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

姓 名：许芷毓 201731102245

谢晓辉 201731102242

|  |
| --- |
| 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26日  专业：计算机科学与技术 班级：1702班 |
| 实验项目名称：第五次实验 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。  5.通过结对编程的完成过程来进一步学习编程及团队合作的思想。  **二、材料与方法**  1.语言要求：Java  2.IDE：Eclipse  3.辅助工具：Git  4.实验要求：  结对编程：  我们已经实现了三个功能：  （1）单独运行yq.exe或者yq.java，给出缺省输入文件的输出；  （2）运行yq [指定输入文件] [指定输出文件]；（按要求排序）  （3）运行yq [指定输入文件][指定输出文件] [指定省份]；（按要求排序）  把个人项目的三个的功能封装为独立模块，并设计单元测试。  **三、实验主要过程与结果**  （一）分析解决思路  1.在原有实验的基础上将函数初步进行封装，参数为2（[指定输入文件] [指定输出文件]）的一个函数与参数为3（[指定输入文件][指定输出文件] [指定省份]）的一个函数。  2.将读入输入文件的内容写入list并将其归入某一函数，后续可调用。  3.将按照人数总数对省份进行排序的这一段单独封装成一个函数。  4.将各个省份内的城市排序的这一段单独封装成一个函数。  （二）实验流程  1.新建工程  ①创建一个新的Java工程Epidemic situation，如图1.1-图1.2所示。    图1.1  2  图1.2 新建java项目Epidemic situation  ②在Epidemic situation工程下面新建一个package包city，如图1.3所示。  3  图1.3 新建包city  ③在package包city下新建一个类Test\_report，如图1.4所示。  4  图1.4 新建类Test\_report  ④在package包city下新建一个类Core，如图1.5所示。    图1.5 新建类Core 图1.6 city包下的类  2.在新建的类页面编写代码，如图1.7-1.8所示。  1  图1.7    图1.8  3.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选择项目名称右击选择【Export...】，出现如图1.9所示的页面，展开【Java】选项，选择【JAR file】点击Next。    图1.9  ②点击Browse按钮，选择生成的JAR文件路径，其它选项可以暂时使用默认的，然后点击Next，如图1.10-1.11所示。  17  图1.10 设置要导出文件的保存路径 图1.11 jar包选项  ③直接点击Next按钮，在该页面的其它选项采用默认即可，点击Browse按钮选择项目中主类(带有main方法的类)，如图1.12所示。  1.2  图1.12 选择主类  ④直接点击【OK】后如图1.13所示,再点击【Finish】按钮，即可完成。  1.3  图1.13 jar清单规格  （2）将java程序转换为exe。  ①找到exe4j下面的bin目录，双击exe4j.exe运行。  10  图2.1 运行exe4j  ②直接点击Next按钮，选择"JAR in EXE" mode，如图2.2所示。  11  图2.2 选择项目类型  ③点击Next按钮，Short name of your application中可随便定义，选择定义为XXTest\_report；Output directory中是用来指定生成exe文件的目录，如图2.3所示。  a  图2.3 配置应用  ④点击Next按钮出现如图2.4所示的页面，点击Advanced Options后点击32bit or 64bit一栏后出现如图2.5所示的页面，勾选Generate 64-bit executable。  b  图2.4 配置可执行文件 图2.5 选择生成的可执行文件的主要结构  ⑤点击Next按钮出现如图2.6所示的页面后再次点击Next，在Executable name中输入要生成exe文件的名称，如图2.7所示。  b  图2.6 可执行清单的配置选项 图2.7 编辑生成的exe文件名  ⑥在点击Next按钮后出现的页面上点击右侧绿色的“+”出现如图2.8所示的页面。在Archive中选择我们第一步生成的jar文件，然后直接点击OK按钮，如图2.9所示；点击Main class 后面的按钮，选择程序的入口即程序含有main方法的类，点击OK按钮即可，如图2.10所示。    图2.8 java调用配置  cd  图2.9 定义类路径入口 图2.10 选择主类  ⑦点击Next，在最低版本要求处填写了1.4，再次点击Next依次出现图2.12、2.13所示的页面，再次点击Next出现图2.14所示页面即可，即成功完成转换。    图2.11 jre配置    图2.12 配置闪屏 图2.13 配置可执行的消息  c  图2.14 转换完成  3  图2.15 成功生成Test\_report.exe  （三）程序代码  package city;  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 Test\_report {  static int *j* =0;  static String *location*;  static int *count*=0;  static String *loc*;  static int *sum*=0;  static String[][] *string*;  //对省份的城市进行分割  static String[][] *rank*=new String[9][3];  //进行省份的排序  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*);  //创建一个列表,用来装按行读取到的内容  static List<String> *list*=new ArrayList<>();    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();  *string* = new String[*list*.size()][];  for(int i = 0;i<*list*.size();i++)  {  *string*[i] = *list*.get(i).split("\\s+");  }  }  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];  }  }  }  }  }  //按照省份对省份内的城市进行排序  //a为一个省份的起始位置，b为一个省份的结束位置  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;  //设定第一个起始位置为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 pro()  {  *loc*=*string*[0][0];  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];  }  }  *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*++;  }  }  public static void area() {  *location*=*pro*[0][0];  for(int j =0 ;j < *pro*.length;j++)  {  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;  }  }  }  }  public static void main(String[] args) throws IOException {  *readFileByLines*(args[0]);  *pro*();  *change*(*pro*);  System.*out*.println("按总人数从多到少的排序方法进行省份的排序：");  for(String[] s :*pro*)  {  for(String r :s)  {  System.*out*.println(r+" ");  }  }  *rank*(*string*);  System.*out*.println("各省份及其所包含城市的范围：");  for(String[] s :*rank*)  {  for(String r :s)  {  System.*out*.print(r+" ");  }  }  *area*();  File file = new File(args[1]);  FileOutputStream out = new FileOutputStream(file);  OutputStreamWriter wr =new OutputStreamWriter(out);  if(args.length<2||args.length>3)  {  System.*out*.println("参数出错");  }  else if(args.length==2)  {  //先写入人数最多的省份并加换行到文档  wr.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]))  {  wr.write(*string*[i][1]+" "+*string*[i][2]+"\r\n");  }  else  {  continue;  }  }  if(j!=*pro*.length-1)  {  *location*=*pro*[j+1][0];  wr.write(*location*+" "+*pro*[j+1][1]+"\r\n\r\n");  }  else  break;  }  }  else  {  *location* = args[2];  wr.write(*location*+" "+*pro*[0][1]+"\r\n\r\n");  for(int j =0 ;j < *string*.length;j++)  {  if(*string*[j][0].equals(*location*))  {  wr.write(*string*[j][1]+" "+*string*[j][2]+"\r\n");  }  else  {  continue;  }  }  }  wr.close();  }  }  【进行封装后】  package city;  import java.io.File;  import java.io.FileOutputStream;  import java.io.IOException;  import java.io.OutputStreamWriter;  public class Core {  public static void main(String[] args) throws IOException {  if(args.length==2)  {  *tip2*(args[0],args[1]);  }  else if(args.length==3)  {  *tip3*(args[0],args[1],args[2]);  }  else  {  System.*out*.println("参数输入有误,请重新输入!");  }  }  public static void tip2(String inputfile,String outputfile) throws IOException {  Test\_report.*readFileByLines*(inputfile);  Test\_report.*pro*();  Test\_report.*change*(Test\_report.*pro*);  Test\_report.*rank*(Test\_report.*string*);  System.*out*.println("各省份及其所包含城市的范围：");  Test\_report.*area*();  File file = new File(outputfile);  FileOutputStream out = new FileOutputStream(file);  OutputStreamWriter wr =new OutputStreamWriter(out);  wr.write(Test\_report.*location*+" "+Test\_report.*pro*[0][1]+"\r\n\r\n");  for(int m = 0;m < Test\_report.*pro*.length;m++)  {  for(int i=0;i<Test\_report.*string*.length;i++)  {  if(Test\_report.*string*[i][0].equals(Test\_report.*pro*[m][0]))  {  wr.write(Test\_report.*string*[i][1]+" "+Test\_report.*string*[i][2]+"\r\n");  }  else  {  continue;  }  }  if(m!=Test\_report.*pro*.length-1)  {  Test\_report.*location*=Test\_report.*pro*[m+1][0];  wr.write(Test\_report.*location*+" "+Test\_report.*pro*[m+1][1]+"\r\n\r\n");  }  else  break;  }  wr.close();  }    public static void tip3(String inputfile,String outputfile,String area) throws IOException {  Test\_report.*readFileByLines*(inputfile);  Test\_report.*pro*();  Test\_report.*change*(Test\_report.*pro*);  Test\_report.*rank*(Test\_report.*string*);  System.*out*.println("各省份及其所包含城市的范围：");  Test\_report.*area*();  File file = new File(outputfile);  FileOutputStream out = new FileOutputStream(file);  OutputStreamWriter wr =new OutputStreamWriter(out);  Test\_report.*location*=area;  wr.write(Test\_report.*location*+" "+Test\_report.*pro*[0][1]+"\r\n\r\n");  for(int n =0 ;n < Test\_report.*string*.length;n++)  {  if(Test\_report.*string*[n][0].equals(Test\_report.*location*))  {  wr.write(Test\_report.*string*[n][1]+" "+Test\_report.*string*[n][2]+"\r\n");  }  else  {  continue;  }  }  wr.close();  }  }  （四）单元测试  **点击工具栏Run按钮，在弹出来的选项当中点击Run Configurations。接下来在弹出的界面当中点击Arguments，并在Program arguments框里填写所传参数（输入文件与输出文件的绝对路径），多参数以空格隔开，如图2.16所示（两个参数）。**  **3**  图2.16  **填好以后点击Apply，再点击Run即可。**  Eclipse中运行结果如图2.17所示。    图2.17  此时在目标文件夹下也成功生成了yq\_out.txt,如图2.18所示。  3.4  3.23.3  图2.18  如果传递的参数为3个（图2.19），则目标文件夹下也会成功生成相应的yq\_out.txt,如图2.20所示。  4  图2.19  4.2  图2.20  （五）结果测试  （1）在cmd中通过cd命令进入实验目录。  13  图3.1 进入实验目录  （2）使用jar -tvf Test\_report.jar命令查看该jar包的内容。    图3.2 查看jar包内容  （3）在前边已成功生成了Test\_report.exe，因此可在cmd中通过输入：Test\_report yq\_in.txt yq\_out.txt 来运行，如图3.3、3.4所示。（两个参数）  2  图3.3  4  4.24.3  图3.4  （5）在cmd中通过输入：Test\_report yq\_in.txt yq\_out.txt 贵州省 来运行，如图3.5、3.6所示。（三个参数）  5  图3.5  5.2  图3.6  （6）在cmd中通过输入：Test\_report yq\_in.txt yq\_out.txt 湖南省 来运行，如图3.7、3.8所示。（三个参数）    图3.7  6.2  图3.8  **四、分析讨论**  1.在编写程序时注意添加注释，有利于理清自己的思维以及可在逐步调试或报错时发挥不可或缺的作用。  2.这次实验比上次实验难度更大，考验的是两个人之间的配合，通过和组员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。  3.因为本次实验用到了封装，我们对java的相关知识进行了复习并更深一步地进行了学习。  **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。