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：2019级 班级：1902

学号：201931101138 姓名：卢澜升

同组人：无

|  |
| --- |
| 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-226 实验时间：2021年 10 月 21 日  姓名：卢澜升 肖振威 专业：软件工程 班级：1902 学号:201931101138  20193110 |
| 实验项目名称：SE实践3 实验成绩： 教师签名：周绪川 |
| **一、实验目的**  把之前的程序进行封装  1、单独运行yq.exe或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）；  **二、材料与方法**  PC机一台、GitHub、vc++ 。  1.实验环境  （1）语言要求  C++  （2）开发环境  内存8G及以上。  （3）工具  Vc++  **三、实验主要过程与结果**  解决思路：对于前几次的实验所写的对于yq\_in\_xx.txt的读入，按要求输出等代码进行封装，并且使用功能块的封装并且继承相应功能完成本次实验中的按要求排序或输出特定省份以及下属城市功能。  2.编程  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；   （3）流程  创建城市类，内部成员为按行读入yq\_in\_04.txt中的城市名称以及人数。  创建省份类，内部成员为城市类数组，省份排序函数，人数统计函数，人数排序函数，以及成员属性：城市，省份总人数。  创建country类，内部成员为建立省份类数组，使用封装好的排序函数以及按要求输出函数  对整体的输入文件进行存储后，按要求操作排序，最后根据在命令行输入文件区别产生不同输出文件。  （4）代码  #include<bits\stdc++.h>  using namespace std;  static int Pptr = 1;  class City{  public:  string Cname;  int Ctot;    public:  City(){};  City(string cname, int ctot) {  Cname = cname, Ctot = ctot;  }  };  class Province{  public:  string Pname;  int Ptot;  int Cnt;  City city[1000];  public:  Province(){};  Province(string pname){  Ptot = 0, Pname = pname;  }  void getPtot(int tot){  Ptot += tot;  }  void sort\_city(Province pro, int cnt);  friend bool operator<(const Province &u, const Province &v){  if(u.Ptot == v.Ptot) return u.Pname < v.Pname;  return u.Ptot > v.Ptot;  }  };  class Country {  public:  Province province[100];    public:  Country(){};  void Sort();  void Show1();  void Show2();  };  Country country;  string res = "", infile = "", outfile = "";  ifstream in;  ofstream out;  void Province :: sort\_city(Province pro, int cnt){  for(int j = 1; j < cnt; j++){  for(int k = j + 1; k < cnt; k++){  string tem = "";  int num = 0;  if(pro.city[j].Ctot < pro.city[k].Ctot){  num = pro.city[k].Ctot;  pro.city[j].Ctot = pro.city[k].Ctot;  pro.city[k].Ctot = num;  }  if(pro.city[j].Ctot == pro.city[k].Ctot){  if(pro.city[j].Cname > pro.city[j].Cname){  tem = pro.city[j].Cname;  pro.city[j].Cname = pro.city[k].Cname;  pro.city[k].Cname = tem;  }  }  }  }  }  void Country :: Sort(){  sort(country.province + 1, country.province + Pptr);  for(int i = 1; i < Pptr; i++){  country.province[i].sort\_city(country.province[i], country.province[i].Cnt);  }  }  void Country :: Show1(){  for(int i = 1; i < Pptr; i++){  out << country.province[i].Pname << " " << country.province[i].Ptot << endl;  for(int j = 1; j < country.province[i].Cnt; j++){  out << country.province[i].city[j].Cname << " " << country.province[i].city[j].Ctot << endl;  }  out << endl;  }  }  void Country :: Show2(){  for(int i = 1; i < Pptr; i++){  if(country.province[i].Pname == res){  out << country.province[i].Pname << " " << country.province[i].Ptot << endl;  for(int j = 1; j < country.province[i].Cnt; j++){  out << country.province[i].city[j].Cname << " " << country.province[i].city[j].Ctot << endl;  }  break;  }  }  }  int main(int argc, char \*argv[]){  int ok = 0, flag = 0;  if(argc == 1){  ok = 1;  }  else{  for(int i = 0; i < strlen(argv[2]); i++){  if(argv[2][i] == '.'){  ok = 1;  break;  }  }  }  if(ok == 0) res = argv[2];  infile = "yq\_in.txt", outfile = "yq\_out.txt";  in.open(infile);  out.open(outfile);  string P, C, tem;  int num = 0;  int Cptr = 1;  while(in >> P >> C >> num){  if(P != tem){  if(tem.length() != 0) country.province[Pptr - 1].Cnt = Cptr;  country.province[Pptr].Pname = P;  tem = P;  Pptr++;  Cptr = 1;  }  if(num == 0) continue;  int id = Pptr - 1;  country.province[id].getPtot(num);  country.province[id].city[Cptr].Cname = C;  country.province[id].city[Cptr].Ctot = num;  Cptr++;  }  country.province[Pptr - 1].Cnt = Cptr;  country.Sort();  if(ok == 1){  country.Show1();  }  else{  country.Show2();  }  return 0;  }   1. 测试实验要求： 2. 直接执行exe文件 无参输出     双击执行    出现输出文件    按要求排序后输出  2.输入yq\_in\_04.txt yq\_out\_04.txt  输出所有省份并按照每个省份的总数据人数排序    输入命令  相应文件夹下出现输出文件    输出yq\_out\_04.txt  查看输出文件是否排序    图三 图四  输入yq\_in\_04.txt yq\_out\_04.txt xx省份  输出文件中筛选出特定省份并且省份内部城市排序    图五 查询广东省  查看对应文件：    图六  完成排序:    图七  查询湖南省：    图八    图九 查询湖南省输出结果  上传到github      上传完成  **四、分析讨论**  在进行输出单一省份后，检查该省份下属城市是否排序时发现并没有达到要求，对代码进行查验后发现问题为，将输入文件储存后的操作并没有指向储存文件的地址，即封装的函数操作中指向的是函数名而非函数名地址，若不指向地址则输出对应文件时将无法修改省份内城市排序  （即城市排序中的pro变量名前应加取地址符指向操作地址）。  本次实验中要求的封装函数以及建立类并继承等操作均为第一次实践，对于功能块封装的可继承性与可复用性有了新的认识，加深了理解。  **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。