西南民族大学

**实验报告**

2019 ------2020 学年第 2 学期

课程名称：软件工程课程设计

学 院：计算机科学与技术

专业：计算机科学与技术

年级：2017级 班级：1702班

姓名：于晴 学号：201731102256

同组人：无

|  |
| --- |
| 西南民族大学学生实验报告  教学单位：计算机科学与技术 实验室名称：BS-226 实验时间：2020年3月21日  专业：计算机科学与技术 班级：1702班 姓名：于晴 学号：201731102256 |
| 实验项目名称：第四次实验 实验成绩： 教师签名：周绪川 |
| 1. 实验要求 2. 语言要求：C/C++，Java等 3. 实验报告内容：解决思路，流程，代码及结果测试； 4. 源码上传至个人git 5. 实验过程及结果   在实验三的基础上继续：  输入文件为yq\_in\_04.txt，输出文件yq\_out\_04.txt,但有如下要求：  每个省后面有一个总数;  输出省按总数从大到小排序；如果两个省总数一样，按拼音（字母）排序；  每个省内各市从大到小排序；如果两个市总数一样，按拼音（字母）排序；  解决思路：  首先要解决的是每个省的求和问题，我们可以选择用三个map来解决该问题，第一个map中存放省份和省份总数信息，第二个map存放每个市和市的数量信息，第三个map用来存放省份和市及数量的信息，我们把各部分信息分离开来就会比较容易做相关的操作。接下来要处理的是排序的问题，不仅省份的总数要排序，省内各个市也需要排序，我们用map的value对map  进行排序，用这种方法就可以进行排序操作： List list = new ArrayList(map.entrySet()); Collections.sort(list, new Comparator() {   @Override   public int compare(Entry e1, Entry e2) {     return e1.getValue().compareTo(e2.getValue());   } });  代码：  **import** java.io.\*; **import** java.util.\*;  **public class** Test {  **public static void** main(String[] args) **throws** IOException {   InputStreamReader reader = **new** InputStreamReader(**new** FileInputStream(args[0]), **"UTF-8"**);  BufferedReader br = **new** BufferedReader(reader);   OutputStreamWriter writer = **new** OutputStreamWriter(**new** FileOutputStream(args[1]), **"UTF-8"**);  BufferedWriter bw = **new** BufferedWriter(writer);   HashMap<String,Integer> pro = **new** HashMap<>();*//<省，每个省份总数>* HashMap<String,Integer> city = **new** HashMap<>();*//<市，每个市的数量>* HashMap<String,HashMap> total = **new** HashMap<>();*//<省，市和每个市的数量>* **int** n = 0;  String title=**""**;  String tmp=**""**;  **while**((tmp = br.readLine())!= **null**) {  **if** (tmp.contains(**"待明确地区"**))**continue**;  String head = tmp.substring(0,3);  String msg = tmp.substring(3).trim();  **if** (title.equals(head)!=**true**){  **if** (title!=**""**){  total.put(title,city);  pro.put(title,n);  n = 0;  city = **new** HashMap<>();  }  }  city.put(msg.substring(0,msg.indexOf(**'\t'**)),Integer.*parseInt*(msg.substring(msg.indexOf(**'\t'**)).trim()));  n += Integer.*parseInt*(msg.substring(msg.indexOf(**'\t'**)).trim());  title = head;  }  total.put(title,city);  pro.put(title,n);  List<Map.Entry<String,Integer>> list\_province = **new** ArrayList<>(pro.entrySet());  list\_province.sort(**new** Comparator<Map.Entry<String, Integer>>() {*//通过比较器按数量进行排序* @Override  **public int** compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {  **return** o2.getValue().compareTo(o1.getValue());  }  });   **for**(**int** i = 0 ;i < list\_province.size();i++){  bw.write(list\_province.get(i).getKey() + **"\t"** + list\_province.get(i).getValue() + **"\r\n"**);  bw.flush();  List<Map.Entry<String ,Integer>> list\_city = **new** ArrayList<>(total.get(list\_province.get(i).getKey()).entrySet());  list\_city.sort(**new** Comparator<Map.Entry<String, Integer>>() {  @Override  **public int** compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {  **return** o2.getValue().compareTo(o1.getValue());  }  });  **for**(**int** j = 0;j < list\_city.size();j++){  bw.write(list\_city.get(j).getKey() + **"\t"** + list\_city.get(j).getValue() + **"\r\n"**);  bw.flush();  }  bw.write(**"\r\n"**);  bw.flush();  }  reader.close();  br.close();  writer.close();  bw.close();  } }  实验结果：  1、首先将java文件打包成jar包，方便之后生成exe文件：  （1）点击File选项的Project Structure选项，如图1所示：    图1   1. 进入Artifacts选项，点击加号，新建一个Test2的jar，如图2所示：     图2   1. 返回编程界面，点击Build选项，选择build artifacts选项，在弹出的窗口中选择当前工程的jar包中的Build选项，如图3所示:     图3   1. 在相应文件夹可观察到已生成jar包，如图4所示：     图4   1. 生成exe可执行程序： 2. 打开exe4j应用程序，如图5所示：   1  图5   1. 输入在网上找到的序列号，注册成功可以使用，如图6所示：   1111  图6   1. 下一步后，选择第二个选项，即用jar来生成可执行的exe文件，如图7所示：   2  图7   1. 选择生成exe的文件夹，如图8所示：   3  图8   1. 选择生成exe的名字为yq.exe，如图9所示：   4  图9   1. 在class path部分点击绿色加号，选择刚才生成的jar包，如图10所示：   5  图10   1. 在Main class选择当前的主函数Test，如图11所示：   8  图11   1. 选择jdk执行的最低版本，我这里选择的是1.8，如图12所示：     图12   1. 点击Advances Options，选择search sequence,添加所需的环境变量等，如图13：     图13   1. 点击下一步，选择默认的VM，如图14所示：   10  图14   1. 继续点击下一步，直至软件完成所有步骤，如图15所示：   11  图15   1. 打开目标所在文件夹，已经生成yq.exe，如图16所示：     图16   1. 测试结果 2. 打开cmd，输入yq D:\SEProject\yq\_in\_04.txt D:\SEProject\yq\_out\_04.txt      1. 观察yq\_out\_04.txt已经输出经yq\_in\_04.txt处理后的结果，如图所示：     yq\_in\_04.txt:    个人git链接：https://github.com/YuQ23333/SE-project   1. 分析讨论   本次的实验也是在上周实验的基础上进行的，不过对我来说确实有很大的难度，但这正是锻炼我们编程能力，打好坚实基础的时机。一开始我在实验三代码的基础上进行改动，发现并不能达到实验的要求，在网上查阅资料后，感觉可以用map来解决问题。以前对Map的了解并不是太深入，本次实验让我对map的了解更加具体，也让我认识到了编程语言的无穷魅力。在网上查阅资料后经同学的帮助和自己的不懈努力下，我也终于得到了想要的结果，让我更加坚定了想要锻炼自己编程能力的内心，或许我还有很多不足，但是我相信通过过我的努力一定可以再接再厉。   1. 教师评阅 |

注：实验报告的内容及格式可由学院根据学科专业特点确定；全校各专业必须使用学校统一封面。