

Facade模式

Facade模式

一 意图

为子系统的一组接口提供一个一致的界面，**Facade**模式定义了一个高层接口，这个接口使得这一子系统更加容易使用。

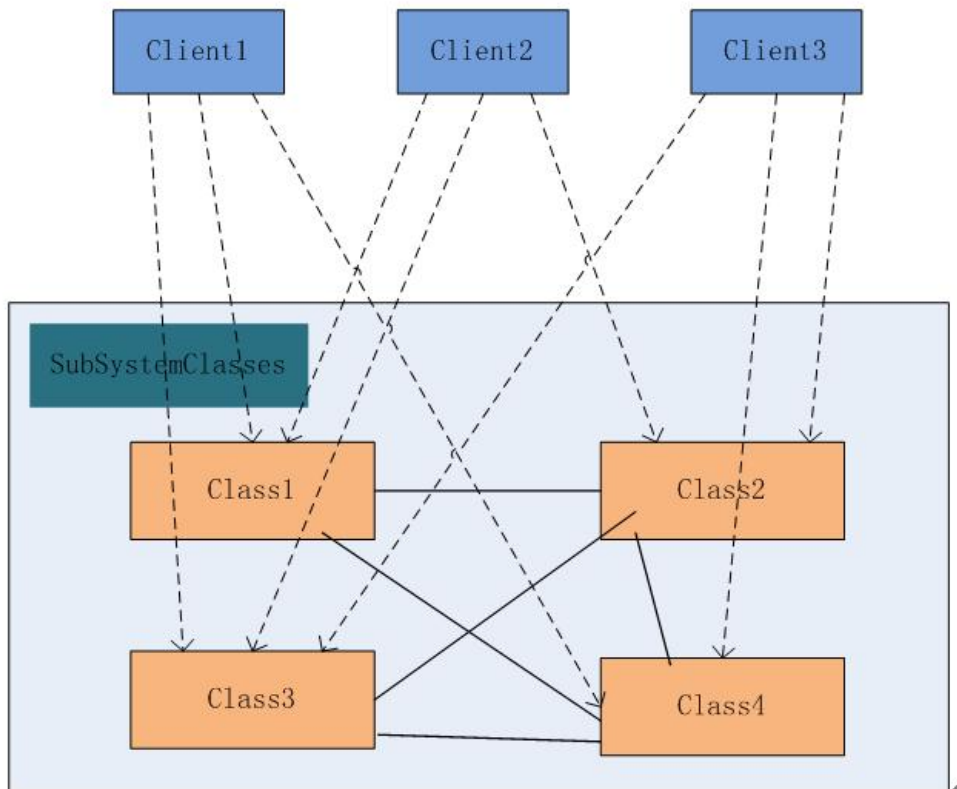
二 动机

将一个系统划分成为若干个子系统有利于降低系统的复杂性。一个常见的设计目标是使子系统间的通信和相互依赖关系达到最小。

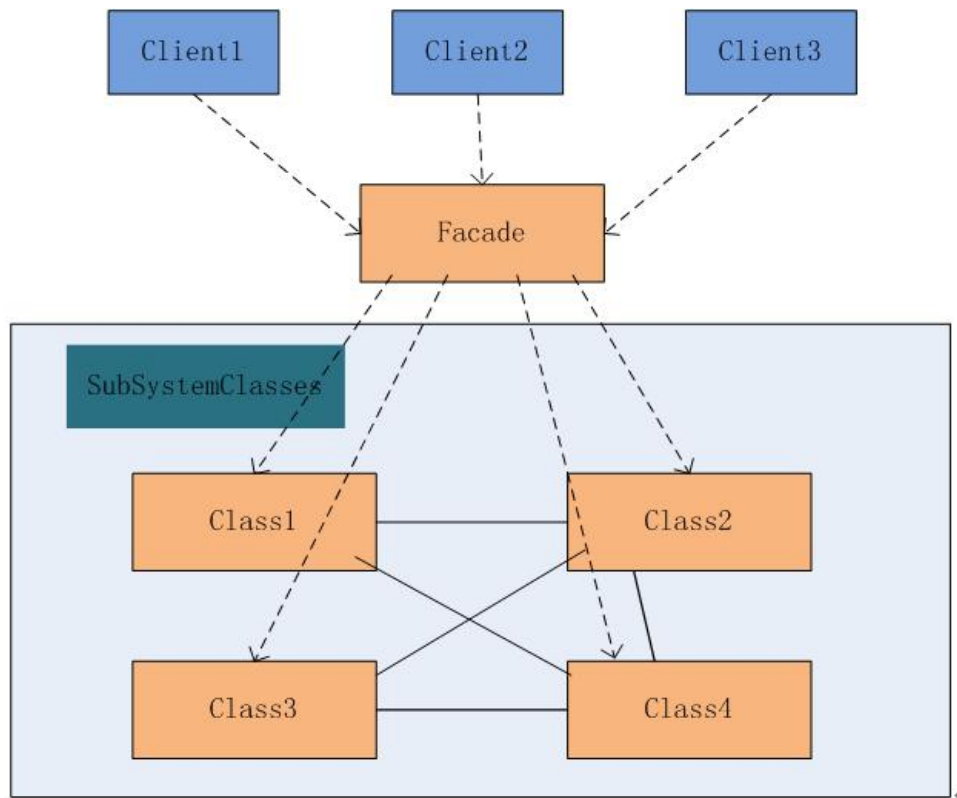
达到该目标的途径之一就是引入一个外观（**Facade**）对象，它为子系统中较一般的设施提供了一个单一而简单的界面。

将各个子系统整合起来作为**Facade**，提供给客户端使用。

看下面这样一个系统：



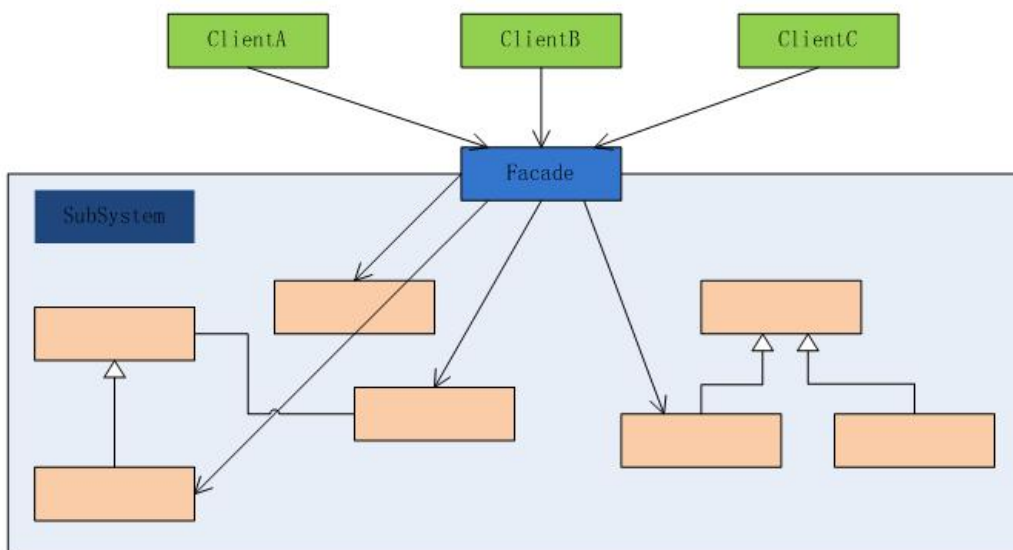
转变成：



三 适用性及其结构

1. 当你要为一个复杂子系统提供一个简单接口时。
2. 客户程序与抽象类的实现部分之间存在着很大的依赖性。
3. 当你需要构建一个层次结构的子系统时，使用**Facade**模式定义子系统中每层的入口点。仅通过**facade**进行通讯。

结构：



Facade:

- 知道哪些子系统类负责处理请求。
- 将客户的请求代理给适当的子系统对象。

Subsystem classes :

- 实现子系统的功能。
- 处理由Facade对象指派的任务。
- 没有**facade**的任何相关信息；即没有指向**facade**的指针。

客户程序通过发送请求给**Facade**的方式与子系统通讯，**Facade**将这些消息转发给适当的子系统对象。

尽管是子系统中的有关对象在做实际工作，但**Facade**模式本身也必须将它的接口转换成子系统的接口。

Facade模式有助于建立层次结构系统，也有助于对对象之间的依赖关系分层。

Facade模式可以消除复杂的循环依赖关系。降低客户-子系统之间的耦合度。

使用**Facade**的客户程序不需要直接访问子系统对象。