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：计科学院 专 业：计算机科学与技术

年 级：2017级 班 级：1702

学 号：201731102140 姓 名：胡倩宇

|  |
| --- |
| 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：BS226 实验时间：2020年3月22日 |
| 实验项目名称：实验4 实验成绩： 教师签名：周绪川 |
| GitHub的地址<https://github.com/huqianyu/software-engineering.git>  **一、实验内容**  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、实验主要过程与结果**  1.思路：在输出文件前，对数据进行排序，先进行省排序，再进行市排序。根据数量冒泡排，如果相等，拼音排序。拼音排需要引入外部jar包，排序完了按顺序输出文件。  2.代码  package main.java.work;  import java.io.\*;  import java.util.ArrayList;  import java.util.Collections;  import java.util.List;  import java.util.Scanner;  public class Main {  private static List<Province> provinces=new ArrayList<>(); //存放读取文件的数据  private static List<Province> sortprovinces=new ArrayList<>(); //存放排序后的数据  public static void main(String[] args) {  Scanner s=new Scanner(System.in); //控制台输入  System.out.println("请输入命令");  while(s.hasNext()) {  String command = s.nextLine(); //取出控制台输入的字符串  String[] commands = command.split("\\s+"); //将输入字符串以空格形式分割  if(commands.length<2) {  System.out.println("您的输入命令有误，请输入yq 格式的命令");  System.out.println("请输入命令");  continue;  }else if(commands.length==2){  if(!commands[0].equals("yq")) {  System.out.println("您的输入命令有误，请输入yq 格式的命令");  System.out.println("请输入命令");  continue;  }else { //输入形式为 yq 输入文件名  int result=readFile(commands[1]); //读取文件  if(result==1) {  System.out.println("文件读取失败，请检查您的文件是否存在，以及文件格式是否是UTF-8");  System.out.println("请输入命令");  continue;  }  //打印输出  printFileData();  //将存放数据清空，避免下次再次读取文件冲突  provinces=new ArrayList<>();  sortprovinces=new ArrayList<>();  System.out.println("请输入命令");  continue;  }  }else if(commands.length==3){ //输入形式为 yq 输入文件名 输出文件名  if(!commands[0].equals("yq")) {  System.out.println("您的输入命令有误，请输入yq 格式的命令");  System.out.println("请输入命令");  continue;  }else {  int result=readFile(commands[1]);  if(result==1) {  System.out.println("您的输入命令有误，请输入yq 格式的命令");  System.out.println("请输入命令");  continue;  }  //排序数据  sortData("0");  wirteFile(commands[2]);  System.out.println("输出文件成功");  //将存放数据清空，避免下次再次读取文件冲突  provinces=new ArrayList<>();  sortprovinces=new ArrayList<>();  System.out.println("请输入命令");  continue;  }  }else if(commands.length==4){ //输入形式为 yq 输入文件名 输出文件名 筛选项  if(!commands[0].equals("yq")) {  System.out.println("您的输入命令有误，请输入yq 格式的命令");  System.out.println("请输入命令");  continue;  }else {  int result=readFile(commands[1]);  if(result==1) {  System.out.println("您的输入命令有误，请输入yq 格式的命令");  System.out.println("请输入命令");  continue;  }  sortData(commands[3]); //排序，加入筛选项参数  wirteFile(commands[2]);  System.out.println("输出文件成功");  provinces=new ArrayList<>();  sortprovinces=new ArrayList<>();  System.out.println("请输入命令");  continue;  }  }  }  }  //读取文件，并存放数据到provinces  public static int readFile(String filepath) {  try {  //输入流  InputStreamReader isr = new InputStreamReader(new FileInputStream("D:\\"+filepath),"UTF-8");  BufferedReader br = new BufferedReader(isr);  String s = null;  while ((s = br.readLine()) != null) {  String[] lineitems = s.split("\\s+");  if (lineitems.length < 3) {  continue;  }  //获取到 如 广东省 广州 25 的数据后，调用addData存放数据  City city = new City();  city.setP\_name(lineitems[0]);  city.setName(lineitems[1]);  city.setSl(Long.valueOf(lineitems[2]));  addData(city);  }  br.close(); //关闭流  } catch (Exception e) {  return 1;  }  return 0;  }  //存放数据  public static void addData(City city) {  //如该城市的省份未创建,增加省份数据  boolean isHaveP=false;  String addp\_cityname=null; //是否是创建省份数据的城市标识,用于下文数量计算判断  for(Province p : provinces) {  if(p.getName().equals(city.getP\_name())) {  isHaveP=true;  break;  }  }  if(!isHaveP) {  addp\_cityname=city.getName();  Province province=new Province();  province.setName(city.getP\_name());  province.setSl(0);  List<City> cities=new ArrayList<>();  cities.add(city);  province.setCityList(cities);  provinces.add(province);  }  //增加城市数据，如果该城市已存在，则数量增加，否则新增  for(Province p:provinces) {  if(city.getP\_name().equals(p.getName())) {  boolean isHaveC=false;  for(City city2:p.getCityList()) {  if(city2.getName().equals(city.getName())) {  isHaveC=true;  p.setSl(p.getSl()+city.getSl());  //如果是创建省份数据的城市标识,则不进行数量增加  if(addp\_cityname !=null && !addp\_cityname.equals(city2.getName())) {  city2.setSl(city2.getSl()+city.getSl());  }  return;  }  }  if(!isHaveC) {  p.getCityList().add(city);  p.setSl(p.getSl()+city.getSl());  return;  }  }  }  }  //排序  public static void sortData(String query) {  //如果查询条件为0，排序所有数据。否则排序跟查询条件相同的数据。  if(!query.equals("0")) {  for(int i=0;i<provinces.size();i++) {  if(provinces.get(i).getName().equals(query)) {  sortprovinces.add(provinces.get(i));  }  }  }else {  sortprovinces=provinces;  }  //省份排序  for(int i=0;i<sortprovinces.size();i++) {  for(int j=i+1;j<sortprovinces.size();j++) {  if(sortprovinces.get(i).getSl()<sortprovinces.get(j).getSl()) {  Collections.swap(sortprovinces, i, j);  }  }  }  //城市排序  PinyinComparator pinyinComparator=new PinyinComparator();  for(Province p:sortprovinces) {  List<City> cities=p.getCityList();  for(int i=0;i<cities.size();i++) {  for(int j=i+1;j<cities.size();j++) {  if(cities.get(i).getSl()<cities.get(j).getSl()) {  Collections.swap(cities, i, j);  }else if(cities.get(i).getSl()==cities.get(j).getSl()) {  int py\_sort=pinyinComparator.compare(cities.get(i), cities.get(j));  if(py\_sort>0) {  Collections.swap(cities, i, j);  }  }  }  }  }  }  // 打印  public static void printFileData() {  for (int i = 0; i < provinces.size(); i++) {  String toppm = provinces.get(i).getName();  System.out.println(toppm);  for (City city : provinces.get(i).getCityList()) {  String item = city.getName() + " " + city.getSl();  System.out.println(item);  }  System.out.println();  }  }  //输出文件  public static void wirteFile(String outfilename) {  try {  FileWriter fileWriter = new FileWriter("D:\\"+outfilename);  for(int i=0;i<sortprovinces.size();i++) {  String toppm="排名:"+(i+1)+" 省份:"+sortprovinces.get(i).getName()+" 数量:"+sortprovinces.get(i).getSl();  fileWriter.write(toppm);  fileWriter.write("\r\n");  for(City city:sortprovinces.get(i).getCityList()) {  String item=city.getName()+" "+city.getSl();  fileWriter.write(item);  fileWriter.write("\r\n");  }  fileWriter.write("\r\n");  }  fileWriter.flush();  } catch (IOException e) {  e.printStackTrace();  }  }  }  3.运行结果    图1 运行结果①    图2 运行结果②    图3 运行结果③    图4 运行结果④  **三、分析讨论**  在本次实验中，对数据的处理有些困难。通过在网上的例子和同学的讲解才把本次实验做出来。希望在后面的实验能将知识掌握好。 |

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。