****

****

**信息学院软件工程系**

**《JAVA程序设计》实验报告**

实验八

**姓名：任宇**

**学号：33920212204567**

**学院：信息学院**

**专业：软件工程**

**完成时间：2024-04-20**

1. **实验目的及要求**

* 熟悉集合类

1. **实验题目及实现过程**
2. 题目一：

编写一个程序，读取一个名字序列，将他们保存在一个set中，然后删除重复的名字。应允许用户输入名字。

1. 实验环境

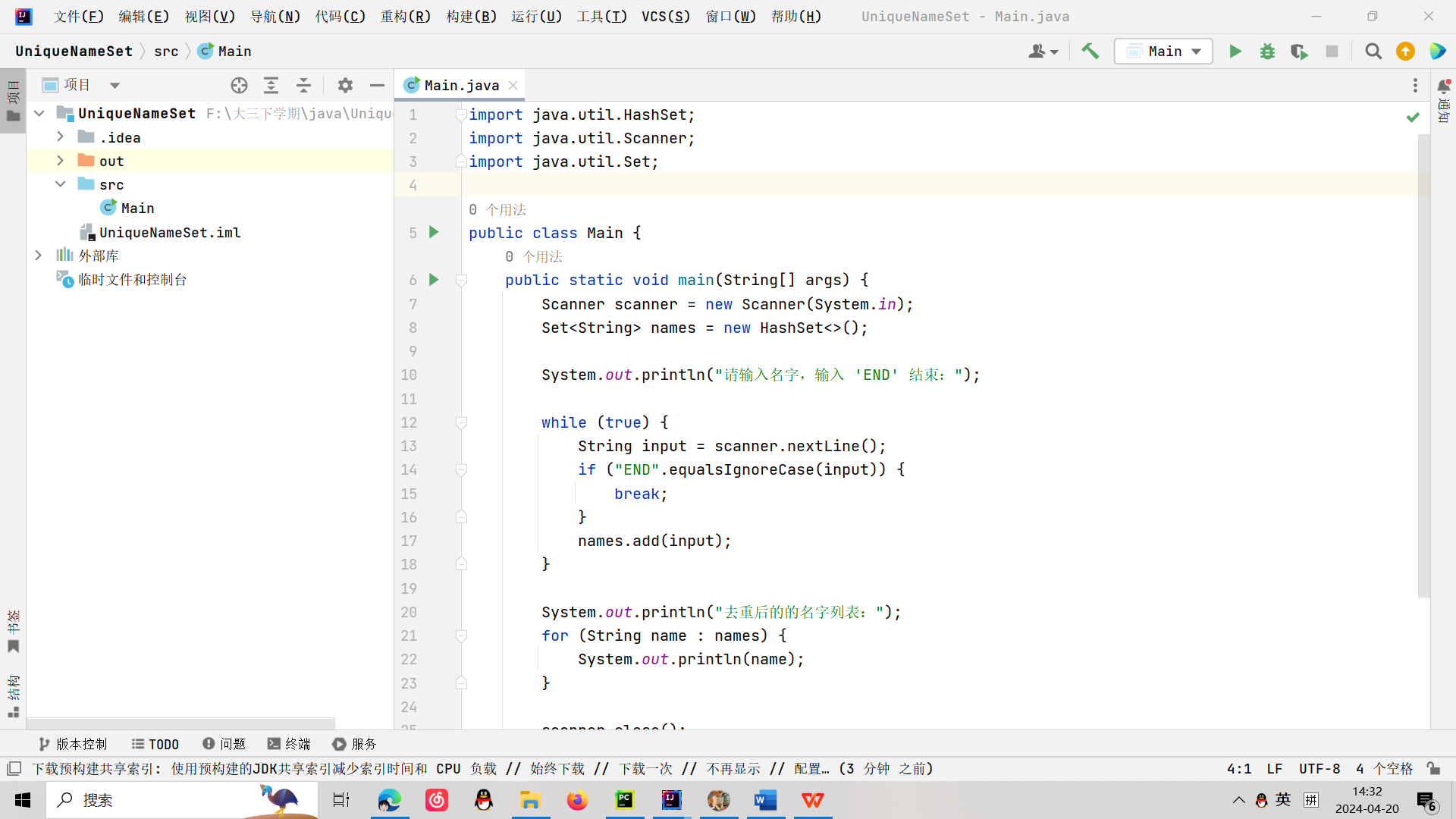
集成开发环境：IDEA Community Edition 2022.3.2

JDK版本：JDK17

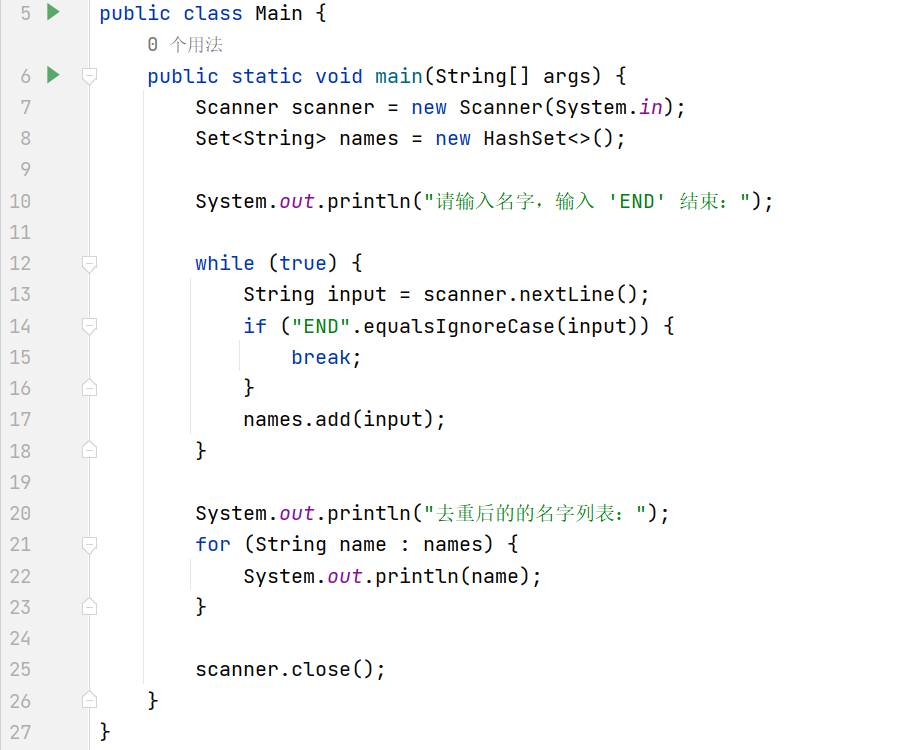
1. 实现过程

对于本题，只设计一个Main类即可完成题目要求。在main方法内部，通过Scanner对象从标准输入读取字符串。每读取一个有效的名字（不是"END"），就将其添加到HashSet中。当读取到"END"时，结束输入，关闭Scanner对象，并遍历HashSet打印所有存储的名字。HashSet<String>用于存储名字。HashSet是一种集合，它不允许重复元素，这自动帮助我们处理了去重的需求。用户输入的每个名字都尝试添加到HashSet中，但由于HashSet的特性，即使用户输入重复的名字，它也只会被保存一次。

1. 过程截图
2. 全屏截图：



1. 主要代码：



1. 运行结果：



1. 题目二：

编写一个程序，它确定并输出用户输入的某个句子中重复的单词数。不区分字母的大小写。忽略标点符号。

1. 实验环境

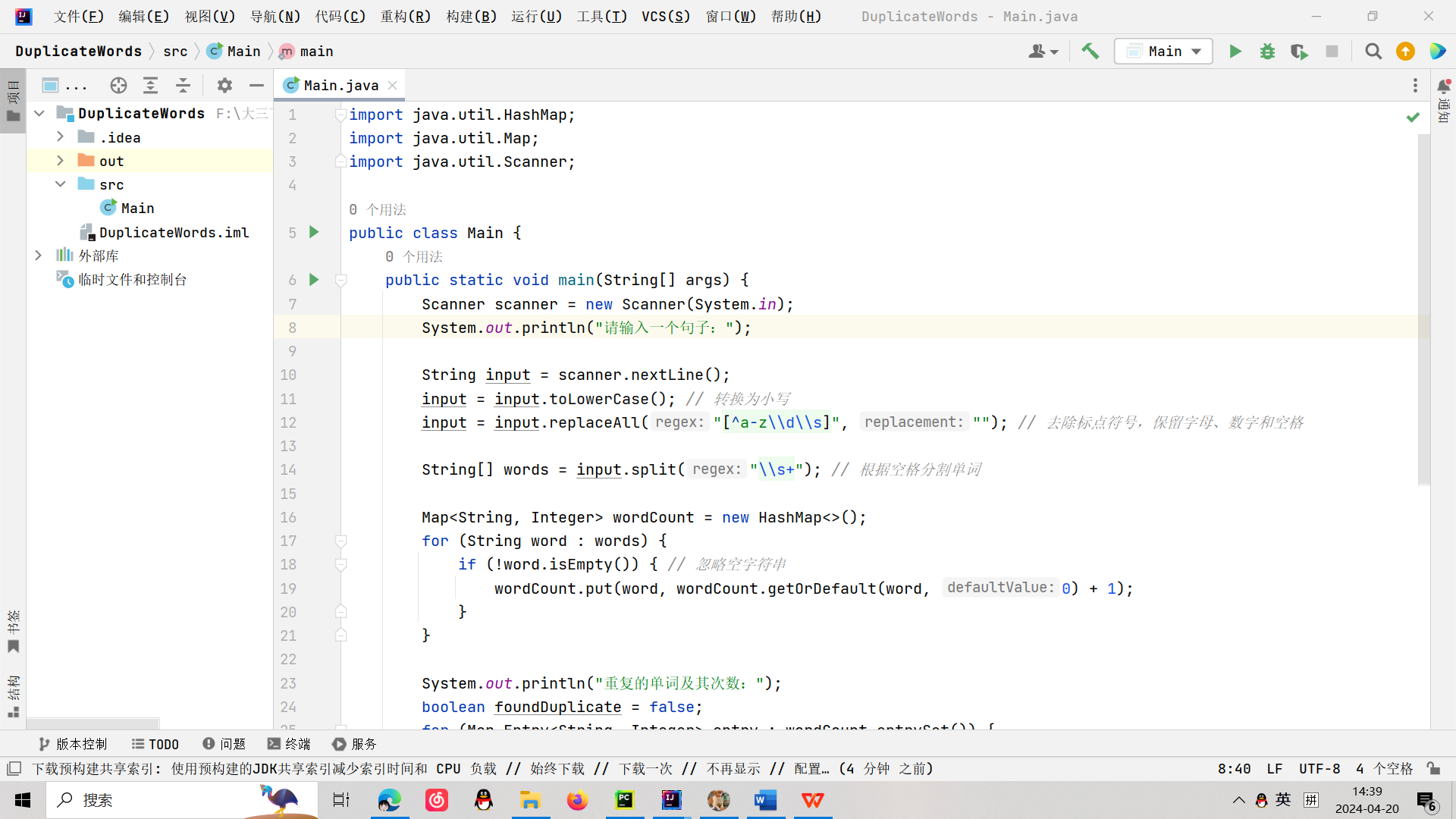
集成开发环境：IDEA Community Edition 2022.3.2

JDK版本：JDK17

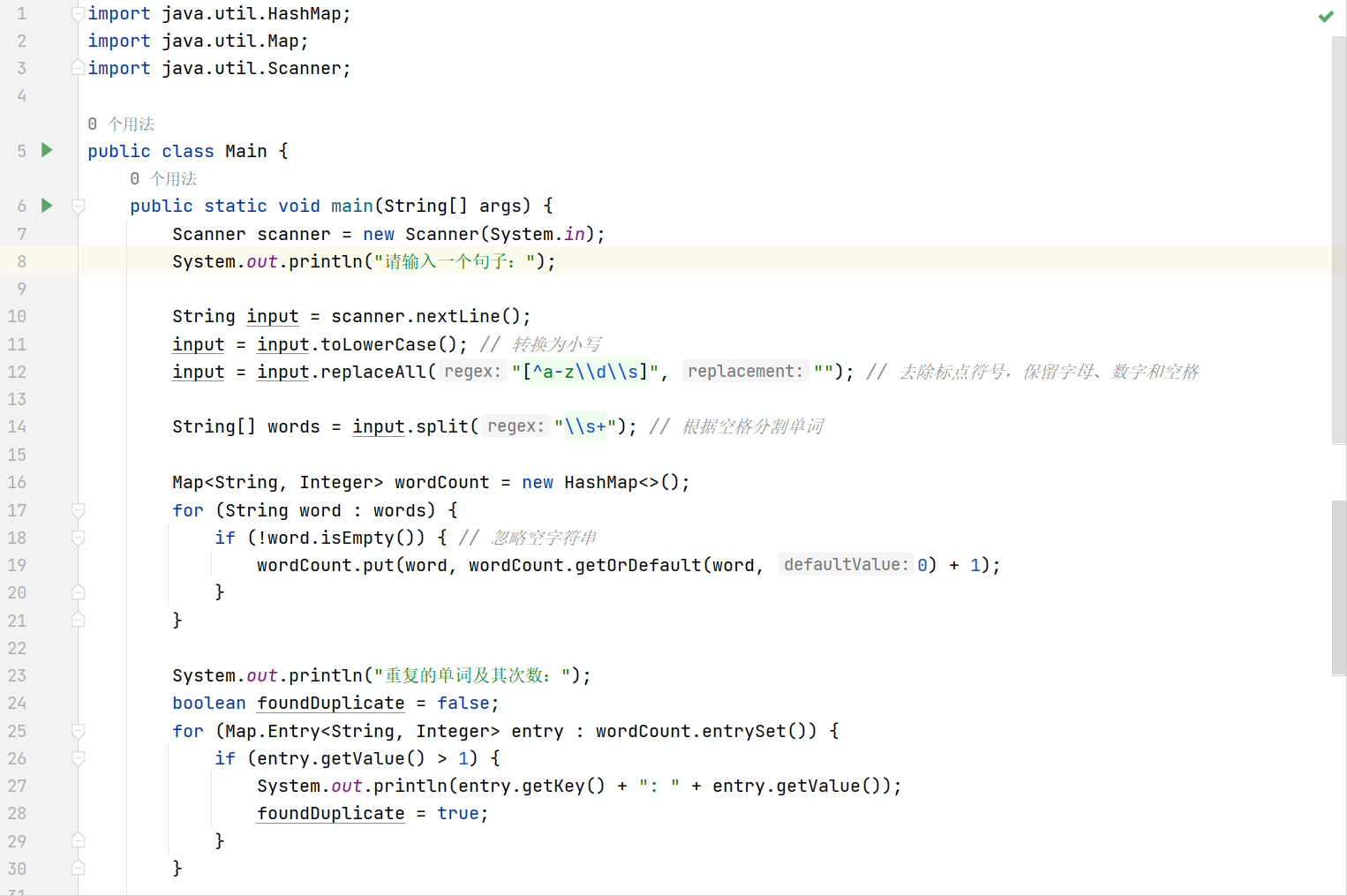
1. 实现过程

程序只包含一个Main类，用于确定并输出用户输入的句子中重复的单词数，忽略字母的大小写并忽略标点符号。首先使用Scanner类获取用户的输入句子。接着将输入句子转换为小写，然后使用正则表达式去除标点符号，只保留字母、数字和空格，并且使用空格分割句子，得到单词数组。再接着使用HashMap<String, Integer>来存储单词及其出现次数。键为单词，值为该单词出现的次数。遍历单词数组，将单词作为键存入HashMap中，如果已存在则增加其出现次数。遍历HashMap，找到出现次数大于1的单词，打印出这些单词及其出现次数。如果没有重复单词，则输出相应的提示信息。

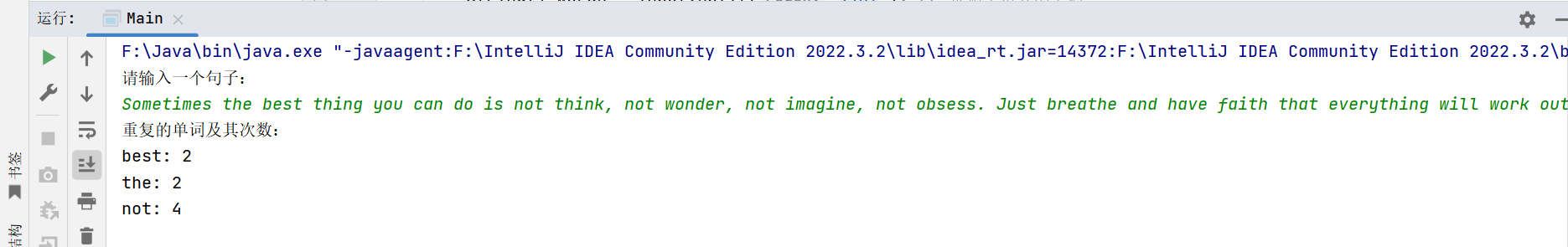
1. 过程截图
2. 全屏截图：



1. 主要代码：



1. 运行结果：



1. 题目三：

修改wordTypeCount.java，使其计算每个字母的出现次数而不是每个单词的出现次数。

1. 实验环境

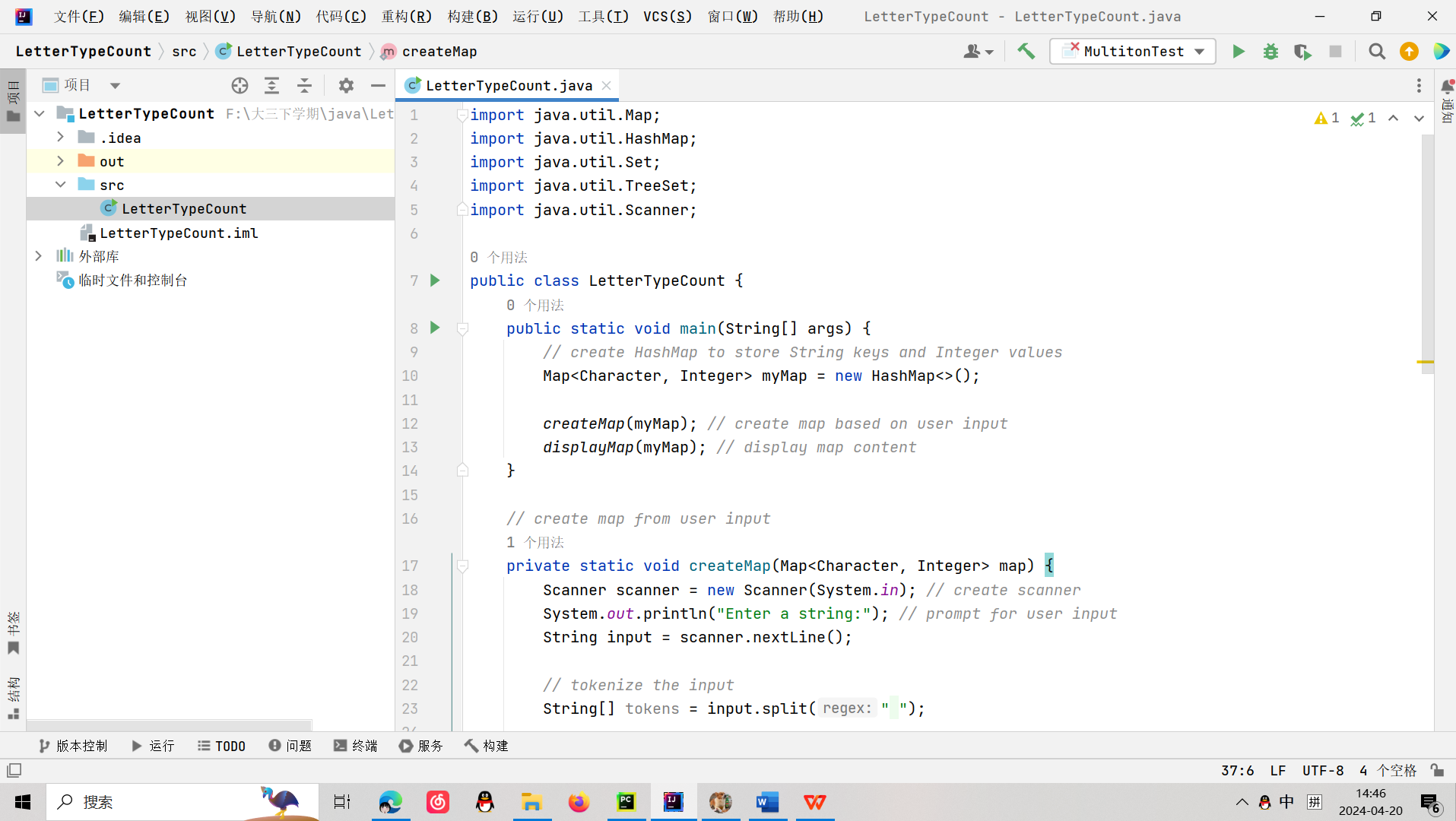
集成开发环境：IDEA Community Edition 2022.3.2

JDK版本：JDK17

1. 实现过程

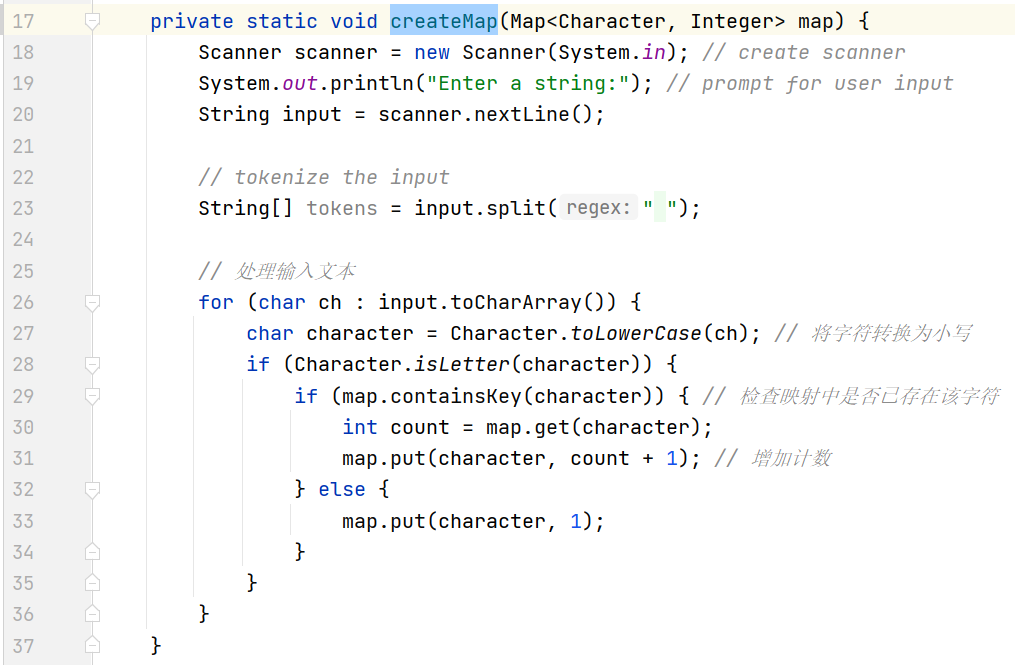
首先将类名从WordTypeCount更改为LetterTypeCount以更准确地反映程序的功能。接着将存储结构从Map<String, Integer>更改为Map<Character, Integer>，以存储单个字母及其出现次数。原本的代码按空格分割输入成单词并处理。新的代码使用input.toCharArray()直接将输入字符串转换为字符数组，然后遍历每个字符。对每个字符，首先转换为小写（Character.toLowerCase(ch)），确保统计时不区分大小写。然后使用Character.isLetter(character)检查每个字符是否为字母，这样非字母字符被自动忽略。如果字母已存在于映射中，则增加其计数；如果不存在，则将其计数设为1。最后遍历Map输出结果。

1. 过程截图
2. 全屏截图：

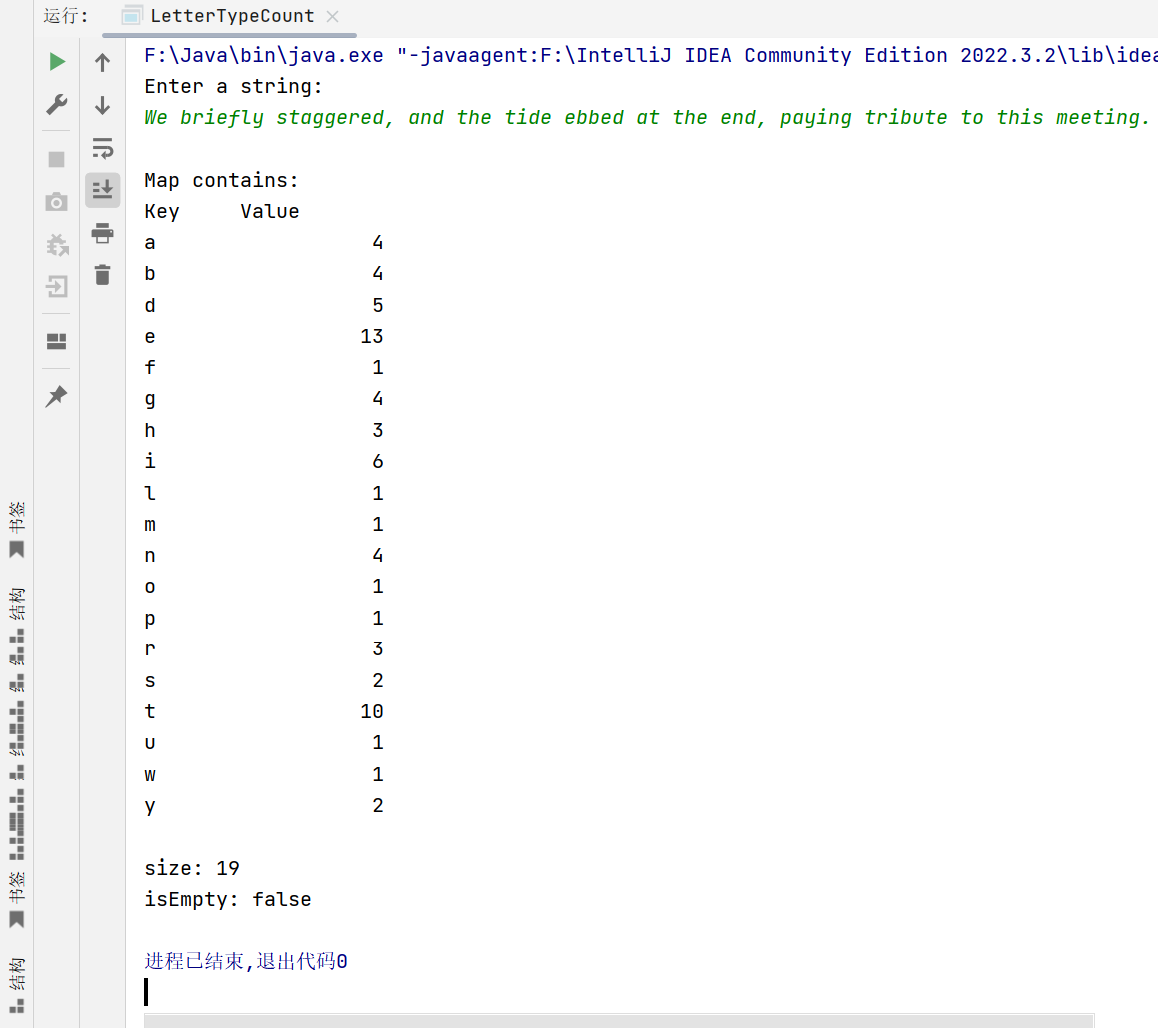


1. 主要代码：

重点修改createMap方法：



1. 运行结果：



1. 题目四：

编写一个程序，它从用户处获得一个整数输入，然后判断它是否为质数。如果不显示该数的全部质因子。质数的因子是 1 和自身。不为质数的数，都具有唯一的因子分解形式。例如54的因子为 2、3、3、3，这些值相乘的结果是 54 。因此对于 54，输出因子应为2和3。需在程序中使用Set。

1. 实验环境

集成开发环境：IDEA Community Edition 2022.3.2

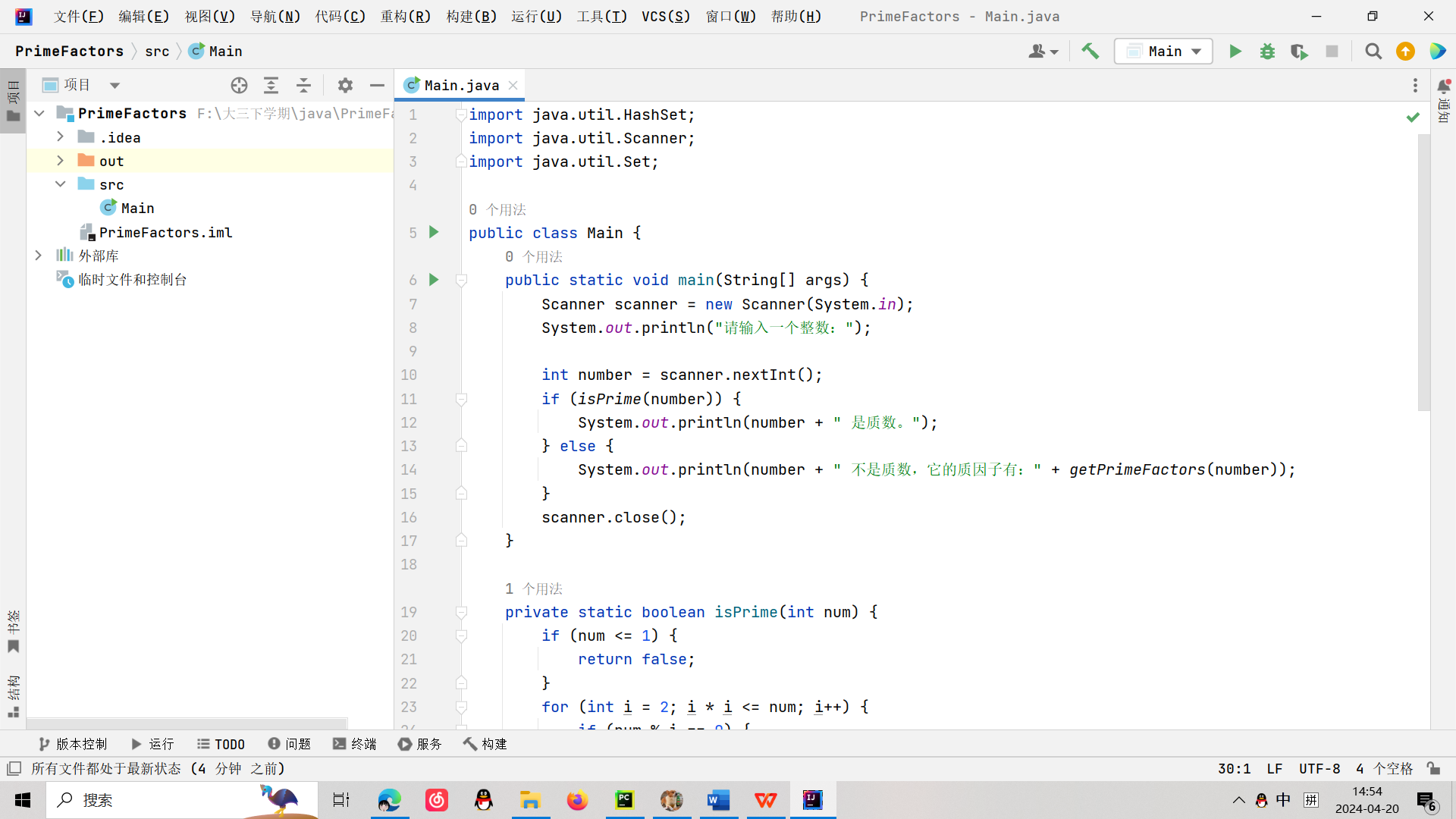
JDK版本：JDK17

1. 实现过程

在这个Java程序中，我使用HashSet来保存因子，确保输出没有重复的因子。程序包含一个主类Main和一个main方法作为程序的入口点，此外还有isPrime(int num)和getPrimeFactors(int number)两个方法。isPrime(int num)用于判断一个数是否是质数。getPrimeFactors(int number)则用于获取一个数的所有质因子并存储在一个Set中。

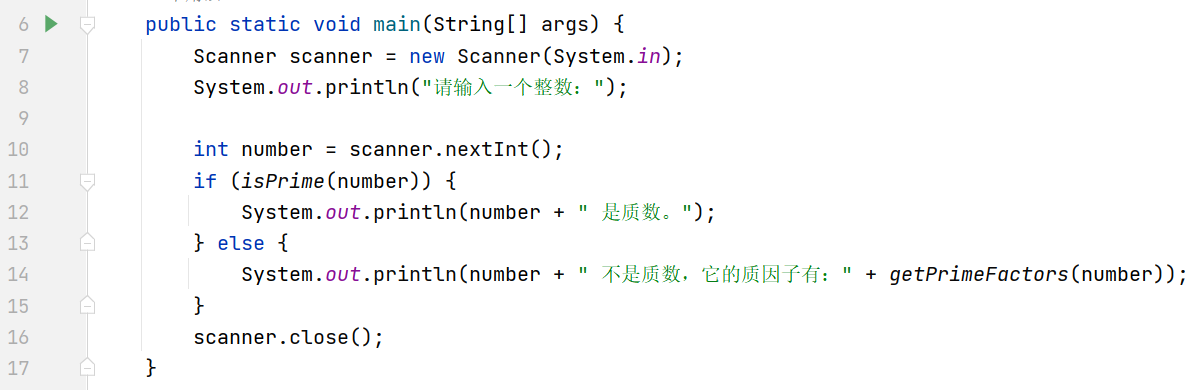
输入的整数首先通过isPrime方法判断是否为质数。如果是质数，输出相应信息。如果不是质数，调用getPrimeFactors方法获取其所有质因子，并输出。

1. 过程截图
2. 全屏截图：

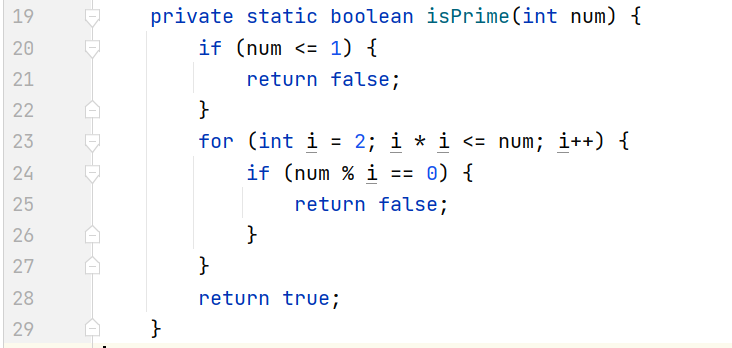


1. 主要代码：

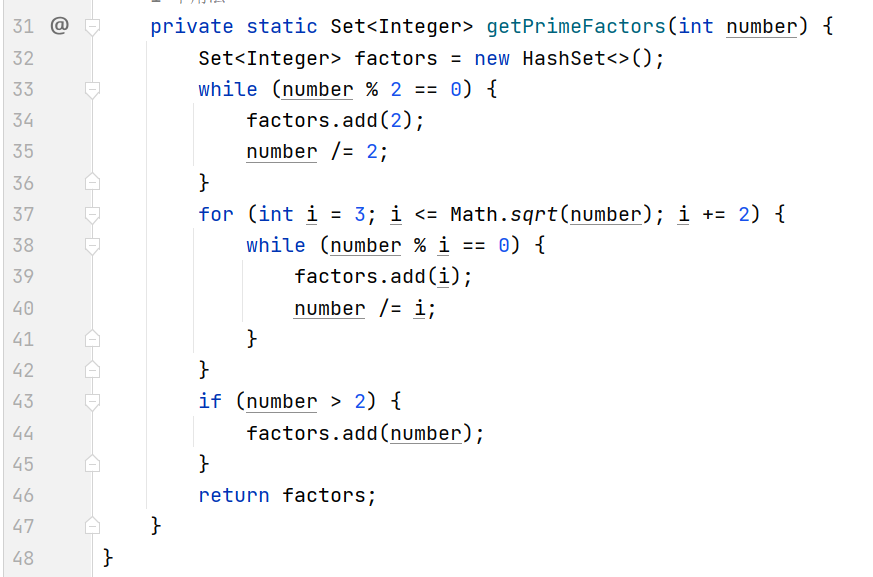
Main方法实现：



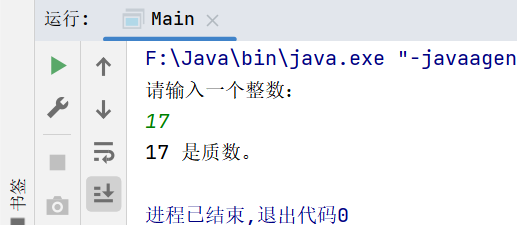
isPrime方法实现：

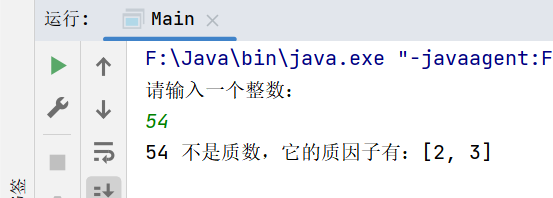


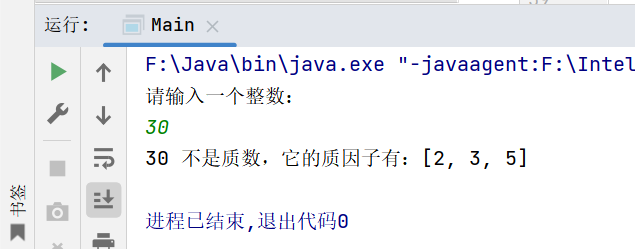
getPrimeFactors方法实现：



1. 运行结果：







**三、实验总结与心得记录**

在这次实验中，我通过具体的编程实践加深了对Java集合类的理解，并提高了自己解决实际问题的编程技能。通过使用HashSet和HashMap等集合类，我体会到了这些数据结构在实际应用中的强大功能，比如自动处理重复数据和进行高效数据检索。