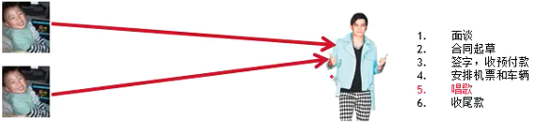
代理模式相当重要 在生活和软件开发中非常重要



代理的作用：通过代理 可以控制对额访问 ---- 比如 周杰伦为例 要求唱歌



可以一个人来做

一个人要做些流程 再来一个都要做 就是中间的唱歌不一样 别的都不做



这就产生了经纪人的概念 就是代理

客户先找经纪人来谈 然后 经济人就安排周杰伦唱歌

如果 这次要求另一个人来唱歌 就换一换

代理叫你什么时候唱 就什么时候唱

核心的唱歌就是真实的角色来做

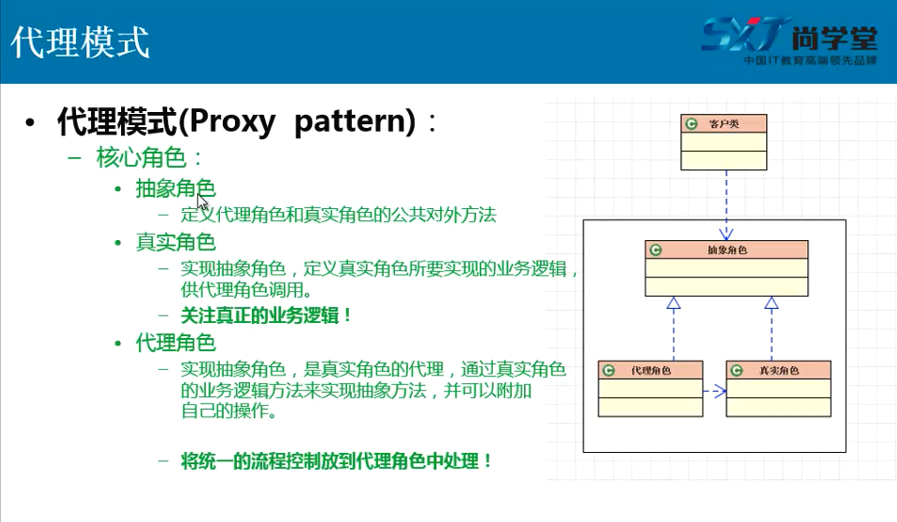
这就是代理模式 实际生活中的应用

比如买房 ---- 比如办理和房产相关的手续 就和中介公司来做

代理最后会变成一个非常重要的角色

和代理有关的一个模式就是面向切面编程 AOP

统一的流程放到代理类中来处理

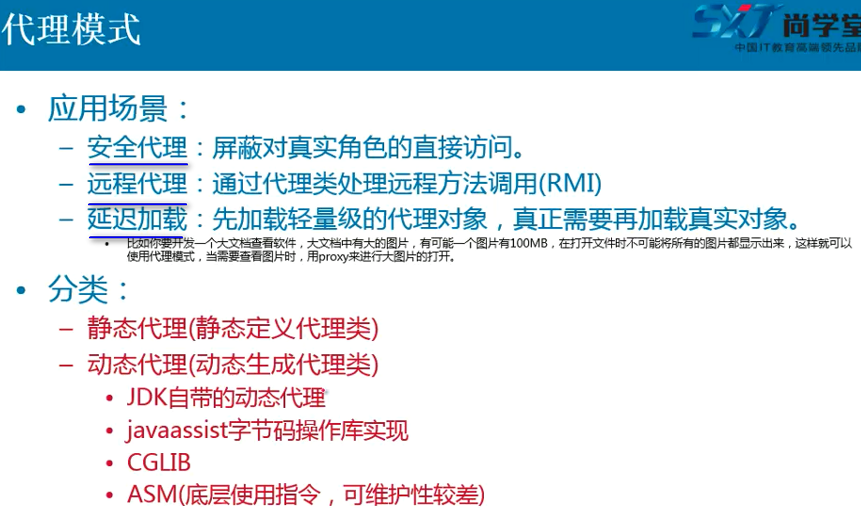


代理角色和真是角色都要遵循相同的抽象角色 遵守同样的接口 ----- 这样对客户来说 实际上 就是代理提供唱歌 调用的是真是角色的唱歌 但是 对外也要有唱歌的接口



代理应用的场景

比如是100M的图片 现在要用这个图片 隔了很久 才要大图片 那我就先给你小图片用



分成了静态代理 和动态代理

静态代理有我们自己定义代理类

动态代理由程序自动生成代理类 ----- 动态代理用的非常多 这个就是动态生成自己码

----- JDK自带动态代理

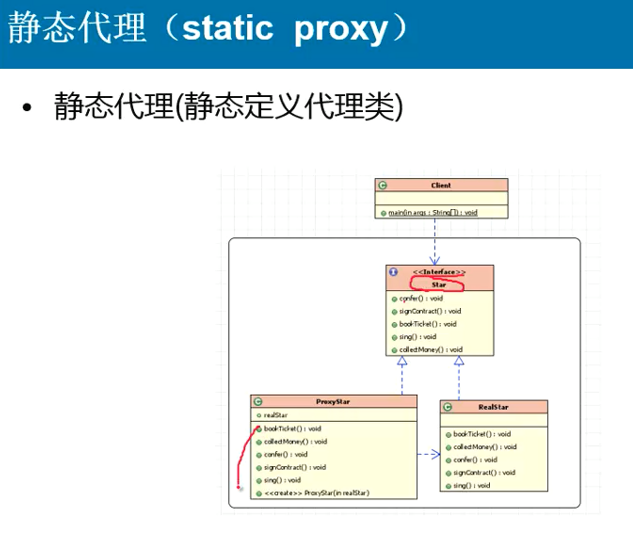
----- javaassist动态生成代理类

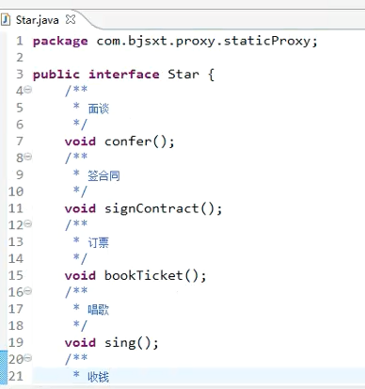
----- cglib

----- asm

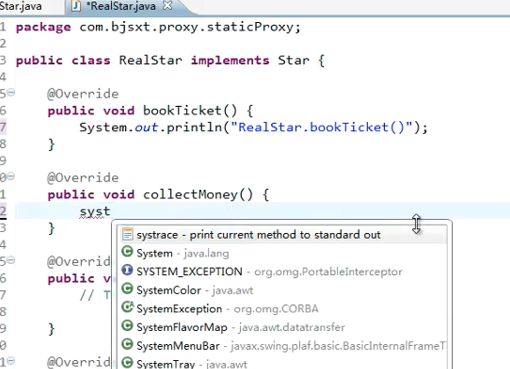
【高淇前面有两节专门讲这个javaassist】

-------------------- 静态代理 -------------------





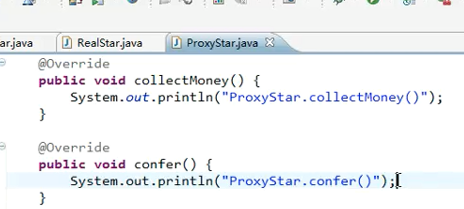
先定义这个真实的角色



就在sing这个方法里面 唱歌这里面

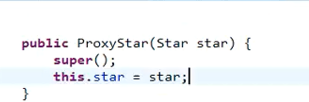


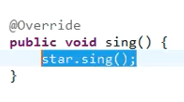
定义一个代理类



注意sing这个方法 代理人不能唱 调用realStar的方法

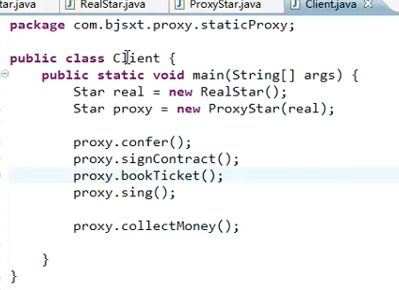
聚合这个对象





测试一下

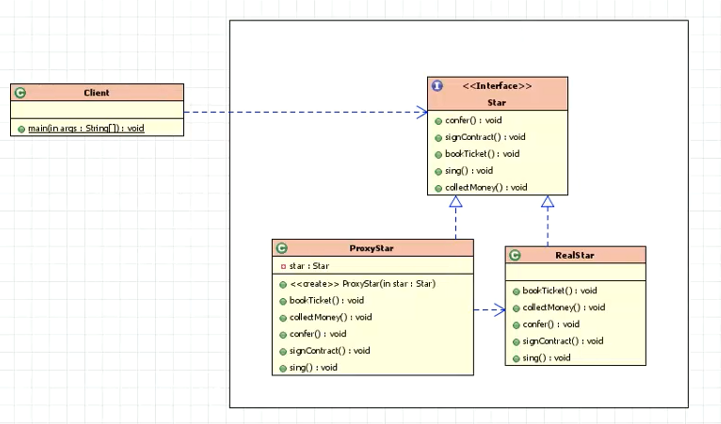
写一个Client



前面都是代理人在做 唱歌才是realStar在唱歌

这样 代理控制了真实的角色

画一下这个类图



下面看看动态代理



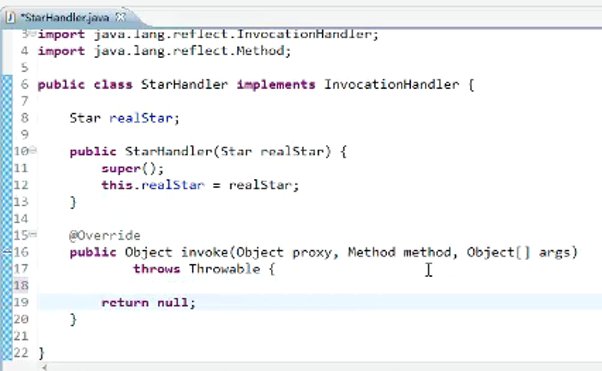
用的范围更普遍 更广泛

这样 动态代理类的内容不需要自己来定义 全部交给工具来生成

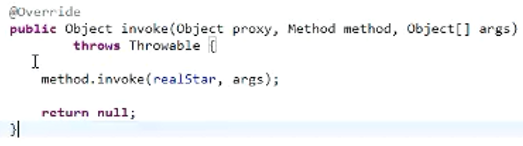


重点介绍JDK中的代理方式 其他之前的教程使用



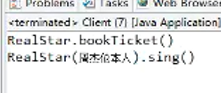


Invoke里面做统一的流程控制



首先需要一个realStar 还有一个是处理器接口对象StarHandler 还需要代理类对象





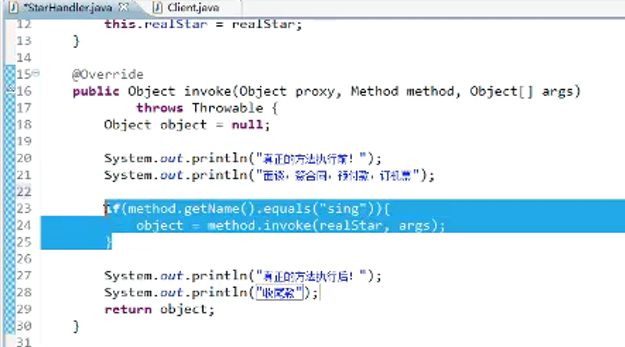
调用了代理类的方法 全部进入invoke方法

调用代理类 都进入invoke方法 我们就可以做统一的流程控制

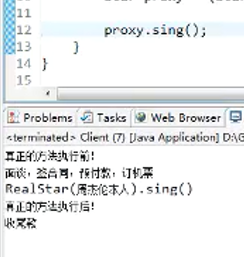
所以 我就看是不是唱歌 只有唱歌 我才去做代理

如果不是 我就做别的处理

有返回值的情况



这种方式 就是JDK中提供的方式



Sign的前后做了相关的处理