

# 接口隔离原则

2020年2月24日 14:39

## 1. 定义

客户端不应该依赖它不需要的接口；一个类对另一个类的依赖应该建立在最小的接口上。接口隔离原则是：建立单一接口，不要建立庞大臃肿的接口，尽量细化接口，接口中的方法尽量少。也就是说，我们要为各个类建立专用的接口，而不要试图去建立一个很庞大的接口供所有依赖它的类去调用。

## 2. 问题由来

类A通过接口I依赖类B，类C通过接口I依赖类D，如果接口I对于类A和类C来说不是最小接口，则类B和类D必须去实现他们不需要的的方法。

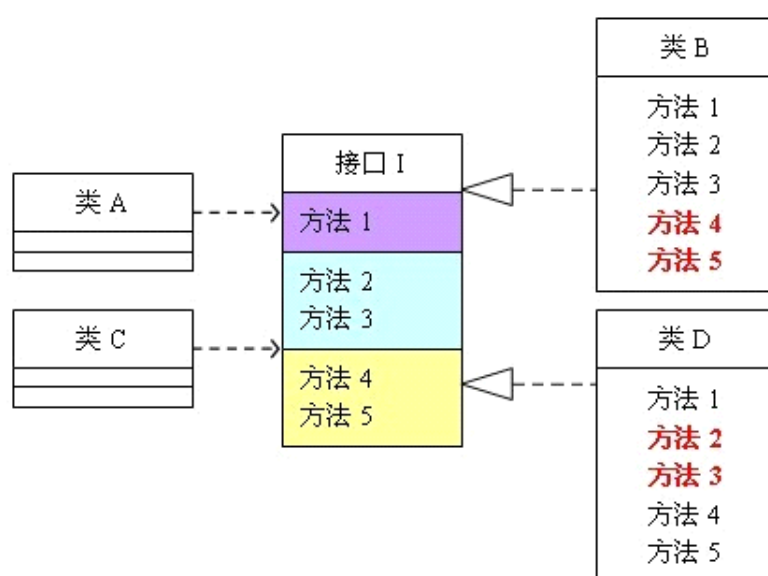


图2.1 接口隔离原则

这个图的意思是：类A依赖接口I中的方法1、方法2、方法3，类B是对类A依赖的实现。类C依赖接口I中的方法1、方法4、方法5，类D是对类C依赖的实现。对于类B和类D来说，虽然他们都存在着用不到的方法（也就是图中红色字体标记的方法），但由于实现了接口I，所以也必须实现这些用不到的方法。

## 3. 解决方案

将臃肿的接口I拆分为独立的几个接口，类A和类C分别与他们需要的接口建立依赖关系。

## 4. 与单一职责区别

1. 单一职责原则原注重的是职责；而接口隔离原则注重对接口依赖的隔离。
2. 单一职责原则主要是约束类，其次才是接口和方法，它针对的是程序中的实现和细节；而接口隔离原则主要约束接口接口，主要针对抽象，针对程序整体框架的构建。

采用接口隔离原则对接口进行约束时，要注意以下几点：

- 接口尽量小，但是要有限度。对接口进行细化可以提高程序设计灵活性是不争的事实，但是如果过小，则会造成接口数量过多，使设计复杂化。所以一定要适度。
- 为依赖接口的类定制服务，只暴露给调用的类它需要的方法，它不需要的方法则隐藏起来。只有专注地为一个模块提供定制服务，才能建立最小的依赖关系。
- 提高内聚，减少对外交互。使接口用最少的方法去完成最多的事情。

运用接口隔离原则，一定要适度，接口设计的过大或过小都不好。