**静态代理和动态代理**

Proxy代理模式是一种结构型设计模式，主要解决的问题是：在直接访问对象时带来的问题。代理是一种常用的设计模式，其目的就是为其他对象提供一个代理以控制对某个对象的访问。代理类负责为委托类预处理消息，过滤消息并转发消息，以及进行消息被委托类执行后的后续处理。

为了保持行为的一致性，代理类和委托类通常会实现相同的接口，所以在访问者看来两者没有丝毫的区别。通过代理类这中间一层，能有效控制对委托类对象的直接访问，也可以很好地隐藏和保护委托类对象，同时也为实施不同控制策略预留了空间，从而在设计上获得了更大的灵活性。

对于动态代理和静态代理，我是通过百度才更好的理解它的。对于老师布置这个员工工资问题，对于静态代理，就是需要我们自己去创建一个接口employee,其他类通过访问这个接口才能输出该有的工资，还有就是在company这个类里面有一个随机数的应用，我写的是5个数的循环，实际运行起来每一次的运行结果都是不一样的。对于动态代理，就是不用用户去创建接口，他有一个类dynamic里面用来做那个动态链接，同样，它也不予要随机数，有一个Proxy.*newProxyInstance*，它的参数就是接口方，接口，连接方，直接就可以动态代理，动态代理的代码，我只输出了一个manager,后续的worker可以再写一个输出函数将他输出就行了。对于java，基本的语法我还余姚更加女里去学，多动手呀