浙江工业大学

大类基础课程大型实验

2017/2018(2)

实验题目 学校开课查询系统

学生姓名 _____

学生学号 _____

学生班级 _____

任课教师 _____

提交日期 _____

计算机科学与技术学院

学校开课查询系统 实验报告

一、 大型实验的内容

学校开课查询系统用于进行学校开课信息的查询以及学生的选课操作，要求完成的主要的功能包括学生管理、课程管理、管理人员管理。

1、设计菜单实现功能选择；

2、能够对课程信息进行输入、修改、删除操作；

3、按给定的条件（编号、名称、任课教师、开课院系等）查询课程信息；

4、能对开课信息按学分、开课学院排序整理；

5、以文件形式保存相关信息，可以读取默认文件中的信息进行查询等操作。

我在此之上添加了学生的退选课操作

二、 运行环境

学校开课查询系统在 codeblocks17.12 平台下开发，操作系统：Windows 10。

硬件环境：(备注：可以查看“计算机”属性)

处理器：Intel(R) Core(TM) i7-2557M CPU @ 1.70GHz 1.70GHz

内存：4.00GB

系统类型：64 位操作系统

三、 实验课题分析（主要的模块功能、流程图）

3.1 学校开课查询的主要功能

学校开课查询主要功能为：学生管理、课程管理、管理人员管理，可以完成查询开课信息，对开课信息进行排序，学生，管理人员登陆，修改学生，课程信息，学生退，选课等操作

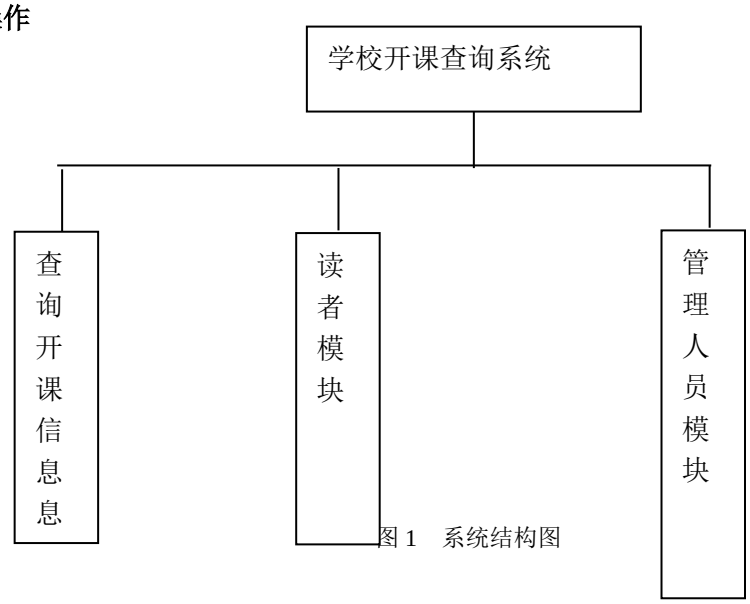


图 1 系统结构图

系统各模块的功能具体描述为：

1、 查询开课信息模块

这部分可以对开课信息进行查询与排序之后输出的操作，但无法对课程进行修改

2、 读者模块

读者必须先进行登录，登录失败之后，无法进行操作，登录成功之后可以查询课程信息，并按编号进行选课，如果已经选了，可以进行退选，并且可以展示个人信息，修改密码

3、 管理员模块

必须先登录，登录失败则退回上一层

登录成功

管理员模块可以对课程信息进行修改，删除，添加操作，也可以对学生信息进行修改添加，删除操作

3.2 系统分析及设计

系统涉及对象有两个基本类：课程类和学生类主要涉及操作如下所示

表 1 人员类涉及的操作

对象	涉及的对象操作	
学生	查询读者信息(个人信息和学生选课情况)	
	查询课程信息	
	退选与选课操作	
管理人员	维护课程信息	添加课程信息
		删除课程信息
		修改课程信息
	维护学生信息	添加学生信息
		修改学生信息
		删除学生信息

可以采用面向对象的方式实现图书管理系统，根据不同的使用权限，使用对象分为学生、管理员与一般访客。系统的主要的类结构如图 2 所示。

学生	课程	学生链表	课程链表	管理员
学生数据	课程数据	链表表头	链表表头	数据
学生类指针（*next）	课程类指针（*next）	链表大小	链表大小	

图 2 系统主要类结构图

分别设计学生类、管理员类，学生链表类，课程链表类，管理员类

用文本文件进行数据的保存，需要保存的数据主要包括图书数据、用户数据（包括读者、、管理员），实现所有的文本操作相关的功能。

3.3 系统的实现

（1）类的编写

系统工程名为：学生开课管理系统。包含了 student 类，course 类，manager 类三个基本类以及相对应的链表类 stulist 类（学生链表类）及 courlist（课程链表类）。而这些人用户都有相应的用户编号和密码，运用时将三个基本类中的用户名和用户密码分别与编号和密码（password）对应。

具体类结构声明如下：

- **student 类：**

```
#ifndef STUDENT_H
#define STUDENT_H
#include<string>
using namespace std;
class student
{
    public:
        student(string,string,int,int,string,int,int * );
        student(string,string,int,int,string);
        virtual ~student();
        void display();//展示个人信息
        void add(int temp);//将课程编号编入 subject 末尾
        student(const student &stu1);
        student *next;
        string getid();//得到学号
        string getname();
        int getage();
        int getey();
        string getpassword();
        int getcounting();
        void changepassword();//修改密码
        void choosecourse(int no);//选课或取消已选的课
```

```
        int existcourse(int no);
        void cover(int no);//从 no 后开始一个一个往前移
        int *subject;//存放学生所选课程的学生编号
        void setdata(string id,string na,int a,int ey,string pa);
    protected:
        string studentid;//学号
        string name;
        int age;
        int enrollyear;//入学年份
        string password;//密码
        int counting;//已选课科目数量
};

#endif // STUDENT_H
```

course 类

```
    class course
    {
        public:
            course *next;//
            course(int,string,string,double,string,string,int ,int    );
            virtual ~course();
            void display();
            void shift(string nm,string tnm,double s,string na,string fa);
            course & operator =(course &c);
            int getnumber();
            string getname();
            string getteacher();
            double getscore();
            string getnature();
            string getfaculty();
            int getcur();
            int getmaxp();

        protected:
            int number;//课程编号
```

```
        string name;
        string teachername;//任课教师
        double score;//学分
        string nature;//课程性质
        string faculty;//开课院系
        int currentperson;//已选该课的人数
        int maxperson;//该科可选的最大人数
};.....//罗列所有类的声明
```

- **manager 类：**

```
class manager
{
    public:
        manager(string ,string);
        virtual ~manager();
        bool login();
        void shiftpass();
    protected:
        string account;
        string password;
};
```

（2）链表的使用

系统实现采用文件的输入输出流对文本数据进行读取与写入，但是由于学生信息、课程信息、都是一个数据的集合，于是对数据的存储组织使用了单向链表。

因为图书管理系统在登录、查找、修改、添加的时候都需要处理大量的数据，所以使用链表十分必要。以课程信息为例，在 course 的基础上定义一个对应的 courlistt 类来管理 kecheng 数据，具体的结构声明如下：

```
class courlist
{
    public:
        courlist();
        virtual ~courlist();
        void add(course &cou);//向末尾插入
        course *findnumber(int no);//按编号查找
        void findname(string n);//按名字查找
        void findfaculty(string f);//按院系查找
        void findteacher(string t);//按教师名字查找
        void ultimatefind();//集合所有查找
        void show();//展示所有课程信息
        void write();//写入文件
```

```
        void sortscore();//按学分排序
        void sortfaculty();//按开课学院排序
        void ultimatesort();//排序
        void change(course *target);
        void removecourse(int no);
    protected:
        course *head;
        int size;

};
```

在运用时，令当前图书的 next 结点指向新的图书结点，即结点的指针 next 保存新的图书结点的地址（如下图 3 所示），以此类推，所有图书信息就通过链表的形式串联起来了。

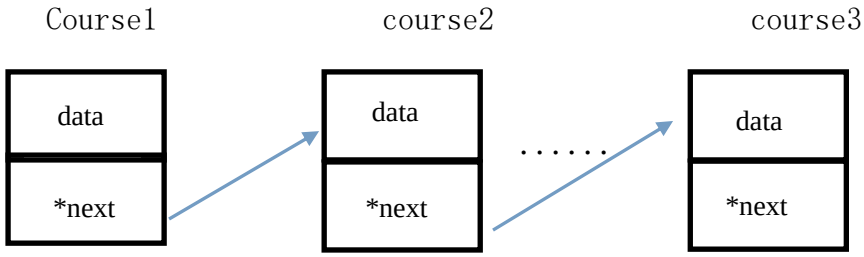


图 3 图书链表的建立

学生类（student）的链表设置与图书的类似，也是通过定义类的指针变量，通过指向下一个类的地址将信息串联起来。即在 student 的基础上定义 stulist，各个类的结构声明如下：

- **stulist 类：**

```
class stulist
{
    public:
        stulist();
        virtual ~stulist();
        void addstu(student & stu2);
        student *search(string no);//°´±àºÅ²éÕÒ
        student * login();
        void write();
        student * findstu(string no);
        void deletenumber(int no);//删除所有选了 no 课的号码
        void changedata(string no);//修改特定学生数据
        void deletedata(string no);//删除特定学生
    protected:
        student *head;
        int size;

};
```

图书管理系统的信息的管理就具体表现为链表的操作。拿图书信息来说，图书信息的查找、修改、添加和删除与链表的查找、修改、添加、删除对应。

- **课程的查找：**

在用户在查询课程时，有四种不同的模式，见图 4：

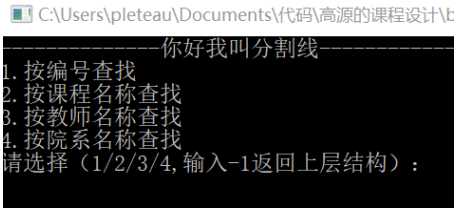


图 4 图书查找界面

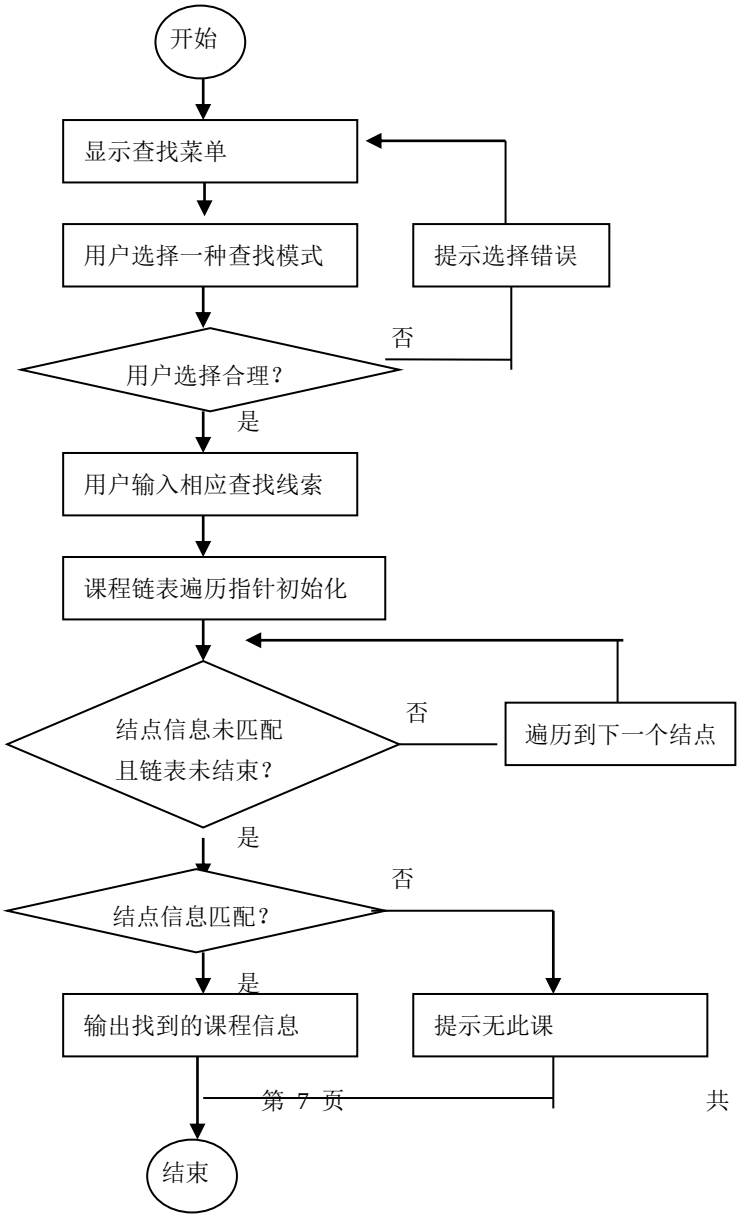
这三种情况都需要对链表中的所有数据进行顺序的搜索。 例如按课程名称的查找，定义一个 Book 类指针变量 p 并对其初始化，使其为 head。

```
course *p=head;
    while(p->getnumber()!=no&&p->next!=NULL){p=p->next;}
    if(p->getnumber()==no){return p;}
    else{cout<<"无此课"<<endl;}
    return NULL;
```

从第一个 Booklist 类的第一个节点开始，会将链表的课程名称和用户输入的书名比较，不一致时，p=p->next，与第二个数据对比，以此类推，直到找到相同的书名，此时调用函数 p->display()输出关于该书籍书籍的所有信息，即它的 7 个变量的数据。

如果没有找到可以匹配的，则 cout<<"无此课"<<endl；并结束循环。

图书查找的流程图如下（备注：选择合适的工具绘制程序流程图）：



……//采用相仿的模式描述其他链表操作的实现

- **课程的修改：**
- **课程的添加：**
- **课程的删除：**
- **学生信息的修改删除添加**

（3）交互界面以及登录菜单的实现

系统运行开始的界面如图 5 所示：

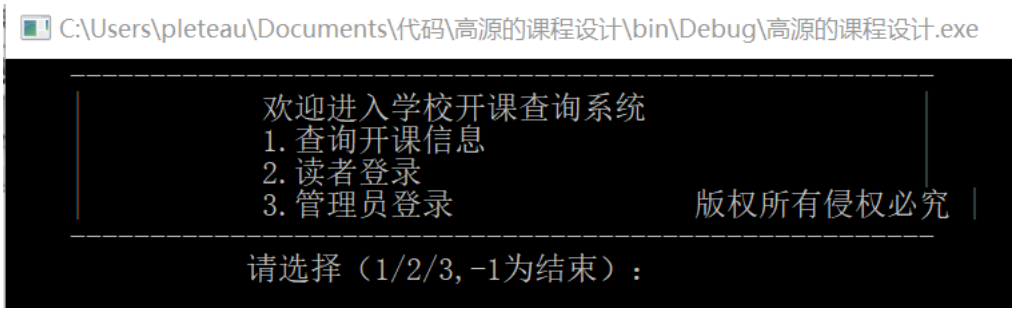
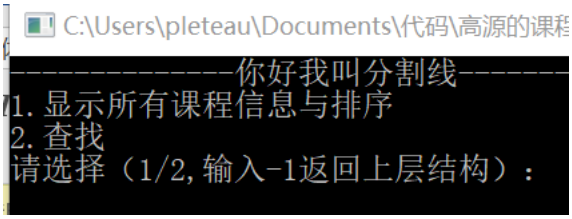


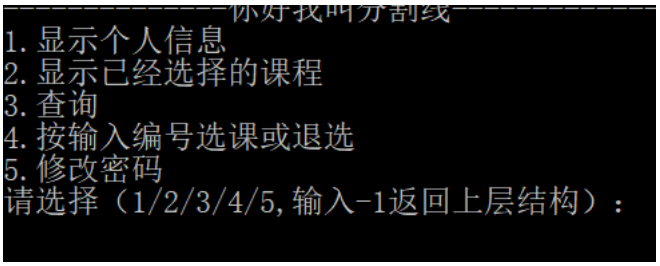
图 5　开始登录界面

主要通过选择结构和循环结构实现界面的前进和后退。例如，第一个登录界面出现 4 个选择：1.查询开课信息，2.读者登录，3.管理员登录-1.结束用 switch case 分别实现，选择之后转到下一个界面。

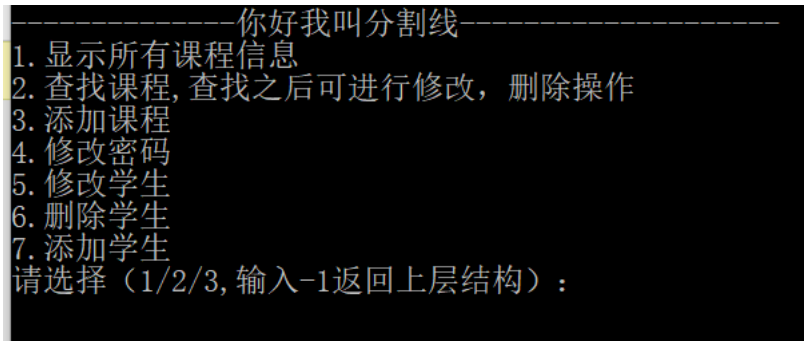
这位查询开课信息界面



这为读者登录成功之后的界面



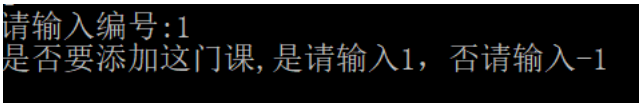
这是管理员登录成功之后的界面



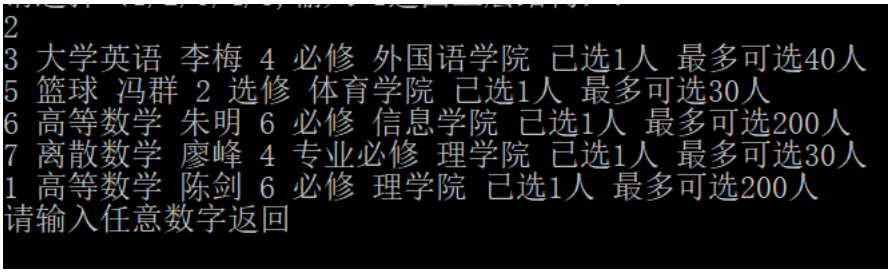
（4）选课的实现

学生数据里有一个整形数组指针 int *subject,在 subject 指向的数组中存放该学生已经选择的课程编号，通过该课程编号来实现展示所有学生已选的课程，学生进行选课操作时，输入课程编号，若该编号不存在在 subject[]中，则向 subject 末尾添加，反之则删除，管理员删除课程时，会从所有已选该课的学生的 subject[]中去掉该编号

如图为选课操作



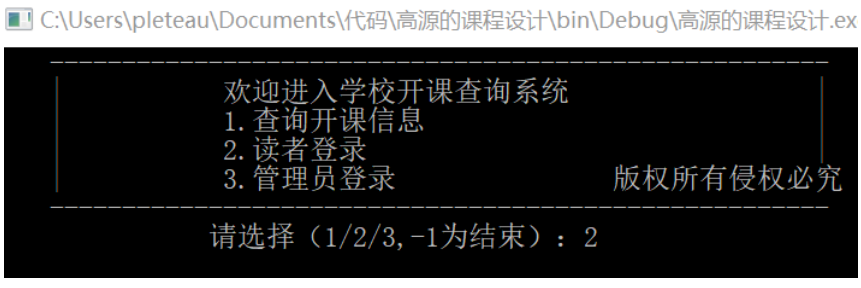
如图为展示学生已选的课



四、 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：

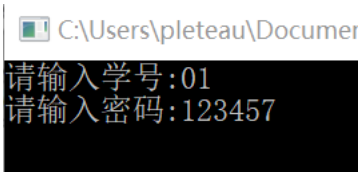
在开始界面输入“2”即可跳转到登陆界面。



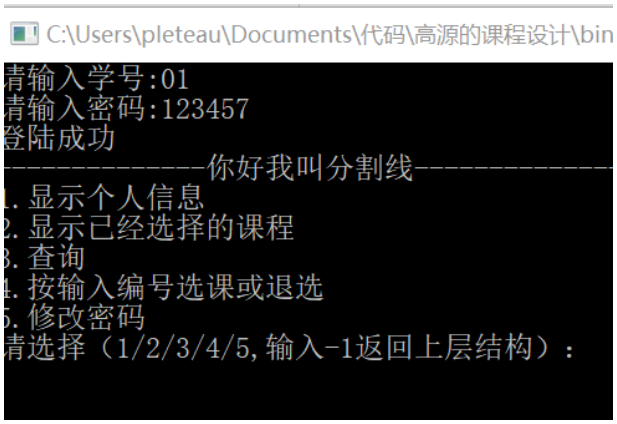
后输入读者信息，如：

读者编号：01

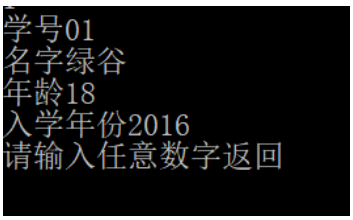
密码：123457



可跳出读者的个人界面

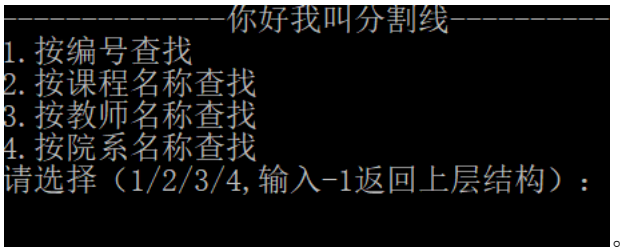


若执行“1”，则显示个人信息。



若执行“2”，则可显示已选择的课程

若执行“3”，则可查询课程信息



。

若执行 4，则可进行退选与选课操作

若执行 5，则可修改密码

```
请输入新的密码:123457
成功
请输入任意数字返回
```

四.问题及解决

遇到的问题及解决方法如下：

- **问题 1：**

问题描述：对课程按院系与学分进行升序排序时

我运用冒泡排序，写了排序算法，编译时没有出错，但程序每次运行到排序时都会卡住无法进行任何操作

解决方法：

我原以为 course 应该有默认的赋值运算符，出错了之后，我尝试了很多种方法

当我重新对赋值运算进行重载之后就没有问题了

以下为该冒泡排序算法

```
void courlist::sortscore()
{
   course temp(80,"test","alisan",7,"选修","虚空",9,1
   course *p1,*p2;
   for(p1=head;p1->next!=NULL;p1=p1->next)
   {
        for(p2=p1->next;p2!=NULL;p2=p2->next)
        {
             if(p1->getscore()>p2->getscore())
             {
                  temp=(*p1);
                  (*p1)=(*p2);
                  (*p2)=temp;
             }
        }
   }
}
```

图 6　调试测试问题 1

…

……

五、　实验总结（优点、不足、收获及体会）

我设计的开课信息查询系统，不仅可以对开课信息进行按名称编号院系等进行查询，也可以对课程进行排序还可以实现选课与退选操作，功能比较完善

存在的缺点主要是对于文件管理方面，我只能实现一学期的选课与查询操作

第二学期时学生无法重新选课，而且没有考虑到一门课由两个老师上的情况

编号方面，也需要考虑到课程编号的意义

通过这次 C++的大型实验，我深刻的明白到：课本知识与实践能力相结合的重要性。会读程序的人并不一定会编程序。要想把一门专业课程学好，必须增强自己的动手实践能力，

而不是一天到晚只知道看书，那种行为只不过是“纸上谈兵”。看再多的书都不如自己亲手试一试。俗话说的好：会打仗的士兵才是好士兵。再者，课本上的知识不一定是完全准确的，只有自己动手进行试验过才知道。实践是检验真理的唯一标准，这话不假。

六、 附录：源代码