# 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 五班**

**学 号 3018216242**

**姓 名 邢思洋**

1. **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。

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

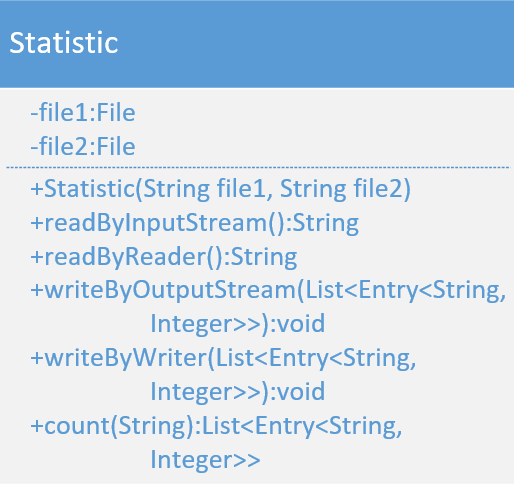
 hello 123

 hi 12

 i 1

1. **设计思路和UML图**

先通过字符流或字节流将文件读入，再将之转化为需要的输出内容，最后通过字符流或字节流将输出文件写出。



1. **源代码**

**package** xsy.lab2;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.OutputStream;

**import** java.io.Reader;

**import** java.io.Writer;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Map.Entry;

**public** **class** Statistic

{

**private** File file1;

**private** File file2;

**public** Statistic(String file1, String file2)

{

**this**.file1 = **new** File(file1);

**this**.file2 = **new** File(file2);

}

**public** String readByInputStream() **throws** IOException

{

**try**

{

InputStream in = **new** FileInputStream(file1);

**try**

{

StringBuffer buf = **new** StringBuffer();

**byte**[] bytes = **new** **byte**[1024];

**int** length = in.read(bytes);

**while** (length != -1)

{

String str = **new** String(bytes, 0, length);

buf.append(str);

length = in.read(bytes);

}

**return** buf.toString();

}

**catch** (IOException e)

{

e.printStackTrace();

}

}

**catch** (FileNotFoundException e)

{

e.printStackTrace();

}

**return** **null**;

}

**public** String readByReader() **throws** IOException

{

**try**

{

Reader in = **new** FileReader(file1);

**try**

{

StringBuffer buf = **new** StringBuffer();

**char**[] chars = **new** **char**[1024];

**int** length = in.read(chars);

**while** (length != -1)

{

buf.append(chars, 0, length);

length = in.read(chars);

}

**return** buf.toString();

}

**catch** (IOException e)

{

e.printStackTrace();

}

}

**catch** (FileNotFoundException e)

{

e.printStackTrace();

}

**return** **null**;

}

**public** List<Map.Entry<String, Integer>> count(String str)

{

Map<String, Integer> map = **new** HashMap<String, Integer>();

StringBuffer buf = **new** StringBuffer();

**for** (**int** i = 0; i < str.length(); i++)

{

**if** (str.charAt(i) != ' ' && str.charAt(i) != '\n')

{

buf.append(str.charAt(i));

}

**else** **if** (buf.length() > 0)

{

**if** (map.containsKey(buf.toString()))

{

map.put(buf.toString(), map.get(buf.toString()) + 1);

buf.setLength(0);

}

**else**

{

map.put(buf.toString(), 1);

buf.setLength(0);

}

}

}

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)

{

**return** o2.getValue() - o1.getValue();

}

});

**return** list;

}

**public** **void** writeByOutputStream(List<Map.Entry<String, Integer>> list) **throws** IOException

{

**if**(!file2.getParentFile().exists())

{

file2.getParentFile().mkdirs();

}

OutputStream out = **new** FileOutputStream(file2);

**try**

{

**for** (Map.Entry s : list)

{

out.write((s.getKey()+ "--" +s.getValue() + "\n").getBytes());

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

}

}

**catch** (FileNotFoundException e)

{

e.printStackTrace();

}

**finally**

{

out.close();

}

}

**public** **void** writeByWriter(List<Map.Entry<String, Integer>> list) **throws** IOException

{

**if**(!file2.getParentFile().exists())

{

file2.getParentFile().mkdirs();

}

Writer out = **new** FileWriter(file2);

**try**

{

**for** (Map.Entry s : list)

{

out.write((s.getKey()+ "--" +s.getValue() + "\n"));

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

}

}

**catch** (FileNotFoundException e)

{

e.printStackTrace();

}

**finally**

{

out.close();

}

}

}

**package** xsy.lab2;

**import** java.util.List;

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.Map;

**public** **class** text

{

**public** **static** **void** statisticByReaderWriter(Statistic sta) **throws** IOException

{

String str = sta.readByReader();

List<Map.Entry<String, Integer>> list = sta.count(str);

sta.writeByWriter(list);

System.***out***.println("统计成功");

}

**public** **static** **void** statisticByInputStreamOutputStream(Statistic sta) **throws** IOException

{

String str = sta.readByInputStream();

List<Map.Entry<String, Integer>> list = sta.count(str);

sta.writeByOutputStream(list);

System.***out***.println("统计成功");

}

**public** **static** **void** main(String[] args) **throws** IOException

{

String file1 = "D:" + File.***separator*** + "demo2020" + File.***separator*** + "了不起的盖茨比英文.txt";

String file2 = "D:" + File.***separator*** + "demo2020" + File.***separator*** + "output.txt";

Statistic sta = **new** Statistic(file1, file2);

*statisticByInputStreamOutputStream*(sta);

// statisticByInputStreamOutputStream(sta);

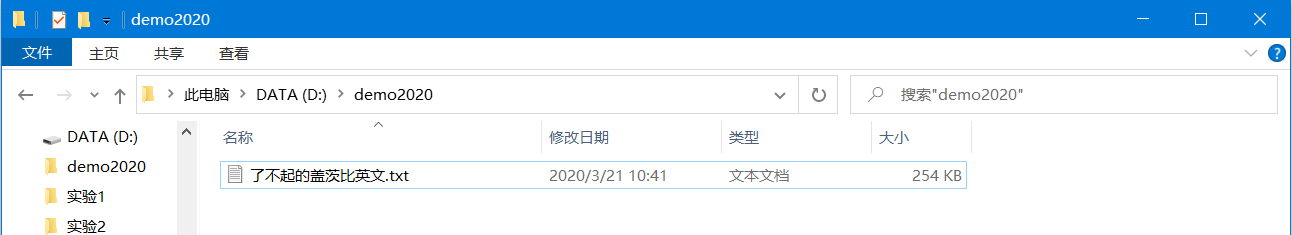
}

}

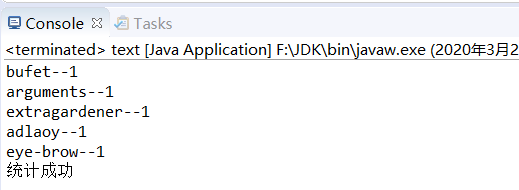
1. **实验结果**

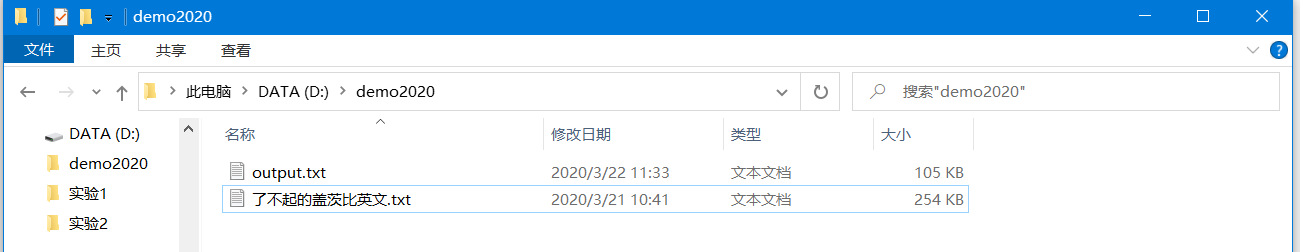
**字节流：**

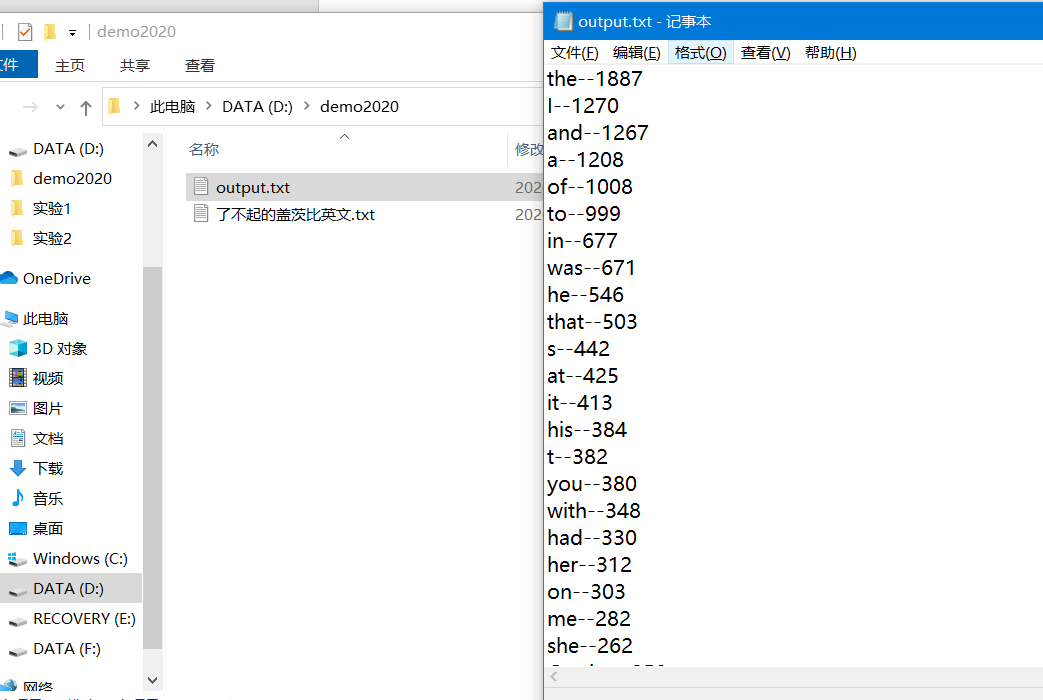
执行前：



执行后：

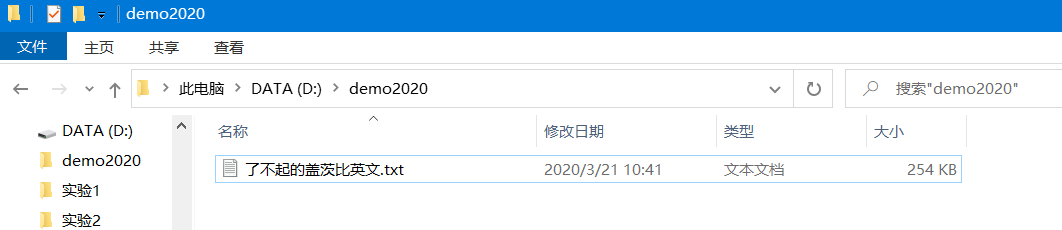






**字符流：**

执行前：



执行后：

