JAVA 编程进阶上机报告

Lab 4 Container and IO

学　　院__智能与计算学部__
专　　业__　软件工程　__
姓　　名_____张瑞安_____
学　　号____3018218063____
年　　级_____2018　_____
班　　级_____4　_____

一、实验要求

1、实验名称

词频统计

2、实现功能

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

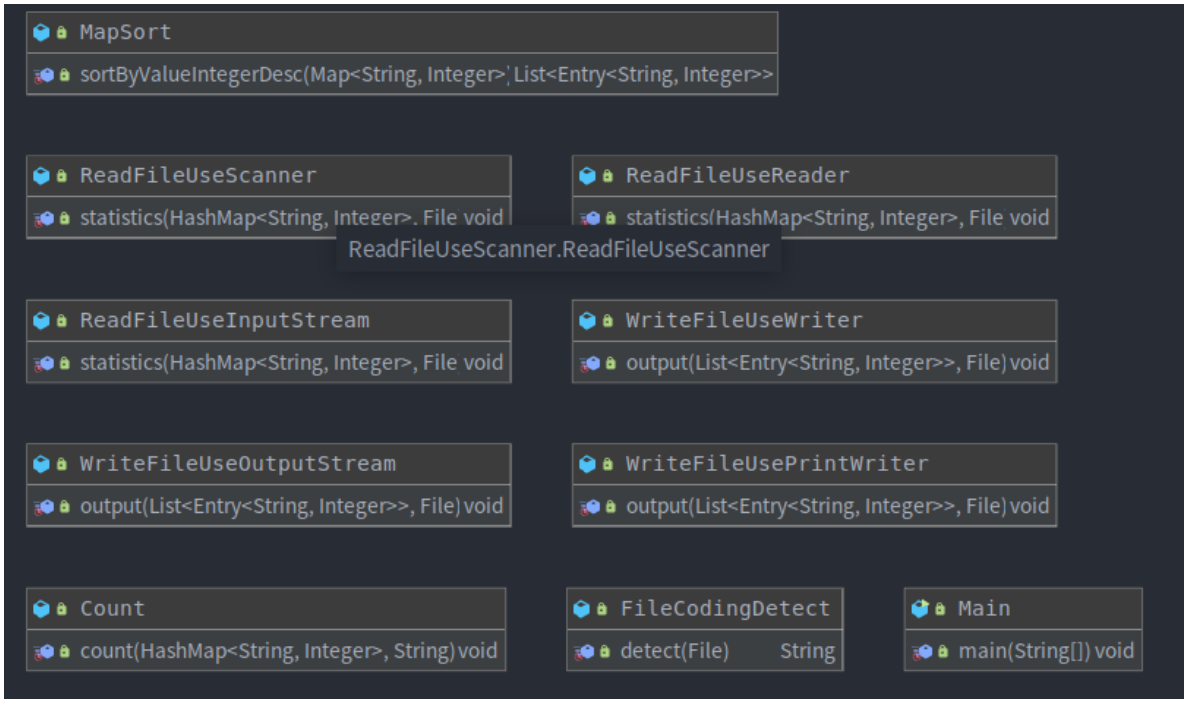
- 尝试使用不同的 stream 进行读文件操作。
- 异常处理（例如文件不存在，文件没有读权限，文件编码错误等）

二、源代码

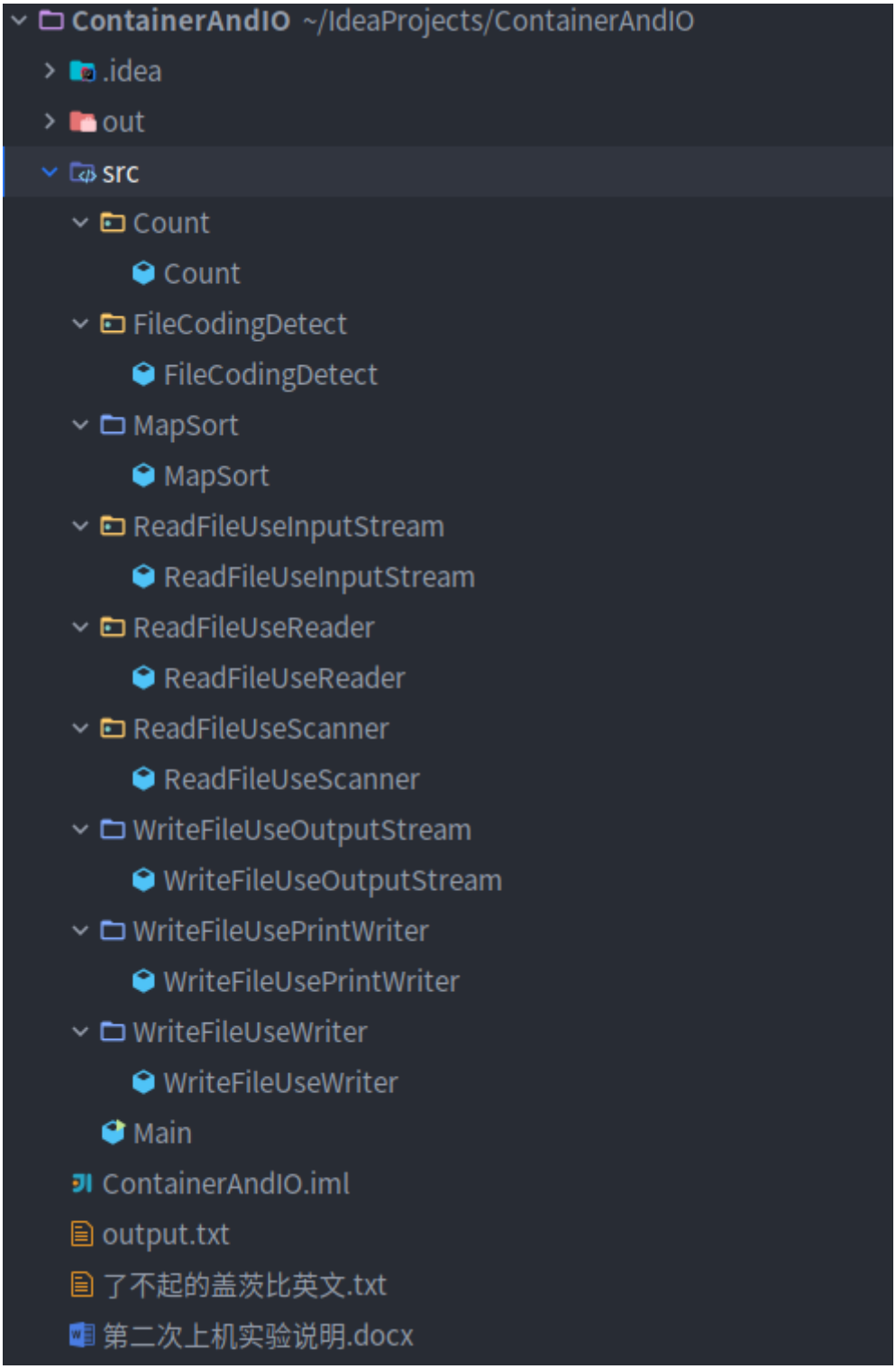
1、设计思路

- 功能与过程分解：
  - 文件权限设置：
    - `了不起的盖茨比英文.txt` 设置当前用户可读
    - 创建 `output.txt` 并设置当前用户可写
  - 文件编码判断
    - 对于 `UTF-8` 编码格式的文本文件，其前3个字节的值就是-17、-69、-65
  - 输入：
    - 可选工具
      - `InputStream`
      - `Reader` (文件中若有中文字符慎用)
      - `Scanner`
    - 防止文件较大带来的内存问题，采用按行输入的方式
  - 统计：
    - 设置 `HashMap<String, Integer>`，按行输入统计，排除的所有单个字母的单词（a和i除外）。
  - 排序：
    - 将 `Map.entrySet()` 转换为 `List`，通过 `Collection` 的比较器实现降序排序
  - 输出：
    - 可选工具
      - `OutputStream`
      - `PrintWriter` (文件中若有中文字符慎用)
      - `Writer` (文件中若有中文字符慎用)
    - 提取 `List` 中排好序的 `Map.Entry` 并逐个输出至 `output.txt`

2、类图


MapSort
sortByValueIntegerDesc(Map<String, Integer>, List<Entry<String, Integer>>
ReadFileUseScanner
statistics(HashMap<String, Integer>, File) void
ReadFileUseReader
statistics(HashMap<String, Integer>, File) void
ReadFileUseScanner.ReadFileUseScanner
ReadFileUseInputStream
statistics(HashMap<String, Integer>, File) void
WriteFileUseWriter
output(List<Entry<String, Integer>>, File) void
WriteFileUseOutputStream
output(List<Entry<String, Integer>>, File) void
WriteFileUsePrintWriter
output(List<Entry<String, Integer>>, File) void
Count
count(HashMap<String, Integer>, String) void
FileCodingDetect
detect(File) String
Main
main(String[]) void

3、文件结构



4、源代码

FileCodingDetect.java

```
package FileCodingDetect;

import java.io.*;

public class FileCodingDetect {

    public static String detect(File file) throws IOException {
        InputStream inputStream = new FileInputStream(file);
        byte[] b = new byte[3];
        inputStream.read(b);
```

```
        inputStream.close();
        if (b[0] == -17 && b[1] == -69 && b[2] == -65)
            return "UTF-8";
        else
            return "others";
    }

}
```

ReadFileUseInputStream.java

```
package ReadFileUseInputStream;

import java.util.*;
import java.io.*;
import Count.Count;

public class ReadFileUseInputStream {

    public static void statistics(HashMap<String, Integer> count, File file) throws IOException {
        FileInputStream fis = new FileInputStream(file);
        InputStreamReader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);

        String line;
        while ((line = br.readLine()) != null) {
            Count.count(count, line);
        }

        fis.close();
        isr.close();
        br.close();
    }
}
```

ReadFileUseReader.java

```
package ReadFileUseReader;

import java.util.*;
import java.io.*;
import Count.Count;

public class ReadFileUseReader {

    public static void statistics(HashMap<String, Integer> count, File file) throws FileNotFoundException,
IOException {
        FileReader fr = new FileReader(file);
        BufferedReader br = new BufferedReader(fr);

        String line;
        while ((line = br.readLine()) != null) {
            Count.count(count, line);
        }

        fr.close();
```

```
        br.close();
    }

}
```

ReadFileUseScanner.java

```
package ReadFileUseScanner;

import java.io.*;
import java.util.*;
import Count.Count;

public class ReadFileUseScanner {

    public static void statistics(HashMap<String, Integer> count, File file) throws FileNotFoundException {
        Scanner sc = new Scanner(file);
        while (sc.hasNextLine()) {
            String line = sc.nextLine();
            Count.count(count, line);
        }
    }
}
```

Count.java

```
package Count;

import java.lang.reflect.Array;
import java.util.*;

public class Count {

    public static void count(HashMap<String, Integer> count, String line) {
        String[] chars = new String[]{
                "b","c","d","e","f","g","h","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z"
        };
        List<String> letters = Arrays.asList(chars);
        String[] words = line.split(" ");
        for (String word : words) {
            if (word.trim().length()>0 && !letters.contains(word)){
                word = word.toLowerCase();
                if (count.containsKey(word)) {
                    count.put(word, count.get(word) + 1);
                } else {
                    count.put(word, 1);
                }
            }
        }
    }
}
```

MapSort.java

```
package MapSort;
```

```
import java.util.*;

public class MapSort {

   public static List<Map.Entry<String,Integer>> sortByValueIntegerDesc(Map<String, Integer>
nowPartTwoData){
      //这里将map.entrySet()转换成list
      List<Map.Entry<String,Integer>> list = new ArrayList<Map.Entry<String,Integer>>
(nowPartTwoData.entrySet());
      //然后通过比较器来实现排序
      Collections.sort(list,new Comparator<Map.Entry<String,Integer>>() {
         //降序排序
         @Override
         public int compare(Map.Entry<String, Integer> o1,
                  Map.Entry<String, Integer> o2) {
            return o2.getValue().compareTo(o1.getValue());
         }
       });
      return list;
   }

}
```

WriteFileUseOutputStream.java

```
package WriteFileUseOutputStream;

import java.io.*;
import java.util.*;

public class WriteFileUseOutputStream {

   public static void output(List<Map.Entry<String, Integer>> list, File file) throws IOException{
      FileOutputStream fos = new FileOutputStream(file);
      for (Map.Entry<String, Integer> e: list) {
         String string = e.getKey()+" "+e.getValue()+"\n";
         byte[] strToByte = string.getBytes();
         fos.write(strToByte);
      }
      fos.close();
   }

}
```

WriteFileUsePrintWriter.java

```
package WriteFileUsePrintWriter;

import java.io.*;
import java.util.*;

public class WriteFileUsePrintWriter {

   public static void output(List<Map.Entry<String, Integer>> list, File file) throws IOException {
      FileWriter fw = new FileWriter(file);
      PrintWriter pw = new PrintWriter(fw);
```

```
        for (Map.Entry<String,Integer> e : list){
            pw.println(e.getKey()+" "+e.getValue());
        }

        pw.close();

    }

}
```

WriteFileUseWritter.java

```
package WriteFileUseWriter;

import java.io.*;
import java.util.*;

public class WriteFileUseWriter {

    public static void output(List<Map.Entry<String, Integer>> list, File file) throws IOException {
        FileWriter fw = new FileWriter(file);
        BufferedWriter bw = new BufferedWriter(fw);

        for (Map.Entry<String,Integer> e : list){
            bw.append(e.getKey()+" "+e.getValue()+"\n");
        }

        bw.close();

    }

}
```

Main.java

```
import FileCodingDetect.FileCodingDetect;
import MapSort.MapSort;
import ReadFileUseInputStream.ReadFileUseInputStream;
import ReadFileUseReader.ReadFileUseReader;
import ReadFileUseScanner.ReadFileUseScanner;
import WriteFileUseOutputStream.WriteFileUseOutputStream;
import WriteFileUsePrintWriter.WriteFileUsePrintWriter;
import WriteFileUseWriter.WriteFileUseWriter;
import org.w3c.dom.ls.LSOutput;

import java.io.*;
import java.util.*;

public class Main {

    public static void main(String[] args) throws IOException {
        File file = new File("了不起的盖茨比英文.txt");
        file.setReadable(true);
        File output = new File("output.txt");
        output.setWritable(true);
```

```
        HashMap<String,Integer> count = new HashMap<>();
        if (FileCodingDetect.detect(file) == "UTF-8"){

            ReadFileUseInputStream.statistics(count,file);

            List<Map.Entry<String, Integer>> list = MapSort.sortByValueIntegerDesc(count);

            output.delete();
            output.createNewFile();

            WriteFileUseOutputStream.output(list,output);

        }else{

//            ReadFileUseInputStream.statistics(count,file);
//            ReadFileUseReader.statistics(count,file);
            ReadFileUseScanner.statistics(count,file);

            List<Map.Entry<String, Integer>> list = MapSort.sortByValueIntegerDesc(count);

            output.delete();
            output.createNewFile();

//            WriteFileUseOutputStream.output(list,output);
//            WriteFileUsePrintWriter.output(list,output);
            WriteFileUseWriter.output(list,output);
        }

        System.out.println("统计成功");

    }

}
```

三、运行结果

（统计词数目过多，仅截图显示部分，查看全部请移步源码目录下查看`output.txt`）

```
the 1990
and 1321
i 1271
a 1262
of 1021
to 1005
he 767
in 708
was 673
it 555
that 504
you 489
at 451
his 422
she 376
with 362
her 339
had 331
on 311
me 284
for 265
gatsby 258
as 244
him 239
but 223
we 221
said 219
my 209
from 198
all 197
out 184
```