西南民族大学

**实验报告**

20 19 ------20 20 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：软件工程

年级： 2018 班级：1801

学号： 201831104039 姓名：孟原硕

同组人：

|  |
| --- |
| 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： 实验时间：2020年9月7日  姓名：孟原硕 专业： 软件工程 班级：1801 学号:孟原硕 |
| 实验项目名称： 实验成绩： 教师签名： |
| 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. **解决思路，流程**   **文件输入时yq\_in.txt，yq\_out以及省份是采用读取一行内容，使用split对空格进行分割，**  **判断分割数组长度，若为3则判定输入了省份，若为2则是判断未输入省份。**  **对于两个输入方法选择了使用类的构造不同使用了方法重载来区分。**  **对于排序，创建一个新的类，利用类存储省名，省内市名与数据，通过遍历得到省内总数据，**  **利用两层循环，把省和市依照数据大小和省名，市名的ascll码进行排序，最后进行输出**     1. **代码**   package s;  import java.io.\*;  import java.util.ArrayList;  import java.util.Scanner;  public class itle {  public static void main(String[] args) throws FileNotFoundException {  Scanner inputs = new Scanner(System.in);  String all = inputs.nextLine();  String[] splitLine = all.split(" ");  String yq\_in = splitLine[0];  String yq\_out = splitLine[1];  if(splitLine.length == 3) {  String prov = splitLine[2];  land b = new land(yq\_in,yq\_out,prov);  }  else {  land a = new land(yq\_in,yq\_out);  }    //C:\\Users\\Administrator\\Desktop\\yq\_in.txt C:\\Users\\Administrator\\Desktop\\yq\_out.txt      }  }  package s;  import java.io.File;  import java.io.FileNotFoundException;  import java.io.FileOutputStream;  import java.io.PrintStream;  import java.util.ArrayList;  import java.util.Scanner;  public class land {    land(String YQ\_in,String YQ\_out,String provs) throws FileNotFoundException{    File file = new File(YQ\_in);  if (file.exists()) {  System.out.println("File already exists");  System.out.println(file);  }  Scanner input = new Scanner(file);  ArrayList<String> province = new ArrayList<String>();  ArrayList<String> city = new ArrayList<String>();  ArrayList<String> data = new ArrayList<String>();  while (input.hasNext()) {  String Province = input.next();  String City = input.next();  String Data = input.next();  if(!City.equals("待明确地区")) {  province.add(Province);  city.add(City);  data.add(Data);  }  }  prov Pro = new prov();  Pro.prov\_name = provs;  for(int j = 0 ; j < province.size() ; j++) {  if(Pro.prov\_name.equals(province.get(j)) == true ) {  Pro.city\_name.add(city.get(j));  Pro.city\_num.add(Integer.parseInt(data.get(j)));  Pro.prov\_num +=Integer.parseInt(data.get(j));  }  }  Pro.Resort();  // for(int i = 0 ; i < Pro.city\_name.size() ; i++) //测试  // {  // System.out.println(Pro.city\_name.get(i)+" "+Pro.city\_num.get(i));  // }  FileOutputStream fs = new FileOutputStream(new File(YQ\_out));  PrintStream p = new PrintStream(fs);  p.println(Pro.prov\_name + " " +Pro.prov\_num);  for(int i = 0 ; i < Pro.city\_name.size() ; i++) //测试  {  p.println(Pro.city\_name.get(i)+" "+Pro.city\_num.get(i));  }  p.close();  }        land(String YQ\_in,String YQ\_out) throws FileNotFoundException{    File file = new File(YQ\_in);  if (file.exists()) {  System.out.println("File already exists");  System.out.println(file);  }  Scanner input = new Scanner(file);  ArrayList<String> province = new ArrayList<String>();  ArrayList<String> city = new ArrayList<String>();  ArrayList<String> data = new ArrayList<String>();  while (input.hasNext()) {  String Province = input.next();  String City = input.next();  String Data = input.next();  if(!City.equals("待明确地区")) {  province.add(Province);  city.add(City);  data.add(Data);  }  }  prov PROV[]= new prov[34];  for(int i = 0 ; i < 34 ; i++) {  prov Pro = new prov();  PROV[i] = Pro;  }    System.out.println(PROV[0].prov\_name);  PROV[0].prov\_name = province.get(0); //先赋一次值  int length = 0;  for(int i = 0 , j = 0 ; j < province.size() ; j++) {  if(PROV[i].prov\_name.equals( province.get(j) ) == false ) { //省名不同,先++再赋值  i++;  length++;  PROV[i].prov\_name = province.get(j);  }  if(PROV[i].prov\_name.equals( province.get(j) )) { //相同或赋值完相同  PROV[i].prov\_num += Integer.parseInt(data.get(j)) ;  PROV[i].city\_name.add(city.get(j));  PROV[i].city\_num.add(Integer.parseInt(data.get(j)));  }  }    for(int i = 0 ; i < length ; i++) { //对市进行排序  PROV[i].Resort();  }      String mid\_name; //对省进行排序  int mid\_num = 0;  ArrayList<String> mid\_city\_name = new ArrayList<>();  ArrayList<Integer> mid\_city\_num = new ArrayList<>();    for(int i = 0 ; i < length ; i++) {  for( int j = i+1 ; j < length ; j++) {  if(PROV[i].prov\_num <= PROV[j].prov\_num)  {  if(PROV[i].prov\_num == PROV[j].prov\_num) {  if(PROV[i].prov\_name.compareTo(PROV[j].prov\_name) > 0) continue;  }  else {  mid\_name = PROV[i].prov\_name;  mid\_num = PROV[i].prov\_num;  mid\_city\_name = PROV[i].city\_name;  mid\_city\_num = PROV[i].city\_num;    PROV[i].prov\_name = PROV[j].prov\_name;  PROV[i].prov\_num = PROV[j].prov\_num;  PROV[i].city\_name = PROV[j].city\_name;  PROV[i].city\_num = PROV[j].city\_num;    PROV[j].prov\_name = mid\_name;  PROV[j].prov\_num = mid\_num;  PROV[j].city\_name = mid\_city\_name;  PROV[j].city\_num = mid\_city\_num;  }  }  }  }      // for(int i = 0 ; i < length ; i++) { //测试  // System.out.println(PROV[i].prov\_name);  // System.out.println(PROV[i].prov\_num);  // for(int j = 0 ; j < PROV[i].city\_name.size() ; j++) {  // System.out.println(PROV[i].city\_name.get(j));  // System.out.println(PROV[i].city\_num.get(j));  // }  //  // }  FileOutputStream fs = new FileOutputStream(new File(YQ\_out)); //输出至文件  PrintStream p = new PrintStream(fs);  for(int i=0; i < length ; i++){  p.print(PROV[i].prov\_name+" ");  p.println(PROV[i].prov\_num);  for(int j = 0 ; j < PROV[i].city\_name.size() ; j++) {  p.print(PROV[i].city\_name.get(j)+" ");  p.println(PROV[i].city\_num.get(j));  }    }  p.close();  }    }  **package** s;  **import** java.util.ArrayList;  **public** **class** prov{  **public** String prov\_name = "";  **public** ArrayList<String> city\_name = **new** ArrayList<>();  **public** ArrayList<Integer> city\_num = **new** ArrayList<>();  **public** **int** prov\_num = 0;    **public** String mid\_name;  **public** **int** mid\_num = 0;  **public** **void** Resort() {  **for**(**int** i = 0 ; i < city\_name.size() ; i++) {  **for**( **int** j = i+1 ; j < city\_name.size() ; j++) {  **if**(city\_num.get(i) <= city\_num.get(j))  {  **if**(city\_num.get(i) == city\_num.get(j)) {  **if**(city\_name.get(i).compareTo(city\_name.get(j)) > 0) **continue**;  }  **else** {  mid\_name = city\_name.get(i);  mid\_num = city\_num.get(i);  city\_name.set(i,city\_name.get(j));  city\_num.set(i,city\_num.get(j));  city\_name.set(j,mid\_name);  city\_num.set(j,mid\_num);  }  }  }  }  }    }   1. **结果测试**   **运行：**  **1.**      2. |
|  |
|  |
|  |

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。