

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 学习中介者模式

一：初识中介者模式

包括：定义、结构、参考实现

二：体会中介者模式

包括：场景问题、使用模式的解决方案

三：理解中介者模式

包括：认识中介者模式、广义中介者、中介者模式的优缺点

四：思考中介者模式

包括：中介者模式的本质、何时选用

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

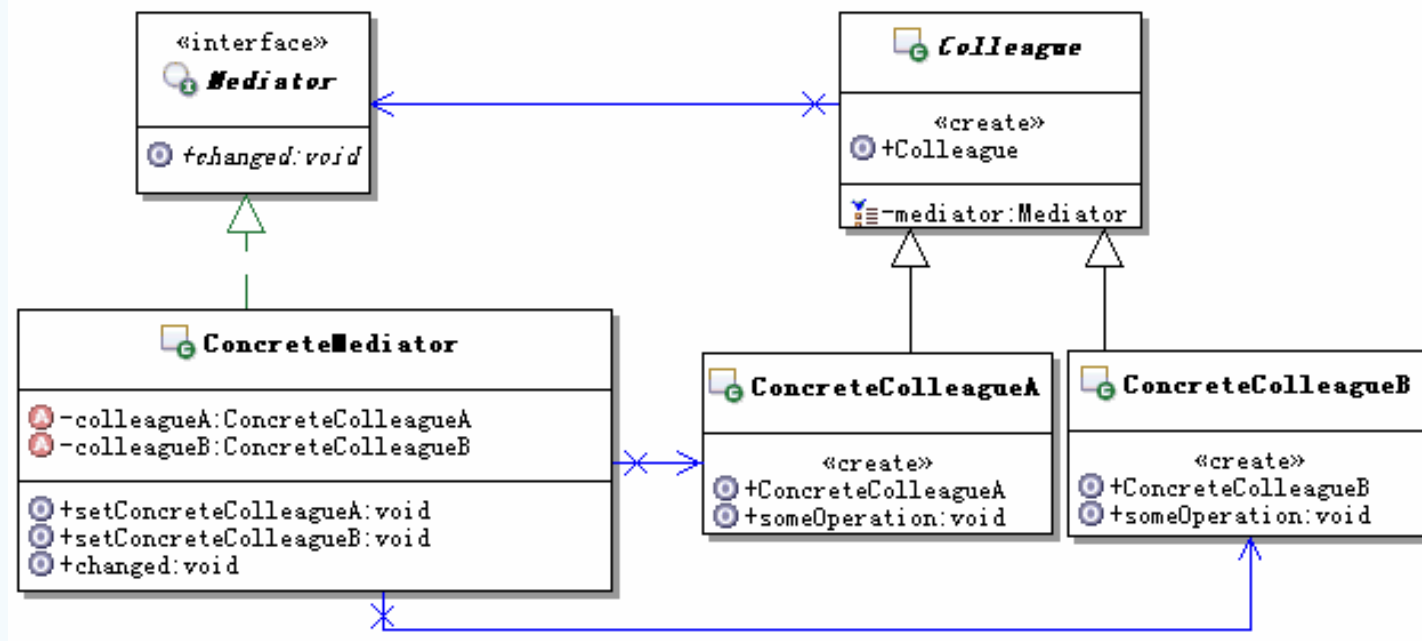
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

初识中介者模式

n 定义

用一个中介对象来封装一系列的对象交互。中介者使得各对象不需要显式地相互引用，从而使其耦合松散，而且可以独立的改变它们之间的交互。

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

初识中介者模式

Mediator:

中介者接口。在里面定义各个同事之间交互需要的方法，可以是公共的通讯方法，比如changed方法，大家都用，也可以是小范围的交互方法。

ConcreteMediator:

具体中介者实现对象。它需要了解并维护各个同事对象，并负责具体的协调各同事对象的交互关系。

Colleague:

同事类的定义，通常实现成为抽象类，主要负责约束同事对象的类型，并实现一些具体同事类之间的公共功能，比如：每个具体同事类都应该知道中介者对象，也就是具体同事类都会持有中介者对象，就可以定义到这个类里面。

ConcreteColleague:

具体的同事类，实现自己的业务，在需要与其它同事通讯的时候，就与持有的中介者通信，中介者会负责与其它同事交互。

做最好的在线学习社区

网 址：<http://sishuok.com>

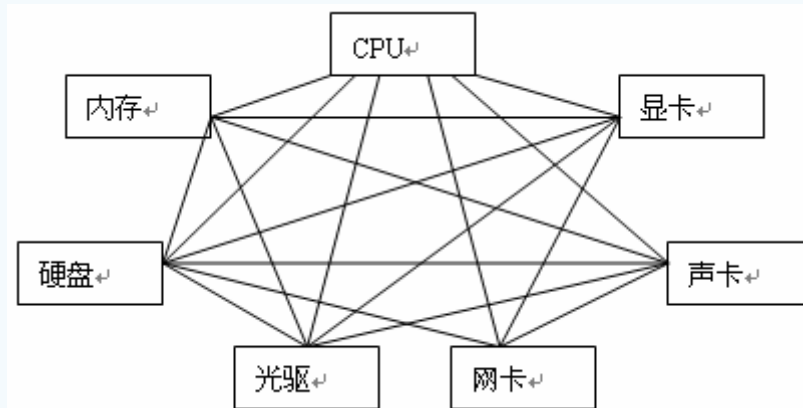
咨询QQ：2371651507

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

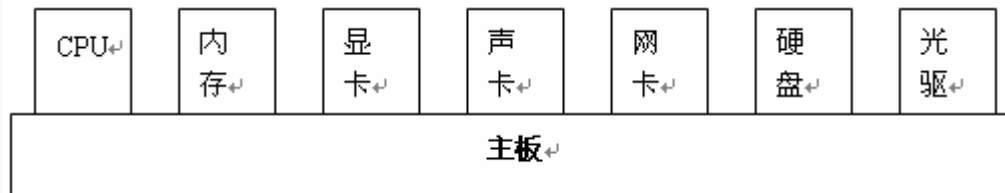
体会中介者模式

n 如果没有主板

如果电脑里面没有了主板，那么各个配件之间就必须自行相互交互，以互相传送数据，理论上说，基本上各个配件相互之间都存在交互数据的可能



有了主板，各个配件的交互完全通过主板来完成，每个配件都只需要和主板交互，而主板知道如何和所有的配件打交道，那就简单多了



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

体会中介者模式

n 存在的问题

如果上面的情况发生在软件开发上呢？

如果把每个电脑配件都抽象成为一个类或者是子系统，那就相当于出现了多个类之间相互交互，而且交互还很繁琐，导致每个类都必须知道所有需要交互的类，也就是我们常说的类和类耦合了，是不是很麻烦？

这样一来，不但开发的时候每个类会复杂，因为要兼顾其它的类，更要命的是每个类在发生改动的时候，需要通知所有相关的类一起修改，因为接口或者是功能发生了变动，使用它的地方都得变，太可怕了吧！

那该如何来简化这种多个对象之间的交互呢？

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

体会中介者模式

n 使用电脑来看电影

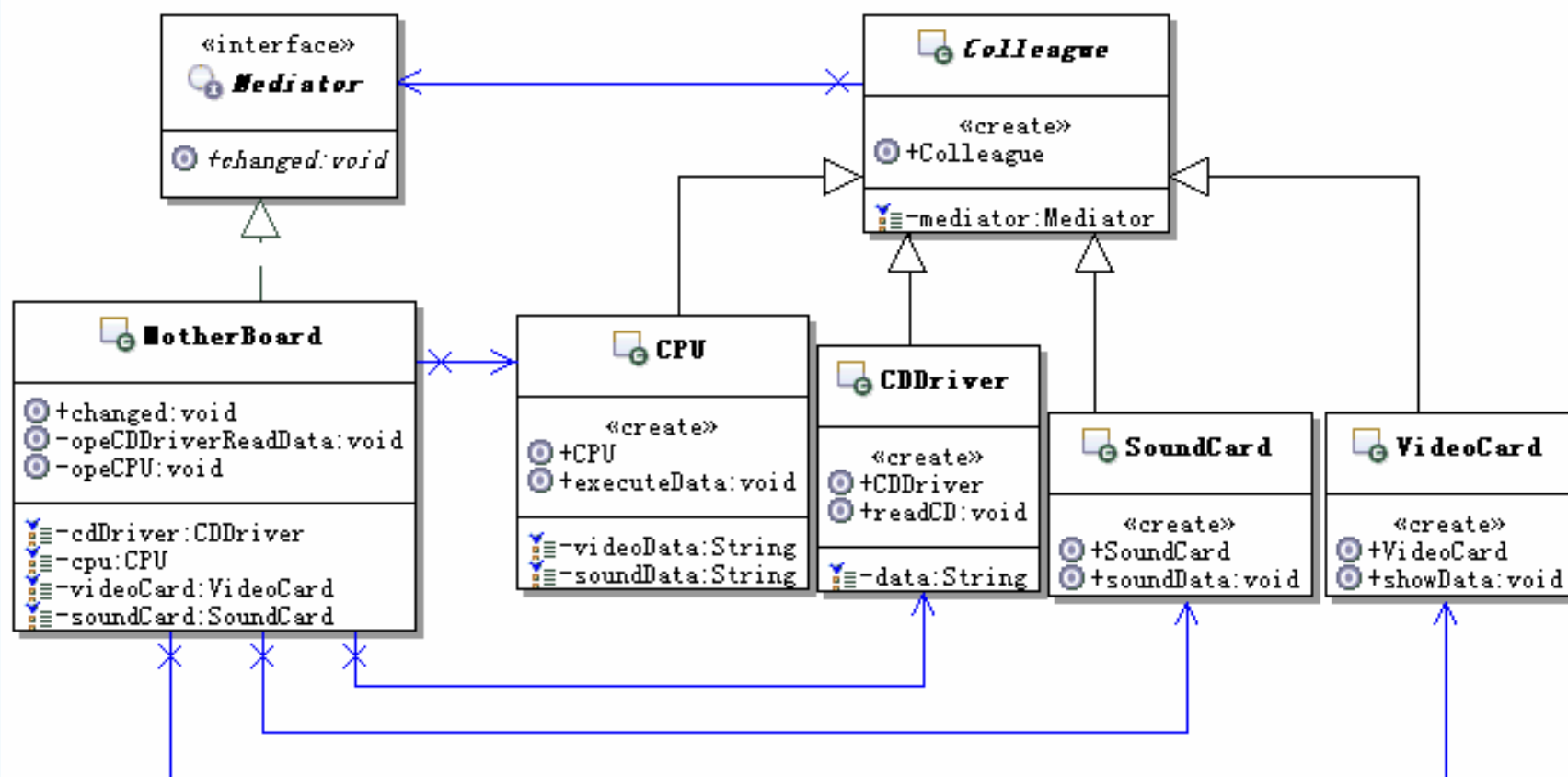
为了演示，考虑一个稍微具体点的功能。在日常生活中，我们经常使用电脑来看电影，把这个过程描述出来，这里仅仅考虑正常的情况，也就是有主板的情况，简化后假定会有如下的交互过程：

- 1: 首先是光驱要读取光盘上的数据，然后告诉主板，它的状态改变了
- 2: 主板去得到光驱的数据，把这些数据交给CPU进行分析处理
- 3: CPU处理完后，把数据分成了视频数据和音频数据，通知主板，它处理完了
- 4: 主板去得到CPU处理过后的数据，分别把数据交给显卡和声卡，去显示出视频和发出声音

当然这是一个持续的、不断重复的过程，从而形成不间断的视频和声音，具体的运行过程不在讨论之列，假设就有如上简单的交互关系就可以了。也就是说想看电影，把光盘放入光驱，光驱开始读盘，就可以看电影了

体会中介者模式

n 使用模式的解决方案



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

理解中介者模式

n 认识中介者模式

1: 中介者模式的功能

中介者的功能非常简单，就是封装对象之间的交互。如果一个对象的操作会引起其它相关对象的变化，或者是某个操作需要引起其它对象的后续或连带操作，而这个对象又不希望自己来处理这些关系，那么就可以找中介者，把所有的麻烦扔给它，只在需要的时候通知中介者，其它的就让中介者去处理就可以了。

反过来，其它的对象在操作的时候，可能会引起这个对象的变化，也可以这么做。最后对象之间就完全分离了，谁都不直接跟其它对象交互，那么相互的关系，全部被集中到中介者对象里面了，所有的对象就只是跟中介者对象进行通信，相互之间不再有联系。

把所有对象之间的交互都封装在中介者当中，无形中还得另外另一个好处，就是能够集中的控制这些对象的交互关系，这样有什么变化的时候，修改起来就很方便。

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

理解中介者模式

2: 需要Mediator接口吗

有没有使用Mediator接口的必要，取决于是否会提供多个不同的中介者实现。如果中介者实现只有一个的话，而且预计中也没有需要扩展的要求，那么就可以不定义Mediator接口，让各个同事对象直接使用中介者实现对象；如果中介者实现不只一个，或者预计中有扩展的要求，那么就需要定义Mediator接口，让各个同事对象来面向中介者接口编程，而无需关心具体的中介者实现。

3: 同事关系

在中介者模式中，要求这些类都要继承相同的类，也就是说，这些对象从某个角度讲是同一个类型，算是兄弟对象。

正是这些兄弟对象之间的交互关系很复杂，才产生了把这些交互关系分离出去，单独做成中介者对象，这样一来，这些兄弟对象就成了中介者对象眼里的同事。

理解中介者模式

4：同事和中介者的关系

中介者对象和同事对象之间是相互依赖的。

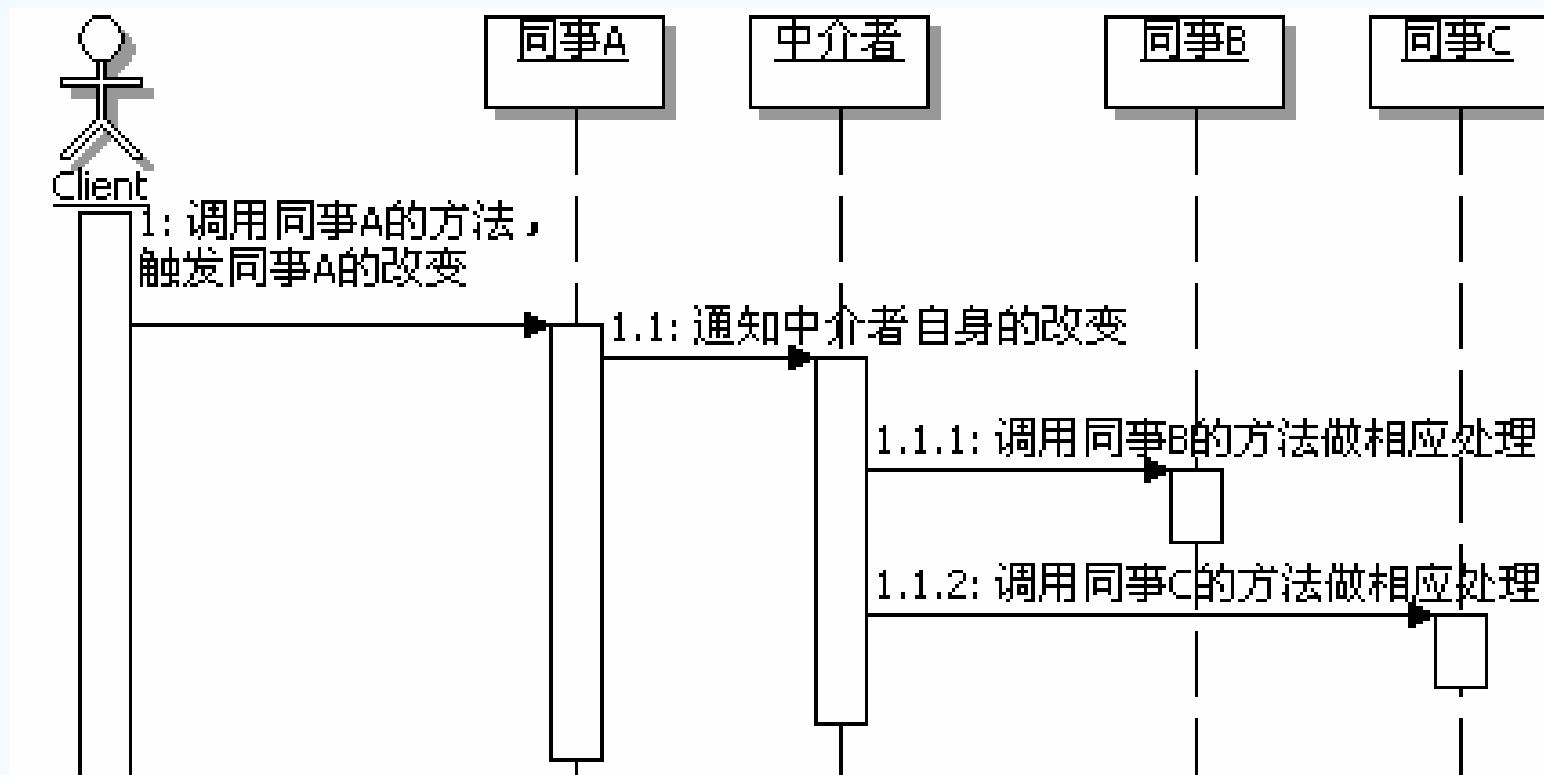
5：如何实现同事和中介者的通信

一种实现方式是在Mediator接口中定义一个特殊的通知接口，作为一个通用的方法，让各个同事类来调用这个方法。

另外一种实现方式是可以采用观察者模式，把Mediator实现成为观察者，而各个同事类实现成为Subject，这样同事类发生了改变，会通知Mediator。Mediator在接到通知过后，会与相应的同事对象进行交互。

理解中介者模式

6: 中介者模式的调用顺序示意图



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

理解中介者模式

n 广义中介者

n 标准的中介者模式在实际使用中的困难

1: 是否有必要为同事对象定义一个公共的父类？

大家都知道，Java是单继承的，为了使用中介者模式，就让这些同事对象继承一个父类，这是很不好的；再说了，这个父类目前也没有什么特别的公共功能，也就是说继承它也得不到多少好处。

在实际开发中，很多相互交互的对象本身是没有公共父类的，强行加上一个父类，会让这些对象实现起来特别别扭。

2: 同事类有必要持有中介者对象吗？

同事类需要知道中介者对象，以便当它们发生改变的时候，能够通知中介者对象，但是否需要作为属性，并通过构造方法传入，这么强的依赖关系呢？

也可以有简单的方式去通知中介对象，比如把中介对象做成单例，直接在同事类的方法里面去调用中介者对象。

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

理解中介者模式

3: 是否需要中介者接口？

在实际开发中，很常见的情况是不需要中介者接口的，而且中介者对象也不需要创建很多个实例，因为中介者是用来封装和处理同事对象的关系的，它一般是没有状态需要维护的，因此中介者通常可以实现成单例。

4: 中介者对象是否需要持有所有的同事？

虽说中介者对象需要知道所有的同事类，这样中介者才能与它们交互。但是是否需要做为属性这么强烈的依赖关系，而且中介者对象在不同的关系维护上，可能会需要不同的同事对象的实例，因此可以在中介者处理的方法里面去创建、或者获取、或者从参数传入需要的同事对象。

5: 中介者对象只是提供一个公共的方法，来接受同事对象的通知吗？

从示例就可以看出来，在公共方法里，还是要去区分到底是谁调过来，这还是简单的，还没有去区分到底是什么样的业务触发调用过来的，因为不同的业务，引起的与其它对象的交互是不一样的。

因此在实际开发中，通常会提供具体的业务通知方法，这样就不用再去判断到底是什么对象，具体是什么业务了。

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

理解中介者模式

n 对标准的中介者模式在实际使用中的改进

基于上面的考虑，在实际应用开发中，经常会简化中介者模式，来使开发变得简单，比如有如下的简化：

- 1: 通常会去掉同事对象的父类，这样可以任意的对象，只要需要相互交互，就可以成为同事；
- 2: 还有通常不定义Mediator接口，把具体的中介者对象实现成为单例；
- 3: 另外一点就是同事对象不再持有中介者，而是在需要的时候直接获取中介者对象并调用；中介者也不再持有同事对象，而是在具体处理方法里面去创建、或者获取、或者从参数传入需要的同事对象。

把这样经过简化、变形使用的情况称为广义中介者。

理解中介者模式

n 广义中介者示例——部门与人员

1: 部门和人员的关系：是多对多的

2: 问题的出现

想想部门和人员的功能交互，举几个常见的功能：

(1) 部门被撤销

(2) 部门之间进行合并

(3) 人员离职

(4) 人员从一个部门调职到另外一个部门

想想要实现这些功能，按照前面的设计，该怎么做呢？

(1) 系统运行期间，部门被撤销了，就意味着这个部门不存在了，可是原来这个部门下所有的人员，每个人员的所属部门中都有这个部门呢，那么就需要先通知所有的人员，把这个部门从它们的所属部门中去掉，然后才可以清除这个部门。

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

理解中介者模式

- (2) 部门合并，是合并成一个新的部门呢，还是把一个部门并入到另一个部门？如果是合并成一个新的部门，那么需要把原有的两个部门撤销，然后再新增一个部门；如果是把一个部门合并到另一个部门里面，那就是撤销掉一个部门，然后把这个部门下的人员移动到这个部门。不管是那种情况，都面临着需要通知相应的人员进行更改这样的问题。
- (3) 人员离职了，反过来就需要通知他所属于的部门，从部门的拥有人员的记录中去除掉这个人员。
- (4) 人员调职，同样需要通知相关的部门，先从原来的部门中去除掉，然后再到新的部门中添加上。

看了上述的描述，感觉如何？

麻烦的根源在什么地方呢？仔细想想，对了，麻烦的根源就在于部门和人员之间的耦合，这样导致操作人员的时候，需要操作所有相关的部门，而操作部门的时候又需要操作所有相关的人员，使得部门和人员搅和在了一起。

做最好的在线学习社区

网 址：<http://sishuok.com>

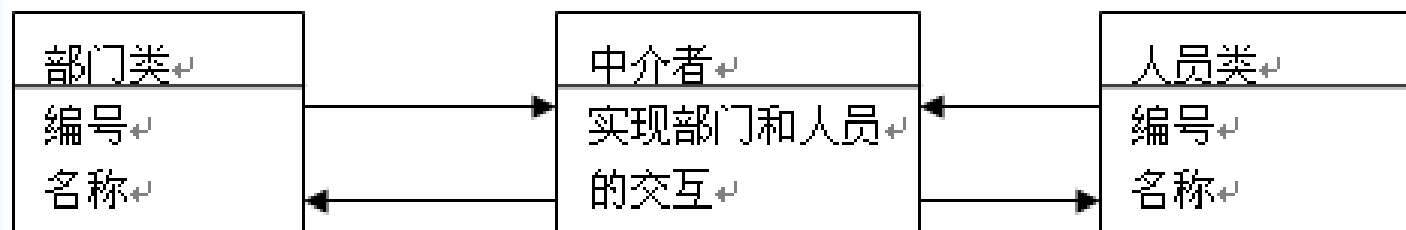
咨询QQ：2371651507

理解中介者模式

3: 中介者来解决

找到了根源就好办了，采用中介者模式，引入一个中介者对象来管理部门和人员之间的关系，就能解决这些问题了。

如果采用标准的中介者模式，想想上面提出的那些问题点吧，就知道实现起来会很别扭。因此采用广义的中介者来解决，这样部门和人员就完全解耦了，也就是说部门不知道人员，人员也不知道部门，它们完全分开，它们之间的关系就完全由中介者对象来管理了。



理解中介者模式

- n 中介者模式的优缺点
 - 1: 松散耦合
 - 2: 集中控制交互
 - 3: 多对多变成一对多
 - 4: 过度集中化

思考中介者模式

n 中介者模式的本质

中介者模式的本质是：封装交互

n 何时选用中介者模式

- 1: 如果一组对象之间的通信方式比较复杂，导致相互依赖、结构混乱，可以采用中介者模式，把这些对象相互的交互管理起来，各个对象都只需要和中介者交互，从而使得各个对象松散耦合，结构也更清晰易懂
- 2: 如果一个对象引用很多的对象，并直接跟这些对象交互，导致难以复用该对象。可以采用中介者模式，把这个对象跟其它对象的交互封装到中介者对象里面，这个对象就只需要和中介者对象交互就可以了

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