

深入浅出基于Java的责任链模式 PDF转换可能丢失图片或格式  
，建议阅读原文

[https://www.100test.com/kao\\_ti2020/227/2021\\_2022\\_\\_E6\\_B7\\_B1\\_E5\\_85\\_A5\\_E6\\_B5\\_85\\_E5\\_c97\\_227671.htm](https://www.100test.com/kao_ti2020/227/2021_2022__E6_B7_B1_E5_85_A5_E6_B5_85_E5_c97_227671.htm) 计算机等级考试训练软件《百宝箱》

一、引言 初看责任链模式，心里不禁想起了一个以前听过的相声：看牙。说的是一个病人看牙的时候，医生不小心把拔下的一个牙掉进了病人嗓子里。病人因此楼上楼下的跑了好多科室，最后无果而终。责任链模式就是这种“推卸”责任的模式，你的问题在我这里能解决我就解决，不行就把你推给另一个对象。至于到底谁解决了这个问题了呢？我管呢！

二、定义与结构 从名字上大概也能猜出这个模式的大概模样系统中将会存在多个有类似处理能力的对象。当一个请求触发后，请求将在这些对象组成的链条中传递，直到找到最合适的“责任”对象，并进行处理。《设计模式》中给它的定义如下：使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。将这些对象连成一条链，并沿着这条链传递该请求，直到有一个对象处理它为止。从定义上可以看出，责任链模式的提出是为了“解耦”，以应变系统需求的变更和不明确性。下面是《设计模式》中给出的适用范围：

- 1) 有多个的对象可以处理一个请求，哪个对象处理该请求运行时刻自动确定。
- 2) 你想在不明确指定接收者的情况下，向多个对象中的一个提交一个请求。
- 3) 可处理一个请求的对象集合应被动态指定。

责任链模式真的能给发送者和接收者之间解耦（这好像很神奇）吗？先来看下它的组成角色。这个问题我会在下面提及。责任链模式由两个角色组成：

- 1) 抽象处理者角色（Handler）：它定义

了一个处理请求的接口。当然对于链子的不同实现，也可以在这个角色中实现后继链。2) 具体处理者角色 ( Concrete Handler ) : 实现抽象角色中定义的接口，并处理它所负责的请求。如果不能处理则访问它的后继者。至于类图不放也罢。毕竟就是一个继承或者实现。三、纯与不纯 责任链模式的纯与不纯的区别，就像黑猫、白猫的区别一样。不要刻意的去使自己的代码来符合一个模式的公式。只要能够使代码降低耦合、提高重用，满足系统需求并能很好的适应变化就好了。正所谓：管它黑猫白猫，抓住老鼠就是好猫！纯的责任链模式，规定一个具体处理者角色只能对请求作出两种动作：自己处理；传给下家。不能出现处理了一部分，把剩下的传给了下家的情况。而且请求在责任链中必须被处理，而不能出现无果而终的结局。反之，则就是不纯的责任链模式。不纯的责任链模式还算是责任链模式吗？比如一个请求被捕获后，每个具体处理者都尝试去处理它，不管结果如何都将请求再次转发。我认为这种方式的实现，算不算是责任链模式的一种倒不重要，重要的是我们也能从中体味到责任链模式的思想：通过将多个处理者之间建立联系，来达到请求与具体的某个处理者的解耦。下面的例子就是采用了上面提到的“不纯的责任链模式”。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](http://www.100test.com)