



Proxy Pattern 代理模式

Tom



代理模式的定义

代理模式（Proxy Pattern）是指为其他对象提供一种代理，以控制对这个对象的访问。

特点：代理对象在客服端和目标对象之间起到中介作用。

属于结构型设计模式。

生活中的代理模式



房产中介



快递小哥



黄牛党

代理模式的适用场景

保护目标对象

增强目标对象

咕泡出品，必属精品

静态代理

显式声明被代理对象

动态代理

动态配置和替换被代理对象

JDK动态代理的实现原理

- 1、拿到被代理类的引用，并且获取它的所有的接口（反射获取）。
- 2、JDK Proxy类重新生成一个新的类，实现了被代理类所有接口的方法。
- 3、动态生成Java代码，把增强逻辑加入到新生成代码中。
- 4、编译生成新的Java代码的class文件。
- 5、加载并重新运行新的class，得到类就是全新类。

CGLib和JDK动态代理对比

1. JDK动态代理是实现了被代理对象的接口，CGLib是继承了被代理对象。
2. JDK和CGLib都是在运行期生成字节码，JDK是直接写Class字节码，CGLib使用ASM框架写Class字节码，Cglib代理实现更复杂，生成代理类比JDK效率低。
3. JDK调用代理方法，是通过反射机制调用，CGLib是通过FastClass机制直接调用方法，CGLib执行效率更高。

CGLib动态代理容易踩的坑

1、无法代理final修饰的方法。

必属精品
咕泡出品
必属精品
咕泡出品
必属精品
咕泡出品
必属精品
咕泡出品

代理模式的优点

代理模式能将代理对象与真实被调用的目标对象分离。

一定程度上降低了系统的耦合程度，易于扩展。

代理可以起到保护目标对象的作用。

增强目标对象的职责

代理模式的缺点

代理模式会造成系统设计中类的数目增加

在客户端和目标对象之间增加了一个代理对象，请求处理速度变慢。

增加了系统的复杂度。

Spring中的代理选择原则

- 1、当Bean有实现接口时，Spring就会用JDK的动态代理。
- 2、当Bean没有实现接口时，Spring选择CGLib。
- 3、Spring可以通过配置强制使用CGLib，只需在Spring的配置文件中加入如下代码：

```
<aop:aspectj-autoproxy proxy-target-class="true"/>
```