& 按位与操作

只有对应的两个二进制数为1时，结果位才为1

1&1=1

1&0=0

| 按位或操作

有一个位为1时，结果位就为1

1|1=1

1|0=1

0|0=0

& 和 && 都可以实现 和 这个功能

区别：& 两边都运算，⽽ && 先算 && 左侧，若左侧为false 那么右侧就不运算，判断语句中

推荐使⽤ &&，效率更⾼

| 和 || 和上⾯类似

区别：||只要满⾜第⼀个条件，后⾯的条件就不再判断，⽽|要对所有的条件进⾏判断

把&&和||

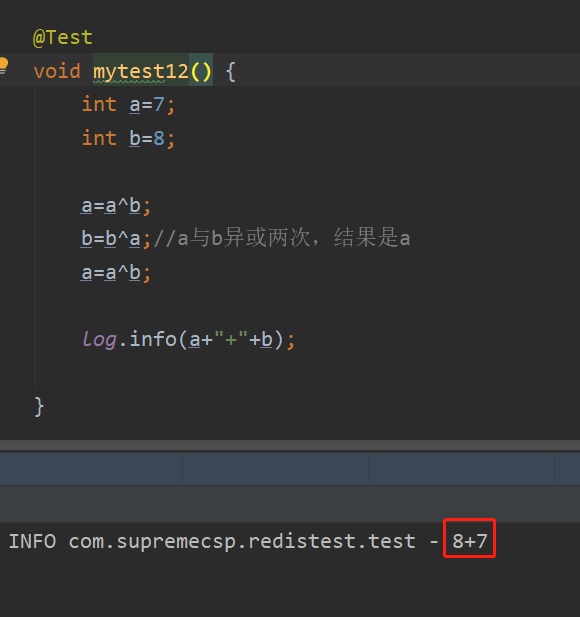
⽤最有效率的⽅法计算2乘以8

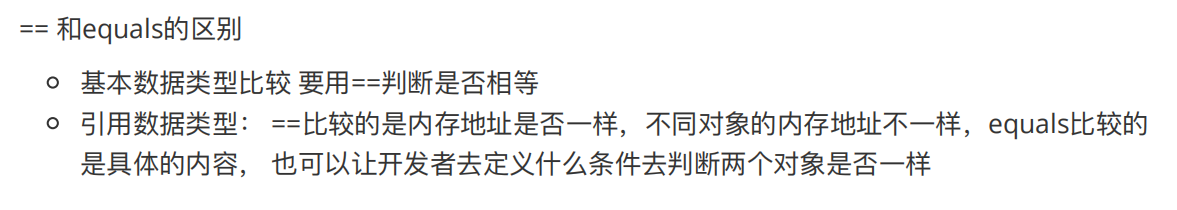
原理：将⼀个数左移n位，相当于乘以2的n次⽅，位运算是CPU直接⽀持的，所以效率⾼

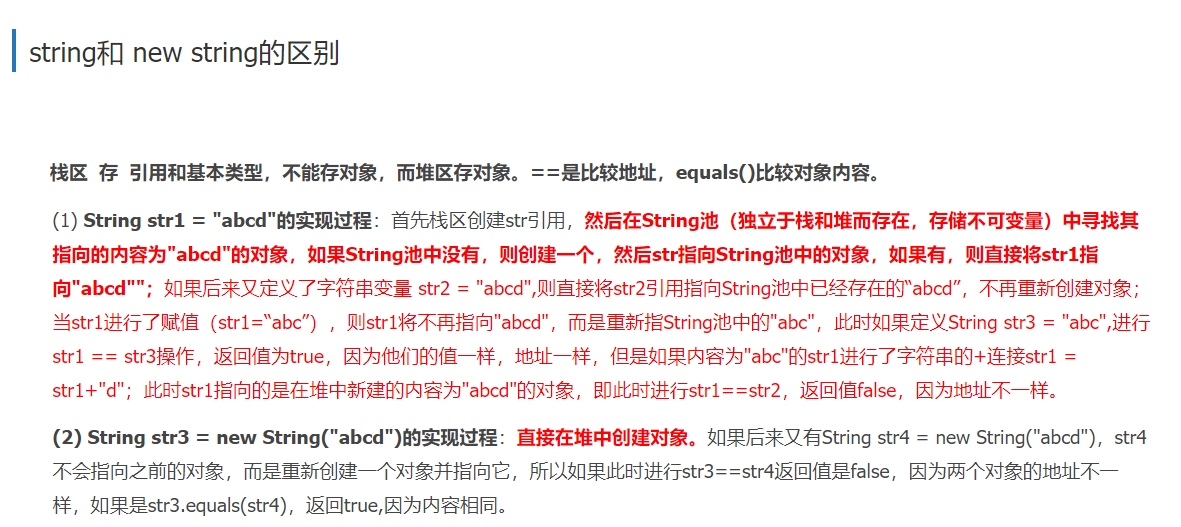
答案：2<<3 向左移3位

直接是⼆进制操作了

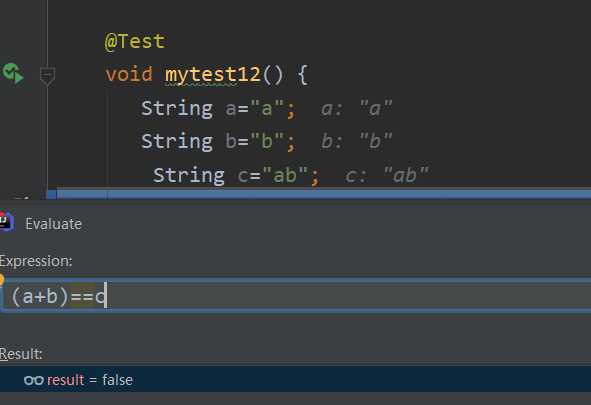
异或运算 (⼀个数与另⼀个数异或两次是其本身， ⼀个数和⾃身异或结果是0 )







一个常量池，一个堆

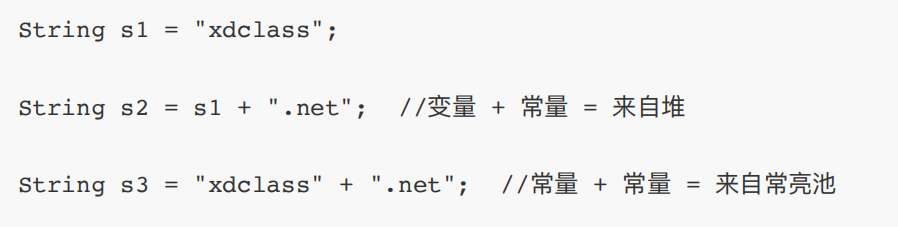


就算是常量池，拼起来结果也不同

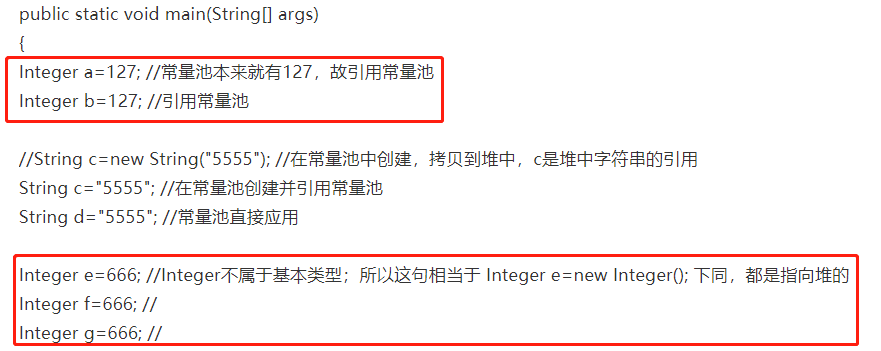
JDK7之后的写法，JDK9⼜进⾏了改良，但是变化不⼤，记住下⾯的写法即可

需要关闭的资源只要实现了java.lang.AutoCloseable，就可以⾃动被关闭

try()⾥⾯可以定义多个资源，它们的关闭顺序是最后在try()

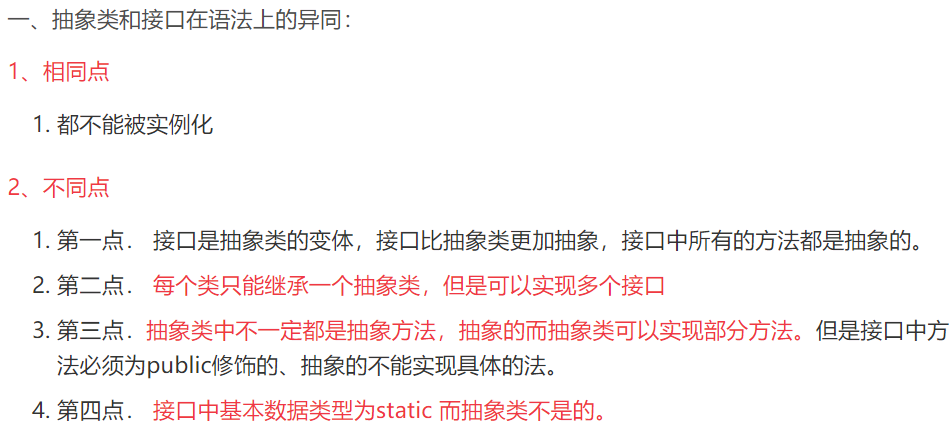


你是在疑问为什么最后三个不相等是吧？整型在常量池中应该是有数值限制，127在范围内，而666在范围外，所以127可以在常量池内创建，而666则是在堆内创建新对象





final修饰的引用类型,引用不可以改变,但是它所指向的对象的内容可以改变（String只有他自己，内容就是本身）



接口也可以是default修饰此时必须实现方法体

继承

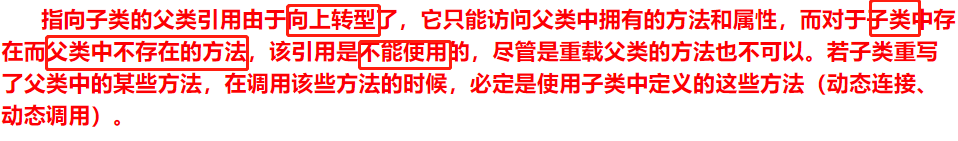
⼦类继承⽗类的特征和⾏为，使得⼦类对象具有⽗类的⽅法和属性

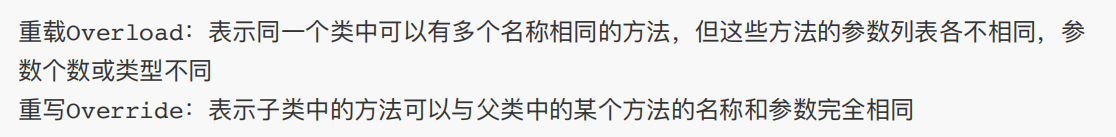
模板设计模式时候抽象类，

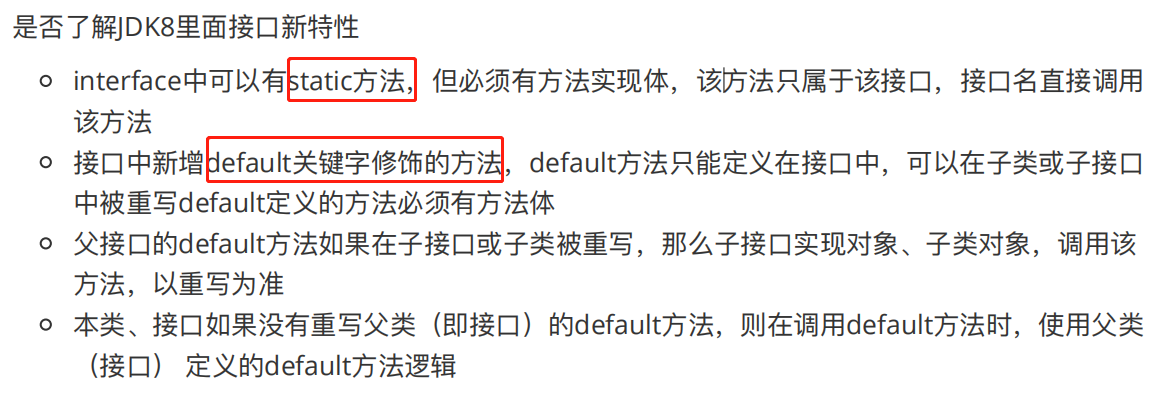
父类先定好方法模板并对模板中的方法，给出一定实现，子类通过重写方法调用父类的模板得到结果

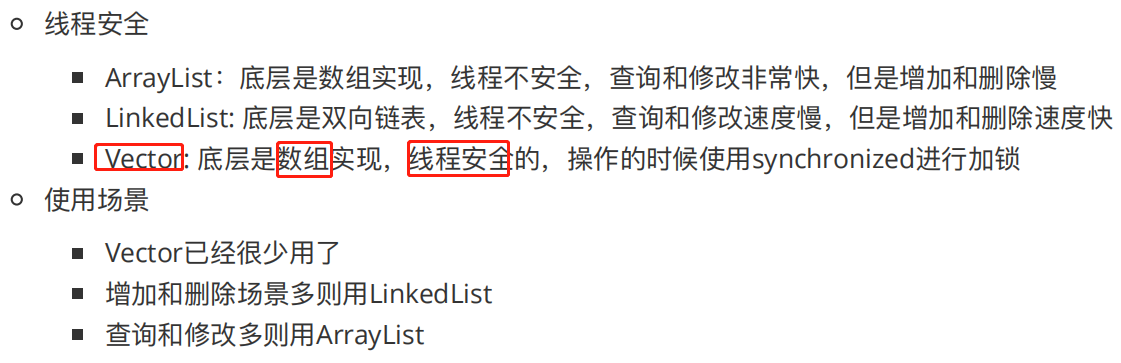


<https://blog.csdn.net/hchhan/article/details/82422721>



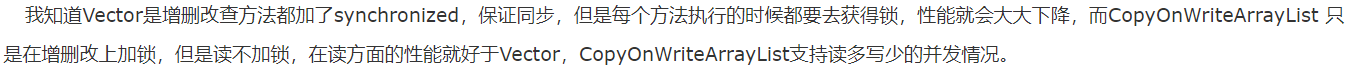


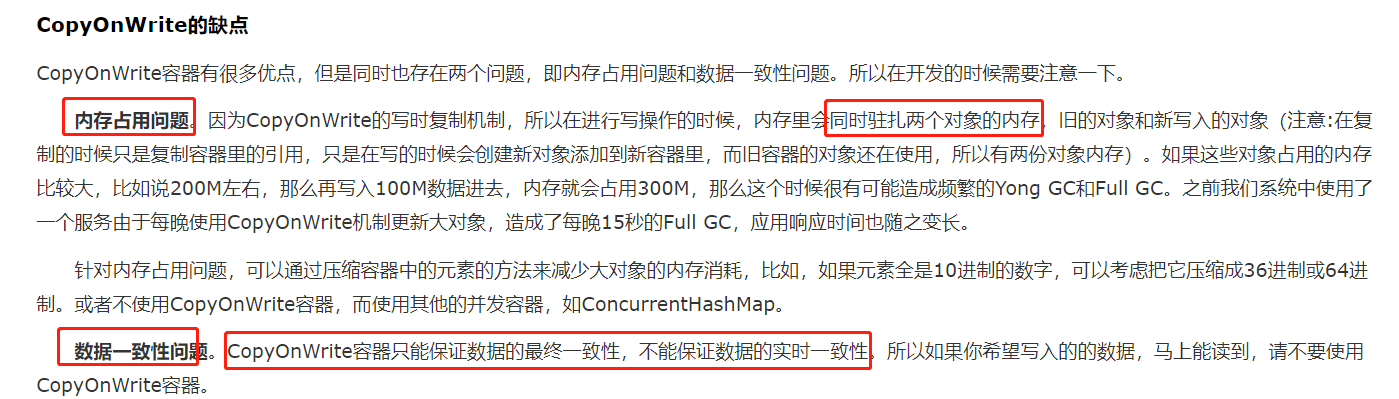


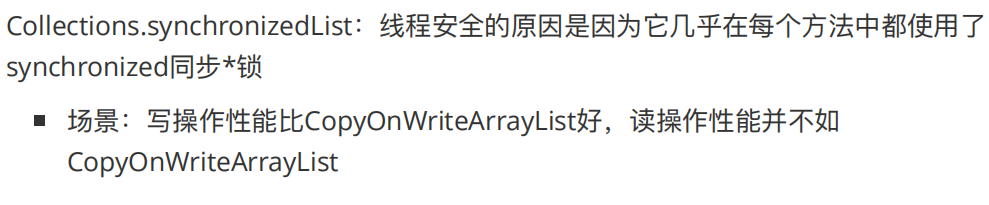


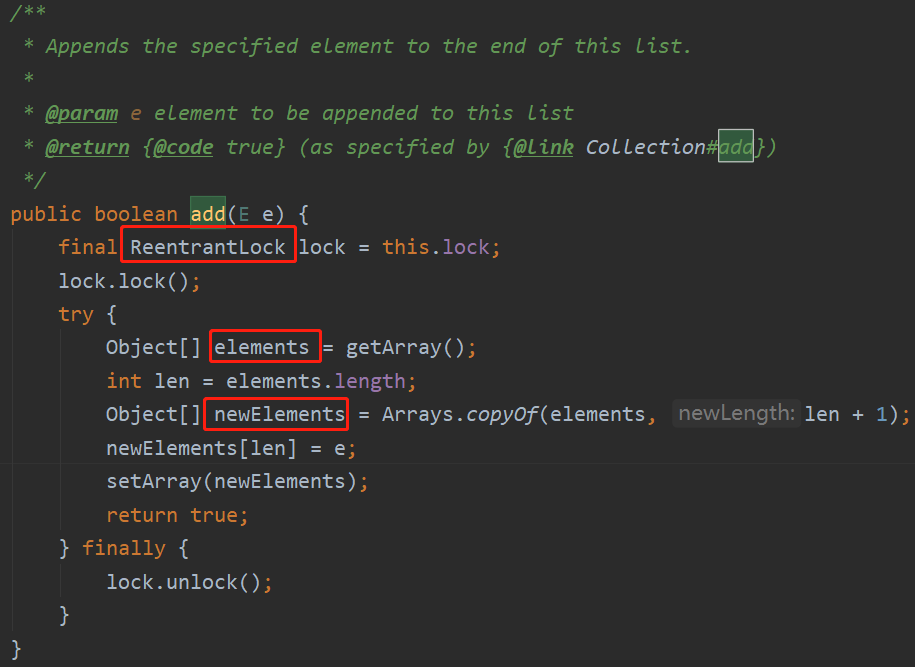


 List<String> list = Collections.synchronizedList(new ArrayList<String>());



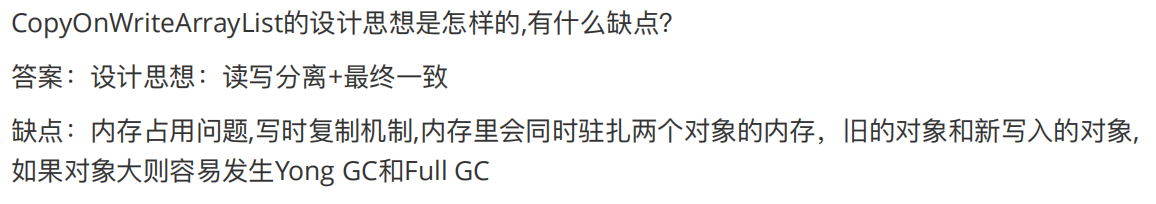


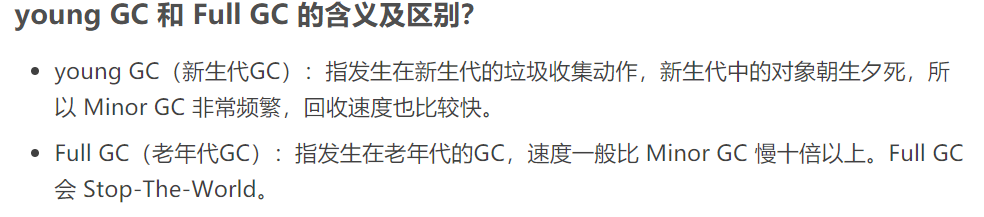




CopyOnWriteArrayList读的时候读旧的，拷贝完才读到新的（读的时候不加锁）

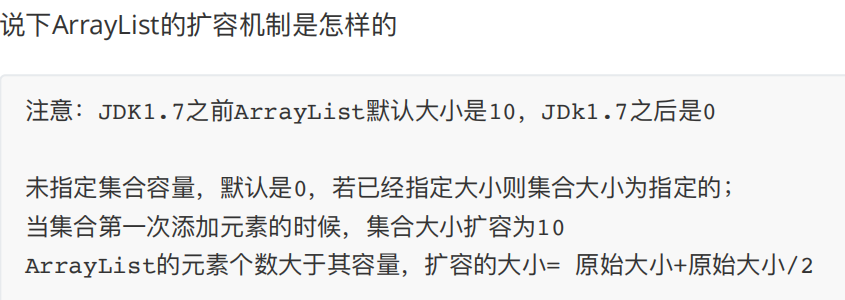
所以才说的最终一致

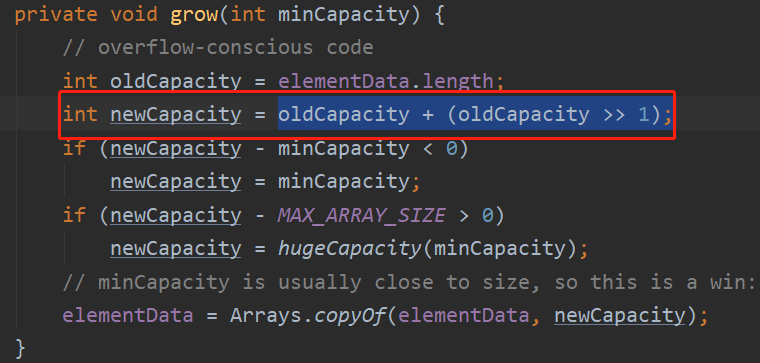


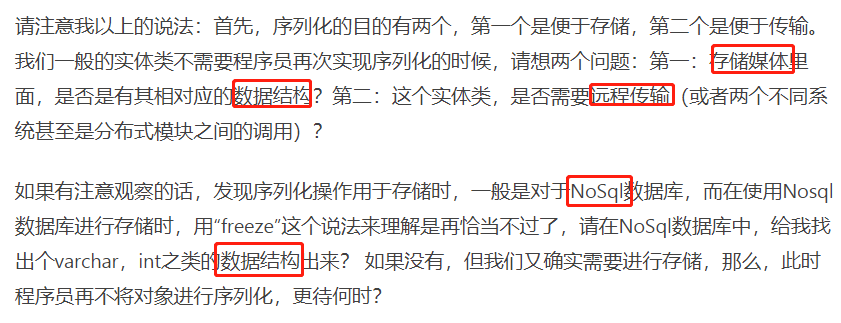


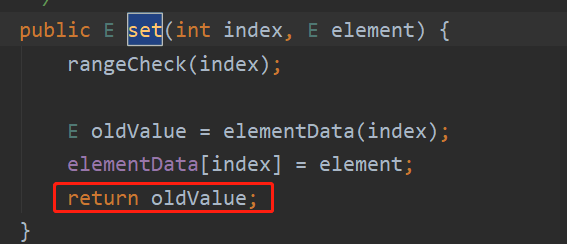


Full GC会触发stop the world,短暂暂停用户线程



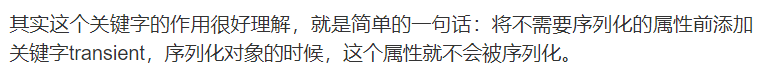


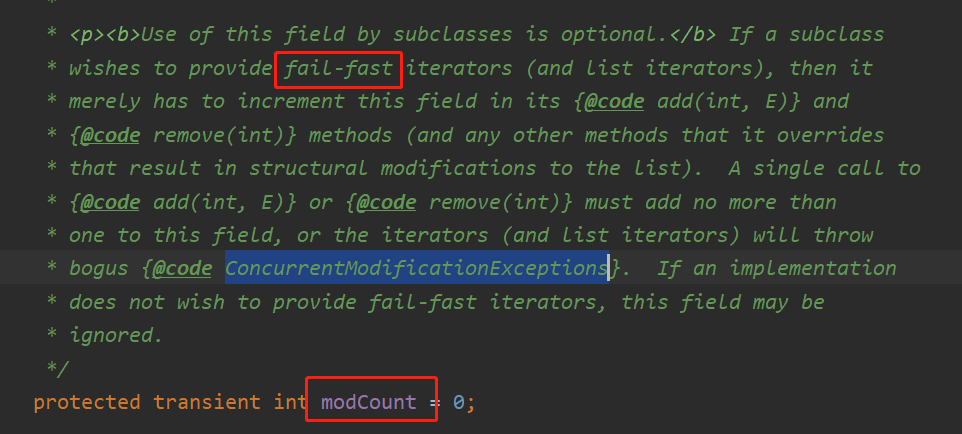




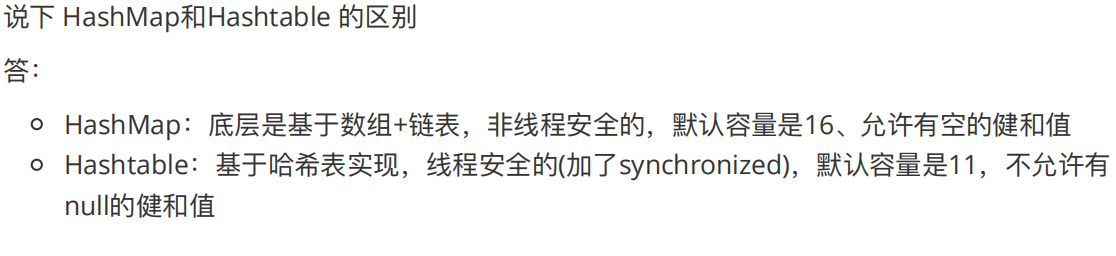
List的set返回Old数据

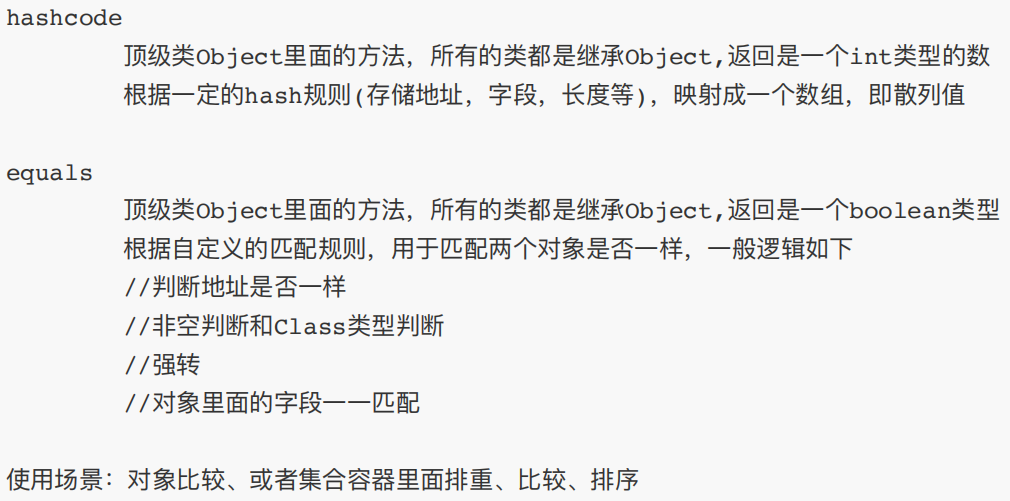






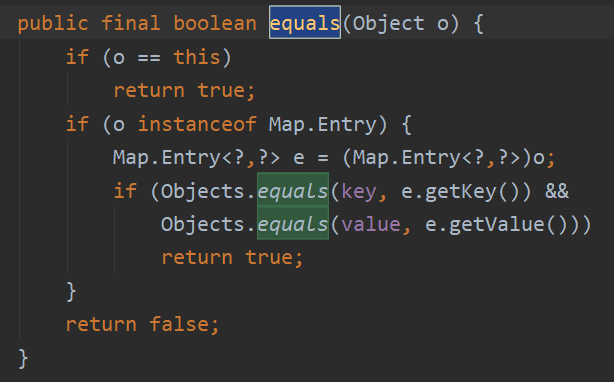
<https://blog.csdn.net/zymx14/article/details/78394464>





Hashcode由于碰撞可能重复但是快

Map equals



重写的equals地址和数值相等都算相等

<https://www.jianshu.com/p/800a44c1d9dd>