

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



# 私塾在线 《研磨设计模式》 ——跟着CC学设计系列精品教程

10101010101010101010101010101

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 本节课程概览

### n 学习职责链模式

#### 一：初识职责链模式

包括：定义、结构、参考实现

#### 二：体会职责链模式

包括：场景问题、不用模式的解决方案、使用模式的解决方案

#### 三：理解职责链模式

包括：认识职责链模式、处理多种请求、功能链、  
职责链模式的优缺点

#### 四：思考职责链模式

包括：职责链模式的本质、何时选用

**做最好的在线学习社区**

网 址：<http://sishuok.com>

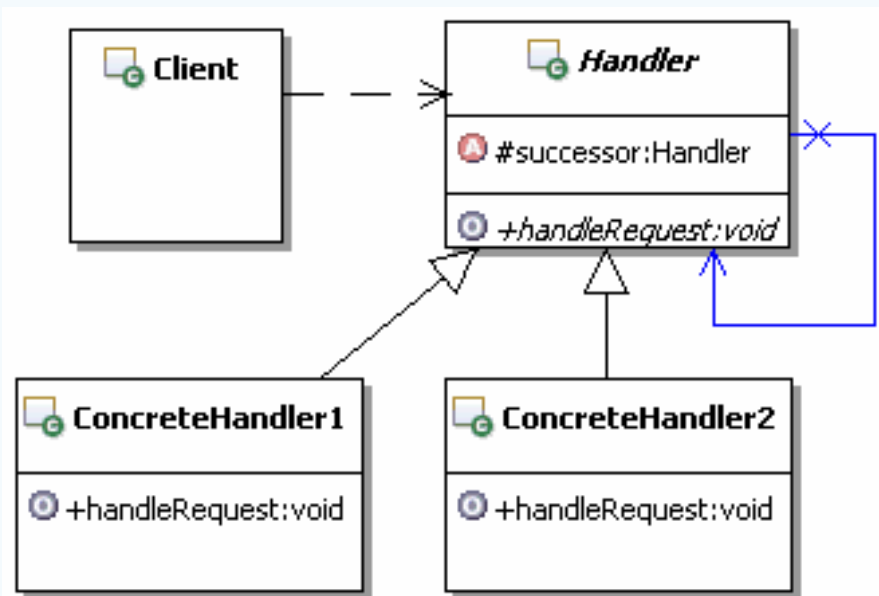
咨询QQ：2371651507

## 初识职责链模式

### n 定义

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。

### n 结构和说明



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 体会职责链模式

Handler:

定义职责的接口，通常在这里定义处理请求的方法，可以在这里实现后继链。

ConcreteHandler:

实现职责的类，在这个类里面，实现对在它职责范围内请求的处理，如果不处理，就继续转发请求给后继者。

Client:

职责链的客户端，向链上的具体处理者对象提交请求，让职责链负责处理。

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 体会职责链模式

### n 申请聚餐费用

来考虑这样一个功能：申请聚餐费用的管理。

申请聚餐费用的大致流程一般是：由申请人先填写申请单，然后交给领导审查，如果申请批准下来了，领导会通知申请人审批通过，然后申请人去财务核领费用，如果没有核准，领导会通知申请人审批未通过，此事也就此作罢了。

不同级别的领导，对于审批的额度是不一样的，比如：项目经理只能审批500元以内的申请；部门经理能审批1000元以内的申请；而总经理可以审核任意额度的申请。

也就是说，当某人提出聚餐费用申请的请求后，该请求会由项目经理、部门经理、总经理之中的某一位领导来进行相应的处理，但是提出申请的人并不知道最终会由谁来处理他的请求，一般申请人是把自己的申请提交给项目经理，或许最后是由总经理来处理他的请求，但是申请人并不知道应该由总经理来处理他申请请求。

那么该怎样实现这样的功能呢？

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

## 体会职责链模式

n 不用模式的解决方案

n 有何问题

上面的实现很简单，基本上没难度。仔细想想，这么实现有没有问题呢？  
仔细分析申请聚餐费用的业务功能和目前的实现，主要面临着如下问题：

- 1: 聚餐费用申请的处理流程是可能会变动的。
- 2: 各个处理环节的业务处理也是会变动的。

如果采用上面的实现，要是处理的逻辑发生了变化，解决的方法，一个是生成一个子类，覆盖requestToProjectManager方法，然后在里面实现新的处理；另一个方法就是修改处理申请方法的源码。

总之都不是什么好方法，也就是说，如果出现聚餐费用申请的处理流程变化的情况，或者是出现各个处理环节的功能变化的时候，上面的实现方式是很难灵活的变化来适应新功能的要求的。

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 体会职责链模式

把上面的问题抽象一下：客户端发出一个请求，会有很多对象都可以来处理这个请求，而且不同对象的处理逻辑是不一样的。对于客户端而言，无所谓谁来处理，反正有对象处理就可以了。

而且在上述处理中，还希望处理流程是可以灵活变动的，而处理请求的对象需要能方便的修改或者是被替换掉，以适应新的业务功能的需要。

请问如何才能实现上述要求？

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 体会职责链模式

### n 使用模式来解决的思路

仔细分析上面的场景，当客户端提出一个聚餐费用的申请，后续处理这个申请的对象，项目经理、部门经理和总经理，自然的形成了一个链，从项目经理à部门经理à总经理，客户端的申请请求就在这个链中传递，直到有领导处理为止。看起来，上面的功能要求很适合采用职责链来处理这个业务。

要让处理请求的流程可灵活的变动，一个基本的思路，那就是动态构建流程步骤，这样随时都可以重新组合出新的流程来。而要让处理请求的对象也要很灵活，那就要让它足够简单，最好是只实现单一的功能，或者是有限的功能，这样更有利于修改和复用。

职责链模式就很好的体现了上述的基本思路，首先职责链模式会定义一个所有处理请求的对象都要继承实现的抽象类，这样就有利于随时切换新的实现；其次每个处理请求对象只实现业务流程中的一步业务处理，这样使其变得简单；最后职责链模式会动态的来组合这些处理请求的对象，把它们按照流程动态组合起来，并要求它们依次调用，这样就动态的实现了流程。

这样一来，如果流程发生了变化，只要重新组合就好了；如果某个处理的业务功能发生了变化，一个方案是修改该处理对应的处理对象，另一个方案是直接提供一个新的实现，然后在组合流程的时候，用新的实现替换掉旧的实现就可以了。

**做最好的在线学习社区**

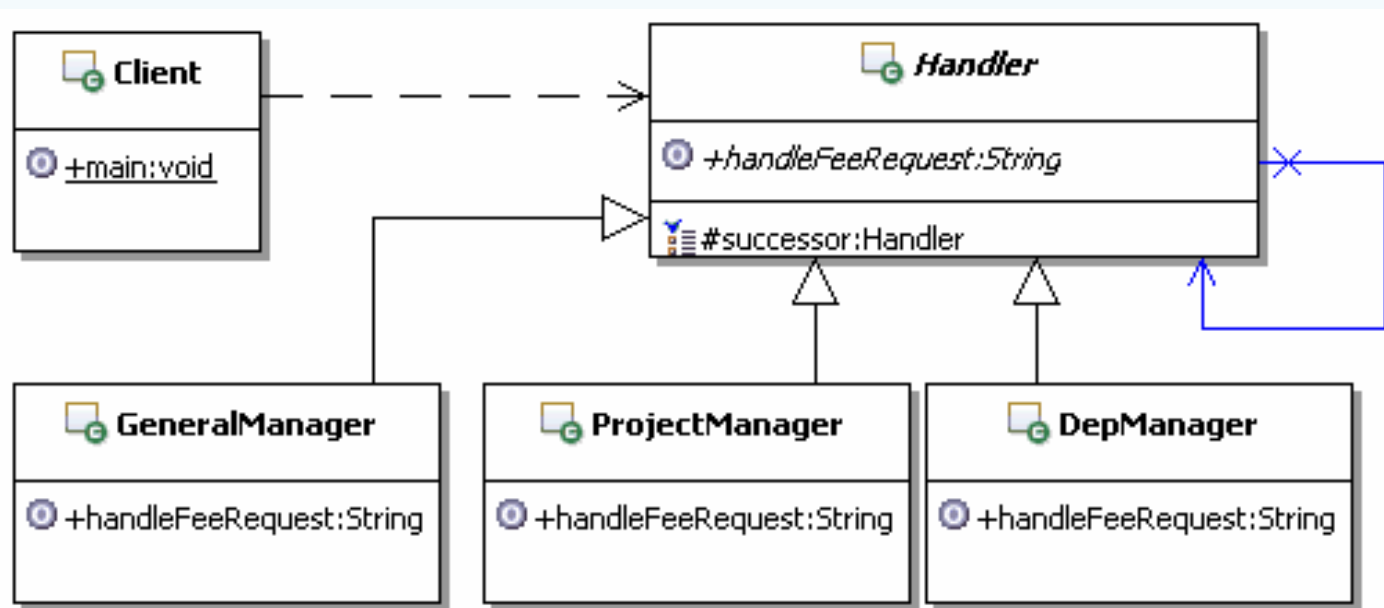
网 址：<http://sishuok.com>

咨询QQ：2371651507



## 体会职责链模式

n 使用模式来解决的类图



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解职责链模式

### n 认识职责链模式

#### 1: 职责链模式的功能

职责链模式主要用来处理：“客户端发出一个请求，有多个对象都有机会来处理这一个请求，但是客户端不知道究竟谁会来处理他的请求”，这样的情况。也就是需要让请求者和接收者解耦，这样就可以动态的切换和组合接收者了。

要注意在标准的职责链模式里面，是只要有对象处理了请求，这个请求就到此为止，不再被传递和处理了。

如果是要变形使用职责链，就可以让这个请求继续传递，每个职责对象对这个请求进行一定的功能处理，从而形成一个处理请求的功能链。

#### 2: 隐式接收者

当客户端发出请求的时候，客户端并不知道谁会真正处理他的请求，客户端只知道他提交请求的第一个对象。从第一个处理对象开始，整个职责链里面的对象，要么自己处理请求，要么继续转发给下一个接收者。

也就是对于请求者而言，并不知道最终的接收者是谁，但是一般情况下，总是会有一个对象来处理的，因此称为隐式接收者。

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解职责链模式

### 3: 如何构建链

职责链的链怎么构建呢？这是个大问题，实现的方式也是五花八门，归结起来大致有以下一些方式。

首先是按照实现的地方来说：

- (1)：可以实现在客户端，在提交请求前组合链，也就是在使用的时候动态组合链，称为外部链；
- (2)：可以在Handler里面实现链的组合，算是内部链的一种；
- (3)：可以在各个职责对象里面，由各个职责对象自行决定后续的处理对象，这种实现方式要求每个职责对象除了进行业务处理外，还必须了解整个业务流程。

按照构建链的数据来源，也就是决定了按照什么顺序来组合链的数据，又分为几种：

- (1)：一种就是在程序里面动态组合；
- (2)：可以通过外部，如数据库来获取组合的数据，这种属于数据库驱动的方式；
- (3)：还有一种方式就是通过配置文件传递进来，也可以是流程的配置文件。

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

## 理解职责链模式

如果是从外部获取数据来构建链，那么在程序运行的时候，会读取这些数据，然后根据数据的要求来获取相应的对象，并组合起来。

还有一种是不需要构建链，因为已有的对象已经自然构成链了，这种情况多出现在组合模式构建的对象树中，这样子对象可以很自然的向上找到自己的父对象。就像部门人员的组织结构一样，顶层是总经理，总经理下面是各个部门的经理，部门经理下面是项目经理，项目经理下面是各个普通员工，自然就可以形成：普通员工à项目经理à部门经理à总经理这样的链。

### 4：谁来处理

职责链中那么多处理对象，到底谁来处理请求呢，这个是在运行时期动态决定的。当请求被传递到某个处理对象的时候，这个对象会按照已经设定好的条件来判断，是否属于自己处理的范围，如果是就处理，如果不是就转发请求给下一个对象。

## 理解职责链模式

### 5：请求一定会被处理吗？

在职责链模式中，请求不一定会被处理，因为可能没有合适的处理者，请求在职责链里面从头传递到尾，每个处理对象都判断不属于自己处理，最后请求就没有对象来处理。这一点是需要注意的。

可以在职责链的末端始终加上一个不支持此功能处理的职责对象，这样如果传递到这里，就会出现提示，本职责链没有对象处理这个请求。



## 理解职责链模式

### n 处理多种请求

前面的示例都是同一个职责链处理一种请求的情况，现在有这样的需求，还是费用申请的功能，这次是申请预支差旅费，假设还是同一流程，也就是组合同一个职责链，从项目经理a传递给部门经理a传递给总经理，虽然流程相同，但是每个处理类需要处理两种请求，它们的具体业务逻辑是不一样的，那么该如何实现呢？

#### 1: 简单的处理方式

要解决这个问题，也不是很困难，一个简单的方法就是为每种业务单独定义一个方法，然后客户端根据不同的需要调用不同的方法，还是通过代码来示例一下。注意这里故意的把两个方法做的有些不一样，一个是返回String类型的值，一个是返回boolean类型的值；另外一个返回到客户端再输出信息，一个是直接在职责处理里面就输出信息。

## 理解职责链模式

### 2: 通用请求的处理方式

上面的实现看起来很容易，但是仔细想想，这样实现有没有什么问题呢？

有一个很明显的问题，那就是只要增加一个业务，就需要修改职责的接口，这是很不灵活的，Java开发中很强调面向接口编程，因此接口应该相对保持稳定，接口一改，需要修改的地方就太多了，频繁修改接口绝对不是个好主意。

那有没有什么好方法来实现呢？分析一下现在变化的东西：

- (1)：一是不同的业务需要传递的业务数据不同；
- (2)：二是不同的业务请求的方法不同；
- (3)：三是不同的职责对象处理这些不同的业务请求的业务逻辑不同

现在有一种简单的方式，可以较好的解决这些问题。首先定义一套通用的调用框架，用一个通用的请求对象来封装请求传递的参数；然后定义一个通用的调用方法，这个方法不去区分具体业务，所有的业务都是这这个方法，那么具体的业务如何区分呢，就是在通用的请求对象里面会有一个业务的标记；到了职责对象里面，愿意处理就跟原来一样的处理方式，如果不愿意处理，就传递到下一个处理对象就好了。

接下来看看如何在不改动现有的框架的前提下，扩展新的业务，这样才能说明这种设计的灵活性。

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

## 理解职责链模式

### n 功能链

在实际开发中，经常会出现一个把职责链稍稍变形的用法。在标准的职责链中，一个请求在职责链中传递，只要有一个对象处理了这个请求，就会停止。

现在稍稍变一下，改成一个请求在职责链中传递，每个职责对象负责处理请求的某一方面功能，处理完成后，不是停止，而是继续向下传递请求，当请求通过很多职责对象处理过后，功能也就处理完了，把这样的职责链称为功能链。

考虑这样一个功能，在实际应用开发中，在进行业务处理之前，通常需要进行权限检查、通用数据校验、数据逻辑校验等处理，然后才开始真正的业务逻辑实现。可以把这些功能分散到一个功能链中，这样做的目的是使程序结构更加灵活，而且复用性会更好，比如通用的权限检查就只需要做一份，然后就可以在多个功能链中使用了。

## 理解职责链模式

有些朋友看到这里，可能会想，这不是可以使用装饰模式来实现吗？没错，可以使用装饰模式来实现这样的功能，但是职责链会更灵活一些，因为装饰模式是在已有的功能上增加新的功能，多个装饰器之间会有一定的联系；而职责链模式的各个职责对象实现的功能，相互之间是没有关联的，是自己实现属于自己处理的那一份功能。

可能有些朋友会想到这很类似于在Web应用开发中的过滤器Filter，没错，过滤器链就类似于一个功能链，每个过滤器负责自己的处理，然后转交给下一个过滤器，直到把所有的过滤器都走完，然后进入到Servlet里面进行处理。最常见的过滤器功能，比如权限检查、字符集转换等，基本上都是Web应用的标配。

接下来在示例中，实现这样的功能：实现商品销售的业务处理，在真正进行销售的业务处理之前，需要对传入处理的数据，进行权限检查、通用数据检查和数据逻辑检查，只有这些检查都能通过的情况下，才说明传入的数据是正确的、有效的数据，才可以进行真正的业务功能处理。

## 理解职责链模式

- n 职责链模式的优缺点
  - 1: 请求者和接收者松散耦合
  - 2: 动态组合职责
  - 3: 产生很多细粒度对象
  - 4: 不一定能被处理



## 思考职责链模式

### n 职责链模式的本质

职责链模式的本质是：分离职责，动态组合

### n 何时选用职责链模式

- 1: 如果有多个对象可以处理同一个请求，但是具体由哪个对象来处理该请求，是运行时刻动态确定的。这种情况可以使用职责链模式，把处理请求的对象实现成为职责对象，然后把它们构成一个职责链，当请求在这个链中传递的时候，具体由哪个职责对象来处理，会在运行时动态判断
- 2: 如果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的话，可以使用职责链模式，职责链模式实现了请求者和接收者之间的解耦，请求者不需要知道究竟是哪一个接收者对象来处理了请求。
- 3: 如果想要动态指定处理一个请求的对象集合，可以使用职责链模式，职责链模式能动态的构建职责链，也就是动态的来决定到底哪些职责对象来参与到处理请求中来，相当于是动态指定了处理一个请求的职责对象集合

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送