

面向对象设计方法

面向对象设计概述

- 软件设计的复杂性：固有的复杂性导致通讯困难、费用超支、时间延期等问题。
- 控制软件复杂性的手段：结构化设计方法、功能分解、抽象、模块化、信息隐藏。
- 结构化设计方法存在的问题：功能与数据分离设计、难以维护、限制软件的可重用性。

基于UML的面向对象设计

- 运用面向对象概念构造系统设计模型，从面向对象分析模型构造设计模型。
- 建模语言：统一建模语言（UML），支持面向对象。

面向对象设计遵循的原则

- 类、对象、封装和信息隐藏、消息、继承、多态性和动态连编、对象之间的联系。
- 分类结构、组成结构、实例连接、消息连接。

面向对象设计的迭代式增量

- 面向对象设计的前提：面向对象分析阶段的制品。
- 设计任务：通过设计将OOA模型精化成OOD模型，并补充实现有关的部分。

描述系统的视角

- 系统的使用实例、逻辑结构、构成、并发性、配置。

4+1视图方法

- 不同架构视图承载不同的系统分析与设计决策，支持不同的目标和用途。

面向对象设计阶段的建模

- 功能、结构、行为、实现、时间等方面的建模。

设计原则

- 模块化、层次分解、耦合、内聚、复用、体系结构、模式。

设计过程

- 确定分析到设计的映射、构造结构模型、行为模型、体系结构模型。

设计类

- 考虑与实现有关的因素，具体描述操作的参数、属性和类型等。

类的精化设计

- 明确定义操作的参数和基本的实现逻辑、属性的类型和可见性、类之间的关系。

基于UML建模的面向对象设计

- 定义组件、构造部署图、设计组件（子系统）与接口。

案例分析

- 系统需求、用例图、在用例场景中选择对象、确定对象交互、设计类以及类之间的关系、确定对象行为的状态图、体系结构设计。