



Object Oriented Analysis & Design

面向对象分析与设计

Lecture_02 建模工具UML

主讲: 姜宁康 博士



■ 4、类图...

- 软件设计的强力武器!
- 是一种静态视图

4.1 类图定义

■ 类class的定义

- 具有相同属性、操作、方法、关系或者行为的一组对象的描述符
- 类是真实世界事物的抽象
- 问题领域的类：在对系统建模时，将会涉及到如何识别业务系统中的事物，这些事物构成了整个业务系统。在UML中，把所有的这些事物都建模为类（class）

■ 对象object的定义

- 当这些事物存在于真实世界中时，它们是类的实例，并被称为对象
- 同一个类的各对象具有
 - 相同的属性，但属性的取值可以不同
 - 提供相同的操作、有相同的语义

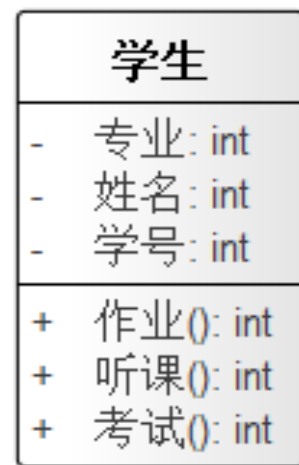
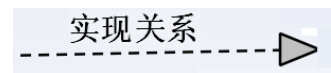
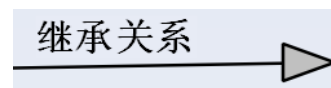
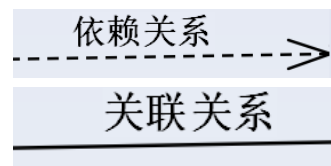
■ 把类相关的元素画在一起，即为类图

4.2 类图中常用的UML元素

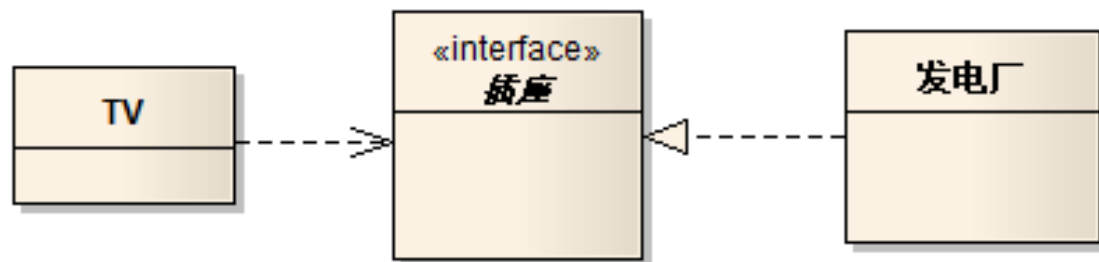
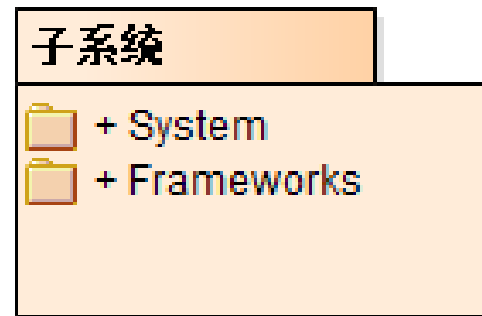
■ 类之间的关系

- 依赖关系
- 关联关系(association)
 - 关联的修饰
 - 名称
 - 角色
 - 多重性
 - 聚合/组合
 - 关联类
 - 连接 (link)
- 继承 | 泛化关系
 - 一般与具体
- 实现关系
 - 类与接口

■ 实例化



类



4.3 类图：问题的引出？

```
public class Person {  
    private String name;  
    private int yearOfBirth;  
  
    public Person(String theName, int theYear) {  
        name = theName;  
        yearOfBirth = theYear;  
    }  
  
    public void jump() {  
        System.out.println(name + " jumps");  
    }  
  
    public int getAge(int thisYear) {  
        return thisYear - yearOfBirth;  
    }  
}
```

创建对象：

```
Person aPerson = new Person("Sue Smith",25);
```

4.3 类图：问题的引出？

■ 一个软件系统

- 有多少个类？
- 每个类需要定义多少个属性？
- 每个类需要定义多少个操作？（方法）
- 类之间的关系？
- 类之间的信息交互、如何完成复杂功能的？

- 以上全部用代码来表示，说的清楚吗？
- 如果要建模，如何表示类、属性、操作、关系...

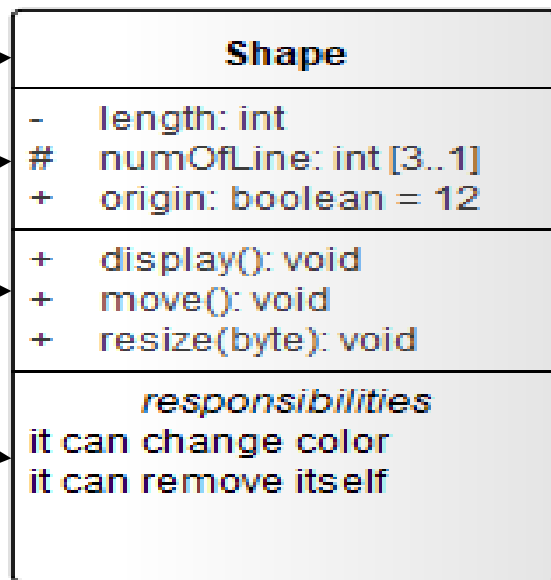
这是
面向对象
设计
要解决的问题

4.4 UML中类的表示

■ 类的图形表示

- 用四部分来表示，它是分为三个分隔区（compartment）的长方形

- 名称naming
- 属性attribute
- 操作operation
- 职责responsibility



- 在显示时，只有名称部分是必须的，其余部分可以隐藏（模型里面还是存在的！）



4.5 类元素的命名

■ 类的命名

- 类名中的每个词的第一个字母通常要大写

例如, Student、Teacher、Book、Store etc.

■ 操作的命名

- 操作名 除第一个词之外的每个词的第一个字母要大写

- 例如, selectCourse();

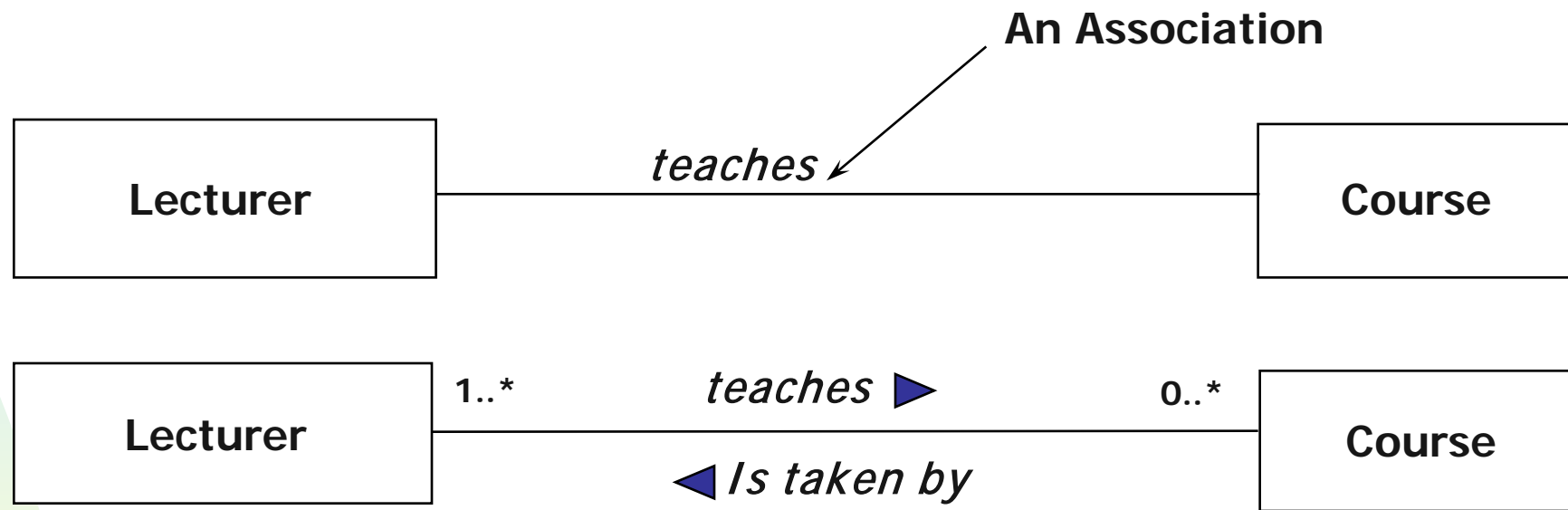
■ 属性的命名

- 属性名 除第一个单词之外的每个单词的第一个字母要大写

例如, myAddress、employeeID、stdNumber

4.6 关联关系的修饰

■ 1. 名称及其方向



■ 2. 角色

- 在关联的每一端，表明相连的类在该关联关系中的作用



- 可见性



4.6 关联关系的修饰

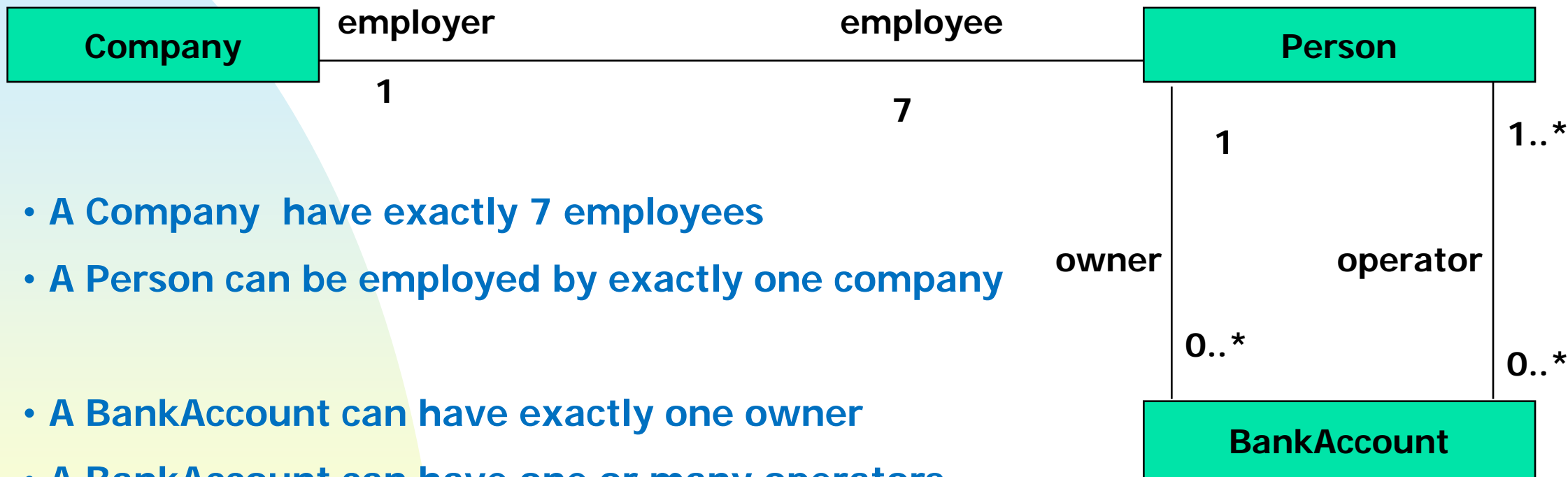
■ 3. 多重性 Multiplicity

- specifies the number of objects that can be involved in a relationship at any point in time
- No default multiplicity. If multiplicity is not explicitly stated then it is undecided

Adornment	Semantics
0..1	Zero or 1
1	Exactly 1
0..*	Zero or more
1..*	1 or more
1..6	1 to 6
1..3, 7..10	1 to 3 or 7 to 10
*	Zero or more

4.6 关联关系的修饰

■ 3. 多重性 Multiplicity



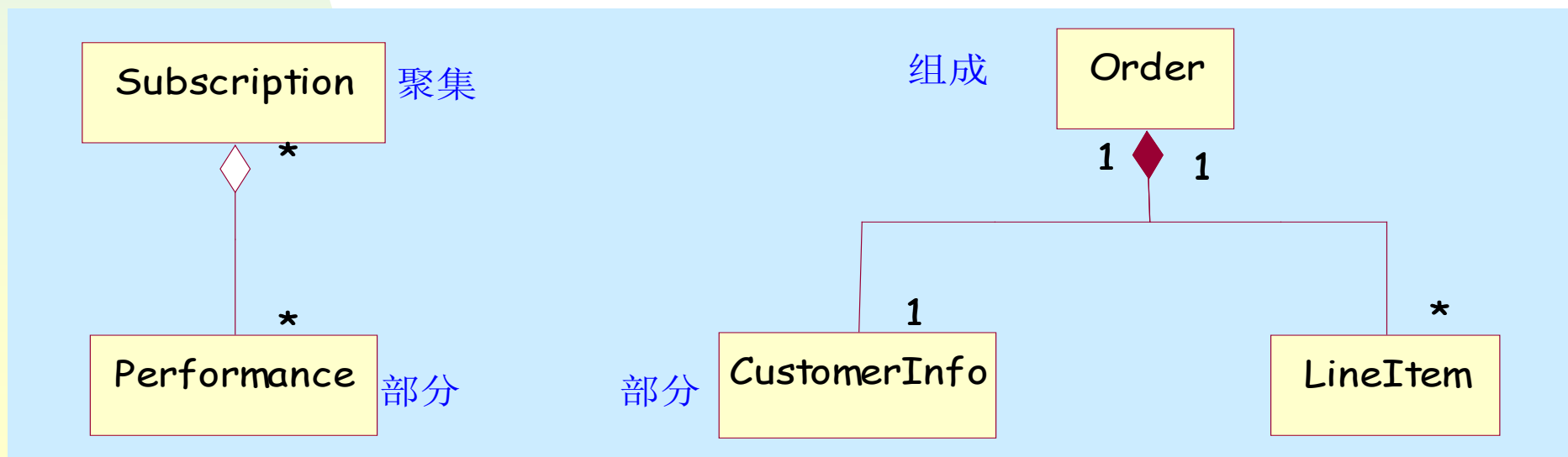
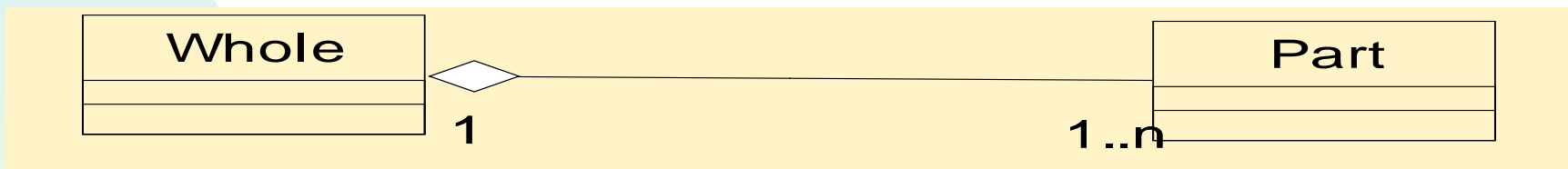
- A Company have exactly 7 employees
- A Person can be employed by exactly one company
- A BankAccount can have exactly one owner
- A BankAccount can have one or many operators
- A Person may have zero to many BankAccounts
- A Person may operate zero to many BankAccounts

4.6 关联关系的修饰

4. 聚合/组合：表示部分与整体的关联关系

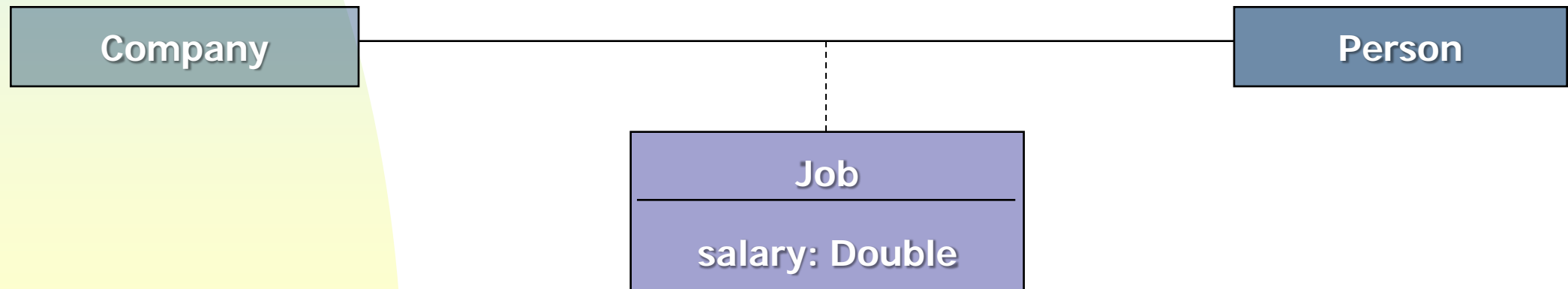
聚合：‘整体’有管理‘部分’的特有的职责；它用端点带有**空菱形**的线段表示，空菱形与‘整体’类相连接

组合：‘整体’拥有‘部分’的生命；它用端点带有**实菱形**的线段表示，实菱形与‘整体’类相连接
每个表示部分的类与表示整体的类之间有单独的关联，但是为了方便起见，连线结合在一起



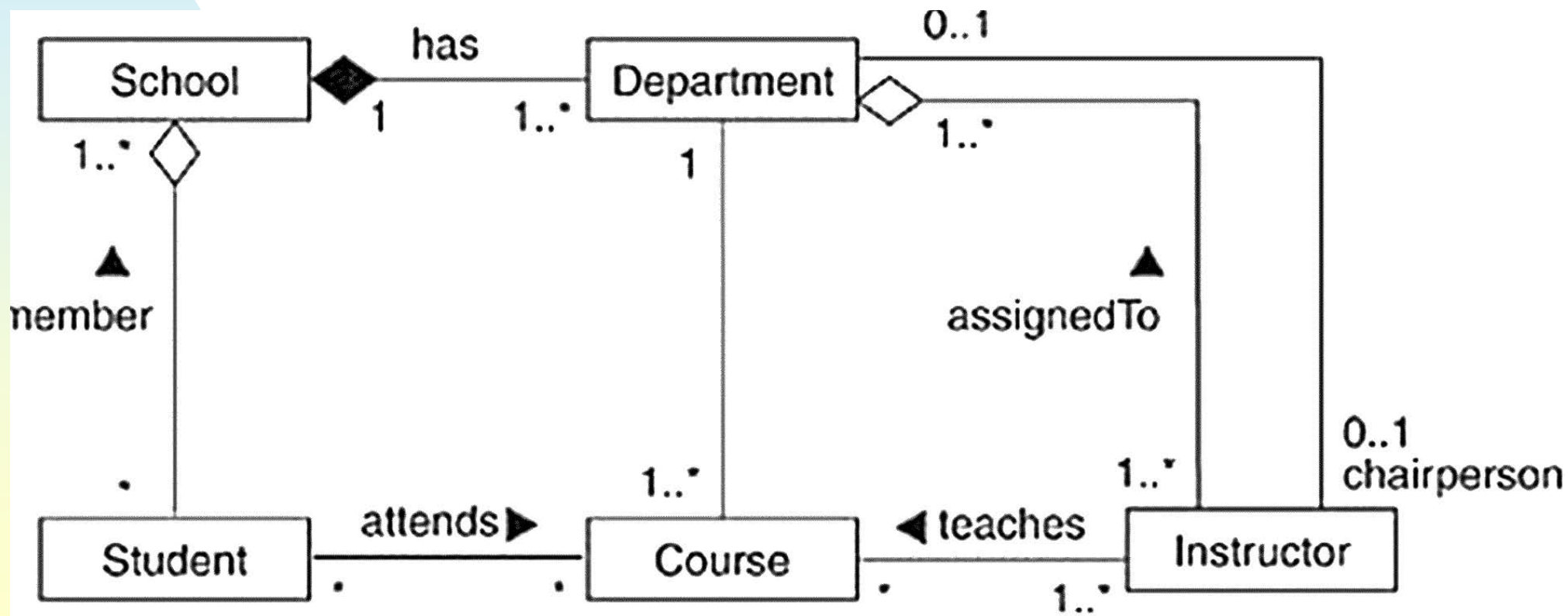
4.7 关联类 Association Class

- Association class is an association that is also a class, and consists of the class, association and the dashed line.
 - It may have attributes, operations and relationships
 - You can use an association class when there is exactly one unique link between any pair of objects at any point in time
 - If a pair of objects may have many links to each other at a given point in time, then you reify the relationship by replacing it with a normal class



4.8 案例：类图解读

- 能看懂
- 会画图



类图小结

■ 类

- 是面向对象分析设计的核心概念，也是难点
- 如何用UML工具来表示类及其相关概念，很重要

■ 但是，更重要的是

- 如何用面向对象的思维方式去思考，去发现类、定义类、定义类之间的关系、定义每个类应该承担的职责！





■ **本讲结束**