设计模式

2016年 秋季学期

何啸

联系方式

- 邮件：hexiao@ustb.edu.cn

- 办公室：机电楼826室

授课及考核方式

- 授课方式
  - 理论教学+课堂讨论
  - 经验研究+课堂汇报
- 考核方式
  - 平时成绩：40%
  - 期末考核：60%

主要内容

- 理论教学
  - 经典设计模式介绍（23个GoF模式+n个常用模式）
- 经验研究
  - 统计研究（包含但不限于）
    - 模式的使用情况
    - 模式与可维护性、可扩展性
    - 模式与软件错误
  - 案例研究（包含但不限于）
    - 模式的应用
    - 新模式的介绍

参考书目

- Design Patterns: Elements of Reusable Object-Oriented Software, by GoF

- Head First设计模式, by 弗里曼

计算的模式

例子

- 假设输入参数list是一个关于整数的数组，请实现下面的功能
  - 求所有项的和
  - 求最小项
  - 求最大连续子串和，及该子串
  - 抽取所有的奇数项

Foldl—计算的模式

- 如果定义foldl函数如下：

$$foldl\ f\ e\ x : xs = foldl\ f\ (f\ e\ x)\ xs$$

```
T foldl(BiFunction<T,E,T> f, T e, List<E> list) {
  if(list.isEmpty()) return e;
  final E x = list.get(0);
  final List<E> xs = list.subList(1, list.length());
  return foldl(f, f.apply(e,x),xs);
}
```

所有的解都可以写成这种形式！

解

- 假设输入参数list是一个关于整数的数组，请实现下面的功能
  - 求所有项的和

$$foldl\ (+)\ 0$$

  - 求最小项

$$foldl\ (min)\ +\infty$$

  - 求最大连续子串和，及该子串

$$foldl\ (\backslash(x,y)\ z\text{->}if\ sum(x) < sum(y) + z\ then\ (x : z, y : z)\ else\ if\ z > 0\ then\ (x, y : z)\ else\ (x, [])\ endif\ endif)\ ([], [])$$

  - 抽取所有的奇数项

$$foldl\ (\backslash x\ y\text{->}if\ odd(y)\ then\ x : y\ else\ x\ endif)\ []$$

抽象

- 计算对象的抽象
  - 类、结构、对象、实体
- 计算过程的抽象
  - 方法、函数
- 计算方法（解决方案）的抽象
  - （高阶）函数

软件模式（Software Pattern）

- 定义
  - 在软件设计中，一种针对某类常见问题所提出的通用的、可复用的解决方案
- 特点
  - 不是一个完整的软件设计，而是一种设计模版
  - 是一种“最佳实践”

软件模式

- 多个层次
  - 体系结构层
  - 类层、构件层
  - 算法层、函数层

设计模式

设计模式

- 从面向对象的视角：
- 一种针对对象、类交互关系的描述，是对在特定场景下出现的某种设计问题的通用解决方案

讨论

为什么需要设计模式?

设计模式的规约

- 名称
- 意图与动机
- 结构与实现
- 效果与副作用

设计模式的使用

- 分清模式的使用场景
- 权衡模式引入的好处和副作用
- 确定 / 划分参与类的角色
- 实现 / 重构代码

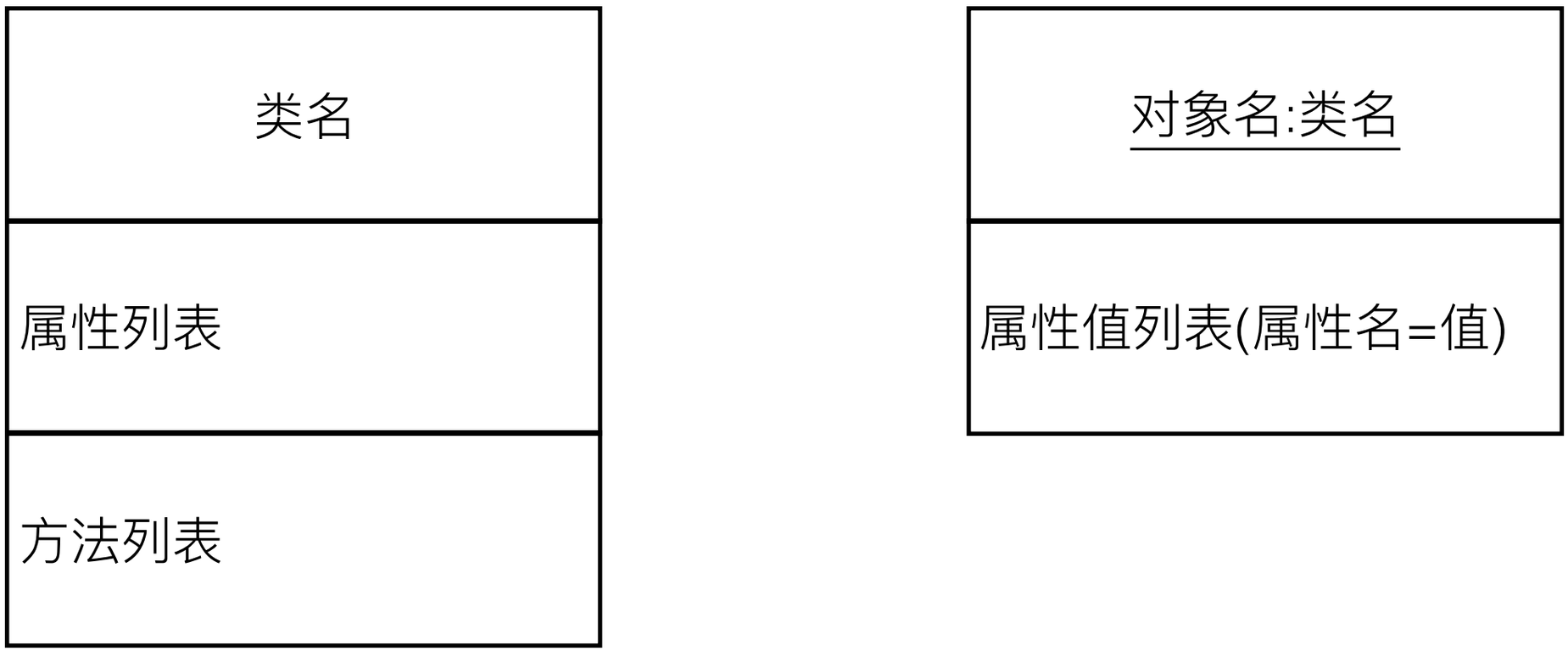
面向对象基础

面向对象

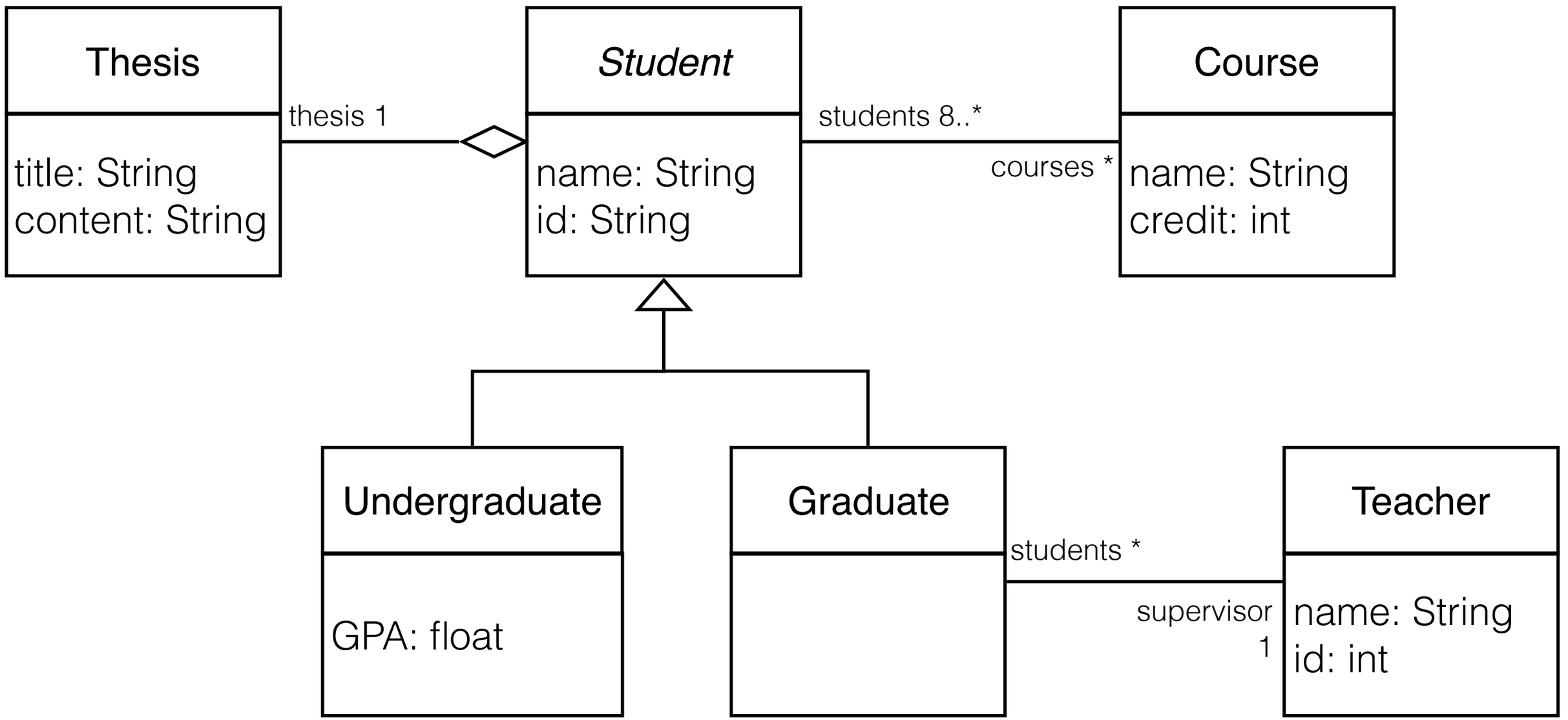
- 利用对象、类、属性、方法、继承、聚合、关联、消息等核心概念刻画和构造软件系统的方法和技术

- 基本设计原则
  - 抽象
  - 分类
  - 封装 / 信息隐藏
  - 高内聚、低耦合

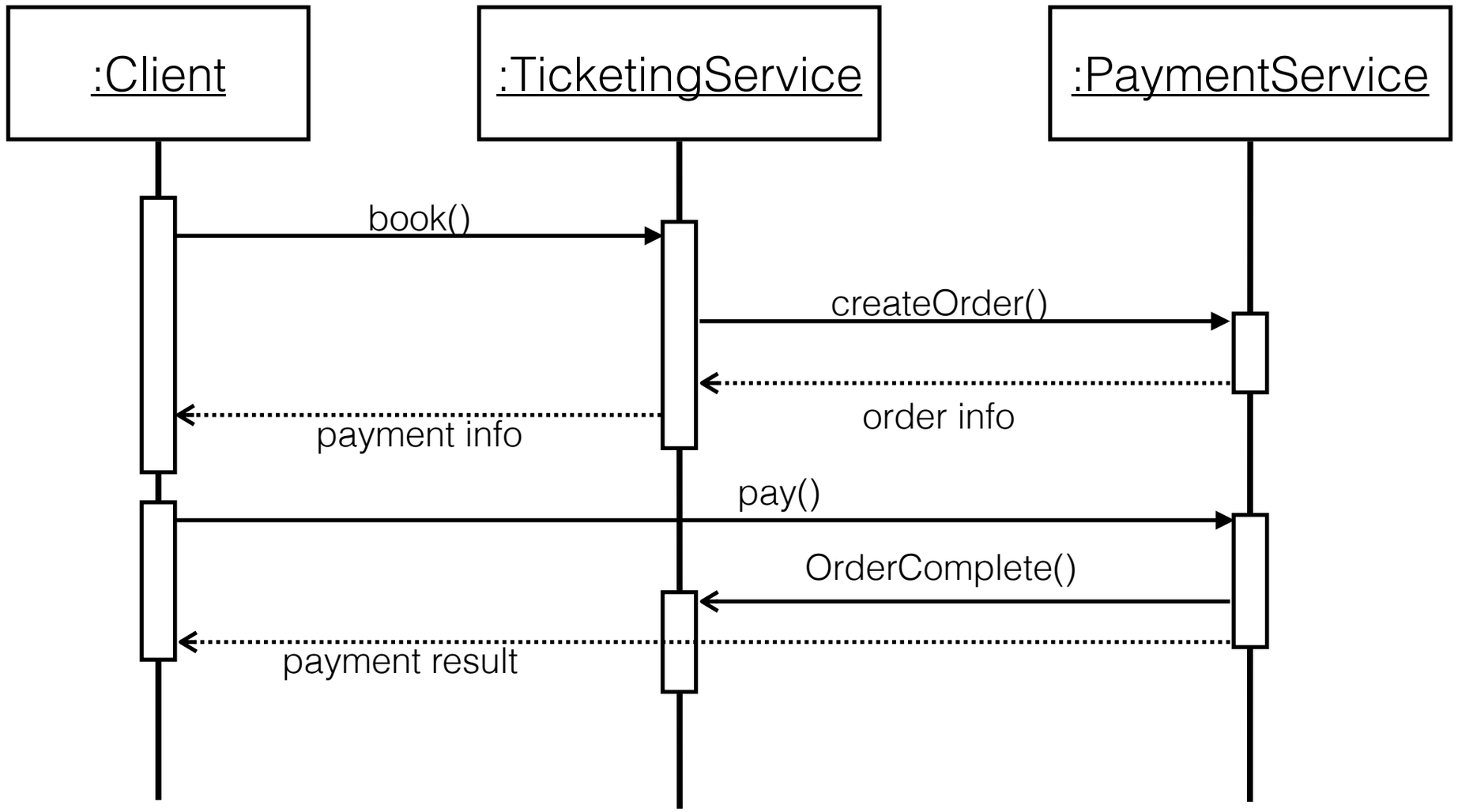
对象和类



继承、聚合、关联


Thesis
title: String
content: String
thesis 1
Student
name: String
id: String
students 8..*
courses *
Course
name: String
credit: int
Undergraduate
GPA: float
Graduate
students *
supervisor
1
Teacher
name: String
id: int

消息、生命线


:Client
:TicketingService
:PaymentService
book()
createOrder()
order info
payment info
pay()
OrderComplete()
payment result