* 设计模式 – 责任链模式（Chain of Responsibility）：

在责任链模式里，由很多对象组成一条链，每个对象持有下个对象的引用。请求在这个

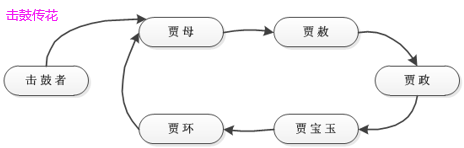
链上传递，直到链上的某一个对象决定处理此请求。

客户并不知道链上的哪一个对象最终处理这个请求，系统可以在不影响客户端的情况下

动态的重新组织链和分配责任。

**处理者有两个选择**：承担责任或者把责任推给下家。

一个请求可以最终不被任何接收端对象所接受。



比如：聚餐费用申请的请求：该请求会在 **项目经理—〉部门经理—〉总经理**这样一条领导处理链上进行传递审批处理。

责任链模式角色：

1）**抽象处理者(Handler)角色**：定义出一个处理请求的接口。如果需要，接口可以定义 出一个方法以设定和返回对下家的引用。这个角色通常由一个Java抽象类或者Java接口实现。上图中Handler类的聚合关系给出了具体子类对下家的引用，抽象方法handleRequest() 规范了子类处理请求的操作。

2）**具体处理者(ConcreteHandler)角色**：具体处理者接到请求后，可以选择将请求处理掉，或者将请求传给下家。由于具体处理者持有对下家的引用，因此，如果需要，具体处理者可以访问下家。

* 设计模式 – 责任链模式Demo：



