



# 软件工程 *SPOC*

## 第八章 面向对象设计

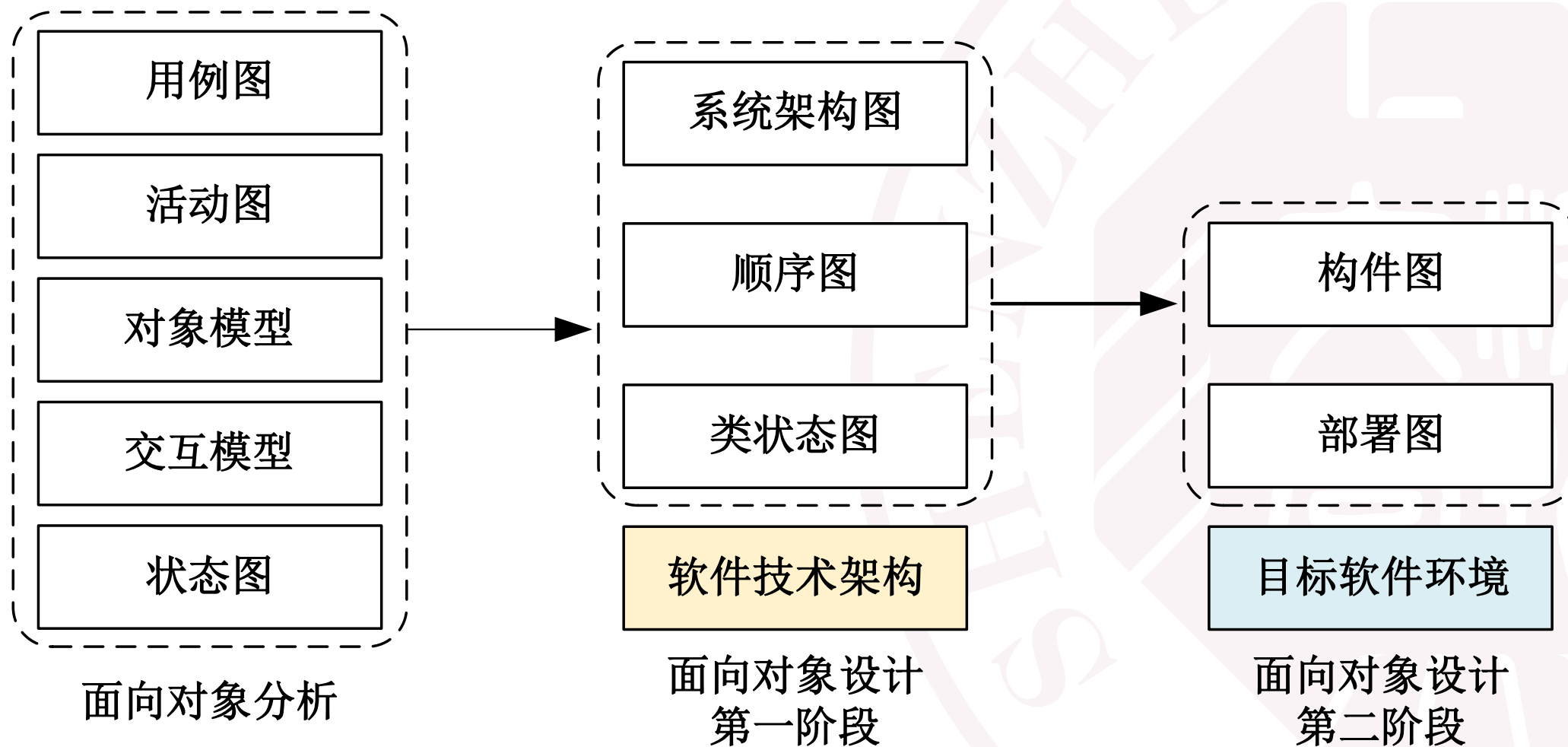
深圳大学 计算机与软件学院 杜文峰

# 问题思考?

分析阶段完成的分析模型能否直接用于软件开发?

面向对象分析的主要目的是理解用户需求，距离软件实现还存在一定的距离；在面向对象设计阶段，软件设计人员需要在面向对象分析的基础上完善软件设计，在原有分析模型的基础上补充与软件实现相关的信息。

# 1.分析模型到设计模型的转换



## 2.面向对象设计原则

### 软件体系架构设计内容

- ◆ 确定目标软件中的类，以及类和类之间的关系；
- ◆ 确定目标软件中类的属性和方法；
- ◆ 确定目标软件中各个构件的组成，以及构件之间的相互关系；
- ◆ 确定各个构件在目标系统中的部署方式。

## 2.1 面向对象的模块独立性

### [1] 内聚

- 属性内聚
- 方法内聚

### [2] 耦合

- 依赖关系
- 关联关系
- 聚合关系
- 组合关系
- 继承关系

## 2.2 面向对象设计启发式规则

- 开放-封闭原则
- 单一职责原则
- Liskov替换原则
- 接口隔离原则
- 迪米特法则/最少知道原则
- 合成/聚合复用原则
- 依赖倒置原则

## 2.3 设计模式

GoF中的设计模式分为创建型模式、结构型模式和行为型模式三种。



## 2.3 设计模式

### [1] 创建型模式

单例模式、原型模式、工厂方法模式、抽象工厂模式和建造者模式

### [2] 结构型模式

代理模式、适配器模式、桥接模式、装饰模式、外观模式、享元模式、组合模式

### [3] 行为型模式

模板方法模式、策略模式、命令模式、职责链模式、状态模式、观察者模式、中介者模式、迭代器模式、访问者模式、备忘录模式、解释器模式



## 3.1 软件架构设计

---

- [1] 确定软件架构模型
- [2] 确定分析模型中类候选者的类型
- [3] 将类候选者加入软件架构框架
- [4] 增加实施过程中必须的其它类
- [5] 软件系统架构优化

## 3.2 类方法设计及类属性设计

充分利用顺序图、通信图、状态图等完成面向对象设计中的类方法设计和类属性设计

## 4.软件构件设计

构件是指可复用的软件组成内容，可以是被封装的对象类、类树、功能模块〔动态链接库、静态链接库〕、软件框架〔*Framwork*〕、软件构架、资源、文档等。

- 构件是在软件生命周期各阶段可以被复用的软件实体。
- 软件构件是具有明确接口的、严格封装的、可被组装的软件制品，它可以不加修改或者基本不加修改地作为一个部件和其它构件一起组装成更大的软件或软件成分。

## 4.1 构件设计原则

---

[1] 有用性

[2] 可用性

[3] 自描述性

[4] 可定制性

[5] 可移植性

## 4.2 软件构件的设计方法

在面向对象设计过程中，软件设计人员可以从主题、功能和资源类型等角度出发来设计和封装构件。

软件设计人员可以根据需要在构件架构中增加协调类，通过协调类来组织构件的其它组成部分来向外提供服务。

## 5.部署设计

---

- [1] 单机模式
- [2] 客户端/服务器模式
- [3] 中间件
- [4] 分布式架构
- [5] 云计算架构模式

## 6. 小结

由于面向对象分析与设计采用相同的建模方法，软件设计人员可以直接从面向对象分析过渡到面向对象设计，并结合实现需要对面向对象分析结果进行补充。

软件设计人员可以结合目标软件需求，参考或者选择经典的软件架构框架来设计适当的软件架构。

## 6. 小结

软件架构模型确定后，软件设计人员可以按照架构模型要求将分析模型中的类添加到目标软件架构中。同时，软件设计人员可以根据需要，在软件架构中增加实现必须的类。

通过实现目标软件的各种用例需求，软件设计人员可以借助交互分析和状态分析，对类候选者的方法和属性进行补充，并对得到的软件架构进行检验。

除了对类架构进行设计以外，软件设计人员也必须结合软件实现需求对目标系统的构件和部署方式进行设计。