# Java动态代理和静态代理

1. 代理是一种模式，提供了对目标对象的间接访问方式，即通过代理访问目标对象。如此便于在目标实现的基础上增加额外的功能操作，前拦截，后拦截等，以满足自身的业务需求，同时代理模式便于扩展目标对象功能的特点也为多人所用。
2. 静态代理：静态代理的实现比较简单，代理类通过实现与目标对象相同的接口，并在类中维护一个代理对象。通过构造器塞入目标对象，赋值给代理对象，进而执行代理对象实现的接口方法，并实现前拦截，后拦截等所需的业务功能。
3. 动态代理：动态代理是指动态的在内存中构建代理对象（需要我们制定要代理的目标对象实现的接口类型），即利用JDK的API生成指定接口的对象，也称之为JDK代理或者接口代理。
4. 总结：静态代理虽然可以做到不对目标对象进行修改的前提下，对目标对象进行功能的扩展和拦截，但是因为代理对象，需要实现与目标对象一样的接口，会导致代理类十分繁多，不易维护，同时一旦接口增加方法，则目标对象和代理类都需要维护。动态代理就是代理对象无需实现接口，免去了编写很多代理类的烦恼，同时接口增加方法也无需再维护目标对象和代理对象，只需在事件处理器中添加对方法的判断即可，不过代理对象不需要实现接口，但是目标对象一定要实现接口，否则无法使用JDK动态代理。