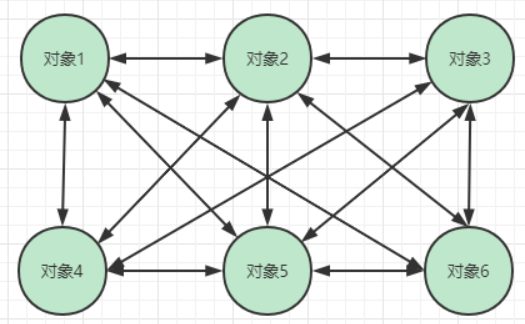
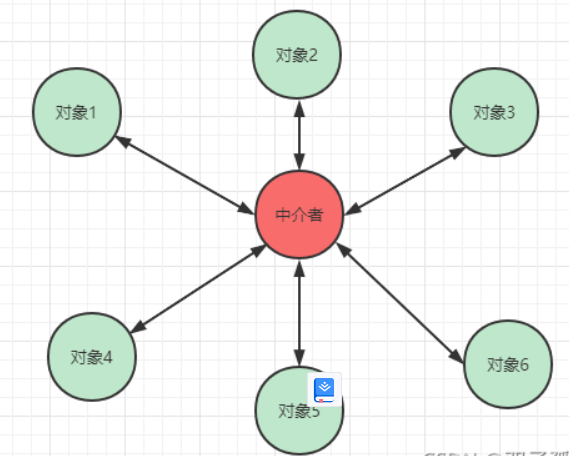
1. 为什么需要中介者模式

假如现在有六个对象，对象与对象之间都需要互相调用，那么它们的关系就会如下图所示：



各个对象之间互相依赖，整个依赖关系形成了网状，关系错综复杂，非常难以管理。这时候如果引入一个中介者，每个对象都只和中介者交互，那么它们之间的关系就会如下图所示：



可以看到，所有的对象之间的关系都归中介者进行统一管理，避免了各个对象之间各自依赖的乱像。

中介者模式（Mediator Pattern）是用来降低多个对象和类之间的通信复杂性。这种模式提供了一个中介类，  
该类通常处理不同类之间的通信，并支持松耦合，使代码易于维护。中介者模式属于\*\*\*行为型模式。  
  
介绍  
  
意图：用一个中介对象来封装一系列的对象交互，中介者使各对象不需要显式地相互引用，从而使其耦合松散，而且可以独立地改变它们之间的交互。  
主要解决：对象与对象之间存在大量的关联关系，这样势必会导致系统的结构变得很复杂，同时若一个对象发生改变，我们也需要跟踪与之相关联的对象，同时做出相应的处理。  
  
何时使用：多个类相互耦合，形成了网状结构。  
  
如何解决：将上述网状结构分离为星型结构。  
  
关键代码：对象 Colleague 之间的通信封装到一个类中单独处理。  
  
应用实例：  
 1、机场调度系统。  
 2、MVC 框架，其中C（控制器）就是 M（模型）和 V（视图）的中介者  
 3、Java Timer 中的中介者模式  
  
优点：  
 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。  
 3、符合迪米特原则。  
  
缺点：中介者会庞大，变得复杂难以维护。  
  
使用场景：  
 1、系统中对象之间存在比较复杂的引用关系，导致它们之间的依赖关系结构混乱而且难以复用该对象。  
 2、想通过一个中间类来封装多个类中的行为，而又不想生成太多的子类。  
  
注意事项：不应当在职责混乱的时候使用。