

# 学生成绩管理系统

班级：无03

学号：2020010708

姓名：王与进

2021 年 6 月 27 日

目录	2
----	---

## 目录

<b>1 系统需求分析</b>	<b>3</b>
<b>2 总体设计</b>	<b>3</b>
2.1 学生模式 . . . . .	3
2.2 教师模式 . . . . .	4
2.3 教务处模式 . . . . .	5
<b>3 详细设计</b>	<b>6</b>
3.1 类模块图 . . . . .	6
3.2 UML图 . . . . .	6
3.3 程序运行逻辑 . . . . .	8
3.4 面向对象特点 . . . . .	8
<b>4 系统调试</b>	<b>8</b>
4.1 程序架构设置 . . . . .	9
4.2 多文件编程 . . . . .	9
4.3 面向对象编程 . . . . .	9
4.4 STL . . . . .	10
4.5 输入输出流 . . . . .	10
4.6 低级错误 . . . . .	10
<b>5 测试结果与分析</b>	<b>10</b>
5.1 主界面 . . . . .	10
5.2 学生模式 . . . . .	11
5.3 教师模式 . . . . .	13
5.4 学生模式与教师模式的交互 . . . . .	15
5.5 教务处模式 . . . . .	18
<b>6 总结</b>	<b>20</b>
<b>7 源代码</b>	<b>20</b>

## 1 系统需求分析

1.该学生成绩管理系统模拟清华大学学生教师信息管理系统和学生成绩管理系统，可以实现学生、教师信息的录入、查询、删除、排序等操作，以及模拟选课、退课、给分、查询成绩等操作。

2.该成绩管理系统应有以下几种访问模式：学生、教师、教务处，以实现不同群体的需求。进入程序后首先会模拟登录过程，选择对应的身份，并输入ID进行登录。系统应以菜单方式工作，并在操作成功或失败时有信息提示。

3.系统应基于ID对信息进行管理。输入ID，输出不同模式下对应的信息。

4.系统应有完整的文件管理体系，将输入的信息自动写入文件；在查询信息时，从文件读取并在终端输出。

## 2 总体设计

### 2.1 学生模式

1.(A)查看个人信息：输出ID、姓名、性别、年级、班级、专业。

2.(B)录入个人信息：输入个人信息，包括年级、班级、系别、性别等（ID和姓名已经事先决定或绑定，不能在此选项中修改）。适用于初次登录。

3.(C)修改个人信息：输入相应字母，修改自身的对应选项。适用于登录后的调整。

4.(D)查看所修课程信息：查看目前所有所修课程的名称、绩点以及均绩。

5.(E)查看所有课程信息：查看目前所有开设课程的信息。

6.(F)选课：输入课程的ID，可以返回课程的开课信息（课程名、开课教师、评分标准等），学生可以根据课程信息选择所需要的课程。

7.(G)退课：输入课程的ID，可以退掉这门课程。

8.(X)实名认证：若此账号初次登录，则会被要求进行实名认证，绑定用户真实姓名。姓名一经绑定将无法修改。

9.(Y)退出登录：退出当前账号，保存当前录入的信息，返回登录界

面。

10.(Z)退出程序：退出当前程序。

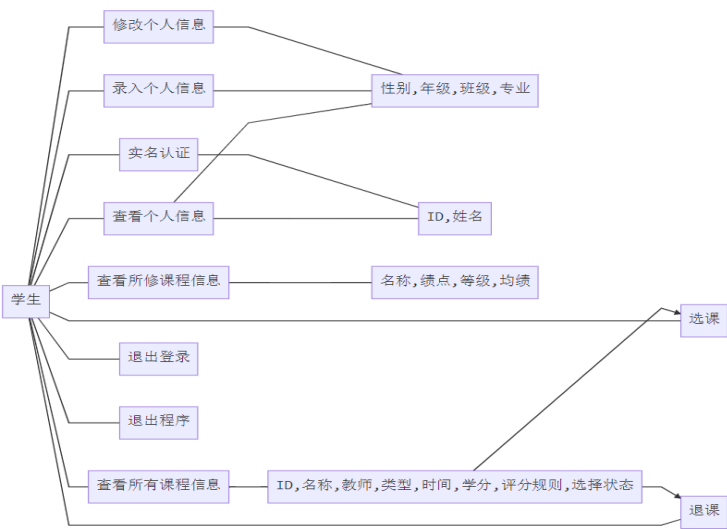


图 1: 学生模块示意图

2.2 教师模式

- 1.(A)查看个人信息：ID、姓名、性别、职称、专业。
- 2.(B)录入个人信息：同学生模式。
- 3.(C)修改个人信息：同学生模式。
- 4.(D)查看所教课程信息：查看课程的ID、任课教师、名称、类型、时间、学分、评分规则。
- 5.(E)查看所有课程信息：同学生模式。
- 6.(F)录入所教课程信息：录入所教课程名称、类型、时间、评分规则等。
- 7.(G)查看学生绩点信息：查看所有报课学生的绩点，可选择是否按照绩点高低排序。
- 8.(H)录入成绩：为所有未获得成绩的报课学生录入各项成绩。
- 9.(X)实名认证：同学生模式。
- 10.(Y)退出登录：同学生模式。

11.(Z)退出程序：同学生模式。

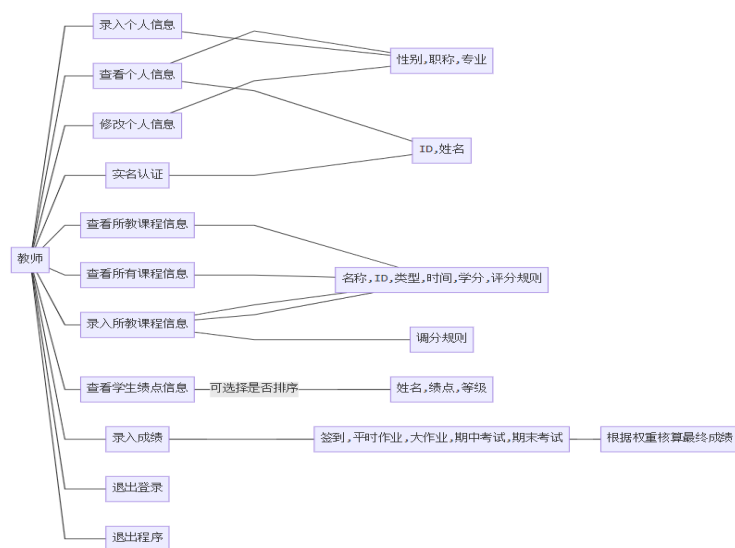


图 2: 教师模块示意图

## 2.3 教务处模式

- 1.(A)查找学生信息：输入学生ID，返回这个学生的所有信息。
- 2.(B)查找教师信息：输入教师ID，返回这个教师的所有信息。
- 3.(C)删除选定信息：输入ID，删除某一学生或某一教师的全部信息。
- 4.(D)删除全部信息：清空数据库。
- 5.(Y)退出登录：同学生模式。
- 6.(Z)退出程序：同学生模式。

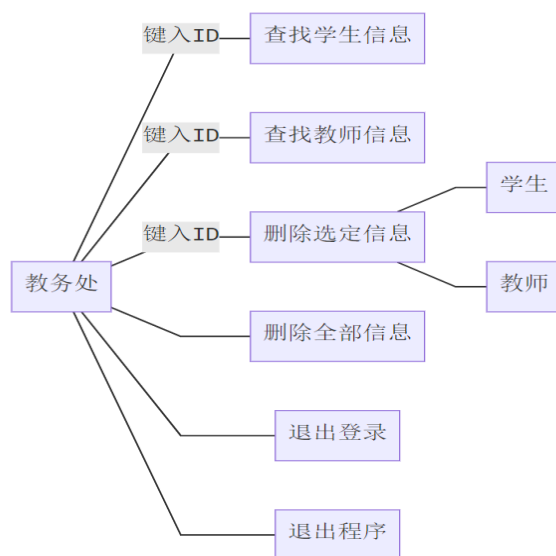


图 3: 教务处模块示意图

### 3 详细设计

#### 3.1 类模块图

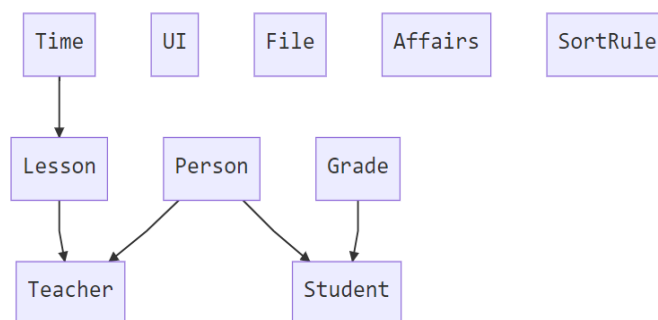


图 4: 类模块图

#### 3.2 UML图

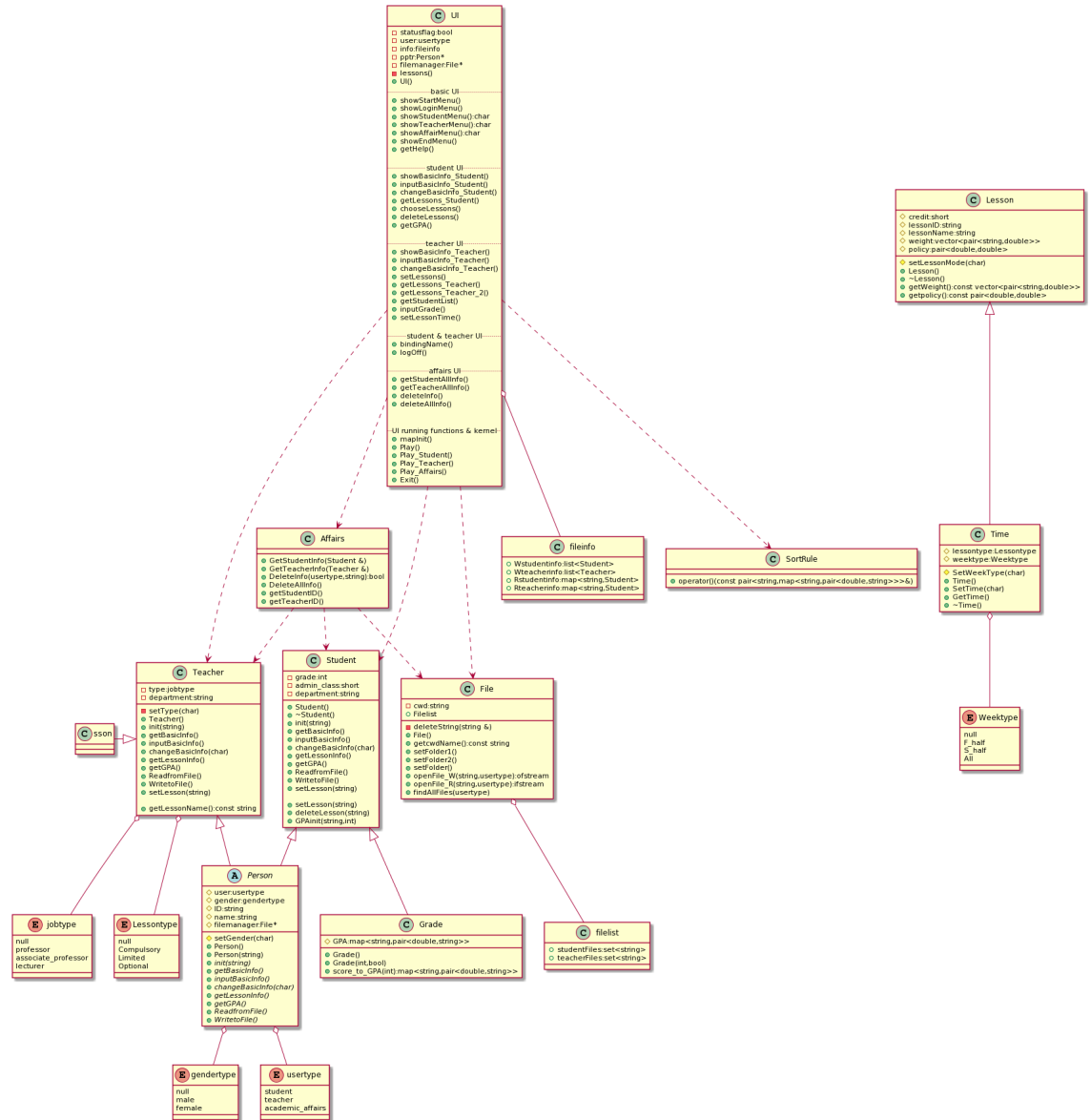


图 5: UML图(完整版详见文件夹中UML.png)

### 3.3 程序运行逻辑

1.系统设置5个全局对象：Student student, Teacher teacher, Affairs affair, File f, UI ui。

2.系统首先会调用全局对象File f的构造函数，读取当前工程所在的目录，创建STUDENT和TEACHER两个文件夹（如果没有的话），完成数据库的初始化。随后，系统会调用UI的构造函数，将每一个文件的信息都写入info中的Rstudentinfo和Rteacherinfo两个map中，便以后查询用。

3.用户在输入对应的身份后，UI会进行不同的初始化操作。若用户选择学生或教师模式，UI中的基类指针pptr会指向对应的全局对象，以便实现多态的功能处理；若用户选择管理员模式，pptr将会被置为nullptr，直接操作affair对象。

4.关于用户信息的初始化：若输入的ID存在于数据库文件中，那么初始化时File f将自动检索ID，打开对应的文件，将文件中的内容写入全局对象，否则将创建一个全新的文件，并提示用户进行信息的录入。

5.每次在执行相应操作时，系统都会返回操作成功或失败的信息；对于选课、退课的操作，系统还可以询问用户是否继续执行该操作；每次操作完毕后，系统将会自动跳转到上一级目录中。

6.当学生或教师的信息发生改动后，该对象将会插入info中的WStudentinfo 或Wteacherinfo两个list中，当退出登录或退出程序后，两个list中的信息将会被依次写入到对应的文档中，下次使用时即可直接读取。

### 3.4 面向对象特点

1.使用Person作为抽象类，继承得到Student类，Teacher类，从而能在UI类中直接使用基类指针pptr对对象的方法进行直接访问。

2.使用多继承，Teacher类继承Lesson类，Lesson类又继承Time类。

3.使用运算符重载，在SortRule类中重载()，从而实现了对GPA的排序。

## 4 系统调试

代码在调试过程中出现了一些问题，我大致将其分为以下几类：



## 4.1 程序架构设置

我的程序架构进行了多次微调。

1.初始时我将Person类派生为3个类：Student, Teacher, Affairs.但后来发现，Affairs类的操作方法与前两个类有很大的差别，并不适用于派生，所以我将其设置为单独的类。

2.一开始我的所有枚举变量是分散在许多头文件里的，但这样在文件之间相互引用时会带来很大的麻烦，于是我就将所有的枚举变量放在enums.h中，给引用带来了较大的方便。

## 4.2 多文件编程

这是我第一次利用多文件编程来解决一个大项目，遇到了一些代表性问题。

1.我一开始将全局对象的定义放在头文件中，但编译时会显示以下信息：“class XX 已经在 YY.obj中定义”。后来我上网搜索资料找到了原因：全局对象的定义尽量不要放在头文件中，这样可能会在多次include后发生重定义。于是我将其放在.cpp文件中，问题解决。

2.后来我的全局对象分散在多个.cpp文件中，但程序在编译运行后会崩溃。经调用堆栈分析，发现是全球对象没有正确初始化所致——分散在多个.cpp文件中导致对象的构造函数不能被正确调用。经过思考后我找到了一种解决方案——将所有的全局对象放在main.cpp中，在需要用到这些全局对象的.cpp文件中，使用extern关键字进行声明（告诉编译器在外部文件中找这些对象的定义）。编译测试后问题解决。

## 4.3 面向对象编程

在这一方面我遇到的主要问题是临时对象、无名对象的问题。在编程时，我收到了这样一条警告：

```
for (it = info.Rstudentinfo.begin(); it != info.Rstudentinfo.end(); it++)
{
    string _lessonname = teacher.getLessonName();
    auto GPAit = it->second.getGPAlist().find(_lessonname); //寻找学生是否报了自己课
    if (GPAit != it->second.getGPAlist().end()) //学生报课
    {
        cout << it->second.getName() << setw(6) << GPAit->second.first << setw(6) << GPAit->second.second << endl;
    }
}
```

图 6: 警告

运行后，程序崩溃：iterators incompatible。我推测是这样的：调用函数所返回的是一个无名的临时对象，若不能够及时赋值就会立刻被销毁，从而无法调用其方法。我将无名对象进行赋值操作后，再调用对应的方法后，程序正常运行，问题解决。

#### 4.4 STL

在此之前我只比较熟悉string和vector的用法，但随后的程序调试中，我结合程序功能对程序中所用的STL容器做了一些修改。例如，读入文件的信息用map储存，便于检索；写入文件的信息用list储存，便于插入；获取目录下的文件名用set储存，保证不会出现重复。这样可以提高程序的运行效率，使程序可读性增强。

#### 4.5 输入输出流

我在程序中设置了大量的输入检测。一开始使用cin.clear();cin.sync();进行输出流的重置，但这样会出现一个bug：当要求输入一个int型整数却输入一个字母时，程序会死循环，一直显示“输入错误”的信息。我在上网查阅资料后，将重置方法调整为cin.clear();cin.ignore();后问题解决。

#### 4.6 低级错误

此外还有一些低级错误，例如在switch-case语句中忘记加break，导致学生教师信息不能被正常设置等。

## 5 测试结果与分析

### 5.1 主界面

主界面如下：

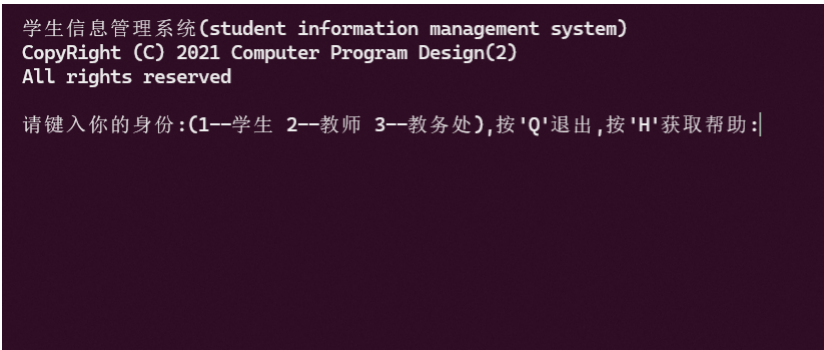


图 7: 主界面

获取帮助:

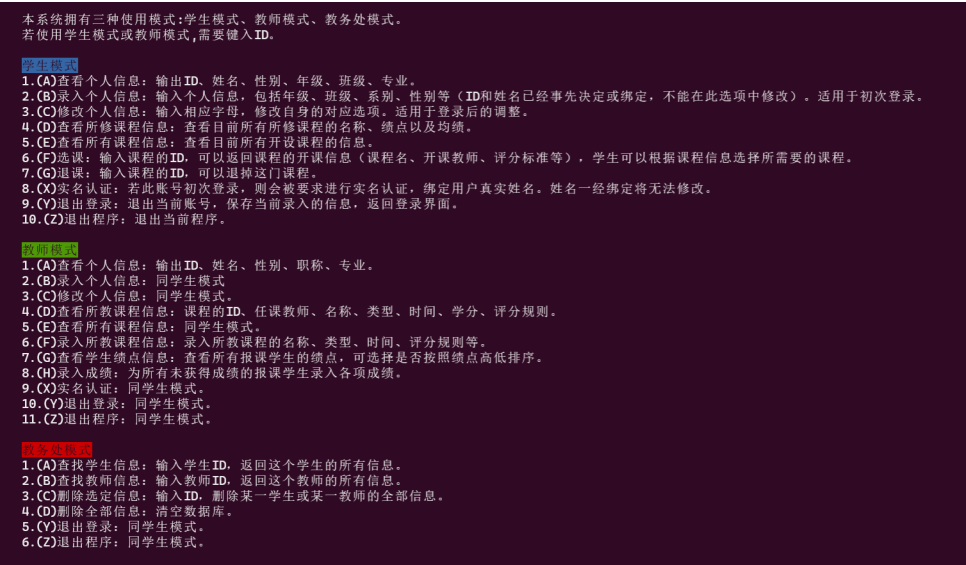


图 8: 帮助

5.2 学生模式

输入一个不存在的学生ID, 进行登录:



图 9: 登录界面

登录后的界面:

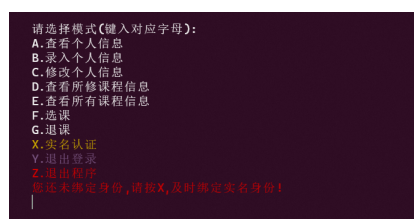


图 10: 学生界面

此时需要进行实名认证:

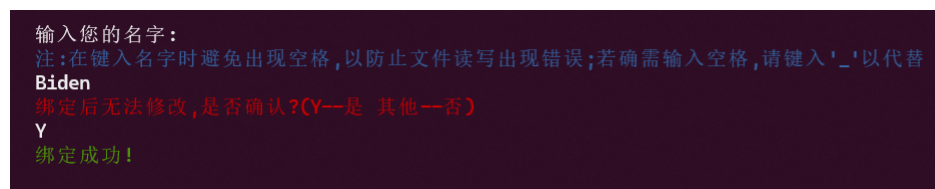


图 11: 实名认证

实名认证成功后, 需要录入个人信息:

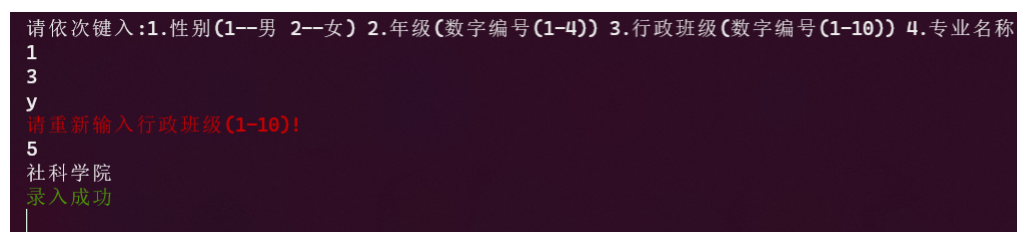


图 12: 输入个人信息

查看个人信息:

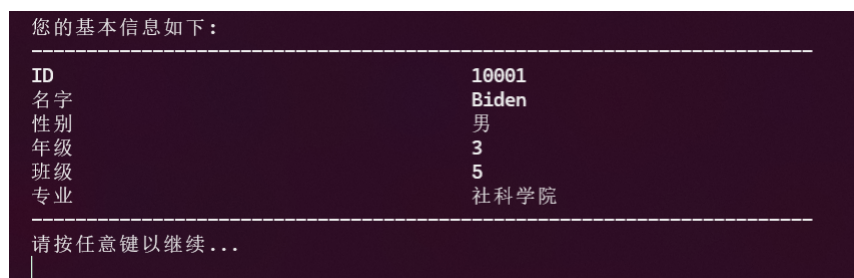


图 13: 查看个人信息

后续也可以对信息进行修改:

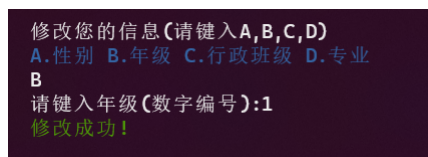


图 14: 修改个人信息

此时若想切换登录身份, 可以选择退出登录, 录入的信息将会自动存入文件中。

### 5.3 教师模式

教师界面, 同样需要先实名认证:

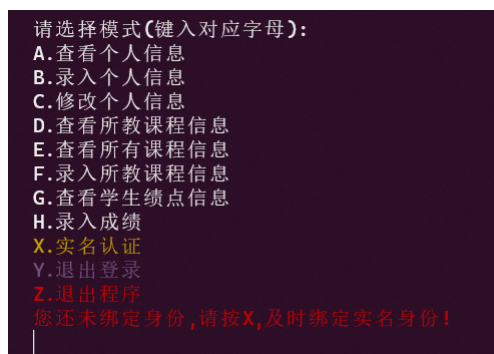


图 15: 教师界面

录入自己的基本信息:

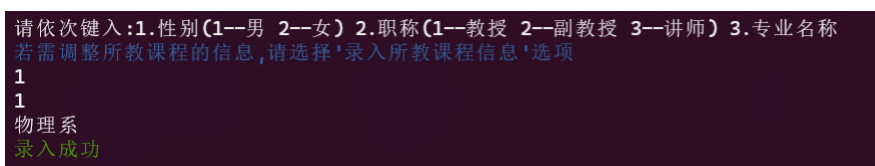


图 16: 输入个人信息

录入自己的授课信息:

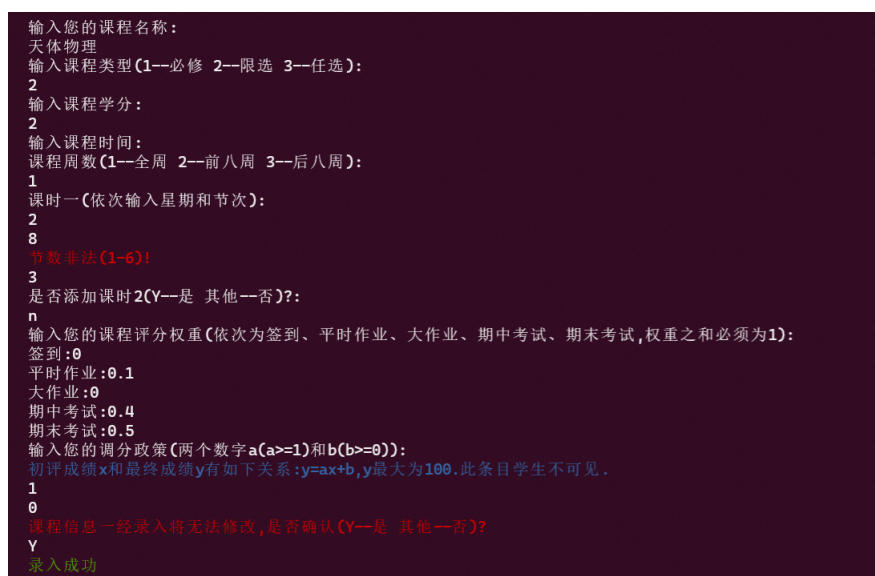


图 17: 输入课程信息

查看信息:



图 18: 查看个人信息

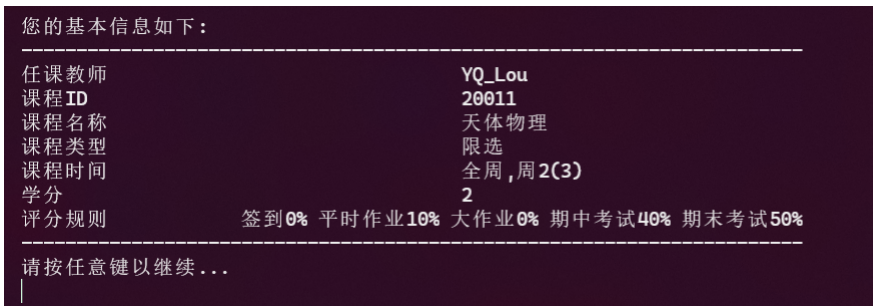


图 19: 查看课程信息

同理，后续也可以对自己的信息进行修改。

5.4 学生模式与教师模式的交互

在学生模式下进入选课，系统会显示所有课程的信息并提示学生选课，此时学生可以键入ID选择自己所需要的课程：

任课教师

课程ID

课程名称

课程类型

课程时间

学分

评分规则

(已选)

YQ\_Lou

20011

天体物理

限选

全周, 周 2(3)

2

签到0% 平时作业10% 大作业0% 期中考试40% 期末考试50%

任课教师

课程ID

课程名称

课程类型

课程时间

学分

评分规则

(未选)

Gauss

20012

微积分A

必修

后八周, 周 3(2)周 5(1)

5

签到0% 平时作业20% 大作业0% 期中考试20% 期末考试60%

键入课程ID,以实现选课操作:

20012

任课教师

课程ID

课程名称

课程类型

课程时间

学分

评分规则

确认选择此课程(Y--是 其他--否)?

Y

选课成功!

是否继续选课(Y--是 其他--否)

Gauss

20012

微积分A

必修

后八周, 周 3(2)周 5(1)

5

签到0% 平时作业20% 大作业0% 期中考试20% 期末考试60%

图 20: 选课

学生的成绩单中会显示下列信息，表示教师还没有进行打分：

您所修课程的GPA如下:

课程名称

量子信息学

天体物理

微积分A

绩点

3.3

等级

B

均绩:3.3

请按任意键以继续...

图 21: 成绩单1

此时可以退出程序，再进入教师界面，为所有报课的学生录入成绩：



```
注:每一项的满分均为100分
请为Biden录入成绩:
签到:0
平时作业:100
大作业:0
期中考试:100
期末考试:100

请为Joe_Biden录入成绩:
签到:0
平时作业:90
大作业:0
期中考试:70
期末考试:80

录入成功
|
```

图 22: 录入成绩

可以查看所有报课学生的成绩，并决定是否开启按GPA高低排序：

```
是否开启按GPA高低排序(Y--是 其他--否)?
Y
所有选课学生的GPA如下:

-----
学生          绩点      等级
-----
Biden          4          A+
特朗普        3.6         B+
Joe_Biden      3.6         B+
-----

请按任意键以继续...
```

图 23: 查看GPA

此时再返回学生界面，可以看到这门课的成绩：

```
您所修课程的GPA如下:

-----
课程名称          绩点  等级
量子信息学        **    **
天体物理          3.3   B
微积分A           4     A+

均绩:3.65
-----

请按任意键以继续...
```

图 24: 成绩单2

若对成绩不满意，可以选择退课：

-----

任课教师

课程ID

课程名称

课程类型

课程时间

学分

评分规则

(未选)

YQ\_Lou

20011

天体物理

限选

前八周,周2(3)

2

签到0% 平时作业10% 大作业0% 期中考试40% 期末考试50%

-----

任课教师

课程ID

课程名称

课程类型

课程时间

学分

评分规则

(已选)

Gauss

20012

微积分A

必修

前八周,周3(2)周5(1)

5

签到0% 平时作业20% 大作业0% 期中考试20% 期末考试60%

-----

输入您想要退掉的课程的ID:

20012

退课后,您将无法获得该课程的学分,确认继续吗(Y--是 其他--否):

Y

退课成功!

图 25: 退课

5.5 教务处模式

教务处界面：

请选择模式(键入对应字母):

A.查找学生信息

B.查找教师信息

C.删除选定信息

D.删除全部信息

Y.退出登录

Z.退出程序

|

图 26: 教务处界面

查看学生的信息：

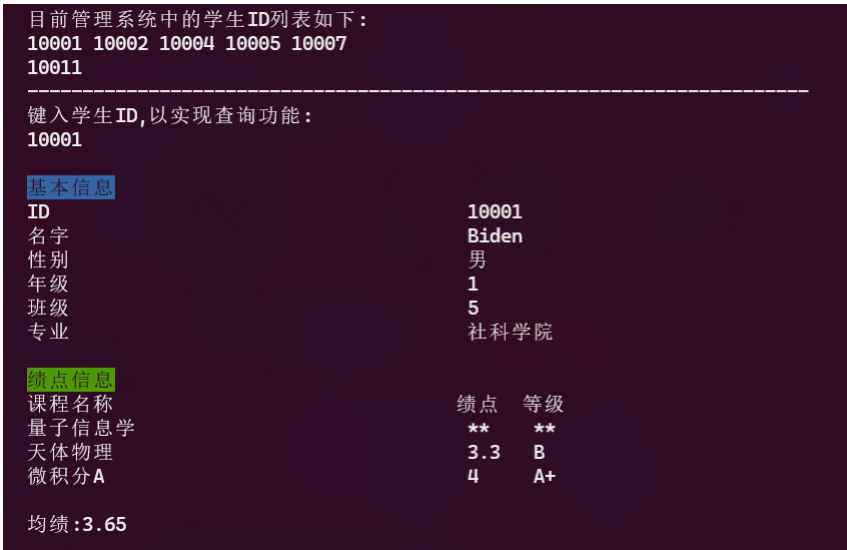


图 27: 学生信息

查看教师的信息:

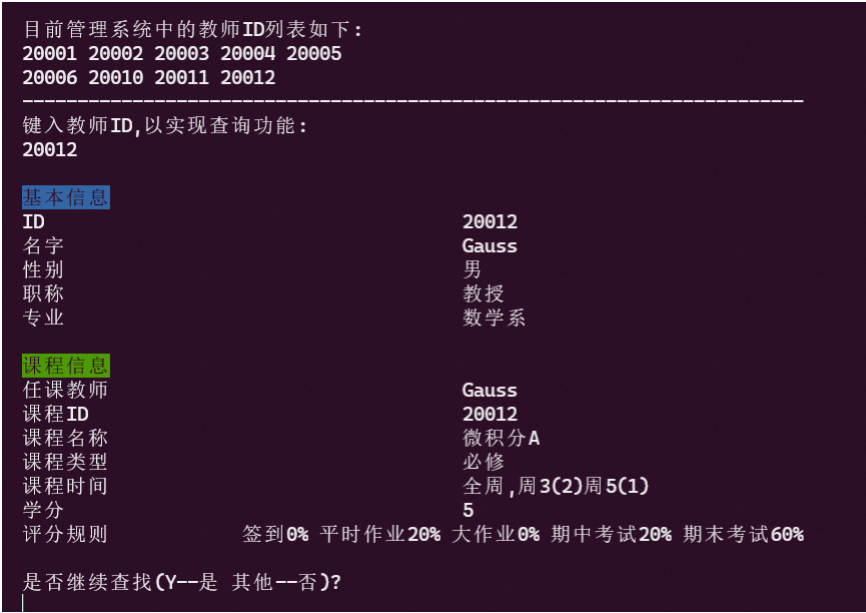


图 28: 教师信息

删除特定信息，删除后文件夹中特定ID的文件会被删除；删除全部信

息，则两个文件夹会被清空。

## 6 总结

这次编程我收益颇多。首先，这是我第一次写一个面向对象的大程序，我对面向对象的一些设计原理，比如类之间的继承包含关系、类的接口与封装、架构设计等有了一些初步的体会。其次，我对C++的一些语法与编程逻辑也多了一些了解，比如多文件编程的方法、STL的使用等。我也认识到编程不仅仅是一门技术，更像是一门艺术，比如如何设计UI，使其界面最适合用户使用，给用户美的享受，这也加强了我进一步学习编程的决心。

总体而言，我的程序并没有使用太多的算法，大多数情况下依赖STL自带的方法，而且大量的函数未经优化，没有将算法复杂度降到最低。有一些小的亮点，比如调整终端字体的显示颜色以便使得用户一目了然；使用正则表达式匹配ID；教师的给分和调分机制；文件管理系统等。但局部的亮点并不能掩盖总体上架构设计的失败，比如Student,Teacher类与UI类出现了耦合；虚函数设置不合理导致程序可读性变差；一些操作没有合理封装。

由于架构设计的不合理，一些原先的计划也未能在代码中得到实现。比如判断学生所选课程是否存在时间上的冲突；根据课程性质为课程设置PF；判断学生是否有退课的权限等。这也是本次程序设计的一个遗憾吧，同时也从反面说明了大型软件开发中架构设计的重要性。

## 7 源代码

见程序源码.pdf