

```

package com.UMediator;

import java.util.*;
public class MediatorPattern
{
    public static void main(String[] args)
    {
        Mediator md=new ConcreteMediator();
        Colleague c1,c2;
        c1=new ConcreteColleague1();
        c2=new ConcreteColleague2();
        c1.setMedium(md);
        c2.setMedium(md);
        c1.send();
        System.out.println("-----");
        c2.send();
    }
}
//抽象中介者
abstract class Mediator
{
    public abstract void register(Colleague colleague);
    public abstract void relay(Colleague cl); //转发
}
//具体中介者
class ConcreteMediator extends Mediator
{
    private List<Colleague> colleagues=new ArrayList<Colleague>();
    public void register(Colleague colleague)
    {
        if(!colleagues.contains(colleague))
        {
            colleagues.add(colleague);
        }
    }
    public void relay(Colleague cl)
    {
        for(Colleague ob:colleagues)
        {
            if(!ob.equals(cl))
            {
                ((Colleague)ob).receive();
            }
        }
    }
}
}

```

```

//抽象同事类
abstract class Colleague
{
    protected Mediator mediator;
    public void setMedium(Mediator mediator)
    {
        this.mediator=mediator;
        mediator.register(this);
    }
    public abstract void receive();
    public abstract void send();
}

//具体同事类
class ConcreteColleague1 extends Colleague
{
    public void receive()
    {
        System.out.println("具体同事类1收到请求。");
    }
    public void send()
    {
        System.out.println("具体同事类1发出请求。");
        mediator.relay(this); //请中介者转发
    }
}

//具体同事类
class ConcreteColleague2 extends Colleague
{
    public void receive()
    {
        System.out.println("具体同事类2收到请求。");
    }
    public void send()
    {
        System.out.println("具体同事类2发出请求。");
        mediator.relay(this); //请中介者转发
    }
}

```