**Java基础**

***1. Java 8大基本数据类型***

整型 int，短整型 short，长整型 long，

字节型 byte，布尔型 Boolean

字符型 char

单精度浮点数 float，双精度浮点数 double。

***2. final关键词***

final关键词适用的范围:数据、方法和类

***2.1 final修饰数据***

编译器常数:在编译期间初始化一个值，它永远不会改变

运行时常数:运行初始化后，并不希望它发生改变

如果是编译期间，final一个基本数据类型，那么会节省一些运行时开销，但是前提是被final修饰的数据必须属于基本数据类型

如果final修饰的是一个对象的变量，那么就说明这个对象的引用句柄是不能变的，不能指向到其他对象，但是对象本身所包含的属性是可以被改变的

final可以编译时设定一个空白值，但是必须在构造函数里面初始化，要不然会编译时就发生异常

final自变量:就是方法的请求参数用final修饰，这就意味着在方法内部，final的变量句柄不能改变其指向。

***2.2 final修饰方法***

不允许子类重写该方法

***2.3 final修饰类***

表明不希望这个类被继承

***3. ArrayList集合和LinkedList集合的优点***

ArrayList实现了长度可变的数组，在内存中分配连续的空间。遍历元素和随机访问元素的效率比较高

LinkedList采用链表存储方式。插入、删除元素时效率比较高

***4. Hashtable和HashMap的异同***

Hashtable继承Dictionary类，HashMap实现Map接口

Hashtable线程安全， HashMap线程非安全

Hashtable不允许null值，HashMap允许null值

***5. ArrayList和Vector的区别***

Vector是线程安全的，也就是说是它的方法之间是线程同步的，而ArrayList是线程序不安全的。单线程访问用ArrayList效率要高一些，如果考虑到多线程并发的情况还是需要用Vector

ArrayList与Vector都有一个初始的容量大小，当存储进它们里面的元素的个数超过了容量时，就需要增加ArrayList与Vector的存储空间,Vector增加的是原来的2倍，而ArrayList是1.5倍

***6. 动态代理***

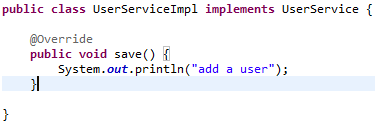
jdk动态代理:JDK的动态代理机制只能代理实现了接口的类

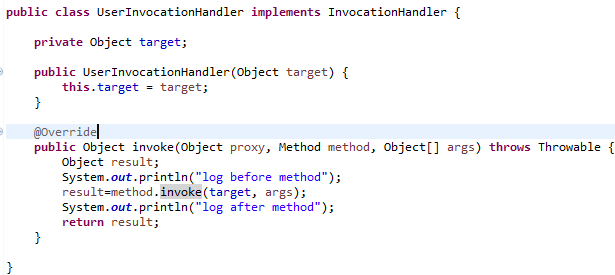
cglib动态代理:cglib是针对类来实现代理的，他的原理是对指定的目标类生成一个子类，并覆盖其中方法实现增强，但因为采用的是继承，所以不能对final修饰的类进行代理。

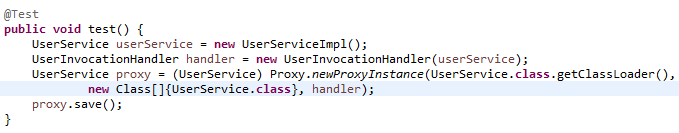
***6.1 jdk动态代理***

做好业务逻辑接口以及实现类，然后在做一个动态代理处理类(就是做一个类去实现InvocationHandler)

在使用的时候，直接用Proxy.newProxyInstance获取动态代理







***6.2 cglib动态代理***

用cglib的Enhancer去将需要代理的对象包裹起来。再利用Enhancer去创建对象

