JAVA 编程进阶上机报告

学　　院 智能与计算学部

专　　业 软件工程

姓　　名 王传安

学　　号 3018216301

年　　级 2018 级

班　　级 软工六班

一、实验要求

编写程序，统计了不起的盖茨比中各个单词出现的频次。

输入:

了不起的盖茨比(英文版).txt （其中一个)

输出:

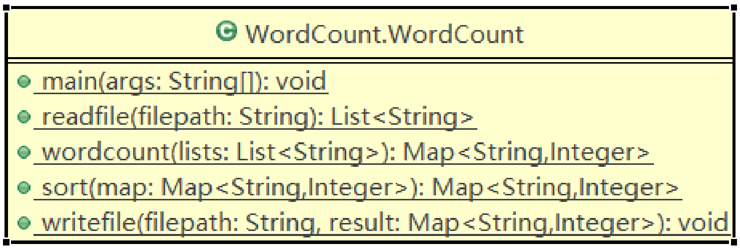
为输入文件，创建一个 output.txt

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、设计思路

UML 图



WordCount.WordCount

main(args: String[]): void

readfile(filepath: String): List<String>

wordcount(lists: List<String>): Map<String,Integer>

sort(map: Map<String,Integer>): Map<String,Integer>

writefile(filepath: String, result: Map<String,Integer>): void

readfile 方法：从传入的路径读取文件内容，并将内容划分为单词，存入 list 中并返回。

wordcount 方法:从传入的 list 中统计每个单词出现的次数,将结果存入一个 map 中并返回。

sort 方法：把传入的 map 根据 value 值进行降序排序，将结果存入一个 map 中并返回

writefile 方法：传入一个要写入文件的路径，和要写入文件的 map，将 map 内的数据格式化写入到指定的文件中，若文件不存在，则自动创建文件。

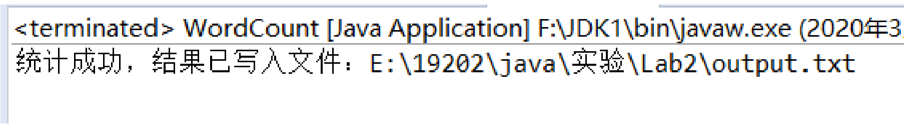
三、源代码

```
package WordCount;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
```

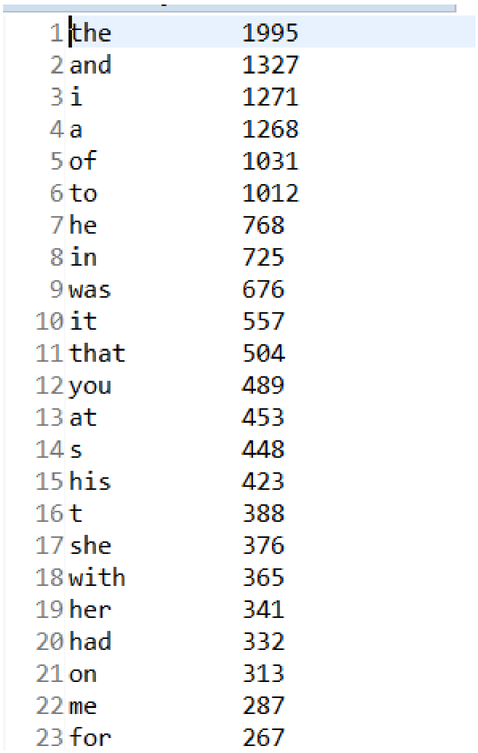
```java
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.Iterator;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;
import java.util.TreeMap;

public class WordCount {
        public static void main(String args[]) throws Exception{
            String filepath = "E:\\19202\\java\\实验\\Lab2\\了不起的盖茨比英文.txt";

            List<String> lists = readfile(filepath);

            Map<String,Integer> map = wordcount(lists);

            Map<String,Integer> result = sort(map);

            String result_file_path = "E:\\19202\\java\\实验\\Lab2\\output.txt";

            writefile(result_file_path, result);

            System.out.println("统计成功，结果已写入文件：" + result_file_path);
        }

        /*
         * 从给定文件中读取数据，划分出单词，并把结果存入一个list中返回
         */
        public static List<String> readfile(String filepath) throws Exception {
            BufferedReader  br = new BufferedReader(new FileReader(filepath));
            List<String> lists = new ArrayList<String>();
            String readline = null;
            while((readline = br.readLine()) != null){
                String[] wordsArr = readline.split("[^a-zA-Z]");

                for (int i=0; i<wordsArr.length; i++){
                    if(wordsArr[i] != ""){
                        lists.add(wordsArr[i]);
                    }
                }
            }
            br.close();
            return lists;
        }

        /*
         * 从传入的 list 中统计单词出现的次数，将结果存入 map 中并返回
         */
        public static Map<String,Integer> wordcount(List<String> lists){
            Map<String,Integer> map=new TreeMap<String,Integer>();
            for(int j=0; j<lists.size(); j++){
                if(lists.get(j) != ""){
                    String key=lists.get(j).toLowerCase();
                    if(map.get(key) != null){
                        map.put(key, map.get(key)+1);
                    }
                    else{
                        map.put(key, 1);
                    }
                }
            }
            return map;
        }

        /*
         * 对传入的 map 根据 value 值降序排序，将结果存入一个新的 map 并返回
         */
        public static Map<String,Integer> sort(Map<String,Integer> map){
            List<Map.Entry<String,Integer>> list = new ArrayList<>(map.entrySet());
            Collections.sort(list, new Comparator<Map.Entry<String,Integer>>(){
                @Override
                public int compare(Map.Entry<String,Integer> o1,
Map.Entry<String,Integer> o2){
                    int compare = (o1.getValue()).compareTo(o2.getValue());
                    return -compare;
                }
            });

            Map<String,Integer> returnMap = new LinkedHashMap<String,Integer>();
            for (Map.Entry<String,Integer> entry : list) {
                returnMap.put(entry.getKey(), entry.getValue());
            }
            return returnMap;
```

```
	}

	/*
	 * 将传入的 map 中的信息格式化写出到传入的指定文件中。
	 */
	public static void writefile(String filepath, Map<String,Integer> result) {
		File file = null;
		FileWriter fw = null;
		try {
			file = new File(filepath);
			if (!file.exists()) {
				file.createNewFile();
			}
			fw = new FileWriter(file);
			for(Entry<String, Integer> entry:result.entrySet()) {
				String str = String.format("%-12s", entry.getKey()) + String.valueOf(entry.getValue()) + "\r\n";
				fw.write(str);
			}
			fw.close();
		} catch (IOException e) {
			e.printStackTrace();
		}
	}

}
```

四、实验结果



```
<terminated> WordCount [Java Application] F:\JDK1\bin\javaw.exe (2020年3
统计成功，结果已写入文件：E:\19202\java\实验\Lab2\output.txt
```



```
the            1995
and            1327
i              1271
a              1268
of             1031
to             1012
he             768
in             725
was            676
it             557
that           504
you            489
at             453
s              448
his            423
t              388
she            376
with           365
her            341
had            332
on             313
me             287
for            267
```