

设计模式:Java设计模式之计数代理模式 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E8_AE_BE_E8_AE_A1_E6_A8_A1_E5_c104_144669.htm 计数代理模式在客户对象调用服务提供者对象上方法的前后执行诸如日志

(logging) 和计数 (counting) 一系列附加功能时很有用。计数代理模式建议把这些附加功能封装在一个单独的对象，这个对象就是指计数代理对象，而不是把这些附加的功能实现放到服务提供者的内部。良好的对象设计的一个特征就是对象要专注于提供特定的功能。换句话说，理想的对象不应该做各种不相干的事情。把诸如日志 (logging) 和计数

(counting) 等类似的功能封装为一个单独的对象，而让服务提供者对象仅提供它自己的特定功能。也就是说，只允许服务提供者对象执行定义良好、特定的任务。计数代理被设计成可以被客户访问的与服务提供者具有相同接口的对象。客户对象不是直接访问服务提供者，而是调用计数代理对象上的方法，计数代理执行必要的纪录日志 (logging) 和计数 (counting) 功能后，再把方法调用传递给服务提供着对象。

如图1 2828o18m1i2q Figure1: Generic Class Association When the Counting Proxy Pattern Is Applied 下面的例子说明了如何在应用程序中利用计数代理。

例子：让我们设计一个Order类，类层次如图2，OrderIF接口声明了getAllOrders读取数据库中所有订单的简单方法。 q0hxcf0kyz9y Figure2: Order Class

Hierarchy public interface OrderIF { public Vector getAllOrders(). }

作为getAllOrders方法实现的一部分，Order类实用了FileUtil工具类从order.txt文件中读取订单项。 public class Order

```
implements OrderIF { public Vector getAllOrders() { FileUtil fileUtil  
= new FileUtil(). Vector v = fileUtil.fileToVector("orders.txt").  
return v. } }
```

100Test 下载频道开通，各类考试题目直接下载。
详细请访问 www.100test.com