

不变模式

整理自：《java与模式》之不变模式

在阎宏博士的《JAVA与模式》一书中开头是这样描述不变（Immutable）模式的：

一个对象的状态在对象被创建之后就不再变化，这就是所谓的不变模式。

不变模式的结构

不变模式可增强对象的强壮型(robustness)。不变模式允许多个对象共享某一个对象，降低了对该对象进行并发访问时的同步化开销。如果需要修改一个不变对象的状态，那么就需要建立一个新的同类型对象，并在创建时将这个新的状态存储在新对象里。

不变模式只涉及到一个类。一个类的内部状态创建后，在整个生命周期都不会发生变化时，这样的类称作不变类。这种使用不变类的做法叫做不变模式。不变模式有两种形式：一种是弱不变模式，另一种是强不变模式。

弱不变模式

一个类的实例的状态是不可改变的；但是这个类的子类的实例具有可能会变化的状态。这样的类符合弱不变模式的定义。要实现弱不变模式，一个类必须满足下面条件：

第一、所考虑的对象没有任何方法会修改对象的状态；这样一来，当对象的构造函数将对象的状态初始化之后，对象的状态便不再改变。

第二、所有属性都应当是私有的。不要声明任何的公开的属性，以防客户端对象直接修改任何的内部状态。

第三、这个对象所引用到的其他对象如果是可变对象的话，必须设法限制外界对这些可变对象的访问，以防止外界修改这些对象。如何可能，应当尽量在不变对象内部初始化这些被引用的对象，而不要在客户端初始化，然后再传入到不变对象内部来。如果某个可变对象必须在客户端初始化，然后再传入到不变对象里的话，就应当考虑在不变对象初始化的时候，将这个可变对象复制一份，而不要使用原来的拷贝。

弱不变模式的缺点是：

第一、一个弱不变对象的子对象可以是可变对象；换言之，一个弱不变对象的子对象可能是可变的。

第二、这个可变的子对象可能可以修改父对象的状态，从而可能会允许外界修改父对象的状态。

强不变模式

一个类的实例不会改变，同时它的子类的实例也具有不可变化的状态。这样的类符合强不变模式。要实现强不变模式，一个类必须首先满足弱不变模式所要求的所有条件，并且还有满足下面条件之一：

第一、所考虑的类所有的方法都应当是final，这样这个类的子类不能够置换掉此类的方法。

第二、这个类本身就是final的，那么这个类就不可能会有子类，从而也就不可能有被子类修改的问题。

“不变”和“只读”的区别

“不变”(Immutable)与“只读”(Read Only)是不同的。当一个变量是“只读”时，变量的值不能直接改变，但是可以在其他变量发生改变的时候发生改变。

比如，一个人的出生年月日是“不变”属性，而一个人的年龄便是“只读”属性，不是“不变”属性。随着时间的变化，一个人的年龄会随之发生变化，而人的出生年月日则不会变化。这就是“不变”和“只读”的区别。

不变模式在JAVA中的应用

不变模式在JAVA中最著名的应用便是java.lang.String类。String类是一个强不变类型，在出现如下的语句时：

```
String a = "test";  
String b = "test";  
String c = "test";
```

JAVA虚拟机其实只会创建这样一个字符串的实例，而这三个String对象都在共享这一个值。

不变模式的优点和缺点

不变模式有很明显的优点：

(1) 因为不能修改一个不变对象的状态，所以可以避免由此引起的不必要的程序错误；换言之，一个不变的对象要比可变的对象更加容易维护。

(2) 因为没有任何一个线程能够修改不变对象的内部状态，一个不变对象自动就是线程安全的，这样就可以省掉处理同步化的开销。一个不变对象可以自由地被不同的客户端共享。

不变模式的缺点：

不变模式唯一的缺点是：一旦需要修改一个不变对象的状态，就只好创建一个新的同类对象。在需要频繁修改不变对象的环境里，会有大量的不变对象作为中间结果被创建出来，再被JAVA垃圾收集器收集走。这是一种资源上的浪费。

在设计任何一个类的时候，应当慎重考虑其状态是否有需要变化的可能性。除非其状态有变化的必要，不然应当将它设计成不变类。