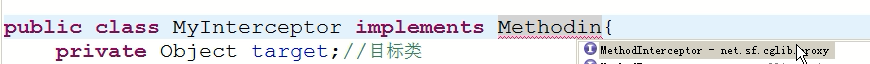
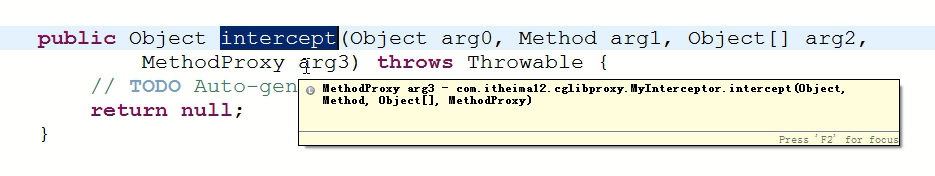


复制过来

修改拦截器





这个intercept方法和jdk proxy中的invoke是一样的

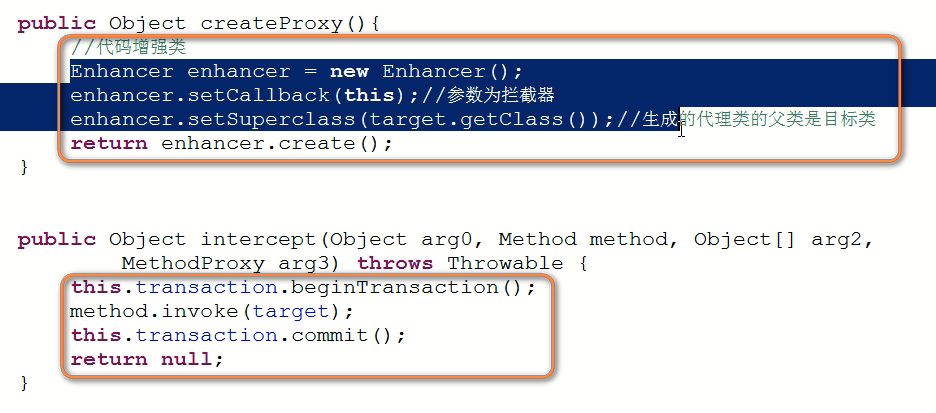
怎么生成代理对象的？

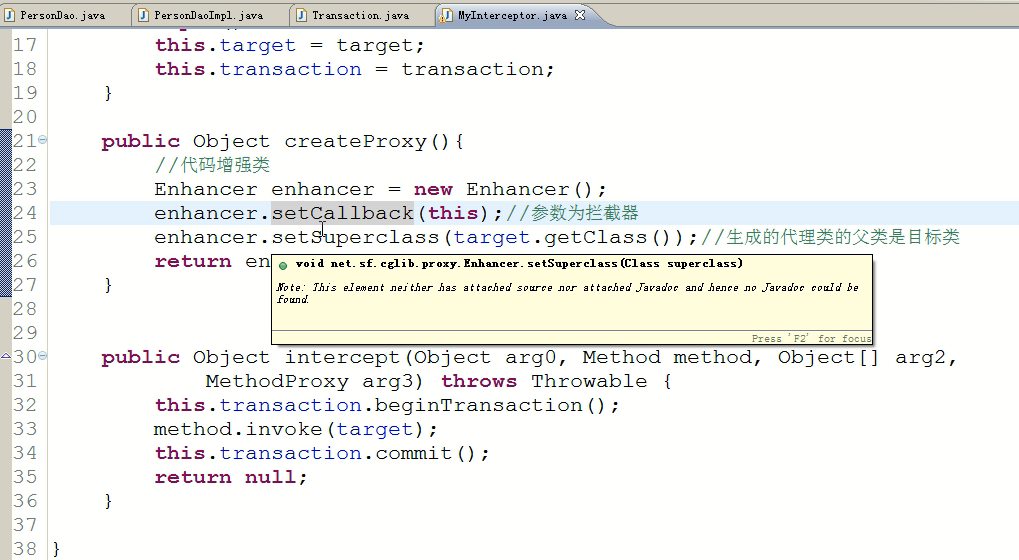
Enhancer enhancer = new Enhancer(); //代码增强类

enhancer.setCallback(this);//this 就是拦截器

enhancer.setSuperclass(target.getClass());//生成的代理类的父类是目标类

return enhancer.create();



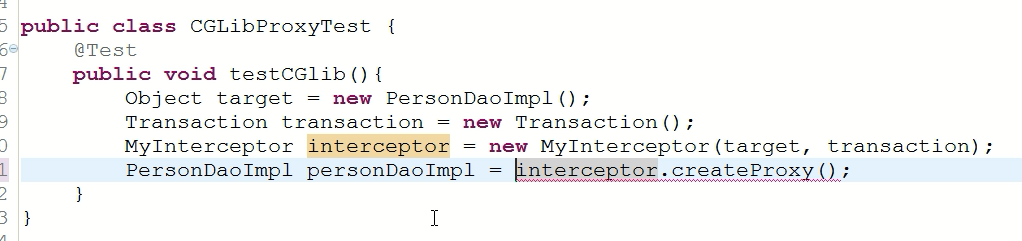


【可以看到 所有的生成过程 全是new出来的这一个enhancer完成的

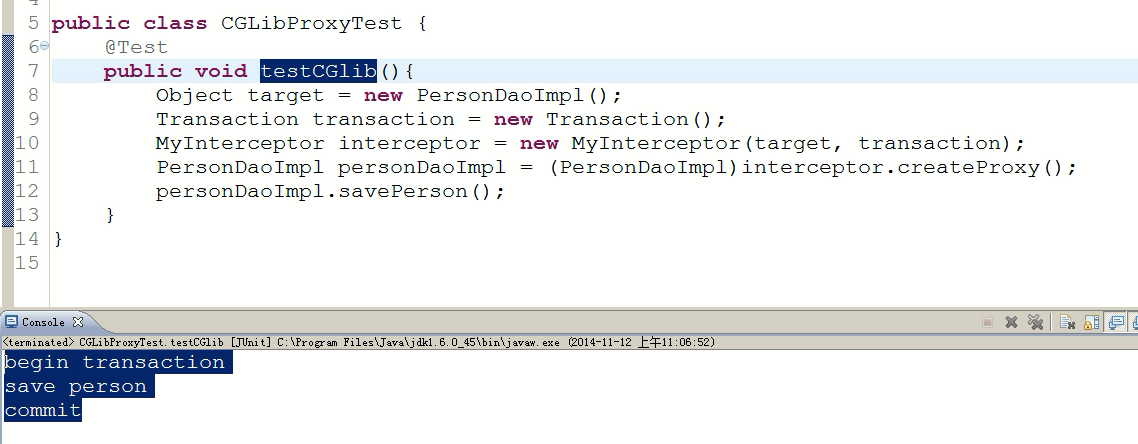
所以 你先要实现一个Interceptor --- net.sf.cglib.proxy 实现intercept方法（实际上就是jdk中的invoke方法） ---- 实现完成之后 在Enhancer中通过setCallback把这个interceptor设置进去 --- -Advice + Pointcut 也就是Advisor完成 （jdk的invocationHandler 和 cglib的Interceptor 都是Advisor 因为在invoke和intercept中 已经把Advice 和 Pointcut都结合好了）

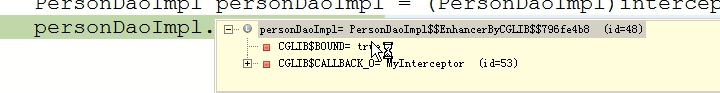
再然后 把你的Advisor和代理类的父类就是目标类 设置到enhancer里面 最后通过enhancer这个第三方类直接使用create生成代理类】

Cglib的动态代理的实现不需要实现接口



这里面PersonDaoImpl没有接口

【目标类就是代理类的父类 所以 强转之后是目标类】



Hibernate中load生成的代理类的子类 ---- 实际上是使用javaassist完成的