

设计模式原则 - 合成复用原则

寂然

官方定义

合成复用原则（Composite Reuse Principle）的官方定义如下：

Try to use composition / aggregation instead of inheritance

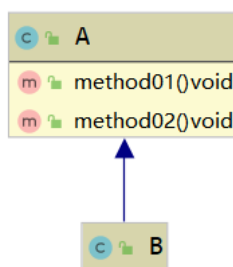
（尽量使用组合/聚合的方式，而不是使用继承）

基本介绍

案例演示 - 如何复用？

现在假设有一个A类，里面有两个方法，有一个类B，想要复用这两个方法，请问有几种方案

方式一：继承



方式二：组合或者聚合？（业务逻辑上）



方式三：依赖关系



案例总结

设计原则总结

- **开闭原则**：要求对扩展开放，对修改关闭
- **里氏替换原则**：不要破坏继承体系
- **依赖倒置原则**：要求面向接口编程
- **单一职责原则**：实现类职责要单一
- **接口隔离原则**：在设计接口的时候要精简单一
- **迪米特法则**：只与直接的朋友的通信
- **合成复用原则**：尽量使用聚合和组合的方式，而不是使用继承

设计原则的核心思想

- 找出应用中可能需要变化之处，独立出来，不要和不需要变化的代码混在一起
- 针对接口编程，而不是针对实现编程
- 为了交互对象的松耦合设计而努力

遵循设计原则:就是为了让程序高内聚，低耦合

下节预告

UML类图

