**Java学习**

1. 代理模式

代理模式是常用的[Java](http://lib.csdn.net/base/java" \o "Java 知识库" \t "http://blog.csdn.net/fighterandknight/article/details/_blank)设计模式，他的特征是**代理类与委托类有同样的接口，代理类主要负责为委托类预处理消息、过滤消息、把消息转发给委托类，以及事后处理消息等**。主要用来做方法的增强，让你可以在不修改源码的情况下，增强一些方法，在方法执行前后做任何你想做的事情。

1. 静态代理

在程序运行前，代理类的.class文件就已经存在了。

*/\*\*  
 \* 账户查询接口  
 \** ***@author*** *wdongsen@linesum.com  
 \** ***@data*** *2017-08-31 16:35  
 \*/*public interface ICount {  
  
 // 查询账户  
 void queryCount();  
}

*/\*\*  
 \* 账户查询接口实现(委托类)  
 \** ***@author*** *wdongsen@linesum.com  
 \** ***@data*** *2017-08-31 16:42  
 \*/*public class CountImpl implements ICount{  
  
 @Override  
 public void queryCount() {  
 System.*out*.println("");  
 }  
}

*/\*\*  
 \* 静态代理  
 \** ***@author*** *wdongsen@linesum.com  
 \** ***@data*** *2017-08-31 16:57  
 \*/*public class CountProxy implements ICount{  
  
 private CountImpl countImpl;  
  
 public CountProxy(CountImpl countImpl) {  
 this.countImpl = countImpl;  
 }  
  
 @Override  
 public void queryCount() {  
 System.*out*.println("处理之前。。。");  
 countImpl.queryCount();  
 System.*out*.println("处理之后。。。");  
 }  
}

1. 动态代理