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：曹丽(201731102105) 吕林桃(201731102187)

|  |
| --- |
| 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： 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 |
| 1. 实验环境   1.语言要求：C/C++，Java，。。。  2.开发环境：内存8G以上  3.IDE建议：VS、IDEA或者Eclipse  4.数据库建议：MySQL或其他关系型数据库  5.辅助工具：UML绘图软件、Visio软件   1. 实验内容   在实验三的基础上继续：  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   1. 实验过程及结果 2. 实验思路：   对于本次实验，首先由我来对原函数进行封装，我将原函数分别封装为参数数量为2时执行的tip2（）函数和参数数量为3时的tip2（）函数，但由于我在实验三代码的基础上只封装了readFileByLines（），导致了后续进行对省和市排序代码的冗余，所以我的组员吕林桃在我的封装结果基础上对我的代码进行了优化，就是将对省和市的排序代码封装成了两个函数，大大减少了代码数量。   1. 实验代码：   第一次封装：  **package** text1;  **import** java.io.File;  **import** java.io.FileOutputStream;  **import** java.io.IOException;  **import** java.io.OutputStreamWriter;  **public** **class** core1 {  **public** **static** **void** main(String[] args) **throws** IOException {  **if**(args.length==2) {  *tip2*(args[0],args[1]);  }//参数数量为2时执行tip2()  **else** **if**(args.length==3) {  *tip3*(args[0],args[1],args[2]);  }//参数数量为3时执行tip3()  **else** {  System.***out***.println("参数输入错误，请重新输入");  }  }  **public** **static** **void** tip2(String inputfile,String outputfile) **throws** IOException {  xcl.*readFileByLines*(inputfile);  xcl.*string* = **new** String[xcl.*list*.size()][];  **for**(**int** i = 0;i<xcl.*list*.size();i++) {  xcl.*string*[i] = xcl.*list*.get(i).split("\\s+");//正则表达式去空字符，\\s+表示多次匹配使用  }  xcl.*loc*=xcl.*string*[0][0];  **for**(**int** r=0;r<xcl.*string*.length;r++) {//统计省份和总数  **if**(xcl.*string*[r][0].equals(xcl.*loc*)){  xcl.*sum*+=Integer.*valueOf*(xcl.*string*[r][2]);  }  **else** {  xcl.*dic*.put(xcl.*string*[r-1][0], xcl.*sum*);  xcl.*sum*=Integer.*valueOf*(xcl.*string*[r][2]);  xcl.*loc*=xcl.*string*[r][0];  }  }  xcl.*dic*.put(xcl.*string*[xcl.*string*.length-1][0], xcl.*sum*);  **for**(Object key :xcl.*dic*.keySet()) {  xcl.*pro*[xcl.*count*][0]=key.toString();  xcl.*pro*[xcl.*count*][1]=xcl.*dic*.get(key).toString();  xcl.*count*++;  }  xcl.*change*(xcl.*pro*);  System.***out***.println("按总人数从多到少排序省份：");  **for**(String[] s :xcl.*pro*) {  **for**(String r :s) {  System.***out***.println(r+" ");  }  }  xcl.*rank*(xcl.*string*);  System.***out***.println("各省份及其范围：");  **for**(String[] s :xcl.*rank*) {  **for**(String r :s) {  System.***out***.print(r+" ");  }  }  xcl.*location*=xcl.*pro*[0][0];  **for**(**int** j =0 ;j < xcl.*pro*.length;j++) {//遍历pro（已经排列好的省份数组）  **for**(**int** i=0;i<xcl.*rank*.length;i++) {//遍历rank（已经确定每个省份范围的数组）  **if**(xcl.*rank*[i][0].equals(xcl.*pro*[j][0])){  xcl.*exchange*(xcl.*string*,Integer.*valueOf*(xcl.*rank*[i][1]),Integer.*valueOf*(xcl.*rank*[i][2]));  }  **else** {  **continue**;  }  }  File file = **new** File(outputfile);  FileOutputStream out = **new** FileOutputStream(file);  OutputStreamWriter wr =**new** OutputStreamWriter(out);//进行输出  wr.write(xcl.*location*+" "+xcl.*pro*[0][1]+"\r\n\r\n");//先写入人数最多的省份加换行到文档  **for**(**int** m = 0;m < xcl.*pro*.length;m++) {  **for**(**int** i=0;i<xcl.*string*.length;i++) {  **if**(xcl.*string*[i][0].equals(xcl.*pro*[m][0])) {  wr.write(xcl.*string*[i][1]+" "+xcl.*string*[i][2]+"\r\n");  }  **else** {  **continue**;  }  }  **if**(m!=xcl.*pro*.length-1) {  xcl.*location*=xcl.*pro*[m+1][0];  wr.write(xcl.*location*+" "+xcl.*pro*[m+1][1]+"\r\n\r\n");  }  **else**  **break**;  }  wr.close();  }  }  //构造参数数量为2时应该执行的函数  **public** **static** **void** tip3(String inputfile,String outputfile,String area) **throws** IOException { //与tip2()不同之处在于多了一个参数area  xcl.*readFileByLines*(inputfile);  xcl.*string* = **new** String[xcl.*list*.size()][];  **for**(**int** i = 0;i<xcl.*list*.size();i++) {  xcl.*string*[i] = xcl.*list*.get(i).split("\\s+");//正则表达式去空字符，\\s+表示多次匹配使用  }  xcl.*loc*=xcl.*string*[0][0];  **for**(**int** r=0;r<xcl.*string*.length;r++) {//统计省份和总数  **if**(xcl.*string*[r][0].equals(xcl.*loc*)){  xcl.*sum*+=Integer.*valueOf*(xcl.*string*[r][2]);  }  **else** {  xcl.*dic*.put(xcl.*string*[r-1][0], xcl.*sum*);  xcl.*sum*=Integer.*valueOf*(xcl.*string*[r][2]);  xcl.*loc*=xcl.*string*[r][0];  }  }  xcl.*dic*.put(xcl.*string*[xcl.*string*.length-1][0], xcl.*sum*);  **for**(Object key :xcl.*dic*.keySet()) {  xcl.*pro*[xcl.*count*][0]=key.toString();  xcl.*pro*[xcl.*count*][1]=xcl.*dic*.get(key).toString();  xcl.*count*++;  }  xcl.*change*(xcl.*pro*);  System.***out***.println("按总人数从多到少排序省份：");  **for**(String[] s :xcl.*pro*) {  **for**(String r :s) {  System.***out***.println(r+" ");  }  }  xcl.*rank*(xcl.*string*);  System.***out***.println("各省份及其范围：");  **for**(String[] s :xcl.*rank*) {  **for**(String r :s) {  System.***out***.print(r+" ");  }  }  xcl.*location*=xcl.*pro*[0][0];  **for**(**int** j =0 ;j < xcl.*pro*.length;j++) {//遍历pro（已经排列好的省份数组）  **for**(**int** i=0;i<xcl.*rank*.length;i++) {//遍历rank（已经确定每个省份范围的数组）  **if**(xcl.*rank*[i][0].equals(xcl.*pro*[j][0])){  xcl.*exchange*(xcl.*string*,Integer.*valueOf*(xcl.*rank*[i][1]),Integer.*valueOf*(xcl.*rank*[i][2]));  }  **else** {  **continue**;  }  }  xcl.*location* = area;//将第三个参数的值赋给location  File file = **new** File(outputfile);  FileOutputStream out = **new** FileOutputStream(file);  OutputStreamWriter wr =**new** OutputStreamWriter(out);  wr.write(xcl.*location*+" "+xcl.*pro*[0][1]+"\r\n\r\n");  **for**(**int** n =0 ;n < xcl.*string*.length;n++) {  **if**(xcl.*string*[n][0].equals(xcl.*location*)){  wr.write(xcl.*string*[n][1]+" "+xcl.*string*[n][2]+"\r\n");  }  **else**{  **continue**;  }  }  wr.close();  }  }  }//构造参数数量为2时应该执行的函数  红色部分为待优化部分  优化后的代码：  package text1;  import java.io.File;  import java.io.FileOutputStream;  import java.io.IOException;  import java.io.OutputStreamWriter;  public class program6 {  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("参数输入错误，请重新输入");  }  }  public static void tip2(String inputfile,String outputfile) throws IOException {  xcl.readFileByLines(inputfile);  //将读出来的内容写入list数组中这一步被归纳进了readFileByLines（）中，后续直接调用：  xcl.pro();//将按照总数对省份进行排序这个步骤单独封装成一个函数pro()，然后直接调用  xcl.change(xcl.pro);  xcl.rank(xcl.string);  System.out.println("各省份及其范围：");  xcl.area();//将各个省份内的城市进行排序这个步骤单独封装成一个函数area()，然后直接调用  File file = new File(outputfile);  FileOutputStream out = new FileOutputStream(file);  OutputStreamWriter wr =new OutputStreamWriter(out);//进行输出  wr.write(xcl.location+" "+xcl.pro[0][1]+"\r\n\r\n");//先写入人数最多的省份加换行到文档  for(int m = 0;m < xcl.pro.length;m++) {  for(int i=0;i<xcl.string.length;i++) {  if(xcl.string[i][0].equals(xcl.pro[m][0])) {  wr.write(xcl.string[i][1]+" "+xcl.string[i][2]+"\r\n");  }  else {  continue;  }  }  if(m!=xcl.pro.length-1) {  xcl.location=xcl.pro[m+1][0];  wr.write(xcl.location+" "+xcl.pro[m+1][1]+"\r\n\r\n");  }  else  break;  }  wr.close();  }  public static void tip3(String inputfile,String outputfile,String area) throws IOException {  xcl.readFileByLines(inputfile);  xcl.pro();  xcl.change(xcl.pro);  xcl.rank(xcl.string);  System.out.println("各省份及其范围：");  xcl.area();  File file = new File(outputfile);  FileOutputStream out = new FileOutputStream(file);  OutputStreamWriter wr =new OutputStreamWriter(out);  xcl.location=area;//记得将第三个参数area的值赋给location，负责location的值会一直停留在广东省，之后无论输入哪个省份查找，都会输出广东省的信息  wr.write(xcl.location+" "+xcl.pro[0][1]+"\r\n\r\n");  for(int n =0 ;n < xcl.string.length;n++) {  if(xcl.string[n][0].equals(xcl.location)){  wr.write(xcl.string[n][1]+" "+xcl.string[n][2]+"\r\n");  }  else{  continue;  }  }  wr.close();  }  }  优化步骤：  1.将读出来的内容写入list数组中这一步被归纳进了readFileByLines（）中，之后只需调用读文件函数即可：    2.将按照总数对省份进行排序这个步骤单独封装成一个函数pro()     1. 将按照数量对各省份内城市进行排序的步骤单独封装成一个函数area()     传参运行成功：  一个参数：    两个参数：      三个参数：      四个参数：    3.Java项目生成jar包  （1）“项目”右键->"Export..."导出，弹出“导出设置”窗口如下：    图一、选择 JAR file  （2）选择要导出的项目，然后设置导出文件路径，选择“main”方法入口类：    图二、点击Browser，选择文件导出路径，这里要注意记得把源文件xcl.java一起导出,   1. 在cmd中检验six.jar是否能运行     图四、输入命令    图五、执行成功，说明该可执行文件可运行  4.使用exe4j生成可执行的.exe文件  （1）启动界面并注册    图五、注册  （2）选择打包方式    图六、"JAR in EXE" mode  （3）应用程序配置    图七、设置程序名称及exe文件输出后保存路径  （4）配置可执行文件：    图八、设置可执行文件名称    图九、根据自己的操作系统位数来配置  （5）配置运行环境    图十、添加JAR包，指定主程序    图十一、配置JRE的版本（1.8）   1. 把其它的所有的directory删除     图十二、删除“JAVA\_HOME”或“JDK\_HOME”    图十三、点击绿色“＋”来选择捆绑的JRE位置  5.生成可执行文件second.exe：    图十四、点击执行该文件      图十五、图十六运行成功，结果正确   1. 讨论分析   **曹丽：**对于这次实验，我对源代码进行了初步的封装，但是却没有在原来代码的基础上做出台啊改变，而通过组员吕林桃的优化，使得代码数量减少，避免了代码的大量冗余，而使原函数保密性变高了，泄露风险也降低了。而在封装过程中，我也遇到了许多问题，比如我忘记了用封装后的类中的第三个参数对location重新赋值，所以第三个参数的值就一直是pro[0][0]，导致后面我查询信息无论输入哪个省份，输出文件里面一直显示广东省，在我将将第三个参数area的值赋给location后，解决了这个问题。还有就是在导出的时候我没有导出源文件xcl.java,只导出了program6.java这个接口，所以无法运行，在组员的帮助下，我重新导出了这两个文件，成功解决了问题。这次实验使我意识到了个人的力量是有限的，在提升自己的同时，也要多多请教别人，学习别人的优点。  **吕林桃：**  github链接：  <https://github.com/caoli617/cl617>   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。