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：计算机科学与工程 专业：计科

年级：2019级 班级：1901

学号：201931101058 姓名：海福生

同组人：无

|  |
| --- |
| 西南民族大学学生实验报告  教学单位：计算机科学与工程学院 实验室名称：BS-222 实验时间：2022 年 3 月 6日  姓名：海福生 专业：计科 班级：1901 学号:201931101058 |
| 实验项目名称：SE实践1 实验成绩： 教师签名：周绪川 |
| **一、实验目的**  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：   1. 每个省后面有一个总数; 2. 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； 3. 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；   **二、材料与方法**  PC机一台、GitHub、idea、exe4j等  **三、实验主要过程与结果**  实验主要是在上次实验的基础上对每个省的数据进行排序并且输出结果，分别输出省和省内市的人数排序  首先我是在上次的基础上将市和人数存放在了hashmap中，在hash表中用value的大小来对hashmap进行排序并且输出，实现方法如下  然后在每个省所管辖的市的人数进行排序的时候将每个省的人数求和    求完和之后 再把省名和省所对应的人数存入hashmap 然后用同样的方法进行排序，排序完了之后开始输出结果 根据三个hashmap对应的NO 进行输出得到最后的结果    最后将输出的数据添加到stringBuilder中 并用输出流进行输出    Txt文件：      源代码：  package SETest;  import java.io.\*; import java.util.\*;  public class test3and4 {  // public static void TreeMapSortByValue(HashMap<String, Integer> map) { // // 将map.entrySet()转换成list // ArrayList<Map.Entry<String, Integer>> list1 = new ArrayList<Map.Entry<String, Integer>>(map.entrySet()); // // 通过比较器来实现排序 // list1.sort(new Comparator<Map.Entry<String, Integer>>() { // @Override // public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) { // // 升序排序 // Integer h1 = o1.getValue(); // Integer h2 = o2.getValue(); // return -Integer.compare(h1, h2);//o1.getValue().compareTo(o2.getValue()); // } // }); // for (Map.Entry<String, Integer> mapping : list1) // System.out.println(mapping.getKey() + " " + mapping.getValue()); // // // }   // 返回省份排名  public static List listSortByValue(HashMap<String, Integer> map) {  // 将map.entrySet()转换成list  List list2 = new ArrayList();  ArrayList<Map.Entry<String, Integer>> list1 = new ArrayList<Map.Entry<String, Integer>>(map.entrySet());  // 通过比较器来实现排序  list1.sort(new Comparator<Map.Entry<String, Integer>>() {  @Override  public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {  // 升序排序  Integer h1 = o1.getValue();  Integer h2 = o2.getValue();  return -Integer.*compare*(h1, h2);//o1.getValue().compareTo(o2.getValue());  }  });  for (Map.Entry<String, Integer> mapping : list1) {  //System.out.println(mapping.getKey() + ":" + mapping.getValue());  list2.add(mapping.getKey());  }   return list2;   }    public static int Sum(HashMap<String, Integer> map) {  ArrayList<Map.Entry<String, Integer>> list1 = new ArrayList<Map.Entry<String, Integer>>(map.entrySet());  // 通过比较器来实现排序  int sum = 0;  for (Map.Entry<String, Integer> mapping : list1) {  sum += mapping.getValue();  }  //System.out.println(sum);  return sum;  }    public static void main(String[] args) throws IOException {  String string = "";  int a = 0;  HashMap<Integer, HashMap<String, Integer>> Sites = new HashMap<>();  ArrayList<String> sb = new ArrayList<>();  ArrayList<String> coll = new ArrayList<>();  HashMap<Integer, String> Sites1 = new HashMap<>(); //存储省份名  try {  // create a reader instance  BufferedReader br = new BufferedReader(new FileReader("E:/软件工程/yq\_in.txt"));  // read until end of file  String line;  while ((line = br.readLine()) != null) {  HashMap SitTest = new HashMap();  SitTest.clear();  coll.clear(); //每次开始前清零coll  coll.add(String.*valueOf*(a));//给省份编号  // 遍历line 将数据分开储存到hashmap  String[] arr = line.split("\\s+", 3);  for (String ss : arr) {  coll.add(ss);  }   /\*  把arrayList里面的值转到hashMap 一个存储编号+ 省份 另一个存储编号+市＋人数  \*/   String NO1 = (String) ((ArrayList<?>) coll).get(0);  int NO = Integer.*valueOf*(NO1).intValue();  String sheng = (String) ((ArrayList<?>) coll).get(1);  String city = (String) ((ArrayList<?>) coll).get(2);  String num = (String) ((ArrayList<?>) coll).get(3);  int num1 = Integer.*valueOf*(num).intValue();  SitTest.put(city, num1);  sb.add(city + " " + num);  Sites1.put(NO, sheng);  Sites.put(NO, SitTest);  a++;  }  //System.out.println(Sites1);  //System.out.println(Sites);  br.close();  } catch (IOException ex) {  ex.printStackTrace();  }    List list = new ArrayList();  for (int i = 0; i < Sites.size(); i++) {  if (!list.contains(Sites1.get(i))) {  list.add(Sites1.get(i));  }  }  HashMap<String, Integer> mapSheng = new HashMap<String, Integer>();  HashMap<String, Integer> map = new HashMap<String, Integer>();  for (int i = 0; i < list.size(); i++) {  map.clear();  // System.out.println( list.get(i));    for (int j = 0; j < Sites1.size(); j++) {  if (Objects.*equals*(list.get(i), Sites1.get(j))) {  map.putAll(Sites.get(j));  // System.out.println("1" + Sites.get(j));  }   }  //System.out.println(map.values());  //TreeMapSortByValue(map);  int sum = *Sum*(map);  //System.out.println("该省份总人数"+sum);  mapSheng.put(list.get(i).toString(), sum);  //System.out.println();  }  //   //排省份  StringBuilder stringBuilder = new StringBuilder();  ArrayList list1 = new ArrayList();   list1 = (ArrayList) *listSortByValue*(mapSheng);   //System.out.println("sb"+list1);    for (int i = 0; i < Sites.size(); i++) {  if (!list1.contains(Sites1.get(i))) {  list1.add(Sites1.get(i));  }  }  HashMap<String, Integer> mapSheng1 = new HashMap<String, Integer>();  HashMap<String, Integer> map1 = new HashMap<String, Integer>();  for (int i = 0; i < list1.size(); i++) {  map1.clear();  System.*out*.println(list1.get(i));  stringBuilder.append(list1.get(i) + "\n");    for (int j = 0; j < Sites1.size(); j++) {  if (Objects.*equals*(list1.get(i), Sites1.get(j))) {  map1.putAll(Sites.get(j));  // System.out.println("1" + Sites.get(j));  }   }  //System.out.println(map.values());  //TreeMapSortByValue(map1);   ArrayList<Map.Entry<String, Integer>> list10 = new ArrayList<Map.Entry<String, Integer>>(map1.entrySet());  // 通过比较器来实现排序  list10.sort(new Comparator<Map.Entry<String, Integer>>() {  @Override  public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {  // 升序排序  Integer h1 = o1.getValue();  Integer h2 = o2.getValue();  return -Integer.*compare*(h1, h2);//o1.getValue().compareTo(o2.getValue());  }  });  for (Map.Entry<String, Integer> mapping : list10) {  System.*out*.println(mapping.getKey() + " " + mapping.getValue());  stringBuilder.append(mapping.getKey() + " " + mapping.getValue() + "\n");  }    int sum1 = *Sum*(map1);  System.*out*.println("该省份总人数" + sum1);  stringBuilder.append("该省份总人数" + sum1 + "\n");  mapSheng1.put(list1.get(i).toString(), sum1);  System.*out*.println();  stringBuilder.append("\n");   //   }  System.*out*.println("这是输出" + stringBuilder);    String content = stringBuilder.toString();  File file = new File("E:/软件工程/yq\_out35555.txt");  if (!file.exists()) {  file.createNewFile();  }  FileWriter fileWriter = new FileWriter(file.getAbsoluteFile());  BufferedWriter bw = new BufferedWriter(fileWriter);  bw.write(content);  bw.close();  System.*out*.println("finish");   } }  **四、分析讨论**  1．对java集合的熟悉程度提高了，但应用起来不是很灵活，需要个人加强编程练习次数。  2.在处理问题时没有头绪就开始上手，导致最后 虽然实现了功能但是代码逻辑欠佳、可读性差、还需要大量的优化。  3.对于排序一开始想的是用类和对象的方法来解决，但写的时候发现需要实例化很多个重复的对象，具体的功能实现也欠佳。  。  **五、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。