西南民族大学

**实验报告**

**2021------2022**学年第**1**学期

课程名称：软件工程

学院：计算机科学与工程 专业：软件工程

年级：2019级 班级：1902

学号：201931101096 姓名：李寒甘

同组人：范旭伟 201931101044

|  |
| --- |
| 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：BS-222 实验时间：2021 年 10 月 20日  姓名：李寒甘 专业：软件工程 班级：1902 学号:201931101096 |
| 实验项目名称：SE实践4 实验成绩： 教师签名：周绪川 |
| **一、实验目的**  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；   **二、材料与方法**  硬件：固态硬盘：256GB，CPU：Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz  内存：8.00 GB （RAM）  软件：操作系统:Windows 10 家庭中文版，VScode。  PC机一台、源文件yq\_in\_04.txt等 。  **三、实验主要过程与结果**  1. 解决思路：  在实验三的基础上，我们将代码分为了三个模块，即三个函数，三个函数的功能分别是读取数据，处理数据，和输出数据，读取数据除了负责将数据读入，还要对数据进行简单的处理，比如去除省份与待明确地区，以及最重要的是将数据处理成处理数据函数所需的格式，读数据模块还负责读取输出文件名以及选取的省份，处理好的数据进入处理数据函数，处理数据模块负责统计每个省份的城市数以及对数据进行排序，最后由输出数据函数写入文件。  2. 代码与解决流程（解决流程详见注释）：  *# coding=gbk*  *import* sys  def GetData():  *#下边这一块是为读数据做准备*      fileInPath = sys.argv[1]                    *# 读取文件的路径*      fileOutPath = "yq\_out\_04.txt"               *# 将输出文件默认置为yq\_out\_04.txt*  *if* len(sys.argv) >= 3:          fileOutPath = sys.argv[2]      provincialName = ""                         *# 将单独选取的省份默认置为yq\_out\_04.txt*  *if* len(sys.argv) >= 4:          provincialName = sys.argv[3]      fileContent = []  *# 下边这一块是读取数据*  *with* open(fileInPath, "r") as f:            *# 打开文件*          data = f.readlines()                    *# 读取文件中每行数据*  *for* fline in data:                      *# 循环读取每行数据*              fileContent.append(fline.split())   *# 将每行数据分割后再存入list中*  *# 下边这一块是为了去除待明确地区以及去除城市带的省份*      Temp = []      province = " "      fileContentFinal = []      default\_string = "待明确地区"  *for* fline in fileContent:  *if* fline[0] != province:                *# 如果读到的省份数据和之前记录的数据不同，则*              Temp.append(fline[0])               *# Temp中添加新省份*              fileContentFinal.append(Temp)       *# fileContentFina添加Temp*              Temp = []                           *# 将Temp置空*              province = fline[0]                 *# 更新省份*              Temp.append(fline[1])               *# Temp中添加城市与数量*              Temp.append(fline[2])              fileContentFinal.append(Temp)       *# fileContentFina添加Temp*              Temp = []  *elif* fline[1] != default\_string:        *# 省份相同，则只存后边的城市与数字，加个判断是为了去除“待明确地区”*              Temp.append(fline[1])              Temp.append(fline[2])              fileContentFinal.append(Temp)              Temp = []  *# 下边这一块是为了将省份将每个省份与属于这个省份的城市防到一个list中，再将这些list放入一个大list中*      fileContentTemp = []      fileContentFinal2 = []  *for* i in range(len(fileContentFinal)):          fileContentTemp.append(fileContentFinal[i])  *if* i == len(fileContentFinal)-1 or (i < len(fileContentFinal)-1 and len(fileContentFinal[i+1]) == 1):              fileContentFinal2.append(fileContentTemp)              fileContentTemp = []  *# 最后返回处理好的结构的读取到的数据以及输出文件名与选取特定省份名*  *return* fileContentFinal2, fileOutPath, provincialName  def ProcessingData(*fileContent*):  *# 这个循环是统计每个省份的城市数，再将数字加到包含省份名的list中*  *for* i in range(len(*fileContent*)):  *fileContent*[i][0].append(str(len(*fileContent*[i])-1))  *# 下边是为按省份的城市数从大到小排序，如果城市数相同，则按省份名字母从大到小排序*  *for* i in range(len(*fileContent*)):  *for* j in range(len(*fileContent*)):              temp = []  *if* int(*fileContent*[i][0][1]) > int(*fileContent*[j][0][1]):                  temp = *fileContent*[i]  *fileContent*[i] = *fileContent*[j]  *fileContent*[j] = temp  *if* int(*fileContent*[i][0][1]) == int(*fileContent*[j][0][1]):  *if* *fileContent*[i][0][0] > *fileContent*[j][0][0]:                      temp = *fileContent*[i]  *fileContent*[i] = *fileContent*[j]  *fileContent*[j] = temp  *# 下边是为城市数从大到小排序，如果城市数相同，则按省份名字母从大到小排序*  *for* i in range(len(*fileContent*)):  *for* j in range(len(*fileContent*[i]))[1:]:              temp = []  *for* k in range(len(*fileContent*[i]))[1:]:  *if* int(*fileContent*[i][j][1]) > int(*fileContent*[i][k][1]):                      temp = *fileContent*[i][j]  *fileContent*[i][j] = *fileContent*[i][k]  *fileContent*[i][k] = temp  *if* int(*fileContent*[i][j][1]) == int(*fileContent*[i][k][1]):  *if* *fileContent*[i][j][0] > *fileContent*[i][k][0]:                          temp = *fileContent*[i][j]  *fileContent*[i][j] = *fileContent*[i][k]  *fileContent*[i][k] = temp  *# 返回处理好的数据*  *return* *fileContent*  def OutData(*fileContent*, *fileOutPath*, *provincialName*):  *# 写入文件*  *with* open(*fileOutPath*, "w") as f:  *for* i in range(len(*fileContent*)):  *# 如果要选定特定省份的情况*  *if* len(*provincialName*) > 0:  *if* *fileContent*[i][0][0] == *provincialName*:  *for* j in range(len(*fileContent*[i])):  *if* len(*fileContent*[i][j]) == 1:                              f.write(*fileContent*[i][j][0]+'\n')  *elif* len(*fileContent*[i][j]) == 2:                              f.write(*fileContent*[i][j][0]+'\t'+*fileContent*[i][j][1]+'\n')                  f.write('\n')  *# 如果不选定特定省份的情况*  *else*:  *for* j in range(len(*fileContent*[i])):  *if* len(*fileContent*[i][j]) == 1:                          f.write(*fileContent*[i][j][0]+'\n')  *elif* len(*fileContent*[i][j]) == 2:                          f.write(*fileContent*[i][j][0]+'\t'+*fileContent*[i][j][1]+'\n')                  f.write('\n')  *if* \_\_name\_\_ == '\_\_main\_\_':  *# 首先运行读取数据模块*      fileContent, fileOutPath, provincialName = GetData()      print(fileContent)      print(fileOutPath, provincialName)  *# 然后运行处理数据模块*      fileContent = ProcessingData(fileContent)      print(fileContent)  *# 最后运行输出数据模块*      OutData(fileContent, fileOutPath, provincialName)  3. 运行结果：  当不指定输出文件名与省份时：    图 3.1 读取数据函数的结果与数据处理函数的结果    图 3.2 写入文件结果  当指定输出文件名与省份时：    图 3.3 读取数据函数的结果与数据处理函数的结果    图 3.4 写入文件结果  将文件上传至GitHub：  图 3.5 Github截图  5.个人Git链接：https://github.com/rrtaitan/Lihangan-work.git  **四、分析讨论**  通过本次实验，我深刻的认识到面向对象编程的优势所在，对于同一个功能，不同的人写出的代码差别很大，在大型的程序编写过程中就会出现代码不兼容的问题，往往解决这个问题所花费的时间比我们写功能模块所花的时间还要多。这次结对编程使我们认识到团队合作的重要性和难点。我们之前编写代码只为了能实现功能，没有考虑代码的整合问题，导致很多代码在软件工程项目开发中失去意义。编写代码要采用面向对象的思想，这样才能让我们自己写的代码和其他人的代码能够相互调用，共同完成一个软件工程项目。  **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。