西南民族大学

**实验报告**

2020 ------2021 学年第1 学期

课程名称：软件工程课程设计

学 院：计算机科学与技术学院 专 业：软件工程

年级：2018级 班级：1801

学 号：201831104075姓 名：叶欣钰

同组人：

|  |
| --- |
| 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： BS-223 实验时间：2020年9月 日  姓名：叶欣钰 专业：软件工程 班级：1801 学号:201831104075 |
| 实验项目名称：软件工程课程设计 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）。  一、实验背景  1.语言要求：C/C++，Java，。。。  2.实验报告内容：解决思路，流程，代码及结果测试；  3.源码上传至个人git   1. 材料与方法 2. 语言要求：C/C++，Java，。。   三、实验主要过程与结果  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；   实现代码如下:  package 作业;  import java.io.\*;  import java.util.ArrayList;  import java.util.Arrays;  import java.util.List;  import java.util.Scanner;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class in\_out\_04 {  public static void main(String[] args) {  // TODO Auto-generated method stub  Scanner in=new Scanner(System.in);  String s1=in.nextLine();  String s2=in.nextLine();  // String s3=in.nextLine();  // "D:\\yq\_in.txt"  // "D://yq\_out.txt"  try {  String encoding="GBK";  File f1=new File(s1);  if (f1.isFile() && f1.exists()) {  // 判断文件是否存在  InputStreamReader isr=new InputStreamReader(new FileInputStream(f1), encoding);  // 考虑到编码格式  BufferedReader br=new BufferedReader(isr);  File f2=new File(s2);  f2.createNewFile();  BufferedWriter bw=new BufferedWriter(new FileWriter(f2, true));  FileOutputStream fos=new FileOutputStream(f2);  PrintStream ps=new PrintStream(fos);  System.setOut(ps);    String line = null;  String[] province=new String[500]; //省  // String[] region=new String[500]; //市  String[] quantity = new String[500]; //数  // String[] province\_region = new String[500]; //省市  String[] region\_quantity = new String[500]; //市数  String[] line3 = new String[3]; //一行的三个数据  String[] a = new String[500]; //数 转  int[] quantit = new int[500]; //数 数  int[] sum = new int[100]; //总数  String[] str\_province = new String[100]; //归纳出的省  String[] str\_region\_quantity = new String[100]; //临时 市 数  int i = 0, j = 0, k = 0, n=-1,m=-1;  while ((line = br.readLine()) != null) {  String resu[] = line.split("\\s+");  for (String r : resu) {  // System.out.println(r);  line3[j] = r;  j++;  }  j = 0;  province[i]=line3[0];  // region[i]=line3[1];  // province\_region[i] = line3[0] + "\t" + line3[1];  region\_quantity[i] = line3[1] + "\t" + line3[2];  quantity[i] = line3[2];  String regEx = "[^0-9]";  Pattern pattern = Pattern.compile(regEx);  Matcher ma = pattern.matcher(quantity[i]);  a[i] = ma.replaceAll("").trim();  quantit[i] = Integer.parseInt(a[i]);  i++;  }  //省排序  String pro=null;  for(i=0;province[i]!=null;i++)  {  if(!province[i].equals(pro))  {  n++;  sum[n]+=quantit[i];  str\_province[n]=province[i];  }  else if(province[i].equals(pro))  {  sum[n]+=quantit[i];  }  pro=province[i];  }  for(i=0;i<=n;i++)  {  for(j=0;j<=n-1-i;j++)  {  if(sum[j]<sum[j+1])  {  int temp=sum[j];  sum[j]=sum[j+1];  sum[j+1]=temp;  String t=str\_province[j];  str\_province[j]=str\_province[j+1];  str\_province[j+1]=t;  }  if(sum[j]==sum[j+1] && str\_province[j].compareTo(str\_province[j+1])>0)  {  int temp=sum[j];  sum[j]=sum[j+1];  sum[j+1]=temp;  String t=str\_province[j];  str\_province[j]=str\_province[j+1];  str\_province[j+1]=t;  }  }  }  //市排序  for(i=0;i<=n;i++)  {  m=-1;  System.out.println(str\_province[i]+"\t"+sum[i]);  for(j=0;province[j]!=null;j++)  {  if(province[j].equals(str\_province[i]))  {  m++;  str\_region\_quantity[m]=region\_quantity[j];  }  }  for(j=0;j<=m;j++)  {  for(k=0;k<=m-j-1;k++)  {  String[] r1=str\_region\_quantity[k].split("\\s+");  int q1=Integer.valueOf(r1[1]);  String[] r2=str\_region\_quantity[k+1].split("\\s+");  int q2=Integer.valueOf(r2[1]);  if(q1<q2)  {  String temp=str\_region\_quantity[k];  str\_region\_quantity[k]=str\_region\_quantity[k + 1];  str\_region\_quantity[k+1]=temp;  }  if(q1==q2 && r1[0].compareTo(r2[0])>0)  {  String temp=str\_region\_quantity[k];  str\_region\_quantity[k]=str\_region\_quantity[k+1];  str\_region\_quantity[k+1]=temp;  }  }  }  for(j=0;j<=m;j++)  {  System.out.println(str\_region\_quantity[j]);  }  System.out.println();  }  isr.close();  } else  System.out.println("找不到指定的文件");  } catch (Exception e) {  // System.out.println("读取文件内容出错");  e.printStackTrace();  }  }  }  实现前如下图：    图1  实现后如下图：    图2    图3    图4    图5    图6    图7  个人git链接：<https://github.com/heartnavyblue>  四、分析讨论  数组声明：数据类型 [] 数组名称 = new 数据类型[长度];  冒泡排序的优点：每进行一趟排序，就会少比较一次，因为每进行一趟排序都会找出一个较大值。如上例：第一趟比较之后，排在最后的一个数一定是最大的一个数，第二趟排序的时候，只需要比较除了最后一个数以外的其他的数，同样也能找出一个最大的数排在参与第二趟比较的数后面，第三趟比较的时候，只需要比较除了最后两个数以外的其他的数，以此类推……也就是说，没进行一趟比较，每一趟少比较一次，一定程度上减少了算法的量。  本次实验设计完成较为顺利，锻炼了我的逻辑思维能力，增加了我的编程经验，提高了我的编程水平，完善了我对一些知识点的认知，有了更进一步的了解，看到了自己的不足，并需要更进一步的努力，此次实验设计使我受益匪浅，我将进一步细致的学习相关知识，并更好的应用于实践中。  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。