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

学 号：201731102245

姓 名：许芷毓

|  |
| --- |
| 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19日  姓名：许芷毓 专业：计算机科学与技术 班级：1702班 学号：201731102245 |
| 实验项目名称：第四次实验 实验成绩： 教师签名：周绪川 |
| **一、实验目的**  1.学习编写程序来对数据文件处理；  2.熟悉Java的文件读写机制；  3.练习输入输出流的使用，掌握磁盘文件的输入输出方法；  4.学习利用Eclipse生成Java可执行程序exe。  **二、材料与方法**  1.语言要求：C/C++，Java  2.IDE：Eclipse  3.辅助工具：Git  4.实验要求：  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；  **三、实验主要过程与结果**  （一）分析解决思路  本次实验，我选择用Java来实现对数据的处理。由于依旧需要实现在命令行中可以指定输入文件和输出文件，因此在实验三基础上，本次实验需要实现以下几个功能：①将每个省份的人数进行汇总②将省份和城市进行排序，排序的优先原则为数量优先级最高，其次是省份或者城市拼音的优先级。因为本次要实现数据的灵活处理，所以使用的方法是数组储存优先级顺序信息和分割省份之间的城市的标记信息。  （二）实验流程  1.新建工程  ①创建一个新的Java工程Test2，如图1-图4所示。    图1    图2    图3    图4 新建的Java工程Test2  ②在Test2工程下面新建一个package包com.test，如图5-图6所示。    图5    图6 新建的包com.test  ③在package包com.test下新建一个Java class文件，如图7-图8所示。    图7    图8 新建的Test4.class  3.编写Java程序处理数据文件yq\_in\_04.txt，如图9所示。      图9 编写程序  4.Java程序打包成exe可执行文件  （1）将Java程序通过Eclipse导成Jar包  ①选择项目Test4.java“右击”，点击“Export”，如图10所示。    图10  ②选择Java下边的JAR file，如图11所示。    图11  ③设置要导出文件的保存路径，如图12所示。    图12 设置要导出文件的保存路径  ④设置项目的入口函数，生成相应的jar文件，如图13-图14所示。    图13 设置项目的入口函数    图14 生成的Test4.jar  （2）通过exe4j将Jar包程序生成exe可执行文件  ①下载exe4j并安装，如图15-图20所示。    图15    图16    图17 选择安装路径    图18    图19    图20 开始安装  ②打开exe4j程序，添加注册码，如图21-图22所示。    图21 打开exe4j程序    图22 添加注册码  ③选择“JAR in EXE”mode，如图23所示。    图23 选择“JAR in EXE”mode  ④设置应用信息和转换后exe文件的保存路径，如图24所示。    图24 设置应用程序配置  ⑤设置转换成的exe文件的名称，如图25所示。    图25 设置转换成的exe文件的名称  ⑥添加刚才生成的jar文件，如图26-图27所示。    图26    图27 添加刚才生成的Test4.jar  ⑦选择项目的入口（主类），设置JRE最低版本和最高版本，以及在配置可执行exe文件的过程中需要配置生成64位的，如图28-图30所示。    图28    图29 设置最小JRE版本    图30 配置生成64位的exe  ⑧直到有提示exe4j has finished...，点击“click here to start the application”便可启动生成的exe文件，如图31所示。    图31  ⑨在之前选择的输出保存路径便可找到打包好的exe文件Test4.exe，如图32所示。    图32 可执行程序Test4.exe  （三）程序代码  package com.test;  import java.text.Collator;  import java.util.\*;  import java.io.BufferedReader;  import java.io.File;  import java.io.FileInputStream;  import java.io.FileOutputStream;  import java.io.IOException;  import java.io.InputStreamReader;  import java.io.OutputStreamWriter;  public class Test4 {  static String[][] rank = new String[9][3];  static String[][] pro = new String[9][2];  static Map<Object,Object> dic = new HashMap<Object,Object>();  static Comparator<Object> com = Collator.getInstance(java.util.Locale.CHINA);  static List<String> list = new ArrayList<>(); //创建一个列表对象list  //按行读取文件  public static void readFileByLines(String filename) throws IOException {  File file = new File(filename);  BufferedReader reader = null;  InputStreamReader inputStreamReader = new InputStreamReader(new FileInputStream  (file), "GBK");  reader = new BufferedReader(inputStreamReader);  String tempString;  while((tempString = reader.readLine())!= null) {  list.add(tempString);  }  inputStreamReader.close();  reader.close();  }  //省份排序  public static void change(String[][] str) {  String[][] exc = new String[str.length][];  for(int j=0 ; j<str.length ; j++) {  for(int k=j+1 ; k<str.length ; k++) {  if(Integer.valueOf(str[j][1]) > Integer.valueOf(str[k][1]))  continue;  else if(Integer.valueOf(str[j][1]) < Integer.valueOf(str[k][1])) {  exc[j] = str[k];  str[k] = str[j];  str[j] = exc[j];  }  else {  if(com.compare(str[k][0],str[j][0])>=0)  continue;  else {  exc[j] = str[k];  str[k] = str[j];  str[j] = exc[j];  }  }  }  }  }  //城市排序  public static void exchange(String[][] str,int a,int b) {  String[][] exc = new String[str.length][];  for(int j=a ; j<b-1 ; j++) {  for(int k=j+1 ; k<b-1 ; k++) {  if(Integer.valueOf(str[j][2])>Integer.valueOf(str[k][2]))  continue;  else if(Integer.valueOf(str[j][2])<Integer.valueOf(str[k][2])) {  exc[j] = str[k];  str[k] = str[j];  str[j] = exc[j];  }  else {  if(com.compare(str[k][1],str[j][1])>=0) {  continue;  }  else {  exc[j] = str[k];  str[k] = str[j];  str[j] = exc[j];  }  }  }  }  }  public static void rank(String[][] str) {  String s = str[0][0];  int count = 0;  rank[count][1] = String.valueOf(0);  for(int b=0 ; b<str.length ; b++) {  if(str[b][0].equals(s)) {  continue;  }  else {  rank[count][0] = str[b-1][0];  rank[count][2] = String.valueOf(b-1);  count = count + 1;  rank[count][1] = String.valueOf(b);  s = str[b][0];  }  }  rank[count][0] = str[str.length - 1][0];  rank[count][2] = String.valueOf(str.length-1);  }  public static void main(String[] args) throws IOException {  readFileByLines(args[0]);  System.out.println(list);  String[][] string = new String[list.size()][];  int sum = 0;  int begin;  int over;  for(int i=0 ; i<list.size() ; i++) {  string[i] = list.get(i).split("\\s+");  }  String loc = string[0][0];  //统计省份和总数  for(int r=0 ; r<string.length ; r++) {  if(string[r][0].equals(loc)) {  sum += Integer.valueOf(string[r][2]);  }  else {  dic.put(string[r-1][0],sum);  sum = Integer.valueOf(string[r][2]);  loc = string[r][0];  }  }  int count = 0;  dic.put(string[string.length-1][0], sum);  for(Object key :dic.keySet()) {  pro[count][0] = key.toString();  pro[count][1] = dic.get(key).toString();  count ++;  }  change(pro);//省份排序顺序列出  System.out.println("将省份按总人数由多到少排序：");  for(String[] s : pro) {  for(String r :s) {  System.out.println(r+" ");  }  }  rank(string);  System.out.println("各省份及其范围：");  for(String[] s : rank) {  for(String r : s) {  System.out.print(r + " ");  }  }  String location = pro[0][0];  //遍历pro  for(int j=0 ; j<pro.length ; j++) {  //遍历rank  for(int i=0 ; i<rank.length ; i++) {  if(rank[i][0].equals(pro[j][0])) {  exchange(string,Integer.valueOf(rank[i][1]),Integer.valueOf(rank[i][2]));  }  else  continue;  }  }  //输出排列好的所有数据  File file = new File(args[1]);  FileOutputStream out = new FileOutputStream(file);  OutputStreamWriter writer = new OutputStreamWriter(out);  if(args.length<2||args.length>3)  System.out.println("参数出错");  else if(args.length == 2) {  writer.write(location + " " + pro[0][1] + "\r\n\r\n");  for(int j=0 ; j<pro.length ; j++) {  for(int i=0 ; i<string.length ; i++) {  if(string[i][0].equals(pro[j][0]))  writer.write(string[i][1] + " " + string[i][2] + "\r\n");  else  continue;  }  if(j != pro.length-1) {  location = pro[j+1][0];  writer.write(location + " " + pro[j+1][1] + "\r\n\r\n");  }  else  break;  }  }  else {  location = args[2];  writer.write(location + " " + pro[0][1] + "\r\n\r\n");  for(int j=0 ; j<string.length ; j++) {  if(string[j][0].equals(location))  writer.write(string[j][1] + " " + string[j][2] + "\r\n");  else  continue;  }  }  writer.close();  }  }  （四）结果测试  1.进入实验文件夹test4，如图33所示。    图33 进入d盘下的test4文件夹  2.指定输入文件yq\_in\_04.txt和输出文件yq\_out\_04.txt，如图34-35所示。    图34 指定输入输出文件    图35 成功输出文件yq\_out\_04.txt  3.打开输出文件yq\_out\_04.txt，输出所有省份的疫情信息，每个省份后面有该省份所有疫情人数的总数，输出省按总数从大到小排序，如果两个省总数一样，按拼音排序，每个省内各市按人数从大到小排序，如图36所示。    图36 输出所有省份的疫情信息  3.指定在yq\_out\_04.txt中输出安徽省的疫情信息，省内各市按人数从大到小排序，如果两个市总数一样，按拼音排序，如图37-图38所示。    图37 指定输出安徽省的疫情信息    图38 输出安徽省的疫情信息  4.指定在yq\_out\_04.txt中输出贵州省的疫情信息，省内各市按人数从大到小排序，如果两个市总数一样，按拼音排序，如图39-图40所示。    图39 指定输出贵州省疫情信息    图40 输出贵州省的疫情信息  **四、分析讨论**  通过这次实验，我懂得了理论与实践相结合是重要的，我更好地掌握了编写程序来对数据文件进行处理，熟悉了Java的文件读写机制以及输入输出流的使用。  在实验过程中，我发现调试对于写好一个程序有帮助的功能，通过调试可以发现一些细微的错误，这样就可以及时的发现并且改正。本次实验也提高了我分析问题和解决问题的能力，能够通过查阅资料获得所需信息，解决实验过程中所遇到的问题。  这次实验比上次实验难度更大，我发现了注释的重要性，注释可以方便排错，捋清思路，知道自己在做什么，以及做到哪个环节了。通过和同学之间的讨论研究以及上网查找相关资料之后，我逐渐找到问题的解决方法，实现了所需的功能，完成了程序的编写。在以后的实验中，我一定会认真仔细的完成。  **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。