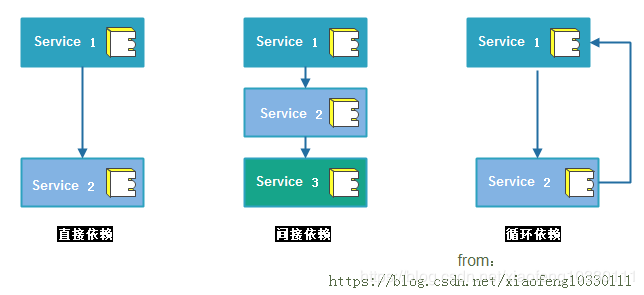
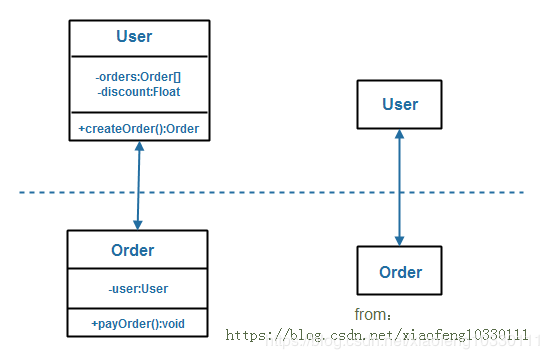
## 管理服务的依赖关系，构建无环依赖关系

依赖关系主要要三种：直接依赖，间接依赖，循环依赖，我们要根据无环依赖原则，知道系统中不应该存在循环依赖，三种依赖的关系图如下：



我们要消除循环依赖，基本思路是通过在两个相互依赖的中间添加中间层，变循环依赖为间接依赖，有三种方式可以做到这一点，分别是上移，下移，回调

以具体例子来讲解：如下图是一个循环依赖，User对象可以存储Order对象并保持Order对象列表，而Order对象需要用到User对象来进行打折discount运算计算折扣后的费率

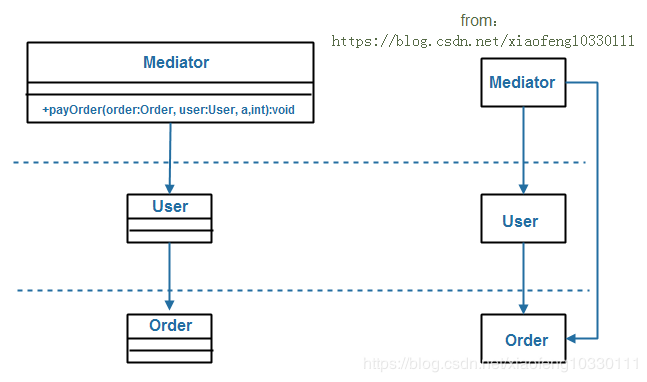


解决方案

#### 1上移切入点：交互部分抽离

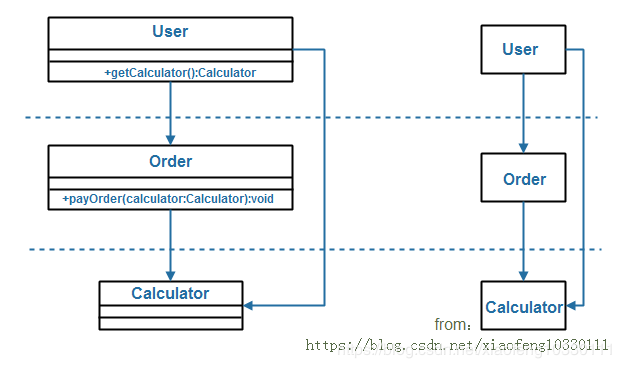
思想试将两个相互依赖的组件中的交互部分抽象出来形成一个新的组件，而新的组件包含原来两个组件的引用，即将循环关系剥离出来并上升到一个更高层次的组件中

对User对象和Order对象循环依赖进行上移，引入一个新的组件Mediator，并提供一个PayOrder()方法对循环依赖进行剥离，方法中使用User和Order作为参数，实现Order中对User信息进行打折金额的计算业务逻辑



#### 下移切入点，依赖关系转移重构

将依赖关系进行重构，重构抽象一个Calculator组件专门包含打折信息和金额计算，该Calculator由User创建，并注入到Order中的pay方法中



#### 回调切入点：接口或抽象类

回调本质上就是一种双向调用模式，在面向对象的语音中，回调通常通过接口或者抽象类来实现。

我们抽象出一个Calculator接口用于封装金额计算逻辑，该接口与Order处于同一个层级，User实现该接口，这样一来，原关系转为间接依赖了。通过依赖注入机制，很容易实现User和Order的有效交互

