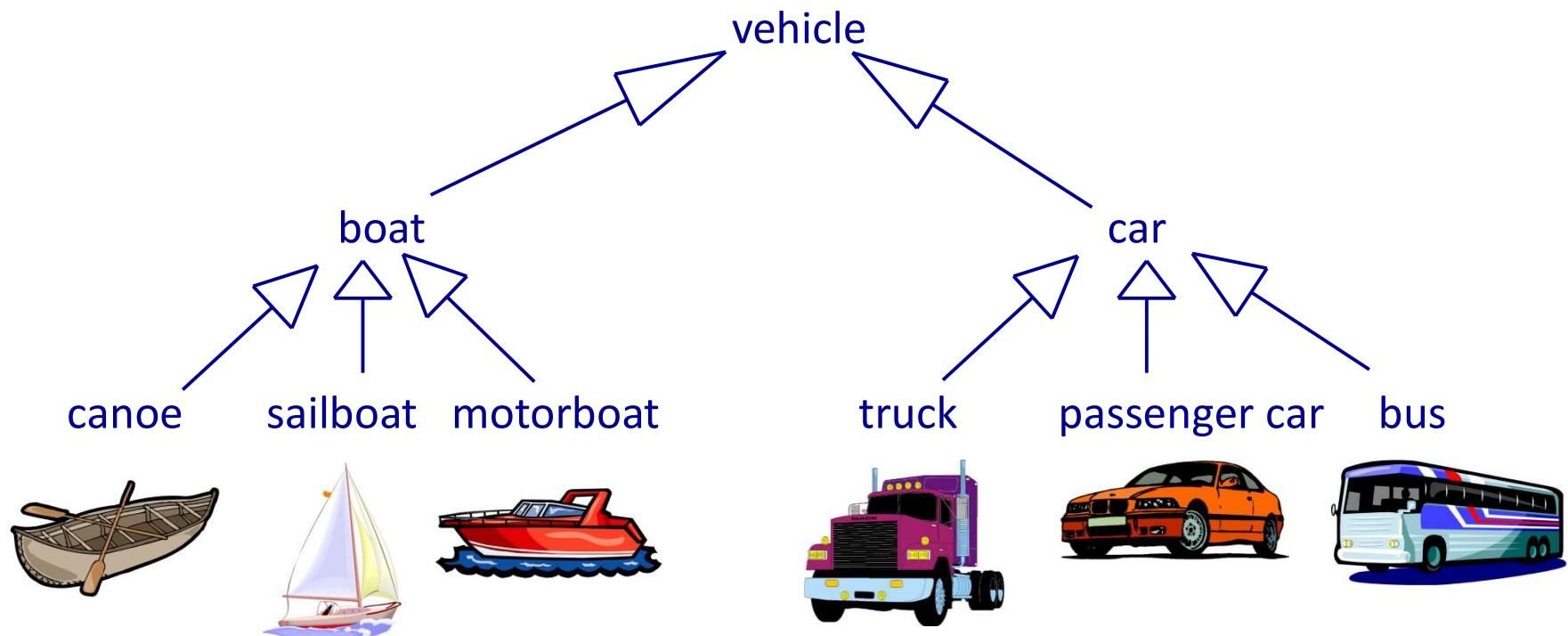
---



# Inheritance

---

- Is-a relationship between generic and specific classes



# Polymorphism

---

- Polymorphism = many forms
- Objects behave differently on the same command



# The virtual world of objects

---



- Software objects can do anything
  - A product can sell itself
  - A mortgage can compute its monthly payment
  - A shoppingcart can check itself out

# Characteristics of objects

---

- Objects are real world entities or concepts in the problem domain.
- Objects are small.
- Objects can do one thing very well, all other things are delegated to other objects.
- Objects
  - Know things (attributes)
  - Can do things (methods)

# Finding classes

---

- Identify the things and concepts in the problem domain
- Noun/verb analysis
  - Nouns become classes or attributes
  - Verbs become methods

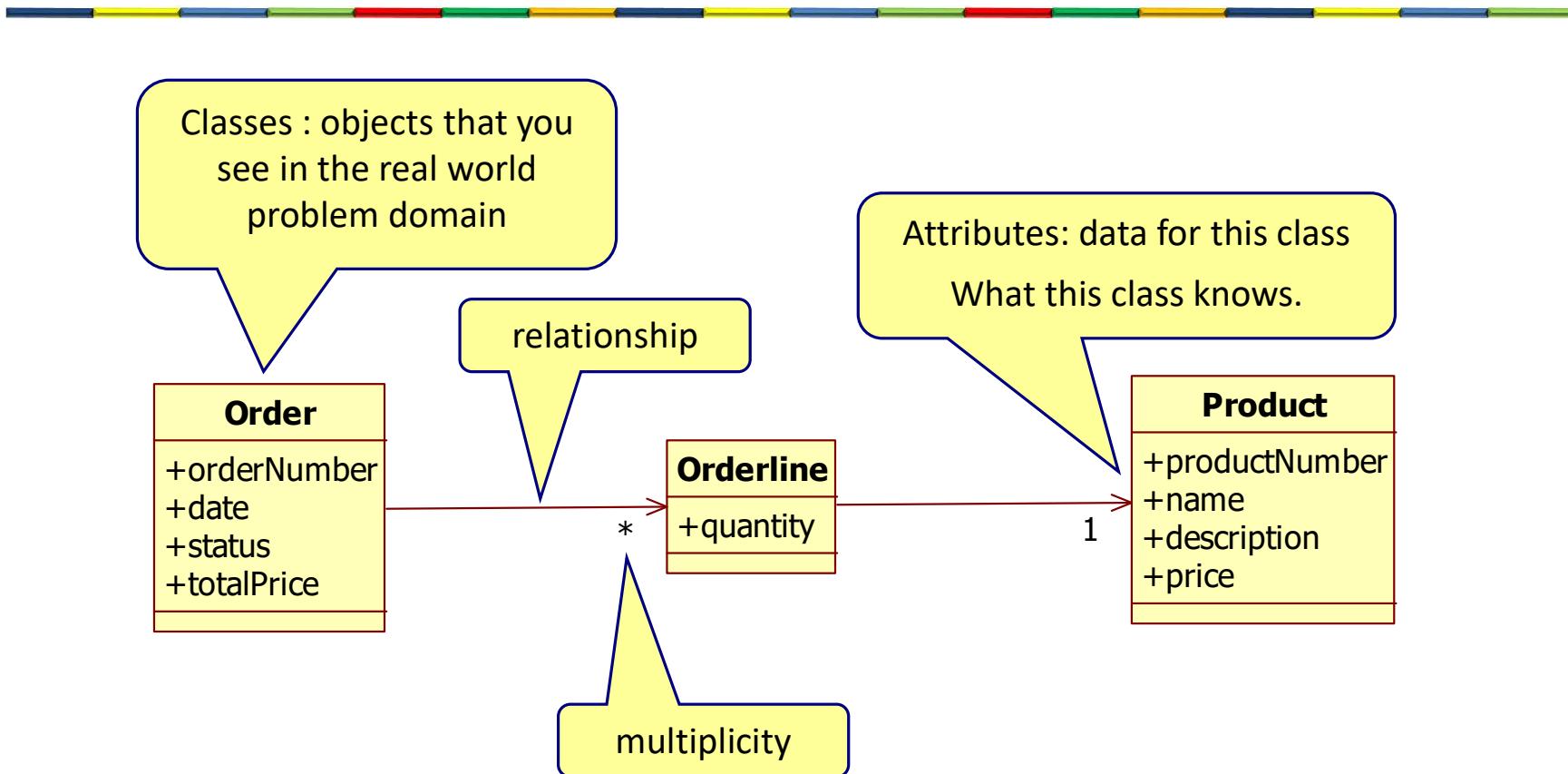
# Exercise

---



# Domain model class diagram

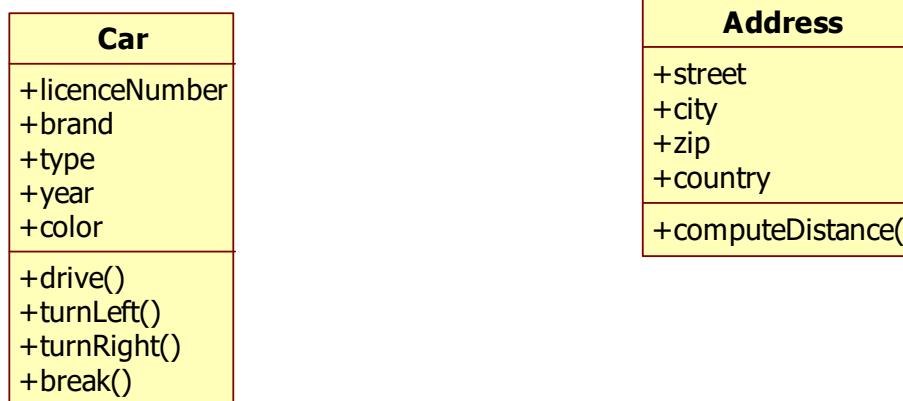
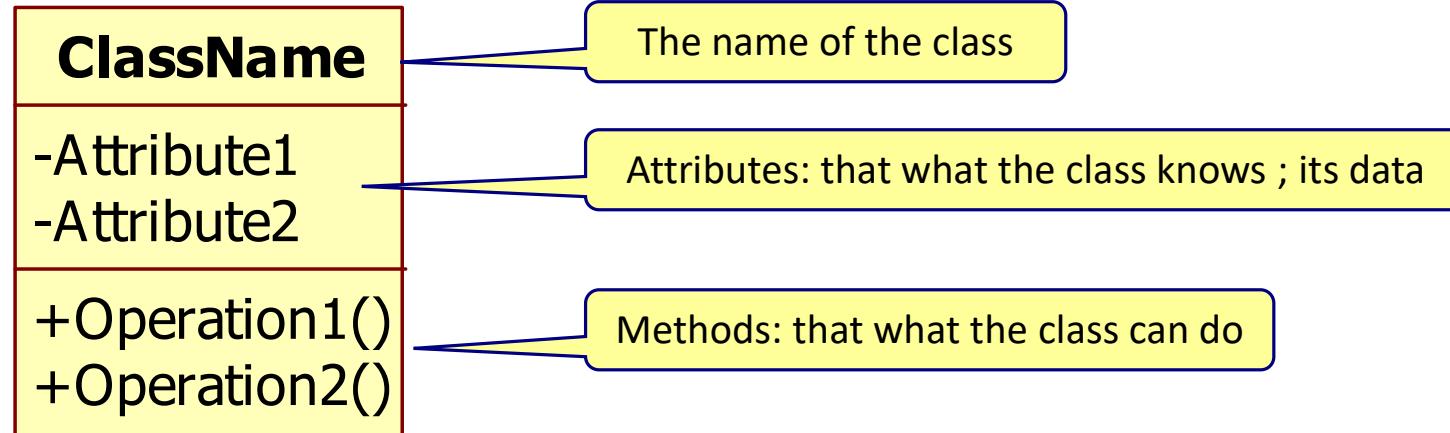
# UML class diagram elements



*Describes the structure of a system by showing the classes, their attributes, and the relationships*

# Class

---



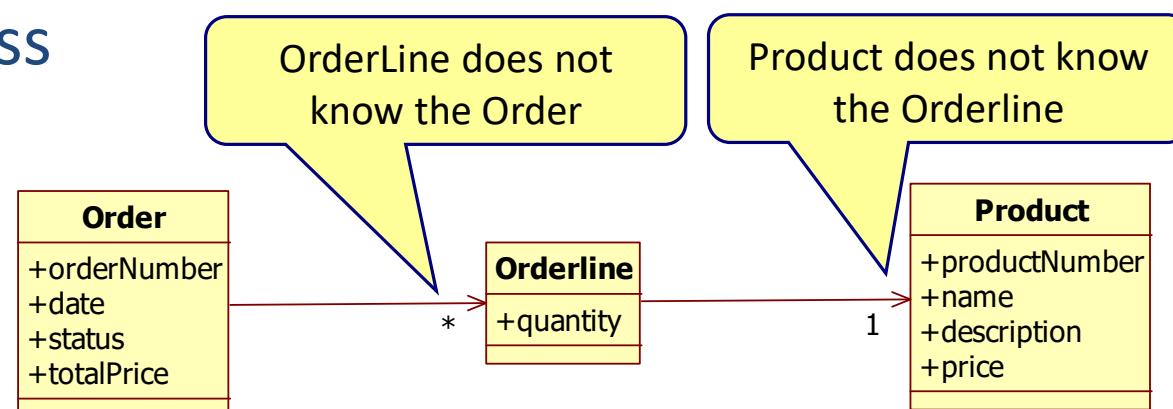
# Association

---

- Defines the relationship between 2 classes
  - Bi-directional: both classes are aware of each other



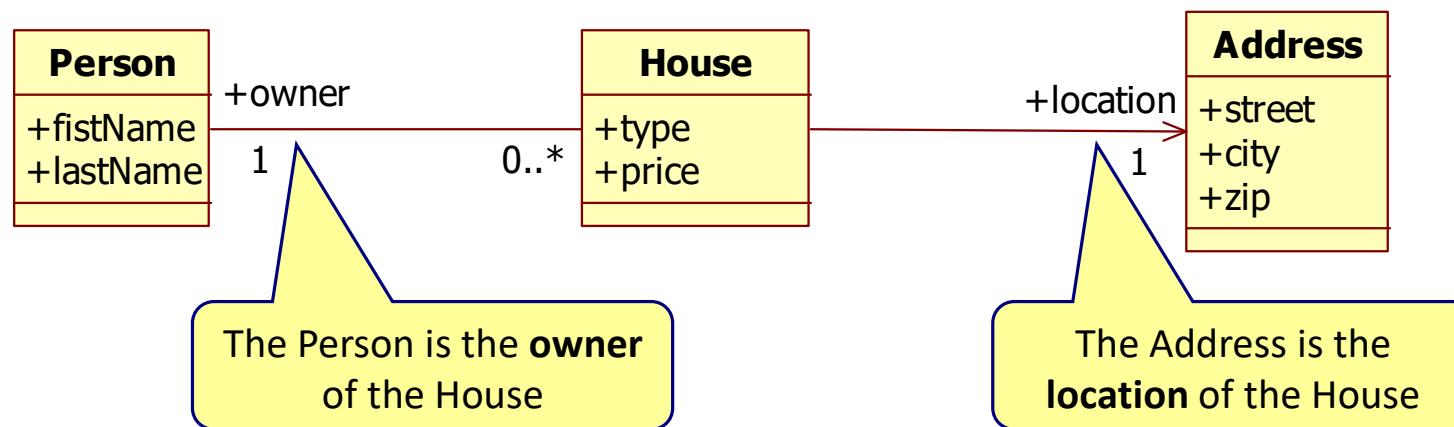
- Uni-directional: only one class is aware of the other class



# Association Role

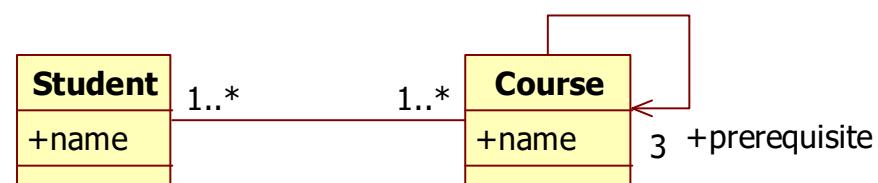
---

- Indicates the role played by the class attached to the end of the association path



# Multiplicity

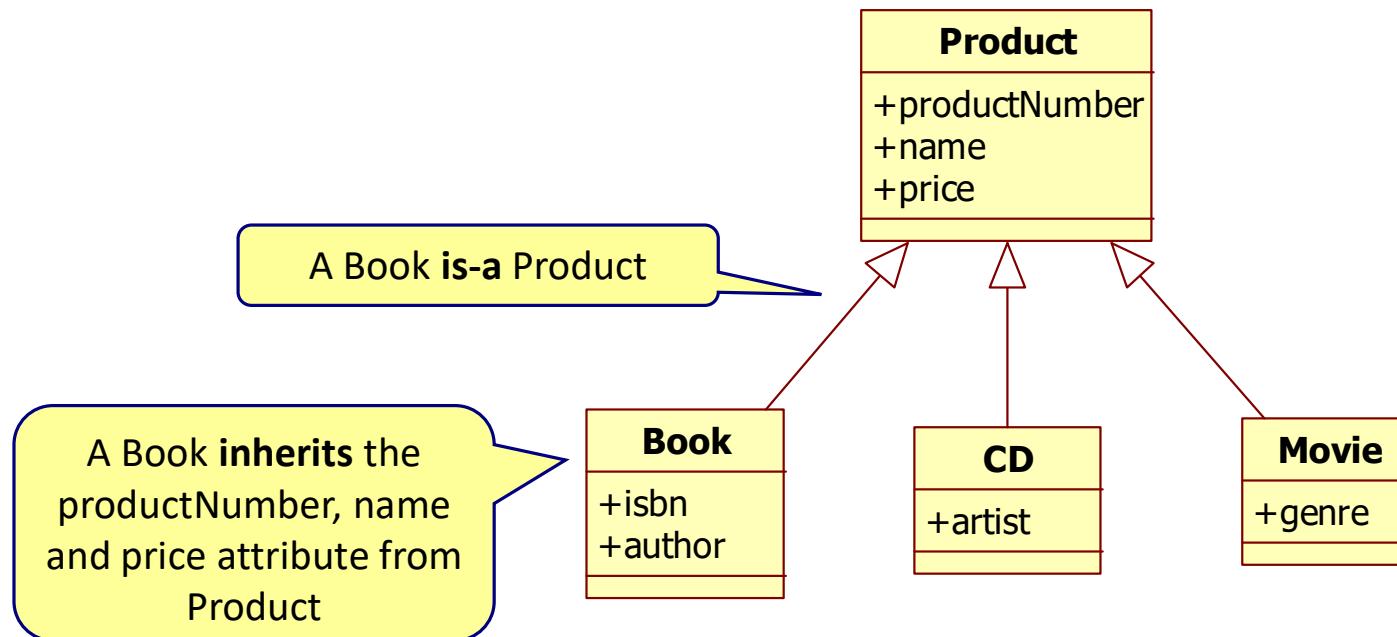
- 0
  - 1
  - 0..1
  - 0..\*
  - 1..\*
  - \*
  - 7
  - 5..7
  - 2..4
- Zero or one
- Zero or more
- One or more
- Zero or more
- Specified range
- Two or four



# Inheritance

---

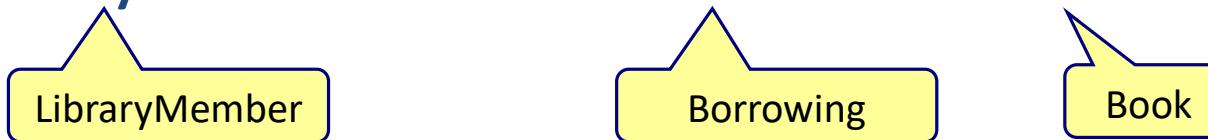
- Is-a relationship



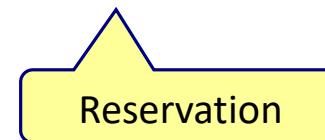
# Finding classes

---

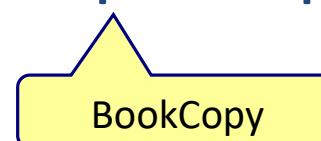
- Library members can borrow books



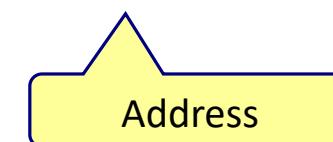
- Library members can reserve books



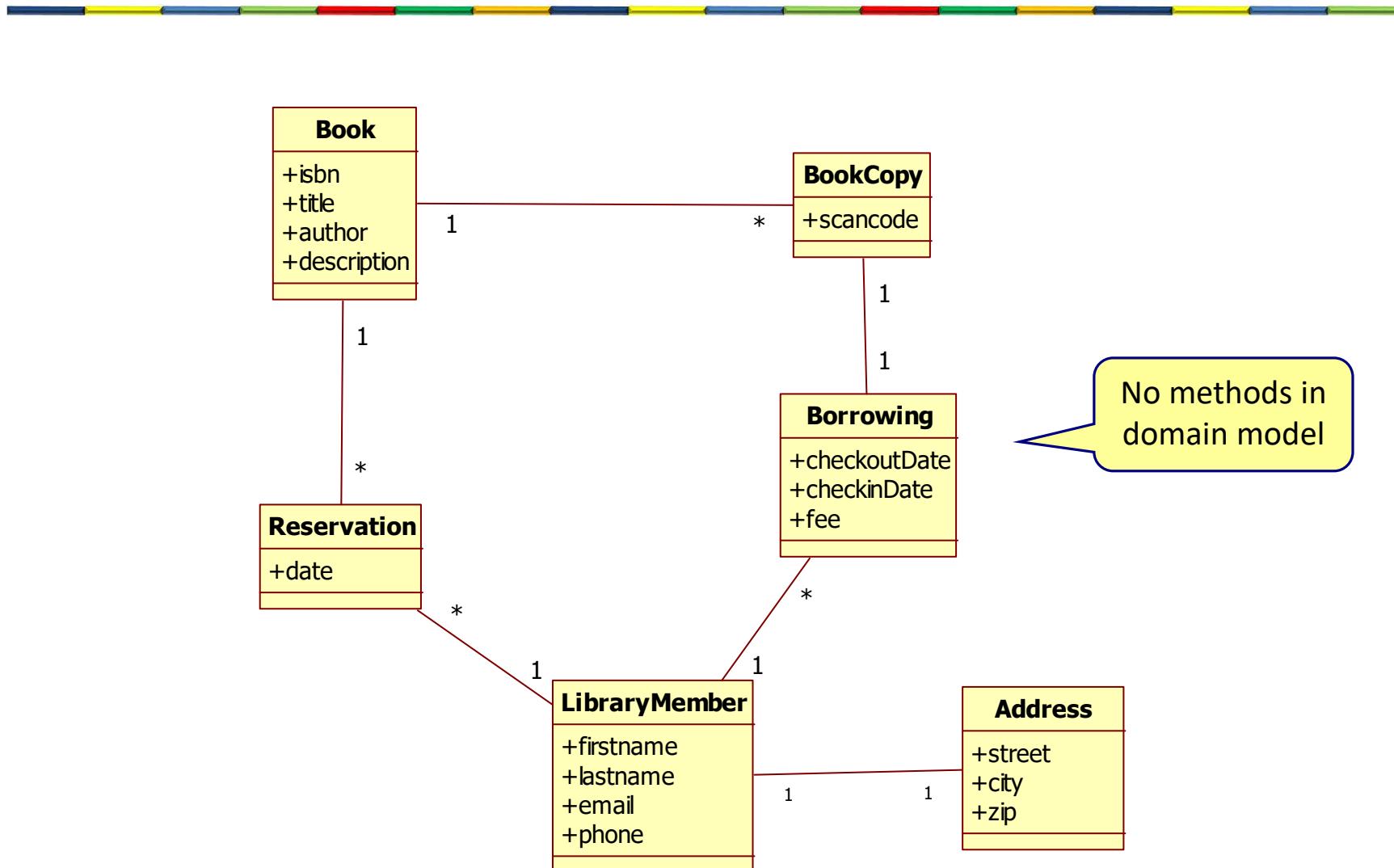
- We can have multiple copies of one book



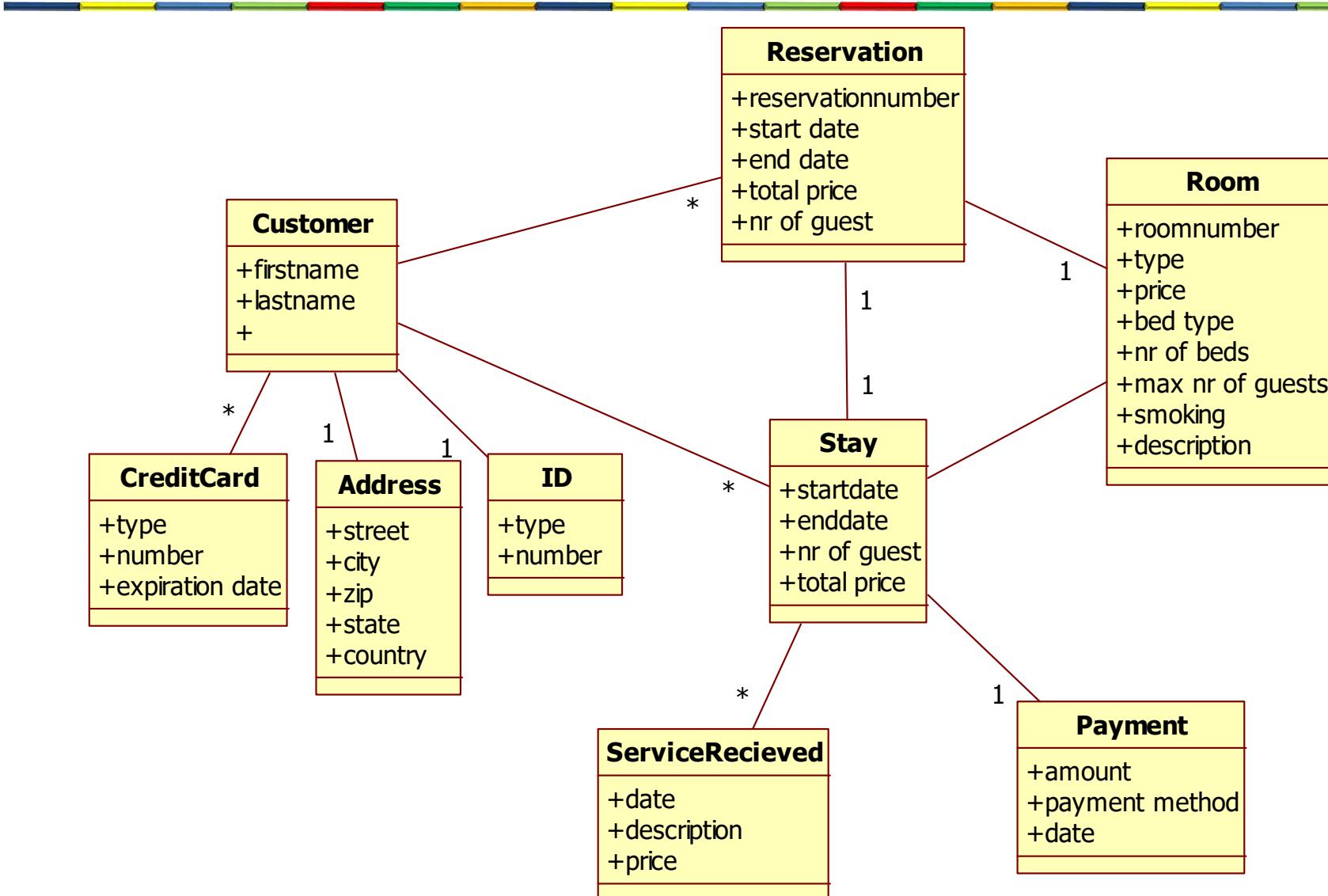
- Library members live at a certain address



# Library system domain model



# Hotel reservation system



# Main point

---

Software engineering

SCI

The domain model shows the entities, attributes and relationships of the problem domain

The human physiology has the same structure as the structure of the Veda and Vedic literature who are expressions of the structure of pure consciousness.