



链滴

GOF 设计模式小白教程之中介者模式

作者: [valarchie](#)

原文链接: <https://ld246.com/article/1567347151264>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

中介者模式 (Mediator)

定义:

定义一个中介对象来封装一系列对象之间的交互，使原有对象之间的耦合松散，且可以独立地改变它之间的交互。中介者模式又叫调停模式，它是迪米特法则的典型应用。

通俗解释:

网游中玩家进行装备买卖的话，都需要满世界喊话。比如求购XX装备，或者卖XX装备要的私聊这样买家与卖家就变成了紧密耦合，所幸的是现在很多网游当中都有交易市场功能。这个交易市场就相当于一个中介者。买家和卖家不再直接联系，而是通过交易市场进行交易。在软件开发中有很多这种例子比如MVC框架中，控制器 (Controller) 就是模型 (Model) 和视图 (View) 的中介者。早期的JS开发中，Model和View就是直接耦合在一起，在维护上面变得十分困难。这里的Model和View就像家和卖家一样。有一句玩笑话就是在软件开发中没有增加一层抽象解决不了的问题。

代码:

中介类：市场类，用于沟通买家和卖家

```
public class Market {  
  
    private Queue<String> goods = new LinkedList<>();  
    // 通过市场卖出东西  
    public void sell(String good) {  
        goods.add(good);  
    }  
  
    // 通过市场买到东西  
    public String buy() {  
        return goods.poll();  
    }  
}
```

卖家类：通过市场卖出商品

```
public class Seller {  
  
    public void sellToMarket(Market market) {  
        market.sell("商品A");  
        System.out.println("向市场卖出一个商品A! ");  
    }  
}
```

买家类：通过市场买入商品

```
public class Buyer {
```

```
public void buyFromMarket(Market market) {  
    System.out.println("从市场买入" + market.buy() + "!");  
}  
  
}
```

测试中介者：买家和卖家只与市场进行交互，通过市场消除他们之间的直接耦合。中介者模式很多变只要消除两个对象的直接交互即可。

```
public class TestMediator {  
  
    public static void main(String[] args) {  
        // 市场类  
        Market market = new Market();  
        // 卖家  
        Seller seller = new Seller();  
        // 卖出两个商品  
        seller.sellToMarket(market);  
        seller.sellToMarket(market);  
        // 买家  
        Buyer buyer = new Buyer();  
        // 从市场买入两个商品  
        buyer.buyFromMarket(market);  
        buyer.buyFromMarket(market);  
  
    }  
  
}
```

运行结果：

```
向市场卖出一个商品A!  
向市场卖出一个商品A!  
从市场买入商品A!  
从市场买入商品A!
```

解析：

1. 降低了类的复杂度，将一对多转化成了一对一。
2. 各个类之间的解耦，符合迪米特原则。
3. 中介者会随着系统内的类增多而变得复杂难以维护。