**JAVA编程进阶上机报告**

****

第二次上机作业

**学 院\_智能与计算学部\_**

**专 业\_\_ 软件工程\_\_\_\_\_**

**姓 名\_\_\_ 李济孚\_\_ \_ \_**

**学 号\_\_ 3018216260\_\_\_**

**年 级\_\_\_\_ 2018\_\_\_\_\_\_**

**班 级\_\_\_\_\_\_ 5\_\_\_\_\_\_\_\_**

# 一、实验要求

要求:

编写程序，统计了不起的盖茨比中各个单词出现的频次。

注意事项

1. 尝试使用不同的 stream 进行读文件操作。

2. 异常处理（例如文件不存在，文件没有读权限，文件编码错误等）

输入:

了不起的盖茨比（英文版）.txt

(其中一个)

输出:

为输入文件，创建一个 output.txt

输出格式如下，单词+空格+频次，结果按照单词的频次倒序排列

hello 123

hi 12

i 1

# 二、源代码

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;

import java.util.Map.Entry;

import java.util.Scanner;

public class FileIO {

HashMap<String, Integer> counts = new HashMap<String, Integer>();

public void count()

{

Scanner scan = null;

try

{

File file = new File("C:\\Users\\Administrator\\Downloads\\了不起的盖茨比英文.txt");

scan = new Scanner(new FileInputStream(file));

while(scan.hasNext())

{

String str = scan.next();

if(counts.containsKey(str))

counts.put(str, counts.get(str)+1);

else

counts.put(str, 1);

}

}catch(Exception e)

{

e.toString();

}finally

{

try {

scan.close();

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

public void OutputResult()

{

String path = "C:\\Users\\Administrator\\Downloads\\result.txt";

BufferedWriter bw = null;

try

{

File file = new File(path);

if(!file.exists())

file.getParentFile().mkdirs();

file.createNewFile();

//wirte

FileWriter fw = new FileWriter(file);

bw = new BufferedWriter(fw);

List<HashMap.Entry<String,Integer>> list = new ArrayList<>(counts.entrySet());

Collections.*sort*(list,new Comparator<HashMap.Entry<String, Integer>>() {

@Override

public int compare(Entry<String, Integer> o1, Entry<String, Integer> o2) {

return o2.getValue()-o1.getValue(); //重写排序规则，小于0表示升序，大于0表示降序

}

});

Iterator<HashMap.Entry<String,Integer>> i = list.iterator();

while(i.hasNext())

{

Entry<String,Integer> ii= i.next();

System.*out*.println(ii.getKey() + " " + ii.getValue());

bw.write(ii.getKey() + " " + ii.getValue());

bw.newLine();

}

bw.flush();

}catch(Exception e)

{

e.toString();

}finally

{

try {

bw.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

public static void main(String[] args)

{

FileIO io = new FileIO();

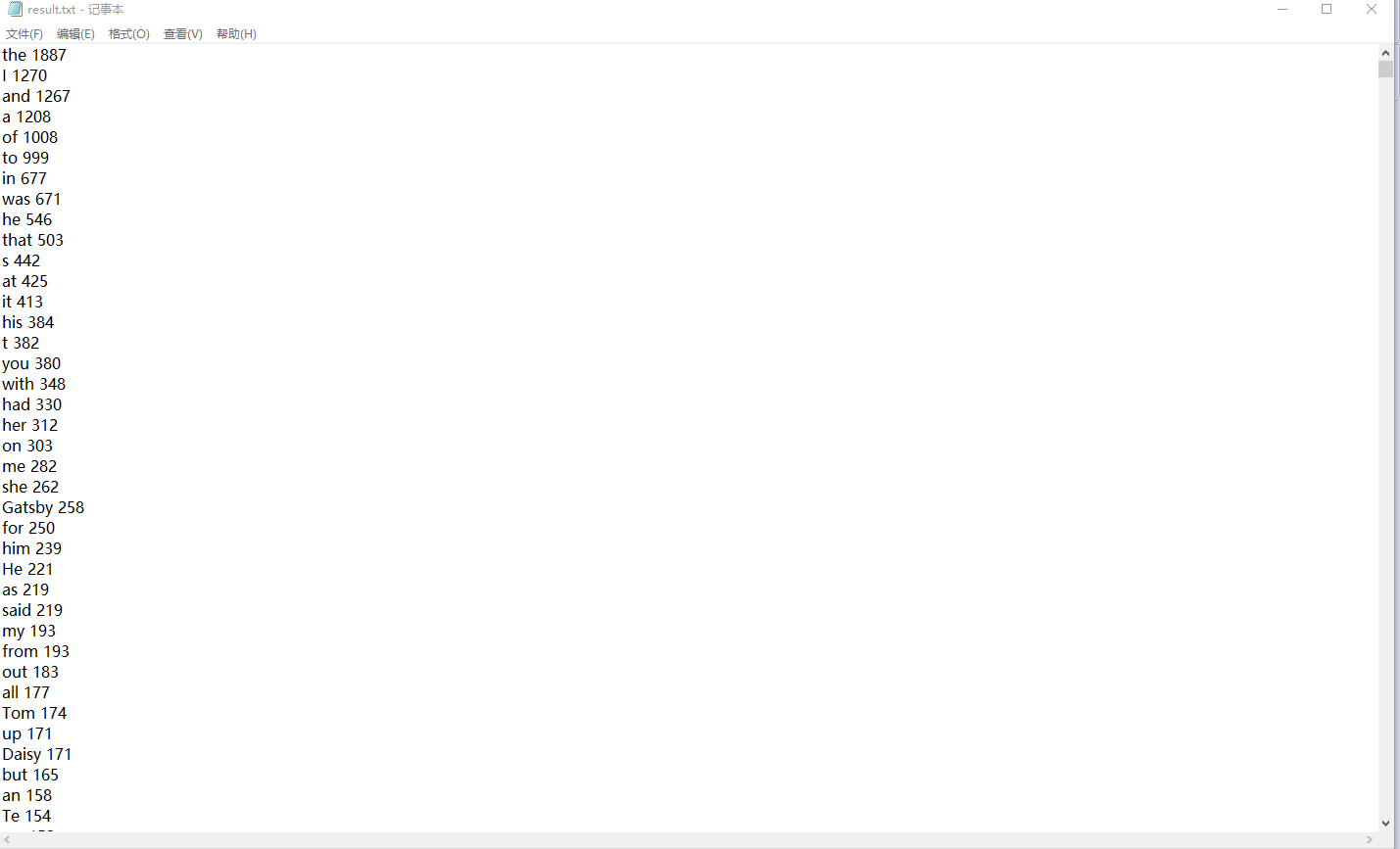
io.count();

io.OutputResult();

}

}

# 三、运行结果



# 四、设计思路和UML

读取文本我们用File类创建一个对象。并开一个FileInputStream的流。

由于要读取文本的所有单词，而但此语单词之间是用空格间隔的，所以我们可以考虑用scanner类中的next()方法，而不是用BufferedReader的nextLine()。而根据要求，我们要将结果输出在一个文本里，所以就在类里面设置两个方法，第一个方法作为读取文本并将结果暂时储存在成员变量HashMap counts中。第二个方法是读取counts并将结果输出在文本里。

在进行结果输出时候，创建文本要用mirks()函数，以保证在没有上级目录的情况下也能生成文件。

细节：在文件读取和文件生成的操作中，都要进行错误处理。而在counts的输出中，先使用list+entry的组合将结果进行排序，再进行输出。

以下是类图

