西南民族大学

**实验报告**

**2021------2022**学年第**1**学期

课程名称：软件工程

学 院：计算机科学与工程 专 业：软件工程

年 级：2019级 班 级：1902

学 号：201931101351 姓 名：周研

同组人：马林泽 201931101155

|  |
| --- |
| 西南民族大学学生实验报告  教学单位：计科学院 实验室名称：BS-222 实验时间：2021 年 10 月 20 日  姓名：周研、马林泽 专业：软件工程 班级：软工1902 学号:201931101351、201931101155 |
| 实验项目名称：SE实践 3 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、IntelliJ IDEA 2021.1.2 x64  **三、实验主要过程与结果**  1.配置实验环境：  （1）开发环境  内存8G。  （2）IntelliJ IDEA 2021.1.2 x64  2. 解决思路：  之前的三次实验实现了三个功能，单独运行yq.java，给出缺省输入文件的输出；运行yq[指定输入文件][指定输出文件]；运行yq[指定输入文件][指定省份]。  这次是在之前实验的基础上要求①每个省后有一个总数；②输出省按总数从大到小排序，若两个省总数一样，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。  首先要做的是将我们的这个程序的功能封装为独立模块然后设计单元测试，不同的代码解决不同层面的问题。加载文件并读取函数loadFile()；读取文件之后要处理文件内容设计函数operateFile()；因为要求输出每个省各市病例数按从大到小排序，设计sortCountry()函数来为各个市排序；统计一个省的总量设计函数 allYQInAProvince()；各个省按病例数从大到小排序sortProvince()；排序解决完还有病例数相同的按拼音（字母）排序，设计函数sortByPY()；然后还有输出文件函数outputFile()；这些功能都是一个独立的模块，可以在命令行测试程序使用，在单元测试框架下使用。  3.源代码：  import java.io.\*;  import java.util.ArrayList;  import java.util.List;  import java.util.Scanner;  public class yq {  static String parameter = null;  static String[] parameters = null;  static int parameterLength = 0;  static int provinceNumber = 0;  //加载文件  public static String loadFile(String[] parameters)throws Exception{  FileInputStream fip = new FileInputStream(new File("src/"+parameters[0]));//文件输入流  InputStreamReader reader = new InputStreamReader(fip, "GBK");  StringBuffer sb1 = new StringBuffer();//读取文件  while (reader.ready()) sb1.append((char) reader.read());  String str = sb1.toString();  return str;  }  //处理文件  public static String[][][] operateFile(String str){  String[] str1 = str.split("\\s+");  int str1length = str1.length;  for (int i = 0;i < str1length;i++) {//处理待明确地区  i++;  if(str1[i].equals("待明确地区")){  str1[i-1] = "\*";  str1[i] = "\*";  i++;  str1[i] = "\*";  }  else i++;  }  List<String> tmp = new ArrayList<String>();  for(String item:str1){ //循环数据  if(item!="\*"){ //把不为\*的字符串存放到list里（去掉了待明确地区）  tmp.add(item);  }  }  str1 = tmp.toArray(new String[0]);  str1length = str1.length;  String province = "";  provinceNumber = 0;  for (int i = 0;i < str1length;i++) {//统计省数  if(str1[i].equals(province)) i++;  else {  province = str1[i];  provinceNumber++;  i++;;  }  i++;  }  province = "阿里嘎多";  int provinceIndex = -1;  int countryIndex = -1;  String data[][][] = new String[provinceNumber][30][2];  for (int i = 0;i < str1length;i++) {//将文件数据封装到三维数组  if(str1[i].equals(province)){  i++;  }  else{  province = str1[i];  countryIndex =0;  provinceIndex++;  data[provinceIndex][countryIndex][0] = str1[i];  data[provinceIndex][countryIndex][1] = "0";  i++;  }  countryIndex++;  data[provinceIndex][countryIndex][0] = str1[i];  i++;  data[provinceIndex][countryIndex][1] = str1[i];  }  for(int i=0;i<data.length;i++) {//遍历三维数组处理null值为0，便于后序排序  for(int j=0;j<data[i].length;j++) {  for(int k=0;k<data[i][j].length;k++)  if(data[i][j][k] == null) data[i][j][k] = "0";  }  }  return data;  }  //各个市按病例数排序  public static String[][][] sortCountry(String[][][] data){  int provinceIndex = 0;  provinceNumber = data.length;  while(provinceIndex < provinceNumber) {//市排序  for (int i = 1; i < 30; i++) {  for (int j = 1; j < 30 - i; j++) {  if (Integer.parseInt(data[provinceIndex][j][1]) < Integer.parseInt(data[provinceIndex][j + 1][1])) {  String temp1, temp2;  temp1 = data[provinceIndex][j][0];  temp2 = data[provinceIndex][j][1];  data[provinceIndex][j][0] = data[provinceIndex][j + 1][0];  data[provinceIndex][j][1] = data[provinceIndex][j + 1][1];  data[provinceIndex][j + 1][0] = temp1;  data[provinceIndex][j + 1][1] = temp2;  }  }  }  provinceIndex++;  }  return data;  }  //统计省的总数  public static String[][][] allYQInAProvince(String[][][] data){  int provinceIndex = 0;  while(provinceIndex < provinceNumber) {//计算省病例数  for (int i = 1; i < 30; i++) {  int add1 = Integer.parseInt(data[provinceIndex][0][1]);  int add2 = Integer.parseInt(data[provinceIndex][i][1]);  add1 += add2;  data[provinceIndex][0][1] = add1+"";  }  provinceIndex++;  }  return data;  }  //各个省按病例数排序  public static String[][][] sortProvince(String[][][] data){  provinceNumber = data.length;  for (int i = 0;i < provinceNumber-1;i++){//省排序  for (int j = 0;j<provinceNumber-1-i;j++)  {  if(Integer.parseInt(data[j][0][1])<Integer.parseInt(data[j+1][0][1])){  for (int p = 0; p < 30;p++){  String temp1, temp2;  temp1 = data[j][p][0];  temp2 = data[j][p][1];  data[j][p][0] = data[j+1][p][0];  data[j][p][1] = data[j+1][p][1];  data[j+1][p][0] = temp1;  data[j+1][p][1] = temp2;  }  }  }  }  return data;  }  //病例数相同的按拼音字母顺序排序  public static String[][][] sortByPY(String[][][] data){  int provinceIndex = 0;  provinceNumber = data.length;  while(provinceIndex < provinceNumber) {//市排序  for (int i = 1; i < 30; i++) {  for (int j = 1; j < 30 - i; j++) {  if(data[provinceIndex][j][0] != "0"){  if (Integer.parseInt(data[provinceIndex][j][1]) == Integer.parseInt(data[provinceIndex][j + 1][1])) {  if(data[provinceIndex][j][0].compareTo(data[provinceIndex][j + 1][0])>0)  {  String temp1, temp2;  temp1 = data[provinceIndex][j][0];  temp2 = data[provinceIndex][j][1];  data[provinceIndex][j][0] = data[provinceIndex][j + 1][0];  data[provinceIndex][j][1] = data[provinceIndex][j + 1][1];  data[provinceIndex][j + 1][0] = temp1;  data[provinceIndex][j + 1][1] = temp2;  }  }  }  }  }  provinceIndex++;  }  return data;  }  //输出文件  public static void outputFile(String[][][] data,String[] parameters,int parameterLength)throws Exception{  FileOutputStream fop = new FileOutputStream(new File("src/"+parameters[1]));//文件输出在src文件夹下  OutputStreamWriter writer = new OutputStreamWriter(fop, "GBK");  if(parameterLength == 2)  {  for (int i = 0;i < provinceNumber;i++)  {  for (int j = 0;j < 30;j++){  if (data[i][j][0] != "0")  {  writer.append(data[i][j][0]+"\t");  writer.append(data[i][j][1]+"\n");  }  }  writer.append("\n");  }  }  if(parameterLength == 3)  {  for (int i = 0;i < provinceNumber;i++)  {  if (data[i][0][0].equals(parameters[2])) {  for (int j = 0; j < 30; j++) {  if (data[i][j][0] != "0") {  writer.append(data[i][j][0] + "\t");  writer.append(data[i][j][1] + "\n");  }  }  writer.append("\n");  }  }  }  writer.close(); // 关闭写入流,同时会把缓冲区内容写入文件  fop.close(); // 关闭输出流,释放系统资源  }  //main函数  public static void main(String[] args) throws Exception {  Scanner scanner = new Scanner(System.in);  System.out.println("按顺序输入参数，使用空格分隔（输入文件名 输出文件名 指定省（可选））");  parameter = scanner.nextLine();  scanner.close();  parameters = parameter.split("\\s+");//分割参数存入参数数组  parameterLength = parameters.length;  String str = loadFile(parameters);  String[][][] data = operateFile(str);  data = sortCountry(data);  data = allYQInAProvince(data);  data = sortProvince(data);  data = sortByPY(data);  outputFile(data,parameters,parameterLength);  }  }  4.结果测试：    图1.1 单元测试框架下测试程序    图1.2 输出省从大到小排序    图1.3 指定输出某个省按病例数从大到小排序  5.个人git链接：  https://github.com/zy-abdl/learngit  https://github.com/18095305783/Panda.git  **四、分析讨论**  这次的实验相较于之前的实验难度比较大，通过对每个代码的整合，将它们封装成独立模块便于后期的维护扩展，这也是软件工程这门课我们要学会的思想。所以对之前的代码的组织结构要精心的整理和优化，在设计每个函数的时候确实比较麻烦，用Java没有python便捷，部分功能之前实验课设计过所以并没有耗费太多的精力，这次的难点主要集中在输出省的总数以及各个市的病例数要按从大到小的顺序排序输出，病例数相同的按拼音（字母）顺序排序，对于Java来说很不友好，在找寻相关资料学习后，最终完成实验。通过这次的实验不仅加强了代码的编写能力，同时也学习到了软件开发中的一些思想，这也是软件工程这门课我们要学习的，受益匪浅。  **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。