**2022 / 2023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： 学 号：**

**指导教师： 蒲 飞 班 级：计科2101、2102、2103 日 期： 2022.11.22**

**【一】上机实验内容及要求**

**实验要求：掌握File类的定义和使用，熟练使用File类的方法来处理文件和文件夹，熟悉文件的读写操作。深入了解自然语言理解的工具HanLP的使用，理解分词的各种使用方法和词性标注的含义，并能使用HanLP各种模块来解决实际问题。**

**实验平台：JDK11+IDEA2021+HanLP**

**实验内容：**

**一、枚举类**

**支付方式有Cash（现金）、WeChatPay（微信）、Alipay(支付宝)、BankCard(银行卡)、CreditCard(信用卡)五种，建立一个枚举类Pay包含这五种支付方式。类Pay有成员变量final String type（支付类型）和double mount（支付金额），构造方法为Pay(String type){…}。类pay实现了接口Message，Message有一个方法show(),要求每个支付方式类对象各自实现了这个方法。在类PayEnum中创建5个枚举类对象，设置支付数额，调用show()方法，输出如下：**

**输出：**

**文本

描述已自动生成**

**代码：**

**二、在指定目录下查找一个文件，如果目录或子目录下有查找的文件，返回文件所在的目录，否则输出 “当前目录下没有此文件”。**

**输出：**

文本

描述已自动生成

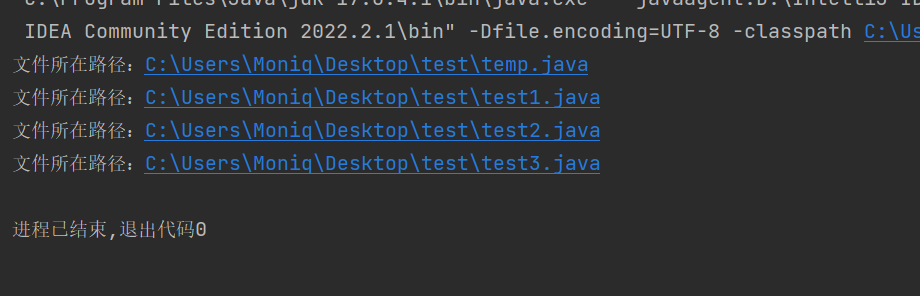
文本

描述已自动生成

**代码：**

**三、判断指定目录及其子目录下是否有后缀名为.java和.jpg的文件，如果有，就输出该文件名称和所在目录路径，如果没有，输出“\*\*目录下没有.java和.jpg文件”。**

**输出：**



**代码：**

**四、拷贝文件夹（内含各种格式的文件和子文件夹）到指定的目录下，如：D:\\a\\b，要求保留源文件夹的目录结构。**

**OddDir：**

**图形用户界面

描述已自动生成**

**NewDir：**

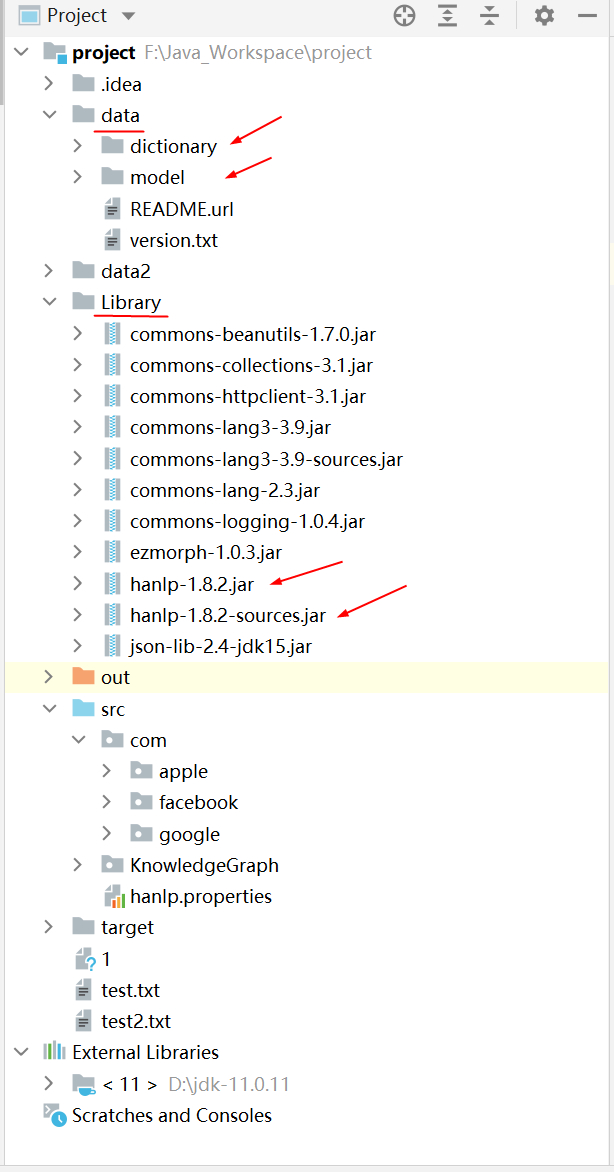
**图形用户界面

描述已自动生成**

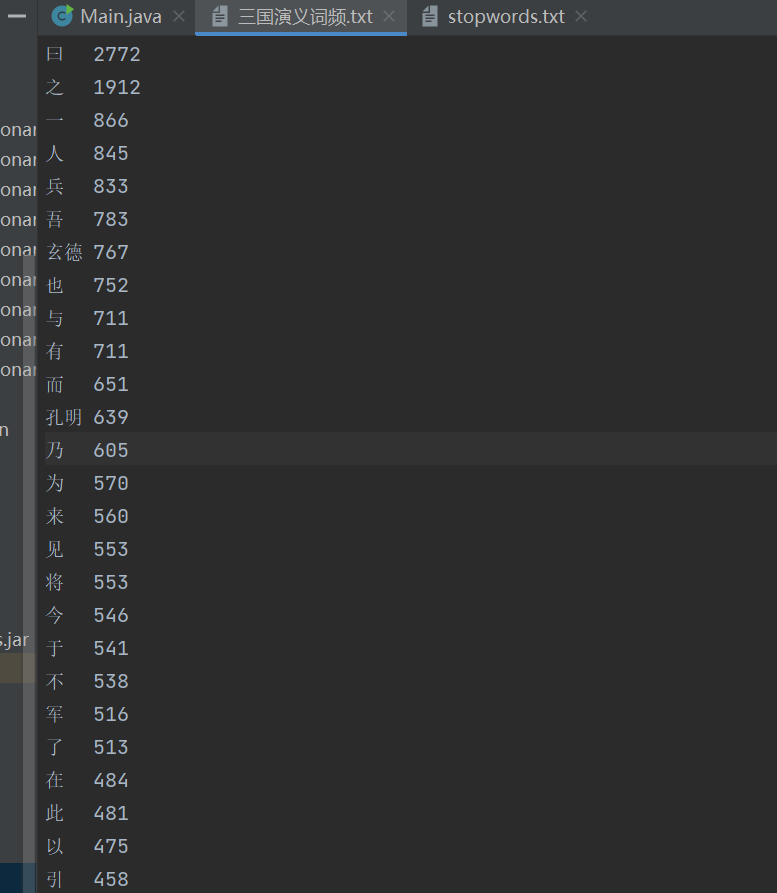
**代码：**

**五、（1）、对文件“三国演义(罗贯中).txt”进行分词，去掉标点符号和停用词，最后统计词频，排序输出到文件“三国演义词频.txt”，并画出词云。**

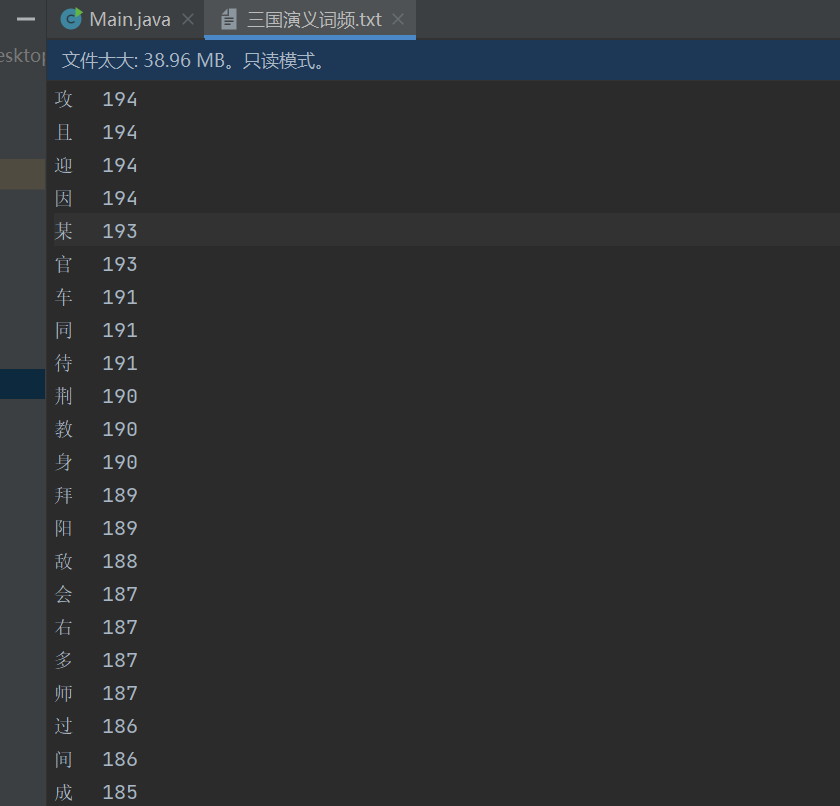
**HanLP配置：**

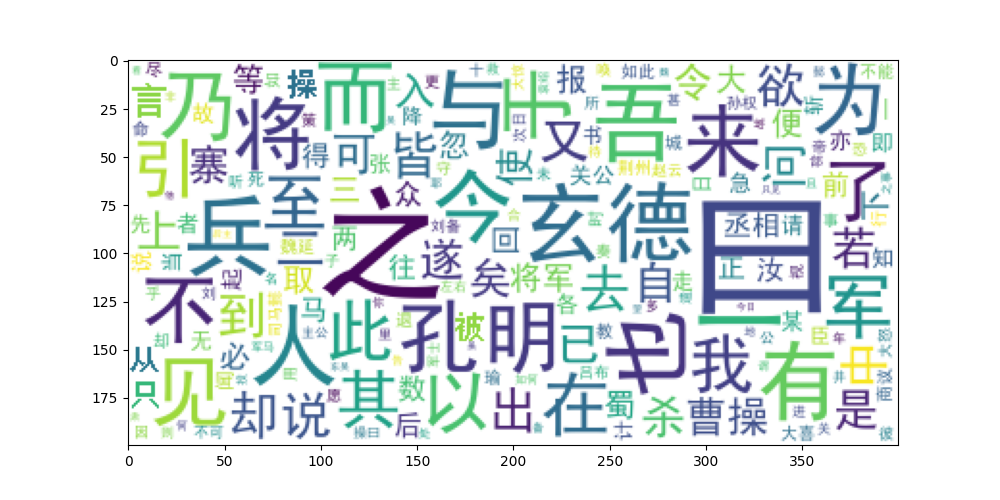


**在项目下建立目录data，把dictionary和model两目录放入data内。建立Library目录将hanlp-1.8.2.jar和hanlp-1.8.2-sources.jar放入，右键点Library，选add as Library。最后将hanlp.properties放在src目录下。**

****

**输出：**



****

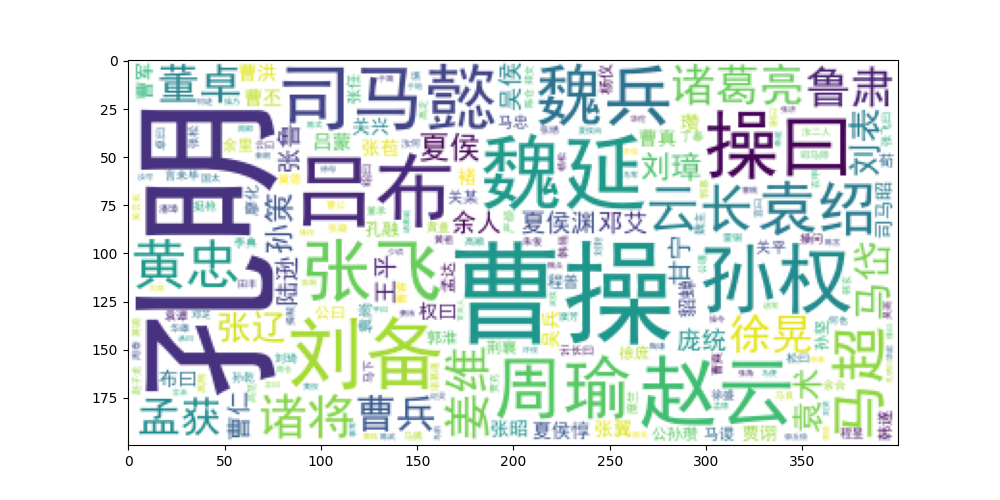
**代码：**

**（2）、统计三国演义小说中有多少个人物，将人物词频排序输出到文件，并画出人物的词频图谱**

**输出：**

**图形用户界面, 文本

中度可信度描述已自动生成**



**代码：**

**人物数：5099**

**画图：**

**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\_stat=pd.read\_csv(**r'data/Personage\_of\_the\_War\_among\_Three\_Kingdoms\_CRF.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()