源代码(若出现bug,请老师删掉注释),分文件

（输stu\_info.txt文件时输入文件所在计算机中的地址）

1. ”main.c”源代码文件

/\*C++中cin>>类似scanf,cout<<类似于printf，endl类似于回车换行\*/

#include<iostream>//c++头文件

#include<vector>

#include<string>

#include<windows.h>

#define sec 1000

enum{CN,Math,Eng,Sci,PE};//枚举类型

#include"main.h"

#include"create\_info.h"

#include"add\_info.h"

#include"Print\_all.h"

#include"modify\_info.h"

#include"search\_info.h"

#include"delete\_info.h"

#include"Na\_Sort\_Print.h"

#include"Sum\_Sort\_Print.h"

#include"save\_eixt.h"

using namespace std;//相当于c++固定格式

void main\_page(vector<student> &T);//相当于传头地址

int main()

{

vector<student> per;//兼有链表与数组的特性的容器；

cout<<"Welcome to the Student management system"<<endl;

Sleep(1.2\*sec);

main\_page(per);

cout<<"input number :";

string filename;//string相当于动态字符串数组

long int N;

cin>>N;

do{

switch(N){

case 1:

filename=create\_info(per);

if(filename=="null") filename.clear();//清空filename里面内容

break;

case 2:add\_info(per,filename);break;

case 3:Print\_all(per);break;

case 4:modify\_info(per);break;

case 5:search\_info(per);break;

case 6:delete\_info(per);break;

case 7:Na\_Sort\_Print(per);break;

case 8:Sum\_Sort\_Print(per);break;

case 9:save\_eixt(per,filename);break;

default :cout<<"Input is incorrect,please reenter"<<endl;

}

Sleep(0.1\*sec);//window.h头文件里面的停顿函数

main\_page(per);

cout<<"input number :";

}while(cin>>N);

return 0;

}

void main\_page(vector<student> &T)//c++中带&为引用,相当于传入首地址

{

cout<<"========================================"<<endl

<<" Student management system "<<endl

<<"========================================"<<endl;

cout<<" 1---create\_info 2---add\_stu\_info"<<endl

<<" 3---Browse\_info 4---modify\_info"<<endl

<<" 5---search\_info 6---delete\_info"<<endl

<<" 7---Na\_Sort\_Print 8---Sum\_Sort\_Print"<<endl

<<" 9---save\_and\_exit Student Total :"<<T.size()<<endl

<<"========================================"<<endl;

}

1. ”main.h”源代码文件

#include<windows.h>

using namespace std;//可以理解为固定格式

string vi\_na();//函数声明；

int snum();//函数声明；

class student{//一个类,类似于结构体但于结构体不同

protected://保护性成员,只允许被类里面函数或带有friend的函数访问

int sum;

private://私有成员只允许被类里面的函数或者带有friend函数访问

string name;

string snum;

public://公有成员,相当于struct里面的变量,public中变量能被外部函数修改,外部函数即非类里面的函数

char sex;

int YR,M,D;//年月日

int score[5];

const string vi\_snum(){return snum;}//用来访问私有成员的接口函数

const string vi\_na(){return name;}//用来访问私有成员的接口函数

SetBa(string x,string s){

name=x;

snum=s;

}//私有成员初始化函数

Soe\_Sum(int N){sum=N;}//访问保护成员Sum的接口函数

Soe\_Sum(){int n=0;for(int i=0;i<5;i++) n+=score[i];sum=n;}//接口函数,算出总分并赋值给sum

const int vi\_sum(){return sum;}//常量函数,不能被修改

friend bool operator <(const student &T,const student &V);//友元函数，能访问类里面的私有成员(private)

friend bool operator <=(const student &T,const student &V);//友元函数，能访问类里面的私有成员(private)

};

void Print(int cnt,int S=1,string x="loading")//参数为string类(类似动态字符数组),int类

{

cout<<x;

for(int i=0;i<cnt;i++){

Sleep(S\*sec);//Sleep为windows.h里面的函数

cout<<".";

if(i==cnt-1) cout<<endl;

}

}

bool operator <(const student &T,const student &V)//重载运算符 < ,使运算符 < 具有更多功能

{

if(T.name<V.name) return true;//重载内容;相当于strcmp(T.name,V.name)

else return false;

}

bool operator <=(const student &T,const student &V)//同上

{

if(T.name<=V.name) return true;

else return false;

}

1. ”create\_info.h”源代码文件

#include<fstream>

#include<cstdio>

using namespace std;

string create\_info(vector<student> &per)//返回类型为string(动态字符串数组)类型

{

if(!per.empty()){//如果per表空,即per里面元素个数为0,则执行if里面内容(元素个数是指学生数)

cout<<"The data has already existed.no need to create"<<endl;

Print(6,0.5,"returning");

string filename;

filename="null";

return filename;//返回filename的内容

}

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*create\_info\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"--------------------------------------"<<endl;

cout<<"| 1---Input data from the keyboard |"<<endl;

cout<<"| 2---Read data from a file |"<<endl;

cout<<"--------------------------------------"<<endl;

int N;

cout<<"input number in front of the operation :";

cin>>N;//类似于scanf

string filename;//定义string类型

if(N==1){//手动输入数据

filename="stu\_info.txt";//文本文件名字及文件所在的计算机中的地址,不输入地址默认为同一目录下

fstream infile(filename.c\_str(),ios::out);//将filename文件清空,用写的方式打开filename文件

if(!infile){

cerr<<"the file has been created to record the data"<<endl;//cerr是向屏幕输出错误信息;

}//如果没有找到,则报错并且新建一个名为filename的文本文件,用来记录信息

int cnt;

cout<<"input number of student :";

cin>>cnt;

for(int i=0;i<cnt;i++){

getchar();

cout<<i+1<<" student :"<<endl;

student stu;//定义了一个student类的stu对象,类似定义结构体

string Name;

cout<<"input stu name.After confirmation,It'll not be changed(eg.Xiao Ming) :";

getline(cin,Name,'\n');//类似于gets,按回车结束输入

cout<<"input sex(B or G) :";

cin>>stu.sex;

string Snum;

int Score[5];

cout<<"input 5 scores"<<endl;

for(int i=0;i<5;i++){

cin>>stu.score[i];

}

cout<<"input snum(After confirmation,It'll not be changed) :";

cin>>Snum;

stu.SetBa(Name,Snum);//调用类里面自定义的初始化函数

cout<<"Input date of birth(eg.2018 01 02) :";

cin>>stu.YR>>stu.M>>stu.D;//从键盘输入年,月,日

infile<<stu.vi\_na()<<endl<<stu.sex<<" "<<stu.vi\_snum()<<" ";//向filename文件中输入信息和空格,这里的infile类似于fprintf

infile<<stu.YR<<" "<<stu.M<<" "<<stu.D<<" ";//infile类似于fprintf

for(int i=0;i<5;i++){

infile<<stu.score[i];

if(i!=4) infile<<" ";//infile类似于fprintf

}

per.push\_back(stu);//向per容器中按先后顺序放入一个student类型(类似于放一个结构体),

Name.clear();//清空Name里面的元素

}

infile.close();//关闭filename文件;

}

else if(N==2){//从文件读入数据

cout<<"input file name :";

cin>>filename;//输入文件名

fstream outfile(filename.c\_str(),ios::in);//用读的方式打开文件

if(!outfile){//若找不到文件,则返回主页面并返回一个字符串;

cerr<<"Not Found,input number in front of the operation"<<endl;

return "null";

}

while(outfile.tellg()!=EOF){//读指针没有到文件末尾时,执行循环

student stu;//定义student类

string Na;char sex;

string snum;//动态字符数组

int score[5];

getline(outfile,Na,'\n');//从文件中读取字符串到Na中

outfile>>stu.sex>>snum;//从文件中读取数据中,类似于fscanf

outfile>>stu.YR>>stu.M>>stu.D;//从文件中读取数据中,类似于fscanf

stu.SetBa(Na,snum);//调用初始化函数来初始化stu对象里面的name和snum

for(int i=0;i<5;i++){

outfile>>stu.score[i];//读入5个成绩,类似于fscanf

}

per.push\_back(stu);//完成从文件中一个学生信息的读入,并将这个学生信息放入per容器里面

}

outfile.close();//关闭文件

}

Print(7,1);

cout<<"Initialization success"<<endl;//输出初始化成功

Sleep(sec);//停顿函数

return filename;//返回文件名

}

1. ”add\_info.h”源代码文件

void add\_info(vector<student> &per,const string &filename){

getchar();

if(per.empty()){

cout<<"you need to create students'info"<<endl;

Sleep(1.1\*sec);

Print(4,1,"returning");

return;

}

here:

student stu;//定义student类

string Name;//(动态字符串数组)

string Snum;//(动态字符串数组)

Print(4);

cout<<"input stu name.After confirmation,It'll not be changed(eg.Xiao Ming) :";

getline(cin,Name,'\n');//输入名字,空格键结束输入;

cout<<"input snum(After confirmation,It'll not be changed) :";

cin>>Snum;//输入学号

int K;

for(K=0;K<per.size()&&Snum!=per[K].vi\_snum()&&Snum!=per[K].vi\_na();K++);//K<per容器中元素个数并且姓名和学号不等于每一个;功能为学生信息查重

if(K!=per.size()){//K不等于per中元素个数

cout<<"Student information has already existed"<<endl

<<"-------------------------------"<<endl

<<"| 1---try again |"<<endl

<<"| 2---return to main page |"<<endl

<<"-------------------------------"<<endl;

int P;

cout<<"input number :";

cin>>P;

switch(P){

case 1:goto here;break;

case 2:return;

}

}

cout<<"input sex(B or G) :";

cin>>stu.sex;//输入性别

int Score[5];

cout<<"input 5 scores"<<endl;

for(int i=0;i<5;i++){

cin>>stu.score[i];

}

stu.SetBa(Name,Snum);//调用student类里面的初始化函数对stu里面的name和snum进行赋值

cout<<"Input date of birth(eg.2018 01 02) :";

cin>>stu.YR>>stu.M>>stu.D;//printf

per.push\_back(stu);//将输入完信息的学生存进per容器末尾;

cout<<"loading";

for(int i=0;i<6;i++){

Sleep(sec);

cout<<".";

if(i==5) cout<<endl;

}

fstream infile(filename.c\_str(),ios::app);//以写和不清空文件的形式打开文件名为filename的文件

getline(infile,Name,'\n');

infile<<" "<<stu.sex<<" "<<Snum<<" ";//这里的infile类似fprintf

infile<<stu.YR<<" "<<stu.M<<" "<<stu.D<<" ";

for(int i=0;i<5;i++){

infile<<stu.score[i]<<" ";

}

infile<<endl;//向文件读入回车

cout<<"added successfully"<<endl;

infile.close(); //关闭文件

}

1. ”Print\_all.h”源代码文件

#include<iomanip>

#include<cstdlib>//兼容了stdlib.h

#include<cstdio>//兼容了stdio.h

void Print\_all(vector<student> &T)//类似于传T容器的首地址(有数组与链表的特性),功能为

{

if(T.empty()){//若T容器元素个数为0

cout<<"there is no students' information"<<endl;

Sleep(1.1\*sec);//window.h中的停顿函数

Print(4,1,"returning");

return;

}

cout<<"-----------------------------------"<<endl//<<endl为输出空格

<<"| Please select the output format |"<<endl

<<"-----------------------------------"<<endl

<<"| 1---Simple mode |"<<endl

<<"| 2---Classic mode |"<<endl

<<"-----------------------------------"<<endl;

//以下为两种不同的输出方式

here:

cout<<"input number :";

int E;

cin>>E;

if(E==1){

printf("\tName\t Snum sex year month day CN\tMath English Science PE\n");

for(int i=0;i<T.size();i++){

cout.width(13);////设置宽度为13

cout<<T[i].vi\_na();////调用类里面访问私有成员name的接口函数

cout.width(15);

cout<<T[i].vi\_snum();////调用类里面访问私有成员snum的接口函数

printf(" %c %3d %02d %02d ",T[i].sex,T[i].YR,T[i].M,T[i].D);//头文件cstdio

for(int j=0;j<5;j++){

printf("%02d\t",T[i].score[j]);

}

cout<<endl;//输出空格

}

}

else if(E==2){

for(int i=0;i<T.size();i++){

cout<<i+1<<"th information :"<<endl

<<" Name :"<<T[i].vi\_na()<<endl

<<" Sex :"<<T[i].sex<<endl

<<" Snum :"<<T[i].vi\_snum()<<endl;

cout<<" birth :";

cout<<setfill('0')<<setw(2)<<T[i].YR<<" "

<<T[i].M<<" "

<<T[i].D<<" "<<endl;

cout<<" five score :"<<endl;

cout<<"--------------------------------------------------"<<endl;

for(int j=0;j<5;j++){//枚举类型CN,Math....

switch(j){

case CN:cout<<"| Chinese ";break;

case Math:cout<<"| Math ";break;

case Eng:cout<<"| English ";break;

case Sci:cout<<"| Science ";break;

case PE:cout<<"| PE ";break;

}

if(j==4) cout<<"|"<<endl;

}

cout<<"---------------------------------------------------"<<endl;

for(int j=0;j<5;j++){

printf("| %5d ",T[i].score[j]);

if(j==4) cout<<"|"<<endl;//输出空格

}

cout<<"---------------------------------------------------";

cout<<endl;

}

}

else{

cout<<"error,reenter"<<endl;

Sleep(sec);

goto here;

}

//cout<<"input Enter button to return.please";

system("pause");//系统库函数中的函数

}

1. ”modify\_info.h”源代码文件

void modi\_page();

void modify\_info(vector<student> &per)//修改信息

{

if(per.empty()){//若per容器中元素个数为0;

cout<<"Not Valid Data"<<endl;

Print(6,0.8);

return;

}

getchar();

string Name;//动态字符串数组

cout<<"input the name that needs to be modified :";

getline(cin,Name,'\n');//输入name,空格结束

int i=0;

for(;i<per.size()&&Name!=per[i].vi\_na();i++);//查找要修改的学生

if(i==per.size()){//如果i等于元素个数,则没找到

cout<<"Not Found"<<endl;

Sleep(1.5\*sec);

cout<<"Returning to the main page"<<endl;

Print(5);

return;

}

Print(5,1.5);

modi\_page();

int N;

while(cin>>N){//类似while(~scanf.....)

switch(N){

case 1:

char sex;

cout<<"input sex :";

cin>>sex;

per[i].sex=sex;

break;

case 2:

int year;

cout<<"input year :";

cin>>year;

per[i].YR=year;

break;

case 3:

int month;

cout<<"input month :";

cin>>month;

per[i].M=month;

break;

case 4:

int Day;

cout<<"input Day :";

cin>>Day;

per[i].D=Day;

break;

case 5:

cout<<"input integer(1.CN 2.Math 3.Eng 4.Sci 5.PE) :";

int x;

cin>>x;

int number;

cout<<"Enter number :";

cin>>number;

switch(x-1){

case CN:per[i].score[CN]=number;break;

case Math:per[i].score[Math]=number;break;

case Eng:per[i].score[Eng]=number;break;

case Sci:per[i].score[Sci]=number;break;

case PE:per[i].score[PE]=number;break;

}

break;

case 6:Print(5);return;

default :cout<<"Input is incorrect,input again"<<endl;

}

Print(5);

cout<<"modify sucessfully"<<endl;

modi\_page();

}

}

void modi\_page()

{

cout<<"====================="<<endl

<<" 1--sex 2--year "<<endl

<<" 3-month 4--Day"<<endl

<<" 5-score 6-exit"<<endl

<<"====================="<<endl;

cout<<"input number :";

}

1. ”search\_info.h”源代码文件

void search\_page();

void search\_info(vector<student> &per)//搜索信息

{

if(per.empty()){//若per容器为空

cout<<"Not Valid Data"<<endl;

Print(6,0.8);

return;

}

getchar();

out:

search\_page();

int N;int i=0;

cin>>N;//printf

getchar();

if(N==1){

string Snum;//动态字符数组,查找学号

cout<<"input the snum that needs to search :";

getline(cin,Snum,'\n');//输入Snum，回车结束输入

for(;i<per.size()&&Snum!=per[i].vi\_snum();i++);//per.size()返回元素个数,vi\_snum()是访问snum的接口函数

}

else if(N==2){

string Name;//动态字符数组,查找学号

cout<<"input the name that needs to search :";

getline(cin,Name,'\n');//输入Name，回车结束输入

for(;i<per.size()&&Name!=per[i].vi\_na();i++);

}

else if(N==3){

cout<<"input Serial number(eg.1st->1,15th->15) :";

int number;

cin>>number;//scanf

i=number-1;

}

else{

cout<<"Input Error"<<endl;//printf,<<endl类似于输出回车

goto out;

}

if(i==per.size()){//i==per的元素个数

cout<<"Not Found"<<endl;

return ;

}

else{

cout<<" Name :"<<per[i].vi\_na()<<endl

<<" Sex :"<<per[i].sex<<endl

<<" Snum :"<<per[i].vi\_snum()<<endl;

cout<<" birth :"<<per[i].YR<<" "<<per[i].M<<" "

<<per[i].D<<" "<<endl;

cout<<" five score :"<<endl;

cout<<"--------------------------------------------------"<<endl;

for(int j=0;j<5;j++){

switch(j){

case CN:cout<<"| Chinese ";break;

case Math:cout<<"| Math ";break;

case Eng:cout<<"| English ";break;

case Sci:cout<<"| Science ";break;

case PE:cout<<"| PE ";break;

}

if(j==4) cout<<"|"<<endl;

}

cout<<"---------------------------------------------------"<<endl;

for(int j=0;j<5;j++){

printf("| %5d ",per[i].score[j]);

if(j==4) cout<<"|"<<endl;

}

cout<<"---------------------------------------------------";

cout<<endl;

}

}

void search\_page()//搜索界面

{

cout<<"\*\*\*\*\*\*\*Search for stus' info\*\*\*\*\*\*\*\*"<<endl

<<"------------------------------------"<<endl

<<" 1---search by the school number"<<endl

<<" 2---search by Name"<<endl

<<" 3---search by Serial number"<<endl//序号搜索

<<"------------------------------------"<<endl;

cout<<"input number :";

}

1. ”delete\_info.h”源代码文件

void delete\_info(vector<student> &per)//删除学生信息

{

if(per.empty()){//元素个数为空

cout<<"Not Valid Data"<<endl;

Print(6,0.9);

return;

}

getchar();

string Name;

string Snum;

cout<<"input the name that needs to delete :";

getline(cin,Name,'\n');

//getchar();//接受键盘输入的回车

cout<<"input the snum that needs to delete :";

getline(cin,Snum,'\n');

vector<student>::iterator it;//迭代器,类似于定义了vector类的一个指针, 迭代器移动用的是++,--

int i=0;

for(;i<per.size()&&Name!=per[i].vi\_na()&&Snum!=per[i].vi\_snum();i++);//指针所指学生和学号与所要删除学生信息不符时执行循环

it=per.begin()+i;

if(it==per.end()){//若指针到容器末尾的后一位,则没找到

cout<<"there is no student that needs to delete"<<endl;

Print(4,1,"returning");

return;

}

else{

it=per.erase(it);//调用vector模板中的erase函数删除it所指向的学生的信息

cout<<"delete sucessfully"<<endl;

Sleep(1.2\*sec);//window.h中的停顿函数

Print(6,1,"returning");

}

}

1. ”Na\_Sort\_Print.h”源代码文件

/\*由于名字可能有重复,所测试数据并不大,所以用了stable排序算法,归并排序 \*/

void Na\_merge(vector<student> &A,int left,int mid,int right,vector<student> &B,vector<student> &C);

void Na\_mergeSort(vector<student> &A,int left,int right,vector<student> &B,vector<student> &C);

void Na\_Sort\_Print(vector<student> &T)

{

if(T.empty()){

cout<<"Not data"<<endl;

Print(5,1,"returning");

return;

}

vector<student> L(1000),R(1000);//定义两个容器,并且每个申请1000\*student结点的空间\*/

Na\_mergeSort(T,0,T.size(),L,R);

Print\_all(T);//调用自定义函数显示全部

}

void Na\_mergeSort(vector<student> &A,int left,int right,vector<student> &B,vector<student> &C)

{

if(left+1<right){

int mid=(left+right)/2;

Na\_mergeSort(A,left,mid,B,C);

Na\_mergeSort(A,mid,right,B,C);

Na\_merge(A,left,mid,right,B,C);

}

}

void Na\_merge(vector<student> &A,int left,int mid,int right,vector<student> &B,vector<student> &C)

{

int n1=mid-left;

int n2=right-mid;

for(int i=0;i<n1;i++) B[i]=A[left+i];

for(int i=0;i<n2;i++) C[i]=A[mid+i];

int i=0,j=0;

string x="zzzzzz";

student c1;//定义一个类型

c1.SetBa(x,x);//调用类中初始化函数去初始化c1中的name,snum

B[n1]=C[n2]=c1;//在链表L,R末尾设置一个哨兵,使得L,R到末尾停止

for(int k=left;k<right;k++){

if(B[i]<=C[j]){//运算符<=被重载,相当于strcmp(B[i],j[i]),以ASCII码表中字符的位置为阿准

A[k]=B[i++];

}else{

A[k]=C[j++];

}

}

}

1. ”Sum\_Sort\_Print.h”源代码文件

/\*由于总分可能有重复,所测试数据并不大,所以用了stable排序算法,归并排序 \*/

void Sum\_merge(vector<student> &A,int left,int mid,int right);

void Sum\_mergeSort(vector<student> &A,int left,int right);

void Sum\_Sort\_Print(vector<student> &T)

{

if(T.empty()){//若T容器元素为0,返回

cout<<"Not data"<<endl;

Print(5,1,"returning");

return;

}

for(int i=0;i<T.size();i++){

T[i].Soe\_Sum();//调用类的函数算出总分

}

vector<student> l(100),r(100);

Sum\_mergeSort(T,0,T.size());

Print\_all(T);

}

void Sum\_mergeSort(vector<student> &A,int left,int right)

{

if(left+1<right){

int mid=(left+right)/2;

Sum\_mergeSort(A,left,mid);

Sum\_mergeSort(A,mid,right);

Sum\_merge(A,left,mid,right);

}

}

void Sum\_merge(vector<student> &A,int left,int mid,int right)

{

vector<student> L(right+10),R(right+10);

int n1=mid-left;

int n2=right-mid;

for(int i=0;i<n1;i++) L[i]=A[left+i];

for(int i=0;i<n2;i++) R[i]=A[mid+i];

int i=0,j=0;

string x="zzzzzz";

student c1;//定义一个类型

c1.SetBa(x,x);//调用类中初始化函数去初始化c1中的name,snum

c1.Soe\_Sum(3000);

L[n1]=R[n2]=c1;//在链表L,R末尾设置一个哨兵,使得L,R到末尾停止

for(int k=left;k<right;k++){

if(L[i].vi\_sum()<=R[j].vi\_sum()){//调用vi\_sum()函数去访问每个学生中的sum

A[k]=L[i++];

}else{

A[k]=R[j++];

}

}

}

1. ”save\_exit.h”源代码文件

void save\_eixt(vector<student> &T,string filename)

{

fstream infile(filename.c\_str(),ios::out);//以写和清空filename方式打开filename

if(!infile){//若没有打开文件,输出错误信息

cerr<<"Open file error"<<endl;

return;

}

for(int i=0;i<T.size();i++){//在这个for中,infile类似fprintf

infile<<T[i].vi\_na()<<endl<<T[i].sex<<" "<<T[i].vi\_snum()<<" ";

infile<<T[i].YR<<" "<<T[i].M<<" "<<T[i].D<<" ";

for(int j=0;j<5;j++){

infile<<T[i].score[j];

if(j!=4) infile<<" ";

}

}

infile.close();//关闭文件

Print(8,0.7);

cout<<"Withdraw from success"<<endl;

Sleep(1.5\*sec);

exit(0);//退出系统

}