1：实现代理模式的底层逻辑是什么？

**TOM:**

代理模式包括静态代理，也包括动态代理，并且也包括cglib和JDKProxy；

代理模式的底层逻辑，其实就是去找一个替身为目标代理对象去完成事情，并且去做一些增强的功能和动作，所以不管是静态代理也好，动态代理也好，都是去新创建一个类，去替原来的类做一些事情，并且在原来类的基础上，在前后做一些增强，这样才是代理模式的底层逻辑，只不过静态代理需要人工，手动去new一个类，相当于是手动挡，动态代理是我们用系统程序自动去生成一个类，相当于是自动挡。

2：为什么JDKProxy一定要求目标对象实现接口，二CGLibProxy对目标对象没有任何要求？

**TOM:**

主要体现在OOP的继承关系上，我们jdk它是用新生成的类去实现目标代理类的接口，因为如果你不实现接口的话，是没办法知道目标对象里面有哪些方法，而且我们去调用代理对象的方法之前，要先把代理对象拿到，并且将代理对象的引用赋值给目标代理对象的接口，如果说我们生成的代理对象不实现目标对象接口的话，这个动作是没办法赋值的，而CGLIB的思路还是满足OOP的关系，它不是通过实现接口，而是通过继承，生成一个新的类去继承目标类，他是通过方法重写的办法去对目标类的方法的功能增强；但是它有什么缺陷呢？它要求目标对象的方法不能加final关键字，大家可想而知，final关键字定义的目的是在于不允许子类重写，而cglib生成的代理类，恰恰是目标对象的子类，所以如果它如果加了final关键字，没办法去重写父类的方法，也就没法实现目标方法的增强。