西南民族大学

**实验报告**

20 20 ------20 21学年第 1学期

课程名称：Java课程设计

学 院：计算机科学与工程学院

专 业：软件工程

年级：2018

班级：软工1801

学 号：201831104032

姓 名：罗薇

同组人：无

|  |
| --- |
| 西南民族大学学生实验报告  教学单位：计算机科学与技术学院 实验室名称： 实验时间：2020年9月24日  姓名：罗薇 专业：软件工程 班级：1801 学号:201831104032 |
| 实验项目名称：软件工程课程设计作业 实验成绩： 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）。  一、实验目的与背景  在实验三的基础上继续：  输入文件为yq\_in.txt,输出文件yq\_out.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，按首字母排序 3. 每个省内市名从大到小排序，如果两个市总是一样，按首字母排序 4. 材料与方法   材料：计算机一台。  环境：Windows 10   1. 实验主要过程与结果   主要代码  **package** newtest;  **import** java.io.File;  **import** java.io.FileReader;  **import** java.io.FileWriter;  **import** java.text.Collator;  **import** java.util.Collections;  **import** java.util.Comparator;  **import** java.util.LinkedList;  **import** java.util.List;  **import** java.util.Scanner;  **import** java.util.jar.JarOutputStream;  **import** javax.swing.JOptionPane;  **public** **class** NewTest {  private final static Comparator<Object> CHINA\_COMPARE = Collator.getInstance(java.util.Locale.CHINA);  **public** **static** **void** main(String[] args) {  // **TODO** 自动生成的方法存根  String path1 = JOptionPane.showInputDialog(null,"请输入路径");  File file = new File(path1);  Scanner in = **new** Scanner(System.***in***);  String input = in.nextLine();  String[] splitLine1 = input.split(" ");//输入一行，空格分割  String yq\_in = splitLine1[0];  String yq\_out = splitLine1[1];  File file = **new** File("D://softwareprotect//" + yq\_in + ".txt");  FileReader fr = **null**;  FileWriter fw = **null**;  **int** flag = 0;  String name = "";  **if**(splitLine1.length == 3) //判断是否输入第三个数据  {  name = splitLine1[2];//市名市splitLine1[2]  flag = 1;  }  **try** {  fr = **new** FileReader(file);//读文件  **char**[] data = **new** **char**[10000];//初始化字符  **int** length = 0;//长度初始化为0  String str = **null**;//初始化字符串为空  **while** ((length = fr.read(data)) > 0) {  str = **new** String(data, 0, length);//str字符串长度是data的长度，内容是data，从0开始  }  String[] splitLine = str.split("\r\n|\t");//换行分割str字符串    **if**(flag == 0) //没有第三个数据的情况  {  ChangeClass[] changeclass = **new** ChangeClass[30];  //List<String> newLine = new LinkedList<>();//Java代码将文件的读取至到StringList中，再将List内容转化成源文件，链表  **for**(**int** i = 0; i < changeclass.length ; i++)//输入的长度  {  changeclass[i] = **new** ChangeClass();    // String province = splitLine[i].substring(0,3);//从0开始，分割成3分（省份）  // String nextProvince = splitLine[i+1].substring(0,3);//下一个省份  // if(i == 0) {  // newLine.add(province);  //  //  // }//源文件添加省份  // newLine.add(splitLine[i].substring(splitLine[i].indexOf("\t") + 1,splitLine[i].length()));  // Collections.sort(newLine, CHINA\_COMPARE);  // if( province.equals(nextProvince) == false)//两个省份比较，不一样就添加下一个省份  // {  // newLine.add("\n");  // newLine.add(nextProvince);  // }  // Collections.sort(newLine, CHINA\_COMPARE);  }  **int** mark\_tmp =0;  **int** index =0;  **int** city\_tmp = 0;  **boolean** bool\_index = **true**;  **for**(**int** i=0;i<splitLine.length;i++) {  String province = splitLine[3\*i];  String nextProvince = splitLine[3\*(i+1)];  changeclass[index].city.add(splitLine[3\*i+1]);  changeclass[index].mark.add(splitLine[3\*i+2]);  city\_tmp++;  mark\_tmp += Integer.*parseInt*(splitLine[3\*i+2]);  **if**(province.equals(nextProvince) == **false**){  changeclass[index].province = splitLine[3\*i];  changeclass[index].total\_city = city\_tmp;  changeclass[index].total\_mark = mark\_tmp;  index++;  mark\_tmp = 0;  city\_tmp = 0;      }  **if**(i == splitLine.length/3-2) {  **if**(province.equals(nextProvince) == **false**) {  bool\_index = **true**;  changeclass[index].province =splitLine[3\*(i+1)];  changeclass[index].total\_city = 1;  changeclass[index].total\_mark = Integer.*parseInt*(splitLine[3\*(i+1)+2]);  changeclass[index].city.add(splitLine[3\*(i+1)+1]);  changeclass[index].mark.add(splitLine[3\*(i+1)+2]);  }  **else** {  bool\_index = **false**;  changeclass[index-1].total\_city += 1;  changeclass[index-1].total\_mark += Integer.*parseInt*(splitLine[3\*(i+1)+2]);  changeclass[index-1].city.add(splitLine[3\*(i+1)+1]);  changeclass[index-1].mark.add(splitLine[3\*(i+1)+2]);  }  }  // StringBuilder sb = new StringBuilder();  // for(int i = 0; i < newLine.size();i++){  // sb.append(newLine.get(i));  // if(newLine.get(i).equals("\n") == false) {  // sb.append("\n");  // }  // }  // String s = sb.toString();  // fw = new FileWriter("D://softwareprotect//" + yq\_out +".txt");  // fw.write(s);  }  **if**(bool\_index == **false**)  index--;  **for**(**int** i = 0; i <= index; i++) {  **for**(**int** j = 0; j <= index-i-1; j++) {  **if**(changeclass[j].total\_mark < changeclass[j+1].total\_mark) {  ChangeClass tmp = **new** ChangeClass();  tmp = changeclass[j];  changeclass[j] = changeclass[j+1];  changeclass[j+1] = tmp;  }  **if**(changeclass[j].total\_mark == changeclass[j+1].total\_mark) {  **if**(changeclass[j].province.compareTo(changeclass[j+1].province) > 0) {  ChangeClass tmp = **new** ChangeClass();  tmp = changeclass[j];  changeclass[j] = changeclass[j+1];  changeclass[j+1] = tmp;  }  }    }  }  **for**(**int** i = 0; i <= index; i++) {  **for**(**int** j = 0; j < changeclass[i].total\_city; j++) {  **for**(**int** k = 0; k < changeclass[i].total\_city-j-1; k++)  {  **if**(Integer.*parseInt*(changeclass[i].mark.get(k))  < Integer.*parseInt*(changeclass[i].mark.get(k+1))) {  String tmpcity, tmpmark;  tmpcity = changeclass[i].city.get(k);  changeclass[i].city.set(k, changeclass[i].city.get(k+1));  changeclass[i].city.set(k+1, tmpcity);  tmpmark = changeclass[i].mark.get(k);  changeclass[i].mark.set(k, changeclass[i].mark.get(k+1));  changeclass[i].mark.set(k+1, tmpmark);  }  **if**(Integer.*parseInt*(changeclass[i].mark.get(k))  == Integer.*parseInt*(changeclass[i].mark.get(k+1))) {  **if**(changeclass[i].city.get(k).compareTo(changeclass[i].city.get(k+1)) > 0) {  String tmpcity, tmpmark;  tmpcity = changeclass[i].city.get(k);  changeclass[i].city.set(k, changeclass[i].city.get(k+1));  changeclass[i].city.set(k+1, tmpcity);  tmpmark = changeclass[i].mark.get(k);  changeclass[i].mark.set(k, changeclass[i].mark.get(k+1));  changeclass[i].mark.set(k+1, tmpmark);  }  }  }  }  }  StringBuilder sb = **new** StringBuilder();  **for**(**int** i = 0; i <= index;i++){  sb.append(changeclass[i].province + " " + changeclass[i].total\_mark + "\n");  **for**(**int** j = 0; j < changeclass[i].total\_city; j++)  sb.append(changeclass[i].city.get(j) + " " + changeclass[i].mark.get(j) + "\n");  sb.append("\n");  }  String s = sb.toString();  fw = **new** FileWriter("D://softwareprotect//" + yq\_out +".txt");  fw.write(s);  }    **else**  {  **int** marktmp = 0;  **int** totalcitytmp = 0;  ChangeClass exchangeclass = **new** ChangeClass();  **for**(**int** i = 0; i < splitLine.length/3; i++){  **if**(name.equals(splitLine[3\*i])) {  exchangeclass.city.add(splitLine[3\*i+1]);  exchangeclass.mark.add(splitLine[3\*i+2]);  totalcitytmp++;  marktmp += Integer.*parseInt*(splitLine[3\*i+2]);  }  }  exchangeclass.total\_city = totalcitytmp;  exchangeclass.province = name;  exchangeclass.total\_mark = marktmp;  **for**(**int** i = 0; i < exchangeclass.total\_city; i++) {  **for**(**int** j = 0; j < exchangeclass.total\_city-i-1; j++) {  **if**(Integer.*parseInt*(exchangeclass.mark.get(j))  < Integer.*parseInt*(exchangeclass.mark.get(j+1))) {  String tmpcity, tmpmark;  tmpcity = exchangeclass.city.get(j);  exchangeclass.city.set(j, exchangeclass.city.get(j+1));  exchangeclass.city.set(j+1, tmpcity);  tmpmark = exchangeclass.mark.get(j);  exchangeclass.mark.set(j, exchangeclass.mark.get(j+1));  exchangeclass.mark.set(j+1, tmpmark);  }  **if**(Integer.*parseInt*(exchangeclass.mark.get(j))  == Integer.*parseInt*(exchangeclass.mark.get(j+1))) {  **if**(exchangeclass.city.get(j).compareTo(exchangeclass.city.get(j+1)) > 0) {  String tmpcity, tmpmark;  tmpcity = exchangeclass.city.get(j);  exchangeclass.city.set(j, exchangeclass.city.get(j+1));  exchangeclass.city.set(j+1, tmpcity);  tmpmark = exchangeclass.mark.get(j);  exchangeclass.mark.set(j, exchangeclass.mark.get(j+1));  exchangeclass.mark.set(j+1, tmpmark);  }  }  }  }  StringBuilder sb = **new** StringBuilder();  sb.append(exchangeclass.province + " " + exchangeclass.total\_mark + "\n");  **for**(**int** i = 0; i < exchangeclass.total\_city; i++) {  sb.append(exchangeclass.city.get(i) + " " + exchangeclass.mark.get(i) + "\n");  }  String s = sb.toString();  fw = **new** FileWriter("D://softwareprotect//" + yq\_out +".txt");  fw.write(s);  }  } **catch** (Exception e) {  e.printStackTrace();  }**finally** {  **try** {  fr.close();  fw.close();  } **catch** (Exception e) {  e.printStackTrace();  }  }  in.close();  }    }  import java.util.ArrayList;  /\*\*  \*  \*/  /\*\*  \* @author v  \*  \*/  public class ExchangeClass {  public String province;  public int total\_mark;  public int total\_city;  public ArrayList<String> city = new ArrayList<>();  public ArrayList<String> mark = new ArrayList<>();  }  代码运行结果部分截图     1. 实验心得与分析   通过这次的学习与应用，基于上次实验的经验，在原有的实验基础上进行功能的增加。总体感觉会有些难度，首先想到的是存入数组再排序，但是这样操作会使代码变得更复杂一些，数组会很多，而且还存入也会比较麻烦。所以我写了一个类，把数据读出来放到数组的类中再排序，然后写入文件。   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。