

**课程设计报告**

**健身俱乐部管理**

|  |  |
| --- | --- |
| 学 院 | 计算机与通信工程学院 |
| 专 业 | 电子信息类/计算机类 |
| 班级序号 |  |
| 学 号 |  |
| 姓 名 |  |
| 指导教师 |  |
| 验收日期 | 2022年12月30日 |

核心知识点清单，由学生确认

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数组 | 链表 | 指针 | 文件读写 | 默认参数 | 函数模板 | 多文件 | 类 | 派生 | 虚函数 | 友元函数 | 重载 | 多继承 |
| √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ |

以下为教师评分表，学生不可填写

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 程序质量  （60%） | 课程设计报告（20%） | 答辩效果  （20%） | 总分 | 等级 |
|  |  |  |  |  |

评分标准与说明：

1. 程序质量（60%）包含程序正确性与所用知识点数量（40%），代码可读性（10%）与界面友好性（10%）。
2. 课程设计报告（20%）要求排版规范，模块设计有文字说明，图、表、代码清单要有序号和名字。
3. 现场答辩（20%）要求根据学生制作的PPT、讲述清晰、回答问题等情况综合评分。
4. 收齐所有纸质报告的同时，要求学委收集所有学生的代码工程、报告电子版和答辩PPT以备存档。

**设计要求：**

**1、问题描述**

《健身俱乐部会员管理》主要对健身俱乐部的员工与会员信息进行管理。将俱乐部人员分为员工和会员，员工分为经理、训练经理、销售经理、健身教练、销售人员等几类，会员分为基础会员和VIP会员。要求具备上述人员信息的添加、删除、修改、查询、浏览、保存等功能，能很好的达到了对健身俱乐部的管理目的。

**2、功能要求**

**（1）添加功能：程序能够添加俱乐部成员记录，提供选择界面供用户选择所要添加的类别。添加记录时，要求编号号要唯一，如果添加了重复编号的人员记录时，则提示数据添加重复并取消添加。**

**（2）查询功能：可根据编号、姓名等信息对已添加的人员记录进行查询，如果未找到，给出相应的提示信息，如果找到，则显示相应的记录信息。**

**（3）显示功能：可显示当前系统中按照分类所有人员的的记录，每条记录占据一行。**

**（4）编辑功能：可根据查询结果对相应的记录进行修改，修改时注意编号的唯一性。**

**（5）删除功能：主要实现对已添加的人员记录进行删除。如果当前系统中没有相应的记录，则提示“记录为空！”并返回操作。**

**（6）统计功能：可分别统计某一类人员信息，也可统计全部人员数量信息。**

**（7）保存功能：可将当前系统中各类记录存入文件中，存入方式任意。**

**（8）读取功能：可将保存在文件中的信息读入到当前系统中，供用户进行使用。**

目录

1 基本功能描述 1

2 设计思路 2

3 软件设计 3

3.1 设计步骤 3

3.2 界面设计 5

3.3 关键功能的实现 5

3.4 程序得意之处 14

3.5 目前存在的问题 14

4 结论与心得体会 15

5 参考文献 16

6 附录 16

6.1 调试报告 16

6.2 测试结果 18

6.3 关键源代码 24

健身俱乐部管理

# 1 基本功能描述

# 《健身俱乐部会员管理》主要对健身俱乐部的员工与会员信息进行管理。将俱乐部人员分为员工和会员，员工分为经理、训练经理、销售经理、健身教练、销售人员等几类，会员分为基础会员和VIP会员。要求具备上述人员信息的添加、删除、修改、查询、浏览、保存等功能，能很好的达到了对健身俱乐部的管理目的。

**（1）添加功能：程序能够添加俱乐部成员记录，提供选择界面供用户选择所要添加的类别。添加记录时，要求编号号要唯一，如果添加了重复编号的人员记录时，则提示数据添加重复并取消添加。**

**（2）查询功能：可根据编号、姓名等信息对已添加的人员记录进行查询，如果未找到，给出相应的提示信息，如果找到，则显示相应的记录信息。**

**（3）显示功能：可显示当前系统中按照分类所有人员的的记录，每条记录占据一行。**

**（4）编辑功能：可根据查询结果对相应的记录进行修改，修改时注意编号的唯一性。**

**（5）删除功能：主要实现对已添加的人员记录进行删除。如果当前系统中没有相应的记录，则提示“记录为空！”并返回操作。**

**（6）统计功能：可分别统计某一类人员信息，也可统计全部人员数量信息。**

**（7）保存功能：可将当前系统中各类记录存入文件中，存入方式任意。**

**（8）读取功能：可将保存在文件中的信息读入到当前系统中，供用户进行使用。**

# 2 设计思路

如图是整个健身俱乐部管理的模块实现流程图，打开健身俱乐部管理程序，输入数据，调用输入数据子程序功能。按下数字按钮后，判断其所属的功能实现，对其进行相关处理，然后在判断是否功能成功正常实现，处理后输出反馈结果。

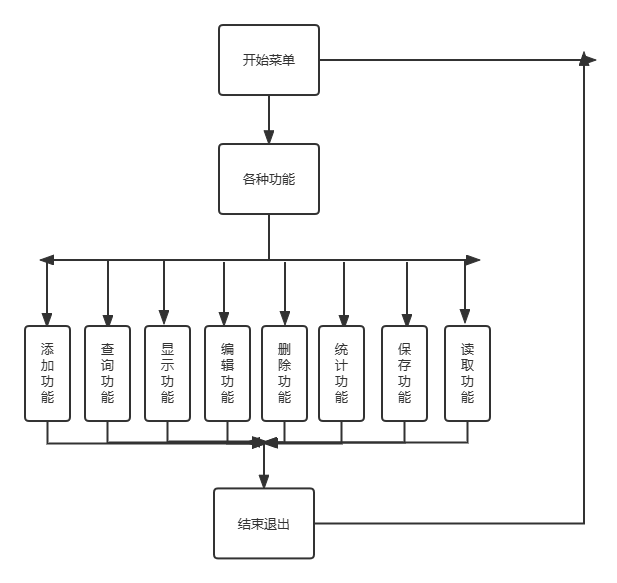
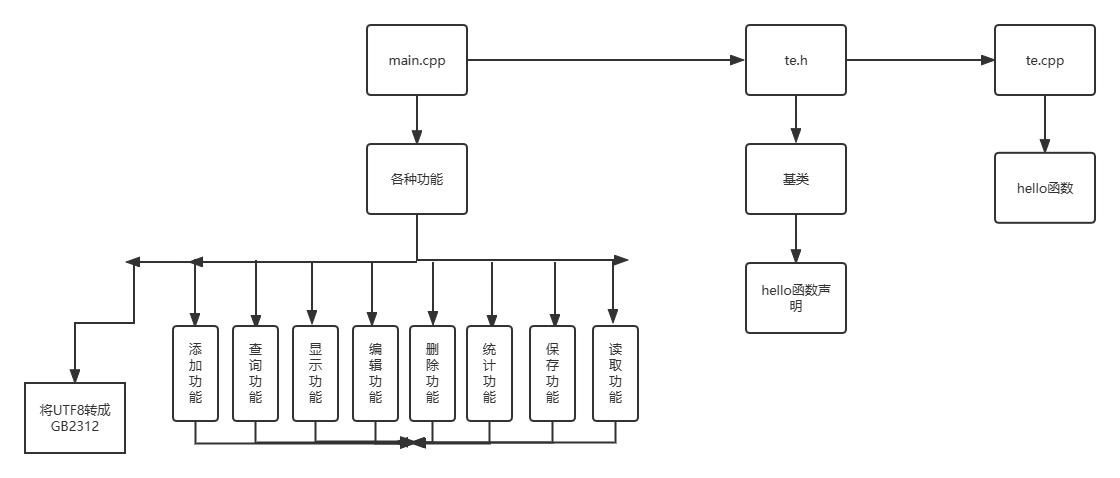


图1 模块实现流程图

# 3 软件设计

## 3.1 设计步骤

下图为工程组织结构图

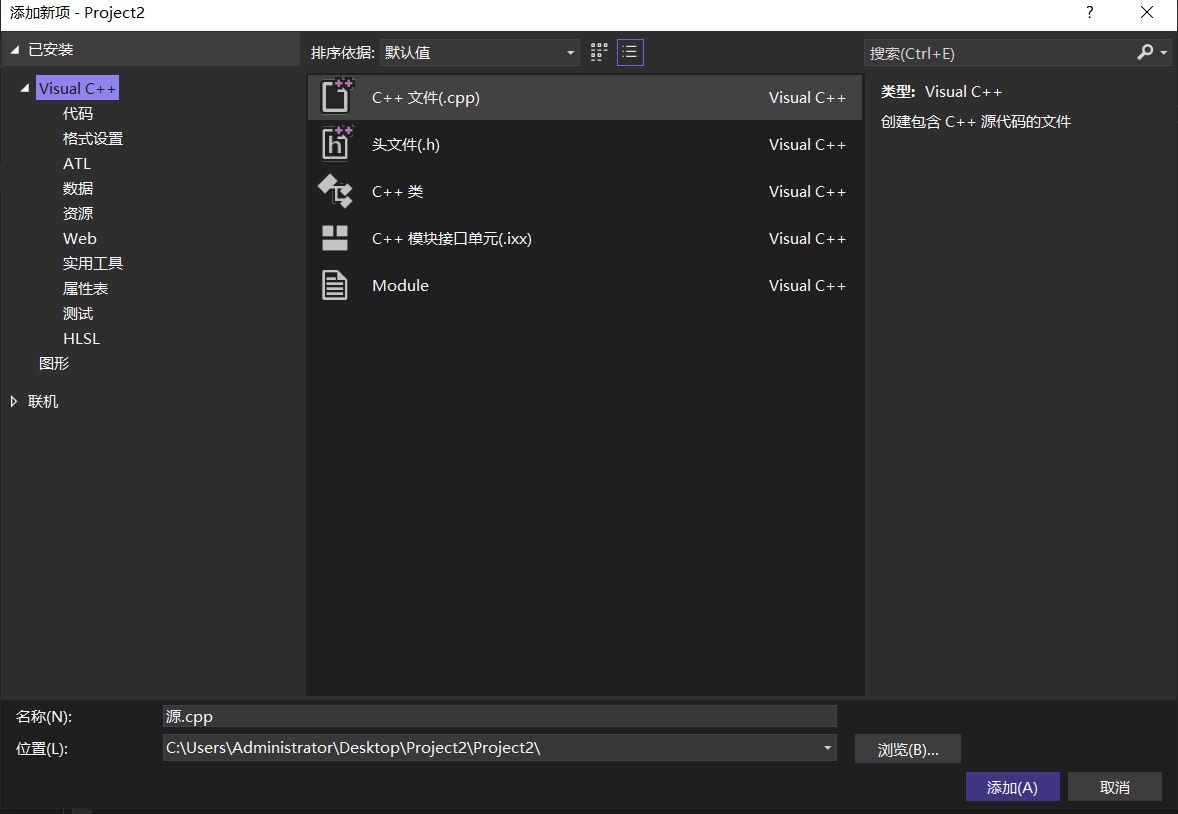


工程组织结构图

1.打开Visual Studio 创建一个新项目，如下设计图

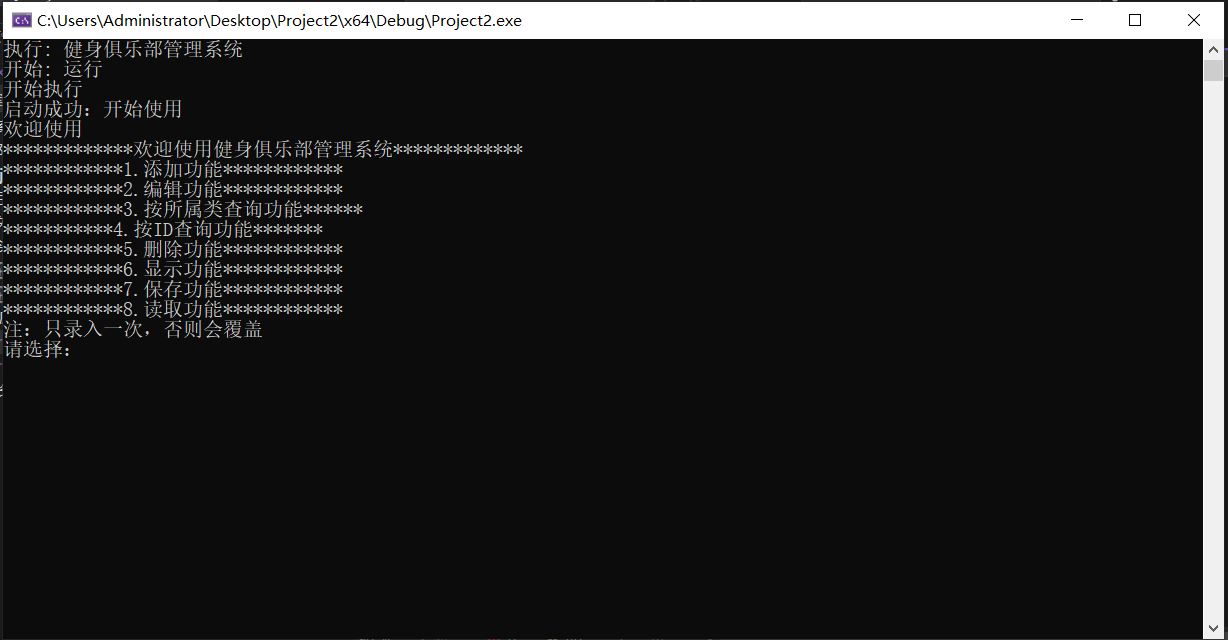


## 为项目添加头文件和源文件，如下设计图



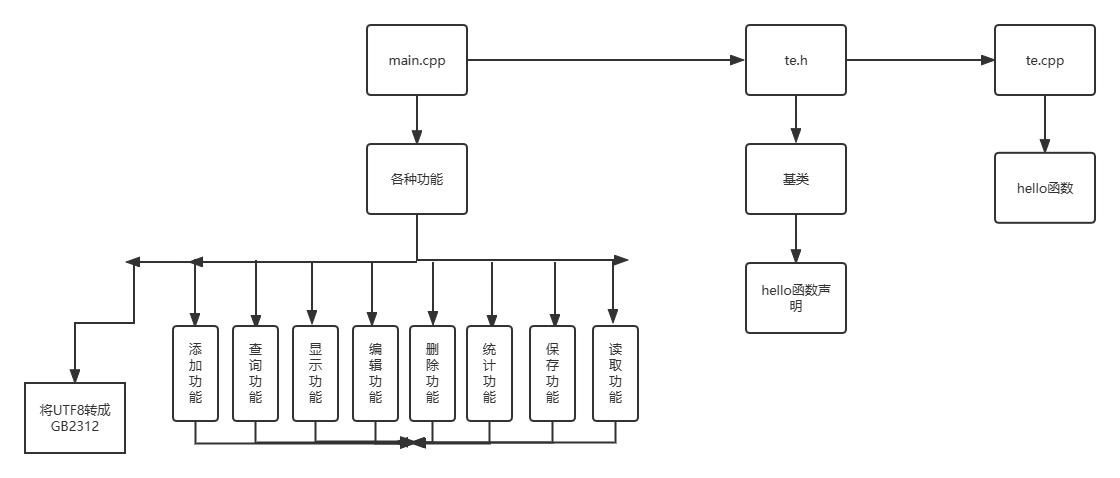
## 3.2 界面设计

运用命令行来实现界面和菜单的设计 如下设计图



## 3.3 关键功能的实现

下图为工程组织结构图



# 工程组织结构图

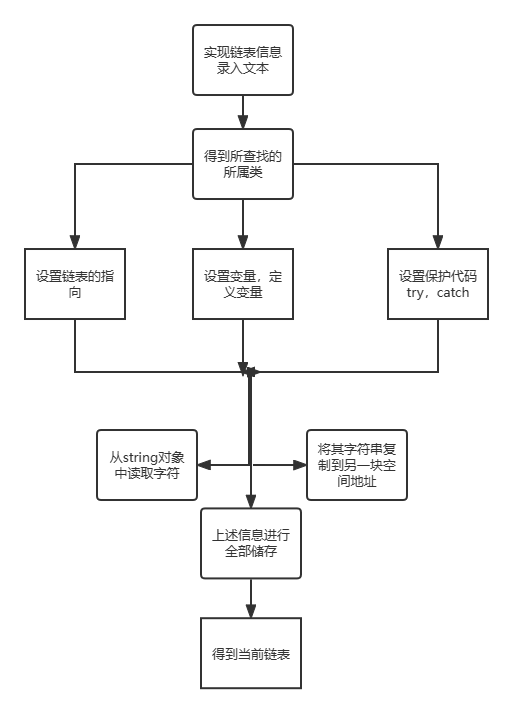
# 1）将UTF8转成GB2312

从文本文件中读取时，编码是[ASCII码](https://so.csdn.net/so/search?q=ASCII%E7%A0%81&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_44517656/article/details/_blank)或者GB2312可能不会出错，但是读取的是UTF8时，就很容易造成乱码，所以每次读取时都最好将读取到的内容转成GB2312。

1. 读取功能的实现

实现链表信息录入文本，得到所查找的所属类，再设置变量，定义变量，设置链表的指向，设置保护代码try，catch。从string对象中读取字符，并将其字符串复制到另一块空间地址，最后对上述信息进行全部储存，得到当前链表。

下图为读取功能的模块的实现流程图

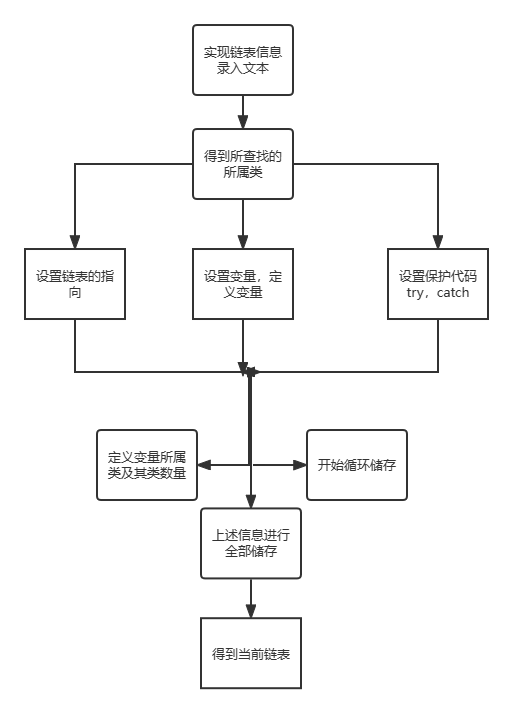


读取功能的模块的实现流程图

3）添加功能的实现

实现链表信息录入文本，得到所查找的所属类，再设置变量，定义变量，设置链表的指向，设置保护代码try，catch。定义变量所属类及其类数量，开始循环储存，最后对上述信息进行全部储存，得到当前链表。

下图为添加功能的模块的实现流程图

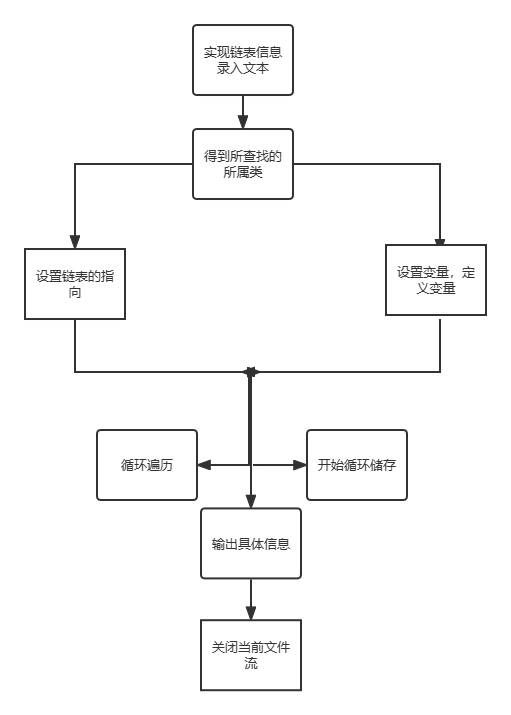


读取功能的模块的实现流程图

4）保存功能的实现

实现链表信息录入文本，得到所查找的所属类，再设置变量，定义变量，设置链表的指向，循环遍历，输出具体信息，关闭当前文件流。

下图为保存功能的模块的实现流程图

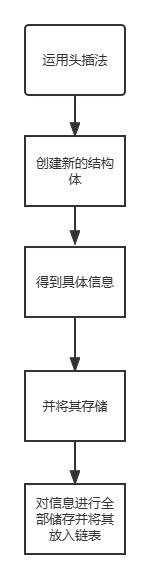


保存功能的模块的实现流程图

1. 编辑功能的实现

运用头插法，创建新的结构体，得到具体信息，并将其存储，对信息进行全部储存并将其放入链表。

下图为编辑功能的模块的实现流程图

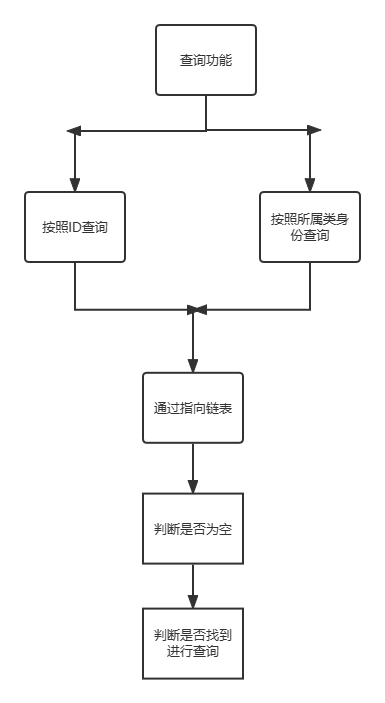


编辑功能的模块的实现流程图

1. 查询功能的实现

将查询功能分为两类，一类为按照ID查询，一类为按照所属类身份查询。二者都是通过指向链表，判断是否为空，判断是否找到进行查询。

下图为查询功能的模块的实现流程图

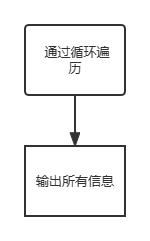


查询功能的模块的实现流程图

1. 显示功能的实现

通过循环遍历，输出所有信息。

下图为显示功能的模块的实现流程图

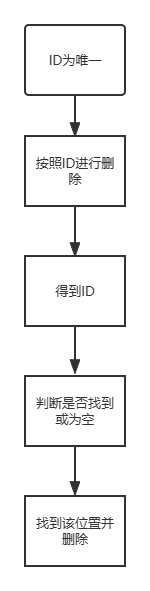


显示功能的模块的实现流程图

1. 删除功能的实现

按照ID进行删除，因为ID为唯一。得到ID，判断是否找到或为空，找到该位置并删除

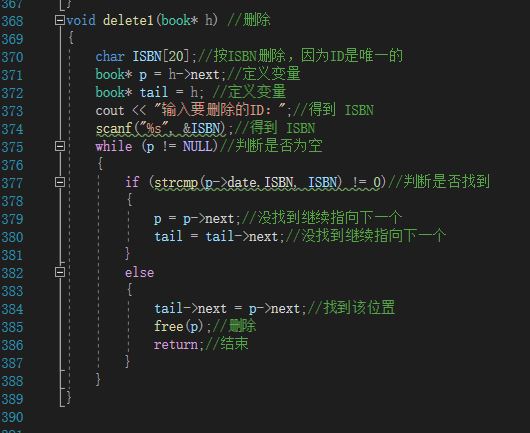
下图为删除功能的模块的实现流程图



删除功能的模块的实现流程图

## 3.4 程序得意之处

下图为程序得意之处，如程序功能图所示。



程序功能图

如上图所示，由于ID的唯一性，故删除功能基于ID查找删除。首先定义链表变量，及其指向，在得到ID后判断是否为空，用strcmp函数判断是否找到。若未找到调整链表令指针不断后移使其指向下一个。若找到该位置则运用free函数释放内存进行删除。最后进行返回。

## 3.5 目前存在的问题

1.运用命令行制作界面。在视觉上令人较为难以接受。存在可改进进行图形化界面的改进提升空间

2.运用命令行制作界面。在鼠标和内容的可交互性上存在不足

# 

# 4 结论与心得体会

经过多番测试，得出结论为设计的健身俱乐部管理系统能够完成功能描述内容中的全部功能。

经过一个学期对《C++程序设计》的学习，我学习到了基本的理论知识，了解到了C++语言程序设计的思想，这些知识都为我的课程实践和进一步的学习打下了坚实的基础。 在为期近两周的C++课程设计中，我体会颇多，学到了很多东西。我加强了对C++程序设计这门课程的认识，并且复习了自己以前学习到的知识。这些都使得我对计算机语言的学习有了更深入的认识!总之，通过这次课程设计，我收获颇丰，相信会为自己以后的学习和工作带来很大的好处。像健身俱乐部管理系统这样的程序设计，经历了平时在课堂和考试中不会出现的问题和考验。而这些问题，这并不是我们平时只靠课本，就可以轻易解决的。所以，锻炼了挑战难题，学会用已掌握的知识去解决具体问题的能力，进一步培养了独立思考问题和解决问题的能力。特别是学会了在C++中如何调试程序的方法。

在老师和同学的帮助下，通过自己的努力，终于完成了这次健身俱乐部管理系统的简单课程设计。经过这段时间的编程，对其中的艰辛，我是深有体会。从刚开始的选择程序、理解程序到后来的调试程序以及改进程序这个过程中，我遇到了各种各样的困难和挫折。但是我坚定信念，对自己充满了信心，想尽一切办法克服重重困难。

通过课程设计的训练，我进一步学习和掌握了对程序的设计和编写，从中体会到了面向对象程序设计的方便和巧妙。懂得了在进行编写一个程序之前，要有明确的目标和整体的设计思想。另外某些具体的细节内容也是相当的重要。这些宝贵的编程思想和从中摸索到的经验都是在编程的过程中获得的宝贵财富。这些经验对我以后的编程会有很大的帮助的，我要好好利用。

# 5 参考文献

[1] 揣锦华.面向对象程序设计与VC++实践.西安电子科技大学出版社，2005

[2] 张建宏，汤练兵，马德骏.计算机程序设计基础-C语言程序设计.科学出版社,2006

[3] 陈清华.Visual C++课程设计案例精选与编程指导.东南大学出版社,2004

[4] CSDN [将UTF8转成GB2312\_Mango酱的博客-CSDN博客\_utf8转gb2312](https://blog.csdn.net/weixin_44517656/article/details/107784008?ops_request_misc=&request_id=&biz_id=102&utm_term=UTF8ToGB&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~sobaiduweb~default-0-107784008.blog_rank_default&spm=1018.2226.3001.4450)

# 6 附录

## 6.1 调试报告

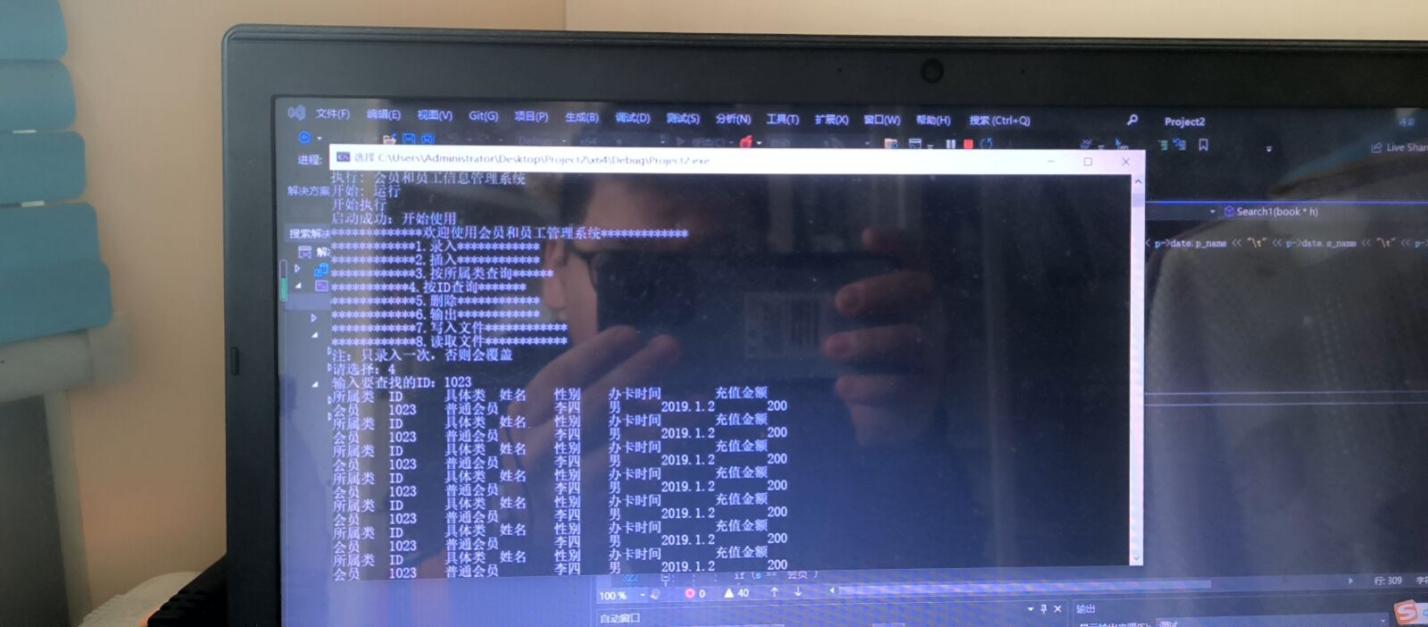
1、编译时提示错误：

Run-Time Check Failure #3 - The variable 'head' is being used without being initialized.

原因为book\* head ;未初始化赋初值，经检查后更改为book\* head = NULL;赋初始值为NULL。

1. 从文本文件中读取时，编码是[ASCII码](https://so.csdn.net/so/search?q=ASCII%E7%A0%81&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_44517656/article/details/_blank)或者GB2312可能不会出错，但是读取的是UTF8时，就很容易造成乱码，所以我们每次读取时都最好将读取到的内容转成GB2312。

故编写函数 UTF8ToGB。

3、

ID查询完信息一直滚动且不返回主菜单。

经检查错误为未检查return；添加return；后错误改正。

## 

## 6.2 测试结果

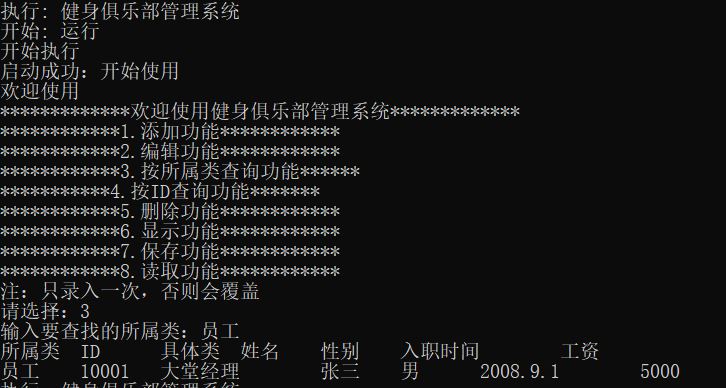
1.输入数字1，实现添加功能。如结果图1。



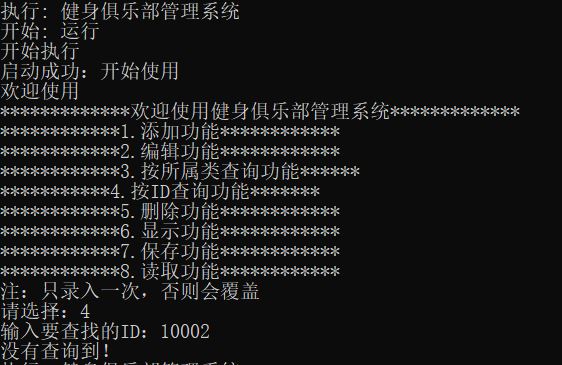
结果图1

1. 输入数字3，实现按所属类查询功能，如结果图2-1.。

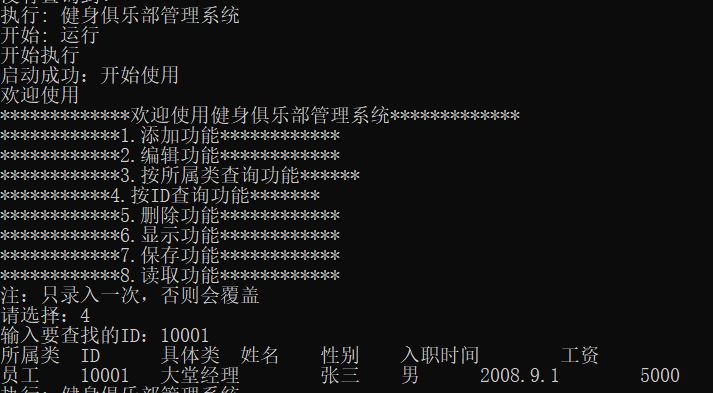
输入数字4，实现按ID查询功能，如结果图2-2, 2-3。



结果图2-1

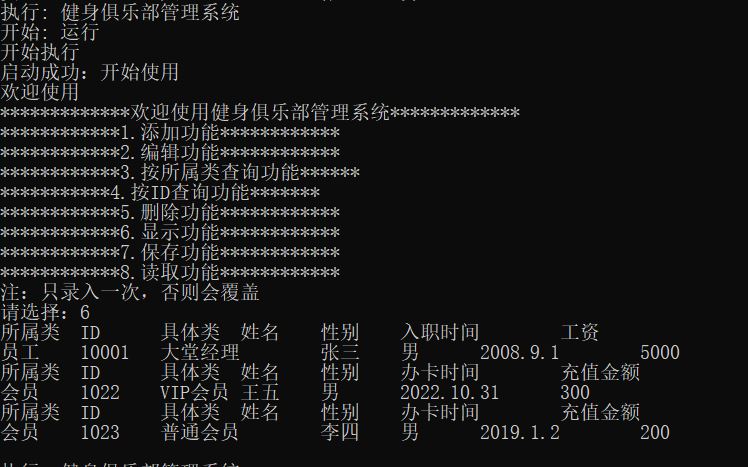


结果图2-2



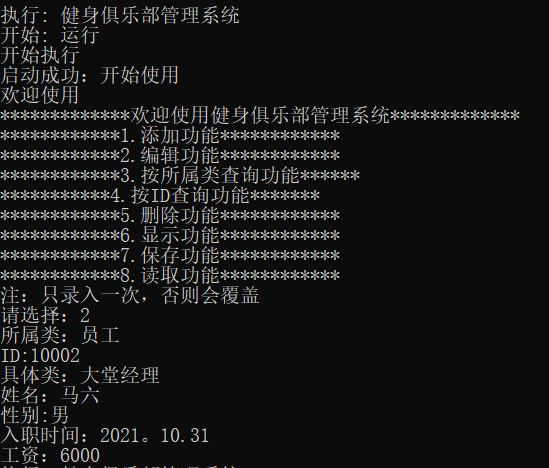
结果图2-3

3.输入数字6，实现显示功能，如结果图3。



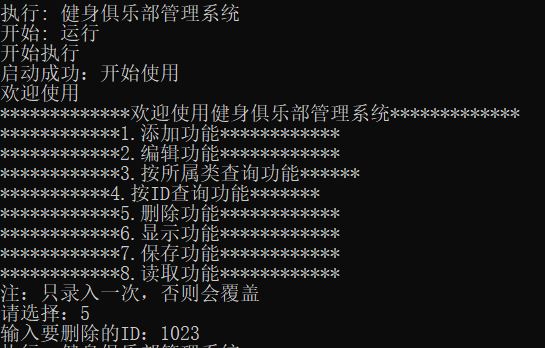
结果图3

4.输入数字2，实现编辑功能，如结果图4。



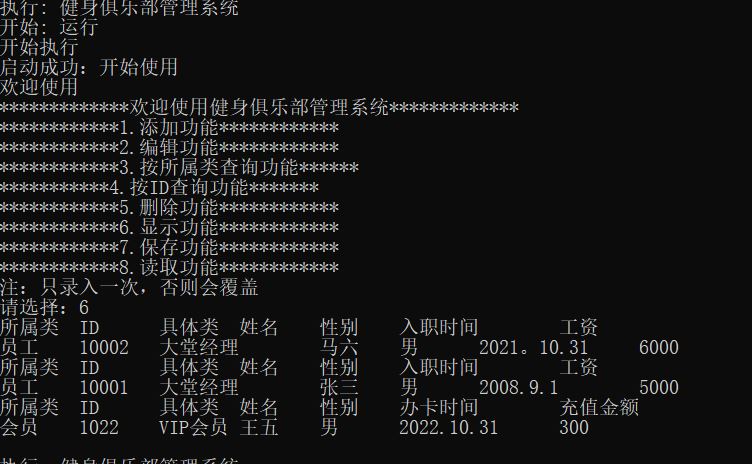
结果图4

5.输入数字5，实现删除功能，如结果图5。



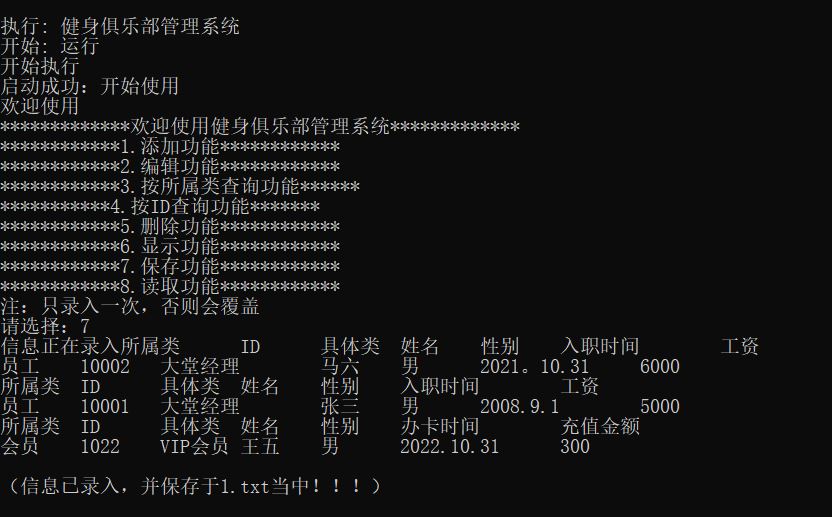
结果图5

6.输入数字6，实现显示功能，如结果图6。

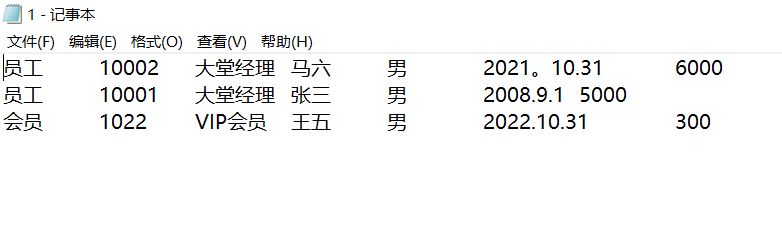


结果图6

7.输入数字7，实现保存功能，如结果图7-1, 7-2。

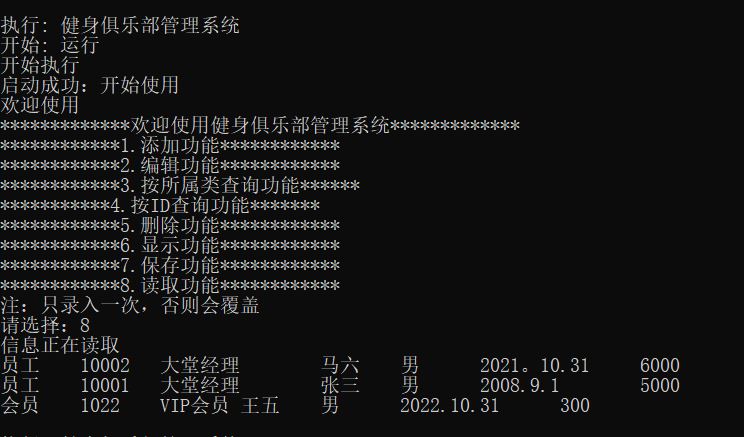


结果图7-1

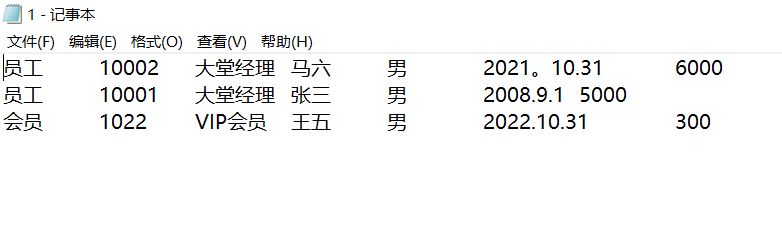


结果图7-2

8.输入数字8，实现读取功能，如结果图8-1， 8-2。



结果图8-1



结果图8-2

## 6.3 关键源代码

1、将UF8T转化为GB312：

string UTF8ToGB(const char\* str)//我们为什么需要这个函数呢，因为我们从文本文件中读取时，编码是ASCII码或者GB2312可能不会出错，

//但是读取的是UTF8时，就很容易造成乱码，所以我们每次读取时都最好将读取到的内容转成GB2312。

{

string result;

WCHAR\* strSrc;

LPSTR szRes;

//获得临时变量的大小

int i = MultiByteToWideChar(CP\_UTF8, 0, str, -1, NULL, 0);

strSrc = new WCHAR[i + 1];

MultiByteToWideChar(CP\_UTF8, 0, str, -1, strSrc, i);

//获得临时变量的大小

i = WideCharToMultiByte(CP\_ACP, 0, strSrc, -1, NULL, 0, NULL, NULL);

szRes = new CHAR[i + 1];

WideCharToMultiByte(CP\_ACP, 0, strSrc, -1, szRes, i, NULL, NULL);

result = szRes;

delete[]strSrc;

delete[]szRes;

return result;

}

2、程序运行时菜单和界面：

void menu()//显示菜单

{

//重载

printData pd;

// 输出整数

pd.print("健身俱乐部", "管理系统");//默认参数

pd.print("运行");

//虚函数

A\* a = new B();

a->fool(); // 在这里，a虽然是指向A的指针，但是被调用的函数(fool)却是B的!

// 派生类 //多继承

extern class Rectangle : public Shape

{

public:

string getArea()

{

return width + height;

}

};

Rectangle Re;

Re.setWidth("开始");

Re.setHeight("使用");

// 输出对象

cout << "启动成功：" << Re.getArea() << endl;

cout << Hello<string>("欢迎使用") << endl;//多文件//函数模板

Box box;

// 使用成员函数设置

box.setWidth("欢迎使用健身俱乐部管理系统");

// 使用友元函数输出

printWidth(box);

cout << "\*\*\*\*\*\*\*\*\*\*\*\*1.添加功能\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*2.编辑功能\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*3.按所属类查询功能\*\*\*\*\*\*\n";

cout << "\*\*\*\*\*\*\*\*\*\*\*4.按ID查询功能\*\*\*\*\*\*\*\n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*5.删除功能\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*6.显示功能\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*7.保存功能\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*8.读取功能\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "注：只录入一次，否则会覆盖\n";

}

3、用户的功能判断

void choose(book\* h)//用户的功能选择判断

{

int i;

int a = 1;

bool flag = FALSE;

while (a > 0)

{

menu(); //显示菜单

cout << "请选择：";

scanf("%d", &i); //得到选择的功能序号

switch (i) //根据序号找到对应的功能

{

case 1:

h = GreatLink(); //录入

flag = TRUE;

break;

case 2:

if (not flag)

{

h = Read\_file();

flag = TRUE;

}

Insert(h);//插入

break;

case 3:

if (not flag)

{

h = Read\_file();

}

Search(h); //按所属类查询

break;

case 4:

if (not flag)

{

h = Read\_file();

}

Search1(h);//按ID查询

break;

case 5:

if (not flag)

{

h = Read\_file();

flag = TRUE;

}

delete1(h);//删除会员信息

break;

case 6:

if (not flag)

{

h = Read\_file();

}

PrintLink(h);//输出所有会员信息

break;

case 7:

if (not flag)

{

h = Read\_file();

}

write\_file(h);//保存会员信息

break;

case 8:

if (not flag)

{

h = Read\_file();

}

Read\_file();//读取会员信息

break;

default:

cout << "无效的命令！\n";

a = -1;//跳出循环条件

break;

}

}

}

4、 读取功能的实现

book\* Read\_file()

{ //实现链表信息录入文本

cout << "信息正在读取" << endl;//得到查找的所属类 ID具体类

ifstream infile("1.txt");

string data;

book\* h, \* tail, \* p; //设置变量

h = tail = (book\*)malloc(sizeof(book)); //定义变量 ，设置链表的指向

h->next = NULL; //创建第一个信息

try

{

// 保护代码

while (getline(infile, data)) {

//cout << "read from file:" << data << endl;

std::istringstream is(data);//作用是从string对象data中读取字符

string n1;

p = (book\*)malloc(sizeof(book)); //得到所属类 ID具体类

string vstr[7];

int cn = 0;

while (is >> n1)

{

/\* n1= UTF8ToGB(n1.c\_str());\*/

vstr[cn] = n1;

cn++;

}

//pstm

strcpy(p->date.name, vstr[0].c\_str());//是将一个字符串复制到另一块空间地址中 的函数，‘\0’是停止拷贝的终止条件，同时也会将 '\0' 也复制到目标空间。

strcpy(p->date.ISBN, vstr[1].c\_str());

strcpy(p->date.price, vstr[2].c\_str());

strcpy(p->date.p\_name, vstr[3].c\_str());

strcpy(p->date.s\_name, vstr[4].c\_str());

strcpy(p->date.t\_name, vstr[5].c\_str());

strcpy(p->date.m\_name, vstr[6].c\_str());

cout << p->date.name << "\t" << p->date.ISBN << "\t" << p->date.price << "\t" << p->date.p\_name << "\t"

<< p->date.s\_name << "\t" << p->date.t\_name << "\t" << p->date.m\_name << endl;

/\* while(is >>n1)

{

cout << n1 << endl ;

p->date.name = n1;

}\*/

p->next = NULL;

tail->next = p; //对于上述的信息进行全部的存储

tail = p; //得到当前的链表

}

}

catch (exception)

{

// catch 块

return h;

}

return h;

}

5、 添加功能的实现

book\* GreatLink() //定义所属类 ID具体类 的录入函数

{

book\* h, \* tail, \* p; //设置变量

try

{

h = Read\_file();

tail = Read\_file();

h = h->next;

tail = tail->next;

}

catch (exception)

{

h = tail = (book\*)malloc(sizeof(book)); //定义变量 ，设置链表的指向

}

//cout << h->date.name << h->date.ISBN << h->date.price << endl;

//h->next = NULL; //创建第一个信息

int n = 0; //定义变量记录所属类 ID具体类 数量

cout << "输入要录入的人数：";

scanf("%d", &n); //得到所属类 ID具体类 数量

for (int i = 0; i < n; i++) //开始循环存储

{

p = (book\*)malloc(sizeof(book)); //得到所属类 ID具体类

cout << "所属类：";

scanf("%s", &p->date.name); //存储所属类 ID具体类

string s;

s = p->date.name;

if (s == "会员")

{

cout << "ID:"; //得到所属类 ID具体类

scanf("%s", &p->date.ISBN); //存储I所属类 ID具体类

cout << "具体类："; //得到所属类 ID具体类

scanf("%s", &p->date.price); //存储所属类 ID具体类

cout << "姓名：";

scanf("%s", &p->date.p\_name); //存储所属类 ID具体类

cout << "性别:"; //得到所属类 ID具体类

scanf("%s", &p->date.s\_name); //存储I所属类 ID具体类

cout << "办卡时间："; //得到所属类 ID具体类

scanf("%s", &p->date.t\_name); //存储所属类 ID具体类

cout << "充值金额："; //得到所属类 ID具体类

scanf("%s", &p->date.m\_name); //存储所属类 ID具体类

}

if (s == "员工")

{

cout << "ID:"; //得到所属类 ID具体类

scanf("%s", &p->date.ISBN); //存储I所属类 ID具体类

cout << "具体类："; //得到所属类 ID具体类

scanf("%s", &p->date.price); //存储所属类 ID具体类

cout << "姓名：";

scanf("%s", &p->date.p\_name); //存储所属类 ID具体类

cout << "性别:"; //得到所属类 ID具体类

scanf("%s", &p->date.s\_name); //存储I所属类 ID具体类

cout << "入职时间："; //得到所属类 ID具体类

scanf("%s", &p->date.t\_name); //存储所属类 ID具体类

cout << "工资："; //得到所属类 ID具体类

scanf("%s", &p->date.m\_name); //存储所属类 ID具体类

}

p->next = tail->next;//对于上述的信息进行全部的存储

tail->next = p;//将信息放入到链表中

//tail = p; //得到当前的链表

}

h->next = tail;

return h; //返回总的链表

}

6、 实现保存功能：

void write\_file(book\* h)

{ //实现链表信息录入文本

FILE\* fp;

cout << "信息正在录入";//得到查找所属类 ID具体类

fp = fopen("1.txt", "w");

if (NULL == fp)

{

cout << "该文件打开失败！";

}

book\* p;//定义变量

//cout << "所属类\tID\t具体类\n";//输出

for (p = h->next; p != NULL; p = p->next)//循环遍历

{

string s;

s = p->date.name;

if (s == "会员")

{

cout << "所属类\tID\t具体类\t姓名\t性别\t办卡时间\t充值金额\t\n";//输出

}

if (s == "员工")

{

cout << "所属类\tID\t具体类\t姓名\t性别\t入职时间\t工资\t\n";//输出

}

cout << p->date.name << "\t";//输出所属类 ID具体类

cout << p->date.ISBN << "\t";//输出ID

cout << p->date.price << "\t";//输出具体类

cout << p->date.p\_name << "\t";//姓名

cout << p->date.s\_name << "\t";//性别

cout << p->date.t\_name << "\t";//

cout << p->date.m\_name << "\t";//

cout << "\n";//换行

fprintf(fp, "%s\t%s\t%s\t%s\t%s\t%s\t%s\n", p->date.name, p->date.ISBN, p->date.price, p->date.p\_name, p->date.s\_name, p->date.t\_name, p->date.m\_name);//打印输出到文件

}

fclose(fp);//关闭当前文件流

cout << "（信息已录入，并保存于1.txt当中！！！）\n\n";

}

7、实现编辑功能：

void Insert(book\* h)//插入(头插法)

{

book\* p; //创建一个新的结构体

p = (book\*)malloc(sizeof(book)); //得到所属类 ID具体类

cout << "所属类：";//存储类名

scanf("%s", &p->date.name);//得到类名

string s;

s = p->date.name;

if (s == "会员")

{

cout << "ID:"; //得到所属类 ID具体类

scanf("%s", &p->date.ISBN); //存储I所属类 ID具体类

cout << "具体类："; //得到所属类 ID具体类

scanf("%s", &p->date.price); //存储所属类 ID具体类

cout << "姓名：";

scanf("%s", &p->date.p\_name); //存储所属类 ID具体类

cout << "性别:"; //得到所属类 ID具体类

scanf("%s", &p->date.s\_name); //存储I所属类 ID具体类

cout << "办卡时间："; //得到所属类 ID具体类

scanf("%s", &p->date.t\_name); //存储所属类 ID具体类

cout << "充值金额："; //得到所属类 ID具体类

scanf("%s", &p->date.m\_name); //存储所属类 ID具体类

}

if (s == "员工")

{

cout << "ID:"; //得到所属类 ID具体类

scanf("%s", &p->date.ISBN); //存储I所属类 ID具体类

cout << "具体类："; //得到所属类 ID具体类

scanf("%s", &p->date.price); //存储所属类 ID具体类

cout << "姓名：";

scanf("%s", &p->date.p\_name); //存储所属类 ID具体类

cout << "性别:"; //得到所属类 ID具体类

scanf("%s", &p->date.s\_name); //存储I所属类 ID具体类

cout << "入职时间："; //得到所属类 ID具体类

scanf("%s", &p->date.t\_name); //存储所属类 ID具体类

cout << "工资："; //得到所属类 ID具体类

scanf("%s", &p->date.m\_name); //存储所属类 ID具体类

}

p->next = h->next;//对于上述的信息进行全部的存储

h->next = p;//将信息放入到链表中

}

8、实现查询功能：

void Search(book\* h)//查找

{

char name[20];

book\* p = h->next; //指向链表

cout << "输入要查找的所属类：";//输出

scanf("%s", &name);//得到查找的所属类 ID具体类

while (p != NULL)//判断是否为空

{

if (strcmp(p->date.name, name) != 0) //判断是否找到

{

p = p->next; //没找到继续指向下一个

}

else //找到输出

{

string s;

s = p->date.name;

if (s == "会员")

{

cout << "所属类\tID\t具体类\t姓名\t性别\t办卡时间\t充值金额\t\n";//输出

}

if (s == "员工")

{

cout << "所属类\tID\t具体类\t姓名\t性别\t入职时间\t工资\t\n";//输出

}

cout << p->date.name << "\t" << p->date.ISBN << "\t" << p->date.price << "\t" << p->date.p\_name << "\t" << p->date.s\_name << "\t" << p->date.t\_name << "\t" << p->date.m\_name << "\n";

return;

}

}

if (p == NULL) //找到底或者为空，则没有找到

{

cout << "没有查询到！\n";//输出

}

}

void Search1(book\* h)//ID查找

{

char ISBN[20];

book\* p = h->next;//指向链表

cout << "输入要查找的ID：";//得到查找的所属类 ID具体类

scanf("%s", &ISBN);

while (p != NULL)//判断是否为空

{

if (strcmp(p->date.ISBN, ISBN) != 0)//判断是否找到 比较两个字符串的大小，一个字符一个字符比较，按ASCLL码比较

{

p = p->next;//没找到继续指向下一个

}

else//找到输出

{

string s;

s = p->date.name;

if (s == "会员")

{

cout << "所属类\tID\t具体类\t姓名\t性别\t办卡时间\t充值金额\t\n";//输出

}

if (s == "员工")

{

cout << "所属类\tID\t具体类\t姓名\t性别\t入职时间\t工资\t\n";//输出

}

cout << p->date.name << "\t" << p->date.ISBN << "\t" << p->date.price << "\t" << p->date.p\_name << "\t" << p->date.s\_name << "\t" << p->date.t\_name << "\t" << p->date.m\_name << "\n";

return;

}

}

if (p == NULL)//找到底或者为空，则没有找到

{

cout << "没有查询到！\n";//输出

}

}

9、显示功能的实现

void PrintLink(book\* h)//输出所有信息

{

book\* p;//定义变量

for (p = h->next; p != NULL; p = p->next)//循环遍历

{

string s;

s = p->date.name;

if (s == "会员")

{

cout << "所属类\tID\t具体类\t姓名\t性别\t办卡时间\t充值金额\t\n";//输出

}

if (s == "员工")

{

cout << "所属类\tID\t具体类\t姓名\t性别\t入职时间\t工资\t\n";//输出

}

cout << p->date.name << "\t" << p->date.ISBN << "\t" << p->date.price << "\t" << p->date.p\_name << "\t" << p->date.s\_name << "\t" << p->date.t\_name << "\t" << p->date.m\_name << "\n";

}

}

10、删除功能的实现：

void delete1(book\* h) //删除

{

char ISBN[20];//按ISBN删除，因为ID是唯一的

book\* p = h->next;//定义变量

book\* tail = h; //定义变量

cout << "输入要删除的ID：";//得到 ISBN

scanf("%s", &ISBN);//得到 ISBN

while (p != NULL)//判断是否为空

{

if (strcmp(p->date.ISBN, ISBN) != 0)//判断是否找到

{

p = p->next;//没找到继续指向下一个

tail = tail->next;//没找到继续指向下一个

}

else

{

tail->next = p->next;//找到该位置

free(p);//删除

return;//结束

}

}

}

1. 多文件编程菜单：

//box类位于#include "te.h"

// 成员函数定义

void Box::setWidth(string wid)

{

width = wid;

}

// 请注意：printWidth() 不是任何类的成员函数

void printWidth(Box box)

{

/\* 因为 printWidth() 是 Box 的友元，它可以直接访问该类的任何成员 \*/

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*" << box.width << "\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

1. 基类和多继承：

// 基类

class Shape

{

public:

void setWidth(string w)

{

width = w;

}

void setHeight(string h)

{

height = h;

}

protected:

string width;

string height;

};

class A

{

public:

virtual void fool()//虚函数

{

cout << "开始执行" << endl;

}

};

class B :public A

{

public:

void fool()

{

cout << "开始执行" << endl;

}

};

class printData//重载

{

public:

void print(string a, string b) {

cout << "执行: " << a + b << endl;

}

void print(string a) {

cout << "开始: " << a << endl;

}

};