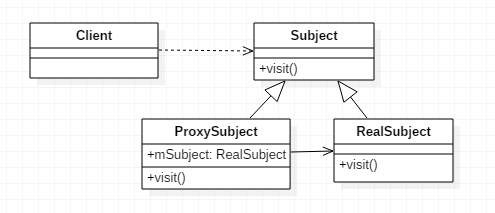
## 代理模式

1. 定义与使用场景

定义：为其他对象提供一种代理以控制对这个对象的访问。

使用场景：当无法或者不想直接访问某个对象或访问某个对象存在困难时可以通过一个代理对象来间接访问，为了保证客户端使用的透明性，委托对象或代理对象需要实现相同的接口。

1. UML类图



Subject：抽象主题类，该类主要职责是声明真实主题与代理的共同接口的方法，该类可以是一个抽象类或者一个接口

RealSubject：真实主题类

ProxySubject：代理类

Client：客户类

1. 简单实现

代理分为两类，静态代理、动态代理

静态代理：代理类的代码由程序员自己或者通过一些自动化工具生成固定的代码再对其进行编译，也就是说在我们的代码运行前代理类的class编译文件就已经存在

动态代理：通过反射机制动态地生成代理者的对象，代理类代理谁将会在运行时决定。通过实现InvocationHandler接口实现动态代理。

1. 源码实现
2. 实战总结