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

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：1902

学号：201931101319 姓名：张洁琴

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 14 日  姓名：张洁琴 专业：计算结科学与技术 班级：1902 学号:201931101319 |
| 实验项目名称：个人编程实践3 实验成绩： 教师签名：周绪川 |
| 一、实验目的 1.锻炼个人编程和对问题的分析，解决能力；  2.进一步熟悉git的使用。 二、材料与方法 1. 搭载Windows系统的电脑；  2. GitHub；  3. Python。 三、实验主要过程与结果 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  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：  每个省后面有一个总数;  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；  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；  代码实现：  import pinyin  import sys  import os  class Area:  def \_\_init\_\_(self, name, num=0):  self.name = name  self.num = num  # 重载大于号和小于号，方便sort函数的调用  def \_\_lt\_\_(self, other):  if self.num != other.num:  return self.num < other.num  else:  # 使用pinyin库就不用写枚举了  return pinyin.get(self.name) < pinyin.get(other.name)  def \_\_gt\_\_(self, other):  if self.num != other.num:  return self.num > other.num  else:  return pinyin.get(self.name) > pinyin.get(other.name)  class Province(Area):  def \_\_init\_\_(self, name, cities=None, num=0):  super().\_\_init\_\_(name, num)  if cities is None:  self.cities = []  def addCity(self, city):  self.num += city.num  self.cities.append(city)  class City(Area):  def \_\_init\_\_(self, name, num, province):  super().\_\_init\_\_(name, num)  self.province = province  def \_\_str\_\_(self):  return '城市名：' + self.name + '\t感染人数：' + str(self.num)  def main():  cities = []  filename = "./yq\_in.txt"  if len(sys.argv) > 1:  filename = sys.argv[1]  # filename = sys.argv[1]  f = open(filename, 'r', encoding='gbk')  line = f.readline()  while line:  oneline = line.split('\t')  province = oneline[0]  city = oneline[1]  num = oneline[2]  cities.append(City(name=city, num=int(num), province=province))  line = f.readline()  f.close()  # 定义一个列表来存储省份对象  provinces = []  province\_temps = [Province(name=None, num=0)]  for i in cities:  if i.province == province\_temps[0].name:  province\_temps[0].cities.append(i)  province\_temps[0].num += i.num  else:  if province\_temps[0].name is not None:  provinces.append(province\_temps[0])  province\_temps[0] = Province(name=i.province, num=i.num)  province\_temps[0].cities.append(i)  provinces.sort(reverse=True)  for i in provinces:  # print(i.name + "\t 感染人数合计: " + str(i.num))  i.cities.sort(reverse=True)  # for j in i.cities:  # print('\t', j)  #数据结构化存储完毕并排好序，开始输出到文件  # 统计命令行参数个数  num\_argv = len(sys.argv)  if num\_argv <= 2:  new\_filename = filename.replace("in", "out")  else:  new\_filename = sys.argv[2]    if os.path.exists(new\_filename):  os.remove(new\_filename)  f = open(new\_filename, 'a', encoding='gbk')  for i in provinces:  if num\_argv > 3:  if i.name != sys.argv[3]:  continue  f.write(i.name + '\t' + str(i.num) + '\n')  for j in i.cities:  f.write(j.name + '\t' + str(j.num) + '\n')  f.write('\n')  if \_\_name\_\_ == '\_\_main\_\_':  main()  结果：  三个参数：    四个参数：      把实验代码文件传到GitHub   四、分析讨论 通过本次实验让我学会了通过cmd命令行去编译运行Python代码，同时学会了使用argv[]得到命令行输入的指令。对Python的文件读写操作有了更深的了解。 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。