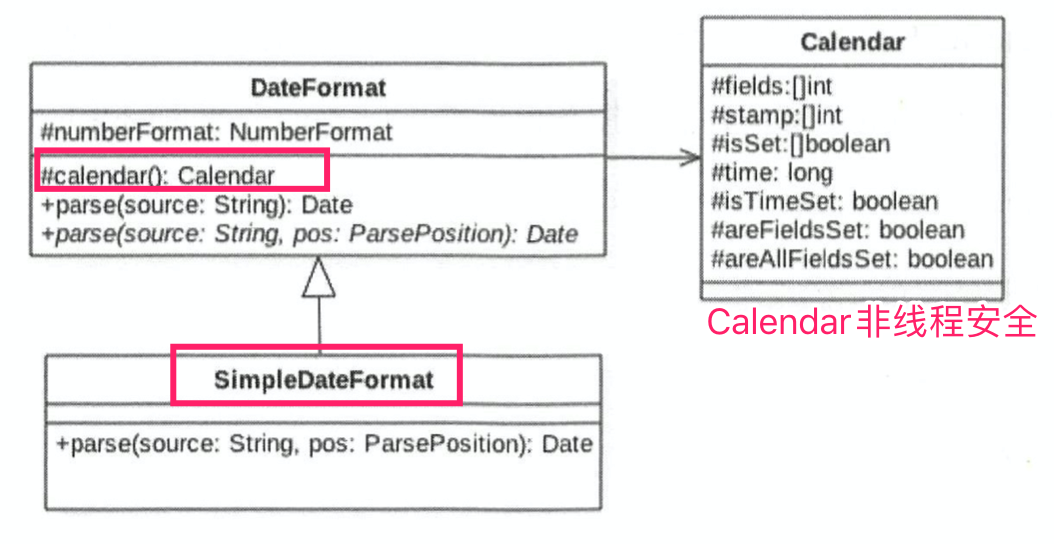
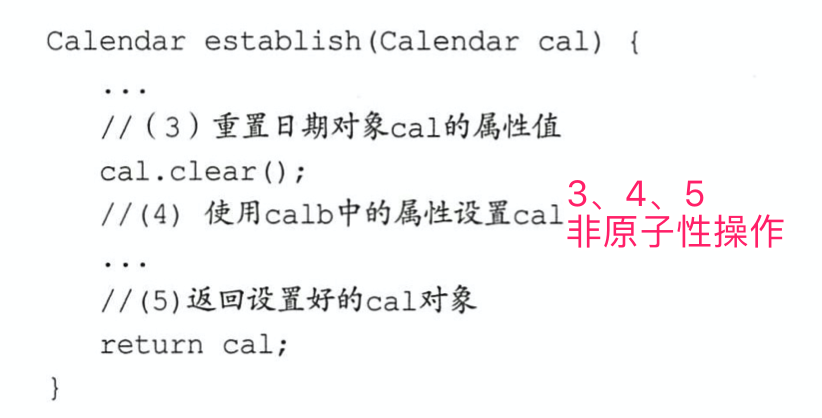
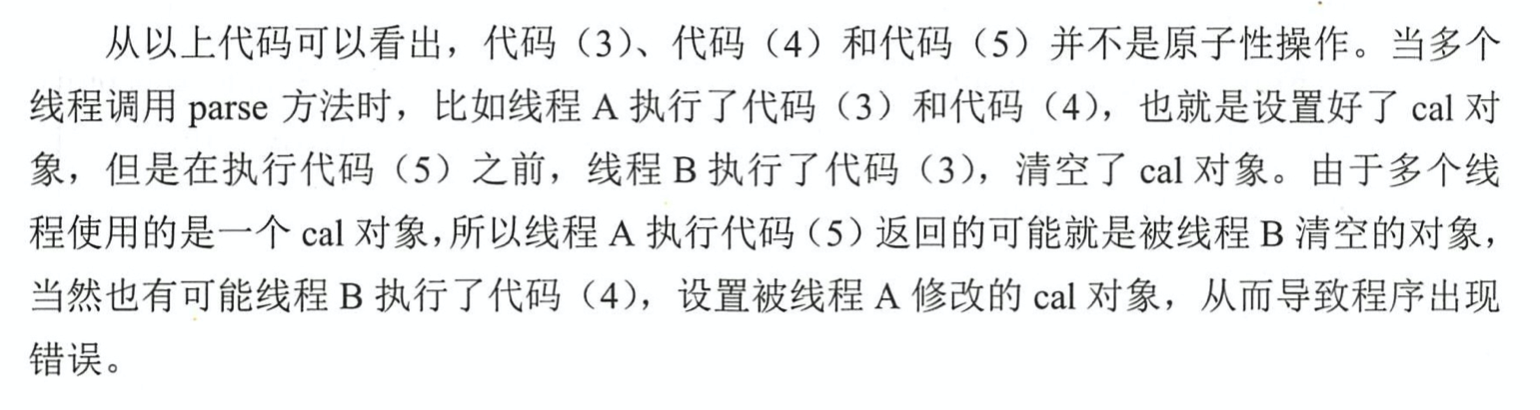
1. java.text.SimpleDateFormat不是线程安全的



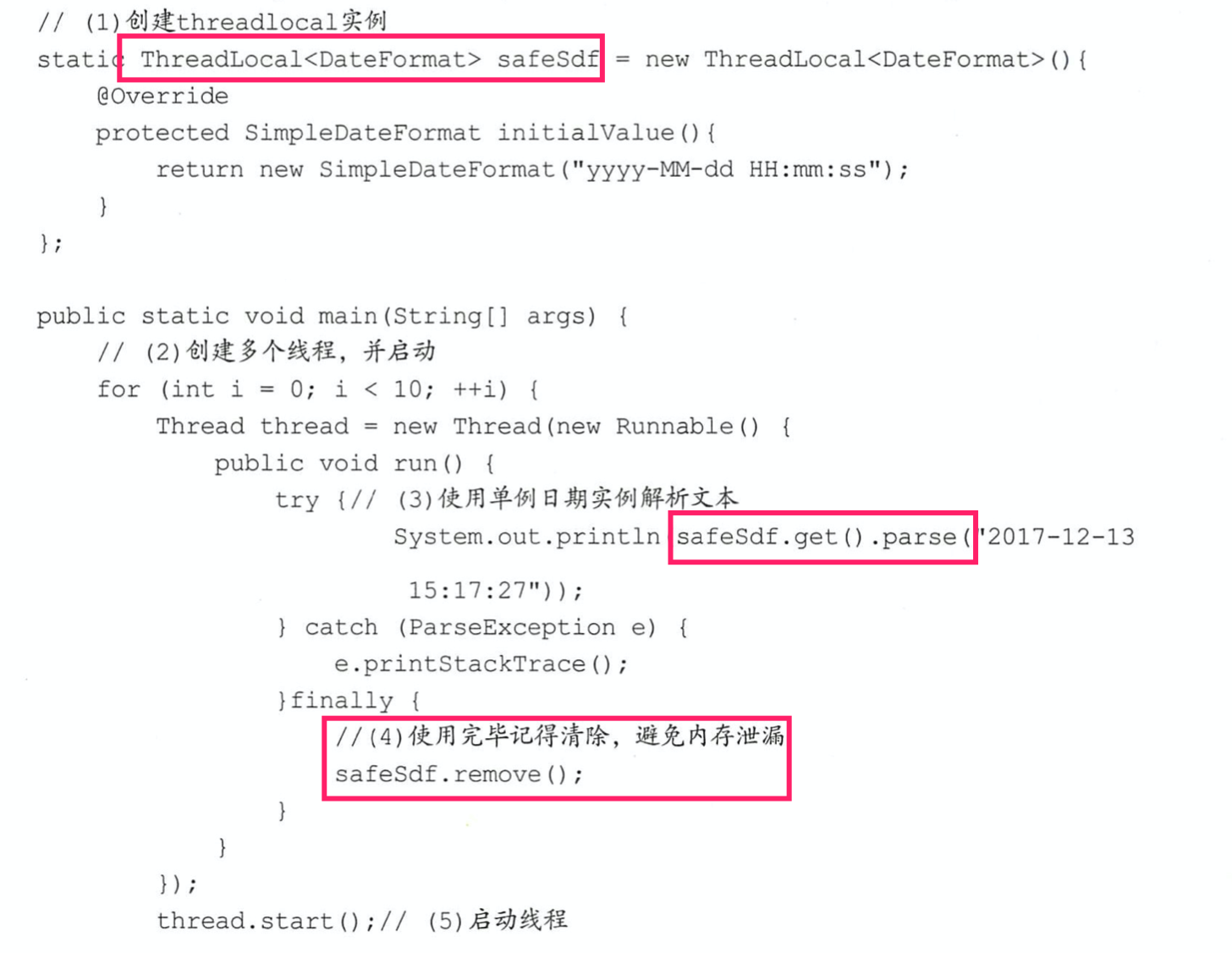




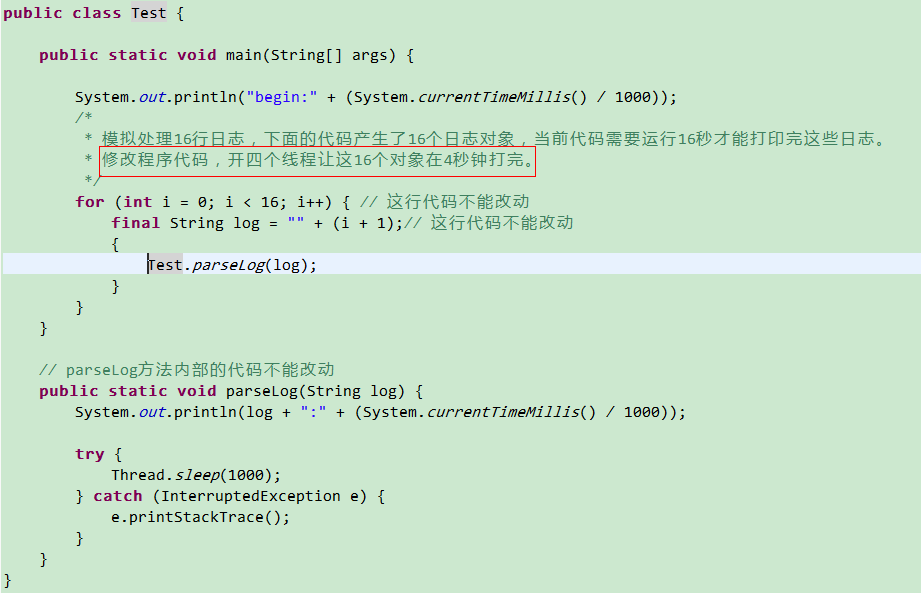


解决办法

* 1. 每次使用都new一个新的SimpleDateFormat实例，缺点是开销会比较大
  2. 调用方法前手动同步，比如synchronized(sdf)，缺点是高并发下线程竞争锁激烈，导致性能下降
  3. 使用ThreadLocal



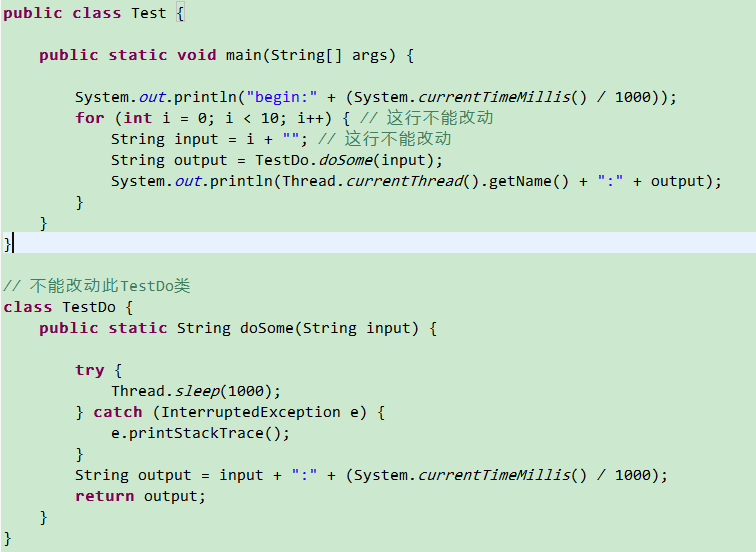
1. 简单面试题
   1. 题目：现有的程序代码模拟产生了16个日志对象，并且需要运行16秒才能打印完这些日志，请在程序中增加4个线程去调用parseLog()方法来分头打印这16个日志对象，程序只需要运行4秒即可打印完这些日志对象。原始代码如下：



* 1. 解答(采用阻塞队列，省略parseLog方法)



1. 题目2：
   1. 现成程序中的Test类中的代码在不断地产生数据，然后交给TestDo.doSome()方法去处理，就好像生产者在不断地产生数据，消费者在不断消费数据。请将程序改造成有10个线程来消费生成者产生的数据，这些消费者都调用TestDo.doSome()方法去进行处理，故每个消费者都需要一秒才能处理完，程序应保证这些消费者线程依次有序地消费数据，只有上一个消费者消费完后，下一个消费者才能消费数据，下一个消费者是谁都可以，但要保证这些消费者线程拿到的数据是**有顺序的**。原始代码如下：



* 1. 解答(省略TestDo类)



1. 题目3
   1. 现有程序同时启动了4个线程去调用TestDo.doSome(key, value)方法，由于TestDo.doSome(key, value)方法内的代码是先暂停1秒，然后再输出以秒为单位的当前时间值，所以，会打印出4个相同的时间值，如下所示：

4:4:1258199615

1:1:1258199615

3:3:1258199615

1:2:1258199615

请修改代码，如果有几个线程调用TestDo.doSome(key, value)方法时，传递进去的key相等（equals比较为true），则这几个线程应互斥排队输出结果，即当有两个线程的key都是"1"时，它们中的一个要比另外其他线程晚1秒输出结果，如下所示：

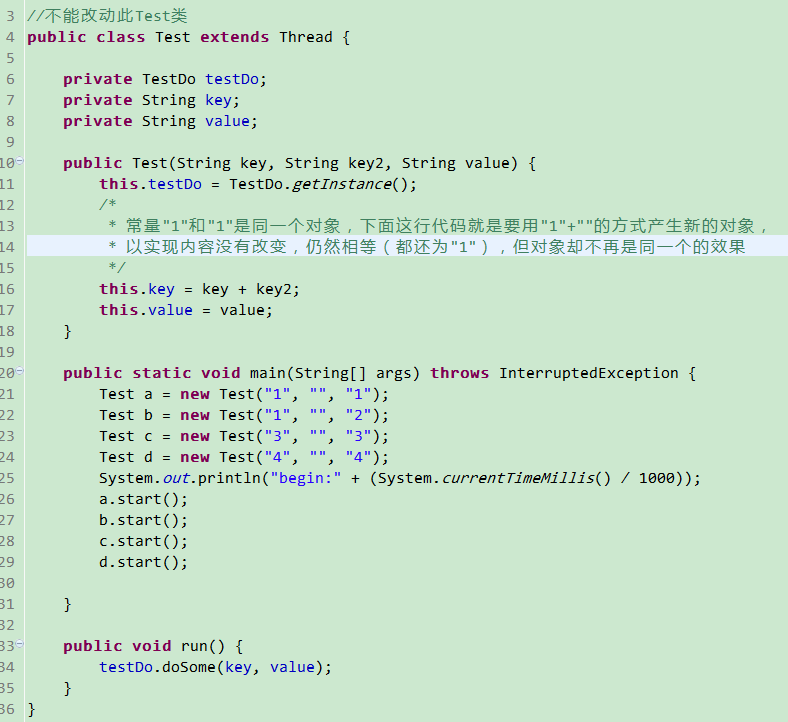
4:4:1258199615

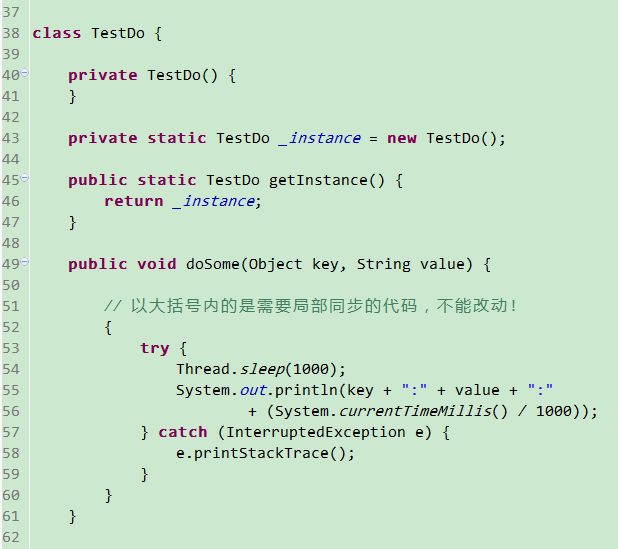
1:1:1258199615

3:3:1258199615

1:2:1258199616

总之，当每个线程中指定的key相等时，这些相等key的线程应每隔一秒依次输出时间值（要用互斥），如果key不同，则并行执行（相互之间不互斥）。原始代码如下：





* 1. 解答(采用同步集合以及互斥)

