

代理模式详解

题目难度：★★★★

知识点标签：设计模式之代理模式

学习时长：20分钟

题目描述

聊一下代理模式，在哪里遇到或者使用过代理模式

解题思路

面试官问题可以从几个方面来回答：代理模式概念，代理模式的三个角色，静态代理，JDK动态代理

代理模式定义

现实生活中经常听到一个词“代理”，比如某某酒品牌什么什么省总代理；还有像现在的明星都有自己的经纪人，有事需要找他们的时候就要先找他们的经纪人，经纪人也相当于是一个代理；再比如打官司都需要找一个律师，有什么问题直接由律师去沟通解决，那律师就是自己的一个代理。

生活中的代理随处可见，Java中也存在一种设计模式，就叫代理模式，并且使用非常普遍，很多著名框架和开源项目的源码中都有大量使用代理模式，比如比较流行的Spring、MyBatis等，都有用到代理模式。特别是MyBatis，代理模式随处可见。那到底什么是代理模式呢？

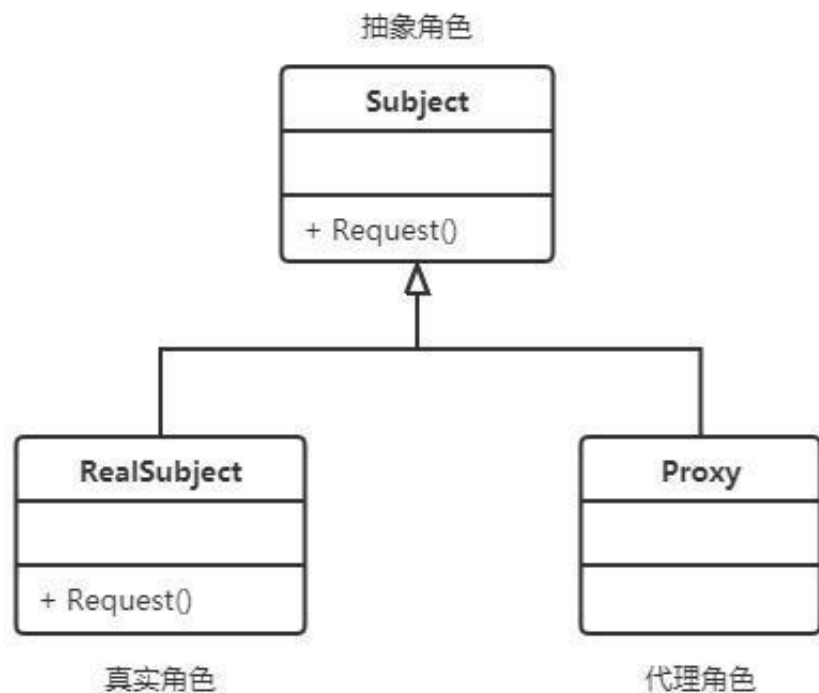
代理模式也叫做委托模式，是指为其他对象提供一种代理以控制对这个对象的访问。代理模式可以分为静态代理和动态代理，动态代理又分为JDK提供的动态代理和CGLIB动态代理

代理模式中的三个角色

代理模式中存在三个角色：

1. 抽象角色：通过接口或抽象类声明真实角色实现的业务方法
2. 代理角色：实现抽象角色，是真实角色的代理，通过真实角色的业务逻辑方法来实现抽象方法，并可以附加自己的操作。
3. 真实角色：实现抽象角色，定义真实角色所要实现的业务逻辑，供代理角色调用。

这三个角色之间的关系可以用一张简单的类图来表示：



静态代理

静态代理：代理角色和真实角色继承同一个父类或实现相同的接口，使用时只需要把真实角色传递给代理角色，代理角色调用真实角色的方法。

例如游戏代练的例子：我们找游戏代练一般会找专业的代练公司，代练公司会有很多代练人员，我们相当于客户端，代练公司相当于代理角色，代练人员就是真实执行任务的真实角色。

静态代理的优缺点：

优点：

1. 真实角色类（业务类）只需要关注业务逻辑本身，这是代理模式的共有优点。
2. 代理类可以在调用业务类的处理方法之前和之后做一些增强性的操作，比如记录日志，管理事务等，这也是代理模式的共有优点。

缺点：

1. 代理类和业务类实现了相同的接口，并且实现了相同的方法，代码冗余。如果接口增加一个方法，所有的代理类和所有的实现类都需要增加这个方法的实现，不易维护。
2. 代理对象只能代理同一种类型的对象，如果要对多种类型的对象进行代理，就要写多个代理类，这就大大增加类文件的数量，不适合在大规模程序中使用

JDK动态代理

动态代理的代理类是在程序运行的时候，有JAVA反射机制动态生成的。JAVA的java.lang.reflect包下提供了一个Proxy类和一个InvocationHandler接口，JDK动态代理的实现主要依靠这两个对象实现。

1. 创建一个类，实现InvocationHandler接口，并重写invoke方法。这个类中持有一个被代理对象的实例target，还有一个invoke方法，这个方法中通过反射机制调用被代理对象target的method方法，所有执行代理对象的方法都会被替换成invoke方法。

2. 创建被代理对象的接口
3. 创建实际要被代理的对象
4. 创建代理对象，并执行要代理的方法

我们可以把 `InvocationHandler` 看做一个中介类，中介类持有一个被代理对象，在 `invoke` 方法中调用了被代理对象的相应方法。

代理类调用自己方法时，通过自身持有的中介类对象来调用中介类对象的 `invoke` 方法，从而达到代理执行被代理对象的方法。也就是说，动态代理通过中介类实现了具体的代理功能。

动态代理的优点

Java 动态代理的优势是实现无侵入式的代码扩展，也就是方法的增强；让你可以在不用修改源码的情况下，增强一些方法；在方法的前后你可以做你任何想做的事情（甚至不去执行这个方法就可以）。此外，也可以减少代码量，如果采用静态代理，类的方法比较多时，得手写大量代码。

总结

代理模式是常用的 java 设计模式，他的特征是代理类与委托类有同样的接口，代理类主要负责为委托类预处理消息、过滤消息、把消息转发给委托类，以及事后处理消息等。代理类与委托类之间通常会存在关联关系，一个代理类的对象与一个委托类的对象关联，代理类的对象本身并不真正实现服务，而是通过调用委托类的对象的相关方法，来提供特定的服务。简单的说就是，我们在访问实际对象时，是通过代理对象来访问的，代理模式就是在访问实际对象时引入一定程度的间接性，因为这种间接性，可以附加多种用途。

认准一手QQ3195303913微信wxywd8