**《C++程序设计》课程设计报告**

|  |  |
| --- | --- |
| 专业： | 计算机类 |
| 班级： | 2020级计算机类1班 |
| 学号： | 2020214675 |
| 姓名： | 张志衡 |
| 学期： | 2020-2021-2 |
| 指导教师： | 朱红梅 |

**目录**

[**运行环境：** 3](#_Toc75441691)

[**软件** 3](#_Toc75441692)

[**硬件** 3](#_Toc75441693)

[**设计题目：** 3](#_Toc75441694)

[**题目1图书管理系统** 3](#_Toc75441695)

[**系统的功能结构图（因为是试用期所以有软件XMind的水印）** 3](#_Toc75441696)

[**算法设计思想和流程图** 5](#_Toc75441697)

[**源程序（类的定义，函数原型及说明）** 7](#_Toc75441698)

[**运行结果和分析** 13](#_Toc75441699)

[**特别的函数：** 14](#_Toc75441700)

[**不足之处：** 18](#_Toc75441701)

[**题目2学生选课管理系统** 20](#_Toc75441702)

[**系统的功能结构图（因为是试用期所以有软件XMind的水印）** 20](#_Toc75441703)

[**算法设计思想和流程图** 21](#_Toc75441704)

[**源程序（类的定义，函数原型及说明）** 23](#_Toc75441705)

[**运行结果和分析** 28](#_Toc75441706)

[**题目3医院病床管理系统** 29](#_Toc75441707)

[**系统需求：** 29](#_Toc75441708)

[**主要功能：** 30](#_Toc75441709)

[**三、 收获及体会** 31](#_Toc75441710)

# **运行环境：**

## **软件**

1. 系统：Windows 10 家庭中文版 64-bit Build 19042;
2. 编译环境：CodeBlocks 20.03（GCC版本8.1.0）。

## **硬件**

1.处理器：AMD Ryzen 5 4600H;

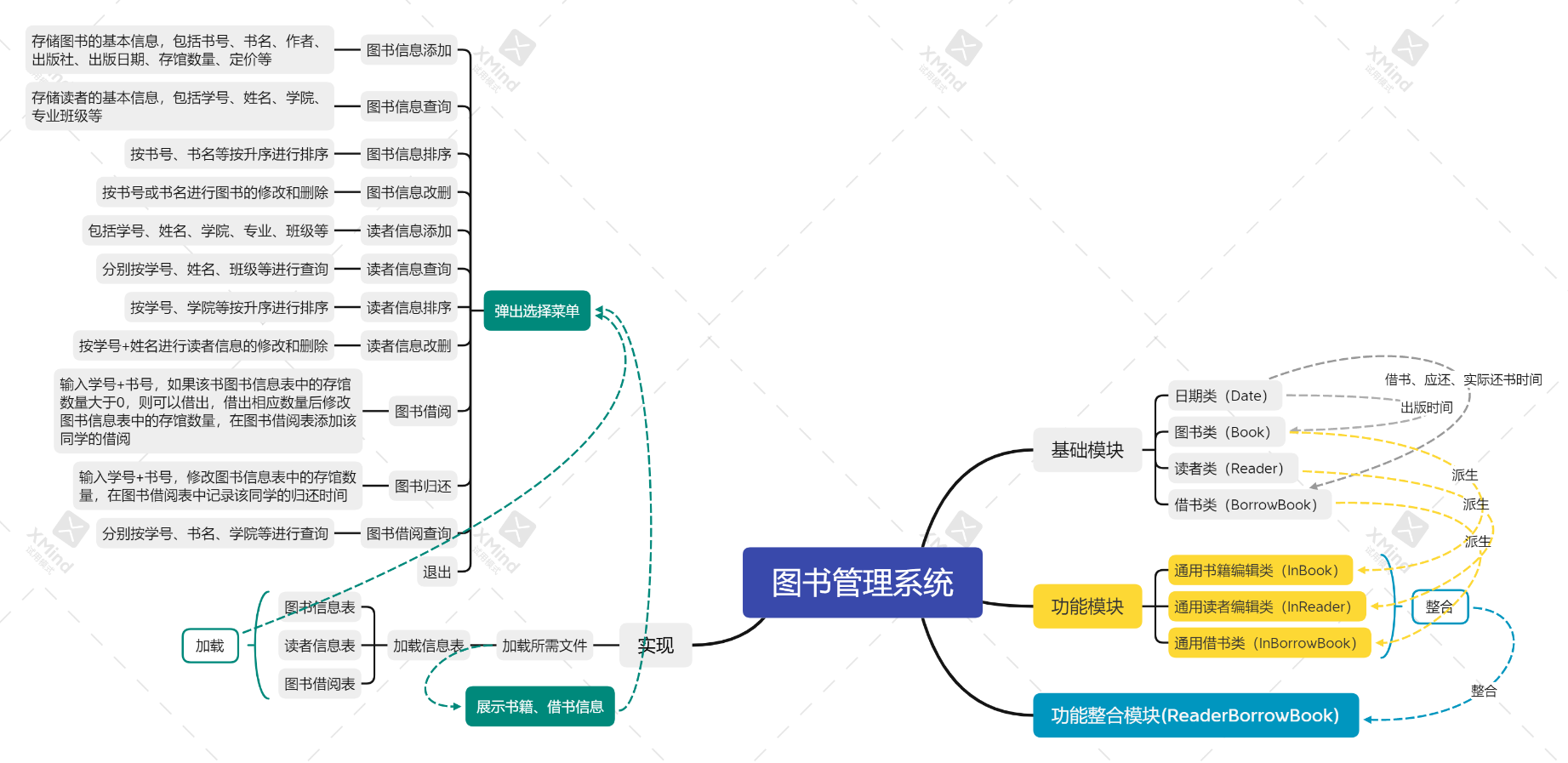
2.机带 RAM：16.0 GB (15.4 GB 可用);

3.系统类型：基于 x64 的处理器。

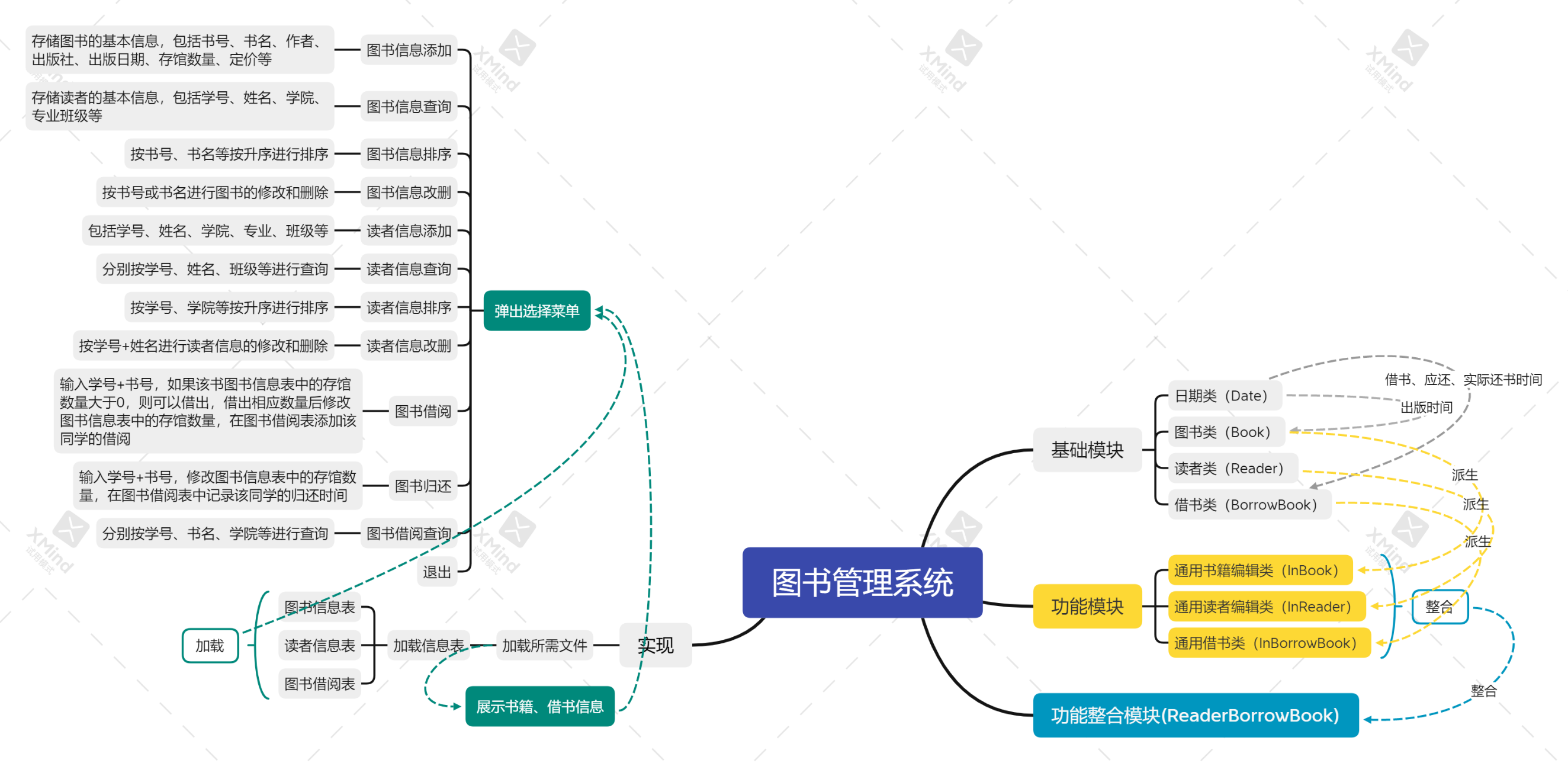
# **设计题目：**

## **题目1图书管理系统**

### **系统的功能结构图（因为是试用期所以有软件XMind的水印）**

**（1）模块部分：**

#### **（2）实现部分**



### **算法设计思想和流程图**

#### **设计思想：**

整体设计思想：先是构建作为基础的Date类、Book类和Reader类，再以这两个类为基础构建实现类。

对于Date类，因为日期是可以进行加减运算的，所以Date类进行了对于“+”的重载。在构造函数中的日期判断和闰年判断使得日期的加减运算变得更安全，在运算符“+”的重载中，有一步二月天数判断，也起到了相关作用。

对于Book类和Reader类，两个类将InBook和ReadBorrowBook声明为友元函数，使得Book类的操作更加方便。

对于BorrowBook类，将InBook声明为友元函数，操作更加方便，也同样体现了逐层递进的效果。

在InBook类中，文件打开时的判断和另起bokkk.txt这个文件增加了安全性，后面的各种查找函数所基于的string类操作（字符串查找函数）大大简化了查找的复杂程度，同时也为后面的增删改函数提供了便捷。其中的日期格式转换更加灵活。

在InReader、InBorrowFile、ReaderBorrowBook类里，通过title来确定count很灵活。

#### **算法流程图**

**不是0**

**是0**

退出

判断序号

DOS窗口显示结果

对文件进行

增/删/改/查

调用InBorrowBook类

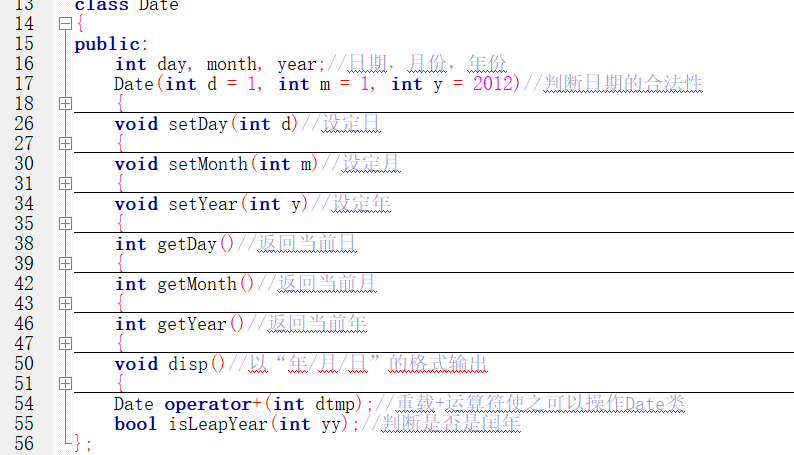
调用InBook类

调用InReader类

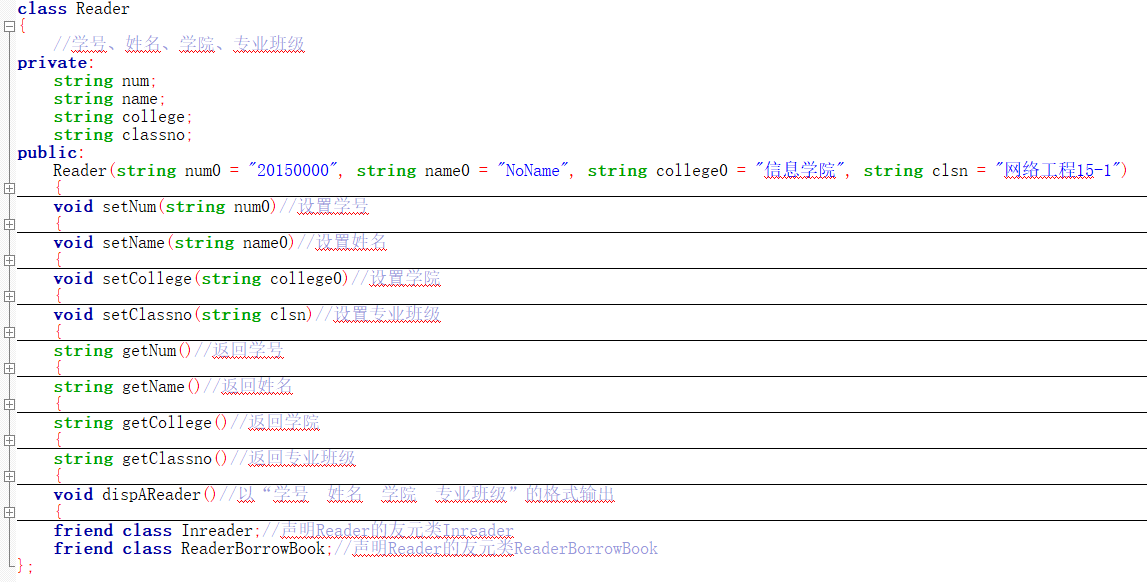
调用ReaderBorrowBook类

输入所选序号

### **源程序（类的定义，函数原型及说明）**



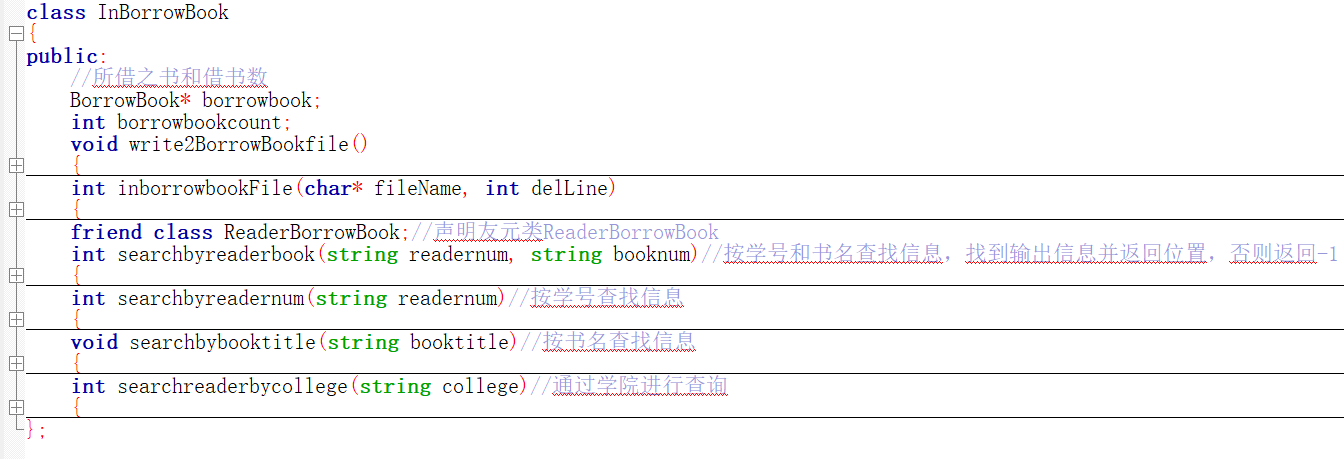
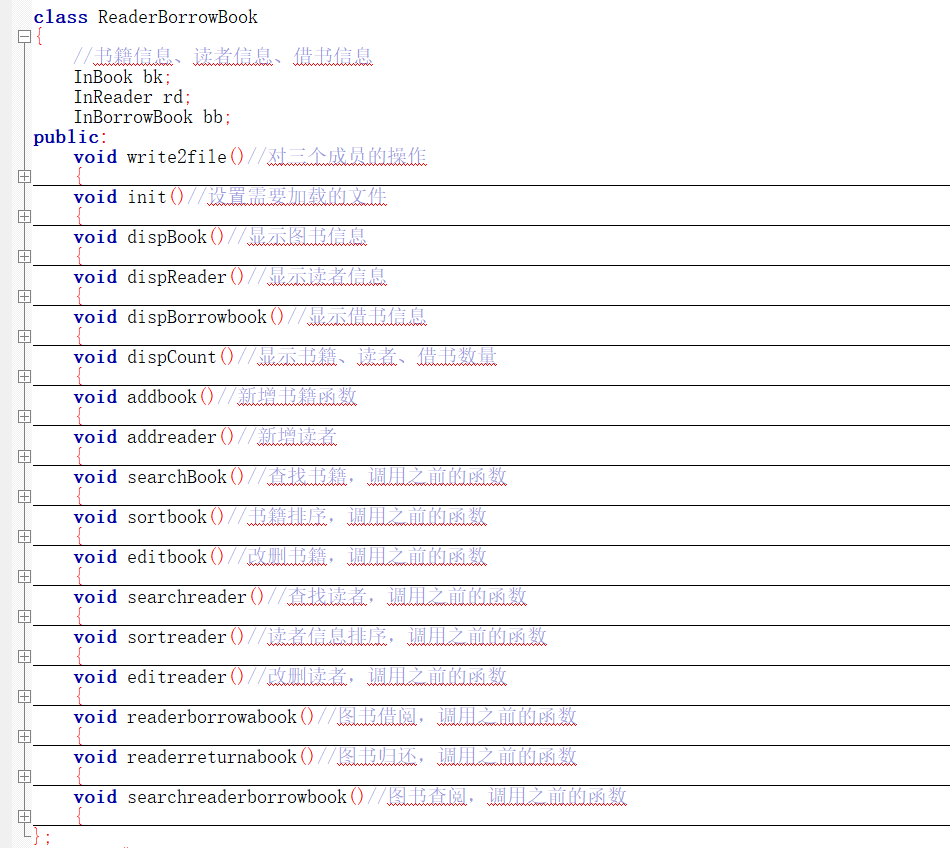










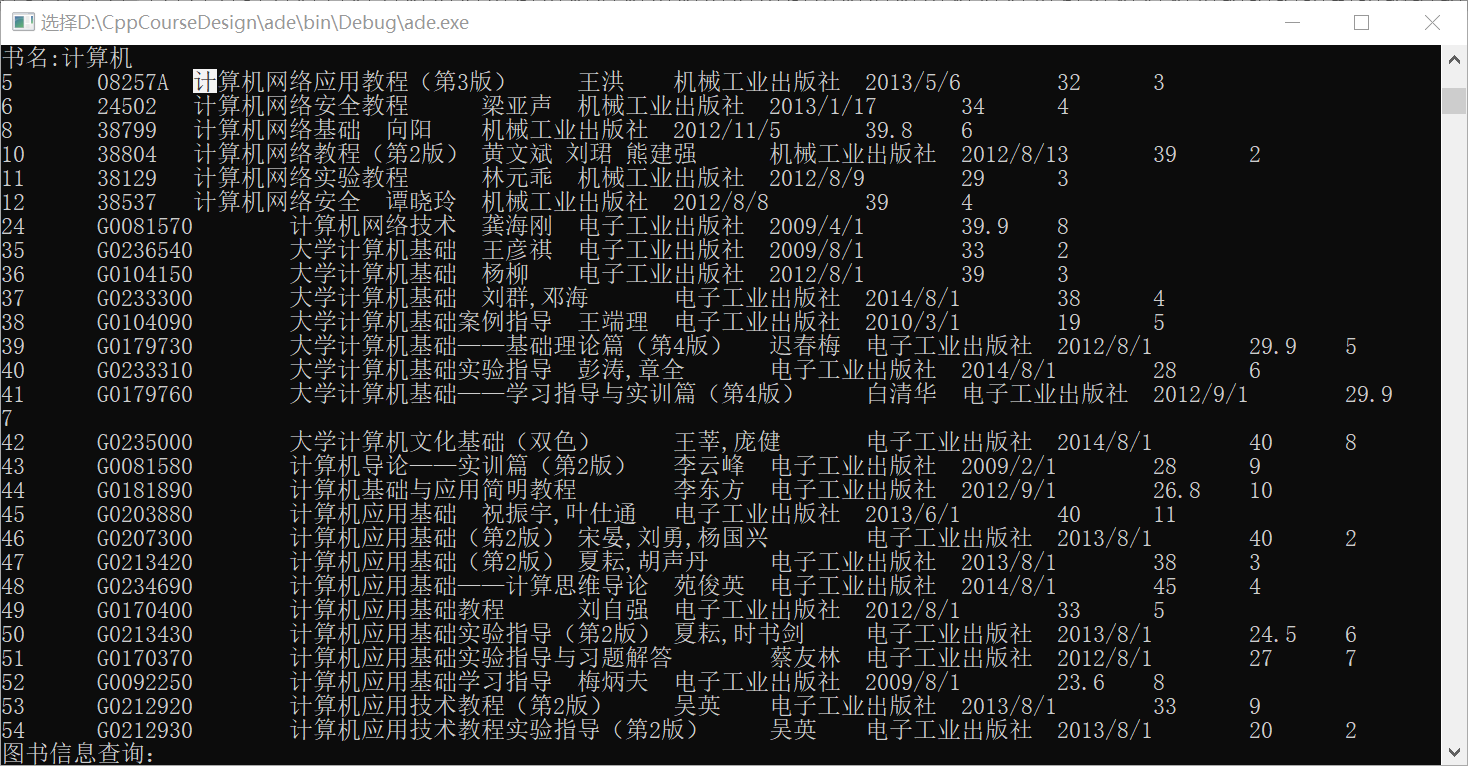
 

### **运行结果和分析**

比如：

1. 2->图书信息查询：分别按书名, 按作者名, 按出版单位等进行查询->按书名查询->输入“计算机”->

输出：

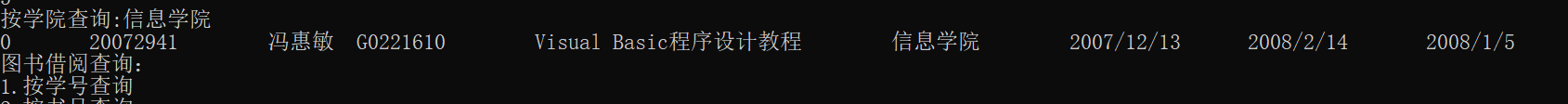


**分析：**调用过程：

ReadBorrowBook::void dispBorrowBook()->

InBook::void searchbookbytitle(string)->

Book:: void dispABook()

(2) 11．图书借阅查询：分别按学号、书名、学院等进行查询->3.按学院查询->输入“信息学院”->输出：

**分析：**调用过程：

ReadBorrowBook::void searchreaderborrowbook()->

BorrowBook::void searchreaderbycollege(string)->

InBorrowBook::int searchreaderbycollege(string)->

BorrowBook::void dispBorrowBook();

### **特别的函数：**

#### **1. 按书名查找书籍**

void searchbookbytitle(string title)//按书名查找书籍，找到返回书的信息，否则提示未找到

{

bool found = false;

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

if ((book[i].getTitle()).find(title) != string::npos)

{

//cout<<book[i].getTitle()<< book[i].getTitle().find(title)<<endl;

cout << i << "\t";

book[i].dispABook();

found = true;

//return i;

}

if (!found)cout << "book title:" << title << " not found." << endl;

//return -1;

}

过程：book[i].getTitle()获取标题，通过查找函数(book[i].getTitle()).find(title) != string::npos判断是不是存在。如果是，则列出该书信息，否则提示未找到，并返回-1。

获取标题

↓

判断书名是不是存在

↓

判断→否→提示未找到

是↓ -

列出信息 -

#### **2. Book类判断日期是否合法的析构函数**

Date(int d = 1, int m = 1, int y = 2012)//判断日期的合法性

{

if (d > 0 && d < 32)day = d;

else day = 1;//限制日在1—31之间，否则设为1

if (m > 0 && m < 13)month = m;

else m = 1;//限制月在1—12之间，否则设为1

if (y > 0)year = y;//限制年大于0，否则设为2012

else year = 2012;

}

过程：判断日月年是否是常规范围，否则改为1。

判断日月年是否是常规范围

↓

判断

否↓

改为1

#### **3. Book类判断是否是闰年的函数**

bool Date::isLeapYear(int yy)//判断是否是闰年

{

if (yy % 400 == 0)

return true;

else if (yy % 100 != 0 && yy % 4 == 0)

return true;

else return false;

}

过程：判断yy如果能被400整除或yy能被4整除但不能被100整除则判断为闰年，返回true，否则返回false。

yy

↓

判断yy如果能被400整除或yy能被4整除但不能被100整除

↓

判断→否→false

是↓ -

true -

#### **4. 通过姓名删除读者**

void deletereaderbyname(string name)//通过姓名删除读者

{

int k = searchreaderbynamep(name);

if (k >= 0)

{

cout << "Reader to be deleted :" << endl;

reader[k].dispAReader();

reader[k] = reader[readercount - 1];

cout << "deletereadername successed:" << endl;

readercount--;

//return readercount;

}

else cout << "deletename failed. No Reader of name=" << name << endl;

}

过程：查找姓名，将要删掉的姓名对应的信息替换为最后一行信息，再将人数-1。

查找姓名

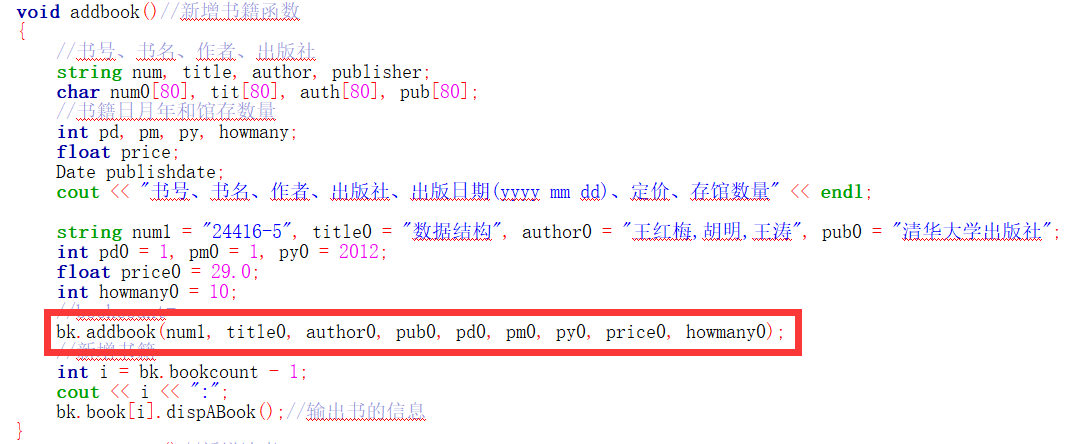
↓

将要删掉的姓名对应的信息替换为最后一行信息

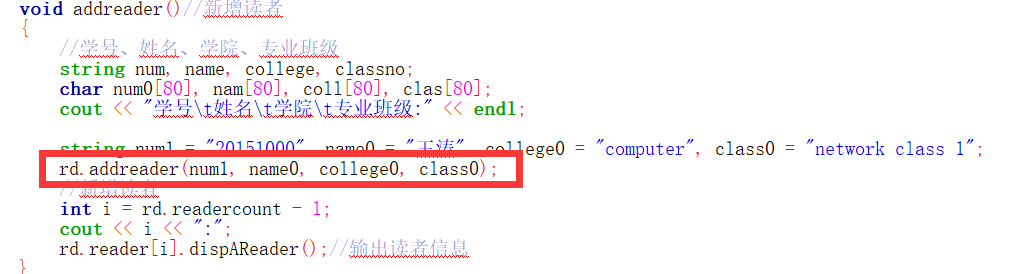
↓

人数改为1

### **不足之处：**

**第一处不足之处：**

ReadBorrowBook::void addbook()函数由于没有输入功能，只能添加同一个书籍信息，ReadBorrowBook::void addreader()亦是如此：



于是，将其改为具有输入功能的：



#### **第二处不足之处：**

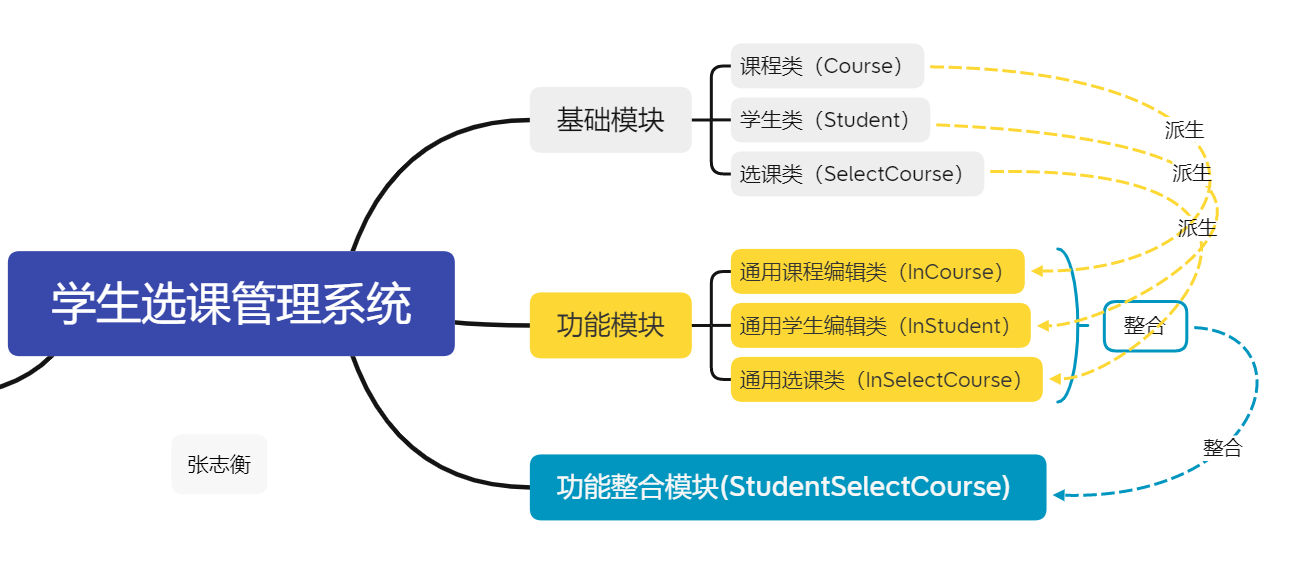


ReadBorrowBook::void readerreturnabook()函数中判断bookpos（书的位置）时忽略了书的位置从0开始计数。导致如果bookpos是0的位置上的书，既不会显示书的信息，又不会提示无书。所以应该为bookpos >= 0以及rdbook >= 0。

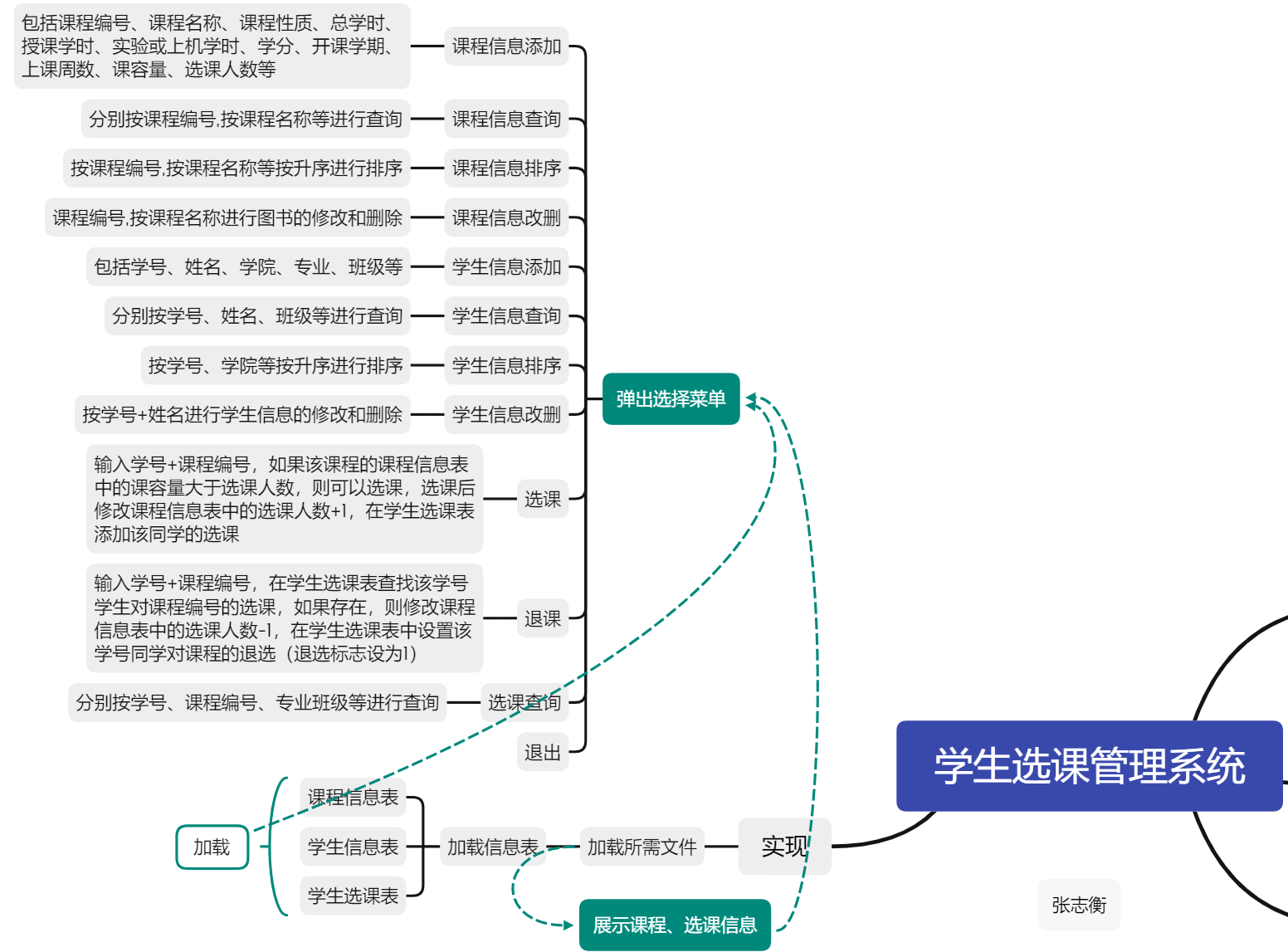
## **题目2学生选课管理系统**

### **系统的功能结构图（因为是试用期所以有软件XMind的水印）**

#### **（1）模块部分：**



#### **（2）实现部分**



### **算法设计思想和流程图**

#### **设计思想：**

整体设计思想：先是构建作为基础的Course类和Student类，再以这两个类为基础构建实现类。对于Date类，因为日期是可以进行加减运算的，所以Date类进行了对于“+”的重载。在构造函数中的日期判断和闰年判断使得日期的加减运算变得更安全，在运算符“+”的重载中，有一步二月天数判断，也起到了相关作用。对于Course类和Student类，两个类将InCourse和ReadSelectCourse声明为友元函数，使得Course类的操作更加方便。对于SelectCourse类，将InCourse声明为友元函数，操作更加方便，也同样体现了逐层递进的效果。

在InCourse类中，文件打开时的判断和另起bokkk.txt这个文件增加了安全性，后面的各种查找函数所基于的string类操作（字符串查找函数）大大简化了查找的复杂程度，同时也为后面的增删改函数提供了便捷。其中的日期格式转换更加灵活。

在InStudent、InSelectFile、StudentSelectCourse类里，通过title来确定count很灵活。

#### **算法流程图:**

**不是0**

**是0**

退出

判断序号

DOS窗口显示结果

对文件进行

增/删/改/查

调用InSelectCourse类

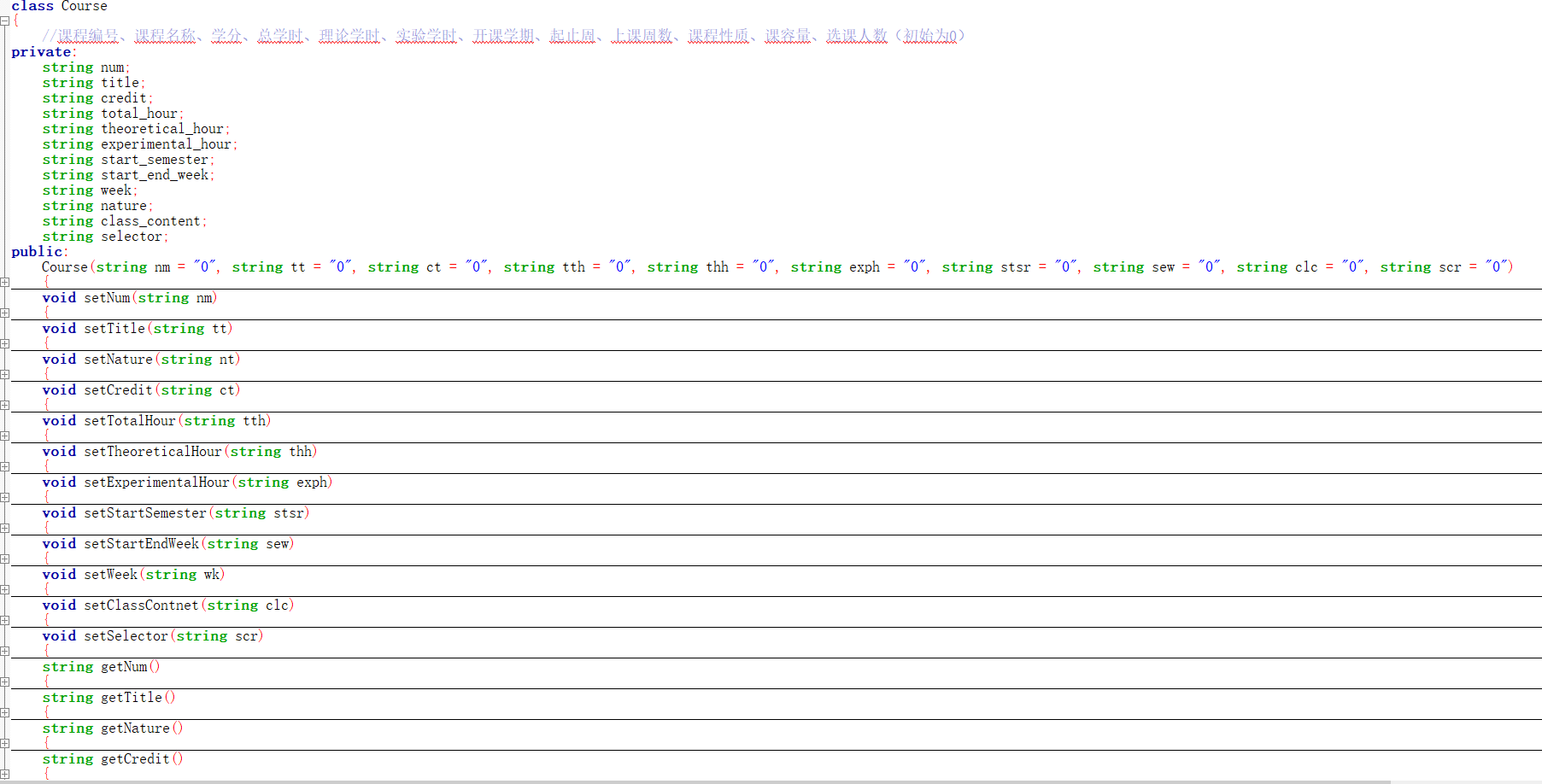
调用InCourse类

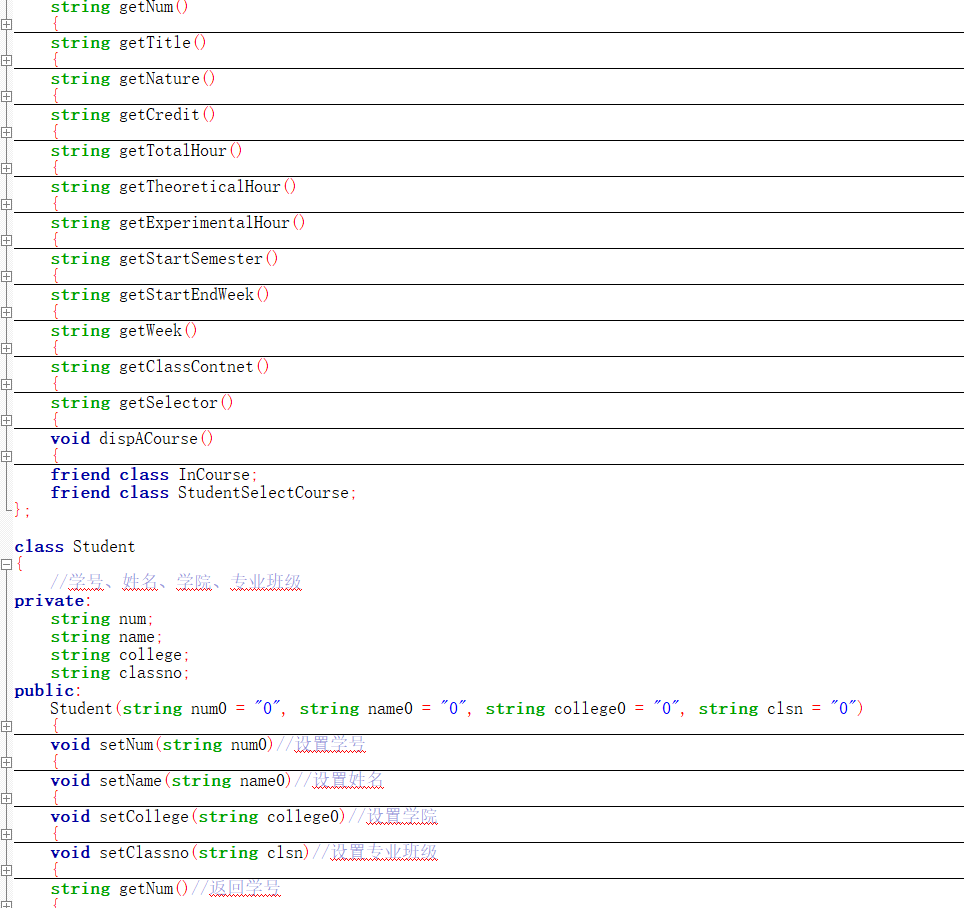
调用InStudent类

调用StudentSelectCourse类

输入所选序号

### **源程序（类的定义，函数原型及说明）**

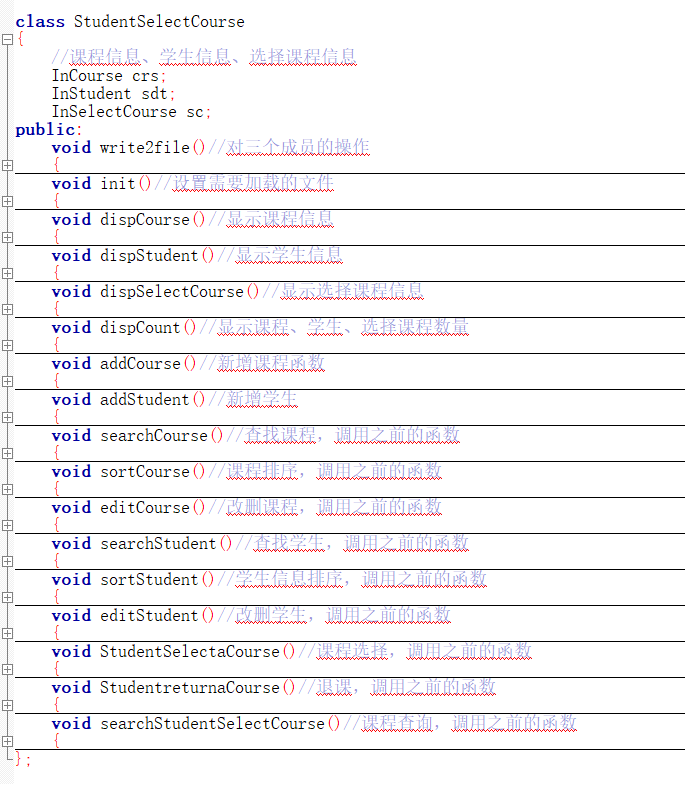
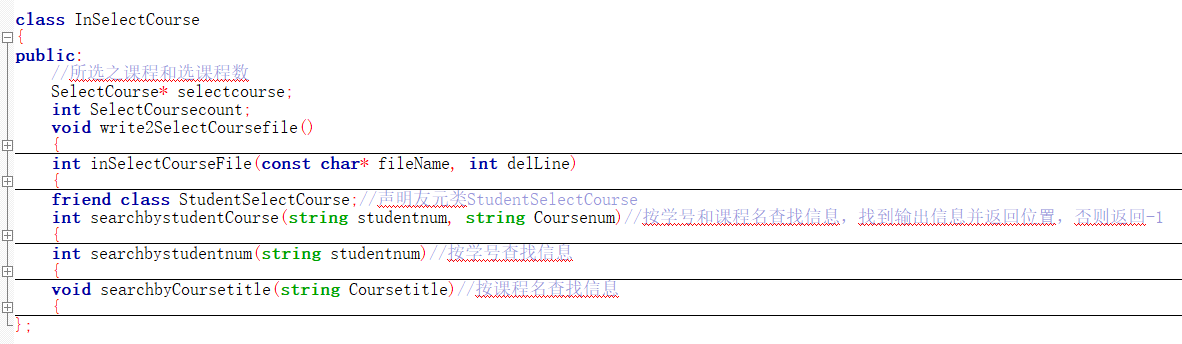








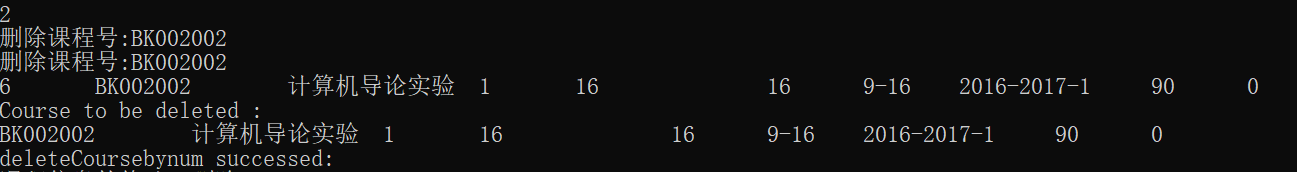




### **运行结果和分析**

比如：

（1）4->课程信息的修改、删除：按课程号或课程名进行课程的修改和删除->2．按课程号删除->输入“BK002002”->输出：



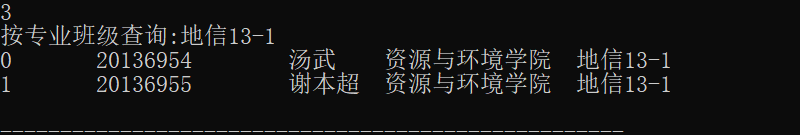
**分析：**

调用过程：

StudentSlectCourse:: void editCourse()->

InCourse:: void deleteCoursebynum(string)->

Course:: void dispACourse()

（2）11-> 11．选课查询：分别按学号、课程名、专业班级等进行查询-> 3.按专业班级查询->地信13-1->输出：

**分析：**

调用过程：

StudentSlectCourse:: void searchStudentSelectCourse()->

InStudent:: void searchStudentbyclassno(string)->

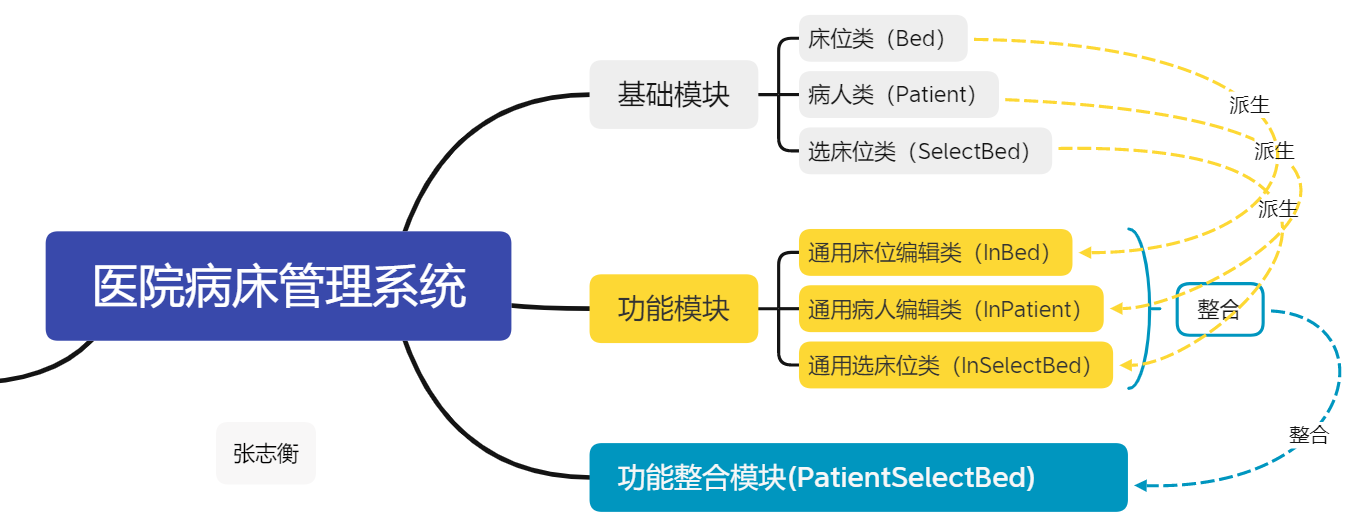
Student:: void dispAStudent()

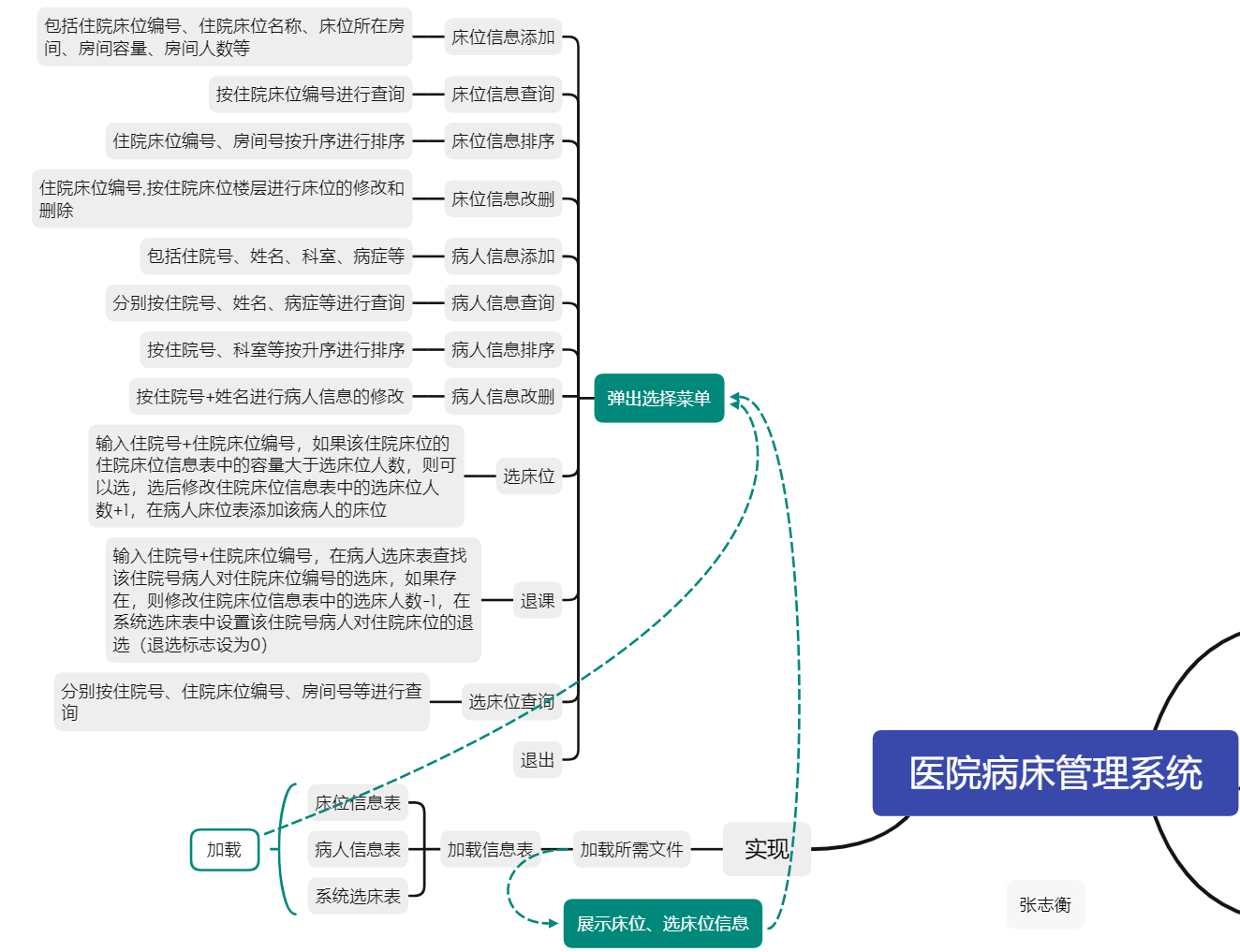
## **题目3医院病床管理系统**

### **系统需求：**

为医院管理人员编写一个医院病床管理系统，医院病床管理系统的设计主要是实现对医院病床的管理和相关操作，包括3个表：2、系统功能结构图：

#### **（1）模块类：**



**（2）功能类**

### **主要功能：**

住院床位信息表——存储住院床位的基本信息，包括住院床位所在房间号、住院床位编号等。

病人信息表——存储病人的基本信息，包括住院号、姓名、科室、病症等。

系统选床表——存储病人选课的相关信息，包括住院号、姓名、住院床位编号、住院房间编号、住院时间、出院标志（0-退住，1-在住）等。

用菜单选择方式完成下列功能：

1．住院床位信息添加功能：包括住院床位编号、住院床位名称、床位所在房间、房间容量、房间人数等。

2．住院床位信息查询：按住院床位编号进行查询。

3．住院床位信息排序：住院床位编号、房间号按升序进行排序。

4．住院床位信息的修改、删除：住院床位编号,按住院床位楼层进行床位的修改和删除。

5．病人信息添加功能：包括住院号、姓名、科室、病症等。

6．病人信息查询：分别按住院号、姓名、病症等进行查询。

7．病人信息排序：按住院号、科室等按升序进行排序。

8．病人信息的修改、删除：按住院号+姓名进行病人信息的修改和删除。

9．系统选床位：输入住院号+住院床位编号，如果该住院床位的住院床位信息表中的容量大于选床位人数，则可以选，选后修改住院床位信息表中的选床位人数+1，在病人床位表添加该病人的床位。

10．病人出院：输入住院号+住院床位编号，在病人选床表查找该住院号病人对住院床位编号的选床，如果存在，则修改住院床位信息表中的选床人数-1，在系统选床表中设置该住院号病人对住院床位的退选（退选标志设为0）。

11．病人床位查询：分别按住院号、住院床位编号、房间号等进行查询。

# **三、 收获及体会**

通过自己修改、制作系统，不仅仅复习了以往的知识，还掌握了新的知识，比如在学生选课管理系统中，string类与int类的互换、MessageBox类的运用。总之，这次课程设计我受益匪浅。