**设计模式之Mediator(中介者)**

[**板桥里人**](http://www.jdon.com/aboutme.htm) http://www.jdon.com 2002/05/05

Mediator中介者模式定义:  
用一个中介对象来封装一系列关于对象交互行为.

为何使用Mediator?  
各个对象之间的交互操作非常多;每个对象的行为操作都依赖彼此对方,修改一个对象的行为,同时会涉及到修改很多其他对象的行为,如果使用Mediator模式,可以使各个对象间的耦合松散,只需关心和 Mediator的关系,使多对多的关系变成了一对多的关系,可以降低系统的复杂性,提高可修改扩展性.

如何使用?

首先 有一个接口,用来定义成员对象之间的交互联系方式:

|  |
| --- |
| public interface Mediator { } |

Meiator具体实现,真正实现交互操作的内容:

|  |
| --- |
| public class ConcreteMediator implements Mediator {  　　 //假设当前有两个成员. 　　 private ConcreteColleague1 colleague1 = new ConcreteColleague1();  　　 private ConcreteColleague2 colleague2 = new ConcreteColleague2();  　　 ...  } |

再看看另外一个参与者:成员,因为是交互行为,都需要双方提供一些共同接口,这种要求在Visitor Observer等模式中都是相同的.

|  |
| --- |
| public class Colleague { 　　 private Mediator mediator; 　　 public Mediator getMediator() {  　　 　　 return mediator; 　　 }  　　 public void setMediator( Mediator mediator ) {  　　 　　 this.mediator = mediator;  　　 } }  public class ConcreteColleague1 { }  public class ConcreteColleague2 { } |

每个成员都必须知道Mediator,并且和 Mediator联系,而不是和其他成员联系.

至此,Mediator模式框架完成,可以发现Mediator模式规定不是很多,大体框架也比较简单,但实际使用起来就非常灵活.

Mediator模式在事件驱动类应用中比较多,例如界面设计GUI.;聊天,消息传递等,在聊天应用中,需要有一个MessageMediator,专门负责request/reponse之间任务的调节.

MVC是J2EE的一个基本模式,View Controller是一种Mediator,它是Jsp和服务器上应用程序间的Mediator.