**2024 / 2025学年 （一）学期 上机实验报告**

**课程名称：面向对象程序设计(Java) 姓 名：沈哲伟 学 号：2312190313**

**指导教师： 蒲 飞 班 级：计科2023级 日 期： 2024.11.15**

**【一】上机实验内容及要求**

**实验要求：掌握String对象的定义和使用，熟练使用字符串的正则表达式来提取所需信息。深入了解HashMap中键值对的概念，能使用HashMap来解决实践问题。进一步，熟练掌握文件的读写。**

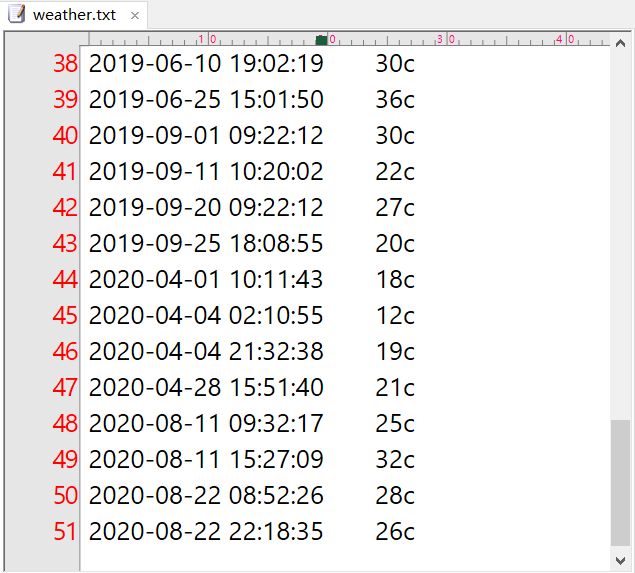
**实验平台：JDK11+IDEA**

**实验内容：**

**一、有不同年份的温度数据temperature.txt，要求：**

1. **日期按年月日升序、温度按降序将温度数据排列出来**

**temperature.txt格式如下：**

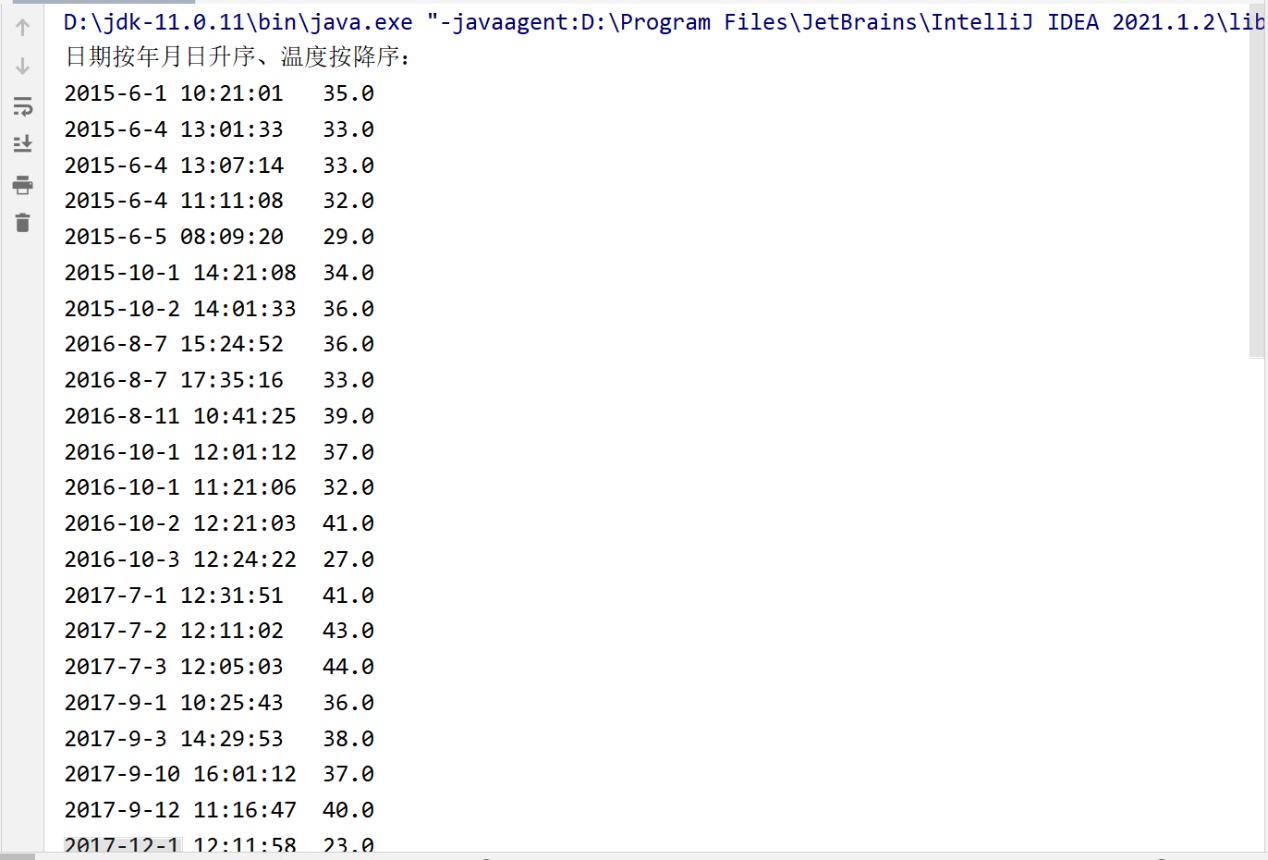


**读文件语句：**

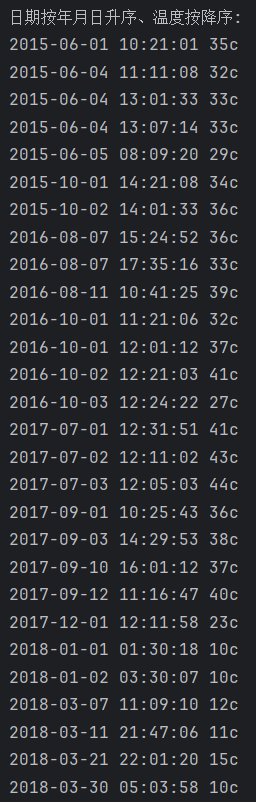
**File file = new File("weather.txt");**

**Scanner scanner = new Scanner(file);**

**输出：**

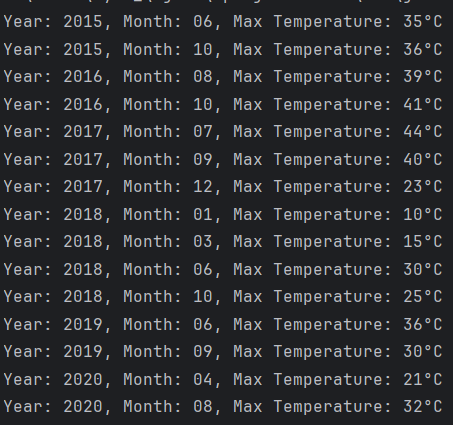


**代码：**

****

**2、输出每年每月最高温度和最低温度**

**代码：**

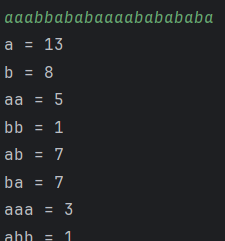
****

**二、从键盘输入一个任意的字符串（如：aaabbababaaaababababa），统计该字符串中长度为i的子串出现次数（i从1开始到上述字符串长度结束）。提示：可使用HashMap实现。**

**输出：**



**代码：**

****

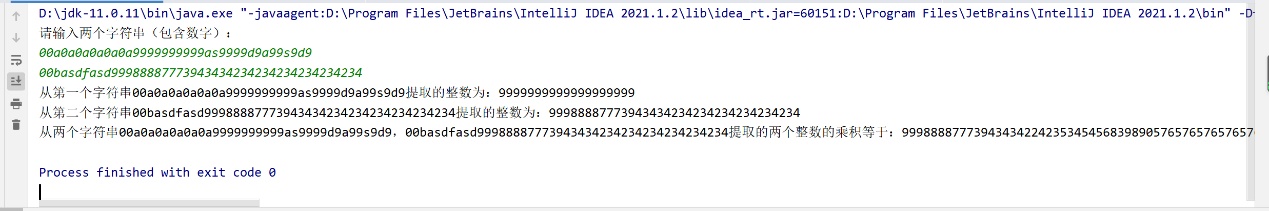
**三、从控制台输入两个字符串，每个字符串包含的字符数极大（如 10000 位），计算这两个字符串的整数乘法运算。所谓的“字符串整数乘法运算”定义为：提取字符串中的数字构成一个数（假设数字一定存在，并且小数点、 E 之类的忽略，仅考虑整数数字）， 计算提取的两个数的乘法。 如输入 a01b3 和 0034 两个字符串，可以解析到13 和34 两个整数（前面的0不要），并输出他们的乘积442。测试用例： 00a0a0a0a0a0a9999999999as9999d9a99s9d9 和00basdfasd99988887773943434234234234234234234**

**两个字符串，输出解析得到的整数，以及他们的乘积。**

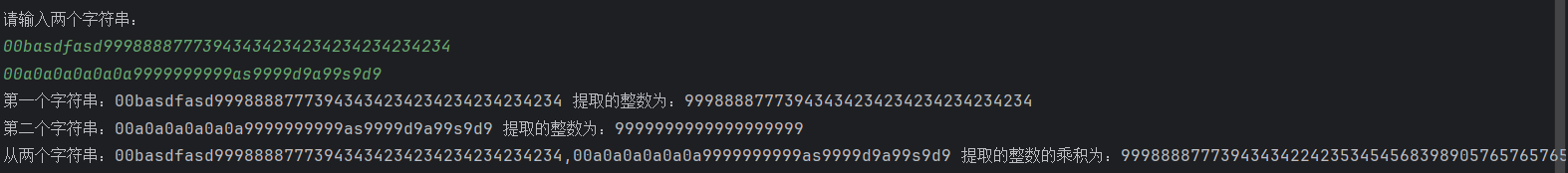
**注：使用BigInteger表示很大的数**

import java.math.BigInteger;

**输出：**



**代码：**

****

**四、对7个文件Lincoln, Abraham - The Writings of Abraham Lincoln Volume 1.txt~Lincoln, Abraham - The Writings of Abraham Lincoln Volume 7.txt中的所有单词进行词频统计，要求去除停用词、去除单词首尾的标点符号，并按词频大小按降序排列写到文件WordCount.txt中。可用HashMap实现单词词频记录。最后，用Python语言根据WordCount.txt文件画出词云图。**

**读文件可用语句：**

Scanner scanner = **null**;

File file = **new** File("Lincoln, Abraham - The Writings of Abraham Lincoln Volume 1.txt");

scanner = **new** Scanner(file);

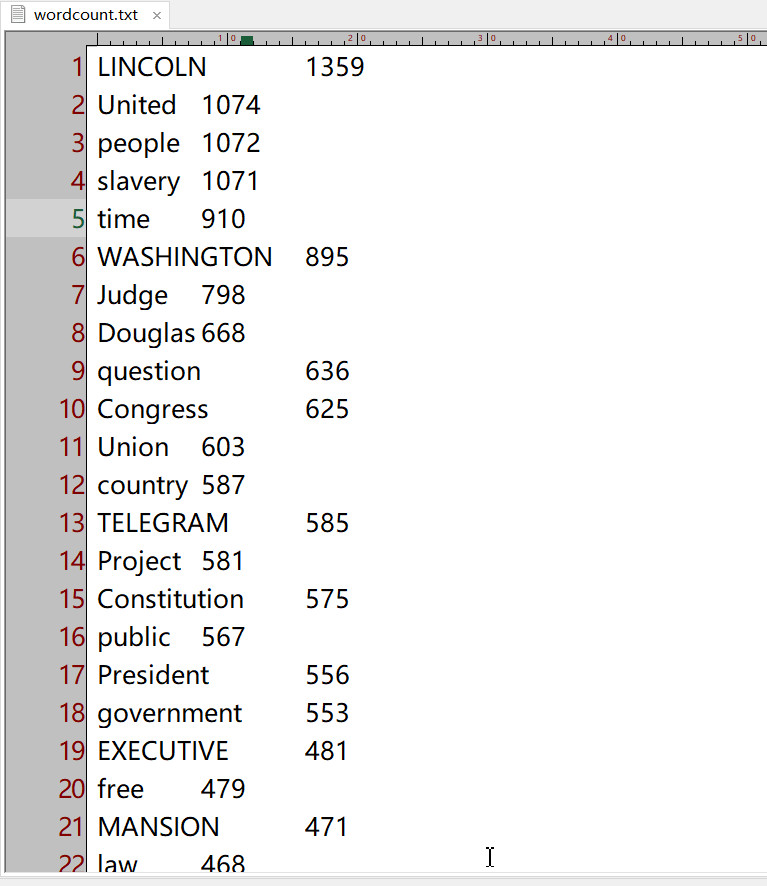
**写文件可用语句：**

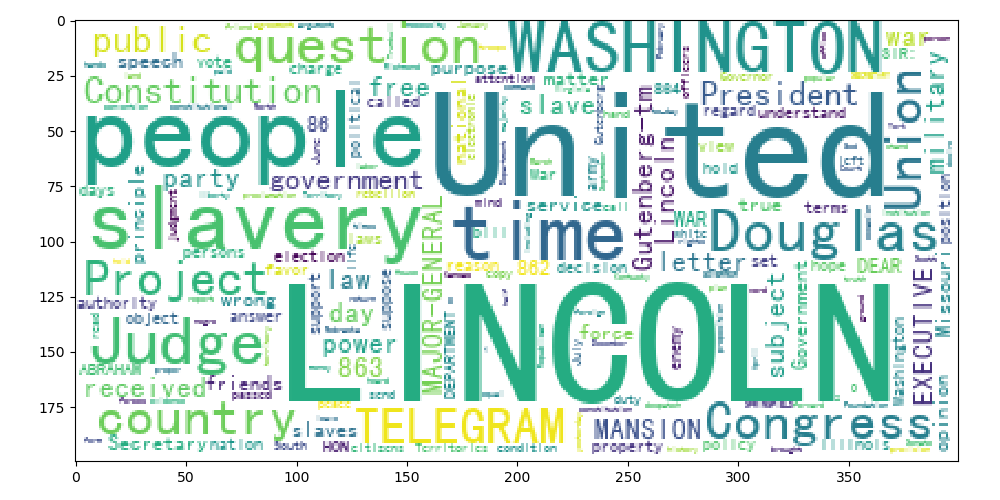
**import** java.io.PrintWriter;

PrintWriter pw = **new** PrintWriter(**"data/wordcount.txt"**);

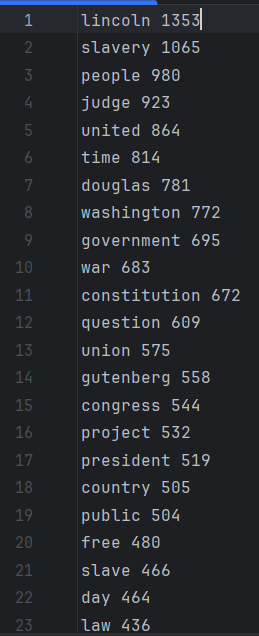
pw.write();

**输出：**



****

**Java代码：**

****

**画图Pyhton代码：**

**import** pandas **as** pd  
**from** wordcloud **import** WordCloud  
**import** matplotlib  
**import** matplotlib.pyplot **as** plt  
matplotlib.rcParams[**'figure.figsize'**]=(10.0,5.0)  
  
wordcloud=WordCloud(font\_path=**'data\simhei.ttf'**,background\_color=**'white'**,max\_font\_size=80)  
word\_stat=pd.read\_csv(**r'data/wordcount.txt'**,sep=**'\t'**,header=**None**,encoding=**'utf-8'**)  
word\_freq={x[0]:x[1] **for** x **in** word\_stat.values}  
wordcloud=wordcloud.fit\_words(word\_freq)  
plt.imshow(wordcloud)  
plt.show()

****