**《Java程序设计》**

**作业报告**

**作业名称：作业2 限时手写代码编程**

**授课教师： 姚俊梅**

**报告人： 杜良衡 学号： 2022150255 班级： 01B**

**报告提交时间： 2024.11.24**

**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：**

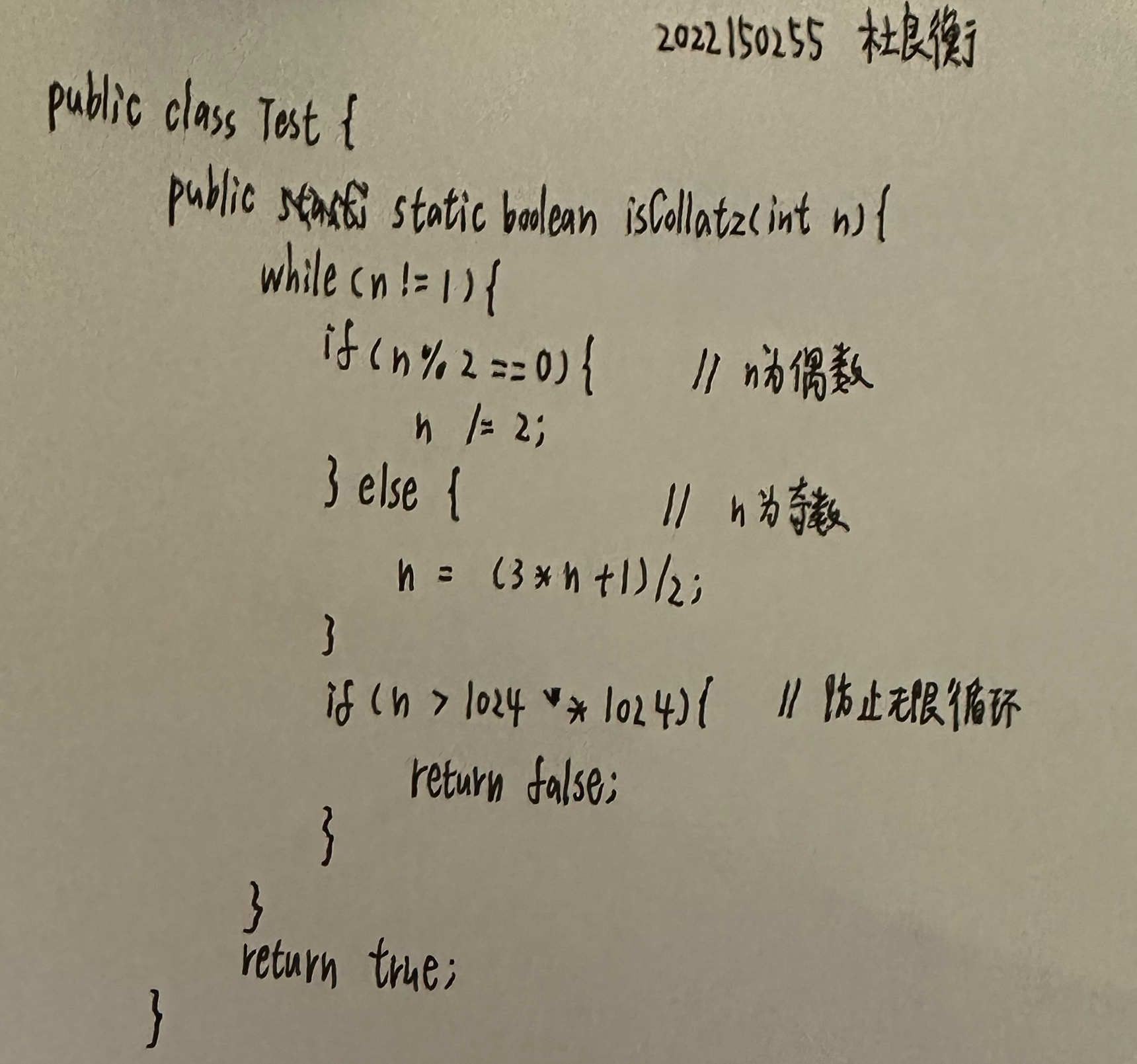
为验证1024（含1024）以内的数是否满足“卡拉兹猜想”，需要遍历数字1到1024，对每个数字，都调用验证“卡拉兹猜想”的方法。

该方法入参为一个自然数n，通过while循环不断改变n的值，直到结果为1，返回true。循环内通过n除2的余数来判断奇偶性：若余数为0，则n为偶数，将n除以2；若余数为1，则n为奇数，将n乘以3加1后再除以2。为防止无限循环，假定n超过1024\*1024则不满足猜想，返回false。

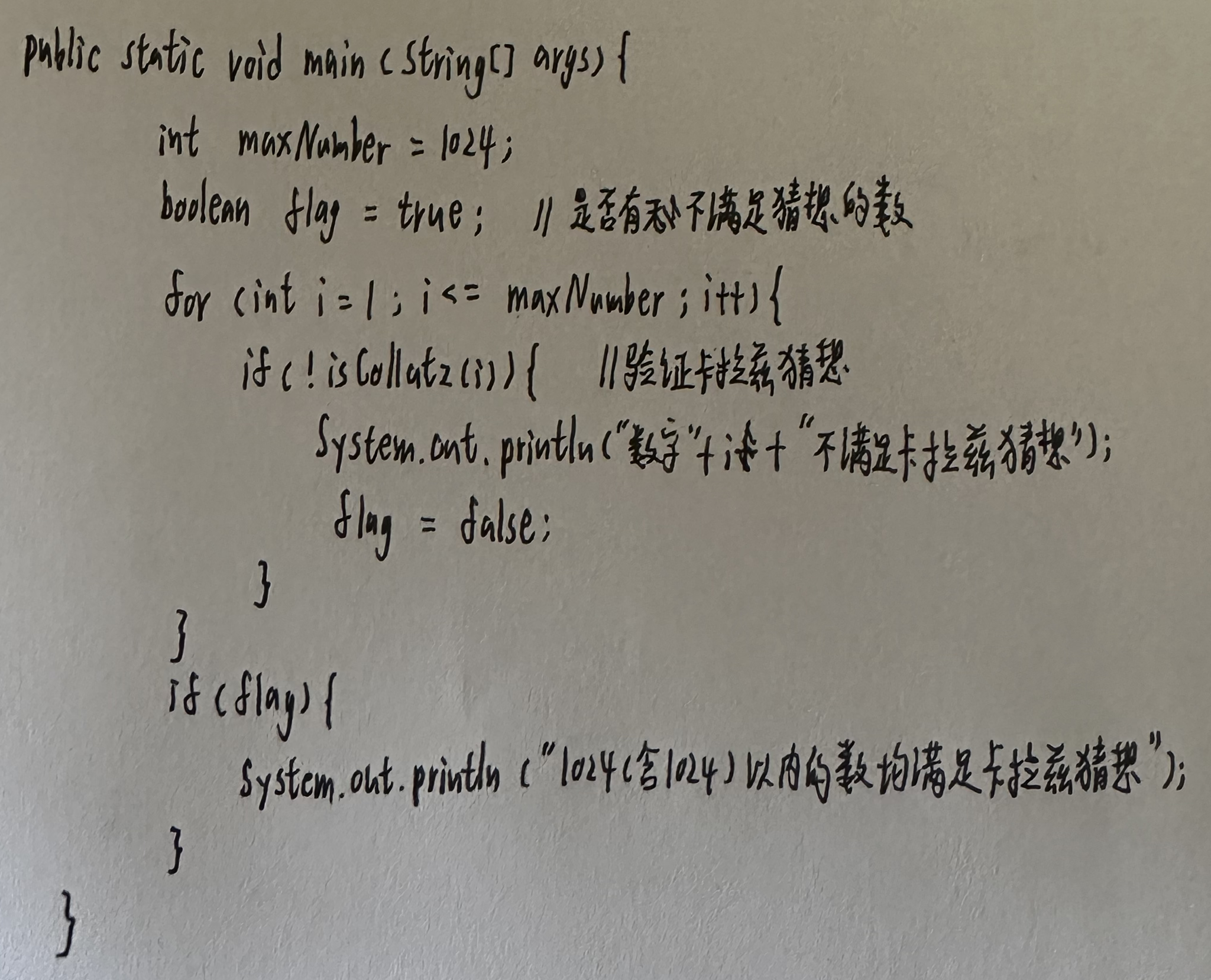
主函数的遍历中会判断验证方法的值是否为false，是的话说明这个数不满足“卡拉兹猜想”，会将其打印出来。

**（1-2）手写代码：**

验证“卡拉兹猜想”的isCollatz方法：



主方法：

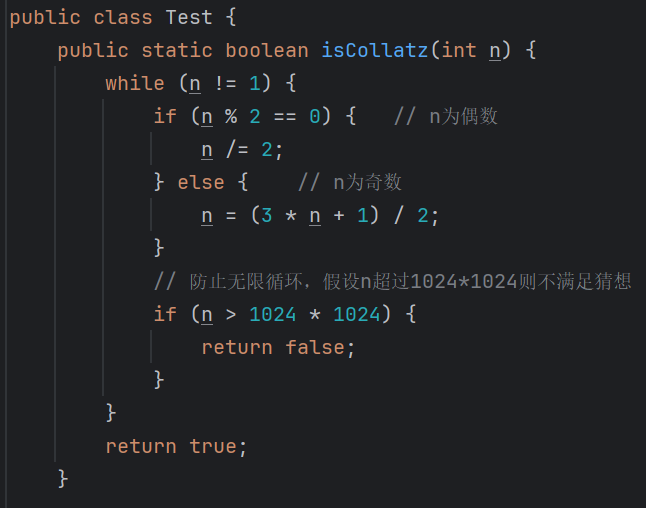


**（1-3）手写代码完成时间：**

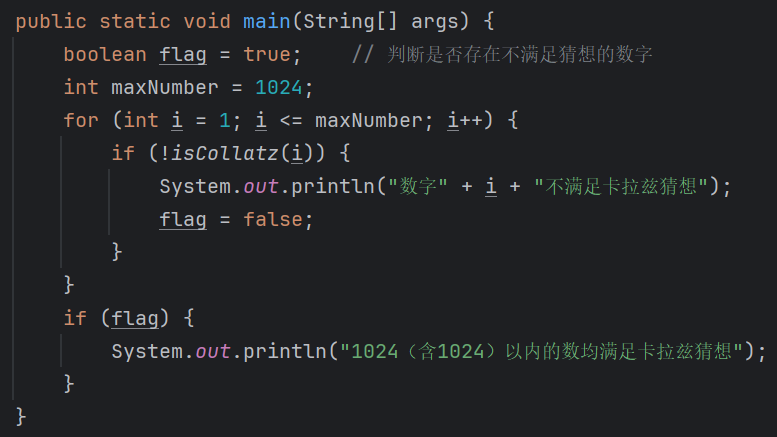
15分钟

**（1-4）机器输入程序代码：**

验证“卡拉兹猜想”的isCollatz方法：

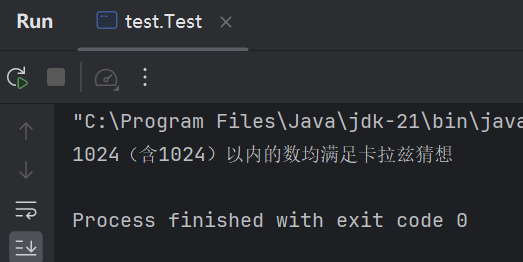
****

主方法：



**（1-5）手写代码的编译运行情况：**

代码输出结果符合预期。

****

**（1-6）程序修改与正确运行结果：**

程序成功运行并输出正确结果，无需进行修改。

**（2）题2编程实现如下：**

**（2-1）编程思路：**

首先为实现线程的交替执行，需要创建一个共享的锁对象lock来同步两个线程的访问。还要创建printNumber标志来控制当前打印数字还是字母。

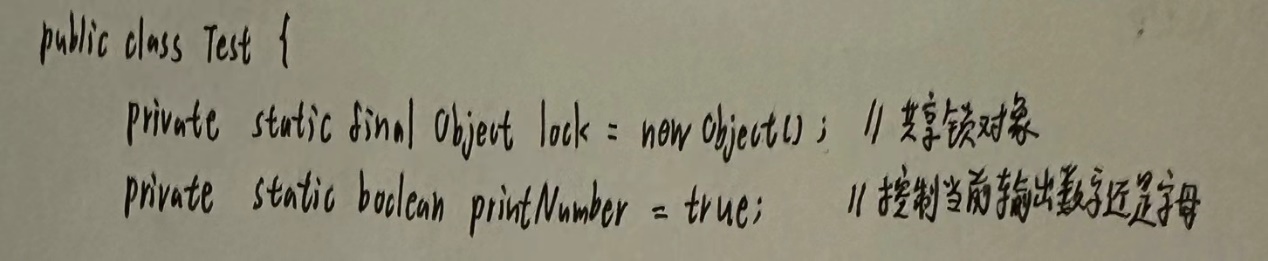
接着创建数字线程和字母线程，通过实现Runnable接口的方法，这两个类中会重写run方法。在重写的run方法内，最外层使用while循环检查是否还有数字或字母需要打印。

线程内利用synchronized块来获取锁，确保同一时间只有一个线程可以执行输出代码块。在输出代码块内会先检查printNumber标志，如果当前不是该线程输出的轮次，则会调用wait方法让出锁；是的话则会输出数字或字母，改变printNumber标志，并调用notifyAll来通知另外一个等待着的打印线程。

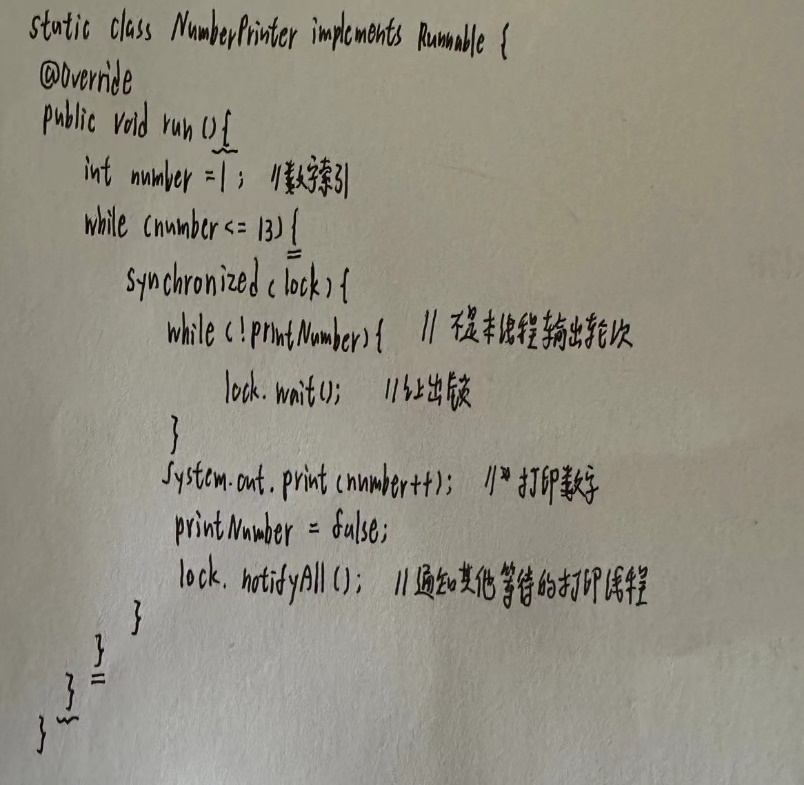
主方法会创建两个Thread对象，并传入实现了Runnable接口的两个类来启动数字和字母打印线程。

**（2-2）手写代码：**

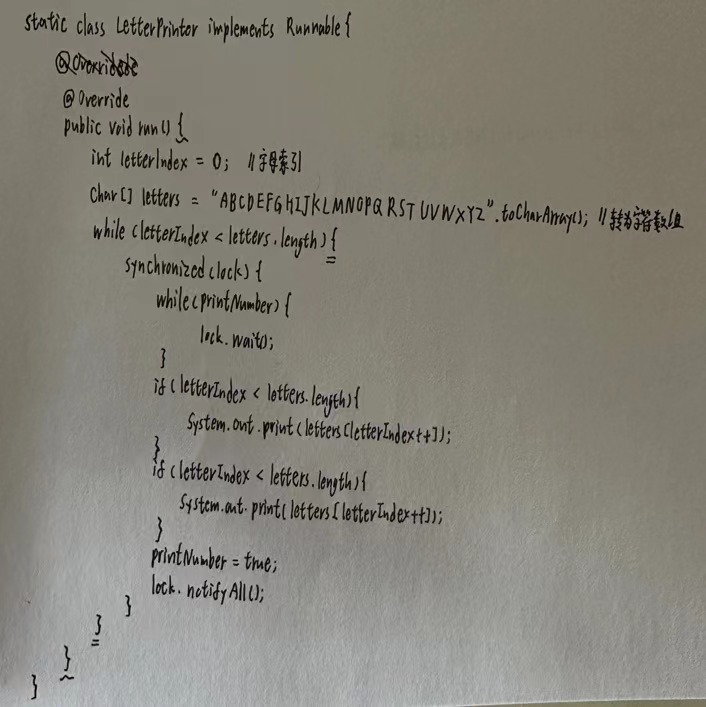
全局变量定义：

****

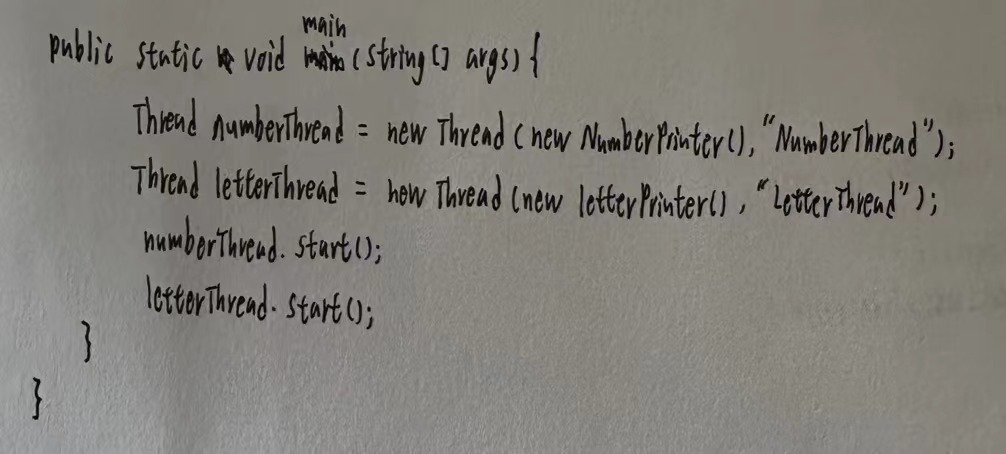
数字打印线程：



字母打印线程：



主方法：

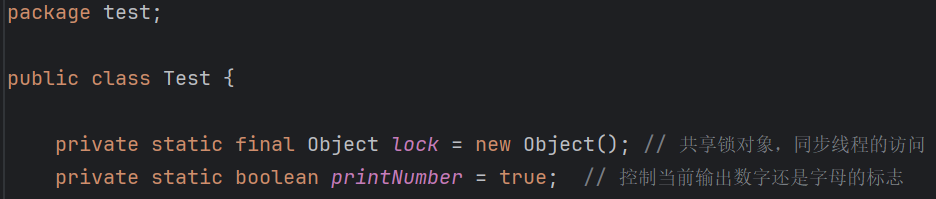


**（2-3）手写代码完成时间：**

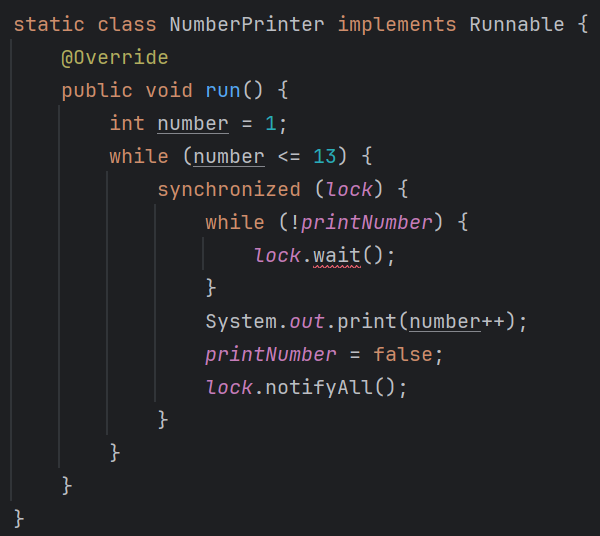
26分钟

**（2-4）机器输入程序代码：**

全局变量定义：



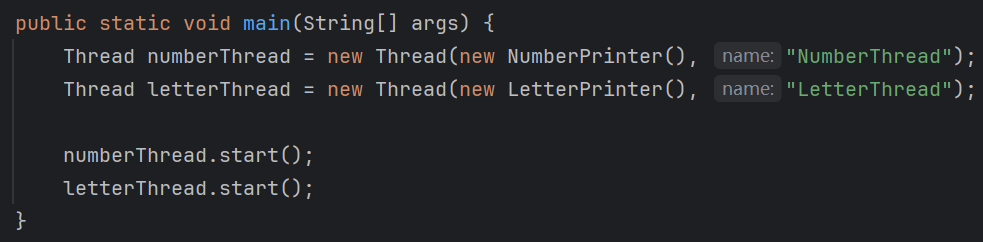
数字打印线程：



字母打印线程：

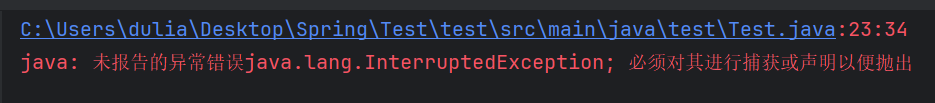


主方法：

****

**（2-5）手写代码的编译运行情况：**

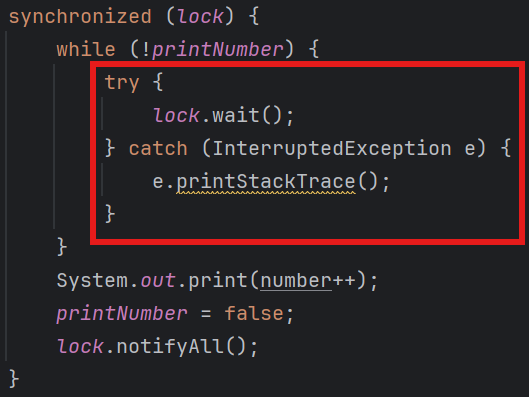
编译该程序时报错，显示没有处理InterruptedException异常。

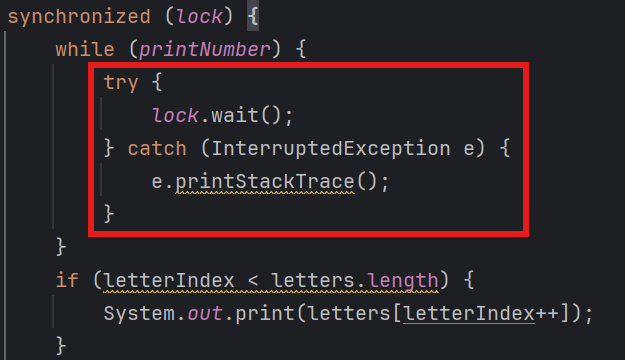
****

**（2-6）程序修改与正确运行结果：**

InterruptedException表示一个正在执行的线程被被另一个线程中断。当一个线程正在执行wait等方法时，如果另一个线程调用了interrupt方法，就会抛出该异常。

此处的解决方法是将两个打印线程中调用wait方法的部分用try-catch来捕获异常，捕获后可以再调用printStackTrace来打印异常的跟踪信息。

****

****

修改后程序成功运行并打印正确结果。

