**《Java程序设计》**

**作业报告**

**作业名称： 作业1 限时手写代码编程**

**授课教师： 毛斐巧**

**报告人： 学号： 班级：**

**报告提交时间：**

**成绩：**

**1.作业内容与要求**：

限时手写代码编程：

(1) 编写Java程序：验证1024（含1024）以内的数是否满足“卡拉兹猜想”，若有不满足“卡拉兹猜想”的数，请打印出来。所谓“卡拉兹猜想”是指对任何一个自然数n，如果它是偶数，那么把它砍掉一半，如果它是奇数，那么把(3n+1)砍掉一半，这样一直反复砍下去，最后一定在某一步得到1。

(2) 编写Java程序：第一个线程输出数字1-13，第二个线程输出字母A-Z，输出的顺序为1AB2CD3EF...13YZ，即每1个数字（如1）后紧跟着2个字母（如AB）。要求线程间实现通信。要求采用实现Runnable接口和Thread类的构造方法的方式创建线程，而不是通过Thread类的子类的方式。在主方法中创建对象，测试效果。

要求：

1. 计时手写代码编写符合上述两道题目要求的程序。建议阅读分析各题编程要求，拟定编程思路，然后，在一空白纸上手写出程序代码，并记录手写该题代码所花时间。将手写出的代码拍照贴在该题“手写代码”位置，要求图片中的源代码清晰可见，无遗漏。**若给出的是机器输入程序而非本人手写代码，该编码部分计0分。**
2. 在任意一个支持java程序开发的集成开发环境中输入自己的“手写代码”（与“手写代码”完全一致，不能修改），截图并在“机器输入程序代码”位置给出在集成开发环境中输入的“机器输入程序”，并运行程序给出程序的实际运行结果。“机器输入程序”必须与“手写代码”完全一致，并据实给出编译运行结果。比如，可能无法运行，有错误提示或警告等，应据实截图给出实际运行情况。
3. 修改程序给出程序正确运行结果。根据错误提示，直接在开发环境中修改程序，直到程序能够正确运行，给出正确运行结果截图。要求在“**程序修改与正确运行结果**”位置指示有哪些错误，进行了哪些修改，并贴出正确运行的结果截图。

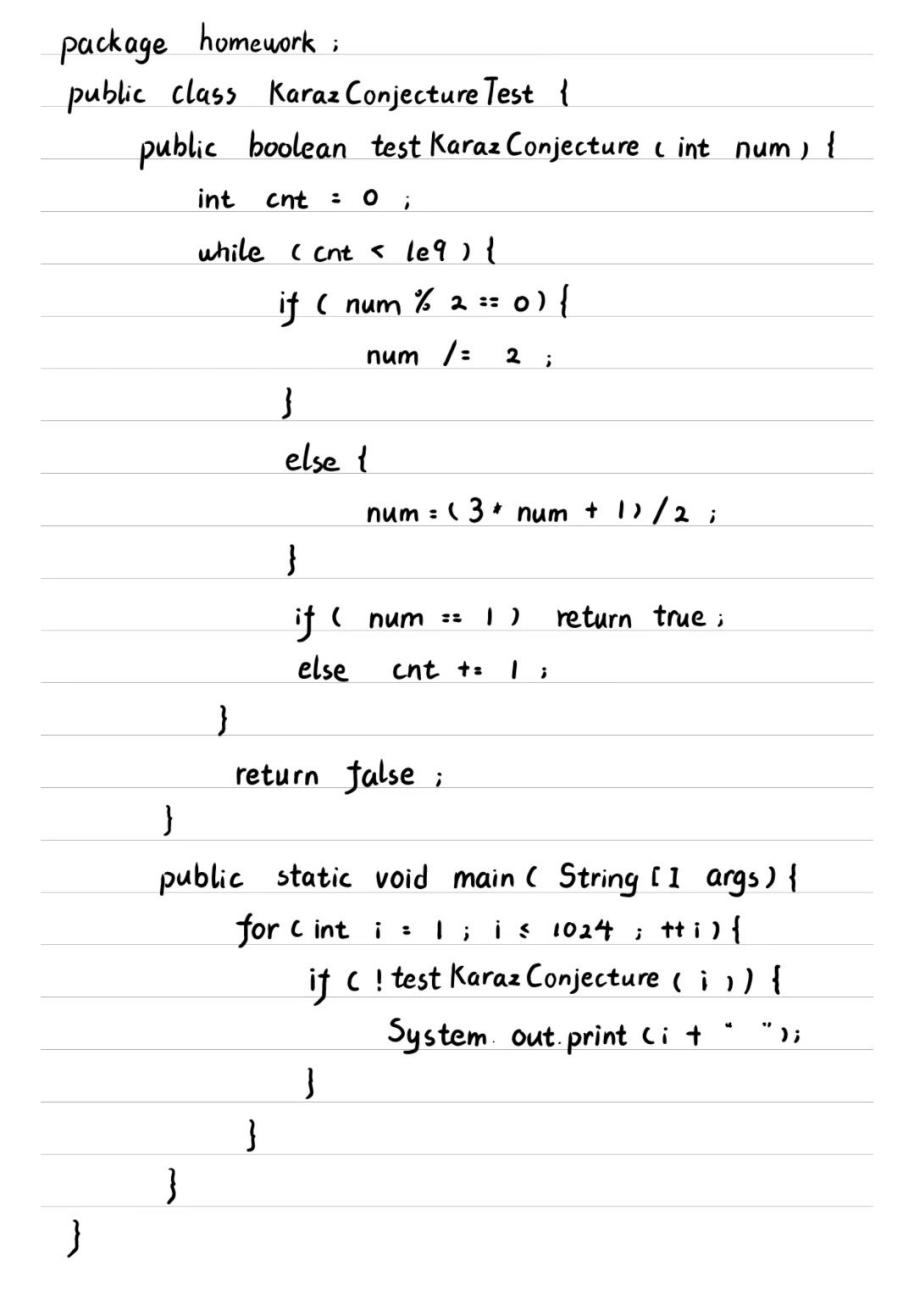
**2.解答报告正文**

**（1）题1编程实现如下：**

**（1-1）编程思路：**

通过遍历1到1024之间的所有自然数 num，对于每个 num，根据卡拉兹猜想的规则进行操作。如果 num 是偶数，则将其除以2；如果 num 是奇数，则将其乘以3加1再除以2。这个过程会一直重复，直到 num 的值变成1。如果某个 num 在规定的步骤内未能得到1，则认为它不满足卡拉兹猜想，打印该数，并继续检查下一个数。

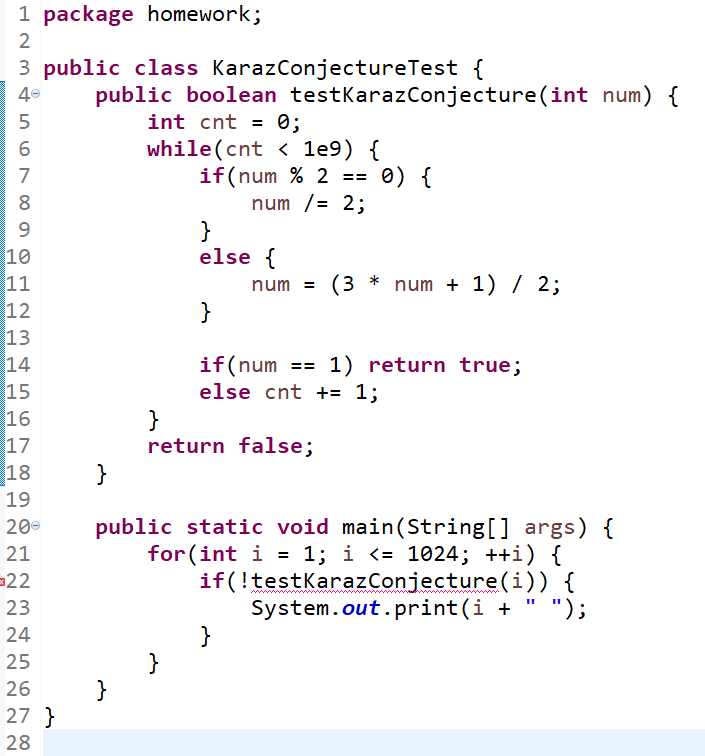
**（1-2）手写代码：**

****

**（1-3）手写代码完成时间：**

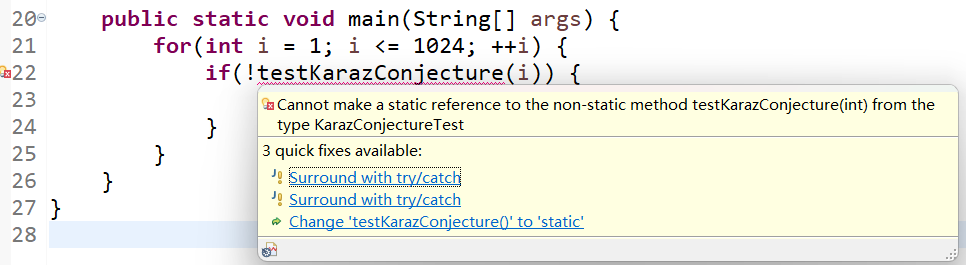
13分钟12秒。

**（1-4）机器输入程序代码：**



**（1-5）手写代码的编译运行情况：**

无法编译，会出现如下报错。这是因为在Java中，testKarazConjecture 方法是实例方法（非静态方法），而在 main 方法中尝试直接调用该方法，这是不允许的。main 方法是静态方法，不能直接访问实例方法，因为静态方法不属于任何特定的对象实例，因此不能引用非静态的方法。

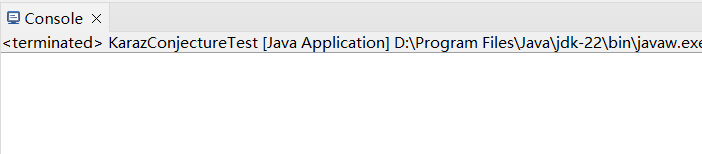


**（1-6）程序修改与正确运行结果：**

将testKarazConjecture改为静态方法，修改后的代码如下图所示。



代码运行结果如下图所示，无输出，说明1024（含1024）以内的数都满足“卡拉兹猜想”。

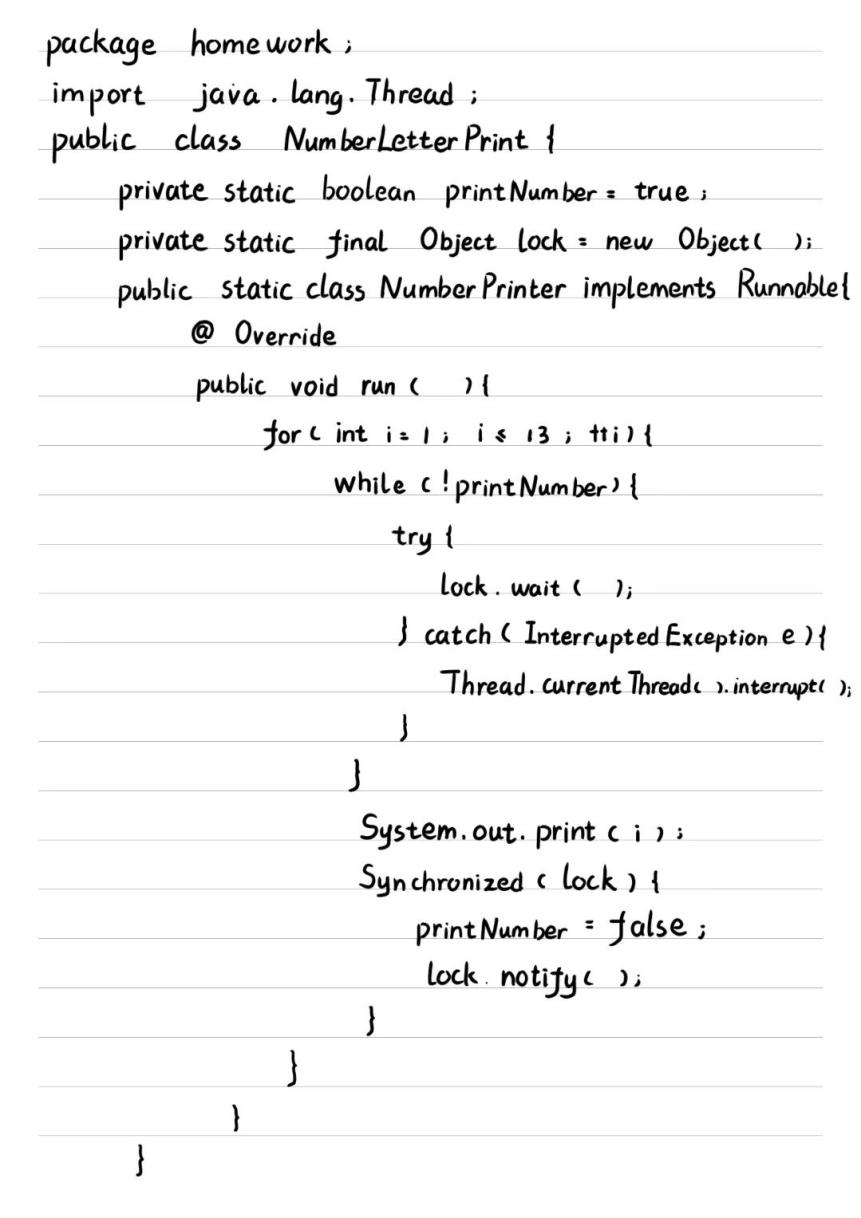


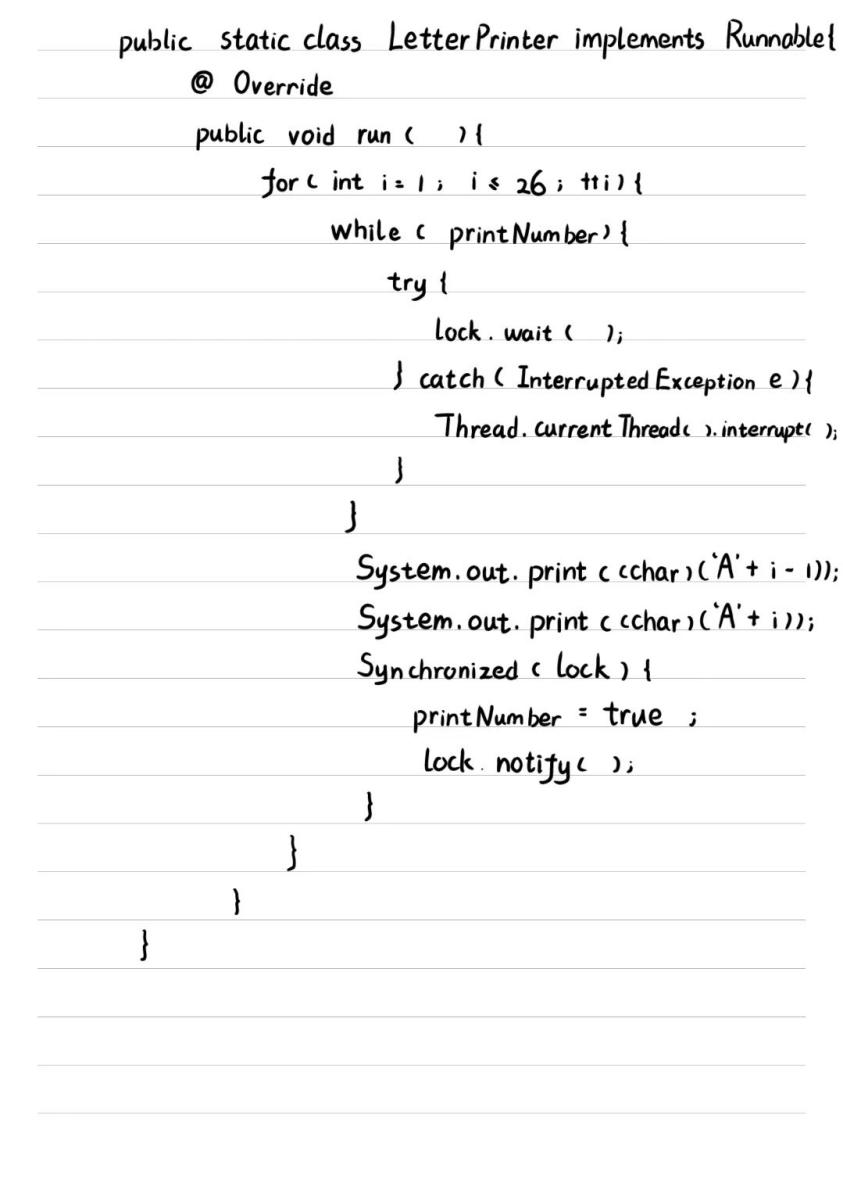
**（2）题2编程实现如下：**

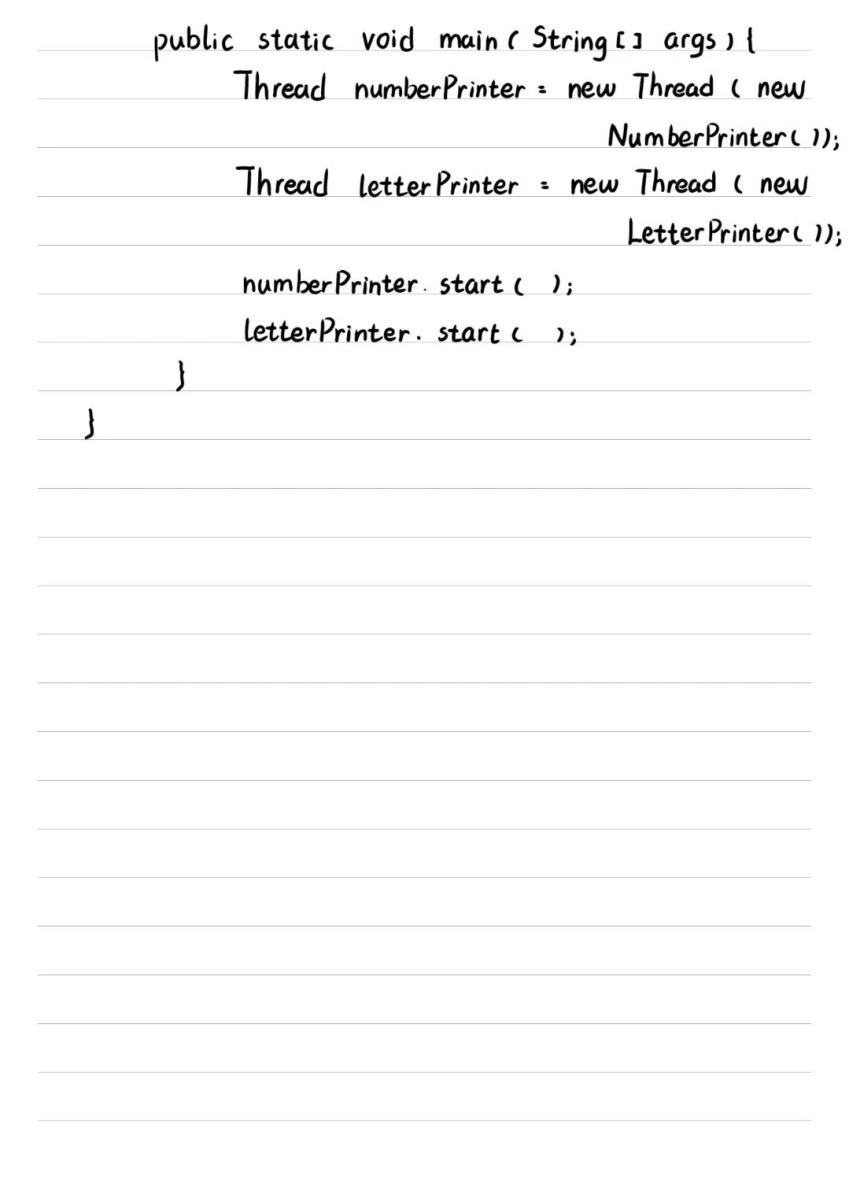
**（2-1）编程思路：**

实现两个线程，数字线程负责输出数字1到13，每次输出一个数字；字母线程负责输出字母 A 到 Z ，每次输出两个字母。通过使用 wait()和 notify() 方法，确保线程之间按照正确的顺序交替执行，数字线程输出一个数字后通知字母线程输出字母，字母线程输出两个字母后再通知数字线程输出下一个数字。最后通过实现 Runnable 接口，创建线程并通过 Thread 类的构造方法启动，确保两个线程交替输出数字和字母。

**（2-2）手写代码：**

****

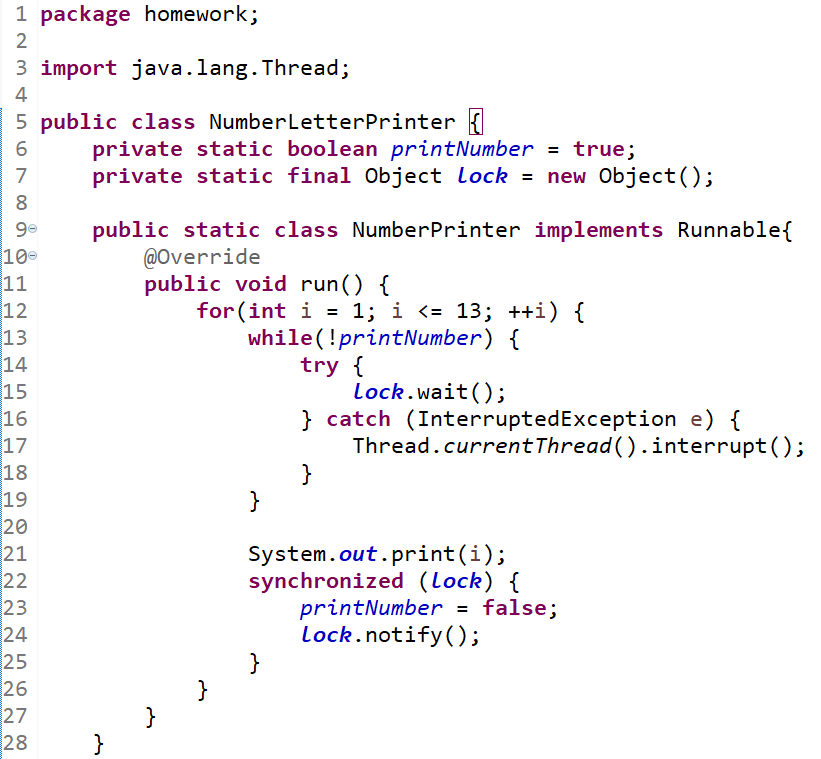
****

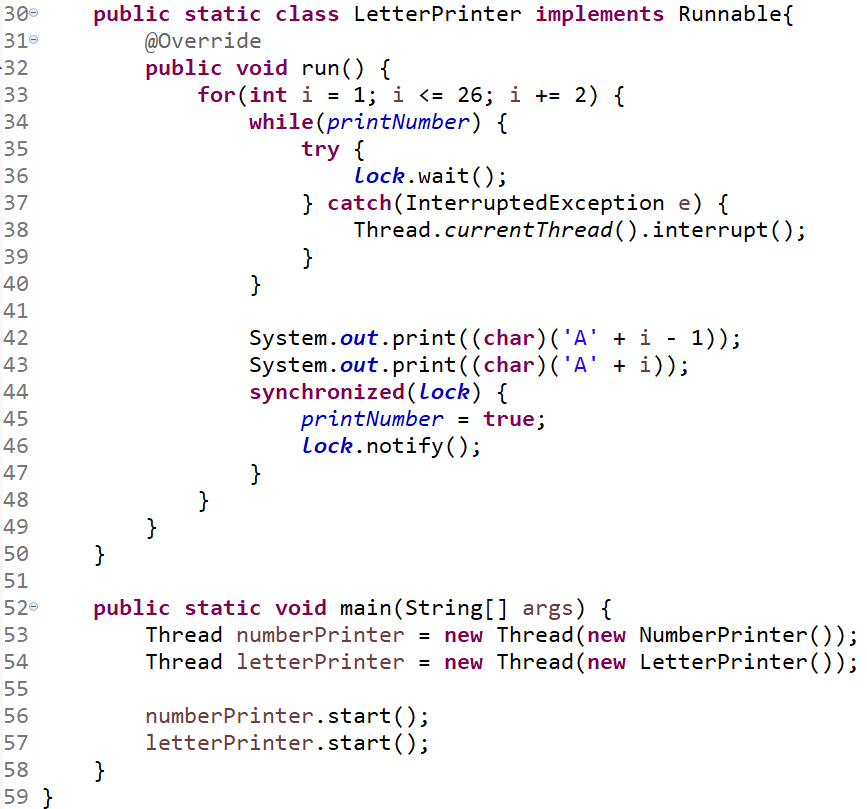
****

**（2-3）手写代码完成时间：**

25分钟34秒。

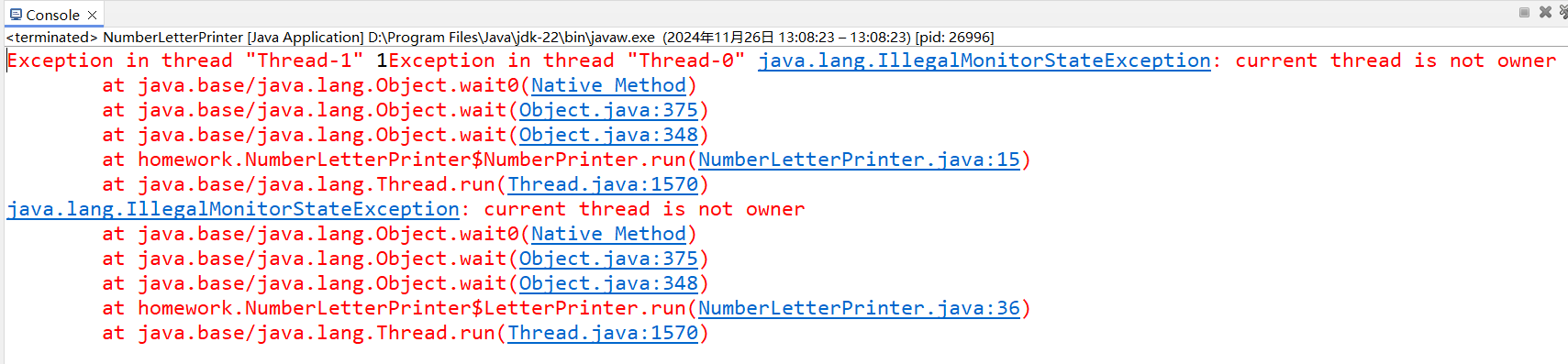
**（2-4）机器输入程序代码：**





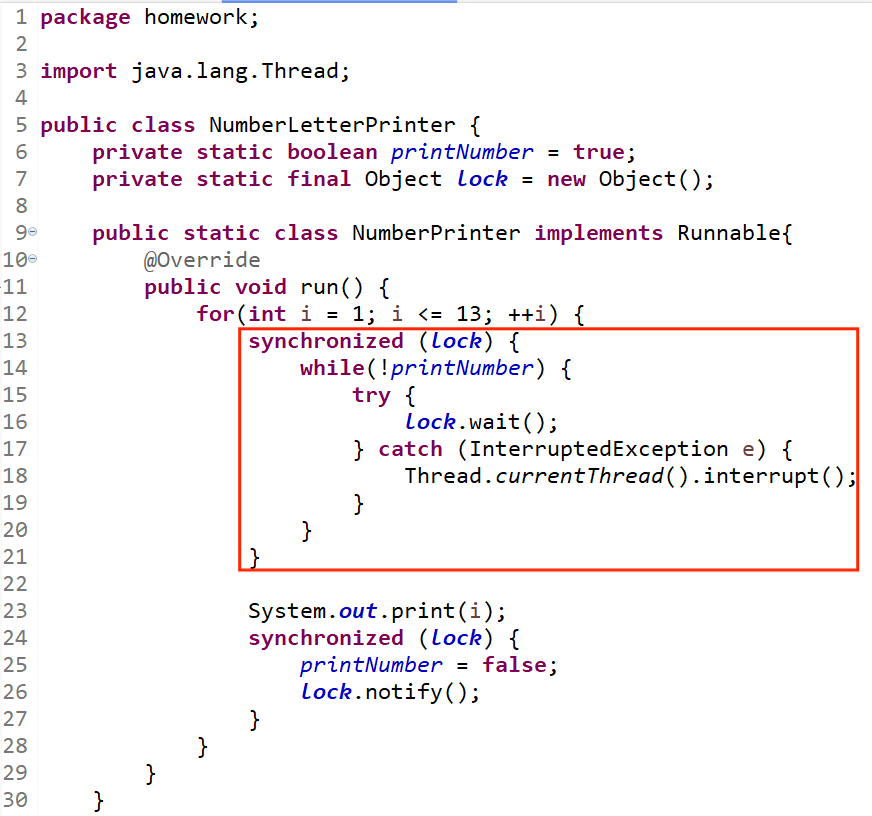
**（2-5）手写代码的编译运行情况：**

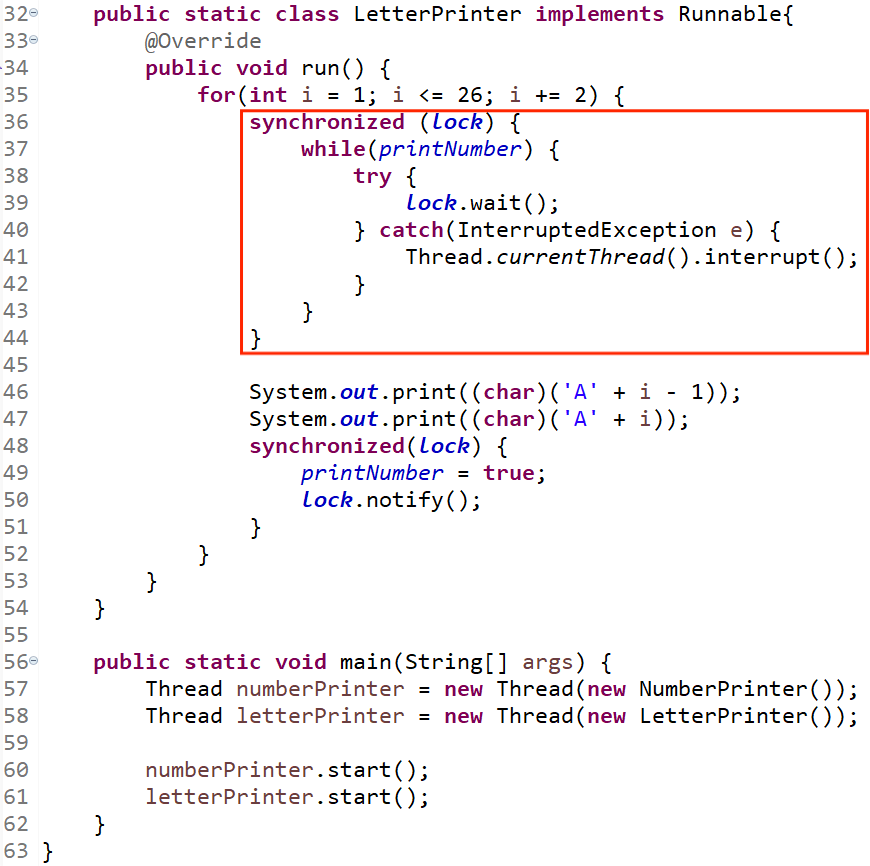
代码运行时出现如下报错。报错 java.lang.IllegalMonitorStateException: current thread is not owner 发生的原因是，线程在调用 wait() 时没有正确持有锁。在 Java 中，调用 lock.wait() 和 lock.notify() 方法的线程必须先获取到该对象的锁，而在 NumberPrinter 和 LetterPrinter 类中的 while 循环中直接进行了 lock.wait() 调用，但没有进行锁的获取。正确的做法应该是先通过 synchronized 块来确保线程持有锁，然后再进行 lock.wait() 或 lock.notify() 操作。换句话说，lock.wait() 和 lock.notify() 必须在 synchronized 块内进行，才能保证当前线程已经拥有该锁。



**（2-6）程序修改与正确运行结果：**

将 while 循环中的 lock.wait() 调用放入 synchronized (lock) 块中，确保在调用 lock.wait()时线程已经持有 lock 锁，修改后的代码如下图所示。





代码运行结果如下图所示。可以看到，程序正确输出了题干要求的 “1AB2CD3EF...13YZ”。

