

代理模式

<https://segmentfault.com/a/1190000011291179>

静态代理与动态代理的区别主要在：

- 静态代理在编译时就已经实现，编译完成后代理类是一个实际的class文件
- 动态代理是在运行时动态生成的，即编译完成后没有实际的class文件，而是在运行时动态生成类字节码，并加载到JVM中

cglib与动态代理最大的区别就是：

- 使用动态代理的对象必须实现一个或多个接口
- 使用cglib代理的对象则无需实现接口，达到代理类无侵入

总结

- 静态代理实现较简单，只要代理对象对目标对象进行包装，即可实现增强功能，但静态代理只能为一个目标对象服务，如果目标对象过多，则会产生很多代理类。
- JDK动态代理需要目标对象实现业务接口，代理类只需实现InvocationHandler接口。
- 动态代理生成的类为 `lass com.sun.proxy.$Proxy4`，cglib代理生成的类为 `class com.cglib.UserDao$$EnhancerByCGLIB$$552188b6`。
- 静态代理在编译时产生class字节码文件，可以直接使用，效率高。
- 动态代理必须实现InvocationHandler接口，通过反射代理方法，比较消耗系统性能，但可以减少代理类的数量，使用更灵活。
- cglib代理无需实现接口，通过生成类字节码实现代理，比反射稍快，不存在性能问题，但cglib会继承目标对象，需要重写方法，所以目标对象不能为final类。