

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 _____ 2 班 _____
学 号 _____ 3018216095 _____
姓 名 _____ 郭紫珊 _____

Lab 2 Container and IO

1 实验目的

编写程序，统计了不起的盖茨比中各个单词出现的频次。

2 设计思路

要统计不同单词出现的频次，我的大体思路是这样的：

- 1) 从txt文件中读取输入流。
- 2) 创造 `HashMap<String,Integer>`，key值是单词，value是单次出现的次数。
- 3) 使用排序算法，对单词出现的次数进行排序。
- 4) 按照单词频次降序排列单词，并把输出流输出到文件中。

2.1 从txt文件中读取输入流，并储存在hashMap中

```
private static Map<String,Integer> readByFileReader(String filename) throws
Exception {
    Map<String,Integer> map = new HashMap<>();
    Reader reader = null;
    String artical = "";
    try {
        StringBuffer buf = new StringBuffer();
        char[] chars = new char[1024];
        reader = new FileReader(filename);
        int readed = reader.read(chars);
        while (readed != -1) {
            buf.append(chars, 0, readed);
            readed = reader.read(chars);
        }
        artical = buf.toString();
    } finally {
        close(reader);
    }
    StringTokenizer st = new StringTokenizer(artical.toString(),"
,?.!:\\\\\"'\"'\n#");
    while(st.hasMoreElements()) {
        String str = st.nextToken().toLowerCase();
        if(map.containsKey(str)) {
            Integer ex = map.get(str)+1;
            map.put(str, ex);
        }else {
            map.put(str,1);
        }
    }
    return map;
}
```

2.2 使用堆排序，排列单词频次

```

private static List<Entry<String,Integer>> sort(Map<String,Integer> record) {
    List<Entry<String,Integer>> list = new
ArrayList<Entry<String,Integer>>(record.entrySet());
    Collections.sort(list,new Comparator<Map.Entry<String,Integer>>() {
        public int compare(Entry<String, Integer> o1, Entry<String,
Integer> o2) {
            return o2.getValue().compareTo(o1.getValue());
        }
    });
    return list;
}

```

2.3 输出成文件

```

private static void output(List<Entry<String,Integer>> list, String
filename) throws IOException {
    File outputFile = new File(filename);
    if(!outputFile.exists()) {
        outputFile.createNewFile();
    }
    OutputStream os = new FileOutputStream(outputFile);
    StringBuffer temp = new StringBuffer();
    for (Entry<String, Integer> e: list) {
        temp.append(e.getKey() + " " + e.getValue()+"\n");
    }
    byte data[] = temp.toString().getBytes();
    os.write(data);
    os.close();
}

```