Object-Oriented Analysis

Xin Feng

Outline

- OOA
- UML diagrams

Object-Oriented Software Development

- Develop the software which is a collection of objects that incorporate both data structure and behavior
- OO software development
  - Object-Oriented Analysis (Requirements specification)
  - Object-Oriented Design (Architectural Design)
  - Object Design (Detailed Design)
  - Object-Oriented Programming (Implementation)
- Essence
  - Identify and organize the application domain objects
  - Do NOT consider the representation in programming languages

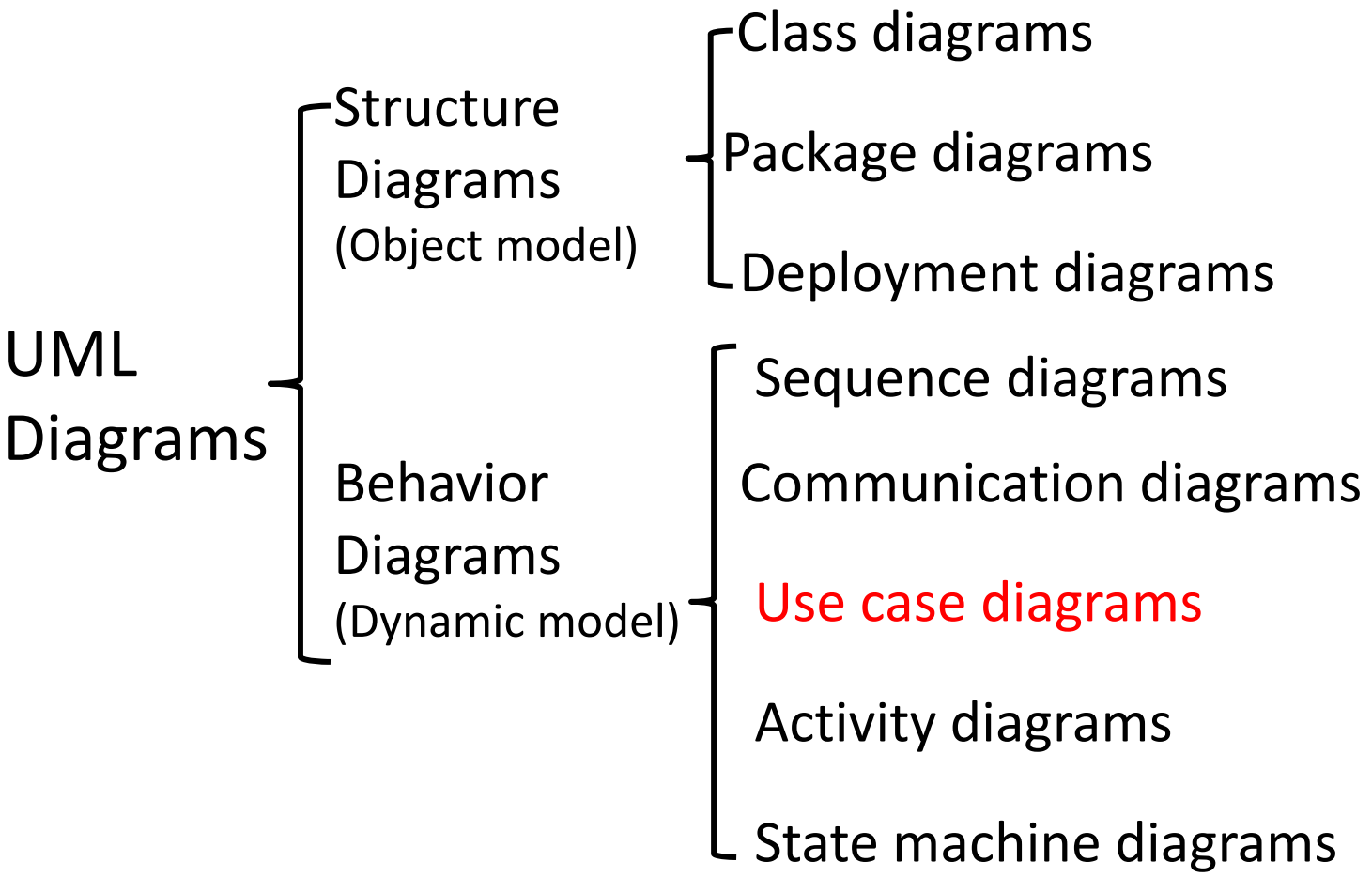
Unified Modeling Language (UML)

- The most popular diagrammatic notation used for object-oriented development
- An integration of several OO modeling methods
- Support from OOA (Object-Oriented Analysis) to OOP (Object-Oriented Programming)
- History
  - James Rumbaugh (OMT for OOA)
  - Grady Booch's (Booch's method for OOD)
  - Ivar Jacobson (OOSE)
  - Rational Software Corporation (IBM)

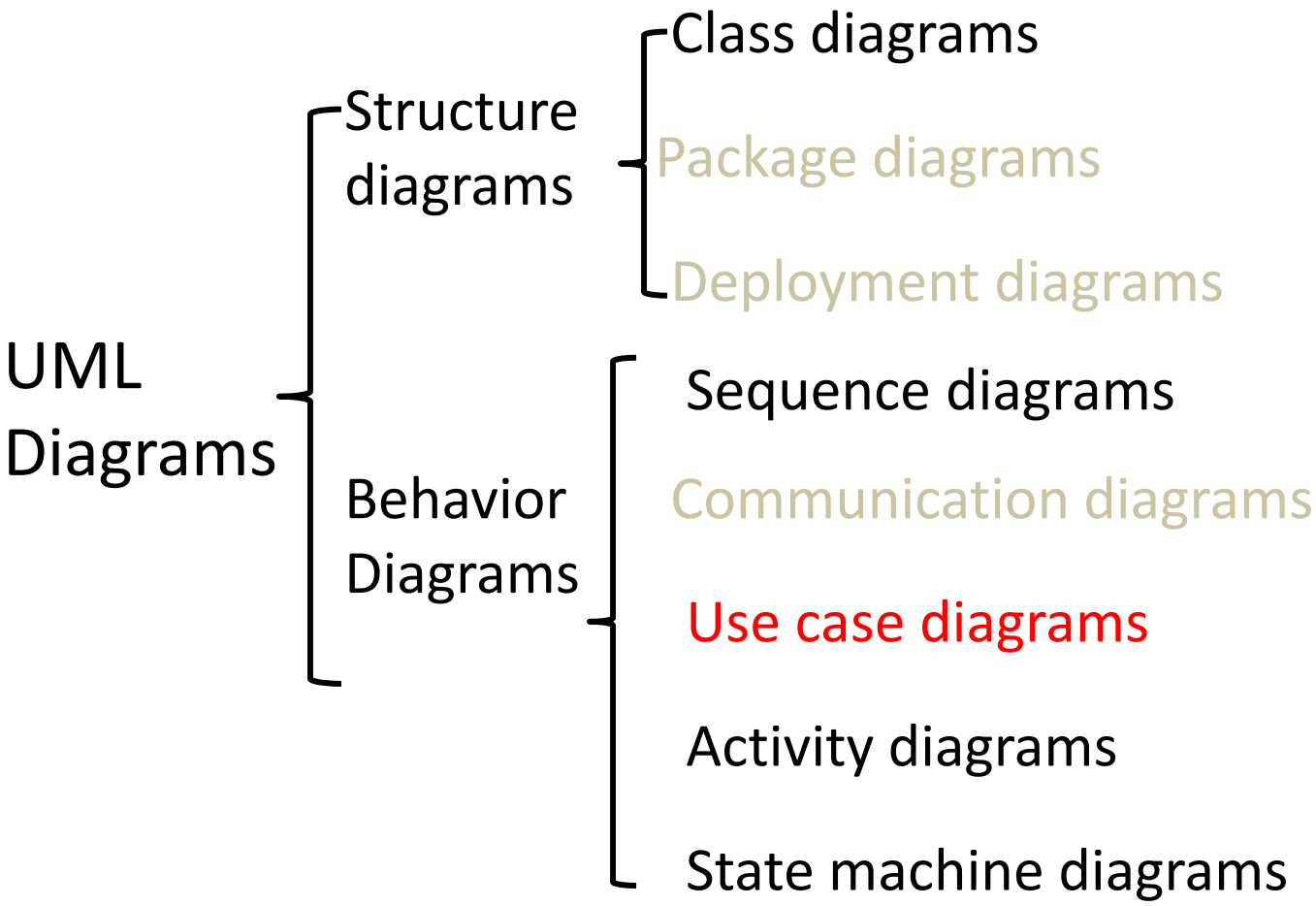
Unified Modeling Language (UML)

Undefined Modeling Language?

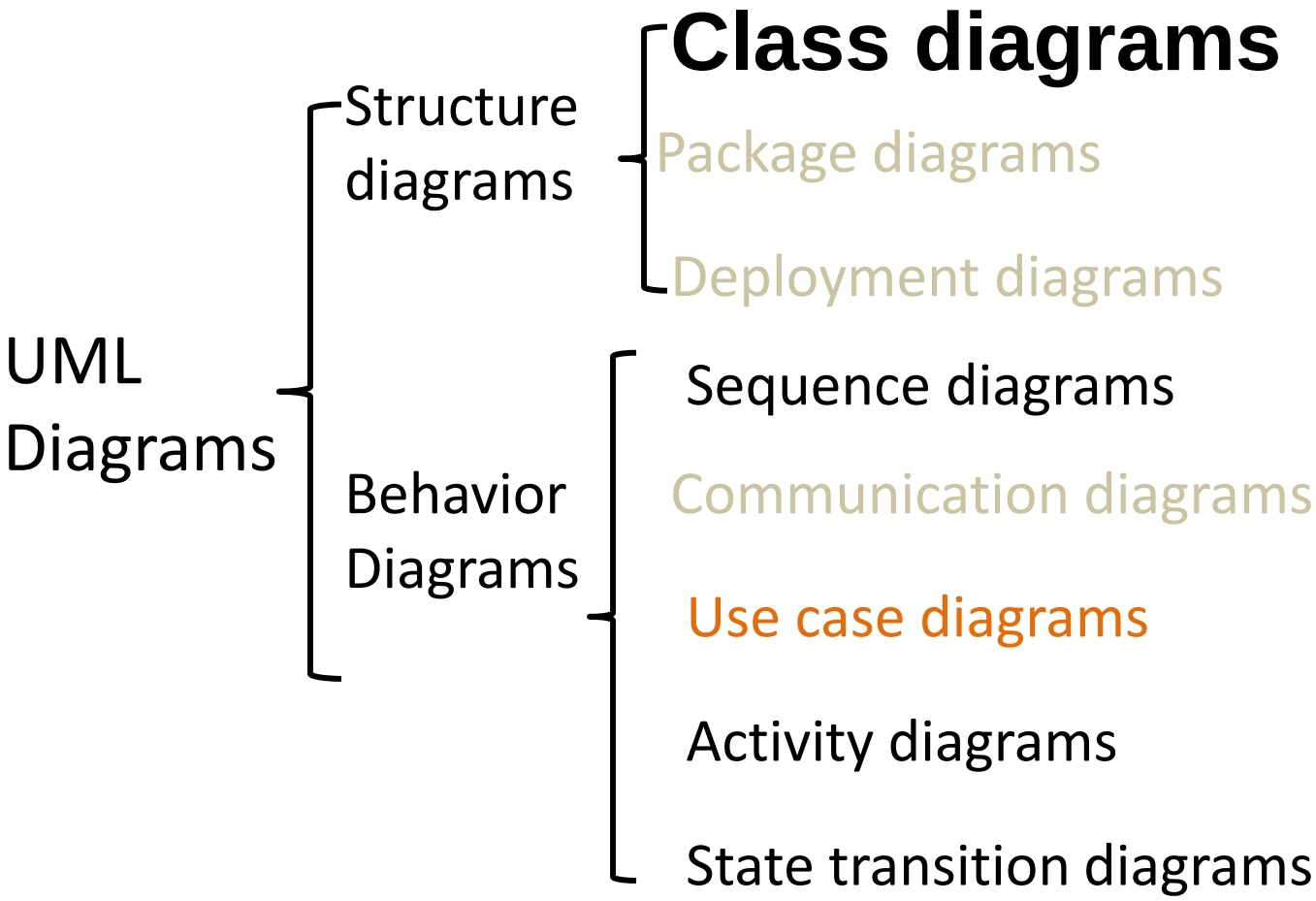
Unified Modeling Language(UML)



Unified Modeling Language (UML)



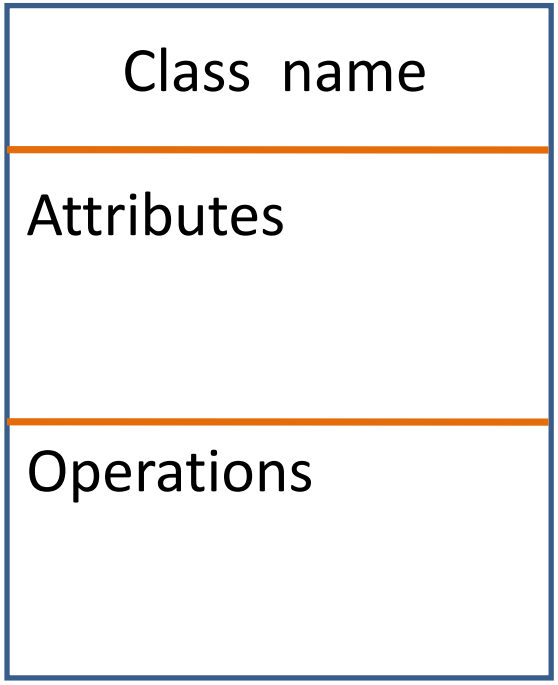
Unified Modeling Language (UML)



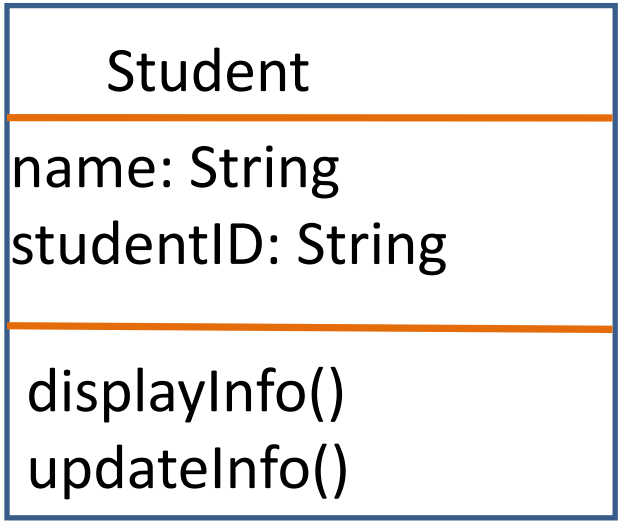
Class Diagrams

- Describe the system in terms of object classes and their associations （关联）
- Natural ways of reflecting （反映）the real-world entities and their relationships
- The class diagrams are the essential （核心）part in Object-Oriented Software Development
- Object diagrams

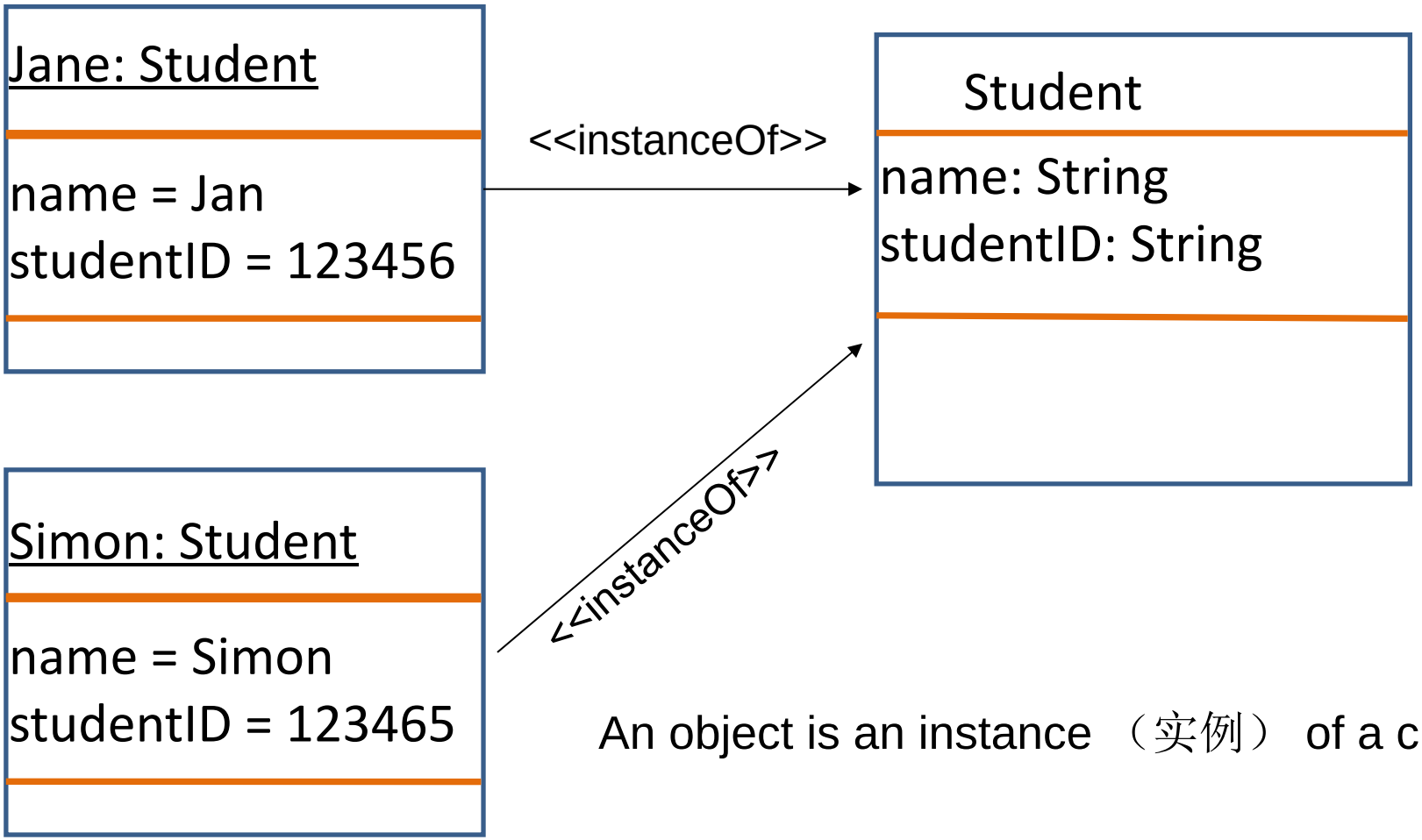
Class Notation



Class - An Example



Object - An Example



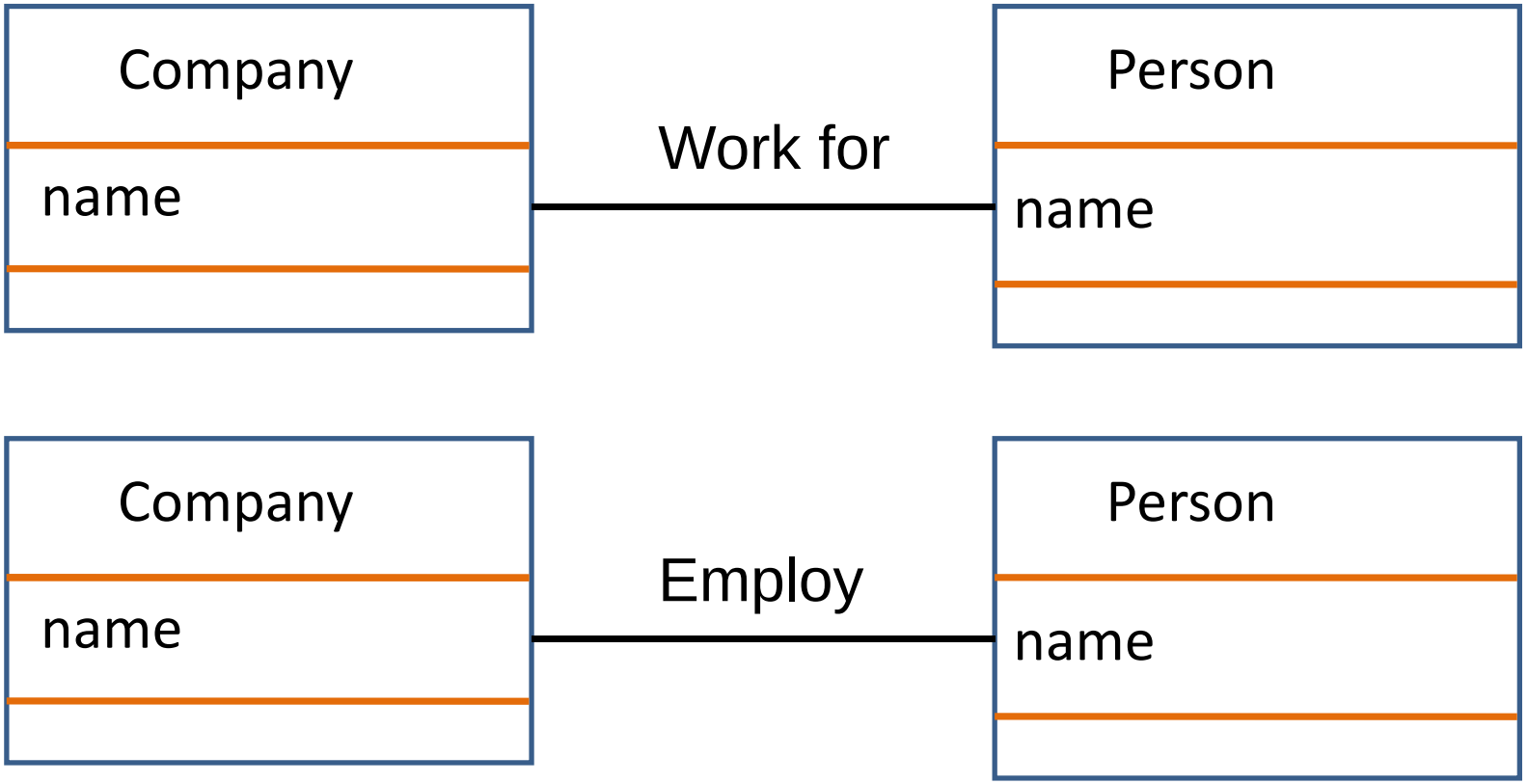
An object is an instance （实例） of a class

Associations and links

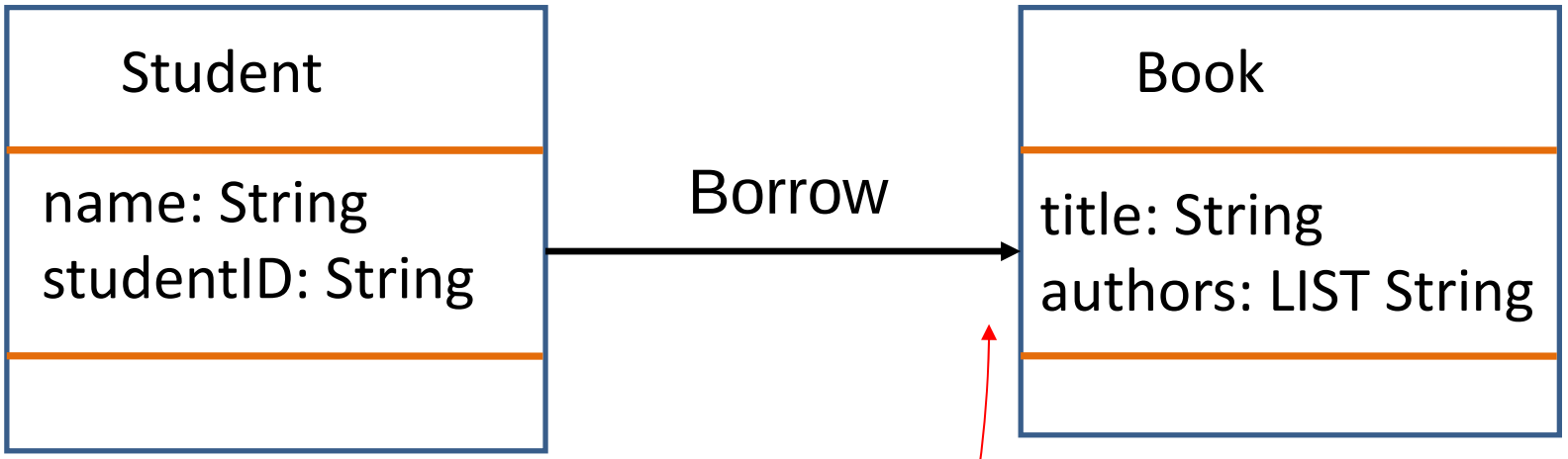
- A link （链接） is
  - a connection between two objects
- An association
  - a relationship between classes and represents a group of links
  - bidirectional （双向） or unidirectional （单向）

Association - Examples

Association - Examples



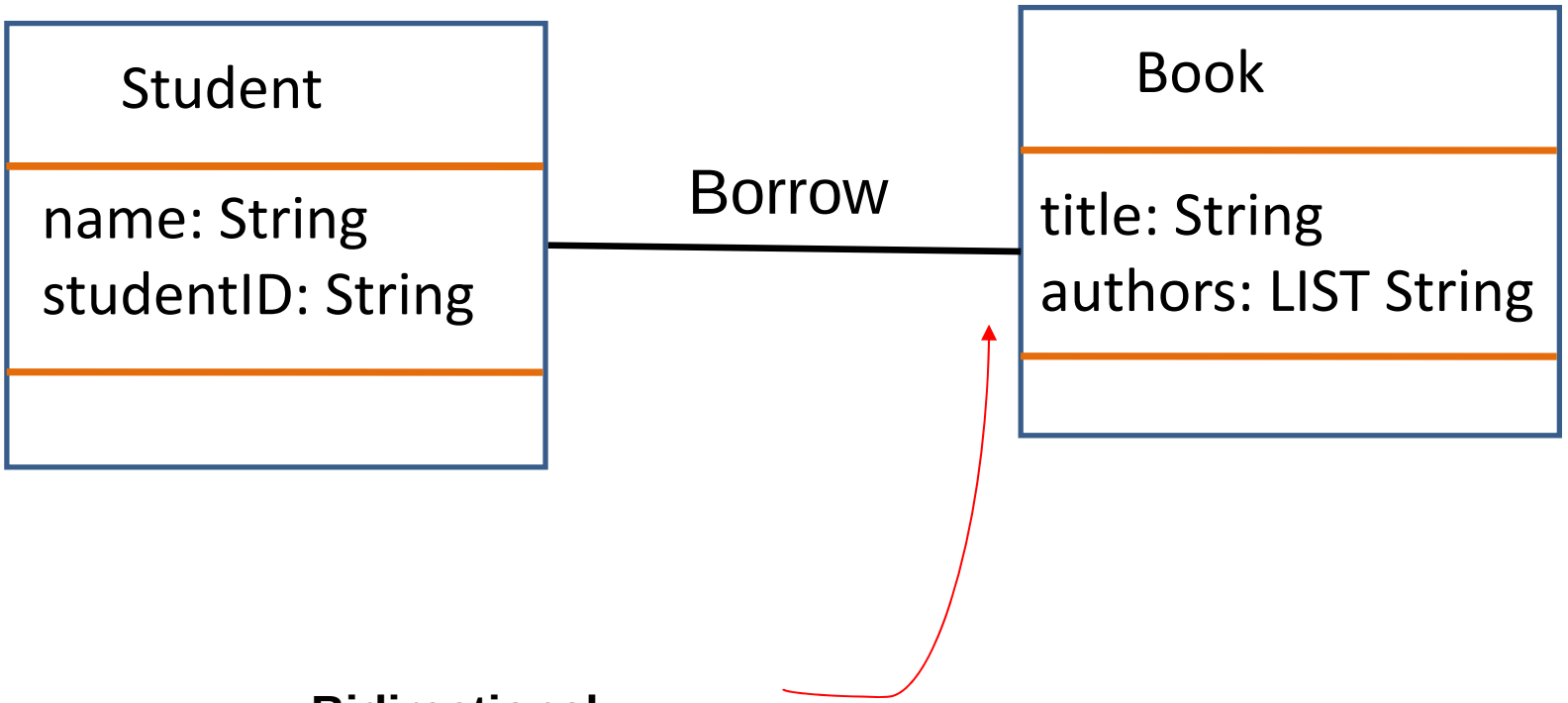
Association



Unidirectional:

A student can query the books he/she borrowed but it is not possible to find which student is this book lent to

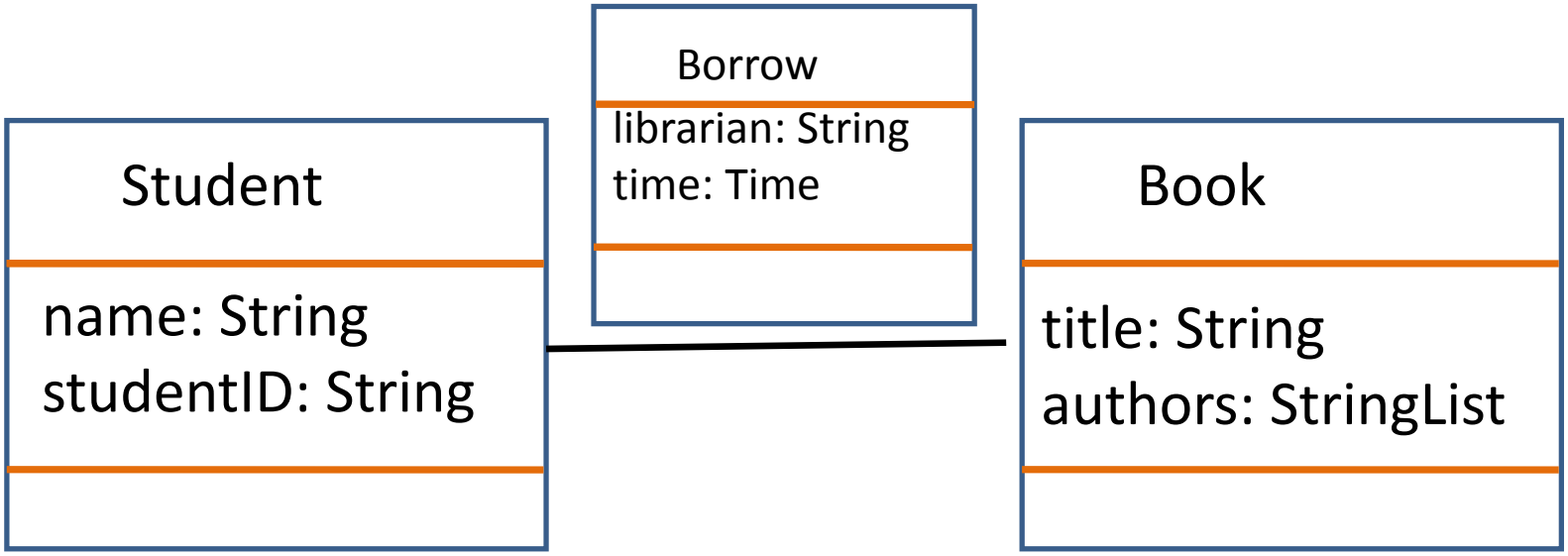
Association



Bidirectional

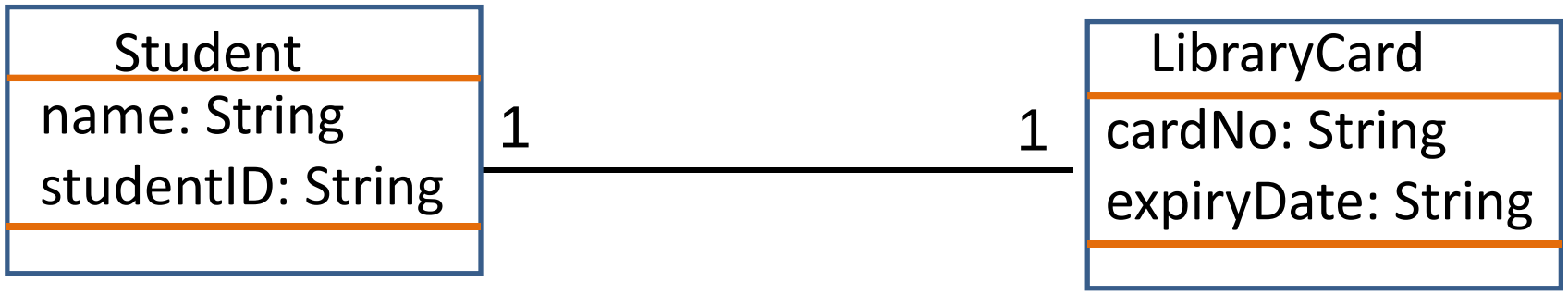
A student can query the books he/she borrowed and
It is possible to find which student is this book lent to

Association classes



An association can have attributes and operations.
The association class cannot exist without its linked classes.

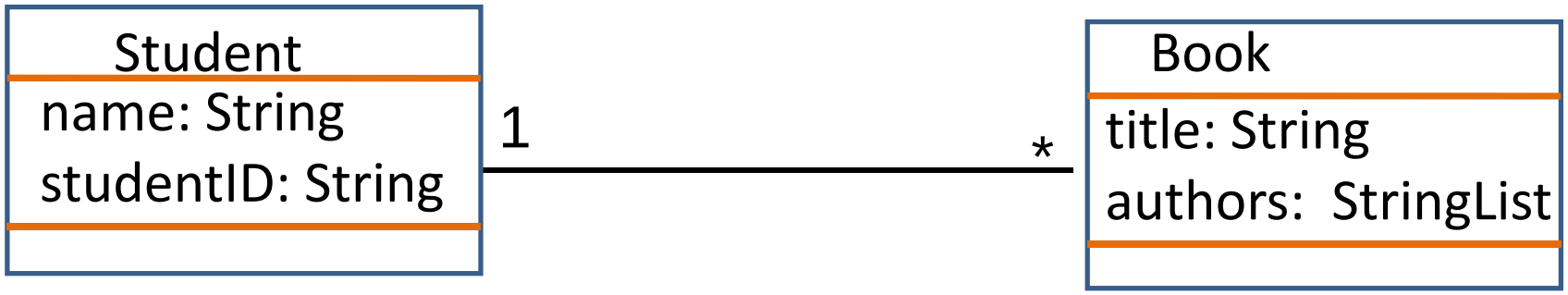
Multiplicity



One to one relationship

One student has only one borrower card and
a library card can only be owned by one student.

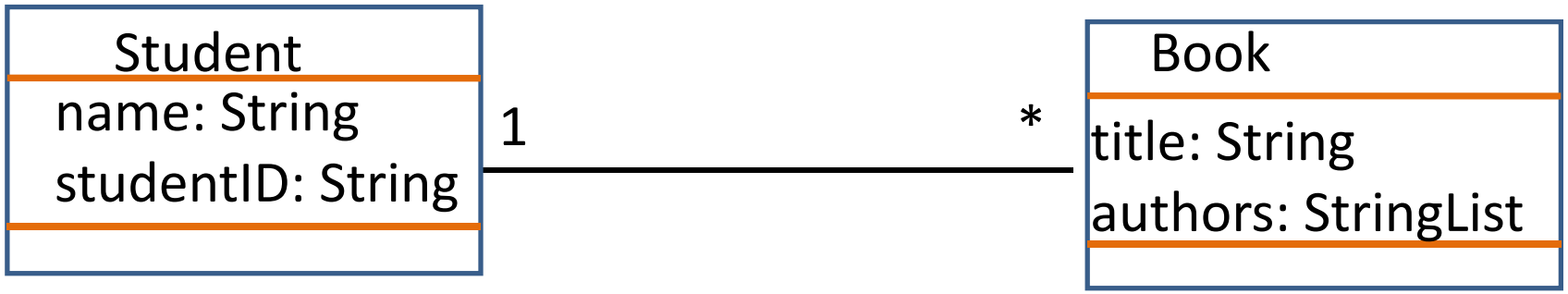
Multiplicity



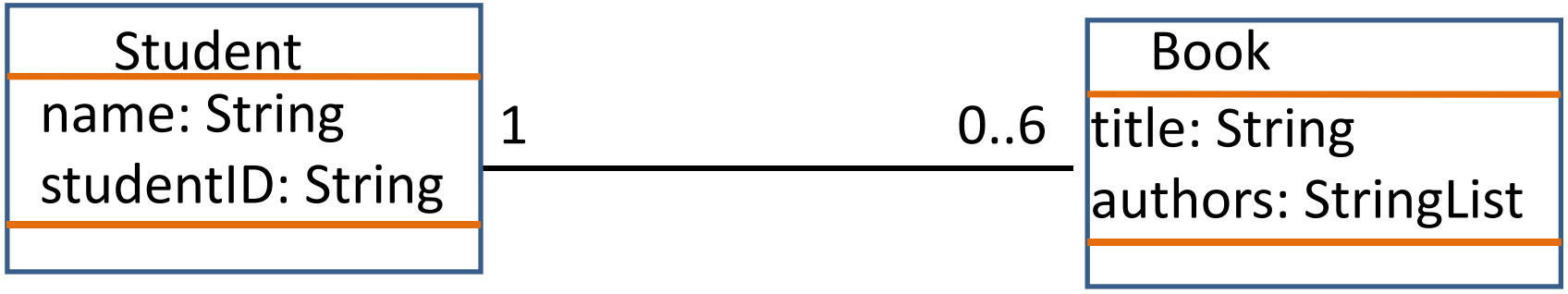
One to many relationship

One student can borrow 0 or many books

Multiplicity

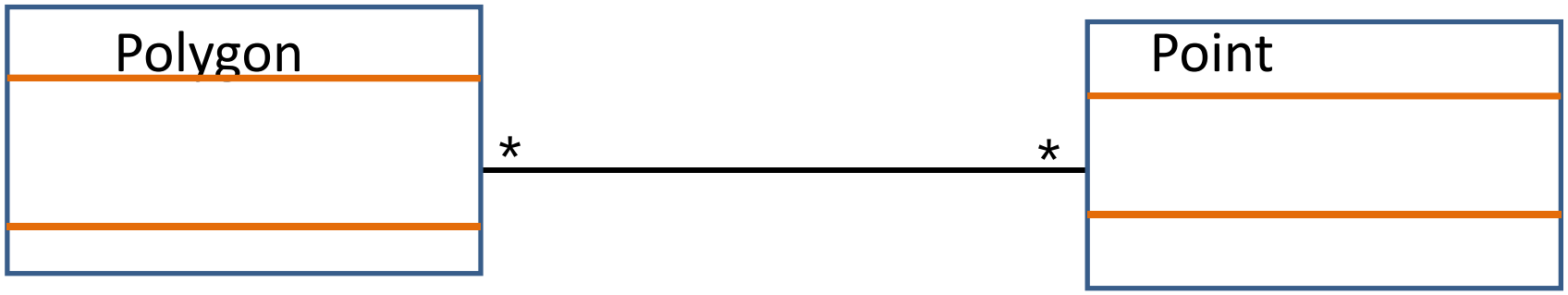


One student can borrow 0 or many books



One student can borrow at most 6 books

Multiplicity

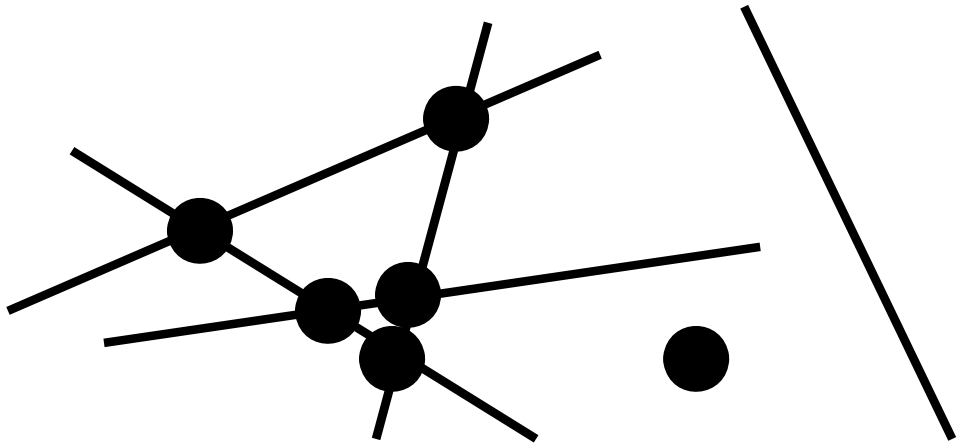


Many to many relationship

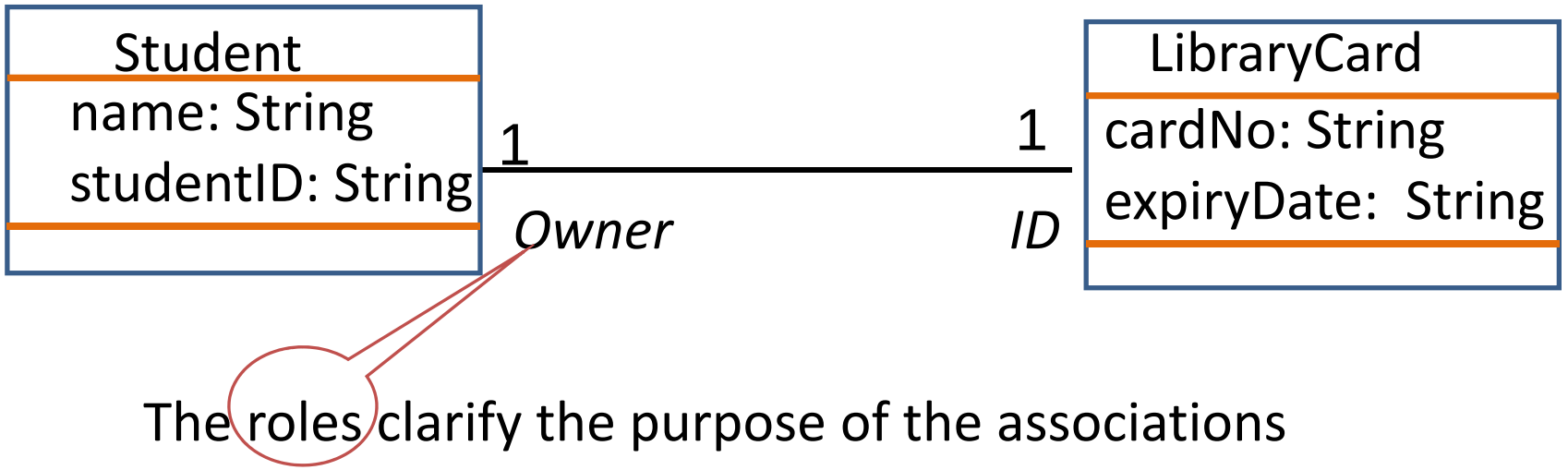
One polygon has many points and one point can be in many polygons （多边形）

Class Exercise

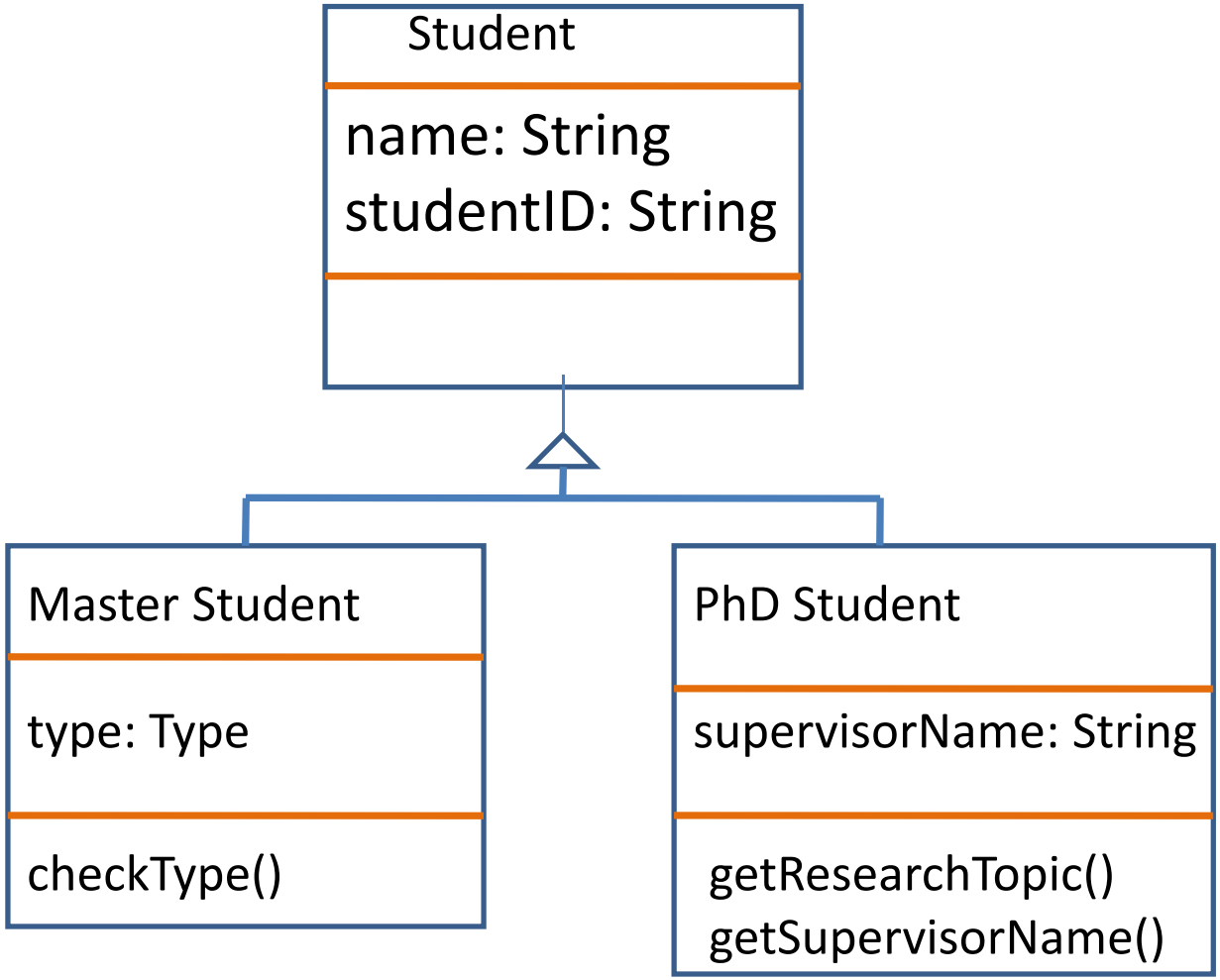
How to represent this?



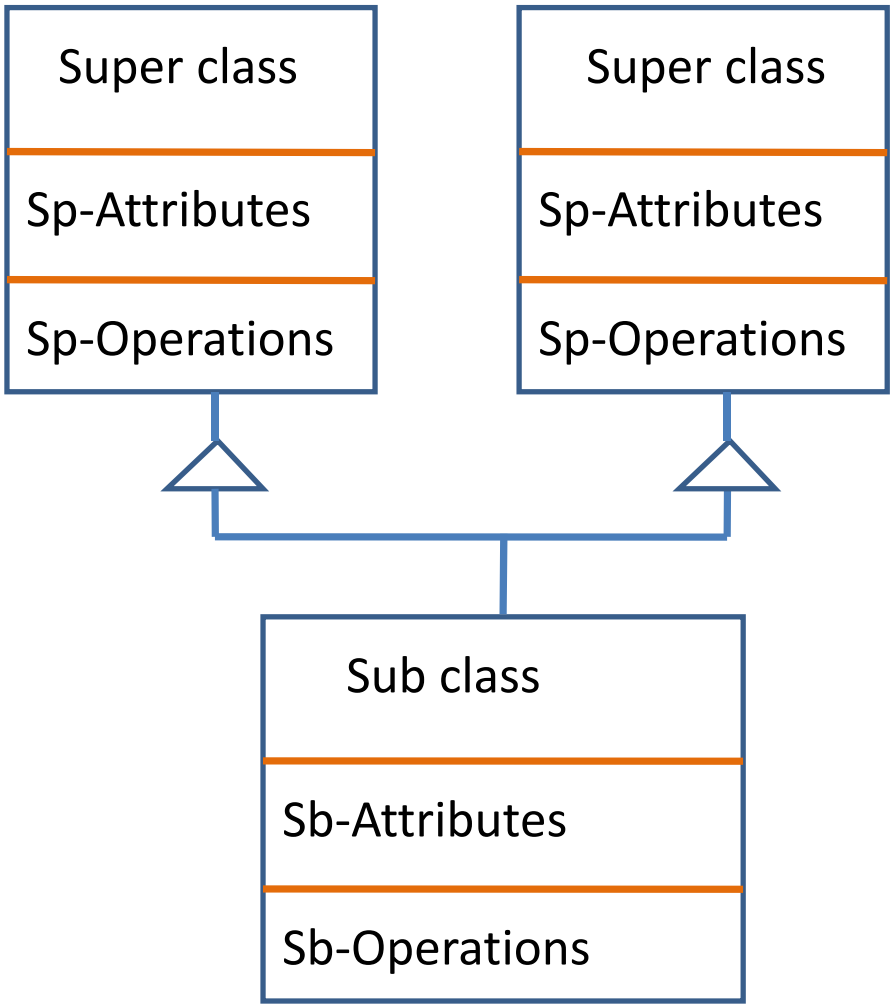
Roles


Student
name: String
studentID: String
1
Owner
1
ID
LibraryCard
cardNo: String
expiryDate: String
The roles clarify the purpose of the associations

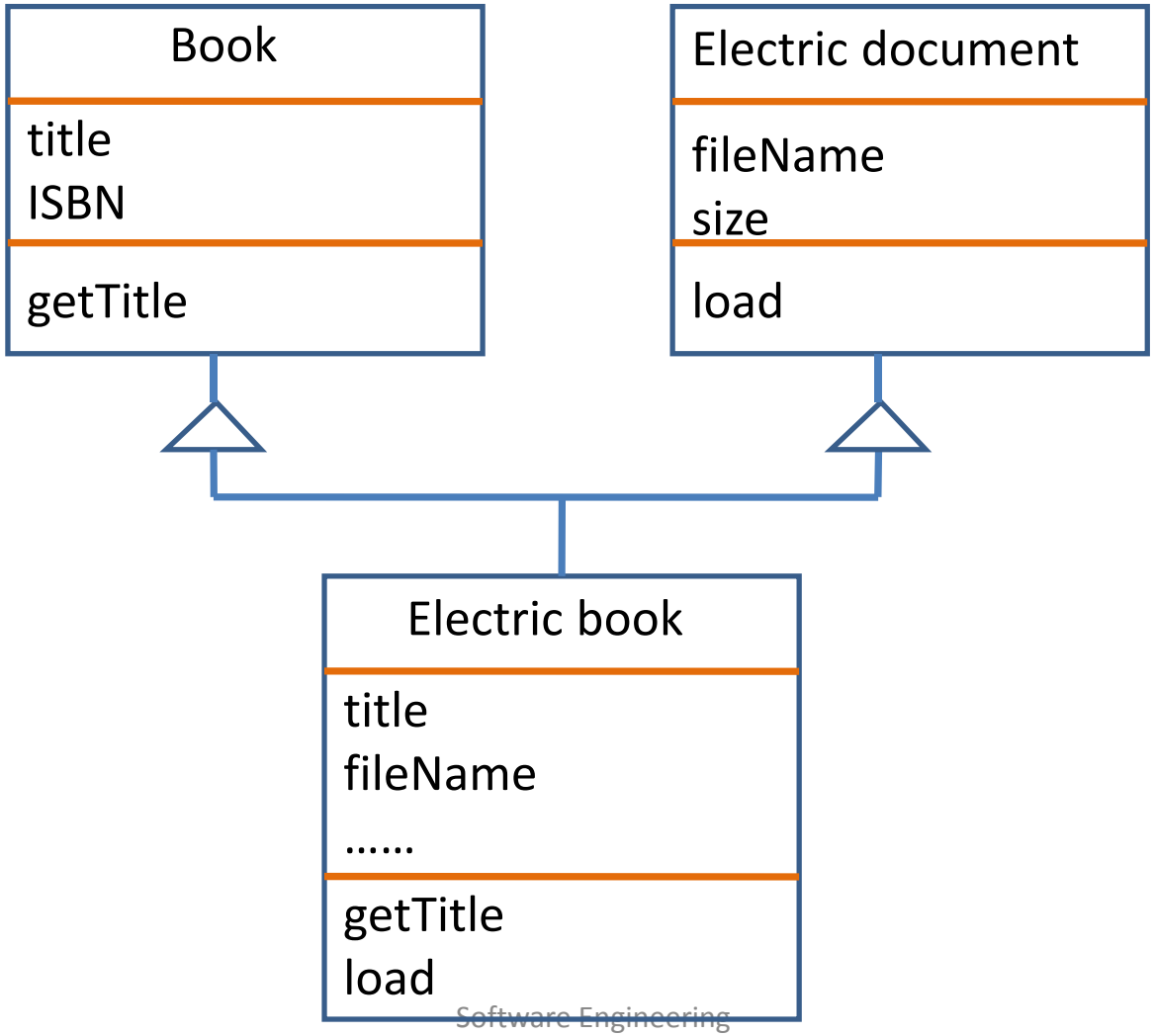
Inheritance



Multiple Inheritance



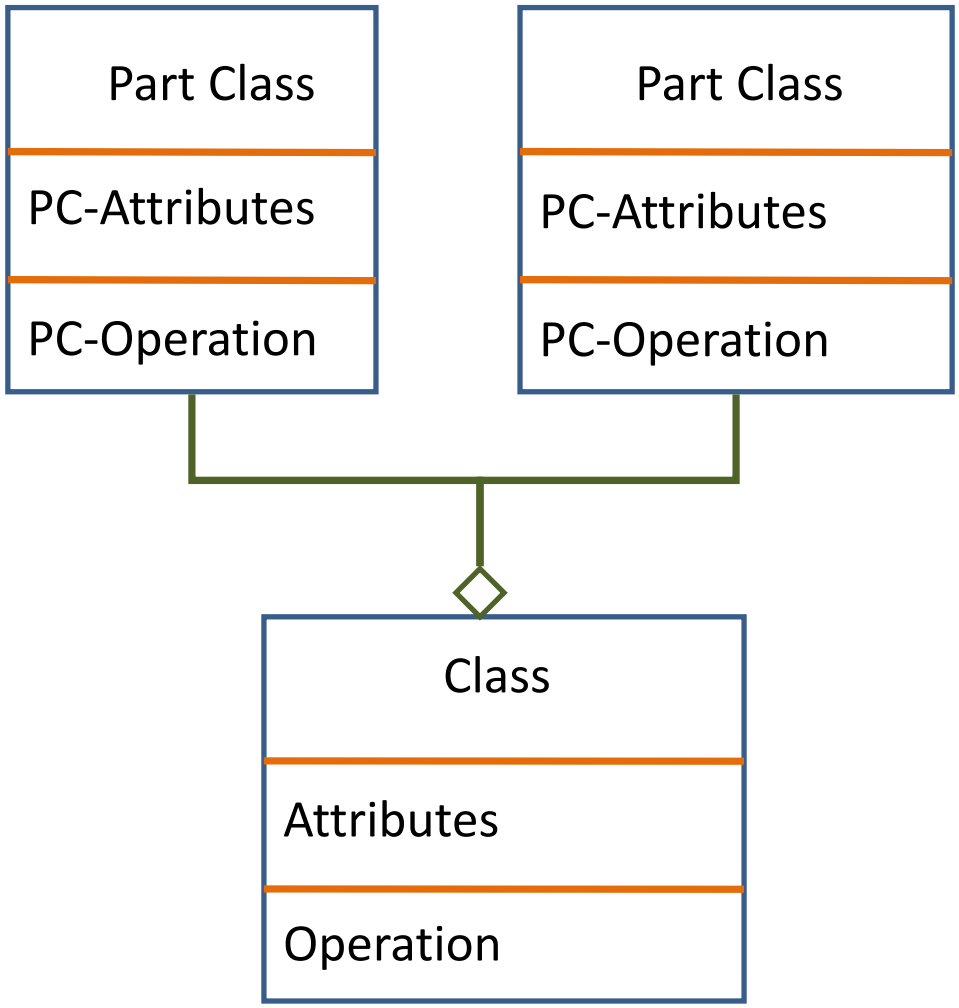
Multiple Inheritance



Multiple Inheritance

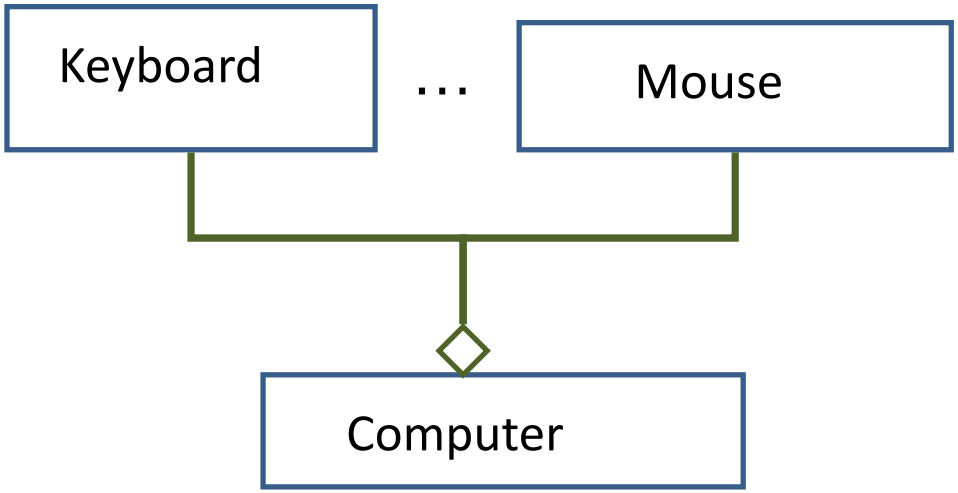
- A class can inherit the attributes and services not only from a single parent class, but also from several super-classes
- Problems
  - Semantic （语义）conflicts （冲突）may occur where attributes/services with the same name in different super-classes have different semantics
  - Class hierarchy （层次） reorganisation （重组） becomes more complex （复杂）

Aggregation



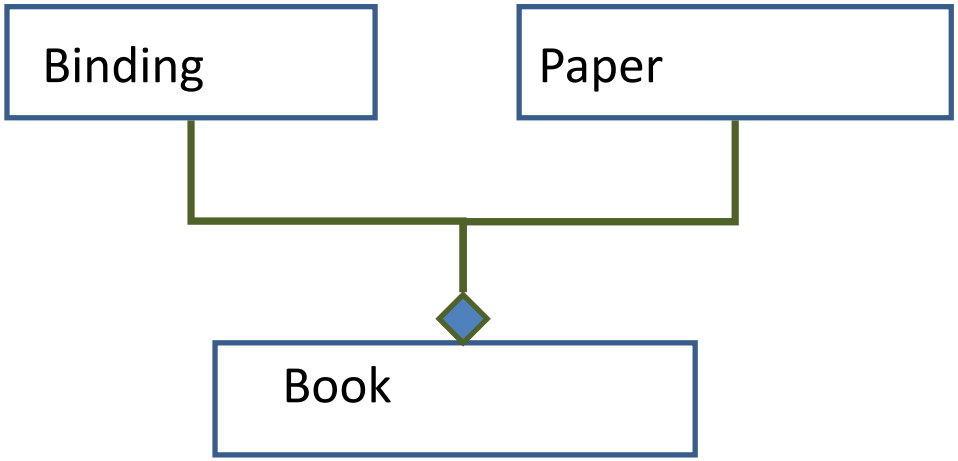
- A class can be composed of (由..组成) other classes
- This relationship is similar to the part-of relationship

Aggregation

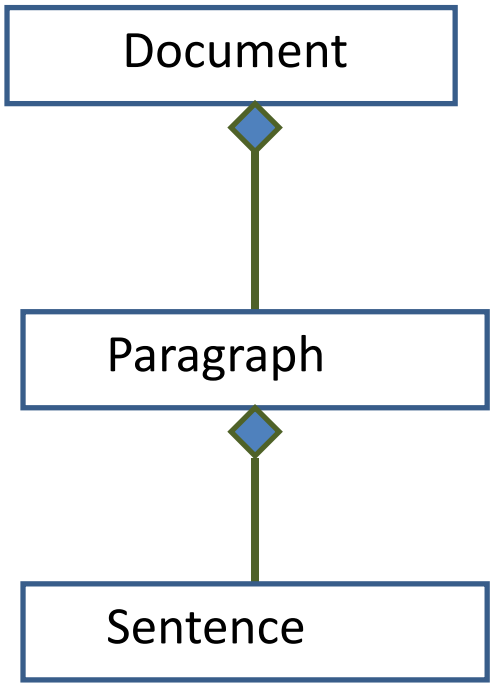


Composition

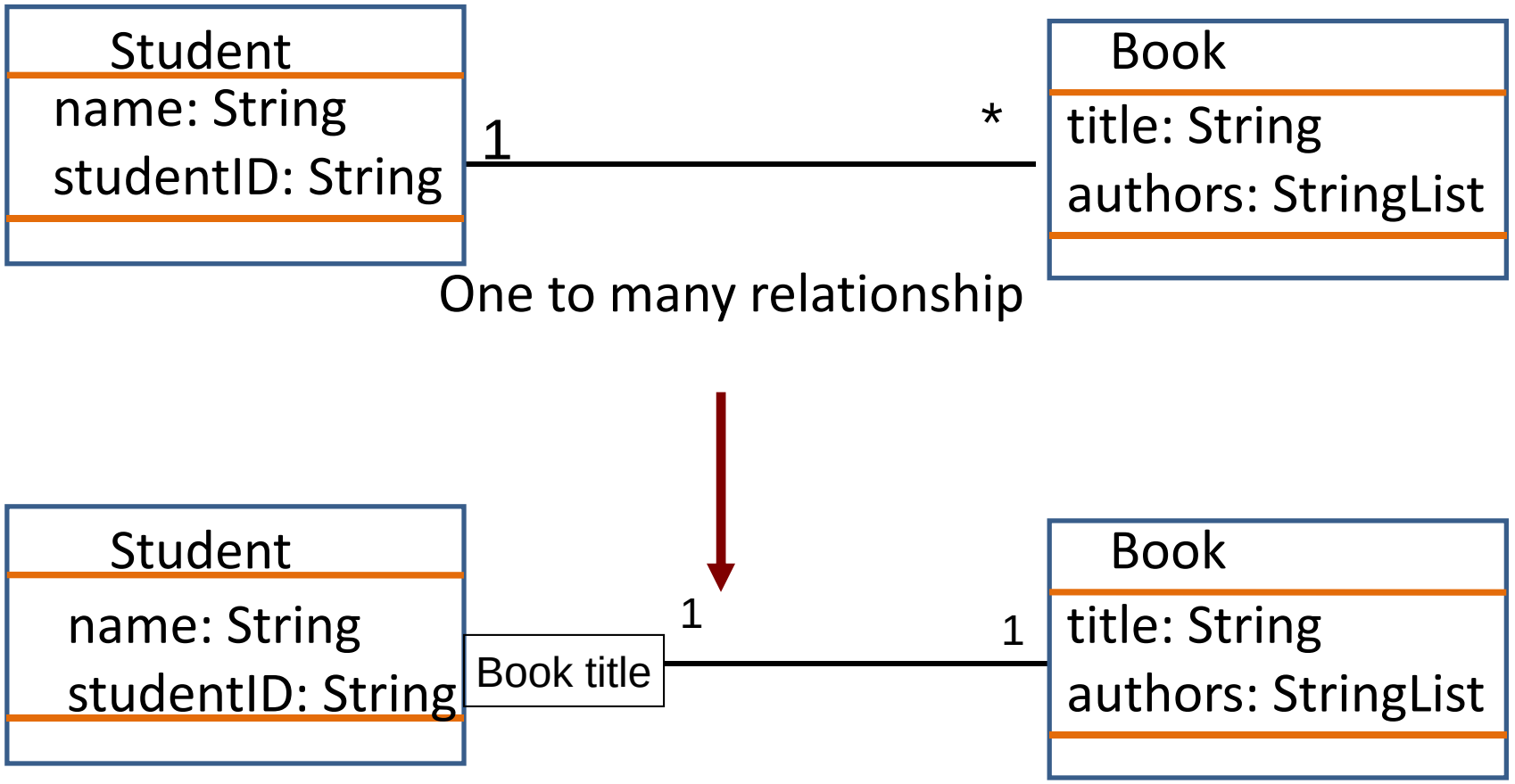
Composition: A form of aggregation with strong ownership



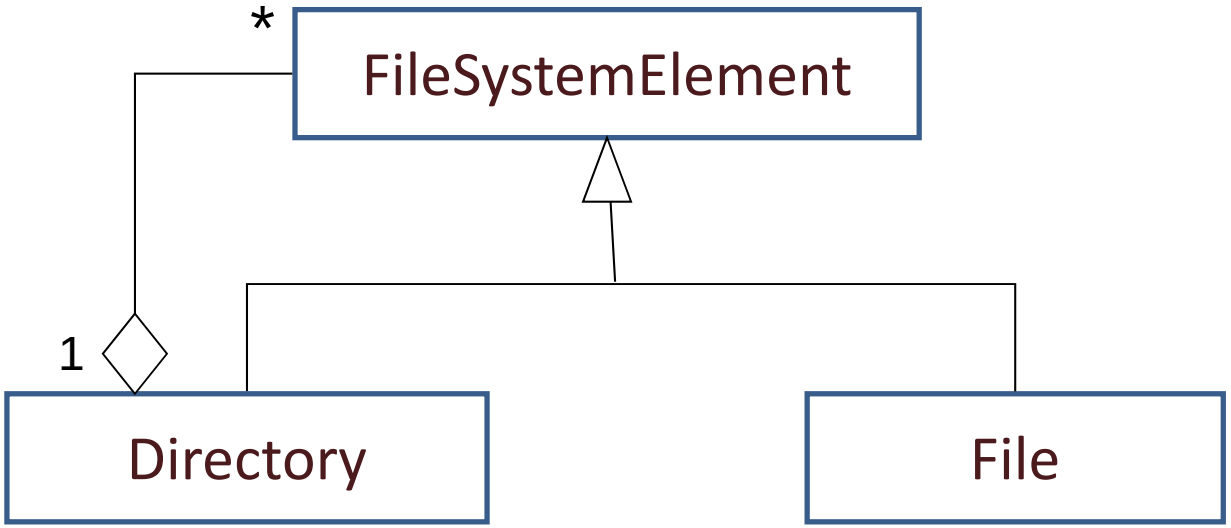
Composition



Qualifier

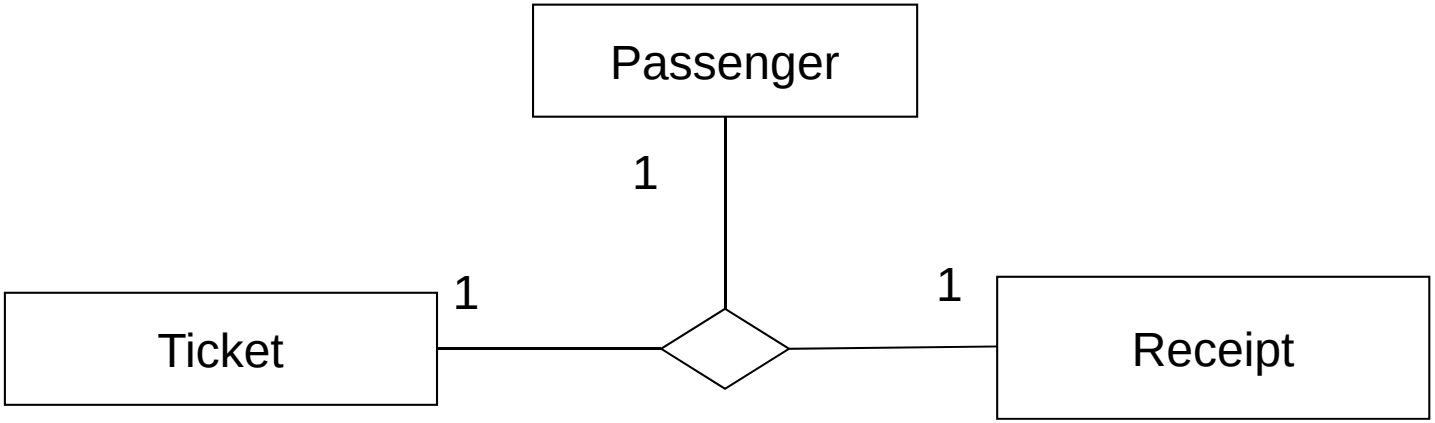

Student
name: String
studentID: String
1
*
Book
title: String
authors: StringList
One to many relationship
Student
name: String
studentID: String
Book title
1
1
Book
title: String
authors: StringList

More Examples



A class diagram

More Examples



A n-ary class diagram

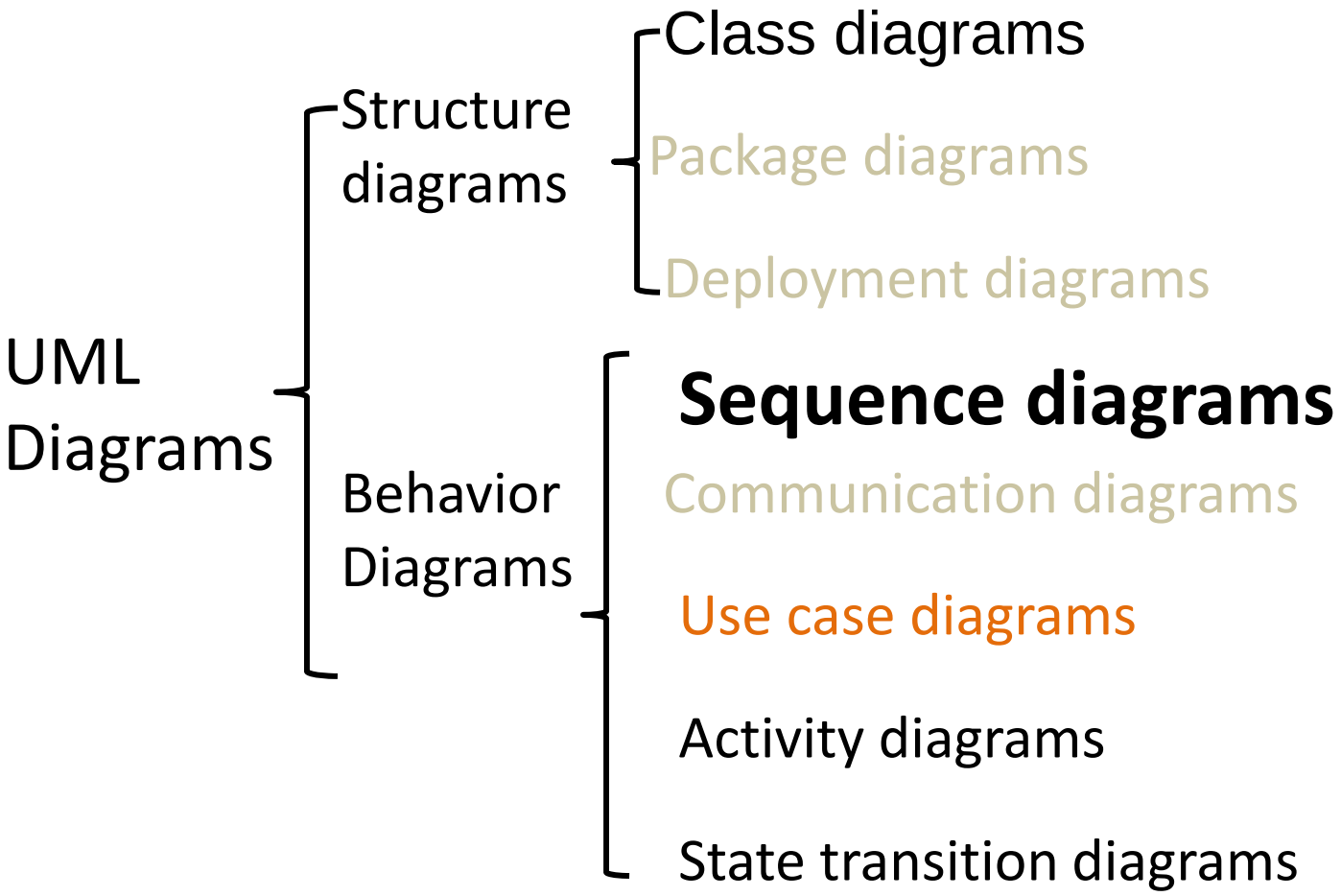
Class Exercises

- An online banking system handles accounts of different types: saving account, time deposit account and investment account.
  - For each account, it has an account number and a balance.
  - A time deposit account includes time deposits. Each time deposit has the start date, maturity date, interest rate and amount.
  - An investment account includes all the investments. Each investment has the type, units, unit price and total amount.
  - A bank customer can own at least one saving account.
  - Customer's information is recorded, including home address and phone number.

Class Exercises

- The money can be transferred from a saving account to a time deposit or an investment account. Each transfer contains transfer date, transfer amount.
- The money from the time deposit can be saved to a saving account at the maturity date.
- In an investment account, the units can be sold or bought. When buy, money is transferred from a saving account; when sold, the money is transferred to a saving account.

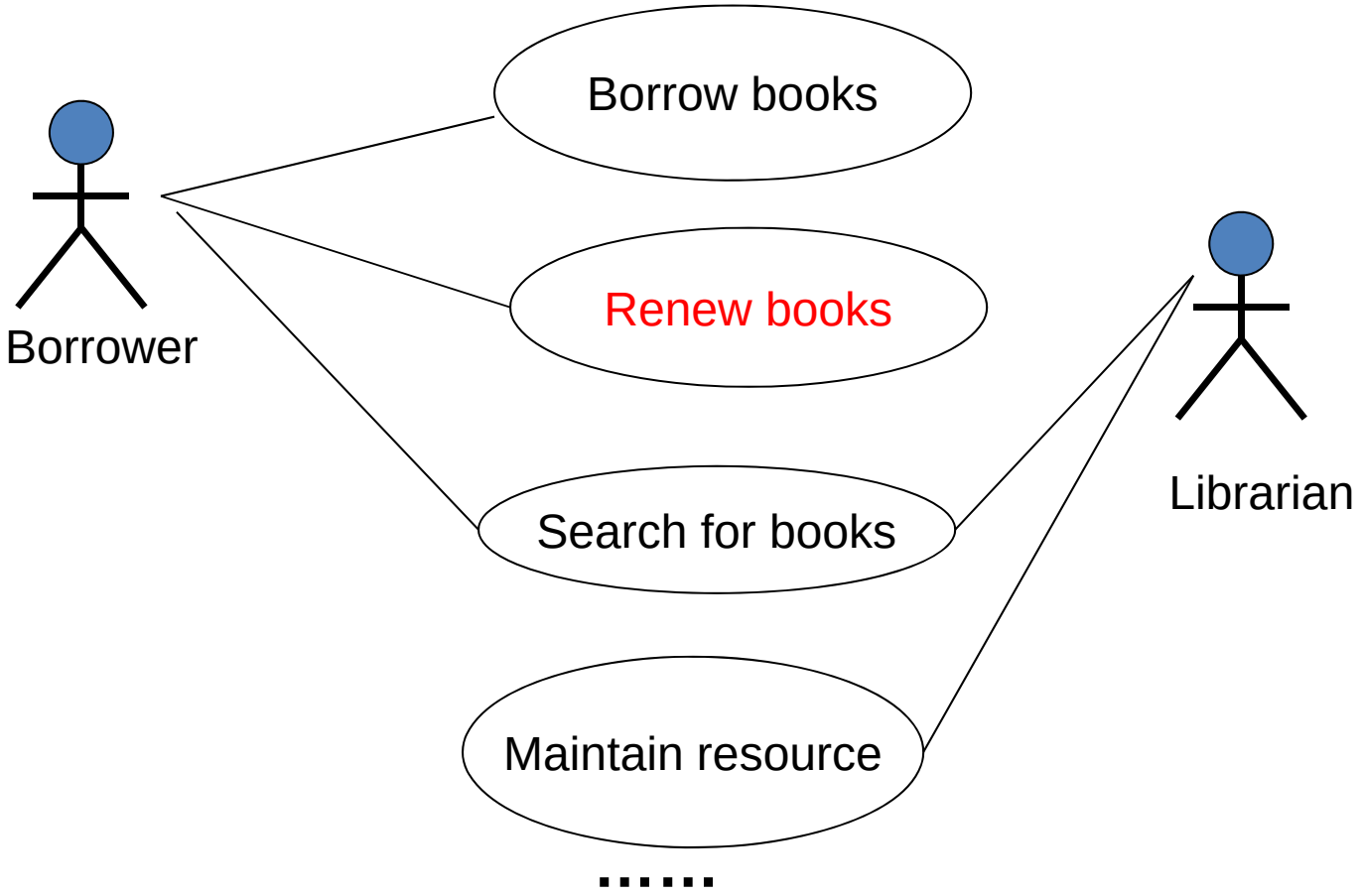
Unified Modeling Language (UML)



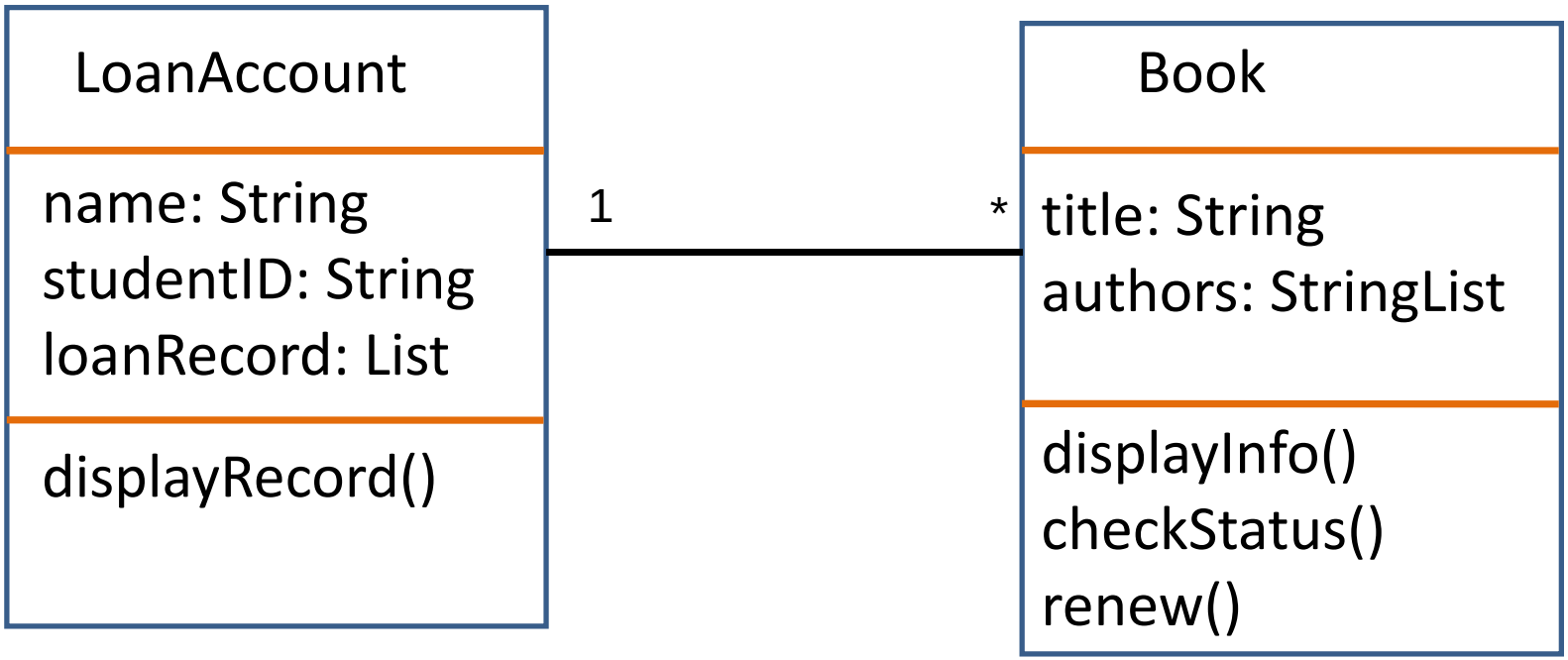
Sequence Diagrams

- For each user case, there is a sequence diagram, helping refine (细化) the use case
- Visualize （显现）the communications between objects
  - Horizontally （水平）among objects
  - Vertically （垂直）in time sequence

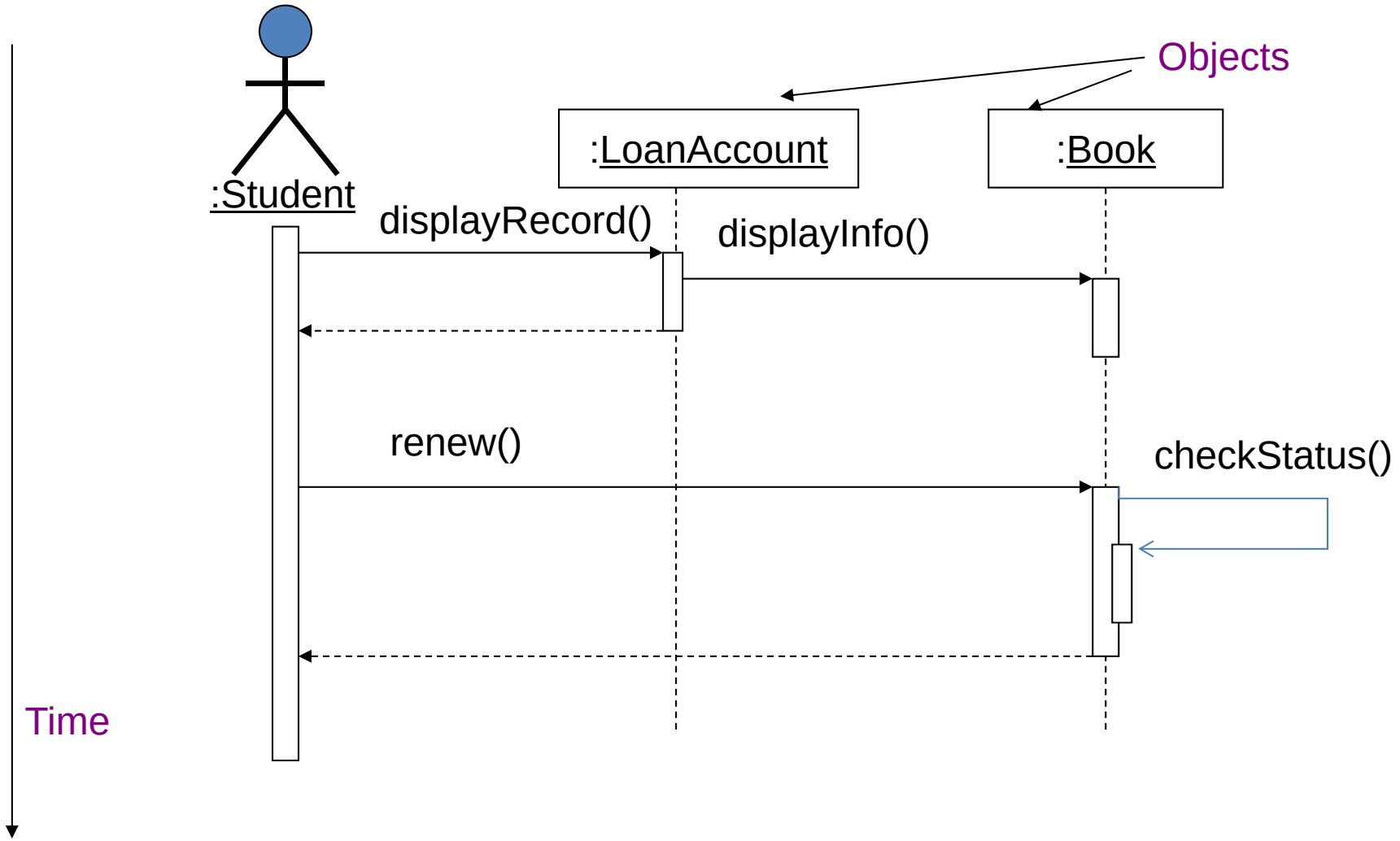
Use Case Diagrams - Library



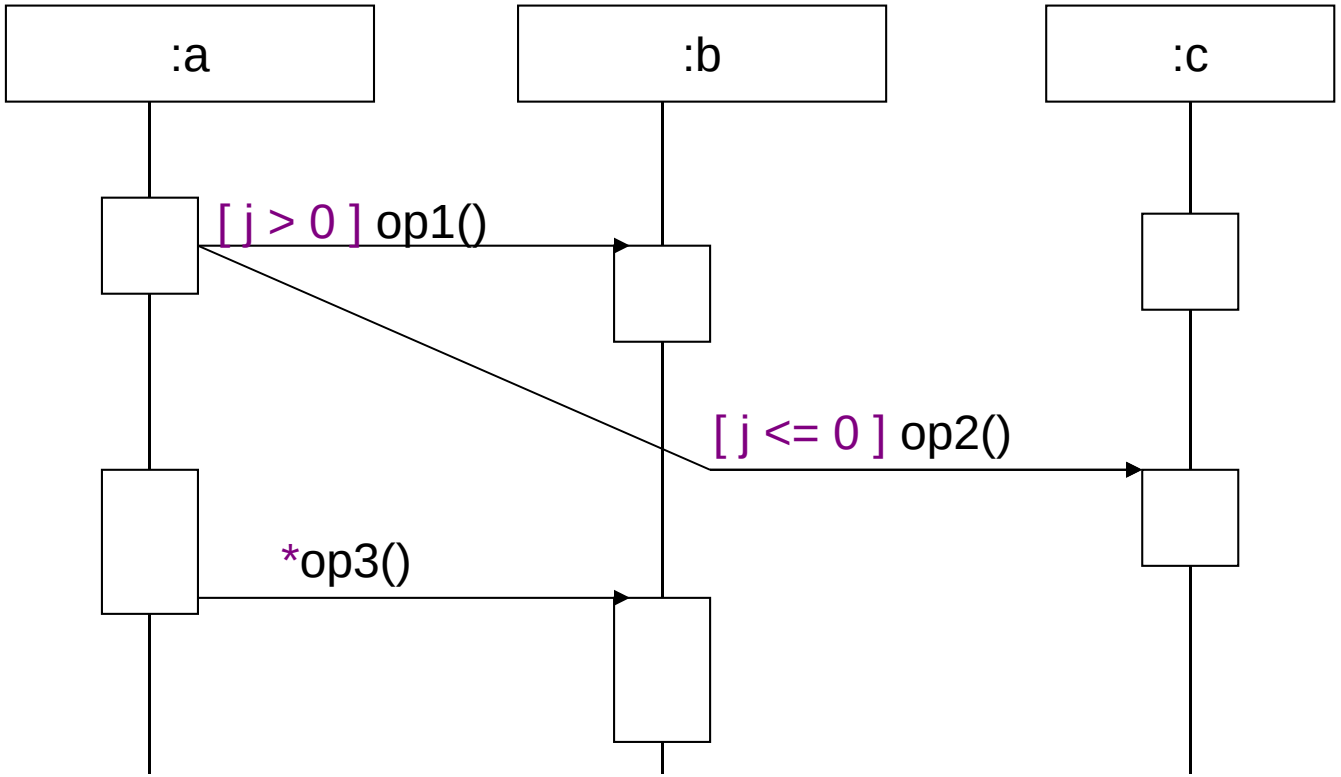
Object Classes in “Renew Book” Use Case



Sequence Diagrams



Sequence Diagrams

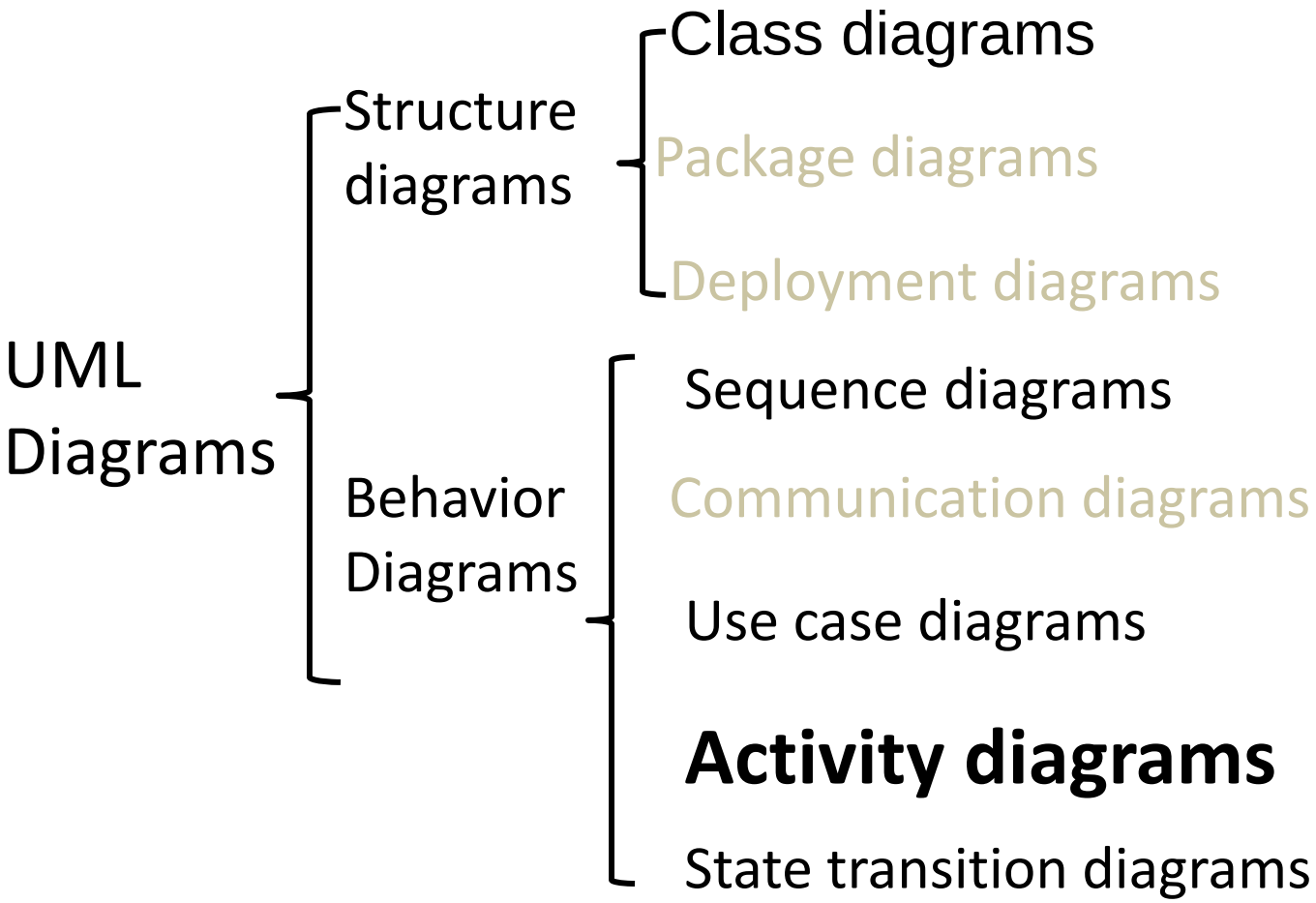


Conditions and iterations in a sequence diagram

Class Exercise

- Please draw the sequence diagram for the transfer between a saving account and a time deposit account. Three classes are involved in this use case: transfer class, saving account class and a time deposit class.

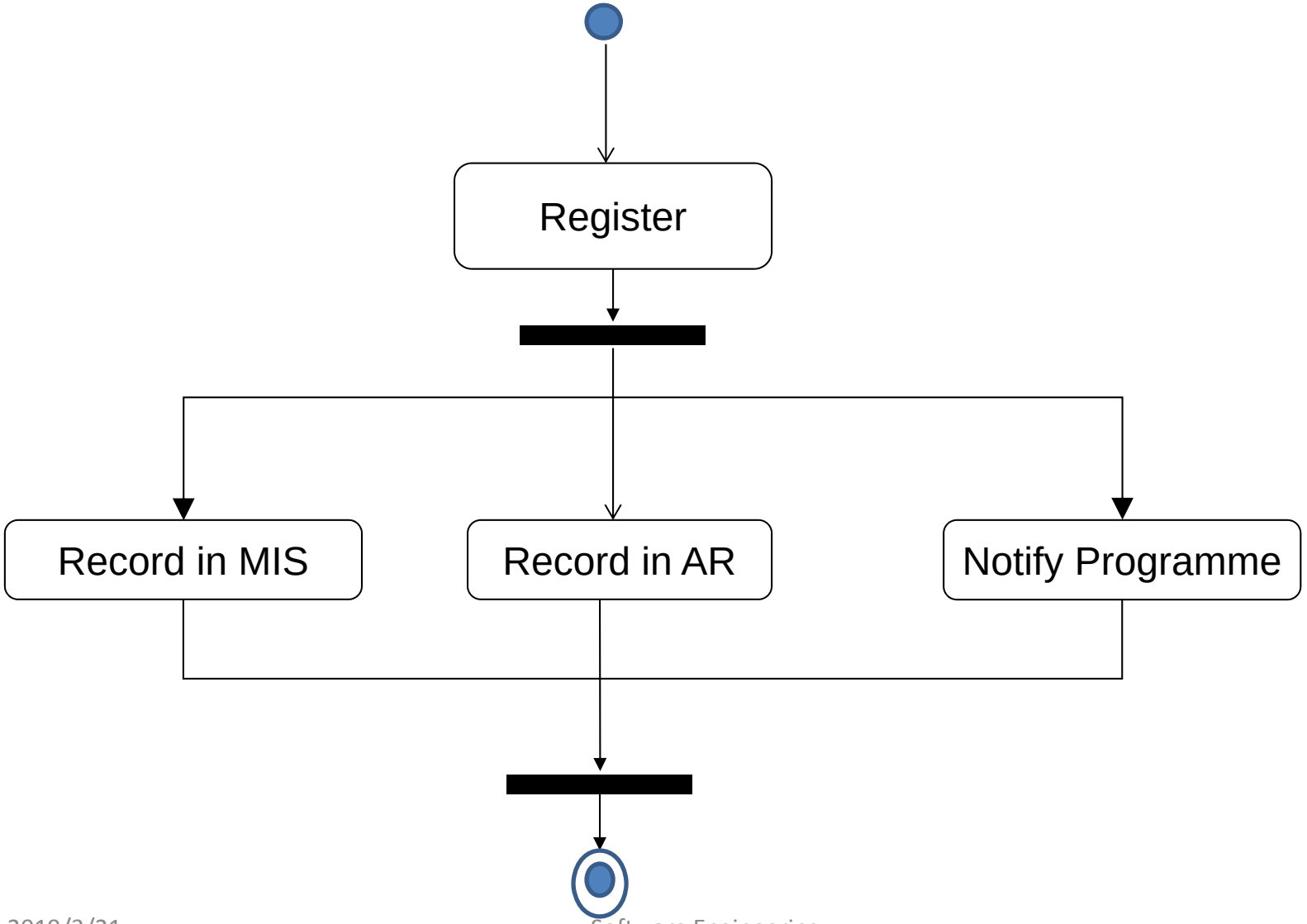
Unified Modeling Language (UML)



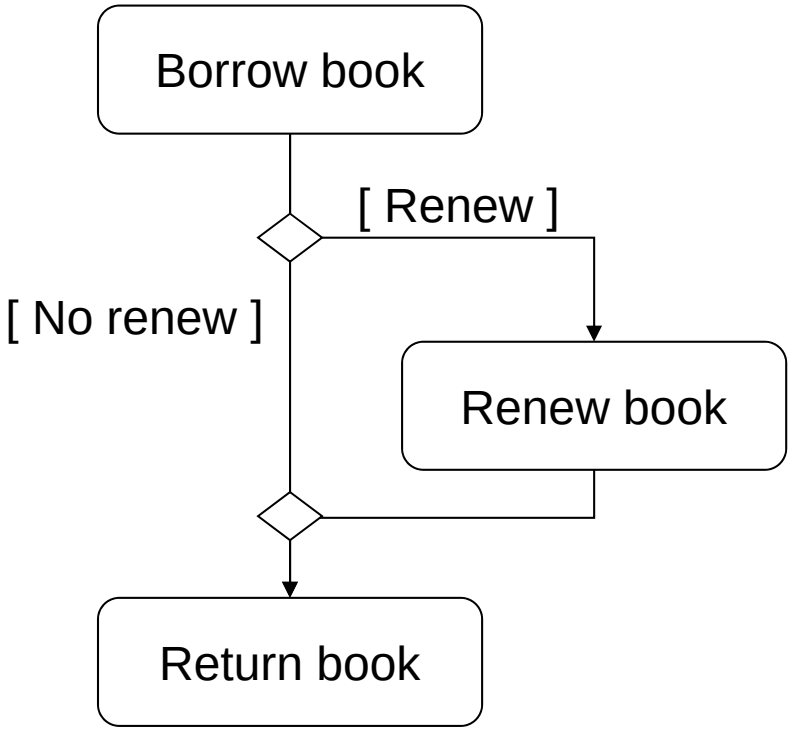
Activity Diagrams

- Describe sequence constraints among use cases
- Describe sequential activities among a group of objects or in an object
- Describe the tasks of a project
- Activity diagrams can describe from objects to the system
- Commonly contains:
  - Activity nodes
  - Flows

Activity Diagrams

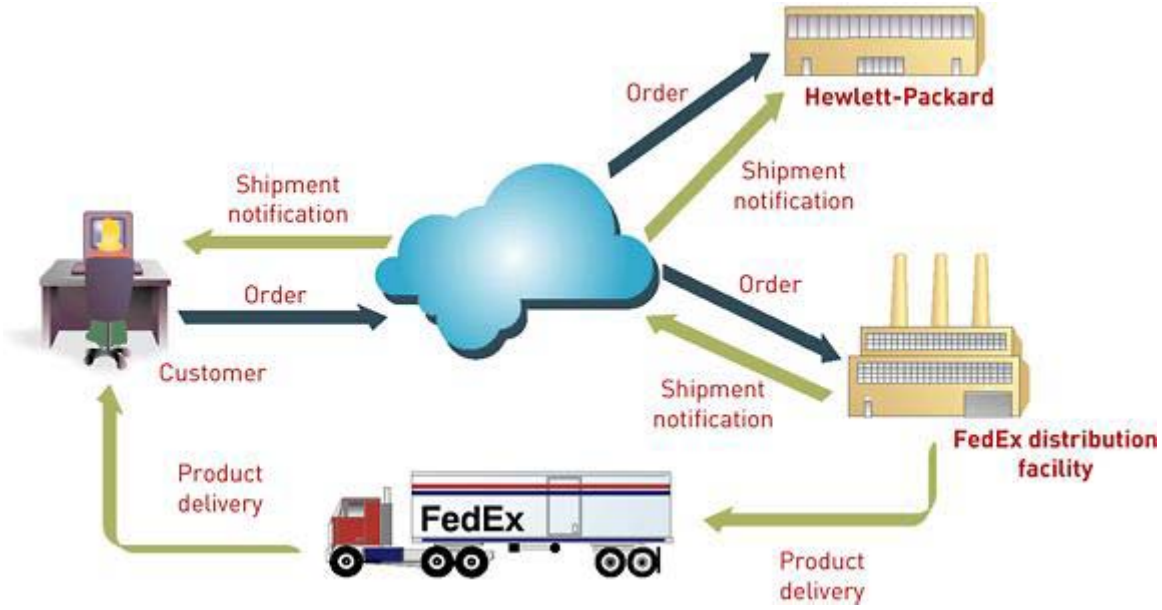


Activity Diagrams (Object Level)

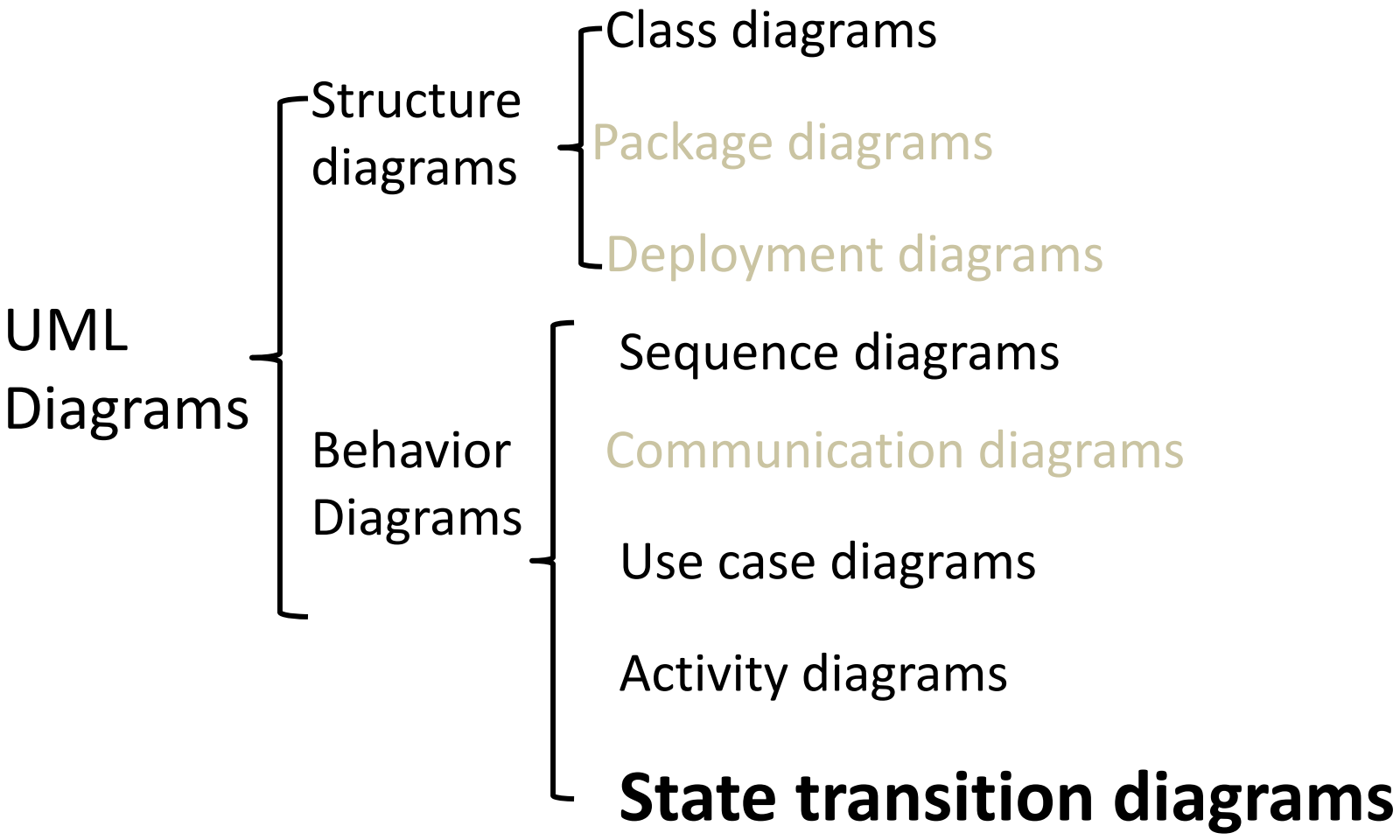


Class Exercise

- Please draw the activity diagram for the flow in the picture.



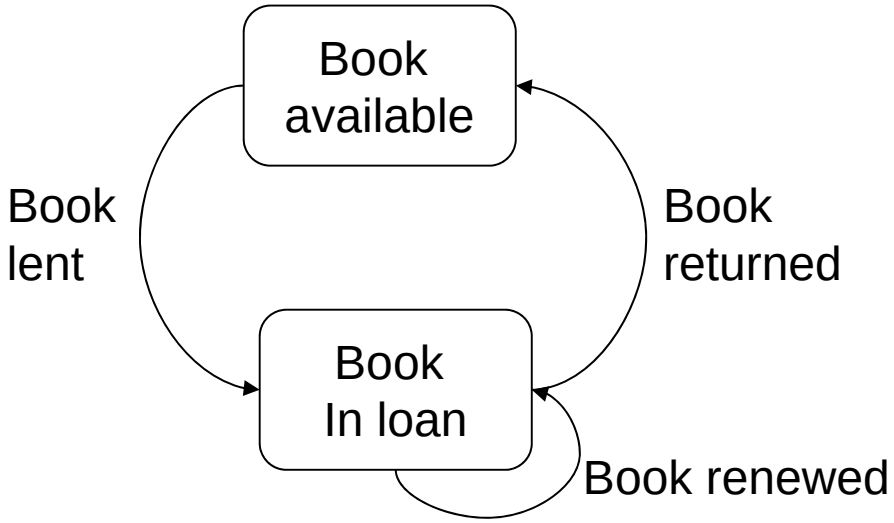
Unified Modeling Language (UML)



State Transition Diagrams

- Transition diagram can be described in multiple levels, from object to system.
- Each class has a state transition diagram
- It helps understand what events, actions and states are possible for an object
- State transition diagram can also be used to describe the whole system (see Lecture 4)

State Transition Diagrams (Object Level)



A state transition diagram for Book

Assignment 4

- Please draw the class diagram from the following description

  Assume that a system is developed for the booking of the rooms in a university.

  –There are four kinds of rooms: classroom, lecture hall, activity and lab. All the rooms have a room no. Lecture halls, classrooms, and lab have a number of available seats while each activity room offers some services (e.g., stereo, beverage, coffee, and tea). The equipment and services can be inquired.

  –Teacher can book any room while students can only book an activity room. When a student books a room, he should give the purpose and referee (i.e., the name of the teacher who supports the booking).

  –A person can book at most three rooms at the same time.

  –A teacher has a name, staff ID, office number, office phone number, and programme. Each student has a student name, student ID, mobile phone number and programme. All phone numbers can be updated.

  –there are two kinds of labs: computer lab and chemestry lab. Each is equipmented with quipments for lab practices.

- Deadline: see iSpace

Summary

- Class diagram
- Use case diagram
- State transition diagram
- Sequence diagram
- Activity diagram