JAVA 编程进阶上机报告

学 院 智能与计算学部

专 业 软件工程

班 级 6 班

学 号 3018216298

姓 名 米思成

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:

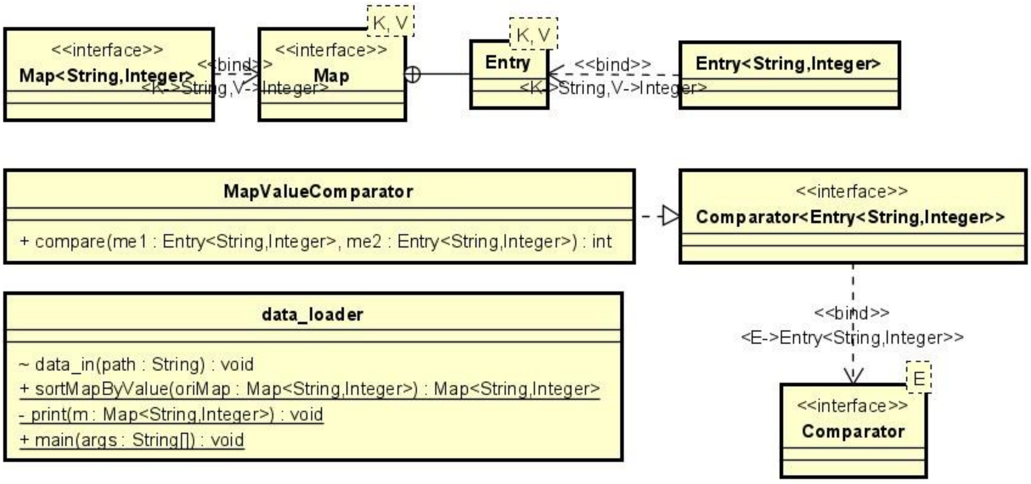
了不起的盖茨比（英文版）.txt

(其中一个)

输出:

为输入文件，创建一个 output.txt

二、UML 图（设计）



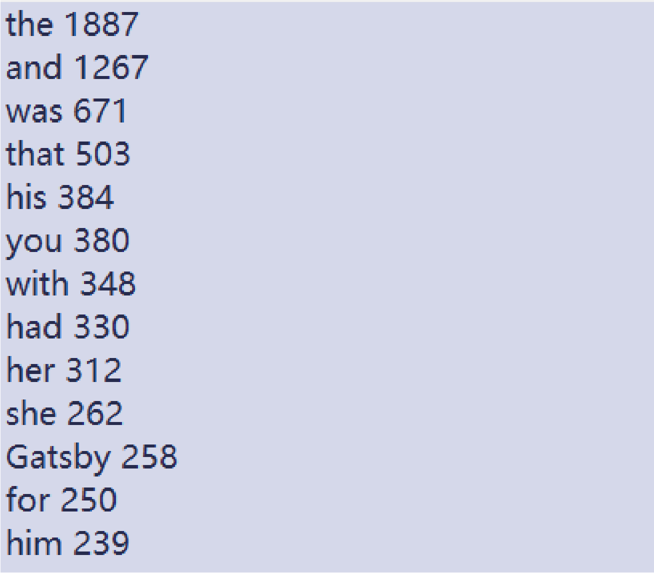
<<interface>>
Map<String,Integer>
<<bind>>
<K->String,V->Integer>
K, V
<<interface>>
Map
K, V
Entry
<<bind>>
<K->String,V->Integer>
Entry<String,Integer>
MapValueComparator
+ compare(me1 : Entry<String,Integer>, me2 : Entry<String,Integer>) : int
<<interface>>
Comparator<Entry<String,Integer>>
<<bind>>
<E->Entry<String,Integer>>
E
<<interface>>
Comparator
data_loader
~ data_in(path : String) : void
+ sortMapByValue(oriMap : Map<String,Integer>) : Map<String,Integer>
- print(m : Map<String,Integer>) : void
+ main(args : String[]) : void

三、源代码

```java
public class data_loader
{
    void data_in(String path) throws IOException
    {
        File txt = new File(path);
        BufferedReader br = null;
        Map<String,Integer> map = new TreeMap<String,Integer>();
        try
        {
            br=new BufferedReader(new FileReader(txt));
        }
        catch (FileNotFoundException e)
        {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        String line="";
        int cnt=0;
        List<String> l = new ArrayList<String>();
        try
        {
            while ((line = br.readLine()) != null)
            {
                //数组里面每个元素是一个单词

                String str[]=line.split(" ");

                /*

                 * 数据处理部分

                 */


                //将所有的大写转为小写并除去所有符号

                for(int i=0;i<str.length;i++) {
                    str[i] = str[i].toLowerCase();

                    String regEx="[\n`~!@#$%^&*()+=|{}':;',\\[\\].<>/?~！
@#￥%……&*（）——+|{}【】‘；：”“’。，、？]";
                    String x = "";
                    str[i] = str[i].replaceAll(regEx,x);
                    l.add(str[i]);
                }
            }


          //统计词频

            ii:
                for(int i=0; i<l.size(); i++) {


                    //判断之前是否出现过这个字符串

                    for(int k=0;k<i;k++) {
                        if(l.get(i).equals(l.get(k)))
                            continue ii;
                    }

                    //count此字符串出现过的次数

                    int temp = 1;
                    for(int j=i+1; j<l.size(); j++) {

                        if(!l.get(i).equals(l.get(j))) {

                        }
                        else {
                            temp++;
                        }
                    }

                    //把字符串添加入Map

                    map.put((String) l.get(i), temp);
                }


            Map<String, Integer> resultMap = sortMapByValue(map);    //按

Key进行排序

            print(resultMap);

        }
        catch (IOException e)
        {
            e.printStackTrace();
            }
        }


    //排序类在这里，还有最后的MapValueComparator类也是

    public static Map<String, Integer> sortMapByValue(Map<String, Integer>
oriMap) {
        if (oriMap == null || oriMap.isEmpty()) {
            return null;
        }
        Map<String, Integer> sortedMap = new LinkedHashMap<String,
Integer>();
        List<Map.Entry<String, Integer>> entryList = new
ArrayList<Map.Entry<String, Integer>>(
                oriMap.entrySet());
        Collections.sort(entryList, new MapValueComparator());

        Iterator<Map.Entry<String, Integer>> iter = entryList.iterator();
        Map.Entry<String, Integer> tmpEntry = null;
        while (iter.hasNext()) {
            tmpEntry = iter.next();
            sortedMap.put(tmpEntry.getKey(), tmpEntry.getValue());
        }
        return sortedMap;
    }



    private static void print(Map<String, Integer> m) {
        Set<Map.Entry<String,Integer>> set = m.entrySet();
        Iterator<Map.Entry<String,Integer>> iter = set.iterator();
        File file = new File("E:/output.txt");
        try {
         file.createNewFile();
         FileWriter fileWriter = new FileWriter(file.getAbsoluteFile());
         BufferedWriter bw = new BufferedWriter(fileWriter);
         while(iter.hasNext()) {
            Map.Entry<String, Integer> ele = iter.next();
            String temp = ele.getKey()+" "+ele.getValue()+"\n";
            bw.write(temp);
        }
            bw.close();
        }catch(IOException e){
           e.printStackTrace();
        }

    }


    public static void main(String[] args) throws IOException
    {
        data_loader L=new data_loader();

        String path="E:/了不起的盖茨比英文.txt";

        L.data_in(path);
    }

}
```

```
class MapValueComparator implements Comparator<Map.Entry<String, Integer>>
{

    @Override
    public int compare(Entry<String, Integer> me1, Entry<String, Integer>
me2) {

        return me1.getValue().compareTo(me2.getValue());
    }
}
```

四、实验结果



```
the 1887
and 1267
was 671
that 503
his 384
you 380
with 348
had 330
her 312
she 262
Gatsby 258
for 250
him 239
```