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：计算机科学与工程学院 专业：计科

年级：2019级 班级：1902

学号：201931101207 姓名：唐鑫

同组人：冯晓

|  |
| --- |
| 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：2021 年 3 月 25 日  姓名：唐鑫 专业：计科 班级：1902 学号:201931101207 |
| 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**  1.封装代码，使其能实现三个功能：   1. 单独运行yq.exe或者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] [指定省份]；（按要求排序）   并设计单元测试。  2.要求：   1. 每个省后有一个总数； 2. 输出省按总数从大到小排序。若两个省总数相同，按拼音（字母）排序； 3. 每个省内各市从大到小排序。若两个市总数相同，按拼音（字母）排序；   **二、材料与方法**  PC机一台、Dev-C++工具等 。  **三、实验主要过程与结果**  1.解决思路  ①使用C++中的类进行封装处理；  ②通过结构体代替原来的数组来存储文件中的变量并增添一个变量类型cnt用来存储各个省中的总数；  ③用int main(int argc,char \*argv[])来控制输入并进行分流进入相应功能块。  2.流程  ①将源代码用类封装起来；  ②声明定义类的各种属性和方法；  ③用int main(int argc,char \*argv[])来读取输入内容并分流；  ④若为缺省输入文件，则直接输出文件；  ⑤若指定了输入输出文件，则按要求进行排序后输出；  ⑥若制定了输入输出文件以及规定了省份，则将该省信息按要求进行排序后输出。  3.代码  #include<iostream>  #include<fstream>  #include<string>  #include<algorithm>  using namespace std;  class node  {  public:  string prov;  string place;  int sum;  int cnt;  };  class imf  {  private:  string src;  string dest;  string pro;  public:  ifstream sfile;  ofstream dfile;  node yq[129];  imf(string a,string b);  imf(string a,string b,string c);  ~imf();  void cinAll();  void traversal();  void sortAll();  void coutProv(string s);  void coutAll();    };  imf::imf(string a,string b)  {  src = a;  dest = b;  sfile.open(src.c\_str());  if(!sfile)  {  cout<<"error opening source file."<<endl;  }  dfile.open(dest.c\_str());  if(!dfile)  {  sfile.close();  cout<<"error opening destination file."<<endl;  }  }  imf::imf(string s,string d,string p)  {  src = s;  dest = d;  pro = p;  sfile.open(src.c\_str());  if(!sfile)  {  cout<<"error opening source file."<<endl;  }  dfile.open(dest.c\_str());  if(!dfile)  {  sfile.close();  cout<<"error opening destination file."<<endl;  }  }  imf::~imf()  {  sfile.close();  dfile.close();  }  void imf::cinAll()  {  string s = yq[0].prov;  int c = 1 , j = 0 ;  for(int i = 0 ; i < 129 ; i++)  {  sfile >> yq[i].prov >> yq[i].place >> yq[i].sum;  if(s == yq[i].prov)  {  yq[i].cnt = 0;  c++;  }  else  {  yq[j].cnt = c;  c = 1;  j = i;  s = yq[i].prov;  }  }  yq[j].cnt = c;  }  void imf::traversal()  {  string s = yq[0].prov;  int c = yq[0].cnt , j=0;  for(int i = 0 ; i < 129 ; i++)  {  if(s == yq[i].prov)  {  yq[i].cnt = c;  }  else  {  s = yq[i].prov;  j = i;  c = yq[i].cnt;  }  }  }  bool cmp(node a,node b)  {  if(a.prov == b.prov)  {  if(a.sum != b.sum)  {  return a.sum > b.sum;  }  else  {  return a.place < b.place;  }  }  else  {  if(a.cnt != b.cnt)  {  return a.cnt > b.cnt;  }  else  {  return a.prov < b.prov;  }  }  }  void imf::sortAll()  {  sort(yq,yq+129,cmp);  }  void imf::coutProv(string s)  {  dfile << s << " ";  int f = 0;  for(int i = 0 ; i < 129 ; i++)  {  if(yq[i].prov == s)  {  if(f == 0)  {  dfile << yq[i].cnt << "个地区" <<endl;  f = 1;  }  dfile << yq[i].place << " " << yq[i].sum << endl;  }  }  }  void imf::coutAll()  {  string s = yq[0].prov;  int f = 0;  for(int j = 0 ; j < 129 ; j++)  {  if(yq[j].prov == s && f == 0)  {  dfile << yq[j].prov << " " << yq[j].cnt << "个地区" <<endl;  f = 1;  dfile << yq[j].place << " " << yq[j].sum <<endl;  }  else if(yq[j].prov == s && f == 1)  {  dfile << yq[j].place << " " << yq[j].sum <<endl;  }  else if(yq[j].prov != s)  {  s = yq[j].prov;  dfile << endl;  dfile << yq[j].prov << " " << yq[j].cnt << "个地区" <<endl;  dfile << yq[j].place << " " << yq[j].sum <<endl;  }  }  }  int main(int argc,char \*argv[])  {  if(argc == 1)  {  imf im1("yq\_in.txt","yq\_out\_01.txt");  im1.cinAll();  im1.coutAll();    }  else if(argc == 4)  {  imf im2(argv[1],argv[2],argv[3]);  im2.cinAll();  im2.traversal();  im2.sortAll();  im2.coutProv(argv[3]);  }  else  {  imf im3(argv[1],argv[2]);  im3.cinAll();  im3.traversal();  im3.sortAll();  im3.coutAll();  }    return 0;  }  4.测试结果  ①命令行输入指令：    图 1 命令行输入指令  ②输出文件：    图 2 输出文件  ③yq\_out\_01.txt文件预览：    图 3 yq\_out\_01.txt文件  ④yq\_out\_02.txt文件预览：    图 4 yq\_out\_02.txt文件  ⑤yq\_out\_03.txt文件预览：    图 5 yq\_out\_03.txt文件  5.个人git链接：https://github.com/XXZAXX/work.git  **四、分析讨论**  由于前三次实验都是用数组来存储文件中的变量，导致这次实验中多了一个统计值不知道该怎么和省份捆绑在一起，所以另起炉灶重新用结构体来存储变量。在用结构体进行编程的时候也遇到了问题，我一开始没有写traversal函数让每个结构体的cnt都为省的总数而是设置成0，这导致在后续的排序中出现了不少问题。  在类的封装过程中，一开始没反应过来定义是需要添加作用域的，也耽误了不少工作，但后来更正过来后边一帆风顺了。  通过此次实验，让我对类的封装有了进一步理解。  **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。