

C++程序设计课程设计

实验报告

**2021/2022(2)**



实验题目 学生选课系统

学生姓名 温家伟

学生学号 202103151422

学生班级 2021大数据分析01

任课教师 姜娓娓

提交日期 2022/6/7

**理学院**

1. **大型实验的内容**

### **学生选课系统用于选课信息的管理。主要的要求包括学生及课程信息的添加、显示、删除、修改、查找和排序。要求使用学习过的C/C++程序设计的知识完成学生选课系统的设计与实现。**

1. **运行环境**

学生选课系统在Visual Studio 2019平台下开发，操作系统：Windows 11。

硬件环境：

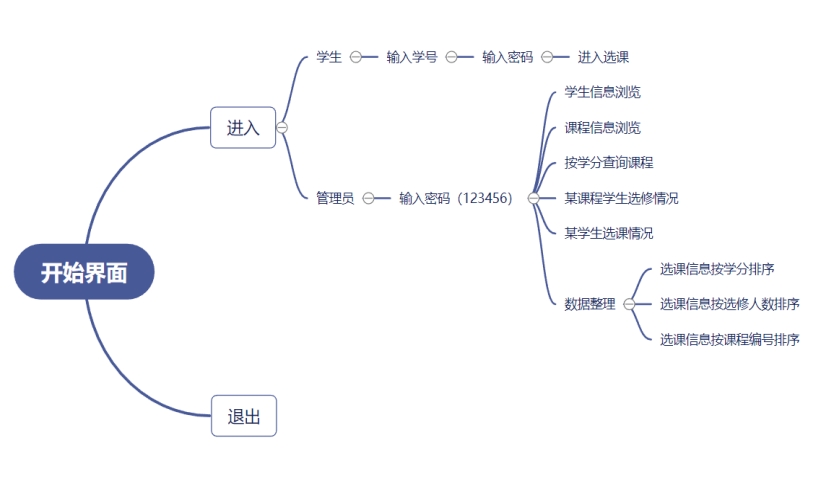
处理器：AMD Ryzen 5 5600H with Radeon Graphics 3.30 GHz

内存：16.0GB

系统类型 64位操作系统

1. **实验课题分析（主要的模块功能、流程图）**

**3.1 学生选课系统的主要功能**

**学生选课系统的主要功能为：**对学生及课程信息的各做操作，可以完成信息的添加、显示、删除、修改和排序的工作。详细的系统功能结构为图1所

**系统各模块的功能具体描述为：**

**1、菜单模块**

打开该系统之后，进入开始界面，根据界面提示，点击进入，选择学生或管理员模式，分别进入对应界面。

1. **管理员**

根据系统的提示，输入管理员密码，进入下一界面：

学生信息浏览：点击即可查看学生总体的选课情况；

课程信息浏览：点击即可查看课程总体情况；

按学分查询课程：根据提示信息，输入学分后，系统查询相应学分的课程并显示；

某课程学生选修情况：根据提示信息，输入课程名字，系统将查询相应课程的学生选课情况；

某学生选课情况：根据提示信息，输入学生名字，系统将查询相应名字的学生的选课情况；

数据整理：根据提示信息，可选择相应的排序规则（按学分排序或是按选修人数排序或是按课程编号排序）。

**3.2 系统分析及设计**

**系统涉及对象有五个基本类：界面类、学生信息类与学生链表类和课程信息类与课程链表类。**

|  |  |
| --- | --- |
| **对象** | **涉及的对象操作** |
| 学生链表 | 创建学生链表、打印学生链表 |
| 课程链表 | 创建课程链表、打印课程链表、排序课程链表 | |
| 初始化类 | 画各个界面，设计退出函数、文字信息函数、鼠标检测函数 | |

**3.3系统的实现**

**（1）类的编写**

系统工程名为：学生选课系统。包含了**界**面类、学生信息类与学生链表类和课程信息类与课程链表类共5个类。

具体类结构声明如下：

**学生信息类**

class Student

{

public:

char num[10];

char name[10];

char password[10];

char \_course[100];

bool operator==(Student& right);

};

**学生链表类**

class StudentNode

{

public:

Student data;

struct StudentNode\* next;

// 单链表打印

void StudentPrint(StudentNode\* plist, int h = 0);

//创建学生链表

StudentNode\* createStudent();

};

**课程信息类**

class CMsg

{

public:

size\_t num;

char name[10];

char kind[10];

size\_t Coursehour;

size\_t credit;

size\_t studentNum;

size\_t semester;

bool operator==(CMsg& right);

};

**课程链表类**

class SListNode

{

public:

CMsg data;

struct SListNode\* next;

// 单链表打印

void SListPrint(SListNode\* plist, int h = 0);

//创建课程链表

SListNode\* createSListCourse();

//按学分排序

void sortByCredit(SListNode\*\* plist);

//按人数排序

void sortByStuNum(SListNode\*\* plist);

};

**初始化类**

class Init

{

public:

//进入与退出界面

//进入---draw2()

void draw1();

//学生与管理员

//学生---draw3()

//管理员---draw4()

void draw2();

//学生的学号与密码输入

void draw3();

//管理员密码输入

void draw4();

//学生选课界面

void draw5();

//学生信息浏览---draw7()

//课程信息浏览---draw8()

//按学分查询课程---draw9()

//某课程学生选修情况---draw10()

//某学生选课情况---draw11()

//数据整理---draw12()

void draw6();

//学生信息浏览

void draw7();

//课程信息浏览

void draw8();

//按学分查询课程

void draw9();

//某课程学生选修情况

void draw10();

//某学生选课情况

void draw11();

// 数据整理界面:

// 选课信息按学分排序---draw13()

// 选课信息按选修人数排序---draw14()

// 选课信息按课程编号排序---draw15()

void draw12();

//选课信息按学分排序

void draw13();

//选课信息按选修人数排序

void draw14();

//选课信息按课程编号排序

void draw15();

//鼠标在某矩形范围内的检测

bool Check(int, int, int, int, ExMessage\*);

//字体设置函数

void settextMSG();

//退出函数

void drawExit();

bool student = 0;

bool administrator = 0;

private:

//背景

void drawBackground();

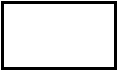
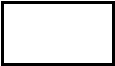
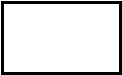
};

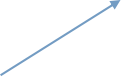
(2)链表的使用

因为学生选课系统在添加、查找、修改、排序的时候都需要处理大量的数据，所以使用链表十分必要。下面是学生链表及课程实现的抽象图。

在运用时，令当前学生链表（课程链表）类的next结点指向新添加的学生（课程）结点，即结点的指针next保存新的学生（课程）结点的地址（如下图所示），以此类推，所有学生（课程）信息就通过链表的形式串联起来了。

Student1 Student2 Student n



 数据 数据

 ......



**课程链表的打印**

void SListNode::SListPrint(SListNode\* plist, int h)

{

SListNode\* cur = plist;

int i = 0;

TCHAR s1[26][5];

TCHAR s2[26][5];

TCHAR s3[26][5];

while (cur)

{

i++;

outtextxy(50, i \* 22 + h, CharToTCHAR(cur->data.name));

outtextxy(150, i \* 22 + h, CharToTCHAR(cur->data.kind));

\_stprintf(s1[i - 1], \_T("%d"), cur->data.Coursehour);

outtextxy(260, i \* 22 + h, s1[i - 1]);

\_stprintf(s2[i - 1], \_T("%d"), cur->data.credit);

outtextxy(320, i \* 22 + h, s2[i - 1]);

\_stprintf(s3[i - 1], \_T("%d"), cur->data.semester);

outtextxy(380, i \* 22 + h, s3[i - 1]);

cur = cur->next;

}

}

**课程链表的创建**

SListNode\* SListNode::createSListCourse()

{

ifstream myfile("Course.txt", ios::in | ios::\_Nocreate);

if (!myfile)

{

cerr << "文件打开失败!" << endl;

abort();

}

CMsg cmsg[26];

//文件内容写入内存

for (int i = 0; i < 26; i++)

{

myfile >> cmsg[i].num;

myfile >> cmsg[i].name;

myfile >> cmsg[i].kind;

myfile >> cmsg[i].Coursehour;

myfile >> cmsg[i].credit;

myfile >> cmsg[i].semester;

myfile >> cmsg[i].studentNum;

}

myfile.close();

SListNode\* head = NULL, \* tail = NULL, \* cur;

head = new SListNode;

if (head == NULL)

{

cout << "No memory!" << endl;

return NULL;

}

else

{

head->data = cmsg[0];

head->next = NULL;

tail = head;

}

for (int i = 1; i < N; i++)

{

cur = new SListNode;

if (cur == NULL)

{

cout << "No memory!" << endl;

return head;

}

else

{

cur->data = cmsg[i];

cur->next = NULL;

tail->next = cur;

tail = cur;

}

}

return head;

}

排序使用了冒泡排序的算法，通过创建临时节点，交换节点的数据域：

**按学分排序**

void SListNode::sortByCredit(SListNode\*\* head)

{

SListNode\* cur = \*head;

while (cur->next)

{

SListNode\* curr = \*head;

while (curr->next)

{

if (curr->data.credit > curr->next->data.credit)

{

CMsg temp = curr->data;

curr->data = curr->next->data;

curr->next->data = temp;

}

curr = curr->next;

}

cur = cur->next;

}

}

**按选修人数排序**

void SListNode::sortByStuNum(SListNode\*\* head)

{

SListNode\* cur = \*head;

while (cur->next)

{

SListNode\* curr = \*head;

while (curr->next)

{

if (curr->data.studentNum > curr->next->data.studentNum)

{

CMsg temp = curr->data;

curr->data = curr->next->data;

curr->next->data = temp;

}

curr = curr->next;

}

cur = cur->next;

}

}

**学生链表的创建**

StudentNode\* StudentNode::createStudent()

{

ifstream myfile("Student.txt", ios::in | ios::\_Nocreate);

if (!myfile)

{

cerr << "文件打开失败!" << endl;

abort();

}

Student student[26] = { 0 };

//读取文件到内存

for (int i = 0; i < 26; i++)

{

myfile >> student[i].num;

myfile >> student[i].name;

myfile >> student[i].password;

myfile >> student[i].\_course;

}

myfile.close();

StudentNode\* head = NULL, \* tail = NULL, \* cur;

head = new StudentNode;

if (head == NULL)

{

cout << "No memory!" << endl;

return NULL;

}

else

{

head->data = student[0];

head->next = NULL;

tail = head;

}

for (int i = 1; i < N; i++)

{

cur = new StudentNode;

if (cur == NULL)

{

cout << "No memory!" << endl;

return head;

}

else

{

cur->data = student[i];

cur->next = NULL;

tail->next = cur;

tail = cur;

}

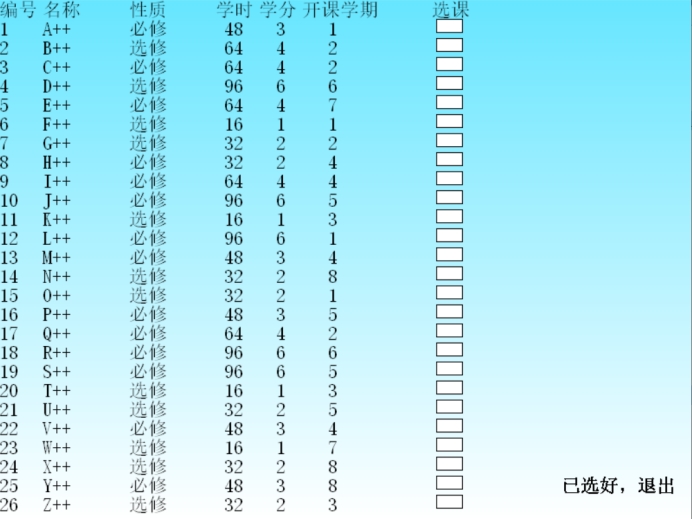
}

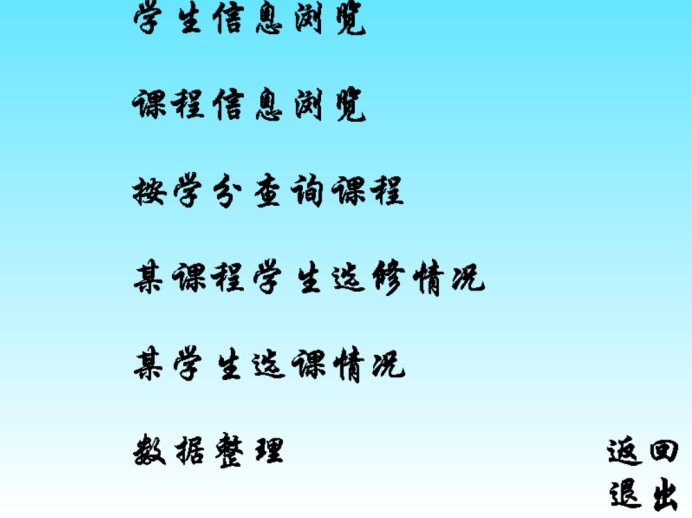
return head;

}

（3）交互界面以及菜单界面的实现

系统运行界面如下图所示：





用户可以根据系统提示，点击相应功能进入对应界面

1. **实验调试、测试、运行记录及分析**

系统在调试测试过程中遇到若干问题，不过经过仔细反复的检查已经消除各种bug。

主要的测试经过如下：

菜单界面如下图所示：

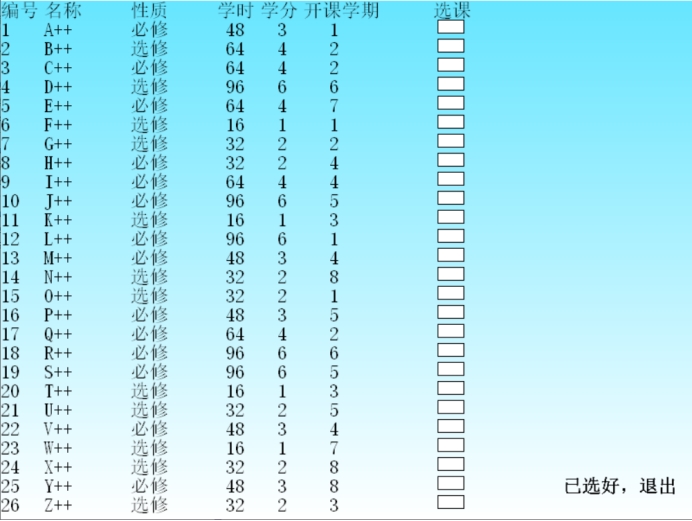
点击进入，跳转到下一页面；点击退出则关闭程序。



点击学生和管理员分别进入对应界面：

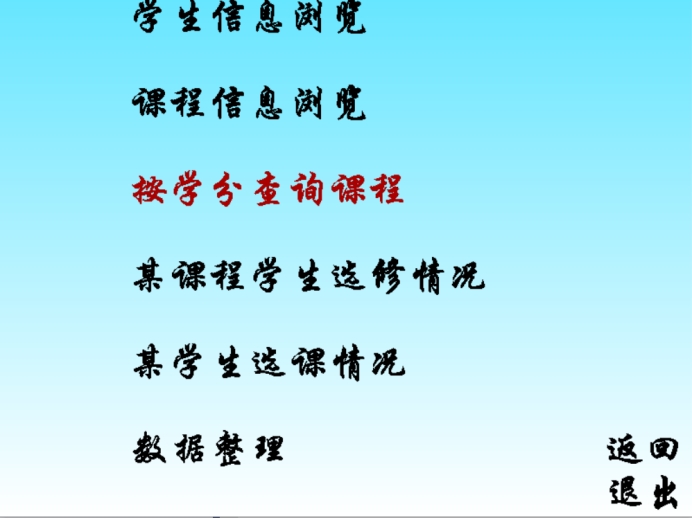


学生选课界面：

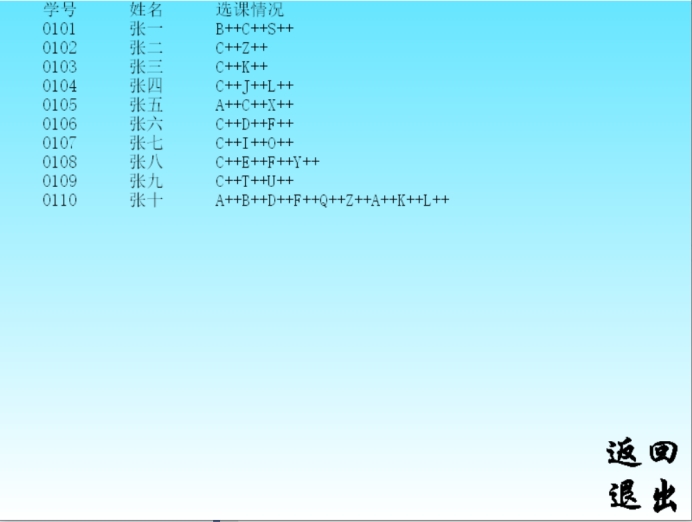


管理员界面：

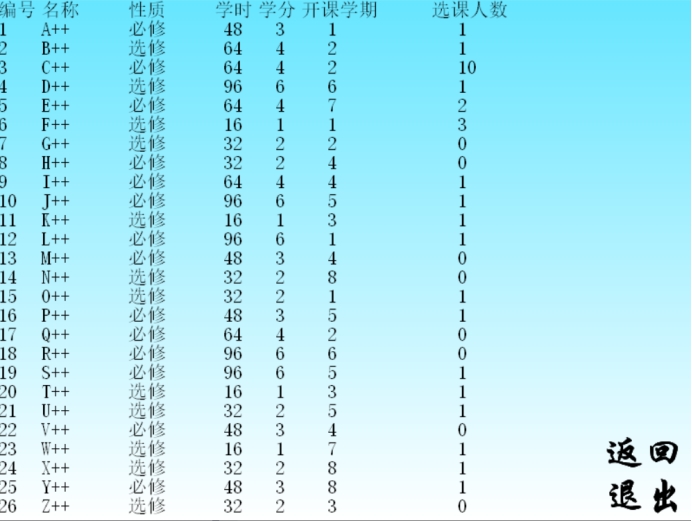
分为一下6个功能：



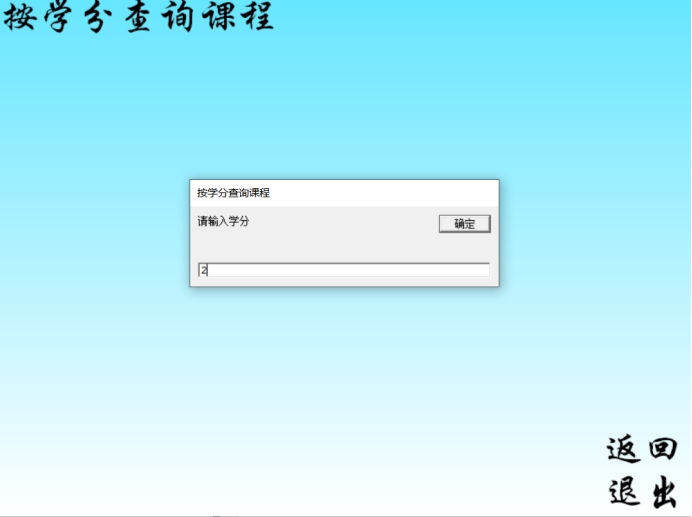
学生信息浏览：

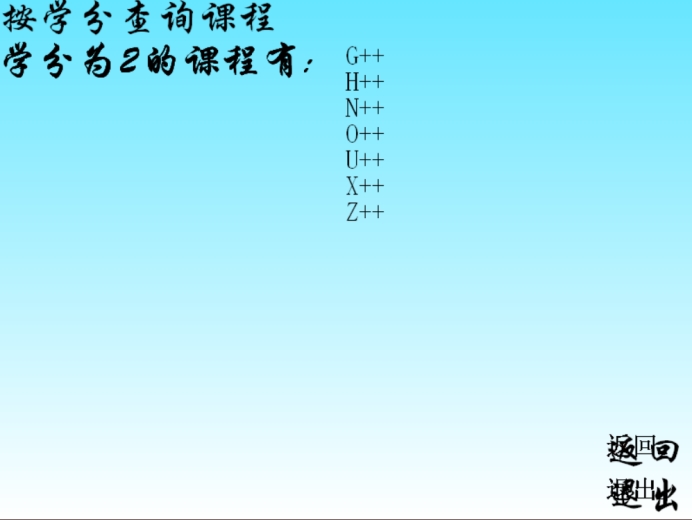


课程信息浏览：



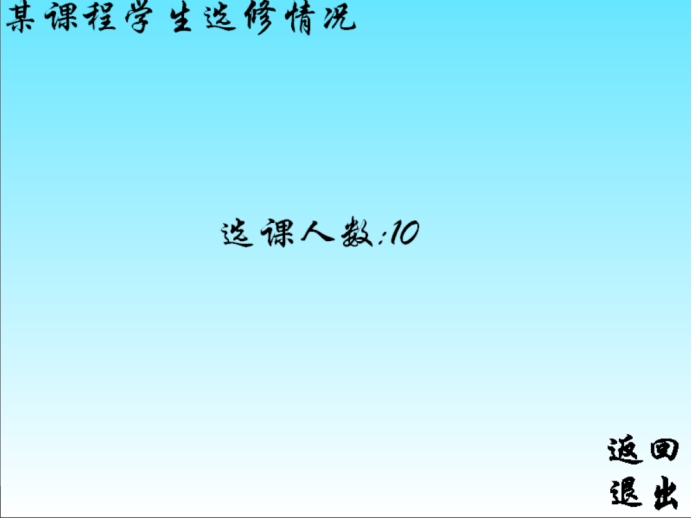
按学分查询课程：





查询某课程学生选修情况：





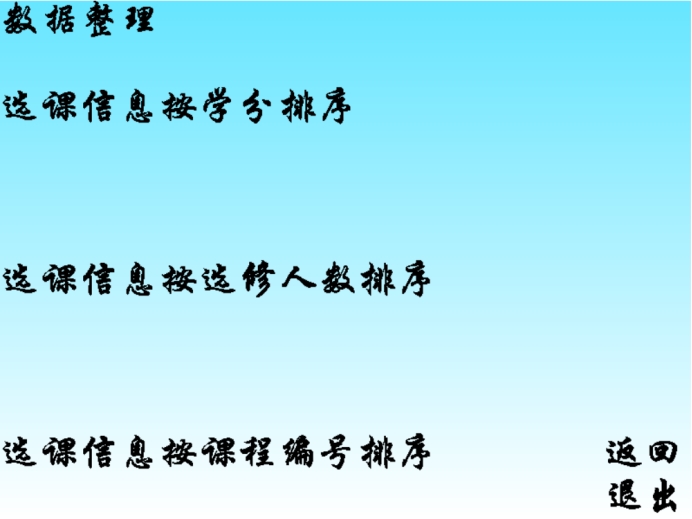
查询某学生选课情况：



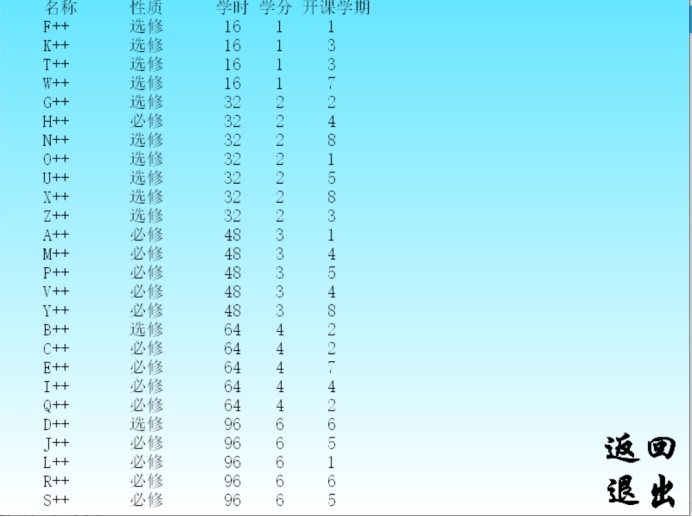


数据整理：

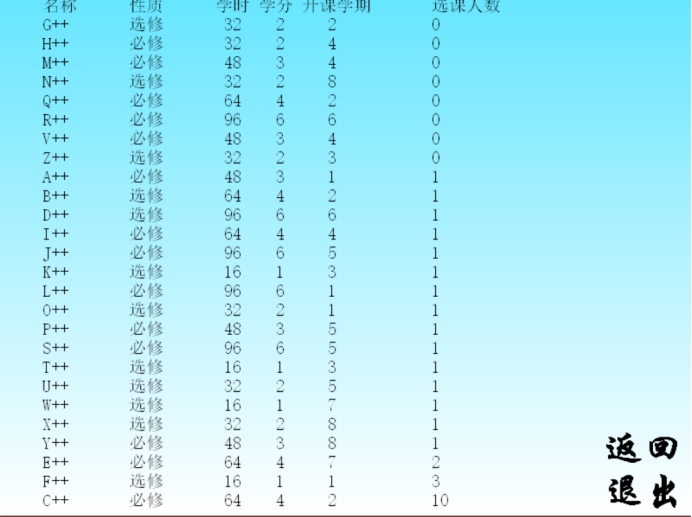
分为3个功能：



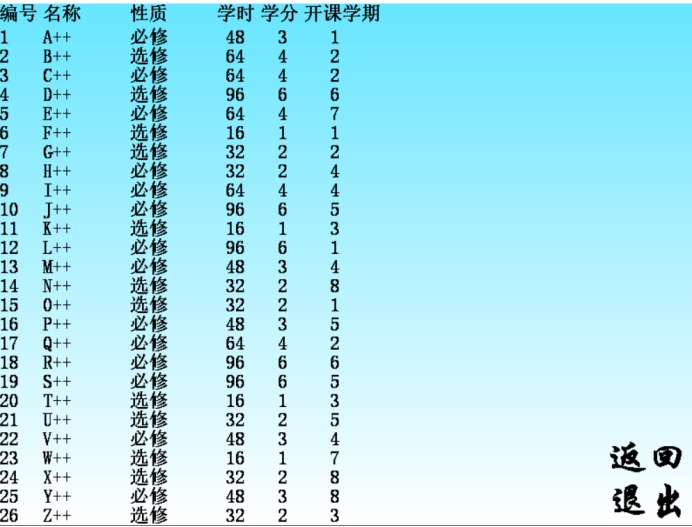
选课信息按学分排序：



选课信息按选修人数排序：



选课信息按课程编号排序：

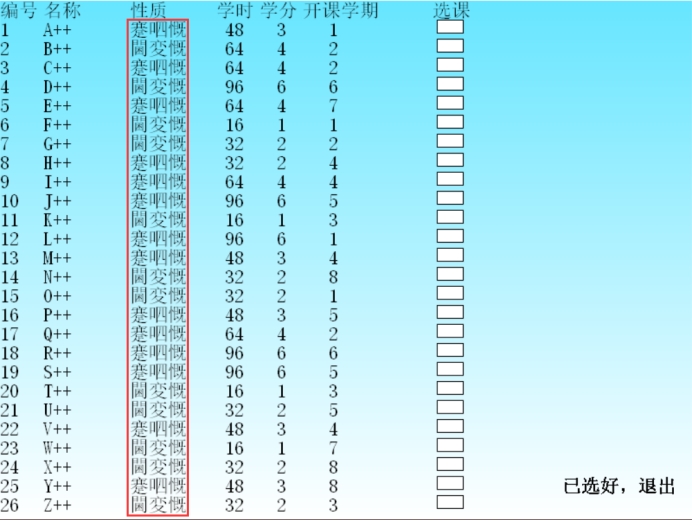


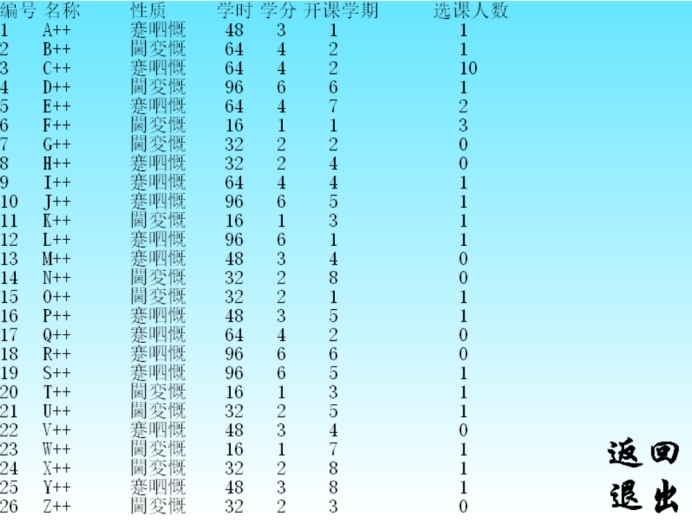
退出界面：



**遇到问题及解决：**

**1.乱码问题**







需要在保存的时候选择ANSI，不然会出现上图的乱码问题。

1. 一个自己常犯的低级错误：遍历链表不写 cur=cur->next;导致指针一直没有移动，这个还调试了好长时间。
2. 返回功能的加入：说实话，我至今对代码中的返回功能不太满意，因为我的不是真正意义上的返回，而是又掉了一遍上一级的界面的函数，但函数传参的过程是需要压栈的，我这样做会导致内存的开销很大，有栈溢出的风险。但这也是我到目前为止能想到的解决返回功能的唯一办法了。

1. **实验总结（优点、不足、收获及体会）**

第一次独自完成这么大的课设，虽然在别人眼中我的题目或许很简单，但对我来说还是很有难度的，其实一开始想用list来存储数据的，又偷懒的想法，但想了想，还是利用这次课设来好好夯实一下自己的基础吧。在写项目过程中，用时最多的就是链表的排序，讲实话，之前完成链表实验的时候，我没有打好基础，当时在写实验排序的时候，在进行交换的时候就只是交换的两个链表的值，而未进行链表的交换，当时也没重视起来，所以在这次排序时，就显得很头痛，这时才认识到自己关于排序方面的知识特别的缺乏，很多排序方法都不知道，而且这些排序方法还都很难理解，还需要自己多多下功夫啊。同时，对于文件的读写也有了更深刻的认识，毕竟当时在这里停滞了好长时间，课设中还是有很多不足的，比如说没有实现指定位置的插入数据，以及在执行各个功能时没有考虑算法的优劣，冗杂重复的代码很多，都没能及时修改，这些都使我认识到，我的C++之路还有很长很长，俗话说的好：吃一堑，长一智。现在多踩些坑，才能为以后打好基础啊。

1. **附录：源代码**

**Init.h**

#pragma once

#include <graphics.h>

#include <conio.h>

class Init

{

public:

//进入与退出界面

//进入---draw2()

void draw1();

//学生与管理员

//学生---draw3()

//管理员---draw4()

void draw2();

//学生的学号与密码输入

void draw3();

//管理员密码输入

void draw4();

//学生选课界面

void draw5();

//学生信息浏览---draw7()

//课程信息浏览---draw8()

//按学分查询课程---draw9()

//某课程学生选修情况---draw10()

//某学生选课情况---draw11()

//数据整理---draw12()

void draw6();

//学生信息浏览

void draw7();

//课程信息浏览

void draw8();

//按学分查询课程

void draw9();

//某课程学生选修情况

void draw10();

//某学生选课情况

void draw11();

// 数据整理界面:

// 选课信息按学分排序---draw13()

// 选课信息按选修人数排序---draw14()

// 选课信息按课程编号排序---draw15()

void draw12();

//选课信息按学分排序

void draw13();

//选课信息按选修人数排序

void draw14();

//选课信息按课程编号排序

void draw15();

//鼠标在某矩形范围内的检测

bool Check(int, int, int, int, ExMessage\*);

//字体设置函数

void settextMSG();

//退出函数

void drawExit();

bool student = 0;

bool administrator = 0;

private:

//背景

void drawBackground();

};

**Init.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include "Init.h"

#include"Cour.h"

#include"Stu.h"

#include<string>

int K = 0;

//多字转宽字 char to wchar\_t

wchar\_t\* char2wchar(const char\* cchar)

{

wchar\_t\* m\_wchar;//定义宽字指针

//取多字长度

int len = MultiByteToWideChar(

CP\_ACP, //代码面

0, //标志

cchar, //多字字符串

strlen(cchar),//多字字符串长度

NULL,//宽字字符串

0);//宽字字符串长度

m\_wchar = new wchar\_t[len + 1];//为宽字指针分配内存

MultiByteToWideChar(CP\_ACP, 0, cchar, strlen(cchar), m\_wchar, len);//复制多字到宽字

m\_wchar[len] = '\0';//字符串结尾

return m\_wchar;//返回指针

}

//宽字转多字 wchar\_t to char

char\* wchar2char(const wchar\_t\* wchar)

{

char\* m\_char;//定义多字指针

int len = WideCharToMultiByte(CP\_ACP, 0, wchar, wcslen(wchar), NULL, 0, NULL, NULL);//取宽字长度

m\_char = new char[len + 1];//为多字指针分配内存

WideCharToMultiByte(CP\_ACP, 0, wchar, wcslen(wchar), m\_char, len, NULL, NULL);//复制宽字到多字

m\_char[len] = '\0';//字符串结尾

return m\_char;//返回多字

}

//初始化背景

//画渐变的天空

void Init::drawBackground()

{

initgraph(800, 600); // 创建绘图窗口，大小为 800\*600

float H = 190; // 色相

float S = 1; // 饱和度

float L = 0.7f; // 亮度

for (int y = 0; y < 600; y++)

{

L += 0.0005f;

setlinecolor(HSLtoRGB(H, S, L));

line(0, y, 800, y);

}

}

//鼠标在某矩形范围内的检测

bool Init::Check(int x1, int x2, int y1, int y2, ExMessage\* m)

{

return (m->x <= x2 && m->x >= x1 && m->y <= y2 && m->y >= y1);

}

//进入与退出界面

//进入---draw2()

void Init::draw1()

{

drawBackground();

settextMSG();

outtextxy(250, 200, \_T("学生选课系统"));

outtextxy(250, 350, \_T("进入"));

outtextxy(434, 350, \_T("退出"));

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

if (Check(250, 348, 350, 398, &m))

{

settextcolor(RED);

outtextxy(250, 350, \_T("进入"));

settextcolor(BLACK);

outtextxy(434, 350, \_T("退出"));

if (m.lbutton)

{

draw2();

break;

}

}

else if (Check(434, 530, 350, 398, &m))

{

settextcolor(BLACK);

outtextxy(250, 350, \_T("进入"));

settextcolor(RED);

outtextxy(434, 350, \_T("退出"));

if (m.lbutton)

{

drawExit();

}

}

else

{

settextcolor(BLACK);

outtextxy(250, 350, \_T("进入"));

settextcolor(BLACK);

outtextxy(434, 350, \_T("退出"));

}

}

}

//学生与管理员

//学生---draw3()

//管理员---draw4()

void Init::draw2()

{

drawBackground();

settextMSG();

outtextxy(250, 270, \_T("学生"));

outtextxy(434, 270, \_T("管理员"));

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

if (Check(250, 348, 270, 318, &m))

{

settextcolor(RED);

outtextxy(250, 270, \_T("学生"));

settextcolor(BLACK);

outtextxy(434, 270, \_T("管理员"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

student = 1;

draw3();

break;

}

}

else if (Check(700, 800, 500, 550, &m))

{

settextcolor(RED);

outtextxy(700, 500, \_T("返回"));

settextcolor(BLACK);

outtextxy(700, 550, \_T("退出"));

outtextxy(250, 270, \_T("学生"));

outtextxy(434, 270, \_T("管理员"));

if (m.lbutton)

{

draw1();

}

}

else if (Check(700, 800, 550, 600, &m))

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(250, 270, \_T("学生"));

outtextxy(434, 270, \_T("管理员"));

settextcolor(RED);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

drawExit();

}

}

else if (Check(434, 578, 270, 318, &m))

{

settextcolor(BLACK);

outtextxy(250, 270, \_T("学生"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

settextcolor(RED);

outtextxy(434, 270, \_T("管理员"));

if (m.lbutton)

{

administrator = 1;

draw4();

break;

}

}

else

{

settextcolor(BLACK);

outtextxy(250, 270, \_T("学生"));

settextcolor(BLACK);

outtextxy(434, 270, \_T("管理员"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

student = 0;

administrator = 0;

}

}

}

//学生的学号与密码输入

void Init::draw3()

{

again:

drawBackground();

// 定义字符串缓冲区，并接收用户输入

wchar\_t s1[10];

InputBox(s1, 10, L"请输入学号", L"学生");

wchar\_t s2[10];

InputBox(s2, 10, L"请输入密码", L"学生");

StudentNode \_list;

StudentNode\* head = \_list.createStudent();

StudentNode\* cur = head;

int k = 0;

while (cur)

{

if (strcmp(wchar2char(s1), cur->data.num) == 0 && strcmp(wchar2char(s2), cur->data.password) == 0)

{

K = k;

break;

}

cur = cur->next;

k++;

if (cur == NULL)

{

goto again;

}

}

}

//管理员密码输入

void Init::draw4()

{

drawBackground();

// 定义字符串缓冲区，并接收用户输入

wchar\_t s[10];

int i = 0;

while (strcmp(wchar2char(s), "123456") != 0)

{

i++;

InputBox(s, 10, L"请输入密码", L"管理员");

if (i == 5)

exit(-1);

}

}

//学生选课界面

void Init::draw5()

{

drawBackground();

LOGFONT f;

gettextstyle(&f); // 获取当前字体设置

f.lfHeight = 22; // 设置字体高度为 22

\_tcscpy(f.lfFaceName, \_T("宋体")); // 设置字体为“华文行楷”

f.lfQuality = ANTIALIASED\_QUALITY; // 设置输出效果为抗锯齿

settextstyle(&f); // 设置字体样式

settextcolor(BLACK);

setbkmode(TRANSPARENT);

outtextxy(0, 0, \_T("编号"));

outtextxy(50, 0, \_T("名称"));

outtextxy(150, 0, \_T("性质"));

outtextxy(250, 0, \_T("学时"));

outtextxy(300, 0, \_T("学分"));

outtextxy(350, 0, \_T("开课学期"));

outtextxy(500, 0, \_T("选课"));

SListNode list;

SListNode\* phead = list.createSListCourse();

for (int i = 1; i <= 26; i++)

{

TCHAR s[5];

\_stprintf(s, \_T("%d"), i);

outtextxy(0, i \* 22, s);

}

list.SListPrint(phead);

for (int i = 1; i <= 26; i++)

{

setlinecolor(BLACK);

fillrectangle(505, i \* 22, 535, i \* 22 + 15);

}

outtextxy(650, 550, \_T("已选好，退出"));

if (student)

{

int arr[26] = { 0 };//放循环外面

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

if (Check(650, 780, 550, 570, &m))

{

settextcolor(RED);

outtextxy(650, 550, \_T("已选好，退出"));

if (m.lbutton)

{

drawExit();

}

}

else

{

settextcolor(BLACK);

outtextxy(650, 550, \_T("已选好，退出"));

}

for (int i = 1; i <= 26; i++)

{

if (Check(505, 535, i \* 22, i \* 22 + 20, &m))

{

setfillcolor(RED);

fillrectangle(505, i \* 22, 535, i \* 22 + 15);

if (m.lbutton)

{

ofstream myfile("Student.txt", ios::out|ios::\_Nocreate|ios::app);

if (!myfile)

{

cerr << "文件打开失败!" << endl;

}

char str[26][4] = { "A++","B++","C++","D++","E++","F++","G++","H++","I++","J++","K++","L++",

"M++","N++","O++","P++","Q++","R++","S++","T++","U++","V++","W++","X++" ,"Y++" ,"Z++" };

arr[i - 1] ^= 1;//神奇的位运算

StudentNode \_list;

StudentNode\* phead = \_list.createStudent();

StudentNode\* cur = phead;

while (cur)

{

if (cur->next==NULL)

{

if (arr[i - 1])

myfile << str[i - 1];

}

cur = cur->next;

}

myfile.close();

}

}

else

{

for (int j = 1; j <= 26; j++)

{

if (arr[j - 1] == 1)

{

setfillcolor(RED);

fillrectangle(505, j \* 22, 535, j \* 22 + 15);

}

else

{

setfillcolor(WHITE);

fillrectangle(505, j \* 22, 535, j \* 22 + 15);

}

}

}

}

}

}

}

//学生信息浏览---draw7()

//课程信息浏览---draw8()

//按学分查询课程---draw9()

//某课程学生选修情况---draw10()

//某学生选课情况---draw11()

//数据整理---draw12()

void Init::draw6()

{

drawBackground();

settextMSG();

outtextxy(150, 0, \_T("学生信息浏览"));

outtextxy(150, 100, \_T("课程信息浏览"));

outtextxy(150, 200, \_T("按学分查询课程"));

outtextxy(150, 300, \_T("某课程学生选修情况"));

outtextxy(150, 400, \_T("某学生选课情况"));

outtextxy(150, 500, \_T("数据整理"));

if (administrator)

{

int arr[6] = { 0,0,0,0,0,0 };

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

if (Check(700, 800, 500, 550, &m))

{

settextcolor(RED);

outtextxy(700, 500, \_T("返回"));

settextcolor(BLACK);

outtextxy(700, 550, \_T("退出"));

outtextxy(150, 0, \_T("学生信息浏览"));

outtextxy(150, 100, \_T("课程信息浏览"));

outtextxy(150, 200, \_T("按学分查询课程"));

outtextxy(150, 300, \_T("某课程学生选修情况"));

outtextxy(150, 400, \_T("某学生选课情况"));

outtextxy(150, 500, \_T("数据整理"));

if (m.lbutton)

{

draw2();

}

}

else if (Check(700, 800, 550, 600, &m))

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(150, 0, \_T("学生信息浏览"));

outtextxy(150, 100, \_T("课程信息浏览"));

outtextxy(150, 200, \_T("按学分查询课程"));

outtextxy(150, 300, \_T("某课程学生选修情况"));

outtextxy(150, 400, \_T("某学生选课情况"));

outtextxy(150, 500, \_T("数据整理"));

settextcolor(RED);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

drawExit();

}

}

else if (Check(150, 430, 0, 50, &m))

{

settextcolor(RED);

outtextxy(150, 0, \_T("学生信息浏览"));

settextcolor(BLACK);

outtextxy(150, 100, \_T("课程信息浏览"));

outtextxy(150, 200, \_T("按学分查询课程"));

outtextxy(150, 300, \_T("某课程学生选修情况"));

outtextxy(150, 400, \_T("某学生选课情况"));

outtextxy(150, 500, \_T("数据整理"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

arr[0] = 1;

draw7();

break;

}

}

else if (Check(150, 430, 100, 150, &m))

{

settextcolor(RED);

outtextxy(150, 100, \_T("课程信息浏览"));

settextcolor(BLACK);

outtextxy(150, 0, \_T("学生信息浏览"));

outtextxy(150, 200, \_T("按学分查询课程"));

outtextxy(150, 300, \_T("某课程学生选修情况"));

outtextxy(150, 400, \_T("某学生选课情况"));

outtextxy(150, 500, \_T("数据整理"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

arr[1] = 1;

draw8();

break;

}

}

else if (Check(150, 480, 200, 250, &m))

{

settextcolor(RED);

outtextxy(150, 200, \_T("按学分查询课程"));

settextcolor(BLACK);

outtextxy(150, 0, \_T("学生信息浏览"));

outtextxy(150, 100, \_T("课程信息浏览"));

outtextxy(150, 300, \_T("某课程学生选修情况"));

outtextxy(150, 400, \_T("某学生选课情况"));

outtextxy(150, 500, \_T("数据整理"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

arr[2] = 1;

draw9();

break;

}

}

else if (Check(150, 570, 300, 350, &m))

{

settextcolor(RED);

outtextxy(150, 300, \_T("某课程学生选修情况"));

settextcolor(BLACK);

outtextxy(150, 0, \_T("学生信息浏览"));

outtextxy(150, 100, \_T("课程信息浏览"));

outtextxy(150, 200, \_T("按学分查询课程"));

outtextxy(150, 400, \_T("某学生选课情况"));

outtextxy(150, 500, \_T("数据整理"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

arr[3] = 1;

draw10();

break;

}

}

else if (Check(150, 480, 400, 450, &m))

{

settextcolor(RED);

outtextxy(150, 400, \_T("某学生选课情况"));

settextcolor(BLACK);

outtextxy(150, 0, \_T("学生信息浏览"));

outtextxy(150, 100, \_T("课程信息浏览"));

outtextxy(150, 200, \_T("按学分查询课程"));

outtextxy(150, 300, \_T("某课程学生选修情况"));

outtextxy(150, 500, \_T("数据整理"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

arr[4] = 1;

draw11();

break;

}

}

else if (Check(150, 330, 500, 550, &m))

{

settextcolor(RED);

outtextxy(150, 500, \_T("数据整理"));

settextcolor(BLACK);

outtextxy(150, 0, \_T("学生信息浏览"));

outtextxy(150, 100, \_T("课程信息浏览"));

outtextxy(150, 200, \_T("按学分查询课程"));

outtextxy(150, 300, \_T("某课程学生选修情况"));

outtextxy(150, 400, \_T("某学生选课情况"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

arr[5] = 1;

draw12();

break;

}

}

else

{

settextcolor(BLACK);

outtextxy(150, 0, \_T("学生信息浏览"));

outtextxy(150, 100, \_T("课程信息浏览"));

outtextxy(150, 200, \_T("按学分查询课程"));

outtextxy(150, 300, \_T("某课程学生选修情况"));

outtextxy(150, 400, \_T("某学生选课情况"));

outtextxy(150, 500, \_T("数据整理"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

}

if (arr[0])

{

draw7();

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

if (Check(700, 800, 500, 550, &m))

{

settextcolor(RED);

outtextxy(700, 500, \_T("返回"));

settextcolor(BLACK);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

draw6();

}

}

else if (Check(700, 800, 550, 600, &m))

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

settextcolor(RED);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

drawExit();

}

}

else

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

}

}

if (arr[1])

{

draw8();

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

if (Check(700, 800, 500, 550, &m))

{

settextcolor(RED);

outtextxy(700, 500, \_T("返回"));

settextcolor(BLACK);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

draw6();

}

}

else if (Check(700, 800, 550, 600, &m))

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

settextcolor(RED);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

drawExit();

}

}

else

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

}

}

if (arr[2])

{

draw9();

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

if (Check(700, 800, 500, 550, &m))

{

settextcolor(RED);

outtextxy(700, 500, \_T("返回"));

settextcolor(BLACK);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

draw6();

}

}

else if (Check(700, 800, 550, 600, &m))

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

settextcolor(RED);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

drawExit();

}

}

else

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

}

}

if (arr[3])

{

draw10();

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

if (Check(700, 800, 500, 550, &m))

{

settextcolor(RED);

outtextxy(700, 500, \_T("返回"));

settextcolor(BLACK);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

draw6();

}

}

else if (Check(700, 800, 550, 600, &m))

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

settextcolor(RED);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

drawExit();

}

}

else

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

}

}

if (arr[4])

{

draw11();

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

if (Check(700, 800, 500, 550, &m))

{

settextcolor(RED);

outtextxy(700, 500, \_T("返回"));

settextcolor(BLACK);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

draw6();

}

}

else if (Check(700, 800, 550, 600, &m))

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

settextcolor(RED);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

drawExit();

}

}

else

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

}

}

if (arr[5])

{

draw12();

}

}

}

//学生信息浏览

void Init::draw7()

{

drawBackground();

LOGFONT f;

gettextstyle(&f); // 获取当前字体设置

f.lfHeight = 20; // 设置字体高度为 48

\_tcscpy(f.lfFaceName, \_T("宋体")); // 设置字体为“华文行楷”

f.lfQuality = ANTIALIASED\_QUALITY; // 设置输出效果为抗锯齿

settextstyle(&f); // 设置字体样式

setbkmode(TRANSPARENT);

settextcolor(BLACK);

StudentNode \_list;

StudentNode\* head = \_list.createStudent();

\_list.StudentPrint(head);

outtextxy(50, 0, \_T("学号"));

outtextxy(150, 0, \_T("姓名"));

outtextxy(250, 0, \_T("选课情况"));

settextMSG();

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

//课程信息浏览

void Init::draw8()

{

drawBackground();

settextMSG();

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

settextMSG();

settextcolor(BLACK);

LOGFONT f;

gettextstyle(&f); // 获取当前字体设置

f.lfHeight = 22; // 设置字体高度为 22

\_tcscpy(f.lfFaceName, \_T("宋体")); // 设置字体为“华文行楷”

f.lfQuality = ANTIALIASED\_QUALITY; // 设置输出效果为抗锯齿

settextstyle(&f); // 设置字体样式

setbkmode(TRANSPARENT);

outtextxy(0, 0, \_T("编号"));

outtextxy(50, 0, \_T("名称"));

outtextxy(150, 0, \_T("性质"));

outtextxy(250, 0, \_T("学时"));

outtextxy(300, 0, \_T("学分"));

outtextxy(350, 0, \_T("开课学期"));

outtextxy(500, 0, \_T("选课人数"));

SListNode list;

SListNode\* phead = list.createSListCourse();

SListNode\* cur = phead;

TCHAR s2[26][5];

int ii = 0;

while (cur)

{

ii++;

\_stprintf(s2[ii - 1], \_T("%d"), cur->data.studentNum);

outtextxy(530, ii \* 22, s2[ii - 1]);

cur = cur->next;

}

for (int i = 1; i <= 26; i++)

{

TCHAR s1[5];

\_stprintf(s1, \_T("%d"), i);

outtextxy(0, i \* 22, s1);

}

list.SListPrint(phead);

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

if (Check(700, 800, 550, 600, &m))

{

settextMSG();

settextcolor(RED);

outtextxy(700, 550, \_T("退出"));

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

if (m.lbutton)

{

drawExit();

}

}

else if (Check(700, 800, 500, 550, &m))

{

settextMSG();

settextcolor(RED);

outtextxy(700, 500, \_T("返回"));

settextcolor(BLACK);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

draw6();

}

}

else

{

settextMSG();

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

}

}

//按学分查询课程

void Init::draw9()

{

drawBackground();

settextMSG();

outtextxy(0, 0, \_T("按学分查询课程"));

wchar\_t s[10];

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

InputBox(s, 10, L"请输入学分", L"按学分查询课程");

SListNode list;

SListNode\* phead = list.createSListCourse();

SListNode\* cur = phead;

int k = 0;

for (int i = 0; i < 26; i++)

{

if (stoi(s) == cur->data.credit)

{

settextMSG();

outtextxy(0, 50, \_T("学分为"));

outtextxy(165, 50, \_T("的课程有:"));

TCHAR t[5];

\_stprintf(t, \_T("%d"), cur->data.credit);

outtextxy(140, 50, t);

LOGFONT f;

gettextstyle(&f); // 获取当前字体设置

f.lfHeight = 30; // 设置字体高度为 20

\_tcscpy(f.lfFaceName, \_T("宋体")); // 设置字体为“宋体”

f.lfQuality = ANTIALIASED\_QUALITY; // 设置输出效果为抗锯齿

settextstyle(&f); // 设置字体样式

setbkmode(TRANSPARENT);

settextcolor(BLACK);

outtextxy(400, 50 + 30 \* k, char2wchar(cur->data.name));

k++;

}

cur = cur->next;

}

}

//某课程学生选修情况

void Init::draw10()

{

drawBackground();

settextMSG();

outtextxy(0, 0, \_T("某课程学生选修情况"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

wchar\_t s[10];

InputBox(s, 10, L"请输入课程名", L"某课程学生选修情况");

SListNode list;

SListNode\* phead = list.createSListCourse();

SListNode\* cur = phead;

for (int i = 0; i < 26; i++)

{

if (strcmp(wchar2char(s), cur->data.name) == 0)

{

TCHAR t[5];

outtextxy(250, 250, \_T("选课人数:"));

\_stprintf(t, \_T("%d"), cur->data.studentNum);

outtextxy(450, 250, t);

}

cur = cur->next;

}

}

//某学生选课情况

void Init::draw11()

{

drawBackground();

settextMSG();

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

wchar\_t s[10];

InputBox(s, 10, L"请输入学生姓名", L"某学生选课情况");

StudentNode \_list;

StudentNode\* phead = \_list.createStudent();

StudentNode\* cur = phead;

for (int i = 0; i < 10; i++)

{

if (strcmp(wchar2char(s), cur->data.name) == 0)

{

settextMSG();

outtextxy(0, 0, \_T("选课情况:"));

LOGFONT f;

gettextstyle(&f); // 获取当前字体设置

f.lfHeight = 48; // 设置字体高度为 48

\_tcscpy(f.lfFaceName, \_T("宋体")); // 设置字体为“华文行楷”

f.lfQuality = ANTIALIASED\_QUALITY; // 设置输出效果为抗锯齿

settextstyle(&f); // 设置字体样式

setbkmode(TRANSPARENT);

settextcolor(BLACK);

outtextxy(0, 50, char2wchar(cur->data.\_course));

}

settextMSG();

cur = cur->next;

}

}

// 数据整理界面:

// 选课信息按学分排序---draw13()

// 选课信息按选修人数排序---draw14()

// 选课信息按课程编号排序---draw15()

void Init::draw12()

{

drawBackground();

settextMSG();

outtextxy(0, 0, \_T("数据整理"));

outtextxy(0, 100, \_T("选课信息按学分排序"));

outtextxy(0, 300, \_T("选课信息按选修人数排序"));

outtextxy(0, 500, \_T("选课信息按课程编号排序"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

if (Check(700, 800, 500, 550, &m))

{

settextcolor(RED);

outtextxy(700, 500, \_T("返回"));

settextcolor(BLACK);

outtextxy(700, 550, \_T("退出"));

outtextxy(0, 100, \_T("选课信息按学分排序"));

outtextxy(0, 300, \_T("选课信息按选修人数排序"));

outtextxy(0, 500, \_T("选课信息按课程编号排序"));

if (m.lbutton)

{

draw6();

}

}

else if (Check(700, 800, 550, 600, &m))

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(0, 100, \_T("选课信息按学分排序"));

outtextxy(0, 300, \_T("选课信息按选修人数排序"));

outtextxy(0, 500, \_T("选课信息按课程编号排序"));

settextcolor(RED);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

drawExit();

}

}

else if (Check(0, 410, 100, 150, &m))

{

settextcolor(RED);

outtextxy(0, 100, \_T("选课信息按学分排序"));

if (m.lbutton)

{

draw13();

}

settextcolor(BLACK);

outtextxy(0, 0, \_T("数据整理"));

outtextxy(0, 300, \_T("选课信息按选修人数排序"));

outtextxy(0, 500, \_T("选课信息按课程编号排序"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

else if (Check(0, 510, 300, 350, &m))

{

settextcolor(RED);

outtextxy(0, 300, \_T("选课信息按选修人数排序"));

if (m.lbutton)

{

draw14();

break;

}

settextcolor(BLACK);

outtextxy(0, 0, \_T("数据整理"));

outtextxy(0, 100, \_T("选课信息按学分排序"));

outtextxy(0, 500, \_T("选课信息按课程编号排序"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

else if (Check(0, 510, 500, 550, &m))

{

settextcolor(RED);

outtextxy(0, 500, \_T("选课信息按课程编号排序"));

if (m.lbutton)

{

draw15();

break;

}

settextcolor(BLACK);

outtextxy(0, 0, \_T("数据整理"));

outtextxy(0, 100, \_T("选课信息按学分排序"));

outtextxy(0, 300, \_T("选课信息按选修人数排序"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

else

{

settextcolor(BLACK);

outtextxy(0, 0, \_T("数据整理"));

outtextxy(0, 100, \_T("选课信息按学分排序"));

outtextxy(0, 300, \_T("选课信息按选修人数排序"));

outtextxy(0, 500, \_T("选课信息按课程编号排序"));

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

}

}

//选课信息按学分排序

void Init::draw13()

{

drawBackground();

settextMSG();

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

SListNode list;

SListNode\* phead = list.createSListCourse();

list.sortByCredit(&phead);

LOGFONT f;

gettextstyle(&f); // 获取当前字体设置

f.lfHeight = 20; // 设置字体高度为 48

\_tcscpy(f.lfFaceName, \_T("宋体")); // 设置字体为“华文行楷”

f.lfQuality = ANTIALIASED\_QUALITY; // 设置输出效果为抗锯齿

settextstyle(&f); // 设置字体样式

setbkmode(TRANSPARENT);

settextcolor(BLACK);

list.SListPrint(phead);

outtextxy(50, 0, \_T("名称"));

outtextxy(150, 0, \_T("性质"));

outtextxy(250, 0, \_T("学时"));

outtextxy(300, 0, \_T("学分"));

outtextxy(350, 0, \_T("开课学期"));

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

settextMSG();

if (Check(700, 800, 550, 600, &m))

{

settextcolor(RED);

outtextxy(700, 550, \_T("退出"));

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

if (m.lbutton)

{

drawExit();

}

}

else if (Check(700, 800, 500, 550, &m))

{

settextcolor(RED);

outtextxy(700, 500, \_T("返回"));

settextcolor(BLACK);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

draw12();

}

}

else

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

}

}

//选课信息按选修人数排序

void Init::draw14()

{

drawBackground();

settextMSG();

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

SListNode list;

SListNode\* phead = list.createSListCourse();

list.sortByStuNum(&phead);

LOGFONT f;

gettextstyle(&f); // 获取当前字体设置

f.lfHeight = 20; // 设置字体高度为 48

\_tcscpy(f.lfFaceName, \_T("宋体")); // 设置字体为“华文行楷”

f.lfQuality = ANTIALIASED\_QUALITY; // 设置输出效果为抗锯齿

settextstyle(&f); // 设置字体样式

setbkmode(TRANSPARENT);

settextcolor(BLACK);

list.SListPrint(phead);

outtextxy(50, 0, \_T("名称"));

outtextxy(150, 0, \_T("性质"));

outtextxy(250, 0, \_T("学时"));

outtextxy(300, 0, \_T("学分"));

outtextxy(350, 0, \_T("开课学期"));

outtextxy(500, 0, \_T("选课人数"));

TCHAR s3[26][5];

int i = 0;

SListNode\* cur = phead;

while (cur)

{

i++;

\_stprintf(s3[i - 1], \_T("%d"), cur->data.studentNum);

outtextxy(500, i \* 22, s3[i - 1]);

cur = cur->next;

}

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

settextMSG();

if (Check(700, 800, 550, 600, &m))

{

settextcolor(RED);

outtextxy(700, 550, \_T("退出"));

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

if (m.lbutton)

{

drawExit();

}

}

else if (Check(700, 800, 500, 550, &m))

{

settextcolor(RED);

outtextxy(700, 500, \_T("返回"));

settextcolor(BLACK);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

draw12();

}

}

else

{

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

}

}

//选课信息按课程编号排序

void Init::draw15()

{

drawBackground();

settextMSG();

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

settextMSG();

settextcolor(BLACK);

LOGFONT f;

gettextstyle(&f); // 获取当前字体设置

f.lfHeight = 22; // 设置字体高度为 22

\_tcscpy(f.lfFaceName, \_T("宋体")); // 设置字体为“华文行楷”

f.lfQuality = ANTIALIASED\_QUALITY; // 设置输出效果为抗锯齿

settextstyle(&f); // 设置字体样式

setbkmode(TRANSPARENT);

outtextxy(0, 0, \_T("编号"));

outtextxy(50, 0, \_T("名称"));

outtextxy(150, 0, \_T("性质"));

outtextxy(250, 0, \_T("学时"));

outtextxy(300, 0, \_T("学分"));

outtextxy(350, 0, \_T("开课学期"));

SListNode list;

SListNode\* phead = list.createSListCourse();

for (int i = 1; i <= 26; i++)

{

TCHAR s[5];

\_stprintf(s, \_T("%d"), i);

outtextxy(0, 5 + i \* 22, s);

}

list.SListPrint(phead, 5);

settextMSG();

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

if (Check(700, 800, 550, 600, &m))

{

settextMSG();

settextcolor(RED);

outtextxy(700, 550, \_T("退出"));

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

if (m.lbutton)

{

drawExit();

}

}

else if (Check(700, 800, 500, 550, &m))

{

settextMSG();

settextcolor(RED);

outtextxy(700, 500, \_T("返回"));

settextcolor(BLACK);

outtextxy(700, 550, \_T("退出"));

if (m.lbutton)

{

draw12();

}

}

else

{

settextMSG();

settextcolor(BLACK);

outtextxy(700, 500, \_T("返回"));

outtextxy(700, 550, \_T("退出"));

}

}

}

//字体设置函数

void Init::settextMSG()

{

LOGFONT f;

gettextstyle(&f); // 获取当前字体设置

f.lfHeight = 48; // 设置字体高度为 48

\_tcscpy(f.lfFaceName, \_T("华文行楷")); // 设置字体为“华文行楷”

f.lfQuality = ANTIALIASED\_QUALITY; // 设置输出效果为抗锯齿

settextstyle(&f); // 设置字体样式

setbkmode(TRANSPARENT);

settextcolor(BLACK);

}

//退出函数

void Init::drawExit()

{

cleardevice();

settextMSG();

settextcolor(WHITE);

outtextxy(300, 275, \_T("谢谢使用！"));

Sleep(3000);

exit(0);

}

**Cour.h**

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <assert.h>

#include<iostream>

#include <graphics.h>

#include <conio.h>

#include <fstream>

#define N 26

using namespace std;

class CMsg

{

public:

size\_t num;

char name[10];

char kind[10];

size\_t Coursehour;

size\_t credit;

size\_t studentNum;

size\_t semester;

bool operator==(CMsg& right);

};

class SListNode

{

public:

CMsg data;

struct SListNode\* next;

// 单链表打印

void SListPrint(SListNode\* plist, int h = 0);

//创建课程链表

SListNode\* createSListCourse();

//按学分排序

void sortByCredit(SListNode\*\* plist);

//按人数排序

void sortByStuNum(SListNode\*\* plist);

};

**Cour.cpp**

void SListNode::SListPrint(SListNode\* plist, int h)

{

SListNode\* cur = plist;

int i = 0;

TCHAR s1[26][5];

TCHAR s2[26][5];

TCHAR s3[26][5];

while (cur)

{

i++;

outtextxy(50, i \* 22 + h, CharToTCHAR(cur->data.name));

outtextxy(150, i \* 22 + h, CharToTCHAR(cur->data.kind));

\_stprintf(s1[i - 1], \_T("%d"), cur->data.Coursehour);

outtextxy(260, i \* 22 + h, s1[i - 1]);

\_stprintf(s2[i - 1], \_T("%d"), cur->data.credit);

outtextxy(320, i \* 22 + h, s2[i - 1]);

\_stprintf(s3[i - 1], \_T("%d"), cur->data.semester);

outtextxy(380, i \* 22 + h, s3[i - 1]);

cur = cur->next;

}

}

SListNode\* SListNode::createSListCourse()

{

ifstream myfile("Course.txt", ios::in | ios::\_Nocreate);

if (!myfile)

{

cerr << "文件打开失败!" << endl;

abort();

}

CMsg cmsg[26];

//文件内容写入内存

for (int i = 0; i < 26; i++)

{

myfile >> cmsg[i].num;

myfile >> cmsg[i].name;

myfile >> cmsg[i].kind;

myfile >> cmsg[i].Coursehour;

myfile >> cmsg[i].credit;

myfile >> cmsg[i].semester;

myfile >> cmsg[i].studentNum;

}

myfile.close();

SListNode\* head = NULL, \* tail = NULL, \* cur;

head = new SListNode;

if (head == NULL)

{

cout << "No memory!" << endl;

return NULL;

}

else

{

head->data = cmsg[0];

head->next = NULL;

tail = head;

}

for (int i = 1; i < N; i++)

{

cur = new SListNode;

if (cur == NULL)

{

cout << "No memory!" << endl;

return head;

}

else

{

cur->data = cmsg[i];

cur->next = NULL;

tail->next = cur;

tail = cur;

}

}

return head;

}

void SListNode::sortByCredit(SListNode\*\* head)

{

SListNode\* cur = \*head;

while (cur->next)

{

SListNode\* curr = \*head;

while (curr->next)

{

if (curr->data.credit > curr->next->data.credit)

{

CMsg temp = curr->data;

curr->data = curr->next->data;

curr->next->data = temp;

}

curr = curr->next;

}

cur = cur->next;

}

}

void SListNode::sortByStuNum(SListNode\*\* head)

{

SListNode\* cur = \*head;

while (cur->next)

{

SListNode\* curr = \*head;

while (curr->next)

{

if (curr->data.studentNum > curr->next->data.studentNum)

{

CMsg temp = curr->data;

curr->data = curr->next->data;

curr->next->data = temp;

}

curr = curr->next;

}

cur = cur->next;

}

}

bool CMsg::operator==(CMsg& right)

{

return num == right.num &&

strcmp(name, right.name) == 0 &&

strcmp(kind, right.kind) == 0 &&

Coursehour == right.Coursehour &&

credit == right.credit &&

studentNum == right.studentNum &&

semester == right.semester;

}

**Stu.h**

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <assert.h>

#include<iostream>

#include <graphics.h>

#include <conio.h>

#include <fstream>

#define N 26

using namespace std;

class Student

{

public:

char num[10];

char name[10];

char password[10];

char \_course[100];

bool operator==(Student& right);

};

class StudentNode

{

public:

Student data;

struct StudentNode\* next;

// 单链表打印

void StudentPrint(StudentNode\* plist, int h = 0);

//创建学生链表

StudentNode\* createStudent();

};

**Stu.cpp**

#include"Stu.h"

TCHAR\* CharToTCHAR1(char\* pChar)

{

TCHAR\* pTchar = nullptr;

int nLen = strlen(pChar) + 1;

pTchar = new wchar\_t[nLen];

MultiByteToWideChar(CP\_ACP, 0, pChar, nLen, pTchar, nLen);

return pTchar;

}

void StudentNode::StudentPrint(StudentNode\* plist, int h)

{

StudentNode\* cur = plist;

int i = 0;

while (cur)

{

i++;

outtextxy(50, i \* 22 + h, CharToTCHAR1(cur->data.num));

outtextxy(150, i \* 22 + h, CharToTCHAR1(cur->data.name));

outtextxy(250, i \* 22 + h, CharToTCHAR1(cur->data.\_course));

cur = cur->next;

}

}

StudentNode\* StudentNode::createStudent()

{

ifstream myfile("Student.txt", ios::in | ios::\_Nocreate);

if (!myfile)

{

cerr << "文件打开失败!" << endl;

abort();

}

Student student[26] = { 0 };

//读取文件到内存

for (int i = 0; i < 26; i++)

{

myfile >> student[i].num;

myfile >> student[i].name;

myfile >> student[i].password;

myfile >> student[i].\_course;

}

myfile.close();

StudentNode\* head = NULL, \* tail = NULL, \* cur;

head = new StudentNode;

if (head == NULL)

{

cout << "No memory!" << endl;

return NULL;

}

else

{

head->data = student[0];

head->next = NULL;

tail = head;

}

for (int i = 1; i < N; i++)

{

cur = new StudentNode;

if (cur == NULL)

{

cout << "No memory!" << endl;

return head;

}

else

{

cur->data = student[i];

cur->next = NULL;

tail->next = cur;

tail = cur;

}

}

return head;

}

bool Student::operator==(Student& right)

{

return strcmp(num, right.num) == 0 &&

strcmp(name, right.name) == 0 &&

strcmp(password, right.password) == 0 &&

strcmp(\_course, right.\_course) == 0;

}

**test.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include"Init.h"

int main()

{

Init init;

init.draw1();

if (init.student)

{

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

init.draw5();

break;

}

}

if (init.administrator)

{

while (true)

{

ExMessage m;

m = getmessage(EM\_MOUSE);

init.draw6();

break;

}

}

return 0;

}