西南民族大学

**实验报告**

**2021------2022**学年第**2**学期

课程名称：软件工程课程设计

学院：计算机科学与工程 专业：计算机科学与技术

年级：2019级 班级：1903

学号：201931101116 姓名：林儒德

同组人：无

|  |
| --- |
| 西南民族大学学生实验报告  教学单位：计算机科学与工程学院 实验室名称：BS-223 实验时间：2022年3月17日  姓名：林儒德 专业：计算机科学与技术 班级：1903 学号:201931101116 |
| 实验项目名称：SE实践3、4 实验成绩： 教师签名：周绪川 |
| **一、实验目的**  **实验3：**在实验二的基础上继续：  1、输入文件名在命令行输入（即输入文件可以指定）；输出文件名在命令行输出（即输出文件可以指定）；  2、可以输出指定省的信息  具体示例：  >yq yq\_in\_03.txt yq\_out\_03.txt  //表示输入文件名为yq\_in\_03.txt，输出文件名为yq\_out\_03.txt，包括所有省份信息，格式和实验二一致  >yq yq\_in\_03.txt yq\_out\_03.txt 浙江省  //输出文件中仅包括浙江省的信息  **实验4：**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**  PC机一台、Python环境。  **三、实验主要过程与结果**  实验3：  实验3在实验2的基础上多了对命令参数的分析。在Python中可以使用sys.argv变量来提取命令的参数，具体如图1所示。    图1  数据的处理过程与实验2一致。不同的是最后的输出部分。输出时要先判断命令中是否指定了省份信息，如果没有指定，则与实验2的格式一致。如果指定了省份，则需要另外输出，具体如图2所示。    图2  第41-46是没有指定省份的情况（PROV is None），实现输出的代码与实验2一致。如果指定了输出省份（第47行，else PROV is not None），则输出部分为代码的第48-52行。  测试代码结果如图3所示，在这里指定了输出的省份为浙江省。    图3  实验4：  该实验中增加了对省份总数的统计，以及排序输出的要求。省份总数的统计如图4所示。    图4  原先的统计数据的字典格式为：{‘prov1’: [[‘city1’, num1], [], …, []], ‘prov2’: [[‘city1’, num1], …], …} 。目标是把改字典的key的格式改成 (‘prov1’, num1) 的形式，方便后续的省份排序。首先遍历原字典的keys（第41行），设置一个变量来累计求和（第42行）。遍历该省份的所有城市，并把总数进行求和（第43-44行）。然后按照设定好的格式构造键（第45行），并把这个键存到一个临时字典中（第46行）。最后用第48行的语法构造目标的字典格式。  排序输出的要求可以很方便地用sorted() 函数来进行实现。在这里把输出和排序给归到了一起来进行实现，如图5所示。    图5  对new\_dict中的键进行排序的方法是sorted(new\_dict.keys(), key=lambda x: int(x[1]), reverse=True)，其中函数中的key参数指明了要按照什么标准进行排序。在这里设置为lambda x: int(x[1])，也就是按照数值来进行排序，reverse=True表明为降序排序。城市部分的排序也类似。再分为未指定省份（PROV is None）和指定了省份（PROV is not None）来分别进行输出。  对该程序进行测试，可以看到未指定省份的情况下按照省份的降序进行排列，并且省份内部的城市也是降序排列的，如图6所示。    图6  再来测试指定省份，如图7所示，城市之间也是按照降序进行输出。    图7  将程序封装成exe后，上传至GitHub。  Github链接：https://github.com/iwajef/21-22-2-SE  **四、分析讨论**  本次实验使用Python实现了命令参数的分析，设计了不同的数据结构，使问题能高效地解决，对排序方法也进行了回顾，为后续更复杂的软件实现打下了基础。  **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。