

# 合成复用原则

2020年2月24日 12:14

## 1. 定义

合成复用原则 (Composite Reuse Principle, CRP) 又叫组合/聚合复用原则 (Composition/Aggregate Reuse Principle, CARP)。它要求在软件复用时, 要尽量先使用组合或者聚合等关联关系来实现, 其次才考虑使用继承关系来实现。如果要使用继承关系, 则必须严格遵循里氏替换原则。合成复用原则同里氏替换原则相辅相成的, 两者都是开闭原则的具体实现规范。

## 2. 重要性

通常类的复用分为**继承复用**和**合成复用**两种, 继承复用虽然有简单和易实现的优点, 但继承复用存在以下缺点。

1. 继承复用破坏了类的封装性。因为继承会将父类的实现细节暴露给子类, 父类对子类是透明的, 所以这种复用又称为“白箱”复用。
2. 子类与父类的耦合度高。父类的实现的任何改变都会导致子类的实现发生变化, 这不利于类的扩展与维护。
3. 它限制了复用的灵活性。从父类继承而来的实现是静态的, 在编译时已经定义, 所以在运行时不可能发生变化。

采用组合或聚合复用时, 可以将**已有对象纳入新对象中, 使之成为新对象的一部分**, 新对象可以调用已有对象的功能, 它有以下优点。

1. 它维持了类的封装性。因为成分对象的内部细节是新对象看不见的, 所以这种复用又称为“黑箱”复用。
2. 新旧类之间的耦合度低。这种复用所需的依赖较少, 新对象存取成分对象的唯一方法是通过成分对象的接口。
3. 复用的灵活性高。这种复用可以在运行时动态进行, 新对象可以动态地引用与成分对象类型相同的对象。

## 3. 实现方法

合成复用原则是通过将已有的对象纳入新对象中, 作为新对象的成员对象来实现的, 新对象可以调用已有对象的功能, 从而达到复用。