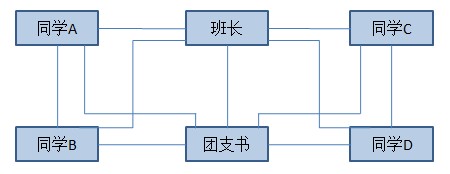
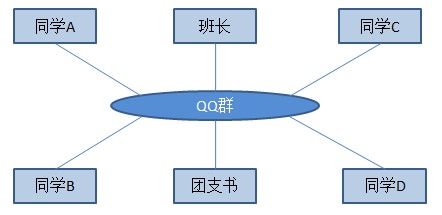
**一、初识中介者模式**

那些年，我们一起上过的大学，班级里有班长，有团书记。想一想如果没有QQ这种通讯工具的话，那么班长或者团支书该怎样下达消息呢？？同时，班级上两个同学之间也可惜沟通啊，沟通一下，院里哪个女生，哪个帅哥呀~~~如果没有QQ的话，大概就是下面的情景：



哎呀呀，看看这个乱那。如果同学的数目多起来就会变成网状的结构啦。原本把一个系 统分割成一些对象是可以增强复用性的，但是现在的情况是，这些兑现之间存在着大量的联系，耦合性极高。这是很不利于复用的，同时这种情况使得系统的灵活性 大大的降低，使得对系统的扩展很难，要是新转来一个学生的话，要改动的地方就多了去了。

如果现在可以使用QQ，那么可以采用另一种方式设计这个系统呢，比如做成星形的结构：

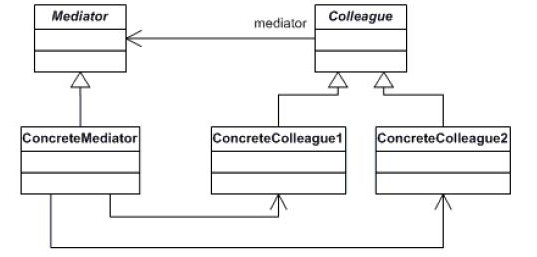


看看这种“星形结构”和“网状结构”的区别吧，显然采用星形结构就可以避免上面的网状结构存在的问题了，实际上这里的QQ就是指的是中介，这样一来每个学生对象就不用存在耦合了，同学之间需要交流可以通过一个QQ群。

本人认为，由原来的网状结构到星形结构的转变是理解中介者模式比较好的途径，下面来具体看看中介者模式，GoF说：

**中介者模式（Mediator Pattern）：**定义一个中介对象来封装系列对象之间的交互。中介者使各个对象不需要显示地相互引用，从而使其耦合性松散，而且可以独立地改变他们之间的交互。

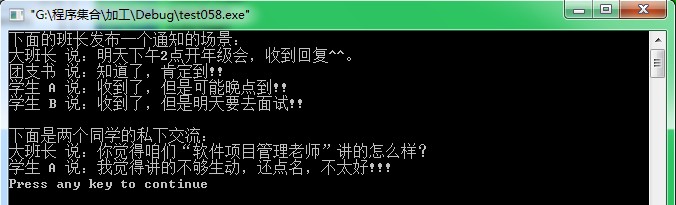
看看结构图的形象描述吧：



对于设计模式这个东西，理解了，应该写一个代码体会一下：看看这些类，比如：Mediator，ConcreteMediator，Colleague和ConcreteColleage1到底是怎么组织的。

http://images.cnblogs.com/OutliningIndicators/ContractedBlock.gifView Code

下面是运行的结果：



看看，这样就利用qq这个平台完成了同学之间的交流。并且同学之间并没有相互联系，联系都是通过qq实现的，如果现在要对象进行扩展，比如也可以通过飞信啊，只需要加入**class FeiXin:public Mediator**就可以了，如果想扩展同学C只需要，**class StudentC:public Colleage**即可。并且在客户端的使用也是很方便的哦~~这时回过头想一想如果不用“中介者模式”的话，系统会是什么样呢？？？

**二、使用中介者模式的场合和优缺点**

**使用终结者模式的场合**

1.一组定义良好的对象，现在要进行复杂的通信。

2.定制一个分布在多个类中的行为，而又不想生成太多的子类。

可以看出，中介对象主要是用来封装行为的，行为的参与者就是那些对象，但是通过中介者，这些对象不用相互知道。呵呵~~~

**使用中介者模式的优点：**

1.降低了系统对象之间的耦合性，使得对象易于独立的被复用。

2.提高系统的灵活性，使得系统易于扩展和维护。

**使用中介者模式的缺点：**

中介者模式的缺点是显而易见的，因为这个“中介“承担了较多的责任，所以一旦这个中介对象出现了问题，那么整个系统就会受到重大的影响