西南民族大学

**实验报告**

2021 ------2022 学年第 2 学期

课程名称：软件工程

学 院：计算机科学与工程学院

专 业：计科

年级：2019 班级：计科1902

学 号：201931101055 姓 名：古丽孜巴

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-223 实验时间：2022 年 3 月15 日  姓名：古丽孜巴 专业：软件工程 班级：计科1902 学号:201931101055 |
| 实验项目名称：实验四 实验成绩：A 教师签名：周绪川 |
| 实验项目报告内容（1、实验背景（目的、意义及原理等）；2、材料与方法；3、实验主要过程与结果；  4、分析讨论；5、教师评阅）。   1. 实验目的：   1.  在实验二和实验三的基础上继续：  1、输入文件名在命令行输入（即输入文件可以指定）；输出文件名在命令行输出（即输出文件可以指定）；  2、可以输出指定省的信息  具体示例：  >yq yq\_in\_03.txt yq\_out\_03.txt  //表示输入文件名为yq\_in\_03.txt，输出文件名为yq\_out\_03.txt，包括所有省份信息，格式和实验二一致  >yq yq\_in\_03.txt yq\_out\_03.txt 浙江省  //输出文件中仅包括浙江省的信息  2．   1. 要求： 2. 每个省后面有一个总数 3. 输出省按总数从大到小排序，如果两个省总数一样，按照拼音排序 4. 每个省内各市从大到小排序；如果两个市总数一样，按拼音排序 5. 实验环境：   Jdk1.8.0   1. 实验三内容及结果： |
| 源代码：  **package** ss;  **import** java.io.\*;  **import** java.util.\*;  **import** java.util.stream.Collectors;  **public** **class** Test {  **public** **static** List<String> *sf* = **new** ArrayList<>();  **public** **static** **void** main(String[] args) **throws** IOException {  List<String> sf1 = **new** ArrayList<>();  Scanner s = **new** Scanner(System.***in***);  String path = "C:\\Users\\admin\\Desktop\\%s";  String in = s.nextLine();  String[] ss = in.split(" ");  String name = "";  **if** (ss.length == 3){  name = ss[2];  }  List<String> outList = **new** ArrayList<>();  List<String> list = *toArrayByFileReader*(String.*format*(path,ss[0]));  **if** (**null** != name && !"".equals(name) ){  //查询  String finalName = name;  List<String> finalOutList = outList;  list.forEach(e -> {  **if** (e.contains(finalName)){  finalOutList.add(e);  }  });  String finalName1 = name;  sf1 = *sf*.stream().filter(e -> e.equals(finalName1)).collect(Collectors.*toList*());  }**else**{  sf1 = *sf*;  outList = list;  }  LinkedHashSet h = **new** LinkedHashSet(sf1);  sf1.clear();  sf1.addAll(h);  *writeFile*(sf1, outList, String.*format*(path,ss[1]));  }  **public** **static** **void** writeFile(List<String> sf2, List<String> outList, String name) **throws** IOException {  PrintWriter pw = **new** PrintWriter(**new** FileWriter(name));  **if** (**null** != outList && outList.size() > 0){  sf2.forEach(e -> {  pw.write(e);  pw.write("\n");  outList.forEach(e2 ->{  **if** (e2.contains(e)) {  pw.write(e2.replace(e, ""));  pw.write("\n");  }  });  });  }  pw.close();  }  **public** **static** List<String> toArrayByFileReader(String name) {  // 使用ArrayList来存储每行读取到的字符串  ArrayList<String> arrayList = **new** ArrayList<>();  **try** {  FileReader fr = **new** FileReader(name);  BufferedReader bf = **new** BufferedReader(fr);  String str;  // 按行读取字符串  **while** ((str = bf.readLine()) != **null**) {  arrayList.add(str);  *sf*.add(str.substring(0,3));  }  bf.close();  fr.close();  } **catch** (IOException e) {  e.printStackTrace();  }  // 返回数组  **return** arrayList;  }  }  实现过程：    输入 输入文件名和输出文件名，自动创造一个名为XX的txt文件：  文件内容为处理实验二的处理文档： |
| 若输出文件名后面有省份：    生成一个名为ll的TXT文件：    文件内容为只有贵州省的内容：     1. 实验内容及结果：   源代码：package ss;  import java.io.\*;  import java.util.\*;  import java.util.stream.Collectors;  public class Test {  // 文件数据  public static List<Data> data = new ArrayList<>();  // 省份数据  public static Set<String> sf = new HashSet<>();  public static void main(String[] args) throws IOException {  List<String> sf1 = new ArrayList<>();  Scanner s = new Scanner(System.in);  String path = "C:\\Users\\admin\\Desktop\\%s";  String in = s.nextLine();  String[] ss = in.split(" ");  String name = "";  if (ss.length == 3){  name = ss[2];  }  toArrayByFileReader(String.format(path,ss[0]));  List<String> dataList = getData(getSfMap(name));  writeFile(dataList, String.format(path,ss[1]));  }  // 一行一行输出字符串到文件  public static void writeFile(List<String> outList, String name) throws IOException {  PrintWriter pw = new PrintWriter(new FileWriter(name));  if (null != outList && outList.size() > 0){  outList.forEach(e ->{  pw.write(e);  pw.write("\n");  });  }  pw.close();  }  // 一行一行读取TXT文档信息  public static void toArrayByFileReader(String name) {  // 使用ArrayList来存储每行读取到的字符串  ArrayList<String> arrayList = new ArrayList<>();  try {  FileReader fr = new FileReader(name);  BufferedReader bf = new BufferedReader(fr);  String str;  // 按行读取字符串  while ((str = bf.readLine()) != null) {  arrayList.add(str);  String[] ss = str.split("\t");  data.add(new Data(ss[0],ss[1],Integer.parseInt(ss[2])));  sf.add(ss[0]);  }  bf.close();  fr.close();  } catch (IOException e) {  e.printStackTrace();  }  }  // 获取省份排序后的list  public static Map<String,Integer> getSfMap(String name){  Map<String,Integer> sfMap = new HashMap<>();  Map<String,Integer> sfDataMap = new LinkedHashMap<>();  // 获取省份的城市值只和组成Map  sf.forEach(e->{  int num = data.stream().filter(data1 -> data1.getSf().equals(e)).mapToInt(Data::getNum).sum();  sfMap.put(e,num);  });  if (null == name || "".equals(name)){  // 对Map中所有省份按照值大小排序，返回排序后字符串  sfMap.entrySet().stream()  .sorted((e1, e2) -> {  int i = e1.getValue() - e2.getValue();  if (i > 0){  return -1;  }else if (i == 0){  return e1.getKey().compareTo(e2.getKey());  }else {  return 1;  }  })  .forEachOrdered(x -> sfDataMap.put(x.getKey(), x.getValue()));  } else {  // 根据名字查询指定身份  sfMap.entrySet().stream().filter(e->e.getKey().equals(name)).forEachOrdered(x -> sfDataMap.put(x.getKey(), x.getValue()));  }  return sfDataMap;  }  // 获取输出字符串List  public static List<String> getData(Map<String,Integer> map){  // 返回最后要输出的字符串List  List<String> dataList = new ArrayList<>();  // 根据省份List对省下城市进行排序  map.forEach((key, value) -> {  dataList.add(key + "\t" + value);  // 根据省份获取当前省份下所有城市的 名称和值 组成Map  Map<String, Integer> csMap = data.stream().filter(data1 -> data1.getSf().equals(key)).collect(Collectors.toMap(Data::getCs, Data::getNum));  // 对Map按照值大小排序  List<String> data = csMap.entrySet().stream().sorted((e1, e2) -> {  int i = e1.getValue() - e2.getValue();  if (i > 0) {  return -1;  } else if (i == 0) {  return e1.getKey().compareTo(e2.getKey());  } else {  return 1;  }  }).map(n -> "\t" + n.getKey() + "\t" + n.getValue()).collect(Collectors.toList());  dataList.addAll(data);  });  return dataList;  }  // 数据实体类  static class Data {  private String sf;  private String cs;  private Integer num;  public Data(String sf, String cs, Integer num) {  this.sf = sf;  this.cs = cs;  this.num = num;  }  public String getSf() {  return sf;  }  public void setSf(String sf) {  this.sf = sf;  }  public String getCs() {  return cs;  }  public void setCs(String cs) {  this.cs = cs;  }  public Integer getNum() {  return num;  }  public void setNum(Integer num) {  this.num = num;  }  }  }  运行截图：    生成相应的文件：       1. 分析讨论   这次实验让我明白了如何写一个有层次，有逻辑的代码，本次实验发现并之前的基础上想做本次实验非常困难，所以在别的帮助下顺利的完成了本次实验。  通过本次实验需要注意输入文件名时需要注意格式的正确性，以及访问文件被拒绝，读文件出现乱码时的解决办法，编译运行时JDK需要1.8.0以上，不然可能不兼容。   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。