// 代理

Target t = new Target();

TargetInterface ti =(TargetInterface) Proxy.newProxyInstance(

t.getClass().getClassLoader(),

t.getClass().getInterfaces(),

new InvocationHandler(){

@Override

public Object invoke(Object proxy, Method method, Object[] args) throws Throwable {

System.out.println("test");

return method.invoke(t, args);

}

} );

ti.todo();

// 核心代码

byte[] classFile = ProxyGenerator.generateProxyClass("$Proxy1", Target.class.getInterfaces());

try(FileOutputStream fos = new FileOutputStream("./Proxy.class")){

fos.write(classFile);

fos.flush();

System.out.println("ok");

}catch (Exception e) {

//TODO: handle exception

}

// <https://www.jianshu.com/p/8dec701d4abe> 详解地址