

# 课 程 设 计 报 告

2019 年 8 月 24 日

# 学生成绩管理系统

## 目 录

1.	系统需求分析.....	1
2.	总体设计.....	3
3.	详细设计.....	5
4.	系统调试.....	10
5.	测试结果与分析 .....	12
6.	总结.....	38
附录：源程序清单 .....		40
核心的 9 个类相关的头文件和源文件 .....		40
MFC 图形界面相关的 15 个类的头文件和源文件 .....		60
第 1 题评分表 .....		118

## 1. 系统需求分析

学生成绩管理系统记录了不同学生的基本信息，包括其姓名、性别、学号，同时会记录每位同学所修课程的课程名称、原始的百分制成绩、学分数、是否记 pf，成绩录入时间，同时依据这些原始录入数据转换出对应绩点以及对应等级储存在系统中，此外还会依据这些数据统计出每位同学所修课程总数、通过的课程总数以及通过的学分数、该同学的平均绩点，系统中全体学生的数量、全体学生的平均绩点。试设计学生成绩管理系统，使之能够实现如下功能：

- (1) 登录：在正式进入系统操作界面前将有一个登录界面，需要输入正确的用户名与密码才可以登录系统进行相关操作，同时为了方便使用，可以为登录直接提供默认用户名与密码；
- (2) 读取数据存储文件中的数据：在界面挂载前自动读取存储在相应文件中的包括学生基本信息和成绩信息的数据进入本系统中，方便使用；
- (3) 实现静态拆分窗口，在操作界面左侧拆分出一个选择窗口，用以选择不同的操作界面，单击不同的栏目即可挂载不同的功能界面；
- (4) 修改登录密码：可以在登录系统后，在用户中心界面显示用户名并可以对登录密码进行修改，修改需要连续输入两次相同的异于原密码的新密码，否则提示错误，拒绝修改；
- (5) 添加新的学生：使用键盘在学生基本信息管理界面相应编辑框中分别输入姓名、学号，并在性别下拉选单中选择对应性别，创建时会由系统各数据合理性，否则无法进行创建；
- (6) 管理系统中已录入的学生的基本信息：通过输入姓名或学号进行对应学生成员的查询，若查询失败则提示相应信息，查询成功则进入管理界面，并会显示查询到的学生已有的基本信息，可进行 2 项管理操作：
  - a) 修改基本信息：可以将查询出的学生的基本信息进行修改；
  - b) 删除基本信息：可以删除该查询出的学生所有信息；
- (7) 学生成绩管理：同样通过输入姓名或学号进行对应学生成员的查询，若查询失败则提示相应信息，查询成功则进入成绩管理界面，并会显示查询到的学生已有的基本信息，可进行 4 项成绩管理操作：
  - a) 增加新的成绩：通过在对应该编辑框中输入课程名称、学分、百分制成绩、成绩录入时间（年-月-日），以及在下拉选单中选择该课程是否记 pf 来增加新的成绩，会对输入信息进行检查，会对错误的成绩录入时间进行报错处理，以

及学分处或百分制成绩处输入不合常理的数据或其他字符亦会报错，均正确方可录入相应新成绩；

- b) 查询已有成绩：通过输入课程名称可以每次查询一门已经录入成功的课程，并显示其所有相关信息；
  - c) 修改已有成绩：可以修改上述查询出的成绩，输入方式与录入新成绩的方式一致；
  - d) 删除已有成绩：可以删除上述查询出的成绩；
- (8) 打印学生成绩单：通过输入姓名或学号进行对应学生成员的查询，若查询失败则提示相应信息，查询成功则打印该学生的成绩：若该学生无成绩，或信息库中没有学生成员，则会反馈相应信息；若有成绩则先将该学生的成绩进行排序后输出，同时也会输出该学生总共修习课程数以及通过的课程数和学分数，并会显示其平均绩点；
- (9) 总体统计信息情况显示：会呈现该系统中现在存储有的全体学生的统计信息，包括姓名、学号、性别、均绩、完成学分数、完成课程数，并显示管理系统中总计学生的数量以及全体学生的平均绩点；
- (10) 设置有系统菜单：可以从开始选项中选择“退出”来退出本系统（也可以之间按系统界面右上角的叉来退出），可以在菜单选项中选择不同界面进行切换；
- (11) 数据自动存储：退出系统时自动将数据写入到对应的数据存储文件中，使得下一次打开时系统可以自动读入数据，方便操作。

## 2. 总体设计

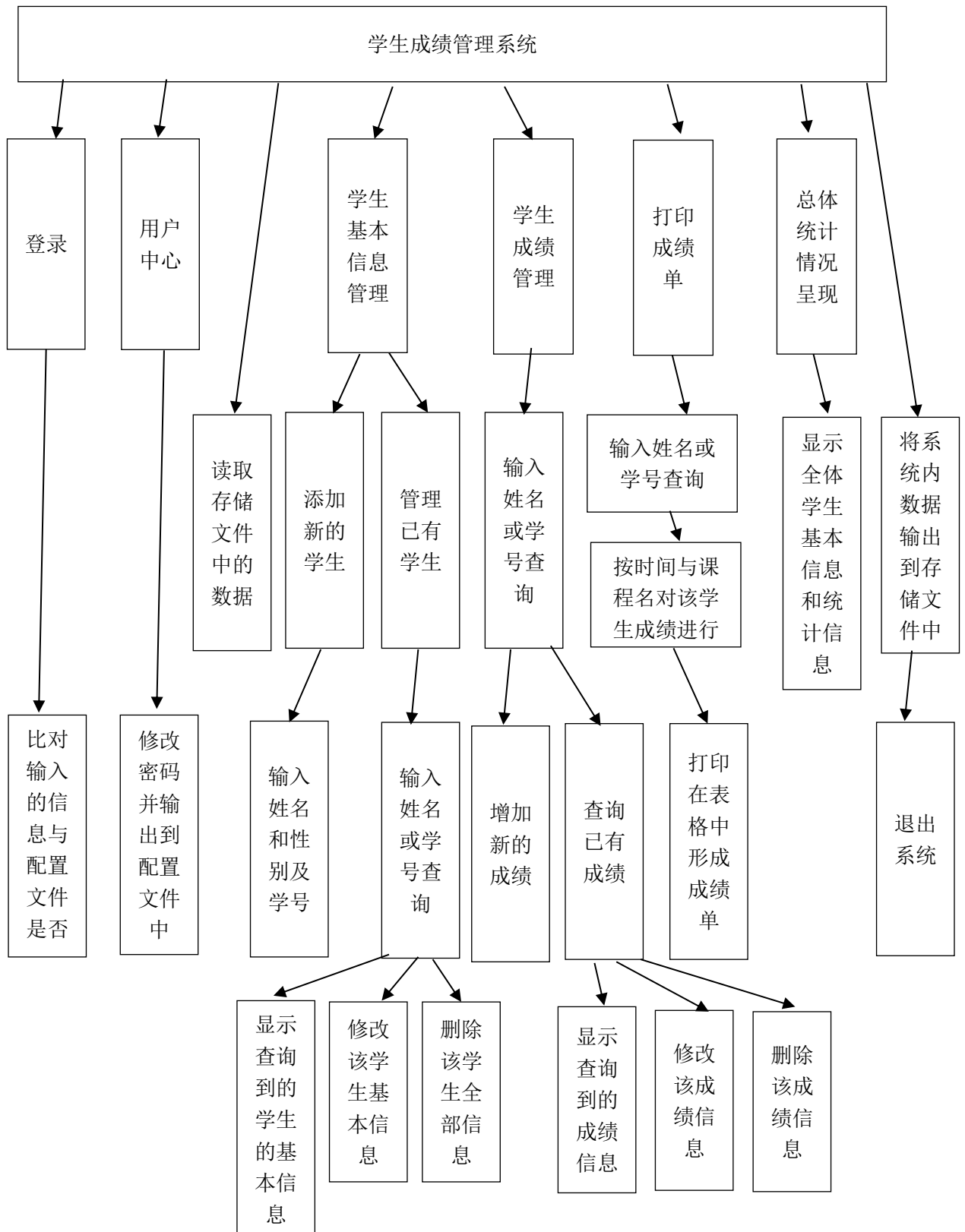
学生成绩管理系统依据划分出的主要界面可以分为 6 个大的功能板块，分别是：a) 输入用户名密码的登录界面；b) 显示用户名并可修改密码的用户中心界面；c) 可以增加、查询、修改、删除学生基本信息的学生基本信息管理界面；d) 可以通过姓名或学号查询到某一个学生后进行成绩的增加、查询、修改、删除的成绩管理界面；e) 可以通过姓名或学号查询到某一个学生后，对其成绩进行排序后输出在表格中，并给出其总体情况的打印成绩单界面；f) 在表格中给出系统中全体学生基本信息和基本成绩情况以及总体成绩情况的总体信息统计界面。

这六大界面具体可以细分为 14 个子功能，分别是：1) 将输入用户名密码与配置文件存储数据比对符合后登录；2) 读取存储文件中的数据；3) 修改用户密码并输出到配置文件中；4) 增加新的学生成员并为其添加基本信息；5) 通过输入姓名或学号进行学生成员的查询；6) 查询后可以修改该学生成员的基本信息；7) 查询后可删除该学生成员全部信息；8) 查询学生后为其添加新的成绩；9) 查询学生后查询已有成绩；10) 查询已有成绩后修改该成绩；11) 查询已有成绩后删除该成绩；12) 查询学生后对其成绩进行排序输出在表格中，打印成绩单；13) 打印全体学生基本信息和统计情况在总体统计表中；14) 退出系统时自动将系统内的数据输出到存储文件中。

其中：学生基本信息包括其姓名、学号、性别，成绩信息包括课程名称、学分、百分制成绩、是否记 pf、成绩录入时间

不同界面下的增删查改等功能只需在对应功能框下可以写的编辑框中输入符合要求的数据，再按对应按钮确认即可。输入要求例如：各输入框不能有未输入的空白情况；学号必须有纯数字构成，且不能与系统中已储存有学号相同；成绩处学分需输入正整数，百分制成绩需输入 0-100 间整数，成绩录入时间需输入真实存在的时间。若不满足这些输入要求会具体提示何输入错误，并要求使用者修正错误的输入方可成功执行相应功能。

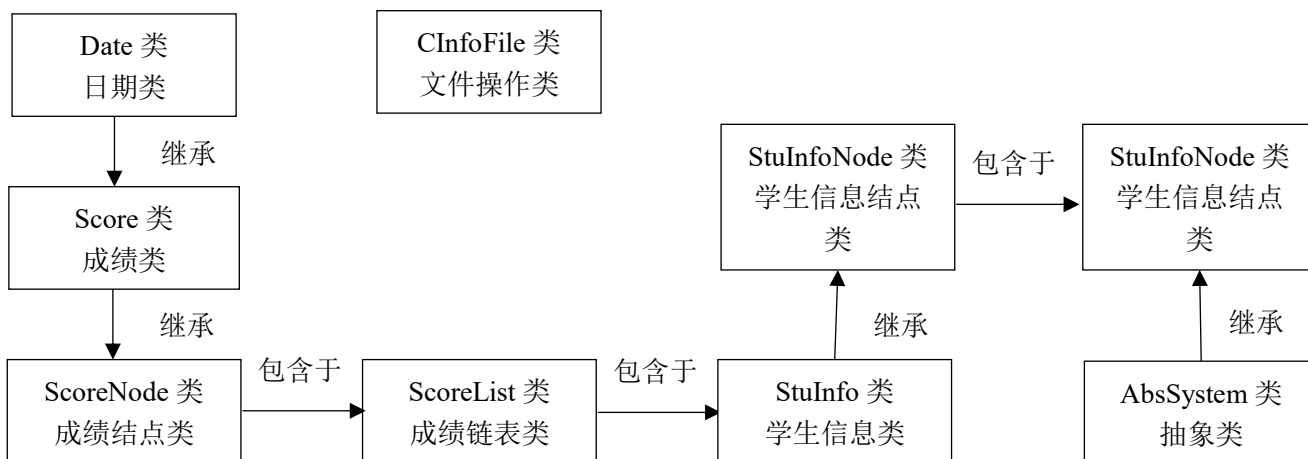
学生成绩管理系统功能模块图：



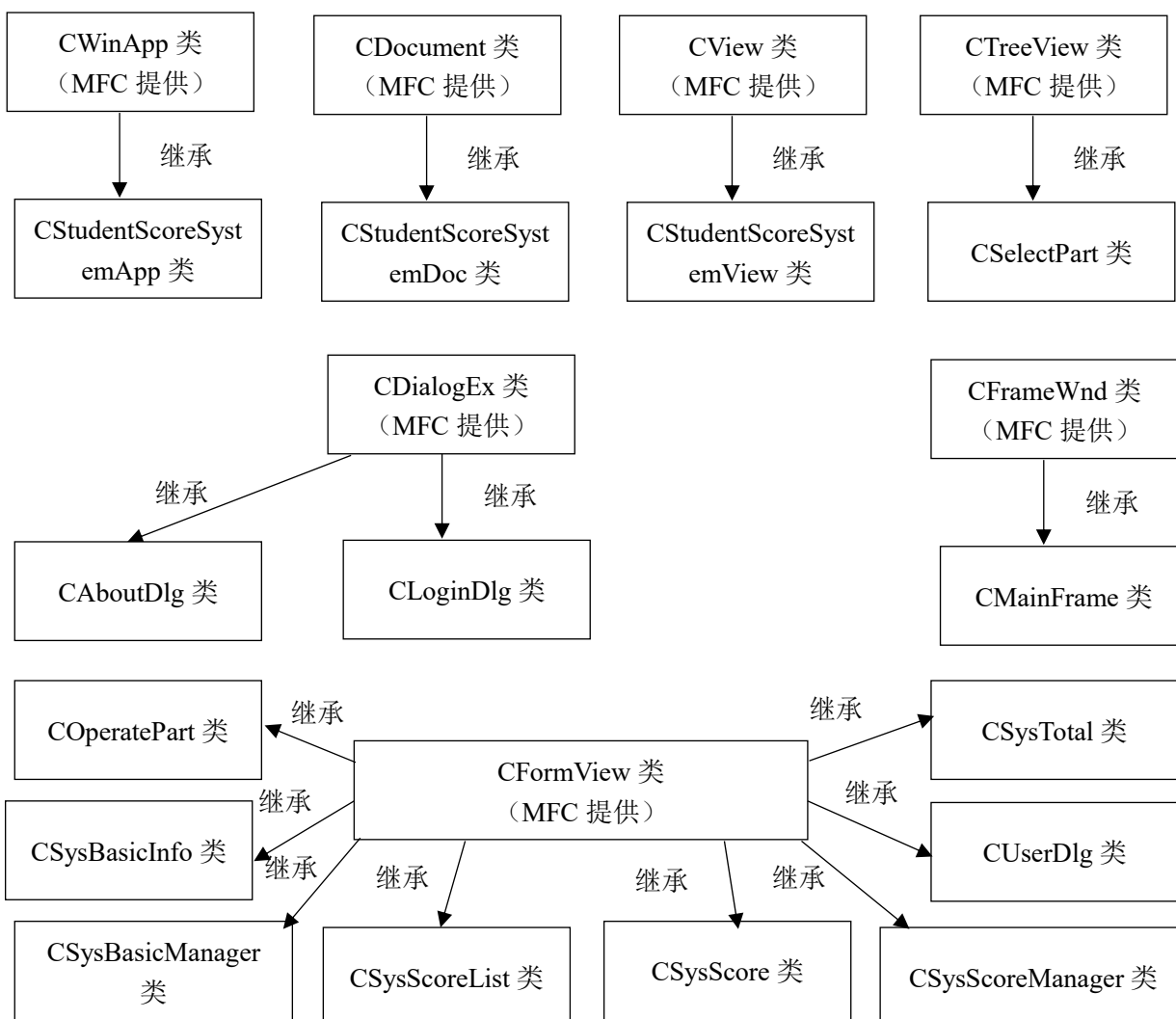
### 3. 详细设计

学生成绩管理系统的类层次图为：

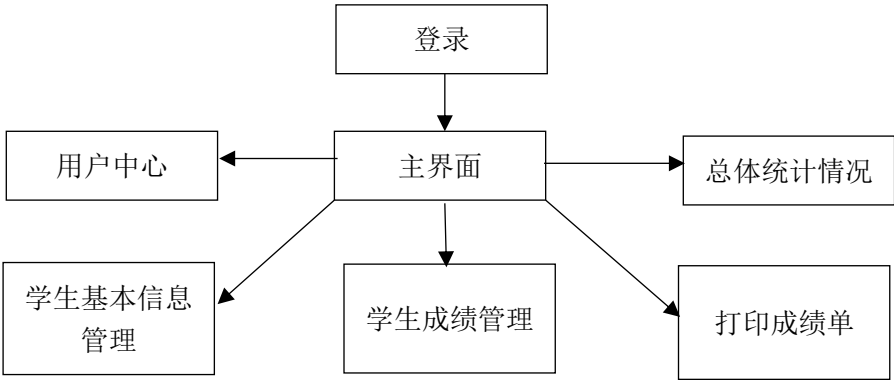
进行数据管理、存储与操作的 9 个核心的类的层次图：



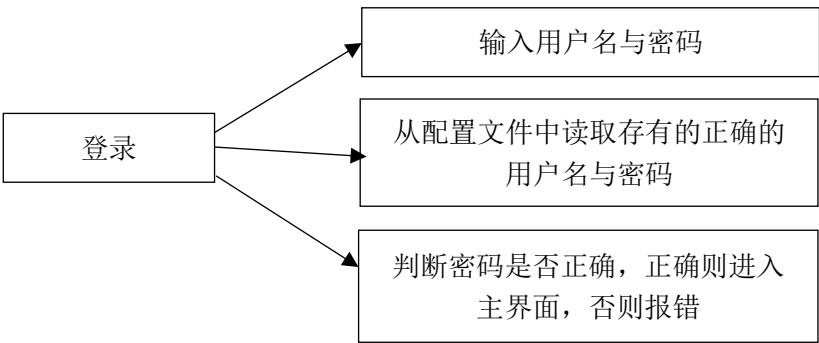
用于图形界面呈现的基于 MFC 类库的类的层次图：



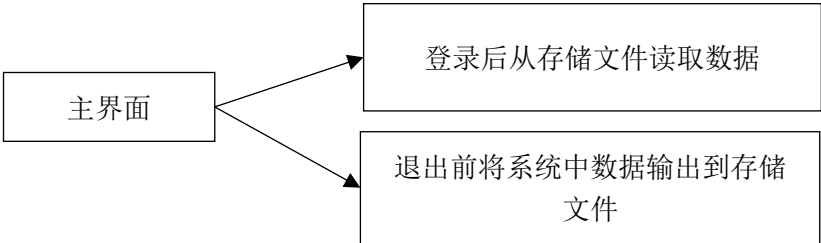
学生成绩管理系统中各功能模块的实现：（由于是图形界面，依托于不同界面展开不同功能的实现与集成）



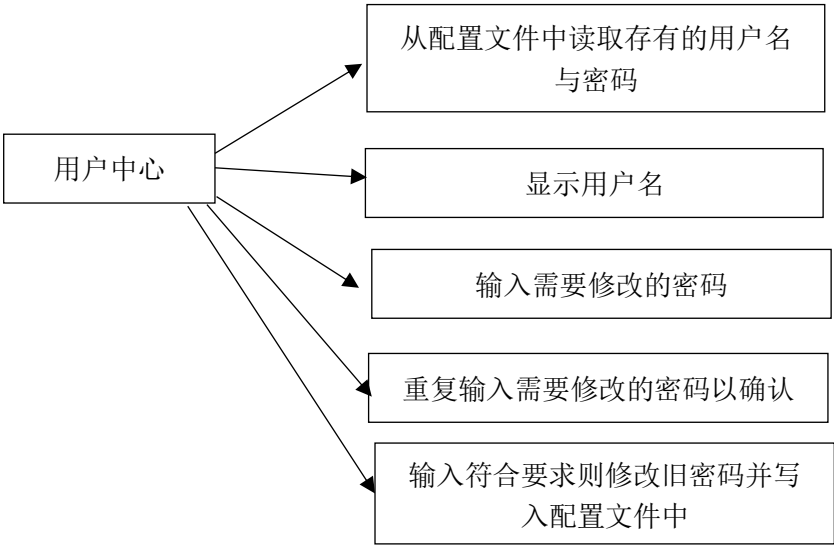
1. 登录界面：



2. 主界面

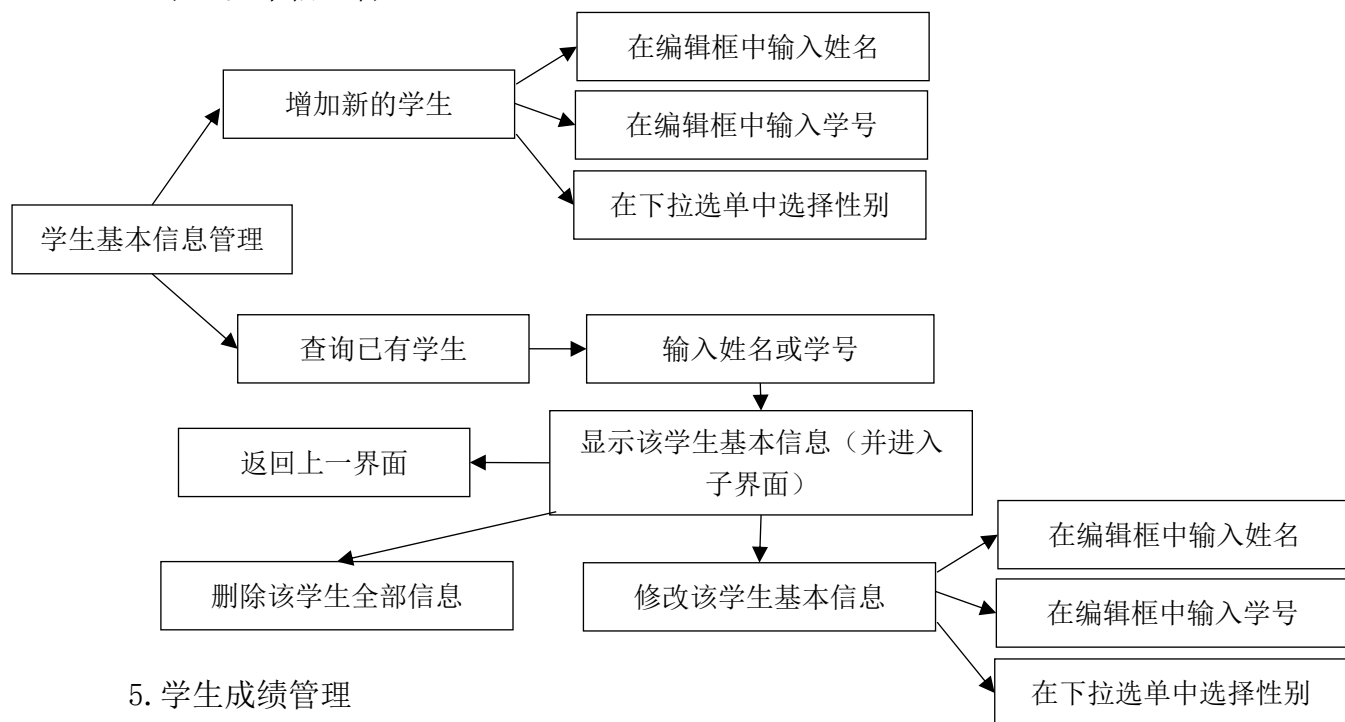


3. 用户中心

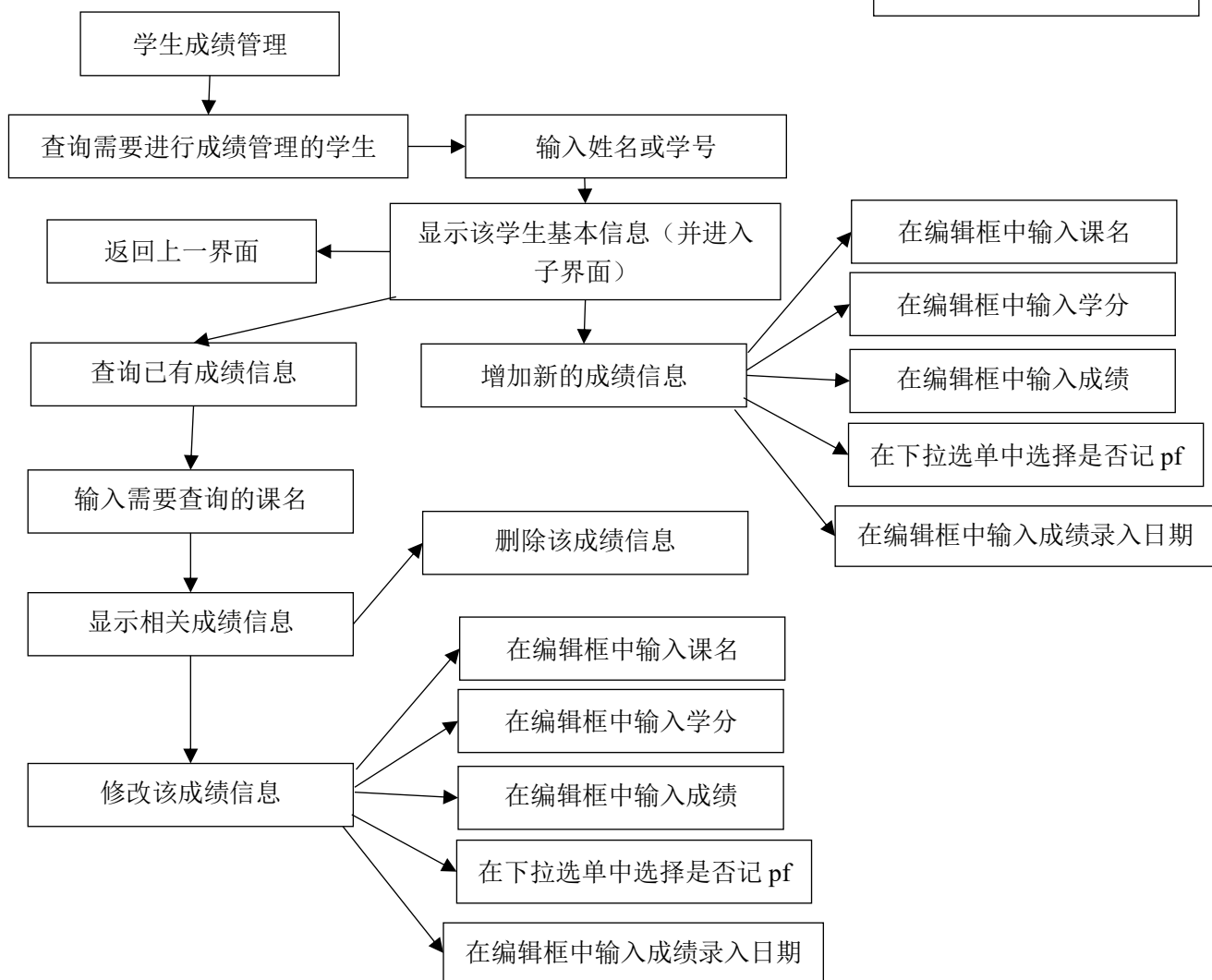




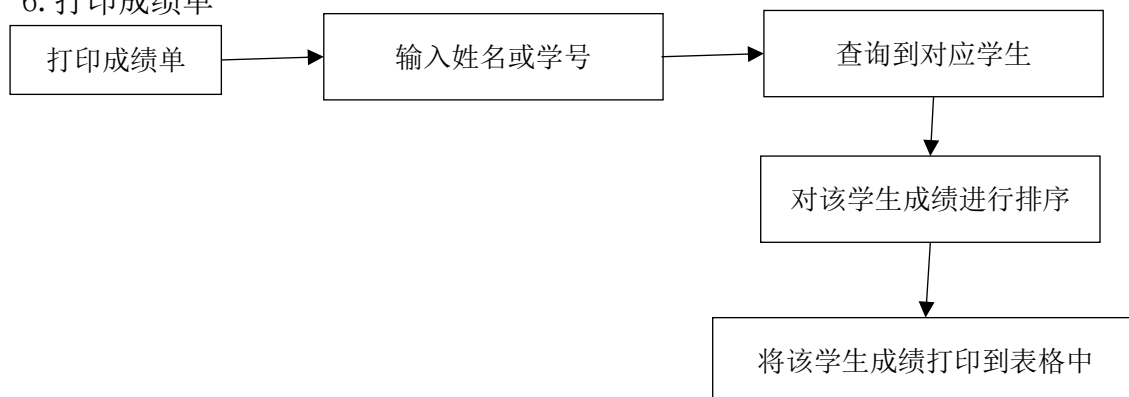
#### 4. 学生基本信息管理



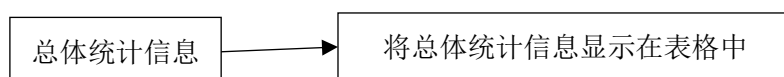
#### 5. 学生成绩管理



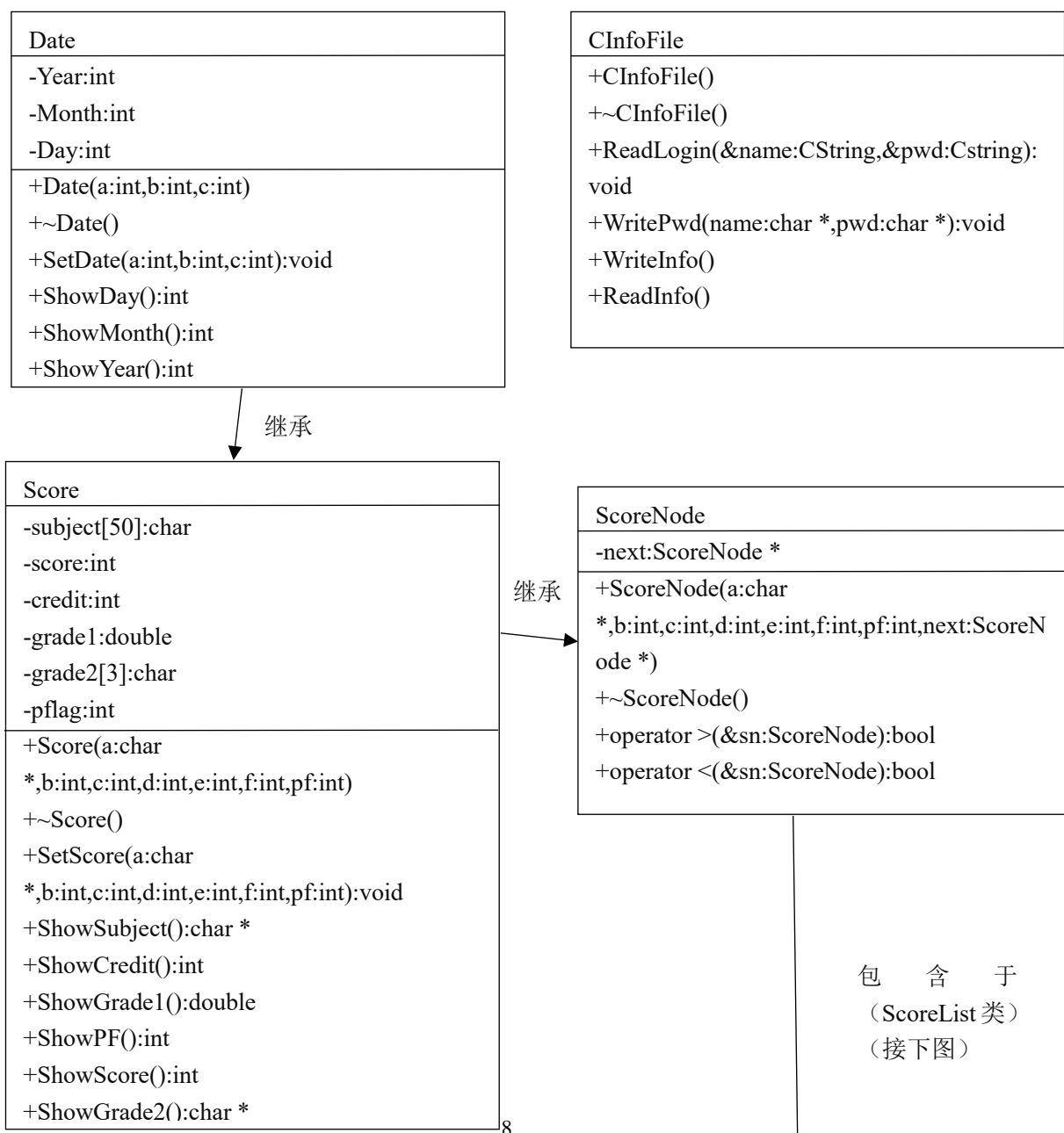
## 6. 打印成绩单

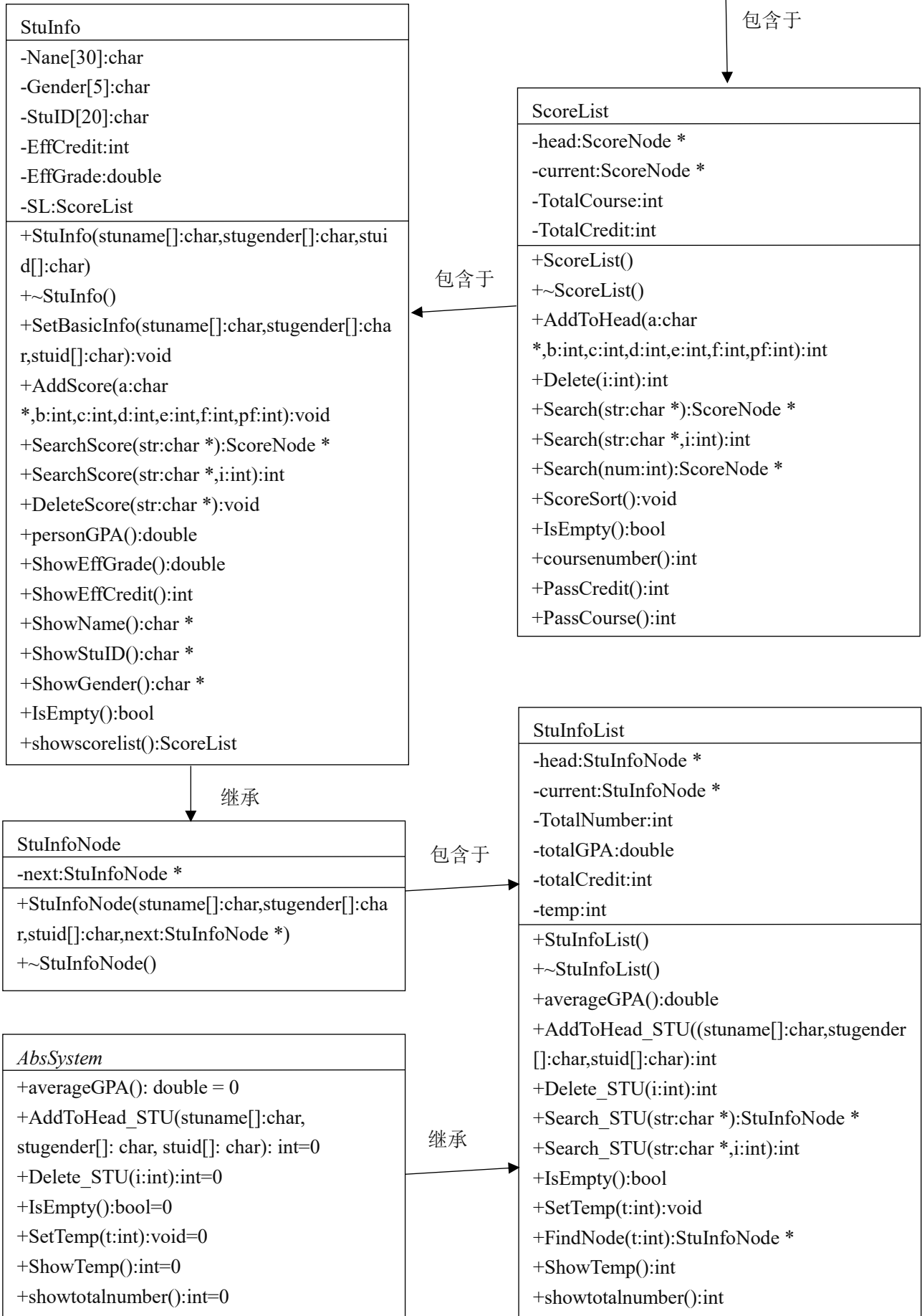


## 7. 总体统计信息



学生成绩管理系统 9 个核心类的 UML 图：





## 4. 系统调试

为了便于系统数据的灵活的使用与删改，使得操作更加方便，同时也使数据的之间的层次关系更加分明，我在设计和实现学生成绩管理系统的时候大量采用了链表这一数据结构，把某学生的所有成绩封在一张链表里，再把所有学生再次封在一张链表里。由于一开始并未考虑好是否要做 MFC 图形界面，所以最先编写的是这个程序最为核心的部分，也就是数据的存储与增删查改以及操作和反馈的部分，基本不涉及外部对数据的输入和数据输出的操作，这样也就使得我把某些内容暂且先搁置了，比如对于数据正确性（即数据是否符合输入的标准要求）的检验，又比如查验链表是否为空等。第一个方面后来采用 MFC 图形界面编程后，放在图形界面类中进行检查，而且由于编辑框选取的是 value 型的 int 或者 CString，使得对于数据符合标准性的检验变得相对更简单，比如检验学号是否为纯数字，只需要用 CString 类自带的一个函数 SpanIncluding()即完成了较好的解决；又比如成绩处需要输入整数，int 型的编辑框在输入不是整数的情况下（比如输入 a）会提示错误，我只需要编写对数据范围作出限制的相关内容即可，比如成绩必须在 0-100 的区间内，而日期是否真实存在也在这一步进行检查，这方便了日期类的设计，如果日期检查通过，才会拿这些日期数据去创建对应的对象，否则则会弹出消息框要求重新输入；其他相关的数据类也是如此。第二个方面最早觉得可能比较麻烦，所以也就先放在一旁，打算后期再处理，但在后来编写 MFC 图形界面的过程中，受到了 MFC 中函数的启发，CString 类下有一个函数叫 IsEmpty()用于判断这个 CString 对象是否为空，同理我也给自己的两个链表类也写了 IsEmpty()函数进行链表是否为空的判断。

一开始做查询功能的时候只是设计返回结点指针的函数，但是后来在编写一些功能的时候发现，如果提供这个结点在链表里的位置可能会更加便于操作，而且也可以有效避免使用指针不当可能出现的问题，因此重载了相同的查询函数，改变了传递参量，返回值为该结点在链表中的位置号，重载也使得调用更加方便了。

因为做了图形界面，并且设计了查询相应信息后可以进入对应的管理界面这样的功能，方便操作和使用，但是在实现的过程中发现了一个问题——前一个父页面的查询数据并没有办法带到子页面中，但是考虑到这个查询出来之后就会立刻跳转到子界面，不经过查询界面是无法进入子界面的这样一个特性，于是在学生信息链表类（即 StuInfoList 类）中增加了一个私有成员 Temp，保存查询到的这个结点的位置号，选择传递位置号自然是因为记录和提取一个 int 型变量显然更加方便而且不易出错，进入子界面后再利用这个储存的 int 数通过相应函数找到对应的结点，从而进行后续操作。

因为要完成 MFC 的原因，将核心类都写在一个.h 文件和一个.cpp 文件中，但是界面上各种内容的实现需要在对应类的.cpp 文件中操作，这就使得我创建的学生信息链表对象需要在这些类中都能够使用，为解决这个问题，我也回顾了上学期的课程内容以及一些相关资料，最后通过引用头文件以及加 `extern` 声明的方式解决了这个问题。

在设计中考虑了文件操作——文件操作使得系统的使用可以变得更加方便，数据可以直接读、存，而不用每次使用都输入一遍，也方便了调试测试的工作。但是一开始并没有写有关文件操作的内容，是因为感觉这样链表套链表的文件操作比较费脑筋——输出还方便，但是读取数据内容实在是一时没有头绪，想不到什么简便的方法。后来在学习 MFC 的各种设计与实现中有了启发，写了一个负责文件操作的类。储存文件除了一个存着用户名与密码的 `login.ini` 配置文件外，有着两个存储相关数据的文件——一个存着学生基本信息，另一个存着所有学生的成绩——这样的实现也和我类的层次与设计有关。在 MFC 学习的过程中，我也发现了 `strtok()` 这样一个非常好用的函数，可以直接将一个字符串依据参数中的分隔符进行不断分隔，这里选用的分隔符是 ‘|’，也是考虑到正常的输入情况下，各种数据都不能是这个字符，这个函数的使用大大简化了我读取数据函数的设计与编写，同时也规划了我写入文件函数的编写——每打印一个信息都用 ‘|’ 分隔开，每打印完一个结点的信息就换行。读取函数把表头读掉之后，进入循环一行一行读，虽然循环条件用了文件对象的 `eof()` 函数，但是考虑到这个函数会“迟钝”，仍然在文件中用了一个 `if-else` 结构来判断读行读到的字符串是否为空，若空则结束循环。设计两个文件是设计先读一个基本信息文件，把学生链表里的每个结点都创建出来，再去读成绩文件，成绩文件每行除了成绩信息外，还有一个判断成绩归属的信息，一开始是打算用姓名的，但是考虑到姓名可能会出现重复的现象，所以最后又修改成学号（在我的设计中学号是不能重复的，重复将会在添加新人员的时候报错而拒绝添加），通过学号查询对应的学生对象来为其添加成绩结点，从而方便高效地实现了文件操作。

为了提高系统的鲁棒性，也做了不少对于数据检查的内容，比如上面提到过的日期检查之类的。但是在调试的时候尝试各式各样的输入，还是出现过均绩数据输出怪异的结果这样一个情况。检查后发现是因为如果一个学生的课程都是记 `pf` 型的课程则由于我设计的返回绩点的函数面对 `pf` 课程会返回值 -1.00，导致计算均绩的函数将出现 -1.00/0 这样的情况，但是在编写计算均绩的函数时又忽略了这种情况，于是用 `if-else` 添加了分支情况，从而对于没有办法计算绩点的情况给出一个负值，在显示的地方也用 `if-else` 结构检查，从而给出不同的正确结果。

## 5. 测试结果与分析

### (1) 使用说明与测试结果

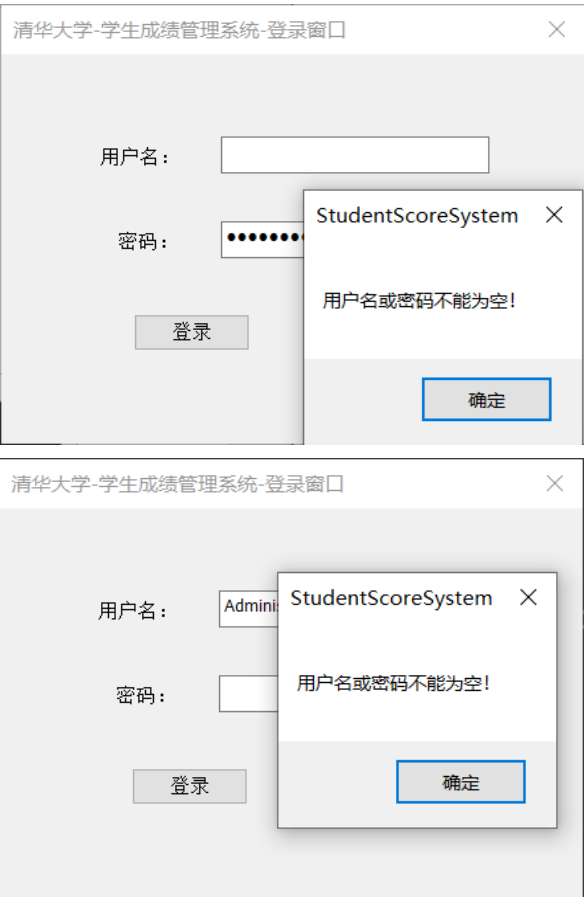
下面将依次介绍与测试我编写的学生成绩管理系统的各个功能，使用说明和测试结果展示如下：

测试数据文件在本小节最后。

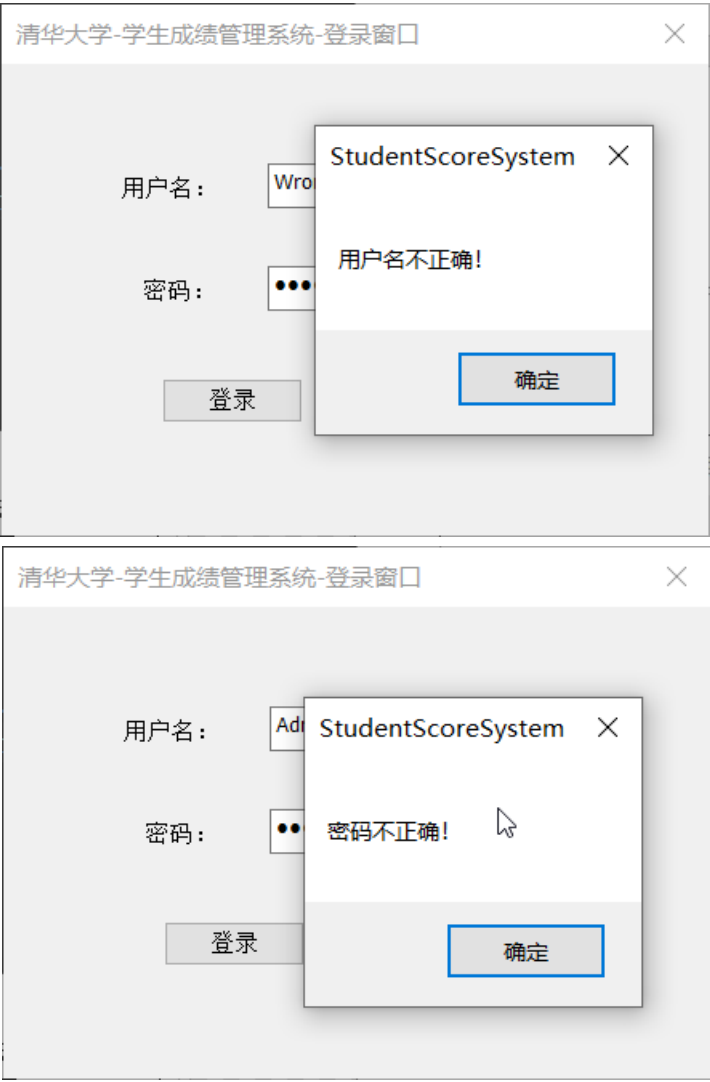
首先运行，将会出现一个登录窗口，需要你输入用户名与密码，为了方便使用与调试，我把正确的用户名与密码直接默认显示上了。



如果我们尝试去掉用户名或者密码，按登录按钮，则会出现如下提示，并拒绝登录进入系统。

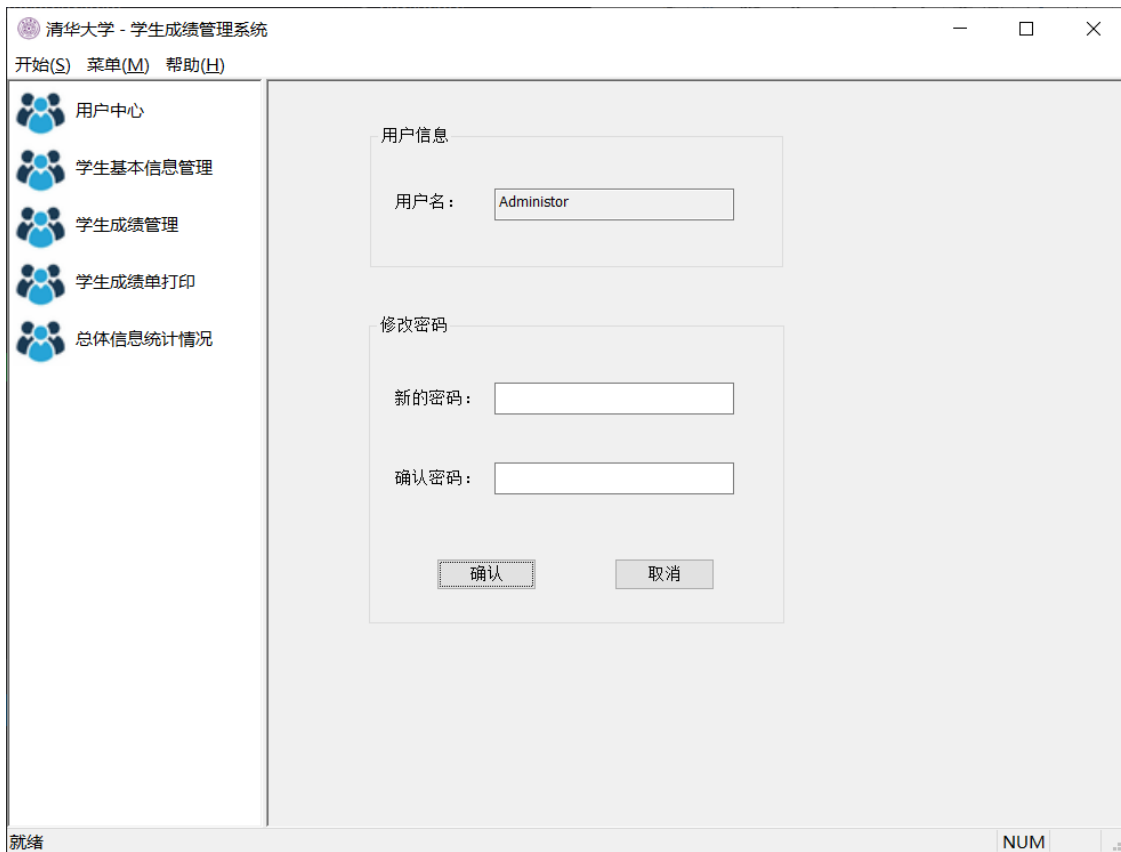


而如果输入错误的用户名或者密码则会出现如下提示，同样拒绝登录系统：

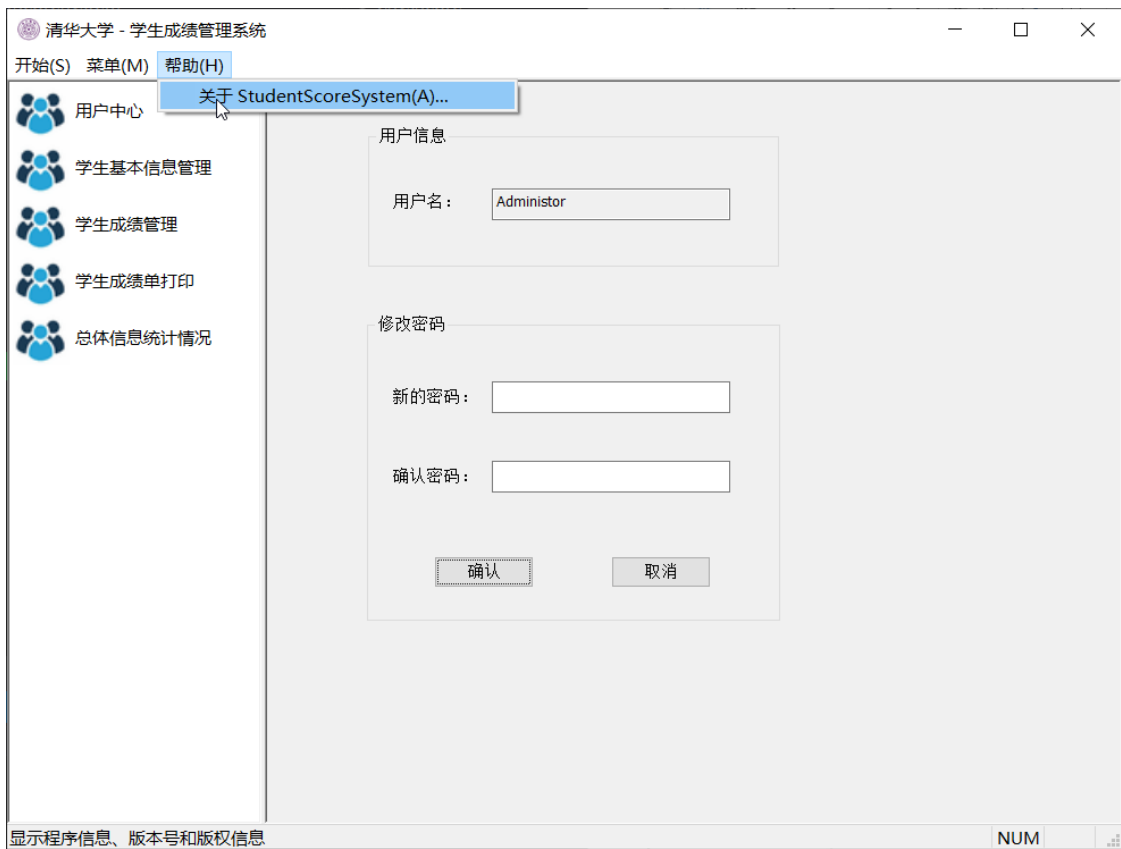


按右上角的×或者直接按退出按钮就会直接结束整个系统的运行。

输入正确的用户名与密码登录系统之后，这个登录窗口会消失，取而代之的是一个新的界面：  
默认挂载的界面是第一个的用户中心界面

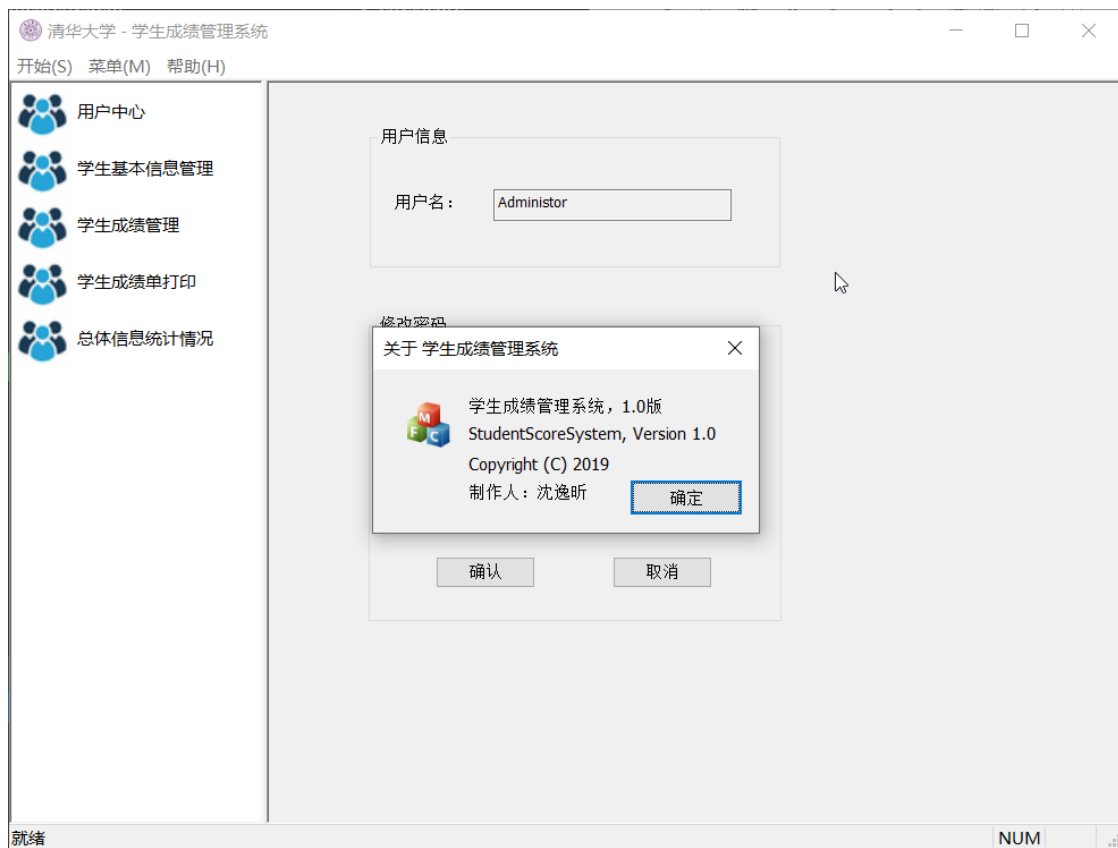


按下上方菜单栏中的帮助选单中的子项

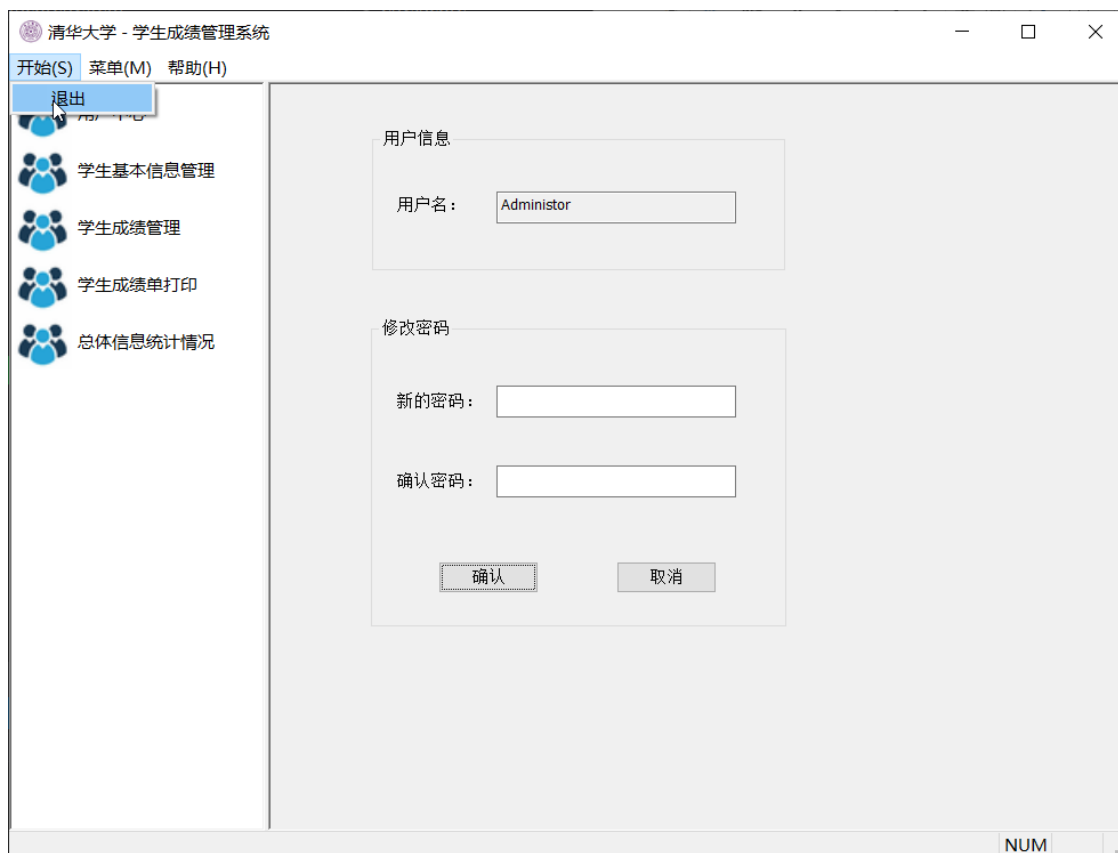


可以看到关于消息框，如下图：

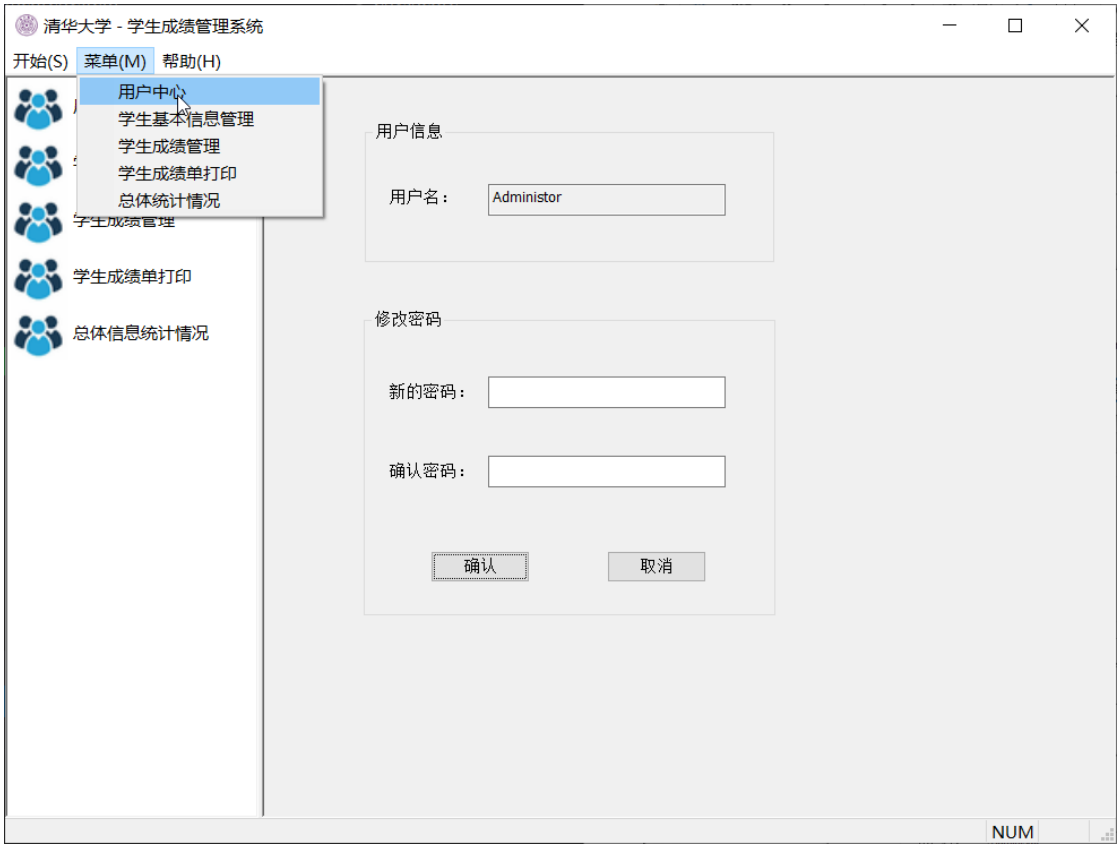




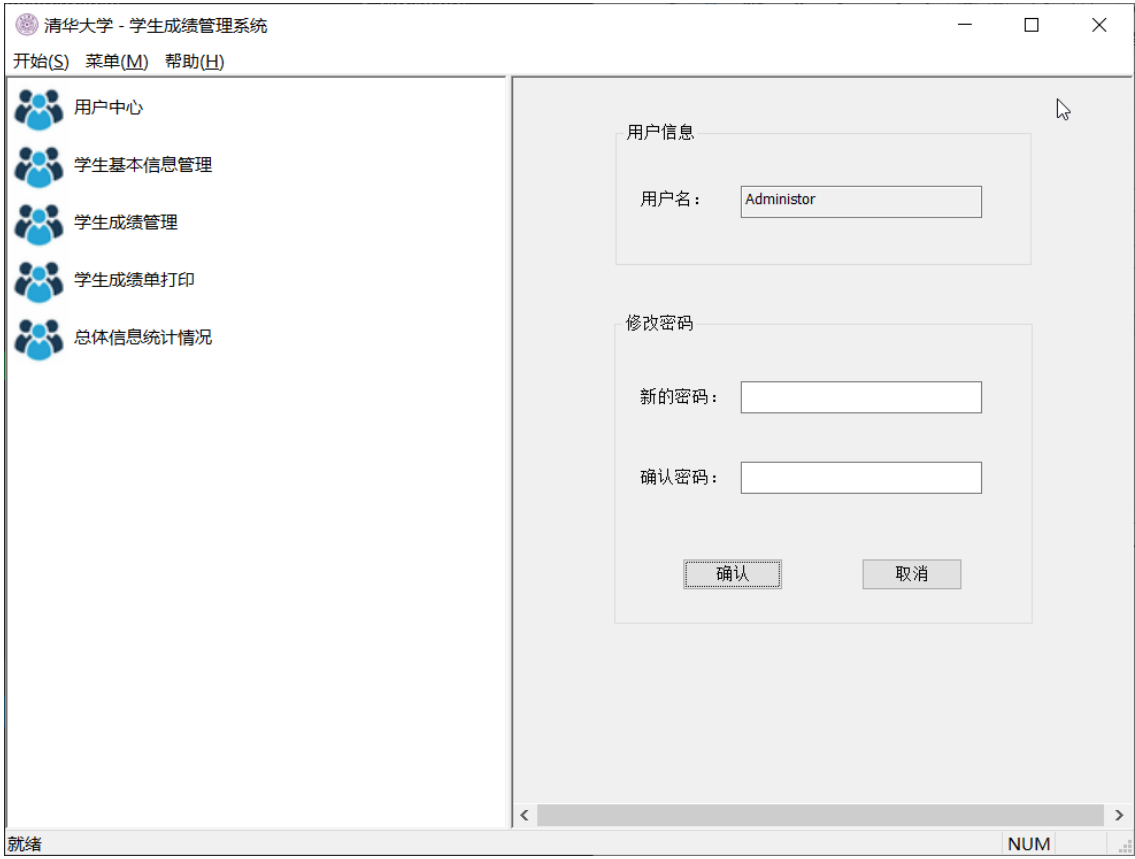
而开始选单下有退出选项，按下即可退出系统，同时会将系统中的数据都输入到存储文件中



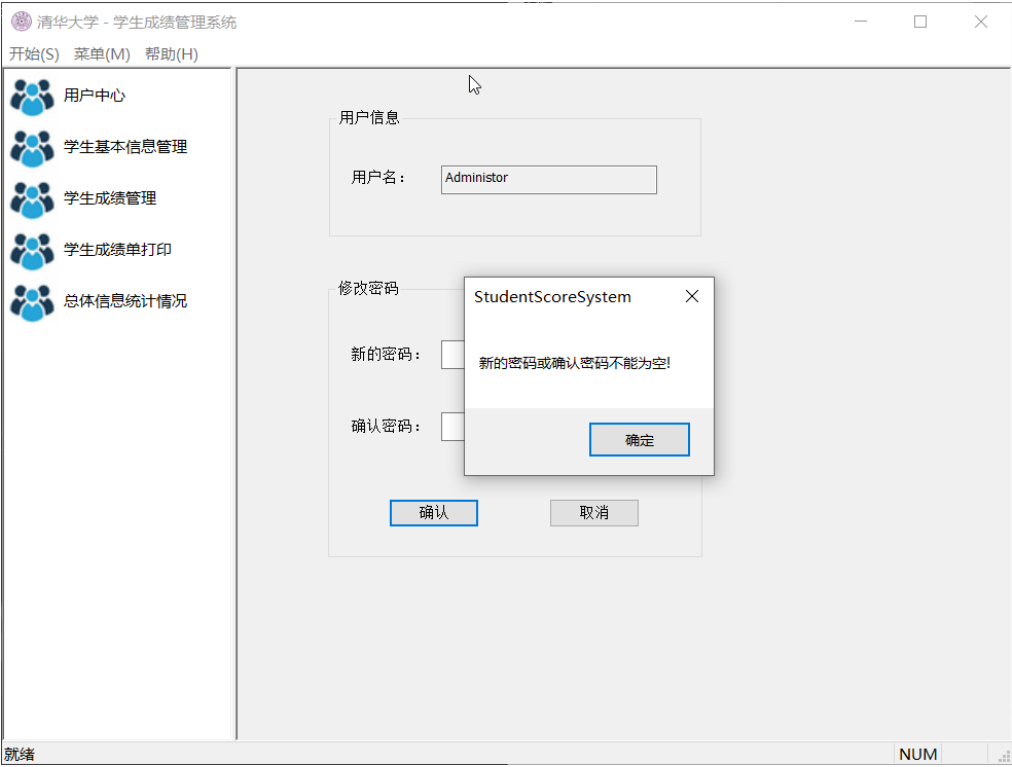
菜单选单中的内容和左边拆分窗口做的界面选择是一样的，可以在菜单处选择不同选项进入不同界面，也可以直接在主界面左边那一栏选择不同的条目进入相应界面。



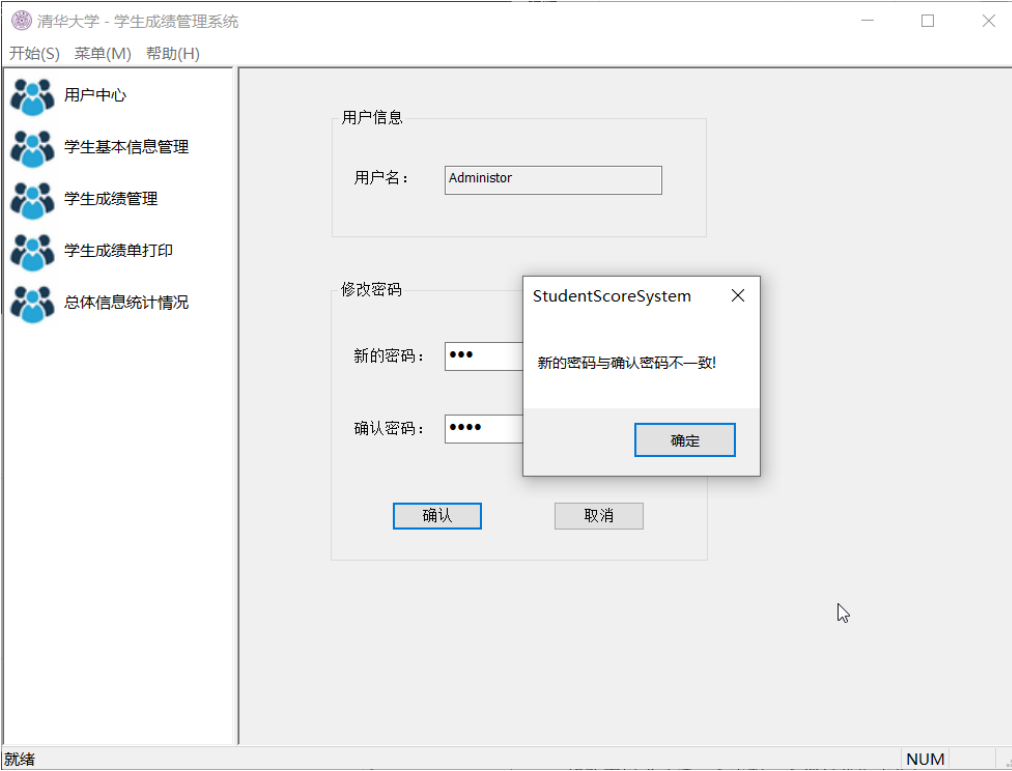
同时左右两个拆分窗口中间的栏是可以拖动而调节左右两框的大小的：



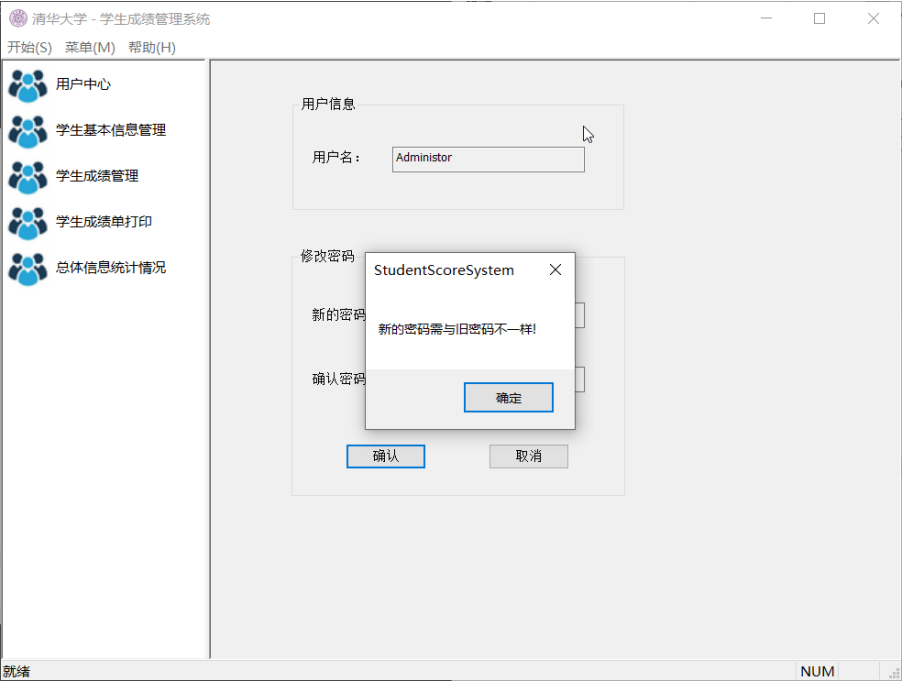
在用户中心界面，会显示用户名，这是我默认设置好的，放在配置文件中，不可更改，而可以更改的是密码，需要在下面修改密码的功能部分输入新的密码并在确认密码处重复输入一遍。同样，此处的输入结果也是加密不可见的。如果空着，将会出现如下情况：



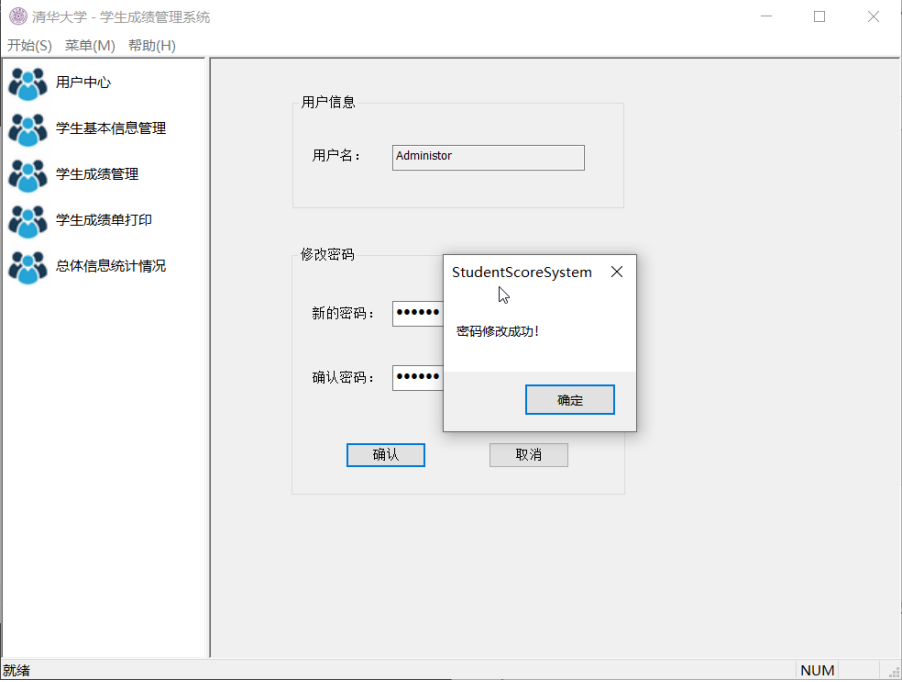
而如果两遍密码不一致将会出现如下情况：



如果新密码与旧密码一样会出现如下情况：



修改密码成功后，会出现一个消息弹框来告知：

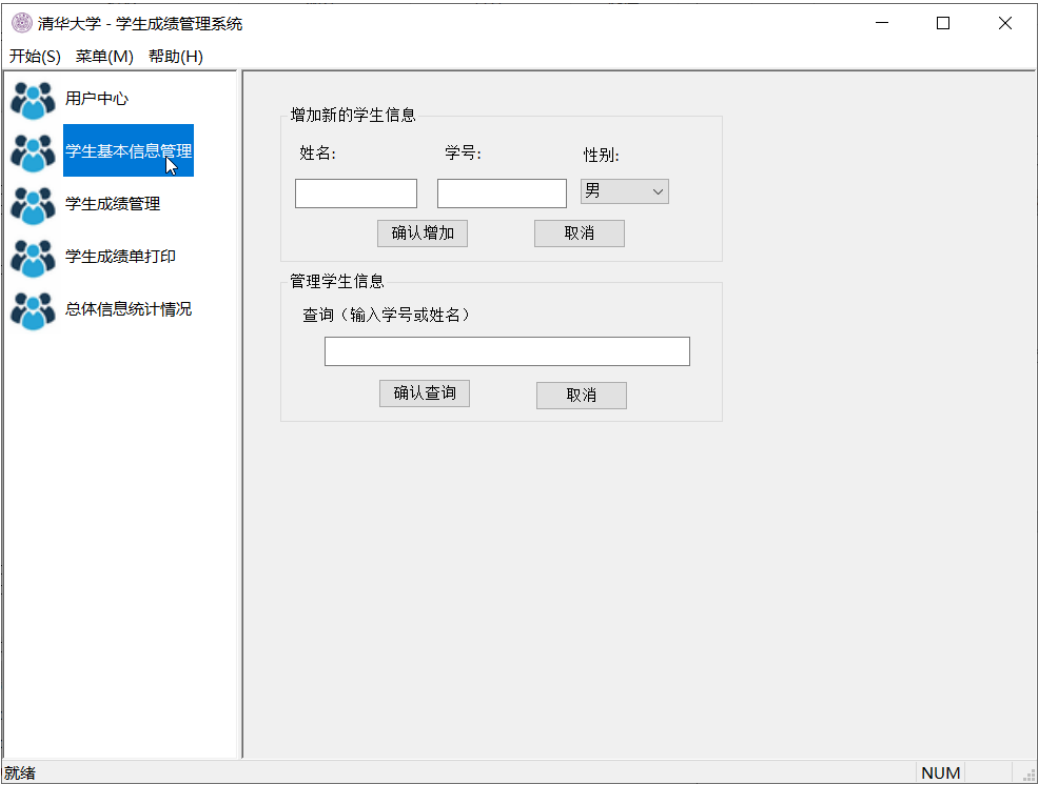


同时我在这一次修改时写入的新密码为 123456，我们可以看到配置文件内容为：



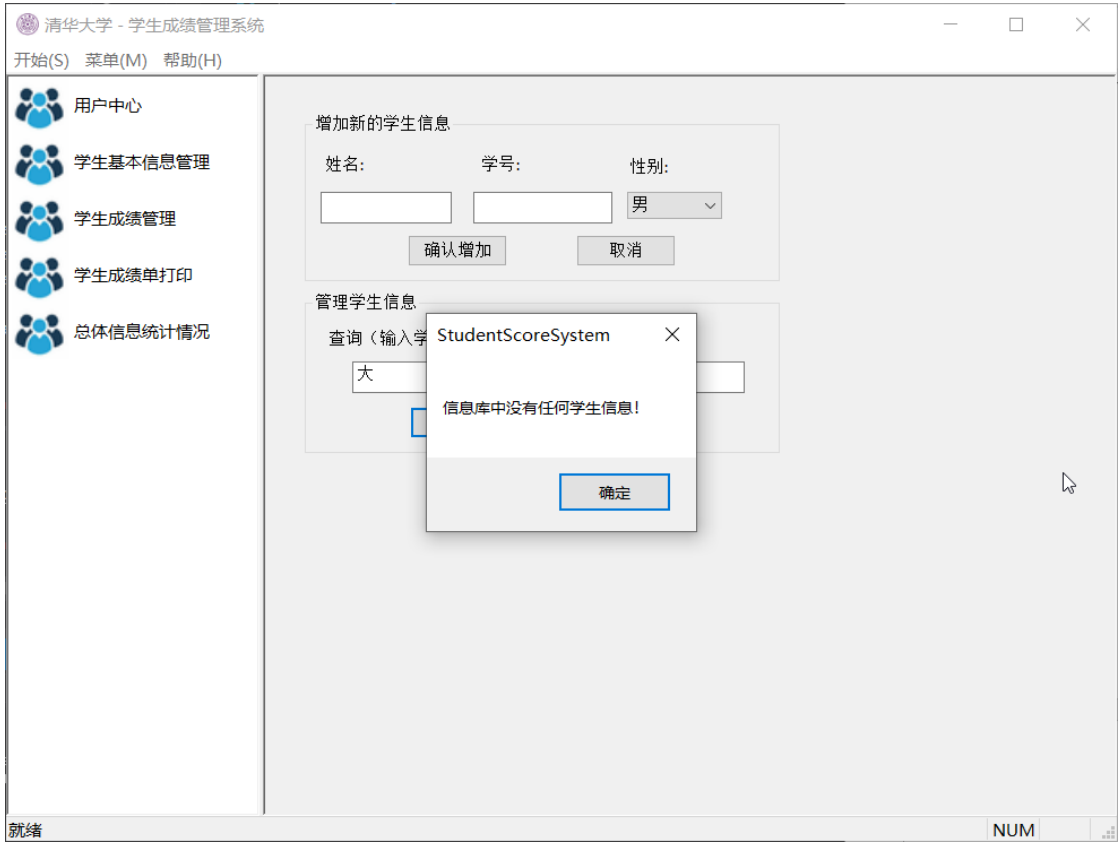
修改密码成功之后或者在未修改时按下取消按钮，将会清空两个编辑框中内容，与刚挂载该界面时一样。

单击左框的学生基本信息管理，将会挂载新的功能界面，其他界面的选择类似，后面不再赘述

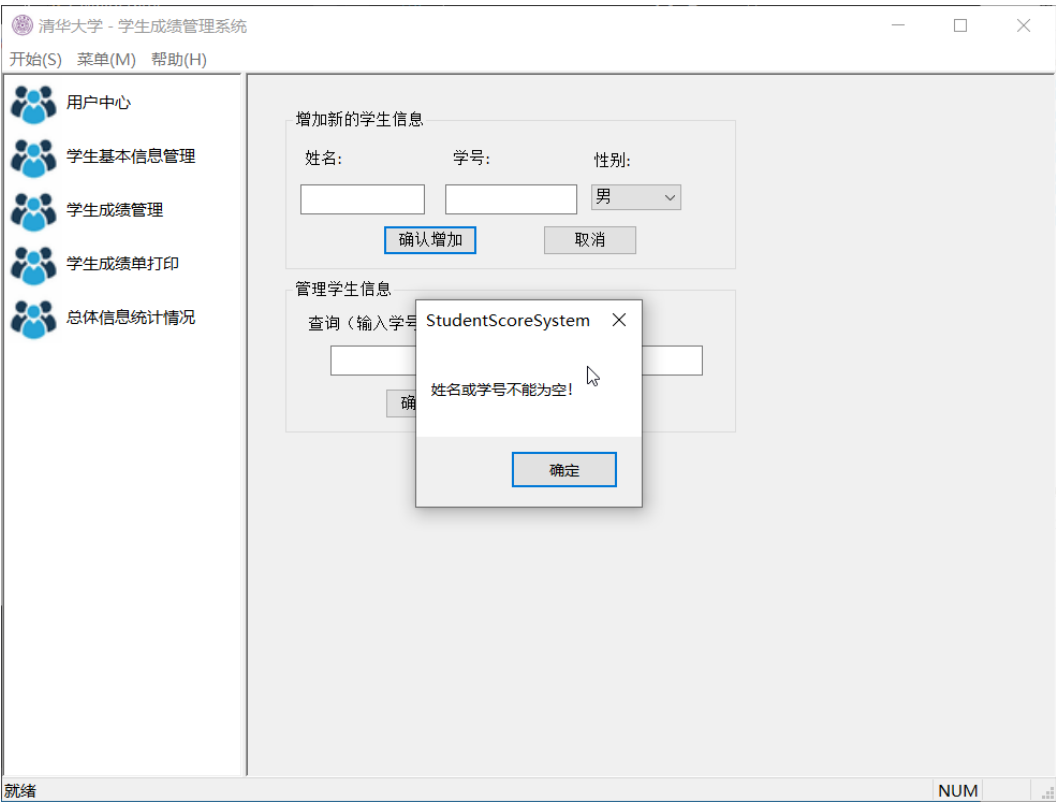


在这一界面中可以增加新的学生信息，同样，增加成功后或按取消按钮可以将各框回复初始状态（即两个编辑框清空，同时性别选单默认回到“男”）

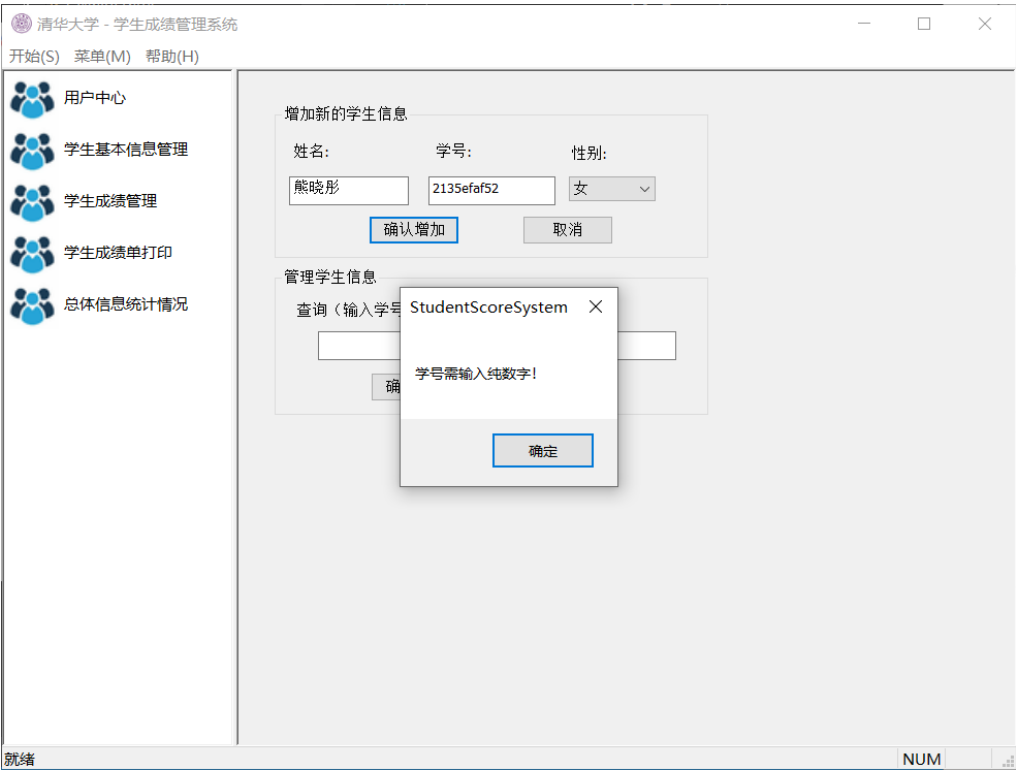
若信息库内没有学生信息，则会提示：



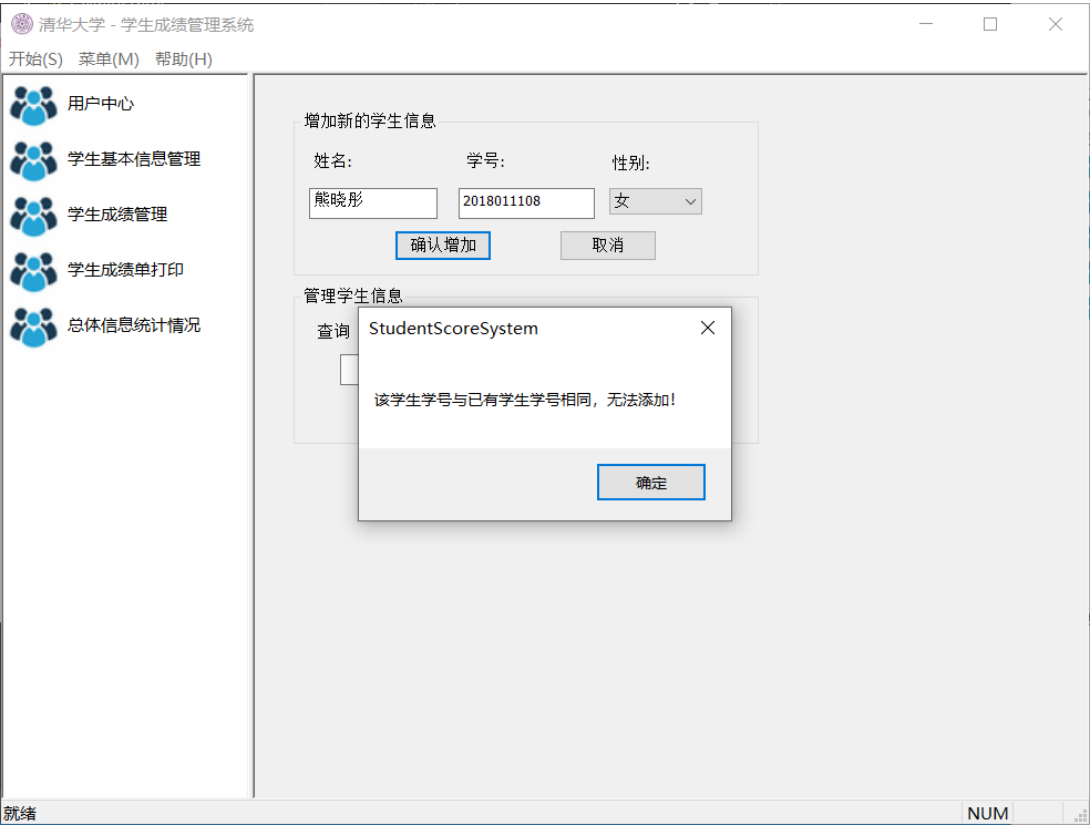
若空着同样会给出弹框：



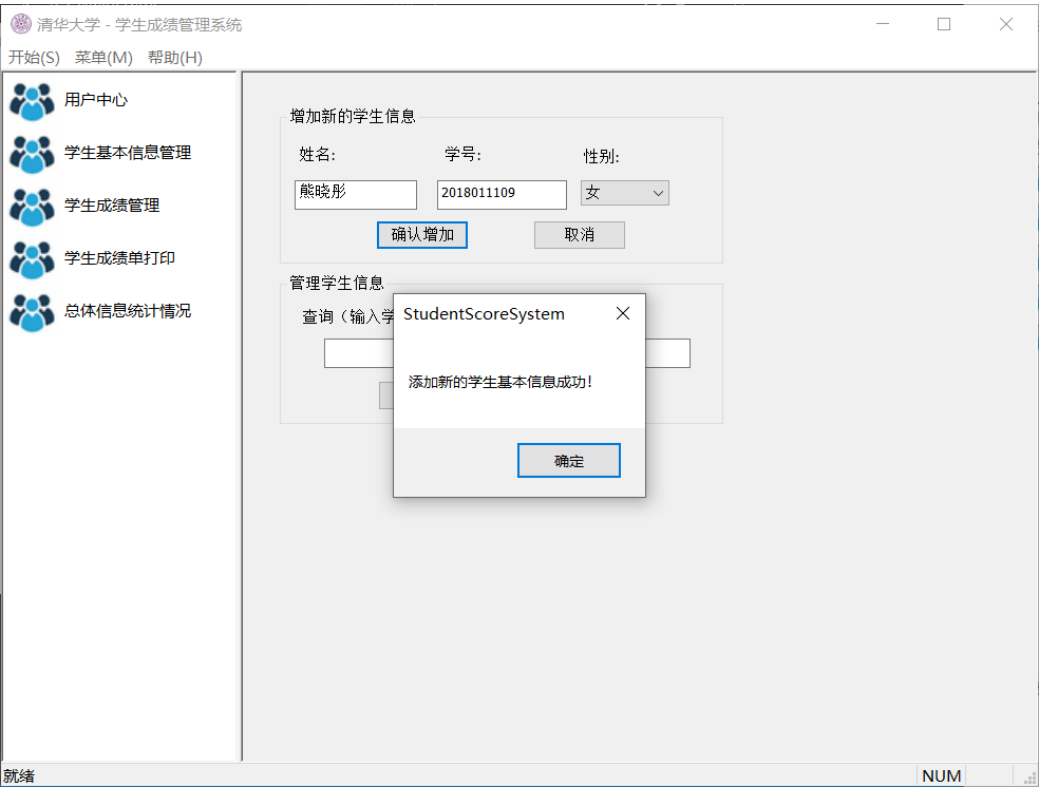
输入一个姓名后，若学号输入不是纯数字，也会弹出消息框提示错误：



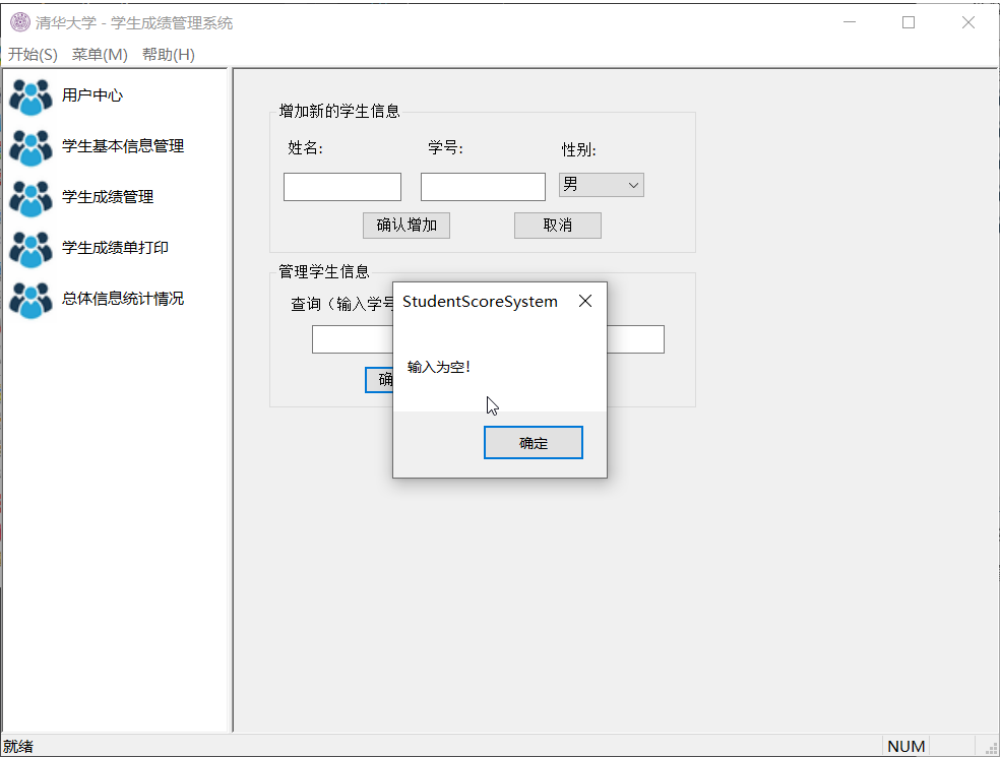
若输入的学号与已有库中的学生成员学号相同也将提示错误无法添加：  
(此处因为之前已经测试过了，所以系统里因文件读取数据的操作已经存在了一些成员)



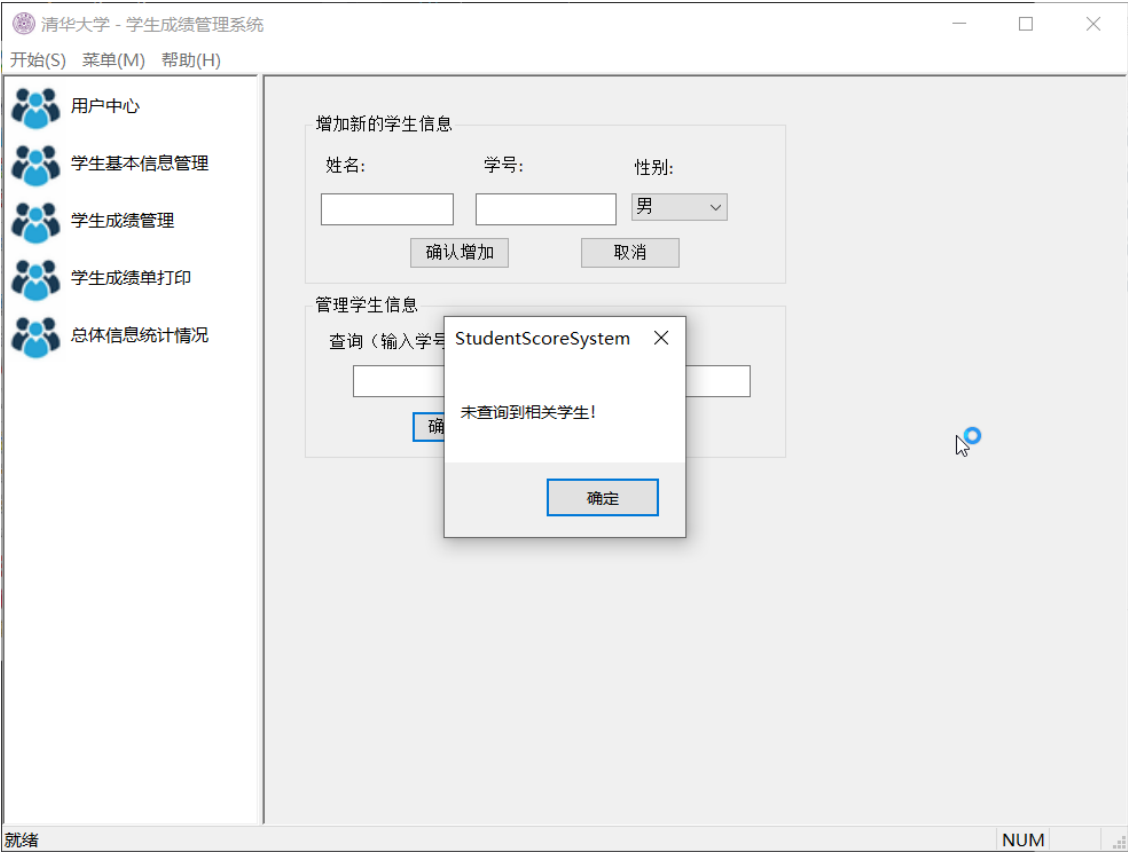
当数据检验都通过时，按下确认增加则会添加成功，并弹出消息框提示：



在该界面，还可以通过输入已有学生的姓名或者学号查询并管理相关学生的信息：  
空着依然会报错：



如果输入并未录入过的姓名或者学号（即查询不到），则会直接清空编辑框内容，同时弹出消息框：





当输入了确实存在的学号或学生姓名后将会跳转到管理学生信息的子界面，该界面只可以通过这种方式进入，会直接在该界面上显示查询结果（即被查询学生的基本信息）

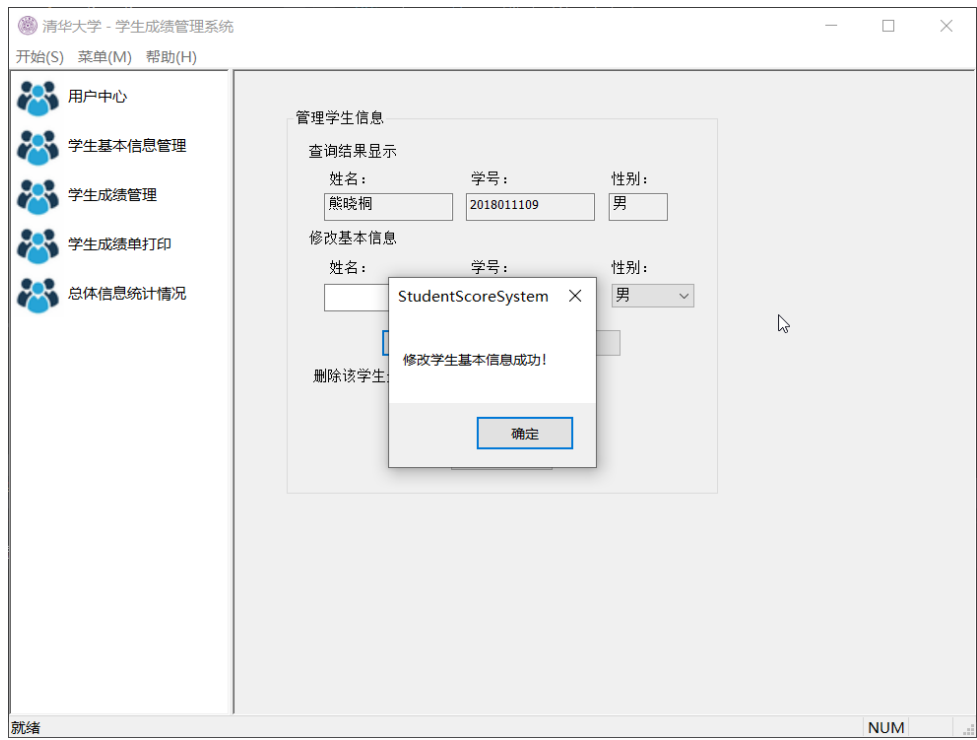


对此学生可以进行修改或删除的操作

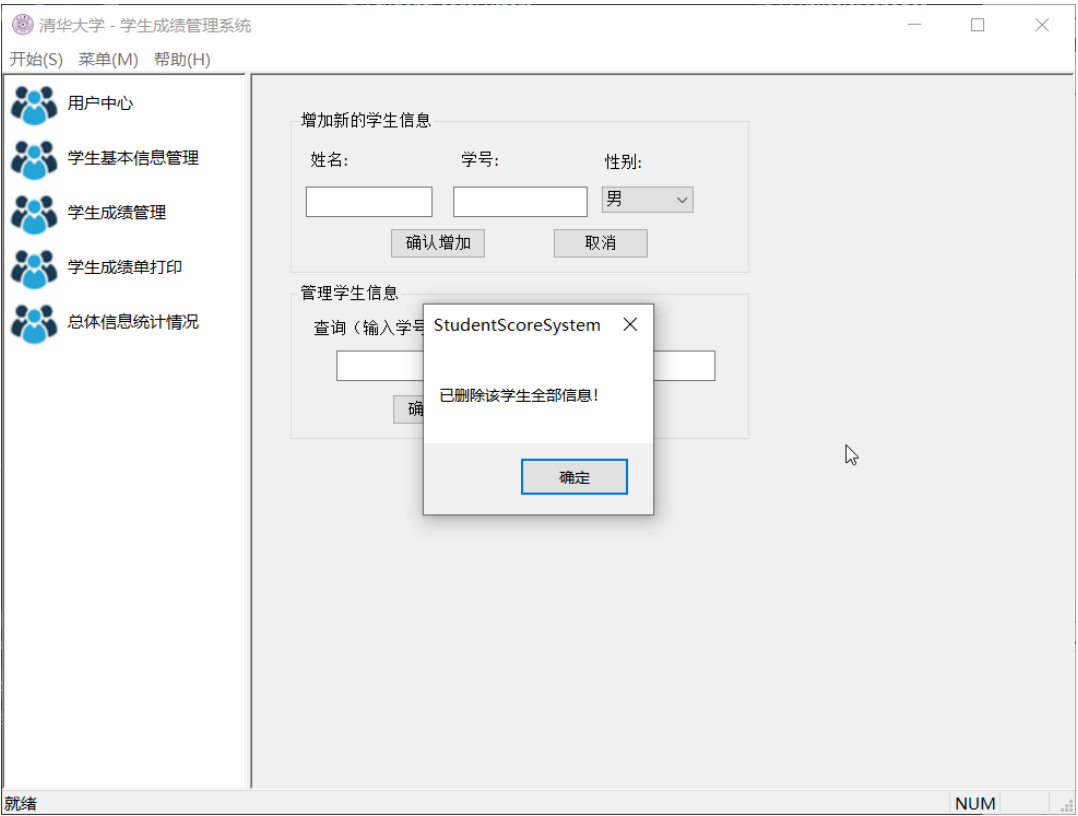
也可以直接按下“返回上一界面”的按钮直接回到父界面

同样，修改操作如果空着编辑框或者数据输入有问题也会同添加新成员一样报错，这里就不再重复操作和截图了。

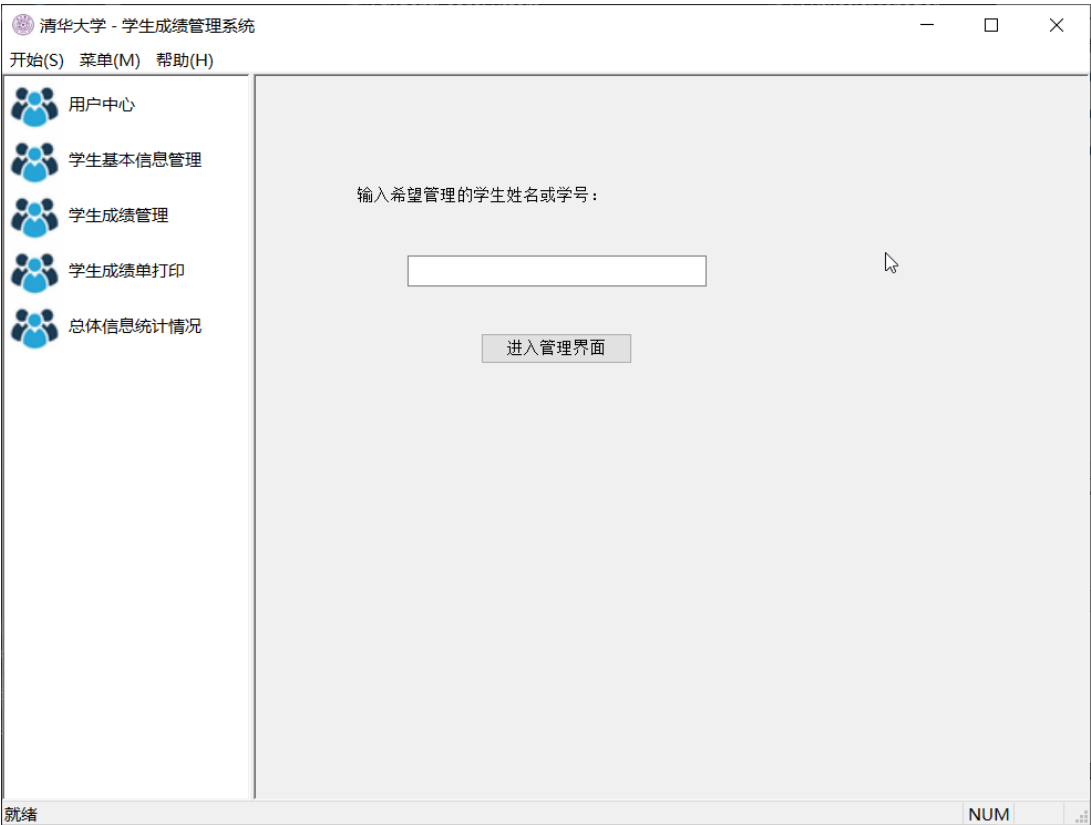
修改成功后则会有弹框提示，同时显示信息也会一并直接发生变化



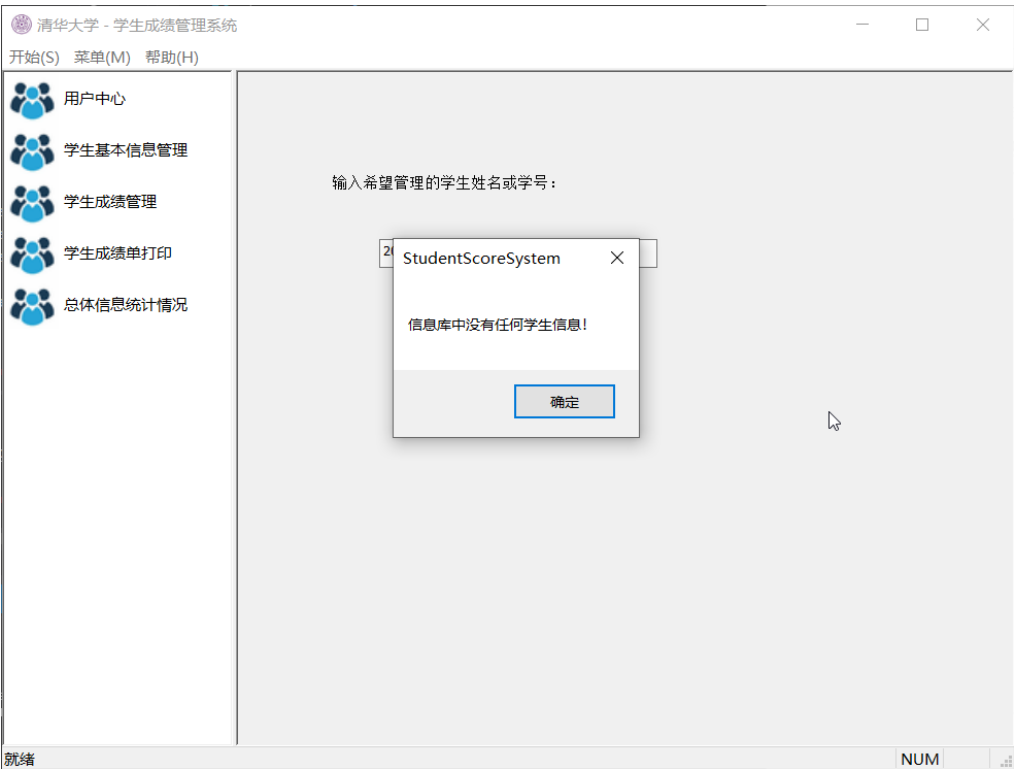
按下确认删除按钮，将可以删除该学生全部信息，同时弹出弹框告知，并直接返回上一界面



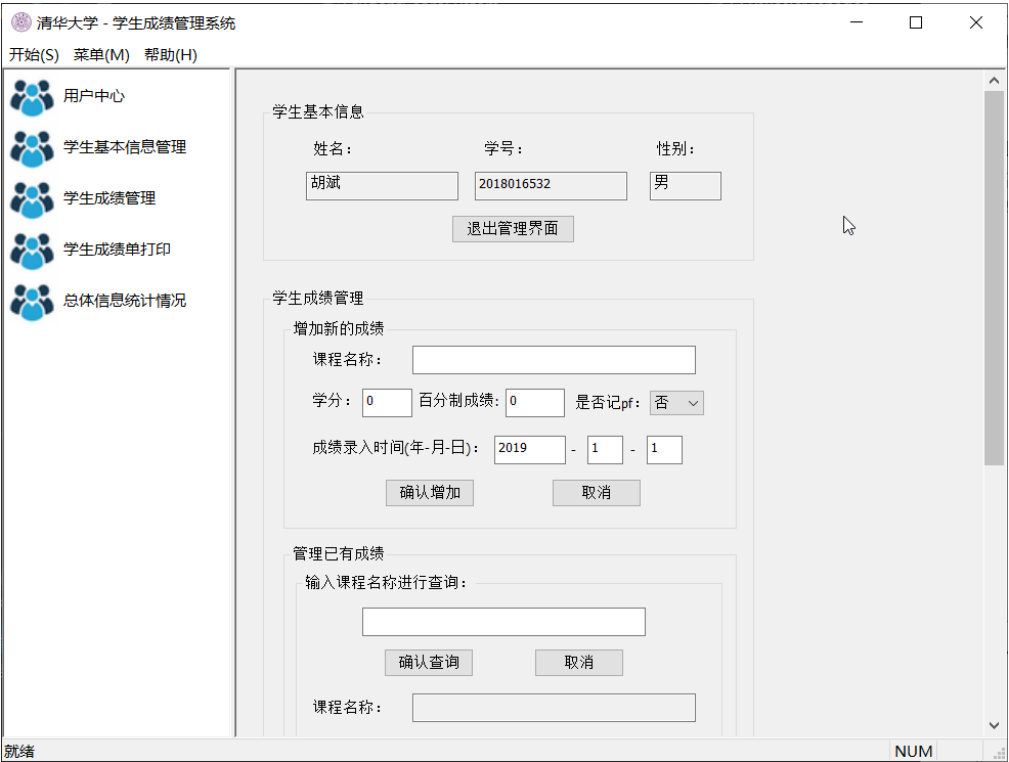
我们再次新建一个学生，用于之后的操作测试。进入学生成绩管理界面



若信息库内没有学生信息，则会提示：



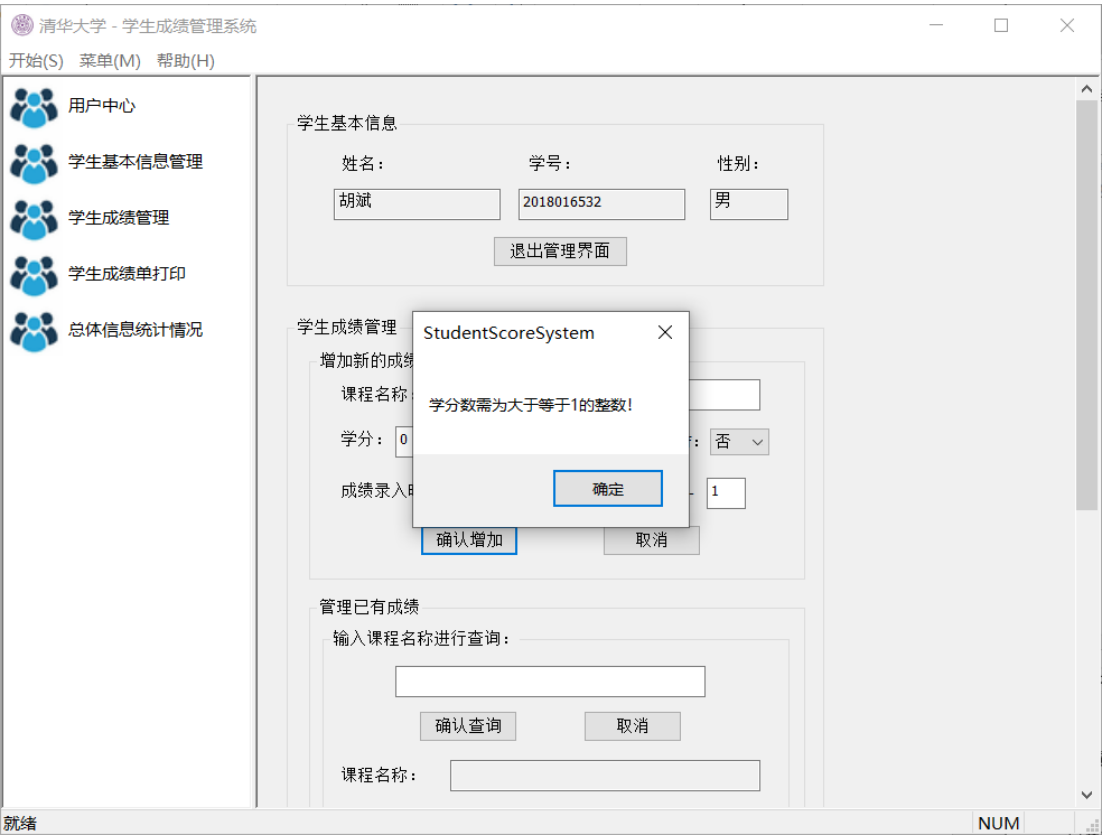
输入姓名或学号进入管理界面。该界面同样会在不可编辑的框中自动显示该学生的基本信息，同时可以按下退出管理界面按钮离开该界面。由于这个界面比较长，内容比较多，所以右侧做了一个滚动条，可以拖拽来进行界面的上下移动。同时可以看到，界面初始化的时候输入成绩的栏目中默认为：学分为0，成绩为0，记pf为否，成绩录入时间 2019-1-1。



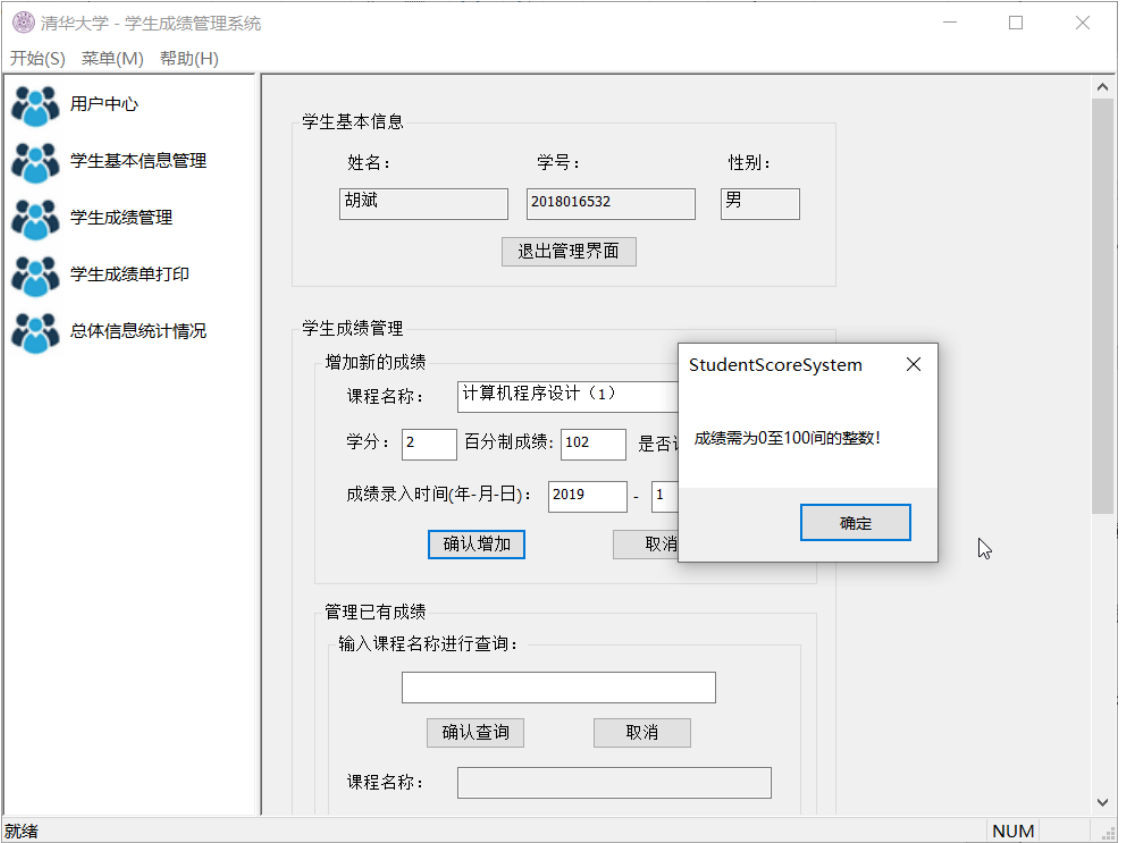
同样，如果课程名空着，会出现如下情况：



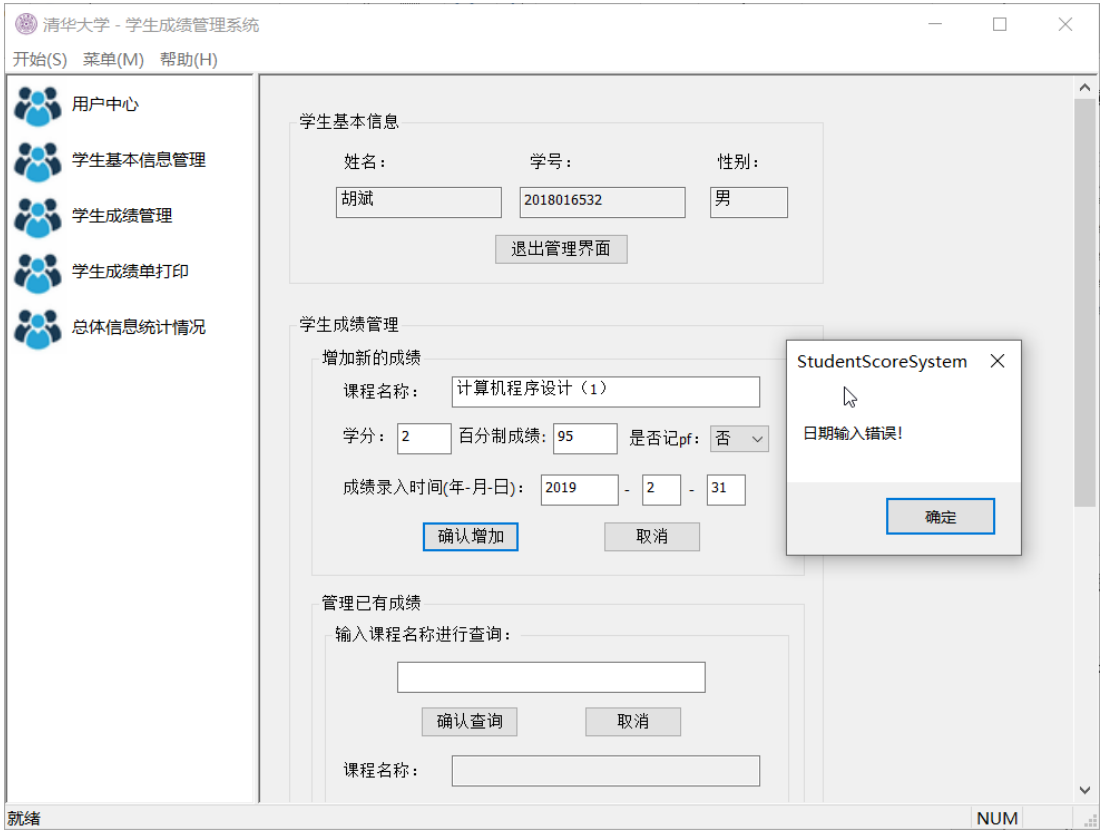
如果学分数输入 0 或负数，也会提示错误：



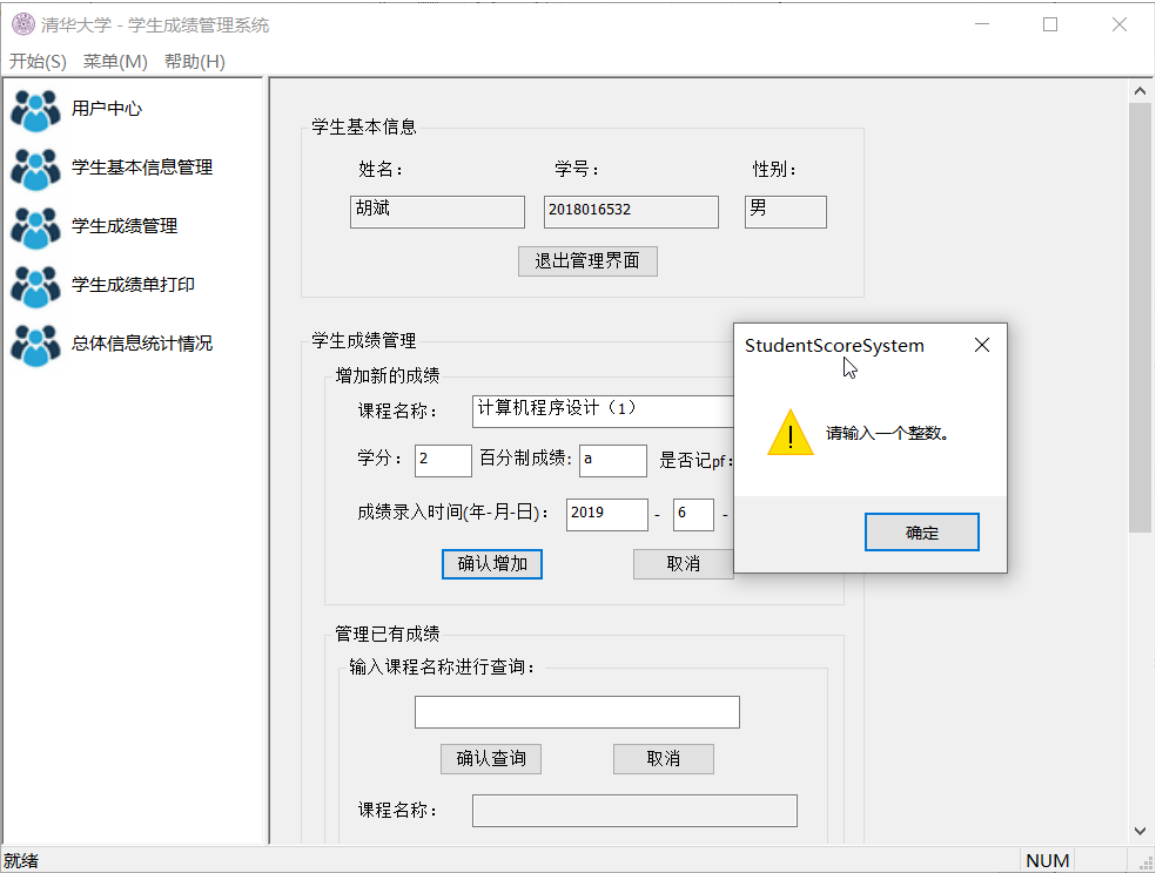
成绩不在 0-100 的区间内也会提示错误：



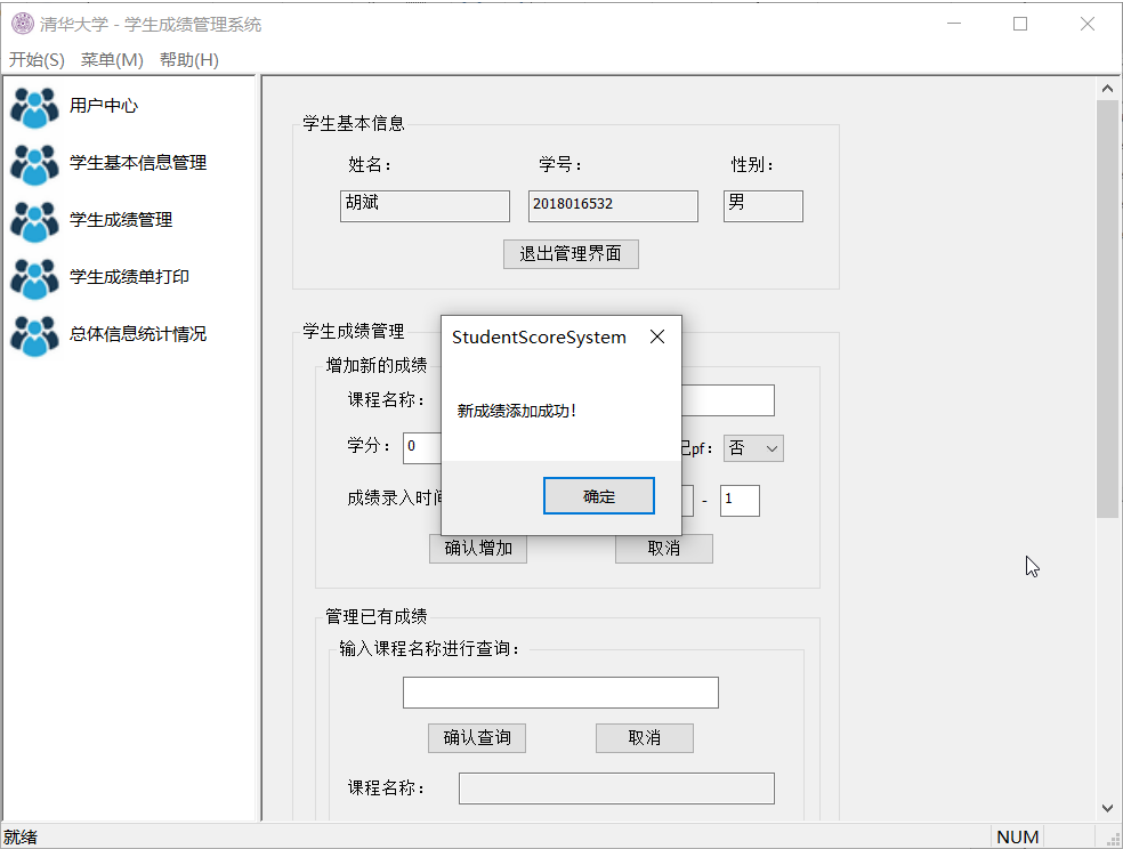
日期输入一个不存在的日子，也会提示错误：



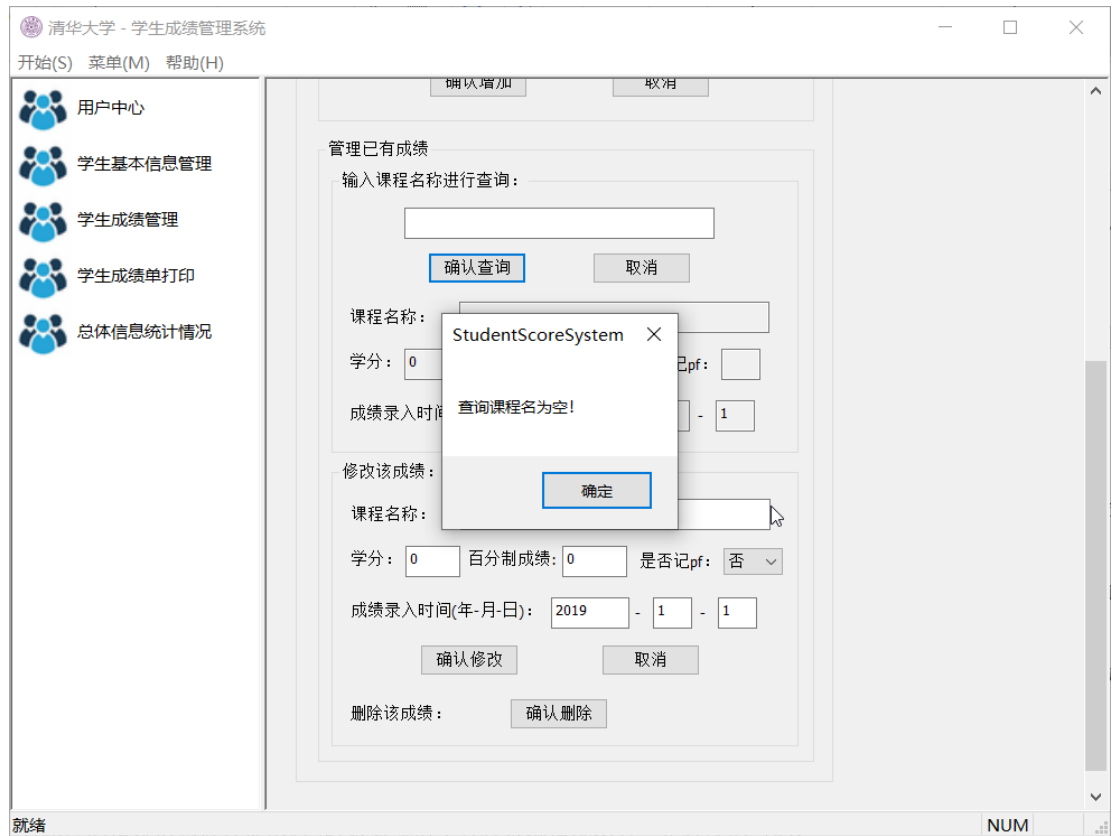
如果在学分或者成绩或者日期处输入整数之外的内容，也会报错：



成绩添加成功后会有弹框提示，并且会将各编辑框初始化：



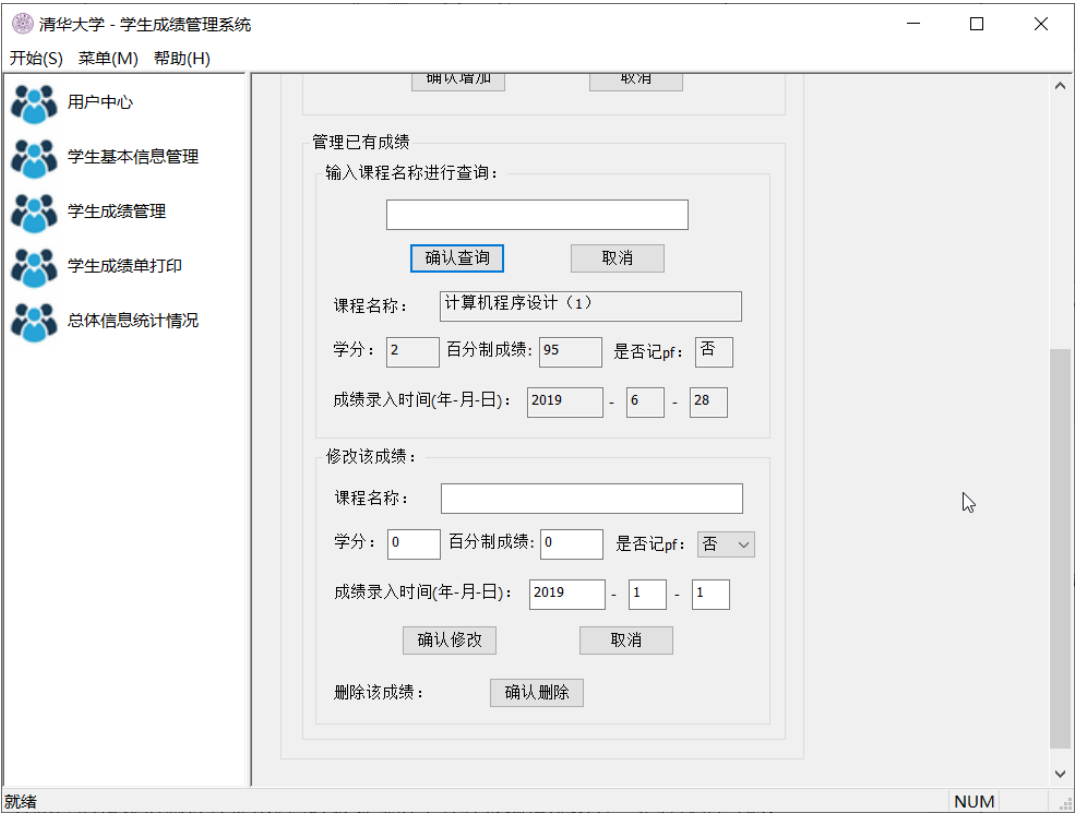
下拉滚动条，可以在管理框的查询部分输入已经录入的课程名进行查询，若输入空则弹框提示错误：



若输入不存在系统中的课程名则会弹框提示错误：

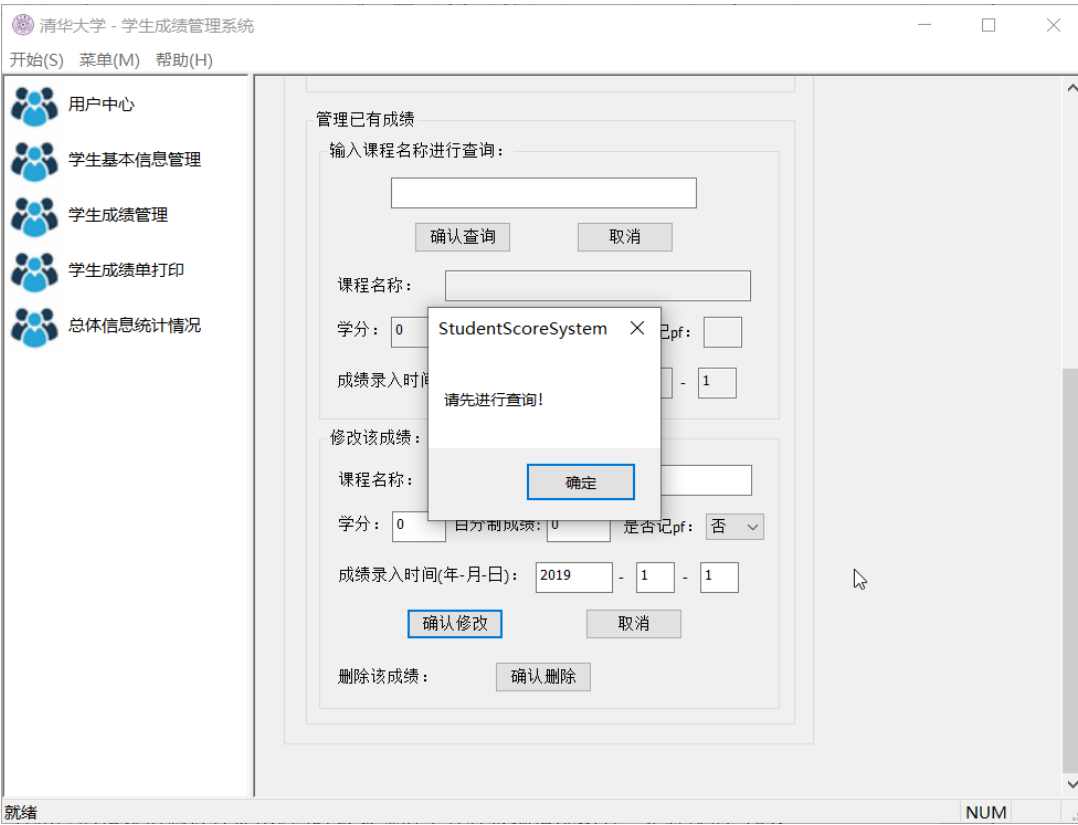


成功查询则会清空查询栏并在下方显示查询出的信息



若按取消，则会把显示窗口的信息初始化

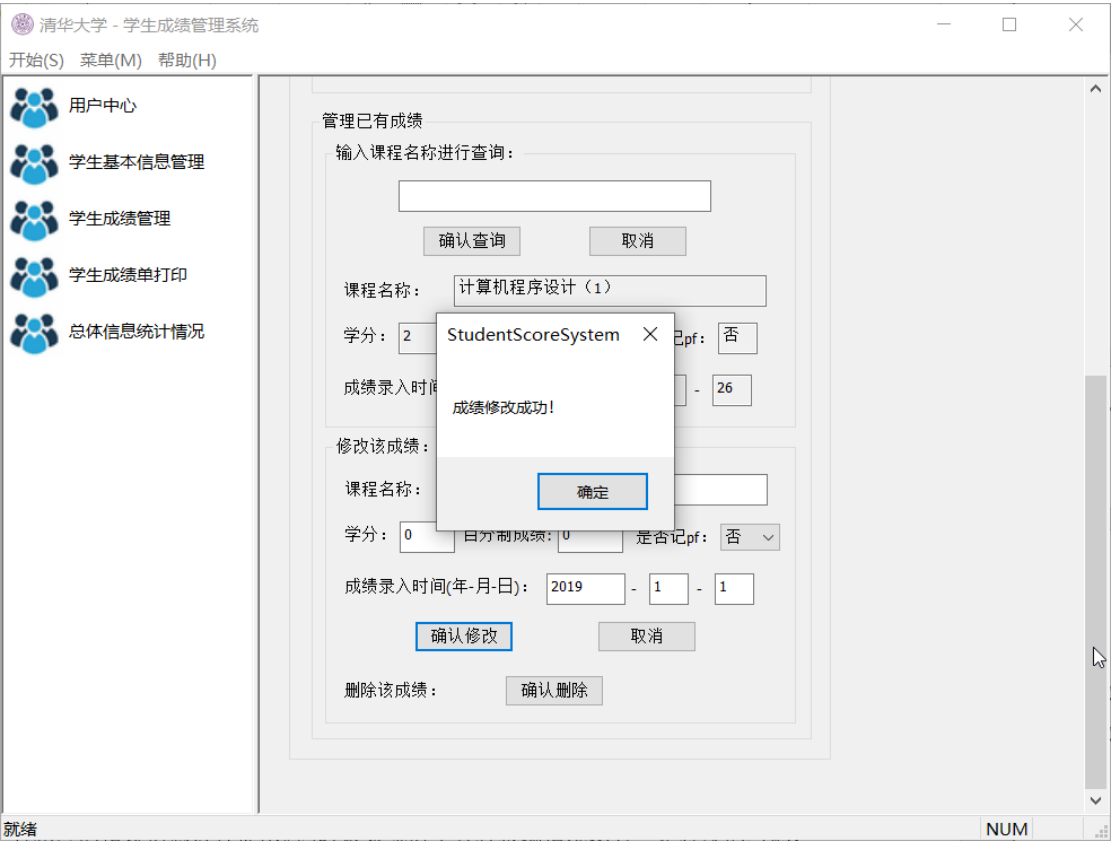
只有先进行查询后才可进行修改或删除操作，否则均会弹框如下：



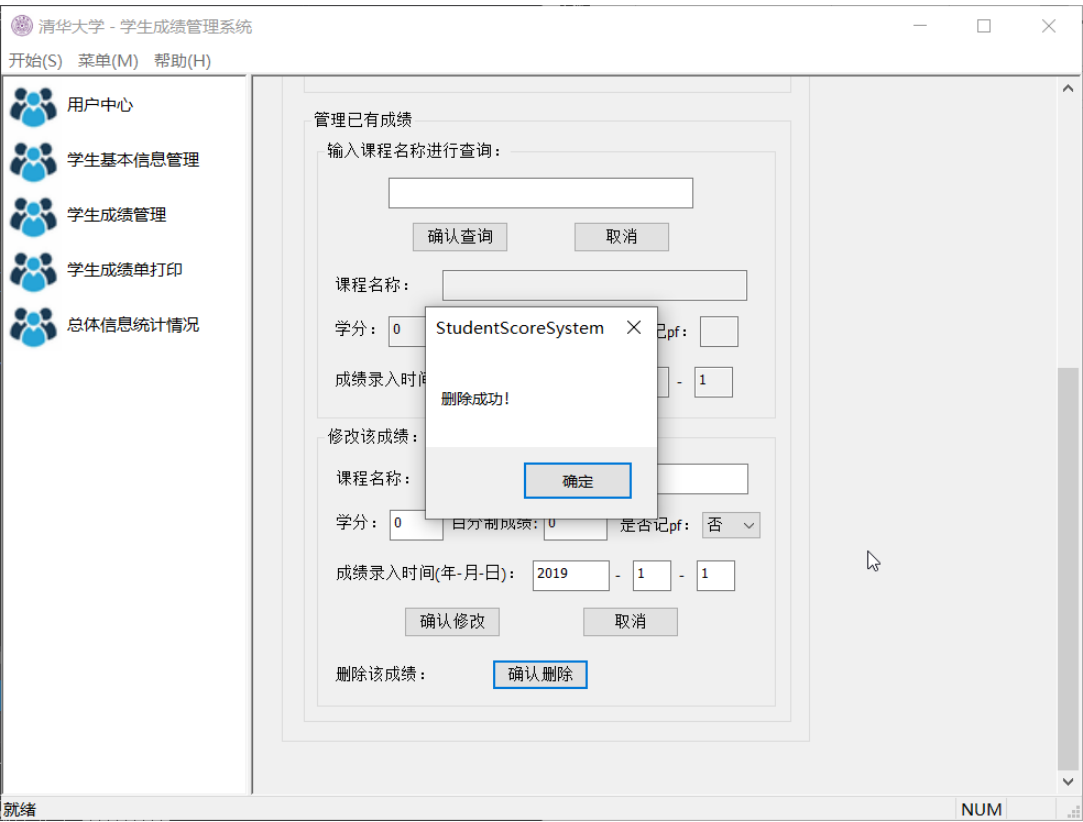


修改课程部分与增加新的成绩相似，取消会初始化修改功能中各编辑框，输入报错同上，此处就不再截图赘述了

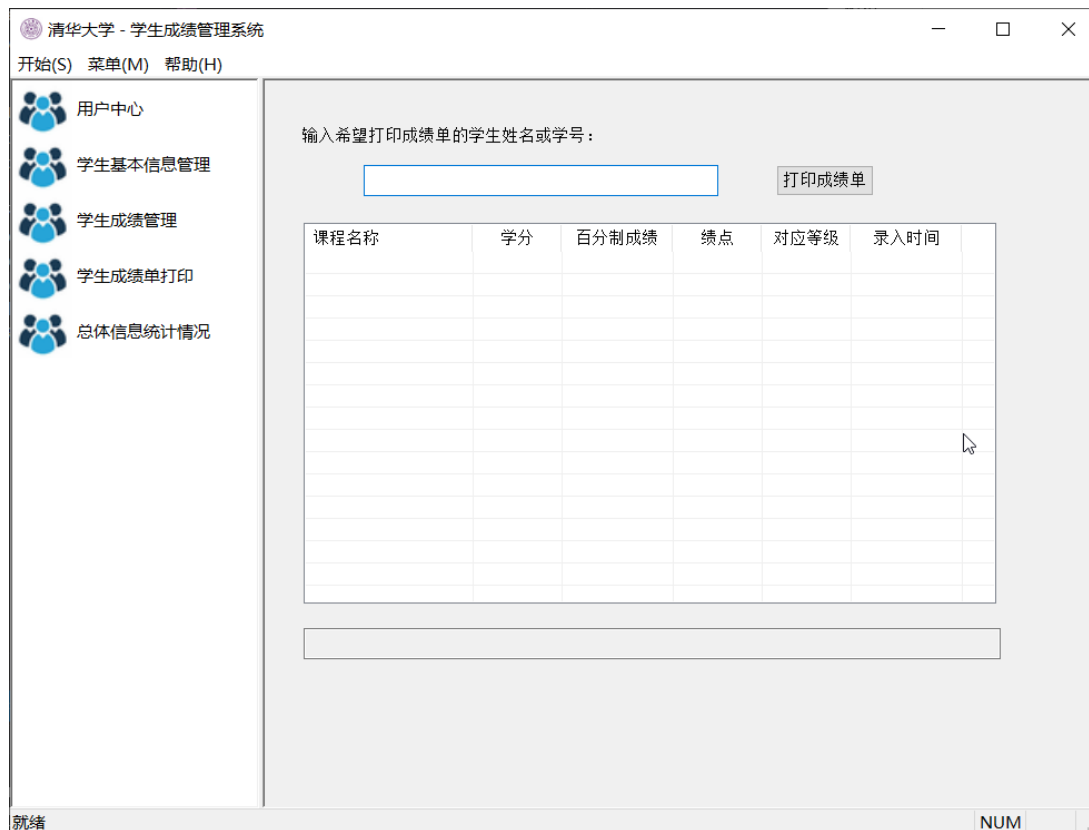
修改成功后同样会弹框提示：



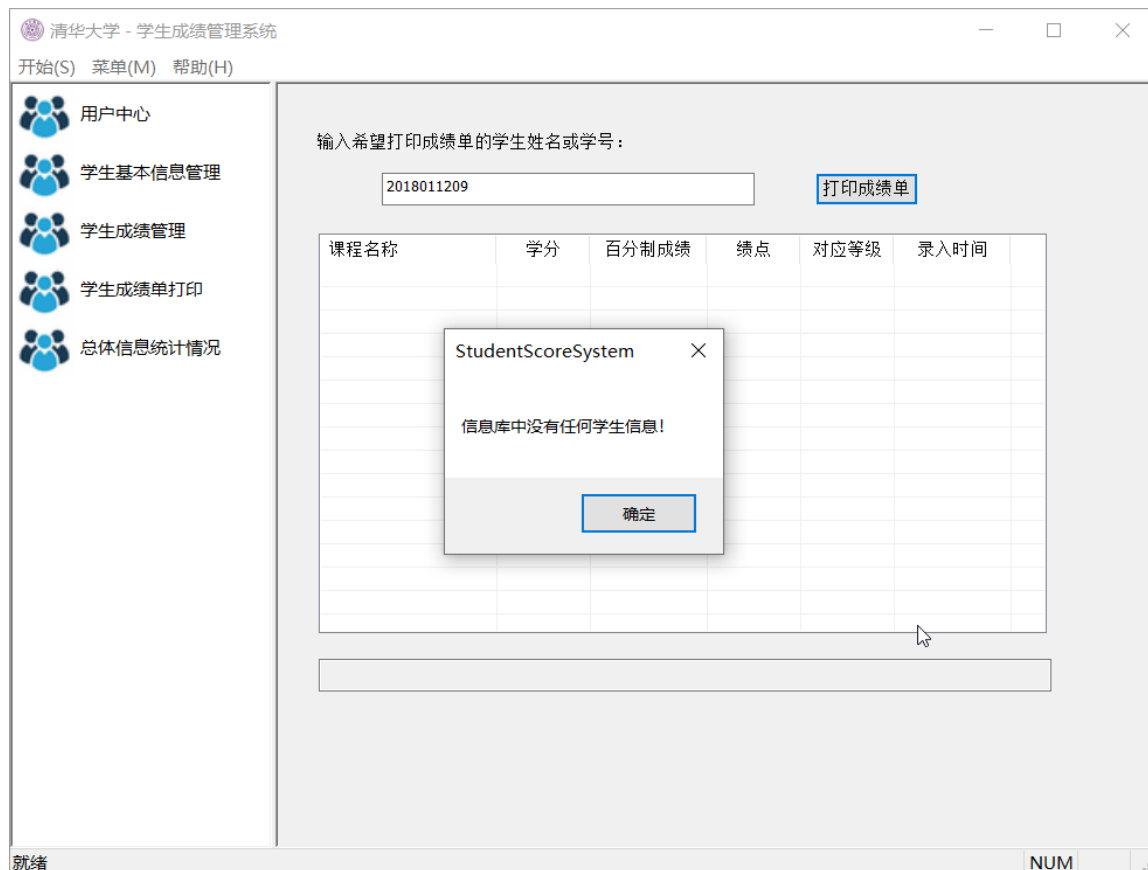
按下删除按钮后，将会初始化修改和显示部分的编辑框，并弹框提示修改成功



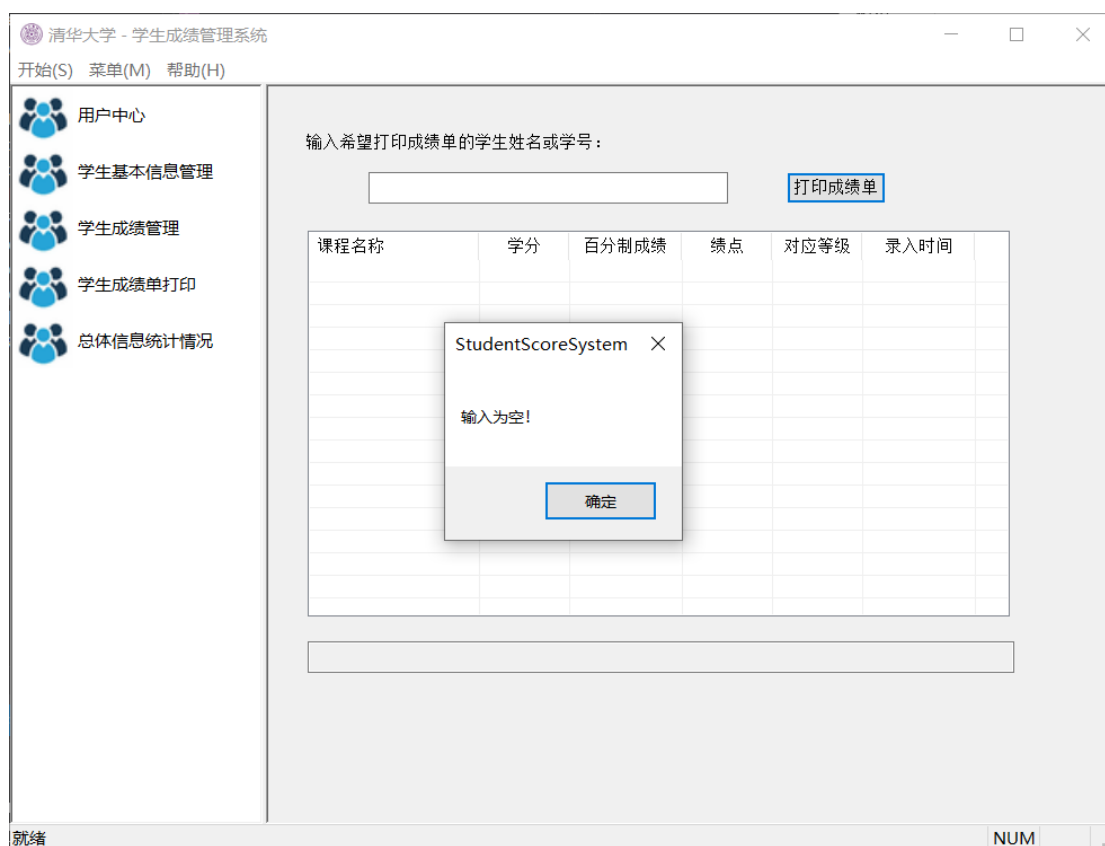
进入学生成绩单打印界面:



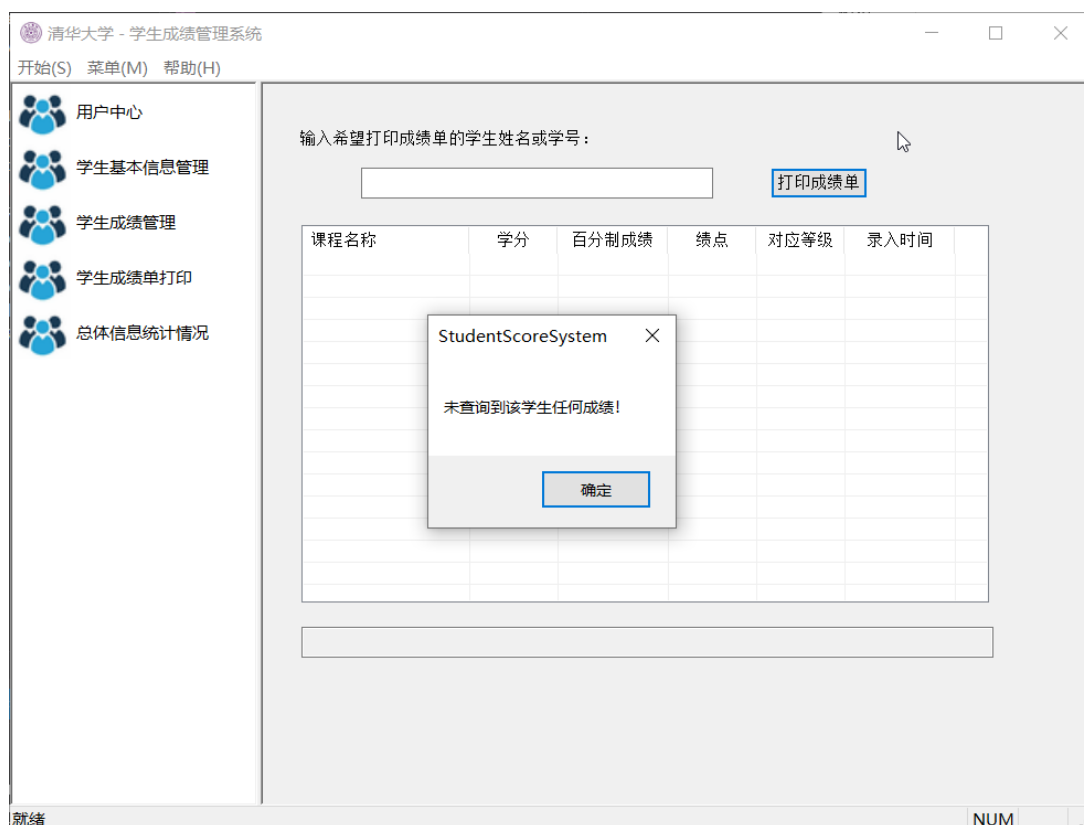
在上方输入希望打印成绩单的学生姓名或学号。若整个系统内没有学生成员，则弹框提示：



在上方输入希望打印成绩单的学生姓名或学号，若空则弹框提示



若输入人员没有成绩（即成绩均为空），选择弹框提示，同时清空查询框：








查询成功则会将其成绩进行排序，第一依据是成绩录入时间，第二依据为课程名称字典序  
如果其课程只有不能计算绩点的 pf 课程，则会显示为

[illegible]

若不只是记 pf 的课程，则会呈现如下，包括成绩信息、修习课程数、绩点等：

清华大学 - 学生成绩管理系统
— □ ×

开始(S)    菜单(M)    帮助(H)

-  用户中心
-  学生基本信息管理
-  学生成绩管理
-  学生成绩单打印
-  总体信息统计情况

输入希望打印成绩单的学生姓名或学号：

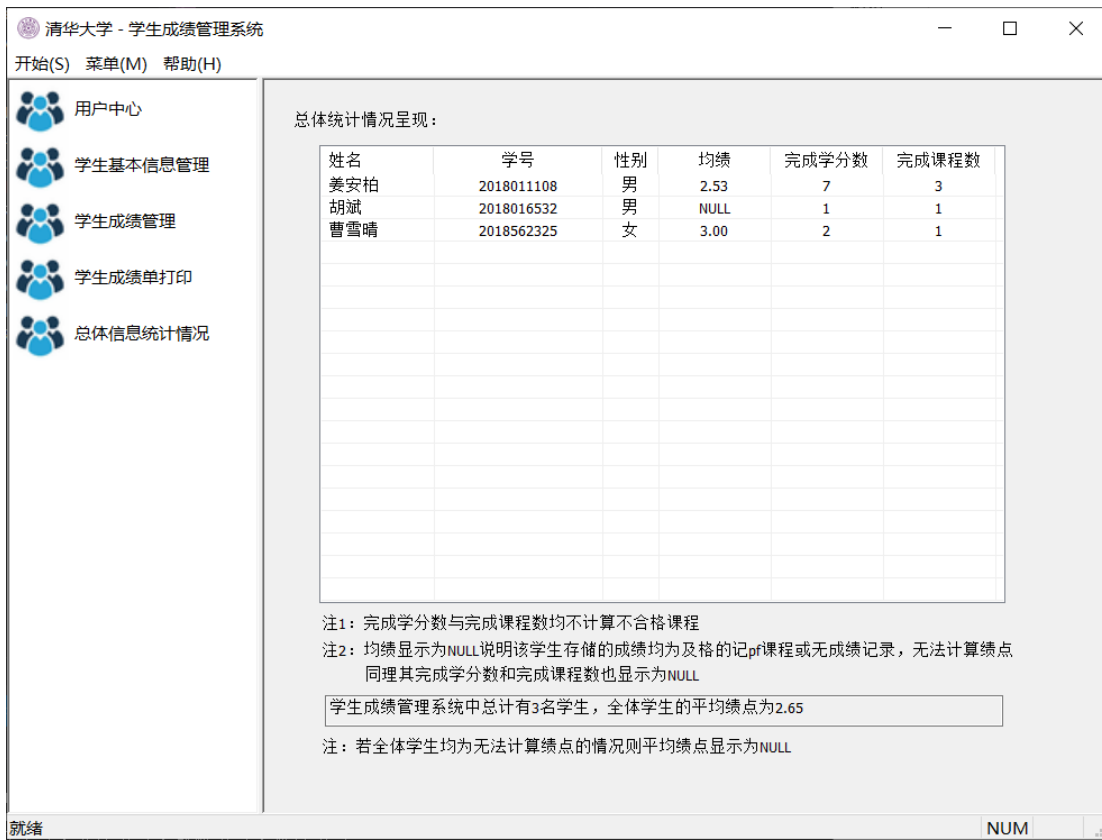
打印成绩单

课程名称	学分	百分制成绩	绩点 记pf	对应等级	录入时间
信息概论	1	60	3.0	P	2018-1-1
离散数学	2	83	3.0	B	2019-5-1
大学物理	4	76	2.3	C+	2019-6-3

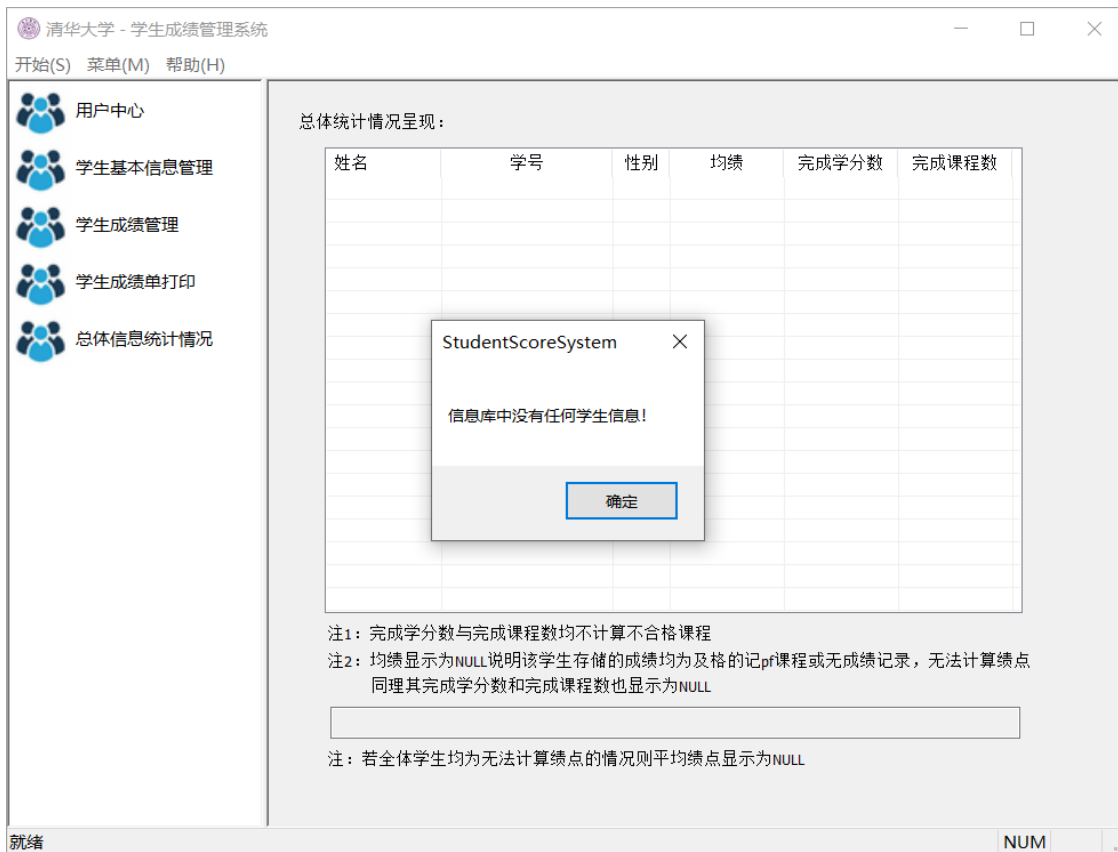
该学生共修3门课程，其中通过课程3门共7学分，平均绩点为2.53

就绪
NUM

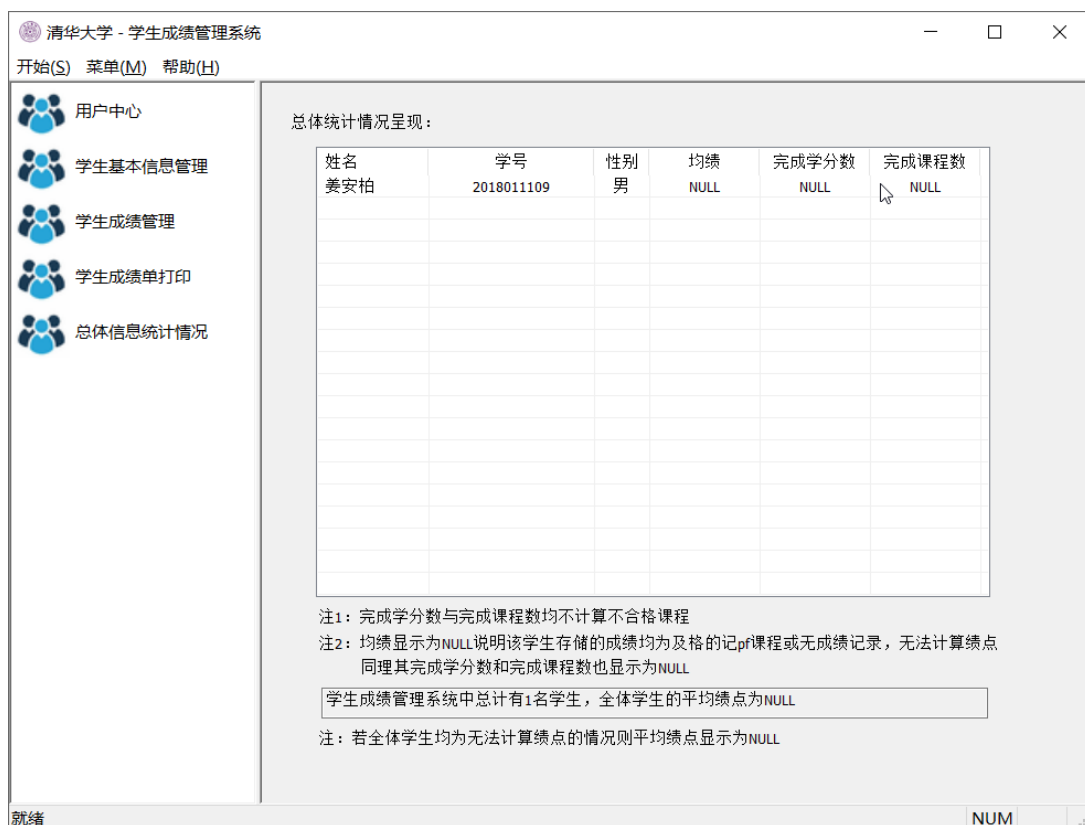
切换到总体信息统计情况直接呈现总体情况，同时下面有注释来说明一些显示内容的含义（即 NULL）



若系统内没有学生成员则会直接弹出一个弹框:



若系统内学生均无法计算绩点则显示如下



按右上角的×或开始选单中的退出按钮退出系统中,系统会将系统中的数据存入相应的数据文件中:

学生基本信息文件: basic.txt



学生成绩文件: record.txt



关闭系统后，将这两个文件作为测试数据文件再次测试。再次打开系统，我们就可以发现，储存在文件中的这些数据就已经在系统中了，可以直接进行操作，并且各项功能均可正常运行。实现了数据存储文件的自动读取。

## （2）测试结果分析

从上述测试与使用说明，我们可以看出，该学生成绩管理系统可以较好地完成学生信息管理以及学生成绩管理的相关功能，同时以图形界面呈现，相较于控制台程序而言使用起来更加贴近真实使用的系统，用户操作更加简单方便。同时该系统也具有一定的鲁棒性，有尚可的错误处理应对能力，可以较好地应对输入数据是否符合规范的问题，并针对不同情况给出不同的弹框进行提示，以便用户修正不合规范的信息内容。同时也考虑到了两张链表——学生链表和成绩链表空着的情况，并优先检查链表是否为空再去检查输入内容，同时弹出消息框提示用户各种情况。此外，设计了多个界面，并可以使用拆分窗口的左栏和菜单进行各界面之间的切换，方便了各功能之间的切换的调用，同时一些需要达到某个前提的功能（比如查询后才可修改或删除），设计成了子界面，只有在父界面达到了要求（比如查询信息正确），才会转入到子界面中，进行相关功能的操作，这样也有效控制了一个界面的规模和内容，使系统显示和操作都变得更简洁，也便于功能直接隔离开来，层次性明显。很多设计也是比较贴近实际，比如姓名可以允许相同，但是学号则要求必须与原有成员不相同，即学号是唯一标识。链表嵌套链表的设计使得各级的增删查改功能都易于实现，操作方便。具有文件操作，在挂载界面之前就自动读取了存储文件中的数据，并在退出系统时自动存储数据到文件中，这方便了系统的多次使用，符合实际操作需求。

系统基本符合要求，但是仍然存在一些缺陷，比如没有很好的应对存储文件遭到破坏的设计，如恶意篡改数据文件（比如让要求不能相同的学号出现相同）则会使得某些功能可能无法实现预想的要求（比如按学号查询则会查询到同样学号中更靠近表头的成员）。

## 6. 总结

该学生成绩管理系统较好地符合了课程设计对于该课题的各项要求，一共设计了 24 个类，我们把 MFC 类暂且放一旁，单考虑 9 个核心的类。

这 9 个类都有数个各自的属性和方法，完全满足类的数量和各自属性方法的数量要求。同时这 9 个类中有一个抽象基类，用于保留接口并将其保护起来，用学生链表类来继承它。这 9 个类中除了 1 个文件类是单独的以外，其他 8 个类都有继承或包含的层次关系，使得数据的结构比较清晰明了，同时具有成绩结点这个类是多重继承的派生类，它是由日期类的派生类成绩类再次派生的类。

为了让成绩打印为成绩单的时候，可以有一个排序，我对于其排序规则定义为先依据日期，早的在前，同日期的依据课程名称的字典序，所以，为了排序的函数方便书写，我为成绩结点类进行了<和>运算符重载，进行后面需要进行成绩结点的比较时，直接使用<和>即可，方便了程序的编写和阅读。由于搜索结点需要依据不同需求返回不同的值，即结点指针或者位置号，所以对于两个链表类的搜索函数都做了函数重载，改变了返回值和参量，方便了后续的使用与其他模块的调用，体现了静态多态性。

依据实际需求，写了文件类，把文件操作从各个核心的处理数据的类中割裂开来，可以读写用户名密码相关的配置文件和读写存储数据的数据文件，使得学生成绩管理系统的实用性更高。考虑到学生成绩管理系统内部数据信息的情况，使用了链表的数据结构，并且学生链表的每个结点下都有一个记录个人成绩的成绩链表，这样使得数据的结构层次非常清晰，方便了写入和读取以及各种传值处理的操作。

这一次大作业核心部分的编写由于大量用到链表，让我对于链表和指针的操作、调用等掌握的更加熟悉深入了，可以较为轻松的编写一些相关的函数和进行一些设计。

这一次的大作业的编写过程也可以说是有一定的模块化的设计与实现的想法。一开始编写了核心类中 8 个负责记录存储数据、处理数据等操作的类，这些类只与数据打交道，而且并不在这些类构造或者是改变值的时候进行数据检查，把这些内容都放在了这八个类构成的体系的外部，也就是图形界面的体系中。相当于图形界面相关的类读取输入值并判断是否符合要求，符合则传入这 8 个类构成的核心数据体系中，并从其中取得相应所需要的输出数值，转化输出。

为了让学生成绩管理系统使用起来更加贴近现实并且更加简单方便，我学习了图形界面的相关技术。由于核心程序是采用 C++ 编写的，所以图形界面的相关技术我就选择了 MFC 来学习。最终不光把系统做成了分界面可视化的，更是做了菜单栏、图标等使得系统更加精致美观的一些设计实现。

在这段时间 MFC 的学习中，确实收获了很多。我深入学习了一些较为底层的相关知识，也练习了一下使用 C 语言来进行底层窗口实现，也学习了消息映射机制，确实让我对于图形界面编程的相关内容有了更加深刻的认识和理解。基于 MFC 的各种控件以及各种界面挂载、拆分窗口、自定义消息处理的学习更是让我的系统得到了进一步的升级，变得更加美观而实用，也让一些设计的实现难



度下降了，同时也让使用者的操作变得简便了——比如起初基于控制台的应用程序设计下，输入性别需要键入英文字符或者中文文字，但是基于 MFC 的图形界面只需要用一个下拉选单，直接选择男或女即可，这样一来使用者方便，二来程序应对错误的处理也将会变得简便，性别处无需识别非法字符，只消读取所选栏目即可。而且图形界面可以把很多功能都集成在一个界面上，不像控制台进入了某个功能你就只能完成这个功能或者取消这个功能以回到上一级的功能选单，图形界面可以把例如查、删、改等功能放在一个界面上，你随时想要调用哪个功能直接完成对应内容的输入后按按钮即可，使用者的自由性也更强了。

此外，MFC 的很多内容也让我自己有了一些设计上的启发，比如之前提到过的判断链表为空的 `IsEmpty()` 函数，就是受 `CString` 类同名函数启发。我之前编写程序都是习惯于在一个 `.cpp` 文件中编写的，把所有类都放在一起，但是自从使用了 MFC 之后，我就把内容分成 `.h` 文件以及 `.cpp` 文件，一个做声明一个做实现，同时把 8 个数据相关的类放在一起，文件操作又单独写成一个类，较好地体现了类的封装性。这么多行的大程序的编写确实也锻炼了我写代码和 `debug` 的能力，熟练地在多个地方插入断点，观察变量的值，以慢慢缩小范围从而锁定造成问题的原因到底出在哪里，然后再解决它，同时也在不断实现新功能的过程中发现一些原有的问题——比如新的功能需要传递某个值进来，但是原先并没有设计，因而就增添上相应函数，这样不断补足完善，直到整个程序最终完成，较好地实现了学生成绩管理系统。

## 附录：源程序清单

（由于源代码较多，所以将其按照设计分成核心的 9 个类和 MFC 图形界面的 15 个类呈现）

项目文件放置在清华网盘如下地址中，若助教有需要也可以查看：

<https://cloud.tsinghua.edu.cn/d/0dade93bce6447f2b0f0/>

### 核心的 9 个类相关的一一对应的头文件和源文件：

头文件：head.h

```
#pragma once
#include<iostream>
#include<iomanip>
using namespace std;
void score_to_grade(int a, double &b, char *s, int c);
double score_to_grade(int a);
bool daywrong(int year, int month, int day);
class AbsSystem //抽象类，保留接口
{
public:
    virtual double averageGPA() = 0;
    virtual int AddToHead_STU(char stuname[], char stugender[], char stuid[])=0;
    virtual int Delete_STU(int i)=0;
    virtual bool IsEmpty()=0;
    virtual void SetTemp(int t) = 0;
    virtual int ShowTemp() = 0;
    virtual int showtotalnumber() = 0;
};
class Date
{
private:
    int Year, Month, Day; //记录该成绩录入的时间
public:
    Date(int a, int b, int c);
    ~Date();
    void SetDate(int a, int b, int c);
    int ShowDay();
    int ShowMonth();
    int ShowYear();
};
class Score : public Date
{
private:
    char subject[50]; //科目名称
    int score; //总评百分制成绩
    int credit; //学分
    double grade1; //绩点
```

```

    char grade2[3];    //对应等级
    int pflag;         //是否记pf的标志,0指记pf
public:
    Score(char *a,int b, int c, int d, int e, int f, int pf);
    ~Score();
    void SetScore(char *a, int b, int c, int d, int e, int f, int pf);
    char *ShowSubject();
    int ShowCredit();
    double ShowGrade1();
    int ShowPF();
    int ShowScore();
    char *ShowGrade2();
};
/*公有继承日期类（即标明本成绩的录入时间），有科目名称、总评百分制成绩、学分、绩点、对应等级等
私有成员，
并有相关构造、析构、设置、打印、传值等成员函数，考虑重载<>运算符； */
class ScoreList;
class ScoreNode : public Score
{
private:
    ScoreNode *next;
public:
    ScoreNode(char *a, int b, int c, int d, int e, int f, int pf, ScoreNode *next);
    ~ScoreNode();
    bool operator > (ScoreNode &sn);    //运算符重载
    bool operator < (ScoreNode &sn);
    friend class ScoreList;
};
class ScoreList
{
private:
    ScoreNode *head, *current;
    int TotalCourse;    //记录总科目数
    int TotalCredit;    //记录总学分数
public:
    ScoreList();
    ~ScoreList();
    int AddToHead(char *a,int b, int c, int d, int e, int f, int pf);
    int Delete(int i);
    ScoreNode *Search(char *str);
    int Search(char *str, int i);    //重载搜索函数
    void ScoreSort();    //排序函数
    bool IsEmpty();
    int coursenum();
    ScoreNode *Search(int num);

```

```

        int PassCredit();           //统计通过的学分总数
        int PassCourse();          //统计通过的课程总数
};

class StuInfo
{
private:
    char Name[30];                 //学生姓名
    char Gender[5];                //性别
    char StuID[20];                //学号
    int EffCredit;                 //去除及格的pf课学分剩下的学分，用于计算学分绩
    double EffGrade;              //去除及格的pf课外剩余课程的绩点与学分乘积和，用于计算学分绩
    ScoreList SL;
public:
    StuInfo(char stuname[], char stugender[], char stuid[]);
    ~StuInfo();
    void SetBasicInfo(char stuname[], char stugender[], char stuid[]);
    void AddScore(char *a, int b, int c, int d, int e, int f, int pf);
    ScoreNode *SearchScore(char *str);
    int SearchScore(char *str, int i);
    void DeleteScore(char *str);
    double personGPA();           //个人均绩
    double ShowEffGrade();
    int ShowEffCredit();
    char * ShowName();
    char * ShowStuID();
    char * ShowGender();
    bool IsEmpty();
    ScoreList showscorelist();
};

class StuInfoList;
class StuInfoNode :public StuInfo
{
private:
    StuInfoNode *next;
public:
    StuInfoNode(char stuname[], char stugender[], char stuid[], StuInfoNode *next);
    ~StuInfoNode();
    friend class StuInfoList;
};

class StuInfoList :public AbsSystem
{
private:
    StuInfoNode *head, *current;
    int TotalNumber; //成员总数（即学生数量）
};

```

```

    double totalGPA;    //记录总的学分绩点积之和, 用于计算全体学生的平均绩点
    int totalCredit;    //记录总的学分之和, 用于计算全体学生的平均绩点
    int temp;           //为界面设计留一个用于界面切换时缓存的重要参数的成员
public:
    StuInfoList();
    ~StuInfoList();
    double averageGPA();
    int AddToHead_STU(char stuname[], char stugender[], char stuid[]);
    int Delete_STU(int i);
    StuInfoNode *Search_STU(char *str);
    int Search_STU(char *str, int i);    //重载搜索函数
    bool IsEmpty();
    void SetTemp(int t);
    StuInfoNode *FindNode(int t);    //依据位置号找到对应结点
    int ShowTemp();
    int showtotalnumber();
};

```

**源文件: source.cpp**

```

#include "stdafx.h"
#include "head.h"

using namespace std;
void score_to_grade(int a, double &b, char *s, int c)    //原始百分制成绩到绩点和对应等级的转换函数
{
    if (c == 1) {
        if (a < 60) { b = 0.0; strcpy(s, "F"); }
        else if (a < 63) { b = 1.0; strcpy(s, "D"); }
        else if (a < 67) { b = 1.3; strcpy(s, "D+"); }
        else if (a < 70) { b = 1.7; strcpy(s, "C-"); }
        else if (a < 73) { b = 2.0; strcpy(s, "C"); }
        else if (a < 77) { b = 2.3; strcpy(s, "C+"); }
        else if (a < 80) { b = 2.7; strcpy(s, "B-"); }
        else if (a < 85) { b = 3.0; strcpy(s, "B"); }
        else if (a < 90) { b = 3.3; strcpy(s, "B+"); }
        else if (a < 95) { b = 3.7; strcpy(s, "A-"); }
        else if (a < 100) { b = 4.0; strcpy(s, "A"); }
        else { b = 4.0; strcpy(s, "A"); }
    }
    else {
        if (a < 60) { b = 0.0; strcpy(s, "F"); }
        else { b = -1; strcpy(s, "P"); }
    }
}

```

```

double score_to_grade(int a)
{
    if (a < 60) return 0.0;
    else if (a < 63) return 1.0;
    else if (a < 67) return 1.3;
    else if (a < 70) return 1.7;
    else if (a < 73) return 2.0;
    else if (a < 77) return 2.3;
    else if (a < 80) return 2.7;
    else if (a < 85) return 3.0;
    else if (a < 90) return 3.3;
    else if (a < 95) return 3.7;
    else return 4.0;
}

bool daywrong(int year, int month, int day) //判断日期是否正确的函数
{
    int d;
    switch (month)
    {
        case 1::
        case 3::
        case 5::
        case 7::
        case 8::
        case 10::
        case 12: d = 31; break;
        case 4::
        case 6::
        case 9::
        case 11: d = 30; break;
        case 2: {
            if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) d = 29;
            else d = 28;
        } break;
    }
    if (day < 1 || day > d) return TRUE;
    else return FALSE;
}

Date::Date(int a, int b, int c) //把数据合理性检查放在输入该数据的地方了
{
    Year = a;
    Month = b;
    Day = c;
}

```

```

Date::~Date() {};
void Date::SetDate(int a, int b, int c)
{
    Year = a;
    Month = b;
    Day = c;
}
int Date::ShowDay()
{
    return Day;
}
int Date::ShowMonth()
{
    return Month;
}
int Date::ShowYear()
{
    return Year;
}

Score::Score(char *a, int b, int c, int d, int e, int f, int pf) :Date(d, e, f) //把数据
合理性检查放在输入该数据的地方了
{
    strcpy(subject, a);
    score = b;
    credit = c;
    pflag = pf;
    score_to_grade(b, grade1, grade2, pf);
}
Score::~Score() {};
void Score::SetScore(char *a, int b, int c, int d, int e, int f, int pf)
{
    Date::SetDate(d, e, f);
    strcpy(subject, a);
    score = b;
    credit = c;
    pflag = pf;
    score_to_grade(b, grade1, grade2, pf);
}
char *Score::ShowSubject()
{
    return subject;
}
int Score::ShowCredit()
{

```

```

        return credit;
    }
    double Score::ShowGrade1()
    {
        return grade1;
    }
    int Score::ShowPF()
    {
        return pflag;
    }
    int Score::ShowScore()
    {
        return score;
    }
    char *Score::ShowGrade2()
    {
        return grade2;
    }

ScoreNode::ScoreNode(char *a, int b, int c, int d, int e, int f, int pf, ScoreNode
*next) :Score(a, b, c, d, e, f, pf)
{
    this->next = next;
}
ScoreNode::~ScoreNode() {};
bool ScoreNode::operator > (ScoreNode &sn)
{
    if (this->ShowYear() != sn.ShowYear())
        return(this->ShowYear() > sn.ShowYear());
    else
    {
        if (this->ShowMonth() != sn.ShowMonth())
            return(this->ShowMonth() > sn.ShowMonth());
        else
        {
            if (this->ShowDay() != sn.ShowDay())
                return(this->ShowDay() > sn.ShowDay());
            else
            {
                if (strcmp(this->ShowSubject(), sn.ShowSubject()) >= 0) return true;
                else return false;
            }
        }
    }
}
}

```



```

bool ScoreNode::operator < (ScoreNode &sn)
{
    if (this->ShowYear() != sn.ShowYear())
        return(this->ShowYear() < sn.ShowYear());
    else
    {
        if (this->ShowMonth() != sn.ShowMonth())
            return(this->ShowMonth() < sn.ShowMonth());
        else
        {
            if (this->ShowDay() != sn.ShowDay())
                return(this->ShowDay() < sn.ShowDay());
            else
            {
                if (strcmp(this->ShowSubject(), sn.ShowSubject()) < 0) return true;
                else return false;
            }
        }
    }
}

```

```

ScoreList::ScoreList()
{
    head = current = NULL;
    this->TotalCourse = 0;
    this->TotalCredit = 0;
}

ScoreList::~~ScoreList() {}

int ScoreList::AddToHead(char *a, int b, int c, int d, int e, int f, int pf)
{
    ScoreNode *p = current;
    current = head;
    ScoreNode *newSN = new ScoreNode(a, b, c, d, e, f, pf, NULL);
    newSN->next = head;
    head = current = newSN;
    this->TotalCourse += 1;
    this->TotalCredit += c;
    return 1;
}

int ScoreList::Delete(int i)
{
    ScoreNode *p, *q;
    if (i == 0) {
        q = head;

```

```

        current = head = head->next;
    }
    else {
        p = current;
        current = head;
        for (int m = 0; m < i - 1; m++)
        {
            if (current == NULL)break;
            else current = current->next;
        }
        if (current == NULL || current->next == NULL)
        {
            current = p;
            return 0;
        }
        else
        {
            q = current->next;
            current = current->next = q->next;
        }
    }
    this->TotalCourse -= 1;
    this->TotalCredit -= q->ShowCredit();
    delete q;
    return 1;
}

ScoreNode *ScoreList::Search(char *str)
{
    ScoreNode *p = head;
    if (p == NULL)
    {
        return NULL;
    }
    else
    {
        while (p!= NULL)
        {
            if (strcmp(str, p->ShowSubject()) == 0) return p;
            p = p->next;
        }
        return NULL;
    }
}

int ScoreList::Search(char *str, int i)    //多加一个参数便于重载，也便于编写代码和调用
{

```

```

ScoreNode *p = head;
int j = 0;
if (p == NULL)
{
    return -1;
}
else
{
    while (p!= NULL)
    {
        if (strcmp(str, p->ShowSubject()) == 0) return j;
        j++;
        p = p->next;
    }
    return -1;
}
}

void ScoreList::ScoreSort()
{
    int n = this->TotalCourse;
    for (int i = 0; i < n ; i++)
    {
        ScoreNode *p = head;
        for (int k = 0; k < n-i ; k++)
        {
            if (p->next == NULL) continue;
            if (*p > *(p->next))
            {
                char *a1 = new char[50];
                char *a2 = new char[50];
                strcpy(a1, p->next->ShowSubject());
                strcpy(a2, p->ShowSubject());
                int b1 = p->next->ShowScore();
                int c1 = p->next->ShowCredit();
                int d1 = p->next->ShowYear();
                int e1 = p->next->ShowMonth();
                int f1 = p->next->ShowDay();
                int pf1 = p->next->ShowPF();
                int b2 = p->ShowScore();
                int c2 = p->ShowCredit();
                int d2 = p->ShowYear();
                int e2 = p->ShowMonth();
                int f2 = p->ShowDay();
                int pf2 = p->ShowPF();
                p->SetScore(a1, b1, c1, d1, e1, f1, pf1);
            }
        }
    }
}

```

```

        (p->next)->SetScore(a2, b2, c2, d2, e2, f2, pf2);
        delete[]a1;
        delete[]a2;
    }
    p = p->next;
}
}
}
bool ScoreList::IsEmpty()
{
    if (head == NULL) return TRUE;
    else return FALSE;
}
int ScoreList::coursenumber()
{
    return this->TotalCourse;
}
ScoreNode *ScoreList::Search(int num)
{
    ScoreNode *p = head;
    for (int i = 0; i < num; i++)
        p = p->next;
    return p;
}
int ScoreList::PassCredit()
{
    ScoreNode *p = head;
    int pc=0;
    for (int i = 0; i < TotalCourse; i++)
    {
        if (p->ShowScore() >= 60) pc += p->ShowCredit();
        p = p->next;
    }
    return pc;
}
int ScoreList::PassCourse()
{
    ScoreNode *p = head;
    int pc = 0;
    for (int i = 0; i < TotalCourse; i++)
    {
        if (p->ShowScore() >= 60) pc += 1;
        p = p->next;
    }
    return pc;
}

```

```

}

StuInfo::StuInfo(char stuname[], char stugender[], char stuid[])
{
    strcpy(Name, stuname);
    strcpy(Gender, stugender);
    strcpy(StuID, stuid);
    EffCredit = 0;
    EffGrade = 0;
}

StuInfo::~StuInfo() {};

void StuInfo::SetBasicInfo(char stuname[], char stugender[], char stuid[])
{
    strcpy(Name, stuname);
    strcpy(Gender, stugender);
    strcpy(StuID, stuid);
}

void StuInfo::AddScore(char *a, int b, int c, int d, int e, int f, int pf)
{
    SL.AddToHead(a, b, c, d, e, f, pf);
    if (pf == 0)
    {
        if (b < 60)
        {
            EffCredit += c;
            EffGrade += c * score_to_grade(b);
        }
    }
    else
    {
        EffCredit += c;
        EffGrade += c * score_to_grade(b);
    }
}

ScoreNode *StuInfo::SearchScore(char *str)
{
    return SL.Search(str);
}

int StuInfo::SearchScore(char *str, int i)
{
    return SL.Search(str, 1);
}

void StuInfo::DeleteScore(char *str)
{
    int i = SL.Search(str, 0);

```

```

ScoreNode *tmp = SL.Search(str);
int c = tmp->ShowCredit();
int b = tmp->ShowScore();
int pf = tmp->ShowPF();
if (pf == 0)
{
    if (b < 60)
    {
        EffCredit -= c;
        EffGrade -= c * score_to_grade(b);
    }
}
else
{
    EffCredit -= c;
    EffGrade -= c * score_to_grade(b);
}
SL.Delete(i);
}

double StuInfo::personGPA()
{
    if (this->EffCredit == 0) return -1;
    else return this->EffGrade / this->EffCredit;
}

double StuInfo::ShowEffGrade()
{
    return this->EffGrade;
}

int StuInfo::ShowEffCredit()
{
    return this->EffCredit;
}

char * StuInfo::ShowName()
{
    return this->Name;
}

char * StuInfo::ShowStuID()
{
    return this->StuID;
}

char * StuInfo::ShowGender()
{
    return this->Gender;
}

bool StuInfo::IsEmpty()

```

```

{
    return SL.IsEmpty();
}

ScoreList StuInfo::showscorelist()
{
    return this->SL;
}

StuInfoNode::StuInfoNode(char stuname[], char stugender[], char stuid[], StuInfoNode
*next) :StuInfo(stuname, stugender, stuid)
{
    this->next = next;
}

StuInfoNode::~~StuInfoNode() {};

StuInfoList::StuInfoList()
{
    head = current = NULL;
    TotalNumber = 0;
    totalGPA = 0;
    totalCredit = 0;
}

StuInfoList::~~StuInfoList() {};
double StuInfoList::averageGPA()
{
    StuInfoNode *p = head;
    if (p == NULL) return 0;
    else
    {
        totalCredit = 0;
        totalGPA = 0;
        while (p != NULL)
        {
            if (p->ShowEffCredit() == 0);
            else
            {
                totalGPA += p->ShowEffGrade();
                totalCredit += p->ShowEffCredit();
            }
            p = p->next;
        }
        if (totalCredit == 0) return -1;
        else return totalGPA / totalCredit;
    }
}

```

```

int StuInfoList::AddToHead_STU(char stuname[], char stugender[], char stuid[])
{
    StuInfoNode *p = current;
    current = head;
    StuInfoNode *newSN = new StuInfoNode(stuname, stugender, stuid, NULL);
    newSN->next = head;
    head = current = newSN;
    this->TotalNumber += 1;
    return 1;
}

int StuInfoList::Delete_STU(int i)
{
    StuInfoNode *p, *q;
    if (i == 0) {
        q = head;
        current = head = head->next;
    }
    else {
        p = current;
        current = head;
        for (int m = 0; m < i - 1; m++)
        {
            if (current == NULL)break;
            else current = current->next;
        }
        if (current == NULL || current->next == NULL)
        {
            current = p;
            return 0;
        }
        else
        {
            q = current->next;
            current = current->next = q->next;
        }
    }
    this->TotalNumber -= 1;
    delete q;
    return 1;
}

StuInfoNode *StuInfoList::Search_STU(char *str)           //这里输入学号或者姓名查询均可
{
    StuInfoNode *p = head;
    if (p == NULL)
    {

```



```

        return NULL;
    }
    else
    {
        while (p!= NULL)
        {
            if ((strcmp(str, p->ShowName()) == 0) || (strcmp(str, p->ShowStuID()) == 0))
return p;
            p = p->next;
        }
        return NULL;
    }
}

int StuInfoList::Search_STU(char *str, int i)
{
    StuInfoNode *p = head;
    int k = 0;
    if (p == NULL)
    {
        return -1;
    }
    else
    {
        while (p!= NULL)
        {
            if ((strcmp(str, p->ShowName()) == 0) || (strcmp(str, p->ShowStuID()) == 0))
return k;
            k++;
            p = p->next;
        }
        return -1;
    }
}

bool StuInfoList::IsEmpty()
{
    if (head == NULL) return TRUE;
    else return FALSE;
}

void StuInfoList::SetTemp(int t)
{
    temp = t;
}

StuInfoNode *StuInfoList::FindNode(int t)
{
    StuInfoNode *q = head;

```

```

        for (int k = 0; k < t; k++)
        {
            q = q->next;
        }
        return q;
    }
    int StuInfoList::ShowTemp()
    {
        return temp;
    }
    int StuInfoList::showtotalnumber()
    {
        return this->TotalNumber;
    }

    StuInfoList sil;          //创建外部对象

```

#### 头文件: InfoFile.h

```

#pragma once
#include <list>
#include <fstream>
#include <iostream>
#include <string>
#define _F_LOGIN "./login.ini"
#define _F_REC "./record.txt"
#define _F_BASIC "./basic.txt"
using namespace std;
class CInfoFile
{
public:
    CInfoFile();
    ~CInfoFile();
    void ReadLogin(CString &name, CString &pwd); //读取登录信息
    void WritePwd(char* name, char* pwd); //修改密码
    void WriteInfo();                       //写入数据到文件中
    void ReadInfo();                         //从文件中读取数据
};

```

#### 源文件: InfoFile.cpp

```

#include "stdafx.h"
#include "InfoFile.h"
#include "head.h"
extern StuInfoList sil;
CInfoFile::CInfoFile()
{
}

```

```

CInfoFile::~CInfoFile()
{
}

//读取登录信息
void CInfoFile::ReadLogin( CString &name, CString &pwd )
{
    ifstream ifs;
    ifs.open(_F_LOGIN);

    char buf[1024] = {0};

    ifs.getline(buf, sizeof(buf));
    name = CString(buf);

    ifs.getline(buf, sizeof(buf));
    pwd = CString(buf);

    ifs.close();
}

//修改密码
void CInfoFile::WritePwd( char* name, char* pwd )
{
    ofstream ofs;
    ofs.open(_F_LOGIN);

    ofs<< name <<endl;
    ofs<< pwd <<endl;

    ofs.close();
}

void CInfoFile::ReadInfo()
{
    ifstream ifs1(_F_BASIC);
    char buf[1024] = { 0 };
    ifs1.getline(buf, sizeof(buf));
    while (!ifs1.eof()) //没到文件结尾
    {
        ifs1.getline(buf, sizeof(buf));
        char nn;
        char *str = strtok(buf, "|"); //以"|"分隔
        if (str != NULL)
        {
            nn = atoi(str);
        }
    }
}

```

```

else
{
    break;
}
str = strtok(NULL, "|");
char *p1 = str;
str = strtok(NULL, "|");
char *p2 = str;
str = strtok(NULL, "|");
char *p3 = str;
sil.AddToHead_STU(p1, p2, p3);
}
ifs1.close();
ifstream ifs2(_F_REC);
char tmp[1024] = { 0 };
ifs2.getline(tmp, sizeof(tmp));
while (!ifs2.eof())
{
    ifs2.getline(tmp, sizeof(tmp));
    int mm;
    char *str = strtok(tmp, "|"); //以"|"分隔
    if (str != NULL)
    {
        mm = atoi(str);
    }
    else
    {
        break;
    }
    str = strtok(NULL, "|");
    char *pp1 = str;
    str = strtok(NULL, "|");
    char *pp2 = str;
    str = strtok(NULL, "|");
    int pp3 = atoi(str);
    str = strtok(NULL, "|");
    int pp4 = atoi(str);
    str = strtok(NULL, "|");
    int pp5 = atoi(str);
    str = strtok(NULL, "|");
    int pp6 = atoi(str);
    str = strtok(NULL, "|");
    int pp7 = atoi(str);
    str = strtok(NULL, "|");
    int pp8 = atoi(str);

```

```

        sil.Search_STU(pp1)->AddScore(pp2, pp3, pp4, pp5, pp6, pp7, pp8);
    }
    ifs2.close();
}

void CInfoFile::WriteInfo()
{
    ofstream ofs1(_F_BASIC);
    if (!sil.IsEmpty())
    {
        char *bt = "序列号|姓名|性别|学号";
        ofs1 << bt << endl;
        for (int i = 0; i < sil.showtotalnumber(); i++)
        {
            ofs1 << i << "|";
            ofs1 << sil.FindNode(i)->ShowName() << "|";
            ofs1 << sil.FindNode(i)->ShowGender() << "|";
            ofs1 << sil.FindNode(i)->ShowStuID() << endl;
        }
    }
    ofs1.close();
    ofstream ofs2(_F_REC);
    if (!sil.IsEmpty())
    {
        char *st = "序列号|学号|课程名称|百分制成绩|学分|年|月|日|pf";
        ofs2 << st << endl;
        for (int i = 0; i < sil.showtotalnumber(); i++)
        {
            if (sil.FindNode(i)->IsEmpty()) continue;
            for (int j = 0; j < sil.FindNode(i)->showscorelist().coursenumber(); j++)
            {
                ofs2 << i << j << "|";
                ofs2 << sil.FindNode(i)->ShowStuID() << "|";
                ofs2 << sil.FindNode(i)->showscorelist().Search(j)->ShowSubject() << "|";
                ofs2 << sil.FindNode(i)->showscorelist().Search(j)->ShowScore() << "|";
                ofs2 << sil.FindNode(i)->showscorelist().Search(j)->ShowCredit() << "|";
                ofs2 << sil.FindNode(i)->showscorelist().Search(j)->ShowYear() << "|";
                ofs2 << sil.FindNode(i)->showscorelist().Search(j)->ShowMonth() << "|";
                ofs2 << sil.FindNode(i)->showscorelist().Search(j)->ShowDay() << "|";
                ofs2 << sil.FindNode(i)->showscorelist().Search(j)->ShowPF() << endl;
            }
        }
    }
    ofs2.close();
}

```

## MFC 图形界面相关的 15 个类——对应的头文件和源文件：

### 头文件：LoginDlg.h

```
#pragma once
class CLoginDlg : public CDialogEx
{
    DECLARE_DYNAMIC(CLoginDlg)
public:
    CLoginDlg(CWnd* pParent = NULL);    // 标准构造函数
    virtual ~CLoginDlg();
    // 对话框数据
    enum { IDD = IDD_LOGIN };
protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持
    DECLARE_MESSAGE_MAP()
private:
    CString m_username;
    CString m_password;
public:
    afx_msg void OnBnClickedButton1();
    virtual BOOL OnInitDialog();
    afx_msg void OnBnClickedButton2();
    virtual void OnOK();
    afx_msg void OnClose();
};
```

### 源文件：LoginDlg.cpp

```
// LoginDlg.cpp : 实现文件
//
#include "stdafx.h"
#include "StudentScoreSystem.h"
#include "LoginDlg.h"
#include "afxdialogex.h"
#include "InfoFile.h"
// CLoginDlg 对话框
IMPLEMENT_DYNAMIC(CLoginDlg, CDialogEx)
CLoginDlg::CLoginDlg(CWnd* pParent /*=NULL*/)
    : CDialogEx(CLoginDlg::IDD, pParent)
    , m_username(_T(""))
    , m_password(_T(""))
{
}
CLoginDlg::~CLoginDlg()
{
}
```

```

void CLoginDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
    DDX_Text(pDX, IDC_EDIT1, m_username);
    DDX_Text(pDX, IDC_EDIT2, m_password);
}

BEGIN_MESSAGE_MAP(CLoginDlg, CDialogEx)
    ON_BN_CLICKED(IDC_BUTTON1, &CLginDlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON2, &CLginDlg::OnBnClickedButton2)
    ON_WM_CLOSE()
END_MESSAGE_MAP()

// CLoginDlg 消息处理程序
void CLoginDlg::OnBnClickedButton1()
{
    //登录按钮点击
    UpdateData(TRUE);
    if (m_username.IsEmpty() || m_password.IsEmpty())
    {
        MessageBox(TEXT("用户名或密码不能为空!"));
        return;
    }
    //获取正确的值
    CInfoFile fp;
    CString crtname, crtpwd;
    fp.ReadLogin(crtname, crtpwd);

    //判断验证
    if (crtname == m_username)
    {
        if (crtpwd == m_password)
        {
            //关闭登录对话框
            CDialog::OnOK();
        }
        else
        {
            MessageBox(TEXT("密码不正确!"));
        }
    }
    else
    {
        MessageBox(TEXT("用户名不正确!"));
    }
}

```

```

}

BOOL CLoginDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();
    //添加默认出现内容
    CInfoFile fp;
    CString crtname, crtpwd;
    fp.ReadLogin(crtname, crtpwd);
    m_username = crtname;
    m_password = crtpwd;
    UpdateData(FALSE);

    return TRUE;
}

void CLoginDlg::OnBnClickedButton2()
{
    exit(0);
}

void CLoginDlg::OnOK()
{
    //CDialogEx::OnOK();
}

void CLoginDlg::OnClose()
{
    //CDialogEx::OnClose();
    exit(0);
}

```

## 头文件: MainFrm.h

```

// MainFrm.h : CMainFrame 类的接口
//
#pragma once
//自定义消息
#define NM_A (WM_USER+10)
#define NM_B (WM_USER+11)
#define NM_C (WM_USER+12)
#define NM_D (WM_USER+13)
#define NM_E (WM_USER+14)
#define NM_F (WM_USER+15)
#define NM_G (WM_USER+16)

class CMainFrame : public CFrameWnd

```



```

{

protected: // 仅从序列化创建
    CMainFrame();
    DECLARE_DYNCREATE(CMainFrame)
// 重写
public:
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
// 实现
public:
    virtual ~CMainFrame();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected: // 控件条嵌入成员
    CStatusBar          m_wndStatusBar;

// 生成的消息映射函数
protected:
    afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
    DECLARE_MESSAGE_MAP()
private:
    CSplitterWnd m_spw;    //拆分窗口对象
    virtual BOOL OnCreateClient(LPCREATESTRUCT lpcs, CCreateContext* pContext);
    afx_msg LRESULT OnMyChange(WPARAM wparam, LPARAM lparam);
public:
    afx_msg void OnClose();
    afx_msg void On32776();
    afx_msg void On32771();
    afx_msg void On32772();
    afx_msg void On32773();
    afx_msg void On32774();
    afx_msg void On32775();
};

```

#### 源文件：MainFrm.cpp

```

// MainFrm.cpp : CMainFrame 类的实现
//

#include "stdafx.h"
#include "StudentScoreSystem.h"
#include "SelectPart.h"

```

```

#include "MainFrm.h"
#include "OperatePart.h"
#include "UserDlg.h"
#include "SysScoreManager.h"
#include "SysScore.h"
#include "SysScoreList.h"
#include "SysBasicInfo.h"
#include "SysTotal.h"
#include "SysBasicManager.h"
#include "head.h"
#include "InfoFile.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#endif
extern StuInfoList sil;
// CMainFrame

IMPLEMENT_DYNCREATE(CMainFrame, CFrameWnd)

BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
    ON_WM_CREATE()
    //ON_MESSAGE响应自定义消息
    ON_MESSAGE(NM_A, OnMyChange)
    ON_MESSAGE(NM_B, OnMyChange)
    ON_MESSAGE(NM_C, OnMyChange)
    ON_MESSAGE(NM_D, OnMyChange)
    ON_MESSAGE(NM_E, OnMyChange)
    ON_MESSAGE(NM_F, OnMyChange)
    ON_MESSAGE(NM_G, OnMyChange)

    ON_WM_CLOSE()
    ON_COMMAND(ID_32776, &CMainFrame::On32776)
    ON_COMMAND(ID_32771, &CMainFrame::On32771)
    ON_COMMAND(ID_32772, &CMainFrame::On32772)
    ON_COMMAND(ID_32773, &CMainFrame::On32773)
    ON_COMMAND(ID_32774, &CMainFrame::On32774)
    ON_COMMAND(ID_32775, &CMainFrame::On32775)
END_MESSAGE_MAP()

static UINT indicators[] =
{
    ID_SEPARATOR,          // 状态行指示器
    ID_INDICATOR_CAPS,
    ID_INDICATOR_NUM,

```

```

        ID_INDICATOR_SCROLL,
    };

// CMainFrame 构造/析构

CMainFrame::CMainFrame()
{
}

CMainFrame::~CMainFrame()
{
}

int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;

    if (!m_wndStatusBar.Create(this))
    {
        TRACE0("未能创建状态栏\n");
        return -1;    // 未能创建
    }
    m_wndStatusBar.SetIndicators(indicators, sizeof(indicators)/sizeof(UINT));
    //设置窗口
    MoveWindow(0, 0, 1000, 750);
    //屏幕居中
    CenterWindow();
    //修改图标
    SetClassLong(m_hWnd, GCL_HICON, (LONG)AfxGetApp()->LoadIconW(IDI_ICON4));
    //设置标题 右侧部分
    SetTitle(TEXT("学生成绩管理系统"));
    return 0;
}

BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
{
    if( !CFrameWnd::PreCreateWindow(cs) )
        return FALSE;
    return TRUE;
}

// CMainFrame 诊断

#ifdef _DEBUG

```

```

void CMainFrame::AssertValid() const
{
    CFrameWnd::AssertValid();
}

void CMainFrame::Dump(CDumpContext& dc) const
{
    CFrameWnd::Dump(dc);
}

#ifdef _DEBUG

// CMainFrame 消息处理程序

BOOL CMainFrame::OnCreateClient(LPCREATESTRUCT lpcs, CCreateContext* pContext)
{
    //return CFrameWnd::OnCreateClient(lpcs, pContext);
    //拆成1行2列
    m_spw.CreateStatic(this, 1, 2);
    //左侧和右侧具体的显示内容
    m_spw.CreateView(0, 0, RUNTIME_CLASS(CSelectPart), CSize(220, 500), pContext);
    m_spw.CreateView(0, 1, RUNTIME_CLASS(COperatePart), CSize(750, 500), pContext);

    return TRUE; //代表自己拆分
}

LRESULT CMainFrame::OnMyChange(WPARAM wparam, LPARAM lparam)
{
    CCreateContext context;
    if (wparam == NM_A)
    {
        //挂载界面
        context.m_pNewViewClass = RUNTIME_CLASS(CUserDlg);
        context.m_pCurrentFrame = this;
        context.m_pLastView = (CFormView *)m_spw.GetPane(0, 1);
        m_spw.DeleteView(0, 1);
        m_spw.CreateView(0, 1, RUNTIME_CLASS(CUserDlg), CSize(750, 500), &context);
        CUserDlg *pNewView = (CUserDlg *)m_spw.GetPane(0, 1);
        m_spw.RecalcLayout();
        pNewView->OnInitialUpdate();
        m_spw.SetActivePane(0, 1);
    }
    else if (wparam == NM_B)
    {

```

```

context.m_pNewViewClass = RUNTIME_CLASS(CSysBasicInfo);
context.m_pCurrentFrame = this;
context.m_pLastView = (CFormView *)m_spw.GetPane(0, 1);
m_spw.DeleteView(0, 1);
m_spw.CreateView(0, 1, RUNTIME_CLASS(CSysBasicInfo), CSize(750, 500), &context);
CSysBasicInfo *pNewView = (CSysBasicInfo *)m_spw.GetPane(0, 1);
m_spw.RecalcLayout();
pNewView->OnInitialUpdate();
m_spw.SetActivePane(0, 1);
}
else if (wparam == NM_C)
{
context.m_pNewViewClass = RUNTIME_CLASS(CSysScore);
context.m_pCurrentFrame = this;
context.m_pLastView = (CFormView *)m_spw.GetPane(0, 1);
m_spw.DeleteView(0, 1);
m_spw.CreateView(0, 1, RUNTIME_CLASS(CSysScore), CSize(750, 500), &context);
CSysScore *pNewView = (CSysScore *)m_spw.GetPane(0, 1);
m_spw.RecalcLayout();
pNewView->OnInitialUpdate();
m_spw.SetActivePane(0, 1);
}
else if (wparam == NM_D)
{
context.m_pNewViewClass = RUNTIME_CLASS(CSysScoreList);
context.m_pCurrentFrame = this;
context.m_pLastView = (CFormView *)m_spw.GetPane(0, 1);
m_spw.DeleteView(0, 1);
m_spw.CreateView(0, 1, RUNTIME_CLASS(CSysScoreList), CSize(750, 500), &context);
CSysScoreList *pNewView = (CSysScoreList *)m_spw.GetPane(0, 1);
m_spw.RecalcLayout();
pNewView->OnInitialUpdate();
m_spw.SetActivePane(0, 1);
}
else if (wparam == NM_E)
{
context.m_pNewViewClass = RUNTIME_CLASS(CSysTotal);
context.m_pCurrentFrame = this;
context.m_pLastView = (CFormView *)m_spw.GetPane(0, 1);
m_spw.DeleteView(0, 1);
m_spw.CreateView(0, 1, RUNTIME_CLASS(CSysTotal), CSize(750, 500), &context);
CSysTotal *pNewView = (CSysTotal *)m_spw.GetPane(0, 1);
m_spw.RecalcLayout();
pNewView->OnInitialUpdate();
m_spw.SetActivePane(0, 1);
}

```

```

    }
    else if (wparam == NM_F)
    {
        context.m_pNewViewClass = RUNTIME_CLASS(CSysScoreManager);
        context.m_pCurrentFrame = this;
        context.m_pLastView = (CFormView *)m_spw.GetPane(0, 1);
        m_spw.DeleteView(0, 1);
        m_spw.CreateView(0, 1, RUNTIME_CLASS(CSysScoreManager), CSize(750, 500), &context);
        CSysScoreManager *pNewView = (CSysScoreManager *)m_spw.GetPane(0, 1);
        m_spw.RecalcLayout();
        pNewView->OnInitialUpdate();
        m_spw.SetActivePane(0, 1);
    }
    else if (wparam == NM_G)
    {
        context.m_pNewViewClass = RUNTIME_CLASS(CSysBasicManager);
        context.m_pCurrentFrame = this;
        context.m_pLastView = (CFormView *)m_spw.GetPane(0, 1);
        m_spw.DeleteView(0, 1);
        m_spw.CreateView(0, 1, RUNTIME_CLASS(CSysBasicManager), CSize(750, 500), &context);
        CSysBasicManager *pNewView = (CSysBasicManager *)m_spw.GetPane(0, 1);
        m_spw.RecalcLayout();
        pNewView->OnInitialUpdate();
        m_spw.SetActivePane(0, 1);
    }

    return 0;
}

void CMainFrame::OnClose()
{
    CInfoFile fps;
    fps.WriteInfo();
    CFrameWnd::OnClose();
}

//以下都是与菜单栏的各项功能相关的函数
//退出
void CMainFrame::On32776()
{
    CInfoFile fps;
    fps.WriteInfo();
    exit(0);
}

```

```

//切换界面
void CMainFrame::On32771()
{
    ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_A, (WPARAM)NM_A, (LPARAM)0);
}

void CMainFrame::On32772()
{
    ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_B, (WPARAM)NM_B, (LPARAM)0);
}

void CMainFrame::On32773()
{
    ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_C, (WPARAM)NM_C, (LPARAM)0);
}

void CMainFrame::On32774()
{
    ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_D, (WPARAM)NM_D, (LPARAM)0);
}

void CMainFrame::On32775()
{
    ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_E, (WPARAM)NM_E, (LPARAM)0);
}

```

## 头文件: OperatePart.h

```

#pragma once
// COperatePart 窗体视图

class COperatePart : public CFormView
{
    DECLARE_DYNCREATE(COperatePart)

protected:
    COperatePart();           // 动态创建所使用的受保护的构造函数
    virtual ~COperatePart();

public:
    enum { IDD = IDD_OPERATEPART };

```

```

#ifdef _DEBUG
    virtual void AssertValid() const;
#endif
#ifdef _WIN32_WCE
    virtual void Dump(CDumpContext& dc) const;
#endif
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

    DECLARE_MESSAGE_MAP()
};

```

**源文件: OperatePart.cpp**

// OperatePart.cpp : 实现文件

```

#include "stdafx.h"
#include "StudentScoreSystem.h"
#include "OperatePart.h"

// COperatePart

IMPLEMENT_DYNCREATE(COperatePart, CFormView)

COperatePart::COperatePart()
    : CFormView(COperatePart::IDD)
{
}

COperatePart::~COperatePart()
{
}

void COperatePart::DoDataExchange(CDataExchange* pDX)
{
    CFormView::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(COperatePart, CFormView)
END_MESSAGE_MAP()

// COperatePart 诊断

```



```

#ifdef _DEBUG
void COperatePart::AssertValid() const
{
    CFormView::AssertValid();
}

#ifdef _WIN32_WCE
void COperatePart::Dump(CDumpContext& dc) const
{
    CFormView::Dump(dc);
}
#endif

#endif // _DEBUG

头文件: SelectPart.h

#pragma once
// CSelectPart 视图

class CSelectPart : public CTreeView
{
    DECLARE_DYNCREATE(CSelectPart)

protected:
    CSelectPart();           // 动态创建所使用的受保护的构造函数
    virtual ~CSelectPart();

public:
#ifdef _DEBUG
    virtual void AssertValid() const;
#endif
#ifdef _WIN32_WCE
    virtual void Dump(CDumpContext& dc) const;
#endif
#endif

protected:
    DECLARE_MESSAGE_MAP()
private:
    CTreeCtrl *m_tcl; //树控件
    CImageList m_iml; //图标列表
public:
    virtual void OnInitialUpdate();
    afx_msg void OnTvnSelchanged(NMHDR *pNMHDR, LRESULT *pResult);
};

```

**源文件: SelectPart.cpp**  
// SelectPart.cpp : 实现文件

```

//

#include "stdafx.h"
#include "StudentScoreSystem.h"
#include "SelectPart.h"
#include "MainFrm.h"

// CSelectPart

IMPLEMENT_DYNCREATE(CSelectPart, CTreeView)

CSelectPart::CSelectPart()
{
}

CSelectPart::~CSelectPart()
{
}

BEGIN_MESSAGE_MAP(CSelectPart, CTreeView)
    ON_NOTIFY_REFLECT(TVN_SELCHANGED, &CSelectPart::OnTvnSelchanged)
END_MESSAGE_MAP()

// CSelectPart 诊断

#ifdef _DEBUG
void CSelectPart::AssertValid() const
{
    CTreeView::AssertValid();
}

#endif

#ifdef _WIN32_WCE
void CSelectPart::Dump(CDumpContext& dc) const
{
    CTreeView::Dump(dc);
}
#endif

// CSelectPart 消息处理程序

void CSelectPart::OnInitialUpdate()
{
    CTreeView::OnInitialUpdate();
}

```

```

//初始化树视图
//获取树控件
m_tcl = &GetTreeCtrl();
//准备图片集合并添加结点
HICON icon = AfxGetApp()->LoadIconW(IDI_ICON2);
m_iml.Create(50, 50, ILC_COLOR32, 1, 1);
m_iml.Add(icon);
m_tcl->SetImageList(&m_iml, TVSIL_NORMAL);
m_tcl->InsertItem(TEXT("用户中心"), 0, 0, NULL);
m_tcl->InsertItem(TEXT("学生基本信息管理"), 0, 0, NULL);
m_tcl->InsertItem(TEXT("学生成绩管理"), 0, 0, NULL);
m_tcl->InsertItem(TEXT("学生成绩单打印"), 0, 0, NULL);
m_tcl->InsertItem(TEXT("总体信息统计情况"), 0, 0, NULL);
}

void CSelectPart::OnTvnSelchanged(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMTREEVIEW pNMTreeView = reinterpret_cast<LPNMTREEVIEW>(pNMHDR);
    *pResult = 0;
    //获取当前选中项
    HTREEITEM hitem = m_tcl->GetSelectedItem();
    CString temp = m_tcl->GetItemText(hitem);
    //发送消息
    if (temp == TEXT("用户中心"))
    {
        //返回窗口的句柄，发送自定义消息，指定附加消息信息，0
        ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_A, (WPARAM)NM_A, (LPARAM)0);
    }
    else if (temp == TEXT("学生基本信息管理"))
    {
        ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_B, (WPARAM)NM_B, (LPARAM)0);
    }
    else if (temp == TEXT("学生成绩管理"))
    {
        ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_C, (WPARAM)NM_C, (LPARAM)0);
    }
    else if (temp == TEXT("学生成绩单打印"))
    {
        ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_D, (WPARAM)NM_D, (LPARAM)0);
    }
    else if (temp == TEXT("总体信息统计情况"))
    {
        ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_E, (WPARAM)NM_E, (LPARAM)0);
    }
}

```

```
}
```

### 头文件: StudentScoreSystem.h

```
// StudentScoreSystem.h : StudentScoreSystem 应用程序的主头文件
//
#pragma once

#ifdef __AFXWIN_H__
    #error "在包含此文件之前包含“stdafx.h”以生成 PCH 文件”
#endif

#include "resource.h" // 主符号

// CStudentScoreSystemApp:
// 有关此类的实现, 请参阅 StudentScoreSystem.cpp
//
class CStudentScoreSystemApp : public CWinApp
{
public:
    CStudentScoreSystemApp();
// 重写
public:
    virtual BOOL InitInstance();
    virtual int ExitInstance();
// 实现
    afx_msg void OnAppAbout();
    DECLARE_MESSAGE_MAP()
};

extern CStudentScoreSystemApp theApp;
```

### 源文件: StudentScoreSystem.cpp

```
// StudentScoreSystem.cpp : 定义应用程序的类行为。
//

#include "stdafx.h"
#include "afxwinappex.h"
#include "afxdialogex.h"
#include "StudentScoreSystem.h"
#include "MainFrm.h"
#include "LoginDlg.h"
#include "StudentScoreSystemDoc.h"
#include "StudentScoreSystemView.h"
#include "head.h"
#include "InfoFile.h"
```

```

extern StuInfoList sil;
#ifdef _DEBUG
#define new DEBUG_NEW
#endif
// CStudentScoreSystemApp

BEGIN_MESSAGE_MAP(CStudentScoreSystemApp, CWinApp)
    ON_COMMAND(ID_APP_ABOUT, &CStudentScoreSystemApp::OnAppAbout)
    // 基于文件的标准文档命令
    ON_COMMAND(ID_FILE_NEW, &CWinApp::OnFileNew)
    ON_COMMAND(ID_FILE_OPEN, &CWinApp::OnFileOpen)
    // 标准打印设置命令
    ON_COMMAND(ID_FILE_PRINT_SETUP, &CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()

// CStudentScoreSystemApp 构造

CStudentScoreSystemApp::CStudentScoreSystemApp()
{
    // 支持重新启动管理器
    m_dwRestartManagerSupportFlags = AFX_RESTART_MANAGER_SUPPORT_ALL_ASPECTS;
#ifdef _MANAGED
    System::Windows::Forms::Application::SetUnhandledExceptionMode(System::Windows::Forms::Unha
ndledExceptionMode::ThrowException);
#endif
    SetAppID(_T("StudentScoreSystem.AppID.NoVersion"));
}

// 唯一的一个 CStudentScoreSystemApp 对象
CStudentScoreSystemApp theApp;

// CStudentScoreSystemApp 初始化

BOOL CStudentScoreSystemApp::InitInstance()
{
    INITCOMMONCONTROLSEX InitCtrls;
    InitCtrls.dwSize = sizeof(InitCtrls);
    InitCtrls.dwICC = ICC_WIN95_CLASSES;
    InitCommonControlsEx(&InitCtrls);
    //在调用基类初始化函数前，先创建登录窗口并且是模态方式
    CLoginDlg logindlg1;
    logindlg1.DoModal();
}

```

```

CWinApp::InitInstance();

// 初始化 OLE 库
if (!AfxOleInit())
{
    AfxMessageBox(IDP_OLE_INIT_FAILED);
    return FALSE;
}

AfxEnableControlContainer();

EnableTaskbarInteraction(FALSE);
SetRegistryKey(_T("应用程序向导生成的本地应用程序"));
LoadStdProfileSettings(4);
CSingleDocTemplate* pDocTemplate;
pDocTemplate = new CSingleDocTemplate(
    IDR_MAINFRAME,
    RUNTIME_CLASS(CStudentScoreSystemDoc),
    RUNTIME_CLASS(CMainFrame),      // 主 SDI 框架窗口
    RUNTIME_CLASS(CStudentScoreSystemView));
if (!pDocTemplate)
    return FALSE;
AddDocTemplate(pDocTemplate);
// 分析标准 shell 命令、DDE、打开文件操作的命令行
CCommandLineInfo cmdInfo;
ParseCommandLine(cmdInfo);
// 调度在命令行中指定的命令。 如果
// 用 /RegServer、/Register、/Unregserver 或 /Unregister 启动应用程序，则返回 FALSE。
if (!ProcessShellCommand(cmdInfo))
    return FALSE;
// 唯一的一个窗口已初始化，因此显示它并对其进行更新
m_pMainWnd->ShowWindow(SW_SHOW);
m_pMainWnd->UpdateWindow();
CInfoFile fps;
fps.ReadInfo();
return TRUE;
}

int CStudentScoreSystemApp::ExitInstance()
{
    AfxOleTerm(FALSE);
    return CWinApp::ExitInstance();
}

// CStudentScoreSystemApp 消息处理程序

```

```

// 用于应用程序“关于”菜单项的 CAboutDlg 对话框

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

// 对话框数据
    enum { IDD = IDD_ABOUTBOX };

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(CAboutDlg::IDD)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

// 用于运行对话框的应用程序命令
void CStudentScoreSystemApp::OnAppAbout()
{
    CAboutDlg aboutDlg;
    aboutDlg.DoModal();
}

// CStudentScoreSystemApp 消息处理程序

头文件：StudentScoreSystemDoc.h

// StudentScoreSystemDoc.h : CStudentScoreSystemDoc 类的接口
//

```

```

#pragma once
class CStudentScoreSystemDoc : public CDocument
{
protected: // 仅从序列化创建
    CStudentScoreSystemDoc();
    DECLARE_DYNCREATE(CStudentScoreSystemDoc)
// 重写
public:
    virtual BOOL OnNewDocument();
    virtual void Serialize(CArchive& ar);
#ifdef SHARED_HANDLERS
    virtual void InitializeSearchContent();
    virtual void OnDrawThumbnail(CDC& dc, LPRECT lprcBounds);
#endif // SHARED_HANDLERS

// 实现
public:
    virtual ~CStudentScoreSystemDoc();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// 生成的消息映射函数
protected:
    DECLARE_MESSAGE_MAP()

#ifdef SHARED_HANDLERS
    // 用于为搜索处理程序设置搜索内容的 Helper 函数
    void SetSearchContent(const CString& value);
#endif // SHARED_HANDLERS
};

```

#### 源文件: StudentScoreSystemDoc.cpp

```

// StudentScoreSystemDoc.cpp : CStudentScoreSystemDoc 类的实现
//

```

```

#include "stdafx.h"
// SHARED_HANDLERS 可以在实现预览、缩略图和搜索筛选器句柄的
// ATL 项目中进行定义，并允许与该项目共享文档代码。
#ifdef SHARED_HANDLERS
#include "StudentScoreSystem.h"
#endif

```



```

#include "StudentScoreSystemDoc.h"

#include <propkey.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// CStudentScoreSystemDoc

IMPLEMENT_DYNCREATE(CStudentScoreSystemDoc, CDocument)

BEGIN_MESSAGE_MAP(CStudentScoreSystemDoc, CDocument)
END_MESSAGE_MAP()

// CStudentScoreSystemDoc 构造/析构

CStudentScoreSystemDoc::CStudentScoreSystemDoc()
{
}

CStudentScoreSystemDoc::~CStudentScoreSystemDoc()
{
}

BOOL CStudentScoreSystemDoc::OnNewDocument()
{
    if (!CDocument::OnNewDocument())
        return FALSE;
    //设置左侧标题
    SetTitle(TEXT("清华大学"));
    return TRUE;
}

// CStudentScoreSystemDoc 序列化

void CStudentScoreSystemDoc::Serialize(CArchive& ar)
{
    if (ar.IsStoring())
    {
    }
    else
    {
    }
}

```

```

#ifdef SHARED_HANDLERS

// 缩略图的支持
void CStudentScoreSystemDoc::OnDrawThumbnail(CDC& dc, LPRECT lprcBounds)
{
    // 修改此代码以绘制文档数据
    dc.FillSolidRect(lprcBounds, RGB(255, 255, 255));

    CString strText = _T("TODO: implement thumbnail drawing here");
    LOGFONT lf;

    CFont* pDefaultGUIFont = CFont::FromHandle((HFONT) GetStockObject(DEFAULT_GUI_FONT));
    pDefaultGUIFont->GetLogFont(&lf);
    lf.lfHeight = 36;

    CFont fontDraw;
    fontDraw.CreateFontIndirect(&lf);

    CFont* pOldFont = dc.SelectObject(&fontDraw);
    dc.DrawText(strText, lprcBounds, DT_CENTER | DT_WORDBREAK);
    dc.SelectObject(pOldFont);
}

// 搜索处理程序的支持
void CStudentScoreSystemDoc::InitializeSearchContent()
{
    CString strSearchContent;
    // 从文档数据设置搜索内容。
    // 内容部分应由“;”分隔

    // 例如:    strSearchContent = _T("point;rectangle;circle;ole object;");
    SetSearchContent(strSearchContent);
}

void CStudentScoreSystemDoc::SetSearchContent(const CString& value)
{
    if (value.IsEmpty())
    {
        RemoveChunk(PKEY_Search_Contents.fmtid, PKEY_Search_Contents.pid);
    }
    else
    {
        CMFCFilterChunkValueImpl *pChunk = NULL;
        ATLTRY(pChunk = new CMFCFilterChunkValueImpl);
    }
}

```

```

        if (pChunk != NULL)
        {
            pChunk->SetTextValue(PKEY_Search_Contents, value, CHUNK_TEXT);
            SetChunkValue(pChunk);
        }
    }
}

```

```

#endif // SHARED_HANDLERS

```

```

// CStudentScoreSystemDoc 诊断

```

```

#ifdef _DEBUG
void CStudentScoreSystemDoc::AssertValid() const
{
    CDocument::AssertValid();
}

```

```

void CStudentScoreSystemDoc::Dump(CDumpContext& dc) const
{
    CDocument::Dump(dc);
}

```

```

#endif // _DEBUG

```

```

// CStudentScoreSystemDoc 命令

```

**头文件: StudentScoreSystemView.h**

```

// StudentScoreSystemView.h : CStudentScoreSystemView 类的接口
//

```

```

#pragma once
class CStudentScoreSystemView : public CView
{
protected: // 仅从序列化创建
    CStudentScoreSystemView();
    DECLARE_DYNCREATE(CStudentScoreSystemView)

```

```

// 特性
public:
    CStudentScoreSystemDoc* GetDocument() const;

```

```

// 重写
public:
    virtual void OnDraw(CDC* pDC); // 重写以绘制该视图

```

```

        virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
protected:
        virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
        virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
        virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);

// 实现
public:
        virtual ~CStudentScoreSystemView();
#ifdef _DEBUG
        virtual void AssertValid() const;
        virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// 生成的消息映射函数
protected:
        DECLARE_MESSAGE_MAP()
};

#ifdef _DEBUG // StudentScoreSystemView.cpp 中的调试版本
inline CStudentScoreSystemDoc* CStudentScoreSystemView::GetDocument() const
{ return reinterpret_cast<CStudentScoreSystemDoc*>(m_pDocument); }
#endif

```

**源文件：StudentScoreSystemView.cpp**

```

// StudentScoreSystemView.cpp : CStudentScoreSystemView 类的实现
//

#include "stdafx.h"
#ifdef SHARED_HANDLERS
#include "StudentScoreSystem.h"
#endif

#include "StudentScoreSystemDoc.h"
#include "StudentScoreSystemView.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// CStudentScoreSystemView

```

```

IMPLEMENT_DYNCREATE(CStudentScoreSystemView, CView)

BEGIN_MESSAGE_MAP(CStudentScoreSystemView, CView)
    // 标准打印命令
    ON_COMMAND(ID_FILE_PRINT, &CView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_DIRECT, &CView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_PREVIEW, &CView::OnFilePrintPreview)
END_MESSAGE_MAP()

// CStudentScoreSystemView 构造/析构

CStudentScoreSystemView::CStudentScoreSystemView()
{
}

CStudentScoreSystemView::~CStudentScoreSystemView()
{
}

BOOL CStudentScoreSystemView::PreCreateWindow(CREATESTRUCT& cs)
{
    return CView::PreCreateWindow(cs);
}

// CStudentScoreSystemView 绘制

void CStudentScoreSystemView::OnDraw(CDC* /*pDC*/)
{
    CStudentScoreSystemDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;
}

// CStudentScoreSystemView 打印

BOOL CStudentScoreSystemView::OnPreparePrinting(CPrintInfo* pInfo)
{
    // 默认准备
    return DoPreparePrinting(pInfo);
}

void CStudentScoreSystemView::OnBeginPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
}

```

```

}

void CStudentScoreSystemView::OnEndPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
}

// CStudentScoreSystemView 诊断

#ifdef _DEBUG
void CStudentScoreSystemView::AssertValid() const
{
    CView::AssertValid();
}

void CStudentScoreSystemView::Dump(CDumpContext& dc) const
{
    CView::Dump(dc);
}

CStudentScoreSystemDoc* CStudentScoreSystemView::GetDocument() const // 非调试版本是内联的
{
    ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CStudentScoreSystemDoc)));
    return (CStudentScoreSystemDoc*)m_pDocument;
}
#endif // _DEBUG

// CStudentScoreSystemView 消息处理程序

```

## 头文件: SysBasicInfo.h

```

#pragma once
#include "afxwin.h"

// CSysBasicInfo 窗体视图

class CSysBasicInfo : public CFormView
{
    DECLARE_DYNCREATE(CSysBasicInfo)

protected:
    CSysBasicInfo();           // 动态创建所使用的受保护的构造函数
    virtual ~CSysBasicInfo();

public:

```

```

        enum { IDD = IDD_BasicInfo };
#ifdef _DEBUG
        virtual void AssertValid() const;
#endif
#ifdef _WIN32_WCE
        virtual void Dump(CDumpContext& dc) const;
#endif
#endif

protected:
        virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

        DECLARE_MESSAGE_MAP()
public:
        afx_msg void OnBnClickedButton9();
private:
        CComboBox m_combo1;
        CString m_newname;
        CString m_newid;
public:
        virtual void OnInitialUpdate();
        afx_msg void OnBnClickedButton3();
        afx_msg void OnBnClickedButton6();

private:
        CString m_searchname;
public:
        afx_msg void OnBnClickedButton10();
};

```

### 源文件：SysBasicInfo.cpp

```

// SysBasicInfo.cpp : 实现文件
//

#include "stdafx.h"
#include "StudentScoreSystem.h"
#include "SysBasicInfo.h"
#include "MainFrm.h"
#include "head.h"

// CSysBasicInfo
extern StuInfoList sil;
IMPLEMENT_DYNCREATE(CSysBasicInfo, CFormView)

CSysBasicInfo::CSysBasicInfo()

```

```

        : CFormView(CSysBasicInfo::IDD)
        , m_newname(_T(""))
        , m_newid(_T(""))
        , m_searchname(_T(""))
    {

    }

CSysBasicInfo::~CSysBasicInfo()
{
}

void CSysBasicInfo::DoDataExchange(CDataExchange* pDX)
{
    CFormView::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_COMBO1, m_combol);
    DDX_Text(pDX, IDC_EDIT1, m_newname);
    DDX_Text(pDX, IDC_EDIT3, m_newid);
    DDX_Text(pDX, IDC_EDIT2, m_searchname);
}

BEGIN_MESSAGE_MAP(CSysBasicInfo, CFormView)
    ON_BN_CLICKED(IDC_BUTTON9, &CSysBasicInfo::OnBnClickedButton9)
    ON_BN_CLICKED(IDC_BUTTON3, &CSysBasicInfo::OnBnClickedButton3)
    ON_BN_CLICKED(IDC_BUTTON6, &CSysBasicInfo::OnBnClickedButton6)
    ON_BN_CLICKED(IDC_BUTTON10, &CSysBasicInfo::OnBnClickedButton10)
END_MESSAGE_MAP()

// CSysBasicInfo 诊断

#ifdef _DEBUG
void CSysBasicInfo::AssertValid() const
{
    CFormView::AssertValid();
}

#endif

#ifdef _WIN32_WCE
void CSysBasicInfo::Dump(CDumpContext& dc) const
{
    CFormView::Dump(dc);
}
#endif
#endif // _DEBUG

```



```
// CSysBasicInfo 消息处理程序
```

```
void CSysBasicInfo::OnBnClickedButton9()
{
    if (sil.IsEmpty())
    {
        MessageBox(TEXT("信息库中没有任何学生信息!"));
        return;
    }
    UpdateData(TRUE);
    if (m_searchname.IsEmpty())
    {
        MessageBox(TEXT("输入为空!"));
        return;
    }
    else
    {
        CStringA stmpl;
        stmpl = m_searchname;
        if (sil.Search_STU(stmpl.GetBuffer(), 1) == -1)
        {
            m_searchname.Empty();
            UpdateData(FALSE);
            MessageBox(TEXT("未查询到相关学生!"));
            return;
        }
        else
        {
            sil.SetTemp(sil.Search_STU(stmpl.GetBuffer(), 1));
            ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_G, (WPARAM)NM_G, (LPARAM)0);
        }
    }
}

void CSysBasicInfo::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    m_combo1.AddString(TEXT("男"));
    m_combo1.AddString(TEXT("女"));
    m_combo1.SetCurSel(0);
}
```

```

void CSysBasicInfo::OnBnClickedButton3()
{
    UpdateData(TRUE);
    if (m_newid.IsEmpty() || m_newname.IsEmpty())
    {
        MessageBox(TEXT("姓名或学号不能为空!"));
        return;
    }
    if (m_newid.SpanIncluding(TEXT("0123456789"))!=m_newid)
    {
        MessageBox(TEXT("学号需输入纯数字!"));
        return;
    }
    CStringA tmp1, tmp2, tmp3;
    tmp1 = m_newname;
    tmp2 = m_newid;
    int index = m_combo1.GetCurSel();
    //根据索引得到具体内容
    CString str;
    m_combo1.GetLBText(index, str);
    tmp3 = str;
    if (sil.Search_STU(tmp2.GetBuffer(), 1) != -1)
    {
        MessageBox(TEXT("该学生学号与已有学生学号相同, 无法添加!"));
        return;
    }
    sil.AddToHead_STU(tmp1.GetBuffer(), tmp3.GetBuffer(), tmp2.GetBuffer());
    MessageBox(TEXT("添加新的学生基本信息成功!"));
    m_newid.Empty();
    m_newname.Empty();
    m_combo1.SetCurSel(0);
    UpdateData(FALSE);
}

void CSysBasicInfo::OnBnClickedButton6()
{
    m_newid.Empty();
    m_newname.Empty();
    m_combo1.SetCurSel(0);
    UpdateData(FALSE);
}

```

```

void CSysBasicInfo::OnBnClickedButton10()
{
    m_searchname.Empty();
    UpdateData(FALSE);
}

```

## 头文件: SysBasicManager.h

```

#pragma once
#include "afxwin.h"
// CSysBasicManager 窗体视图

class CSysBasicManager : public CFormView
{
    DECLARE_DYNCREATE(CSysBasicManager)

protected:
    CSysBasicManager();           // 动态创建所使用的受保护的构造函数
    virtual ~CSysBasicManager();

public:
    enum { IDD = IDD_BasicInfoManager };
#ifdef _DEBUG
    virtual void AssertValid() const;
#endif
#ifdef _WIN32_WCE
    virtual void Dump(CDumpContext& dc) const;
#endif
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

    DECLARE_MESSAGE_MAP()
public:
    afx_msg void OnBnClickedButton6();
private:
    CComboBox m_combo2;
public:
    virtual void OnInitialUpdate();
private:
    CString m_showname;
    CString m_showid;
    CString m_showgender;
    CString m_changename;
    CString m_changeid;
public:

```

```

        afx_msg void OnBnClickedButton5();
        afx_msg void OnBnClickedButton4();
        afx_msg void OnBnClickedButton3();
};

```

### 源文件: SysBasicManager.cpp

```

// SysBasicManager.cpp : 实现文件
//

#include "stdafx.h"
#include "StudentScoreSystem.h"
#include "SysBasicManager.h"
#include "MainFrm.h"
#include "head.h"

// CSysBasicManager
extern StuInfoList sil;
IMPLEMENT_DYNCREATE(CSysBasicManager, CFormView)

CSysBasicManager::CSysBasicManager()
    : CFormView(CSysBasicManager::IDD)
    , m_showname(_T(""))
    , m_showid(_T(""))
    , m_showgender(_T(""))
    , m_changename(_T(""))
    , m_changeid(_T(""))
{
}

CSysBasicManager::~CSysBasicManager()
{
}

void CSysBasicManager::DoDataExchange(CDataExchange* pDX)
{
    CFormView::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_COMBO2, m_combo2);
    DDX_Text(pDX, IDC_EDIT1, m_showname);
    DDX_Text(pDX, IDC_EDIT3, m_showid);
    DDX_Text(pDX, IDC_EDIT5, m_showgender);
    DDX_Text(pDX, IDC_EDIT6, m_changename);
    DDX_Text(pDX, IDC_EDIT4, m_changeid);
}

BEGIN_MESSAGE_MAP(CSysBasicManager, CFormView)

```

```

        ON_BN_CLICKED(IDC_BUTTON6, &CSysBasicManager::OnBnClickedButton6)
        ON_BN_CLICKED(IDC_BUTTON5, &CSysBasicManager::OnBnClickedButton5)
        ON_BN_CLICKED(IDC_BUTTON4, &CSysBasicManager::OnBnClickedButton4)
        ON_BN_CLICKED(IDC_BUTTON3, &CSysBasicManager::OnBnClickedButton3)
    END_MESSAGE_MAP()

// CSysBasicManager 诊断

#ifdef _DEBUG
void CSysBasicManager::AssertValid() const
{
    CFormView::AssertValid();
}

#endif

#ifdef _WIN32_WCE
void CSysBasicManager::Dump(CDumpContext& dc) const
{
    CFormView::Dump(dc);
}
#endif
#endif // _DEBUG

// CSysBasicManager 消息处理程序

void CSysBasicManager::OnBnClickedButton6()
{
    ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_B, (LPARAM)NM_B, (LPARAM)0);
}

void CSysBasicManager::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    m_showname = CString(sil.FindNode(sil.ShowTemp())->ShowName());
    m_showid = CString(sil.FindNode(sil.ShowTemp())->ShowStuID());
    m_showgender = CString(sil.FindNode(sil.ShowTemp())->ShowGender());
    m_combo2.AddString(TEXT("男"));
    m_combo2.AddString(TEXT("女"));
    m_combo2.SetCurSel(0);
    UpdateData(FALSE);
}

```

```

void CSysBasicManager::OnBnClickedButton5()
{
    sil.Delete_STU(sil.ShowTemp());
    m_showname.Empty();
    m_showid.Empty();
    m_showgender.Empty();
    m_changename.Empty();
    m_changeid.Empty();
    m_combo2.SetCurSel(0);
    UpdateData(FALSE);
    ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_B, (LPARAM)NM_B, (LPARAM)0);
    MessageBox(TEXT("已删除该学生全部信息!"));
}

void CSysBasicManager::OnBnClickedButton4()
{
    m_changename.Empty();
    m_changeid.Empty();
    m_combo2.SetCurSel(0);
    UpdateData(FALSE);
}

void CSysBasicManager::OnBnClickedButton3()
{
    UpdateData(TRUE);
    if (m_changeid.IsEmpty() || m_changename.IsEmpty())
    {
        MessageBox(TEXT("姓名或学号不能为空!"));
        return;
    }
    if (m_changeid.SpanIncluding(TEXT("0123456789")) != m_changeid)
    {
        MessageBox(TEXT("学号需输入纯数字!"));
        return;
    }
    CStringA tmp1, tmp2, tmp3;
    tmp1 = m_changename;
    tmp2 = m_changeid;
    if ((sil.Search_STU(tmp2.GetBuffer(), 1) != -1) && (sil.Search_STU(tmp2.GetBuffer(), 1) !=
sil.ShowTemp()))
    {
        MessageBox(TEXT("该学生学号与已有其他学生学号相同，无法修改!"));
    }
}

```

```

        return;
    }
    int index = m_combo2.GetCurSel();
    CString str;
    m_combo2.GetLBText(index, str);
    tmp3 = str;
    sil.FindNode(sil.ShowTemp())->SetBasicInfo(tmp1.GetBuffer(), tmp3.GetBuffer(),
tmp2.GetBuffer());
    m_showname = CString(sil.FindNode(sil.ShowTemp())->ShowName());
    m_showid = CString(sil.FindNode(sil.ShowTemp())->ShowStuID());
    m_showgender = CString(sil.FindNode(sil.ShowTemp())->ShowGender());
    m_changeid.Empty();
    m_changename.Empty();
    m_combo2.SetCurSel(0);
    UpdateData(FALSE);
    MessageBox(TEXT("修改学生基本信息成功!"));
}

```

## 头文件: SysScore.h

```

#pragma once

// CSysScore 窗体视图

class CSysScore : public CFormView
{
    DECLARE_DYNCREATE(CSysScore)

protected:
    CSysScore();           // 动态创建所使用的受保护的构造函数
    virtual ~CSysScore();

public:
    enum { IDD = IDD_Score };
#ifdef _DEBUG
    virtual void AssertValid() const;
#endif
#ifdef _WIN32_WCE
    virtual void Dump(CDumpContext& dc) const;
#endif
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

    DECLARE_MESSAGE_MAP()
public:

```

```

        afx_msg void OnBnClickedButton1();
private:
    CString m_search1;
};

```

#### 源文件: SysScore.cpp

```

// SysScore.cpp : 实现文件
//

#include "stdafx.h"
#include "StudentScoreSystem.h"
#include "SysScore.h"
#include "MainFrm.h"
#include "head.h"

// CSysScore
extern StuInfoList sil;
IMPLEMENT_DYNCREATE(CSysScore, CFormView)

CSysScore::CSysScore()
    : CFormView(CSysScore::IDD)
    , m_search1(_T(""))
{
}

CSysScore::~CSysScore()
{
}

void CSysScore::DoDataExchange(CDataExchange* pDX)
{
    CFormView::DoDataExchange(pDX);
    DDX_Text(pDX, IDC_EDIT1, m_search1);
}

BEGIN_MESSAGE_MAP(CSysScore, CFormView)
    ON_BN_CLICKED(IDC_BUTTON1, &CSysScore::OnBnClickedButton1)
END_MESSAGE_MAP()

// CSysScore 诊断

#ifdef _DEBUG
void CSysScore::AssertValid() const
{

```



```

        CFormView::AssertValid();
    }

#ifdef _WIN32_WCE
void CSysScore::Dump(CDumpContext& dc) const
{
    CFormView::Dump(dc);
}
#endif
#endif // _DEBUG

// CSysScore 消息处理程序

void CSysScore::OnBnClickedButton1()
{
    if (sil.IsEmpty())
    {
        MessageBox(TEXT("信息库中没有任何学生信息!"));
        return;
    }
    UpdateData(TRUE);
    if (m_search1.IsEmpty())
    {
        MessageBox(TEXT("输入为空!"));
        return;
    }
    else
    {
        CStringA stmpl;
        stmpl = m_search1;
        if (sil.Search_STU(stmpl.GetBuffer(), 1) == -1)
        {
            m_search1.Empty();
            UpdateData(FALSE);
            MessageBox(TEXT("未查询到相关学生!"));
            return;
        }
        else
        {
            sil.SetTemp(sil.Search_STU(stmpl.GetBuffer(), 1));
            ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_F, (LPARAM)NM_F, (LPARAM)0);
        }
    }
}
}

```

头文件: SysScoreList.h

```

#pragma once
#include "afxcmn.h"
// CSysScoreList 窗体视图

class CSysScoreList : public CFormView
{
    DECLARE_DYNCREATE(CSysScoreList)

protected:
    CSysScoreList();           // 动态创建所使用的受保护的构造函数
    virtual ~CSysScoreList();

public:
    enum { IDD = IDD_ScoreList };
#ifdef _DEBUG
    virtual void AssertValid() const;
#endif
#ifdef _WIN32_WCE
    virtual void Dump(CDumpContext& dc) const;
#endif
#ifdef _WIN32_WCE
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

    DECLARE_MESSAGE_MAP()
private:
    CListCtrl m_list1;
public:
    virtual void OnInitialUpdate();
    afx_msg void OnLvnItemchangedList2(NMHDR *pNMHDR, LRESULT *pResult);
private:
    CString m_listsearch;
public:
    afx_msg void OnBnClickedButton1();
private:
    CString m_showperson;
};

源文件: SysScoreList.cpp
// SysScoreList.cpp : 实现文件
//

#include "stdafx.h"
#include "StudentScoreSystem.h"
#include "SysScoreList.h"
#include "head.h"

```

```

// CSysScoreList
extern StuInfoList sil;
IMPLEMENT_DYNCREATE(CSysScoreList, CFormView)

CSysScoreList::CSysScoreList()
    : CFormView(CSysScoreList::IDD)
    , m_listsearch(_T(""))
    , m_showperson(_T(""))
{
}

CSysScoreList::~CSysScoreList()
{
}

void CSysScoreList::DoDataExchange(CDataExchange* pDX)
{
    CFormView::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_LIST2, m_list1);
    DDX_Text(pDX, IDC_EDIT1, m_listsearch);
    DDX_Text(pDX, IDC_EDIT3, m_showperson);
}

BEGIN_MESSAGE_MAP(CSysScoreList, CFormView)
    ON_NOTIFY(LVN_ITEMCHANGED, IDC_LIST2, &CSysScoreList::OnLvnItemchangedList2)
    ON_BN_CLICKED(IDC_BUTTON1, &CSysScoreList::OnBnClickedButton1)
END_MESSAGE_MAP()

// CSysScoreList 诊断

#ifdef _DEBUG
void CSysScoreList::AssertValid() const
{
    CFormView::AssertValid();
}

#endif

#ifdef _WIN32_WCE
void CSysScoreList::Dump(CDumpContext& dc) const
{
    CFormView::Dump(dc);
}

#endif
#endif // _DEBUG

```

```
// CSysScoreList 消息处理程序
```

```
void CSysScoreList::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    //设置风格
    m_list1.SetExtendedStyle(m_list1.GetExtendedStyle() | LVS_EX_FULLROWSELECT |
LVS_EX_GRIDLINES);
    //设置表头
    CString column[] = { TEXT("课程名称"), TEXT("学分"), TEXT("百分制成绩"), TEXT("绩点"),
TEXT("对应等级"), TEXT("录入时间") };
    m_list1.InsertColumn(0, column[0], LVCFMT_CENTER, 150);
    m_list1.InsertColumn(1, column[1], LVCFMT_CENTER, 80);
    m_list1.InsertColumn(2, column[2], LVCFMT_CENTER, 100);
    m_list1.InsertColumn(3, column[3], LVCFMT_CENTER, 80);
    m_list1.InsertColumn(4, column[4], LVCFMT_CENTER, 80);
    m_list1.InsertColumn(5, column[5], LVCFMT_CENTER, 100);
}
```

```
void CSysScoreList::OnLvnItemchangedList2(NMHDR *pNMHDR, LRESULT *pResult)
{
    LPNMLISTVIEW pNMLV = reinterpret_cast<LPNMLISTVIEW>(pNMHDR);
    *pResult = 0;
}
```

```
void CSysScoreList::OnBnClickedButton1()
{
    UpdateData(TRUE);
    if (sil.IsEmpty())
    {
        MessageBox(TEXT("信息库中没有任何学生信息!"));
        return;
    }
    if (m_listsearch.IsEmpty())
    {
        MessageBox(TEXT("输入为空!"));
        return;
    }
    CStringA ltmp1;
    ltmp1 = m_listsearch;
```

```

if (sil.Search_STU(ltmp1.GetBuffer(), 1) == -1)
{
    m_listsearch.Empty();
    UpdateData(FALSE);
    MessageBox(TEXT("未查询到相关学生!"));
    return;
}
if (sil.Search_STU(ltmp1.GetBuffer()) ->IsEmpty())
{
    m_listsearch.Empty();
    UpdateData(FALSE);
    MessageBox(TEXT("未查询到该学生任何成绩!"));
    return;
}
//打印成绩
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().ScoreSort();//待测试
m_list1.DeleteAllItems();
for (int num = 0; num < sil.Search_STU(ltmp1.GetBuffer())->showscorelist().coursenumber();
num++)
{
    m_list1.InsertItem(num,
CString(sil.Search_STU(ltmp1.GetBuffer())->showscorelist().Search(num)->ShowSubject()));
    CString c11, c12, c13, c15;
    c11.Format(TEXT("%d"),
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().Search(num)->ShowCredit());
    c12.Format(TEXT("%d"),
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().Search(num)->ShowScore());
    if (sil.Search_STU(ltmp1.GetBuffer())->showscorelist().Search(num)->ShowGrade1() == -
1)
    {
        c13.Format(TEXT("记pf"));
    }
    else
    {
        c13.Format(TEXT("%.1f"),
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().Search(num)->ShowGrade1());
    }
    c15.Format(TEXT("%d-%d-%d"),
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().Search(num)->ShowYear(),
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().Search(num)->ShowMonth(),
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().Search(num)->ShowDay());
    m_list1.SetItemText(num, 1, c11);
    m_list1.SetItemText(num, 2, c12);
    m_list1.SetItemText(num, 3, c13);
    m_list1.SetItemText(num, 4,

```

```

CString(sil.Search_STU(ltmp1.GetBuffer())->showscorelist().Search(num)->ShowGrade2()));
    m_list1.SetItemText(num, 5, c15);
}
CString slt;
if (sil.Search_STU(ltmp1.GetBuffer())->personGPA() == -1)
{
    slt.Format(TEXT("该学生共修%d门课程，其中通过课程%d门共%d学分，平均绩点为NULL"),
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().coursenumber(),
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().PassCourse(),
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().PassCredit());
}
else
{
    slt.Format(TEXT("该学生共修%d门课程，其中通过课程%d门共%d学分，平均绩点为%.2f"),
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().coursenumber(),
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().PassCourse(),
sil.Search_STU(ltmp1.GetBuffer())->showscorelist().PassCredit(), sil.Search_STU(ltmp1.GetBuffer(
))->personGPA());
}
m_showperson = slt;
UpdateData(FALSE);
}

```

## 头文件：SysScoreManager.h

```

#pragma once
#include "afxwin.h"
// CSysScoreManager 窗体视图

class CSysScoreManager : public CFormView
{
    DECLARE_DYNCREATE(CSysScoreManager)

protected:
    CSysScoreManager();           // 动态创建所使用的受保护的构造函数
    virtual ~CSysScoreManager();

public:
    enum { IDD = IDD_ScoreManager };
#ifdef _DEBUG
    virtual void AssertValid() const;
#endif
#ifdef _WIN32_WCE
    virtual void Dump(CDumpContext& dc) const;
#endif
#ifdef _DEBUG
    virtual void Dump(CDumpContext& dc) const;
#endif
}

```

```

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

    DECLARE_MESSAGE_MAP()
public:
    afx_msg void OnBnClickedButton1();
    virtual void OnInitialUpdate();
private:
    CString m_showname1;
    CString m_showidl;
    CString m_showgender1;
    CComboBox m_combo3;
    CString m_newcoursename;
    int m_newcredit;
    int m_newcoursescore;
    int m_newyear;
    int m_newmonth;
    int m_newday;
public:
    afx_msg void OnBnClickedButton6();
    afx_msg void OnBnClickedButton5();
private:
    CString m_searchcourse;
    CString m_showcoursename;
    int m_showcredit;
    int m_showcoursescore;
    CString m_showpf;
    int m_showyear;
    int m_showmonth;
    int m_showday;
    CString m_changecoursename;
    int m_changecredit;
    int m_changecoursescore;
    CComboBox m_combo4;
    int m_changeyear;
    int m_changemonth;
    int m_changeday;
public:
    afx_msg void OnBnClickedButton12();
    afx_msg void OnBnClickedButton13();
    afx_msg void OnBnClickedButton11();
    afx_msg void OnBnClickedButton7();
    afx_msg void OnBnClickedButton8();
};

```

源文件: SysScoreManager.cpp

```

// SysScoreManager.cpp : 实现文件
//

#include "stdafx.h"
#include "StudentScoreSystem.h"
#include "SysScoreManager.h"
#include "MainFrm.h"
#include "head.h"

extern StuInfoList sil;
// CSysScoreManager

IMPLEMENT_DYNCREATE(CSysScoreManager, CFormView)

CSysScoreManager::CSysScoreManager()
    : CFormView(CSysScoreManager::IDD)
    , m_showname1(_T(""))
    , m_showid1(_T(""))
    , m_showgender1(_T(""))
    , m_newcoursename(_T(""))
    , m_newcredit(0)
    , m_newcoursescore(0)
    , m_newyear(0)
    , m_newmonth(0)
    , m_newday(0)
    , m_searchcourse(_T(""))
    , m_showcoursename(_T(""))
    , m_showcredit(0)
    , m_showcoursescore(0)
    , m_showpf(_T(""))
    , m_showyear(0)
    , m_showmonth(0)
    , m_showday(0)
    , m_changecoursename(_T(""))
    , m_changecredit(0)
    , m_changecoursescore(0)
    , m_changeyear(0)
    , m_changemonth(0)
    , m_changeday(0)
{
}

CSysScoreManager::~CSysScoreManager()
{
}

```



```
}
```

```
void CSysScoreManager::DoDataExchange(CDataExchange* pDX)
```

```
{  
    CFormView::DoDataExchange(pDX);  
    DDX_Text(pDX, IDC_EDIT1, m_showname1);  
    DDX_Text(pDX, IDC_EDIT3, m_showid1);  
    DDX_Text(pDX, IDC_EDIT6, m_showgender1);  
    DDX_Control(pDX, IDC_COMBO2, m_combo3);  
    DDX_Text(pDX, IDC_EDIT11, m_newcoursename);  
    DDX_Text(pDX, IDC_EDIT9, m_newcredit);  
    DDX_Text(pDX, IDC_EDIT10, m_newcoursescore);  
    DDX_Text(pDX, IDC_EDIT2, m_newyear);  
    DDX_Text(pDX, IDC_EDIT7, m_newmonth);  
    DDX_Text(pDX, IDC_EDIT8, m_newday);  
    DDX_Text(pDX, IDC_EDIT19, m_searchcourse);  
    DDX_Text(pDX, IDC_EDIT17, m_showcoursename);  
    DDX_Text(pDX, IDC_EDIT15, m_showcredit);  
    DDX_Text(pDX, IDC_EDIT16, m_showcoursescore);  
    DDX_Text(pDX, IDC_EDIT18, m_showpf);  
    DDX_Text(pDX, IDC_EDIT12, m_showyear);  
    DDX_Text(pDX, IDC_EDIT13, m_showmonth);  
    DDX_Text(pDX, IDC_EDIT14, m_showday);  
    DDX_Text(pDX, IDC_EDIT25, m_changecoursename);  
    DDX_Text(pDX, IDC_EDIT23, m_changecredit);  
    DDX_Text(pDX, IDC_EDIT24, m_changecoursescore);  
    DDX_Control(pDX, IDC_COMBO3, m_combo4);  
    DDX_Text(pDX, IDC_EDIT20, m_changeyear);  
    DDX_Text(pDX, IDC_EDIT21, m_changemonth);  
    DDX_Text(pDX, IDC_EDIT22, m_changeday);  
}
```

```
BEGIN_MESSAGE_MAP(CSysScoreManager, CFormView)
```

```
    ON_BN_CLICKED(IDC_BUTTON1, &CSysScoreManager::OnBnClickedButton1)  
    ON_BN_CLICKED(IDC_BUTTON6, &CSysScoreManager::OnBnClickedButton6)  
    ON_BN_CLICKED(IDC_BUTTON5, &CSysScoreManager::OnBnClickedButton5)  
    ON_BN_CLICKED(IDC_BUTTON12, &CSysScoreManager::OnBnClickedButton12)  
    ON_BN_CLICKED(IDC_BUTTON13, &CSysScoreManager::OnBnClickedButton13)  
    ON_BN_CLICKED(IDC_BUTTON11, &CSysScoreManager::OnBnClickedButton11)  
    ON_BN_CLICKED(IDC_BUTTON7, &CSysScoreManager::OnBnClickedButton7)  
    ON_BN_CLICKED(IDC_BUTTON8, &CSysScoreManager::OnBnClickedButton8)  
END_MESSAGE_MAP()
```

```
// CSysScoreManager 诊断
```

```

#ifdef _DEBUG
void CSysScoreManager::AssertValid() const
{
    CFormView::AssertValid();
}

#endif

#ifdef _WIN32_WCE
void CSysScoreManager::Dump(CDumpContext& dc) const
{
    CFormView::Dump(dc);
}
#endif

// CSysScoreManager 消息处理程序

void CSysScoreManager::OnBnClickedButton1()
{
    ::PostMessage(AfxGetMainWnd()->GetSafeHwnd(), NM_C, (WPARAM)NM_C, (LPARAM)0);
}

void CSysScoreManager::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    m_showname1 = CString(sil.FindNode(sil.ShowTemp())->ShowName());
    m_showid1 = CString(sil.FindNode(sil.ShowTemp())->ShowStuID());
    m_showgender1 = CString(sil.FindNode(sil.ShowTemp())->ShowGender());
    m_showcoursename.Empty();
    m_combo3.AddString(TEXT("是"));
    m_combo3.AddString(TEXT("否"));
    m_combo3.SetCurSel(0);
    m_newday = 1;
    m_newmonth = 1;
    m_newyear = 2019;
    m_combo4.AddString(TEXT("是"));
    m_combo4.AddString(TEXT("否"));
    m_combo4.SetCurSel(0);
    m_changeday = 1;
    m_changemonth = 1;
    m_changeyear = 2019;
    m_showday = 1;
    m_showmonth = 1;
    m_showyear = 2019;
}

```

```

        UpdateData(FALSE);
    }

```

```

void CSysScoreManager::OnBnClickedButton6()
{
    m_newcoursename.Empty();
    m_newcoursescore = 0;
    m_newcredit = 0;
    m_newday = 1;
    m_newmonth = 1;
    m_newyear = 2019;
    m_combo3.SetCurSel(0);
    UpdateData(FALSE);
}

```

```

void CSysScoreManager::OnBnClickedButton5()
{
    UpdateData(TRUE);
    if (m_newcoursename.IsEmpty())
    {
        MessageBox(TEXT("课程名为空!"));
        return;
    }
    if (m_newcredit < 1)
    {
        MessageBox(TEXT("学分数需为大于等于1的整数!"));
        return;
    }
    if (m_newcoursescore < 0 || m_newcoursescore > 100)
    {
        MessageBox(TEXT("成绩需为0至100间的整数!"));
        return;
    }
    if (m_newyear < 1)
    {
        MessageBox(TEXT("年份输入错误!"));
        return;
    }
    if (m_newmonth < 1 || m_newmonth > 12)
    {
        MessageBox(TEXT("月份输入错误!"));
        return;
    }
}

```

```

        if (daywrong(m_newyear, m_newmonth, m_newday))
        {
            MessageBox(TEXT("日期输入错误!"));
            return;
        }
        CStringA ntmp;
        ntmp = m_newcoursename;
        int index = m_combo3.GetCurSel();
        CString str;
        m_combo3.GetLBText(index, str);
        if (sil.FindNode(sil.ShowTemp())->SearchScore(ntmp.GetBuffer(), 1) != -1)
        {
            MessageBox(TEXT("该课程名与已有课程名相同, 无法添加!"));
            return;
        }
        if (str == TEXT("是"))
        {
            sil.FindNode(sil.ShowTemp())->AddScore(ntmp.GetBuffer(), m_newcoursescore,
            m_newcredit, m_newyear, m_newmonth, m_newday, 0);
        }
        else
        {
            sil.FindNode(sil.ShowTemp())->AddScore(ntmp.GetBuffer(), m_newcoursescore,
            m_newcredit, m_newyear, m_newmonth, m_newday, 1);
        }
        m_newcoursename.Empty();
        m_newcoursescore = 0;
        m_newcredit = 0;
        m_newday = 1;
        m_newmonth = 1;
        m_newyear = 2019;
        m_combo3.SetCurSel(0);
        UpdateData(FALSE);
        MessageBox(TEXT("新成绩添加成功!"));
    }

void CSysScoreManager::OnBnClickedButton12()
{
    m_searchcourse.Empty();
    m_showcoursename.Empty();
    m_showcredit = 0;
    m_showcoursescore = 0;
    m_showpf.Empty();
    m_changecoursename.Empty();
}

```

```

        m_changecoursescore = 0;
        m_changecredit = 0;
        m_combo4.SetCurSel(0);
        m_changeday = 1;
        m_changemonth = 1;
        m_changeyear = 2019;
        m_showday = 1;
        m_showmonth = 1;
        m_showyear = 2019;
        UpdateData(FALSE);
    }

```

```

void CSysScoreManager::OnBnClickedButton13()
{
    m_changecoursename.Empty();
    m_changecoursescore = 0;
    m_changecredit = 0;
    m_combo4.SetCurSel(0);
    m_changeday = 1;
    m_changemonth = 1;
    m_changeyear = 2019;
    UpdateData(FALSE);
}

```

```

void CSysScoreManager::OnBnClickedButton11()
{
    UpdateData(TRUE);
    if (sil.FindNode(sil.ShowTemp())->IsEmpty())
    {
        MessageBox(TEXT("该学生成绩数据为空，无法进行查询！"));
        return;
    }
    if (m_searchcourse.IsEmpty())
    {
        MessageBox(TEXT("查询课程名为空！"));
        return;
    }
    CStringA ntmp2;
    ntmp2 = m_searchcourse;
    if (sil.FindNode(sil.ShowTemp())->SearchScore(ntmp2.GetBuffer(), 1) == -1)
    {
        MessageBox(TEXT("未查询到该课程！"));
        return;
    }
}

```

```

    }
    m_showcoursename =
CString(sil.FindNode(sil.ShowTemp())->SearchScore(ntmp2.GetBuffer())->ShowSubject());
    m_showcredit = sil.FindNode(sil.ShowTemp())->SearchScore(ntmp2.GetBuffer())->ShowCredit();
    m_showcoursescore =
sil.FindNode(sil.ShowTemp())->SearchScore(ntmp2.GetBuffer())->ShowScore();
    m_showyear = sil.FindNode(sil.ShowTemp())->SearchScore(ntmp2.GetBuffer())->ShowYear();
    m_showmonth = sil.FindNode(sil.ShowTemp())->SearchScore(ntmp2.GetBuffer())->ShowMonth();
    m_showday = sil.FindNode(sil.ShowTemp())->SearchScore(ntmp2.GetBuffer())->ShowDay();
    if (sil.FindNode(sil.ShowTemp())->SearchScore(ntmp2.GetBuffer())->ShowPF() == 1)
    {
        m_showpf = CString("否");
    }
    else
    {
        m_showpf = CString("是");
    }
    m_searchcourse.Empty();
    UpdateData(FALSE);
}

```

```

void CSysScoreManager::OnBnClickedButton7()
{
    if (m_showcoursename.IsEmpty())
    {
        MessageBox(TEXT("请先进行查询!"));
        return;
    }
    CStringA ntmp3;
    ntmp3 = m_showcoursename;
    sil.FindNode(sil.ShowTemp())->DeleteScore(ntmp3.GetBuffer());
    m_searchcourse.Empty();
    m_showcoursename.Empty();
    m_showcredit = 0;
    m_showcoursescore = 0;
    m_showpf.Empty();
    m_changecoursename.Empty();
    m_changecoursescore = 0;
    m_changecredit = 0;
    m_combo4.SetCurSel(0);
    m_changeday = 1;
    m_changemonth = 1;
    m_changeyear = 2019;
    m_showday = 1;

```

```

        m_showmonth = 1;
        m_showyear = 2019;
        UpdateData(FALSE);
        MessageBox(TEXT("删除成功!"));
    }

```

```

void CSysScoreManager::OnBnClickedButton8()
{
    UpdateData(TRUE);
    if (m_showcoursename.IsEmpty())
    {
        MessageBox(TEXT("请先进行查询!"));
        return;
    }
    CStringA ntmp4;
    ntmp4 = m_showcoursename;

    if (m_changecoursename.IsEmpty())
    {
        MessageBox(TEXT("课程名为空!"));
        return;
    }
    if (m_changecredit < 1)
    {
        MessageBox(TEXT("学分数需为大于等于1的整数!"));
        return;
    }
    if (m_changecoursescore < 0 || m_changecoursescore>100)
    {
        MessageBox(TEXT("成绩需为0至100间的整数!"));
        return;
    }
    if (m_changeyear < 1)
    {
        MessageBox(TEXT("年份输入错误!"));
        return;
    }
    if (m_changemonth < 1 || m_changemonth>12)
    {
        MessageBox(TEXT("月份输入错误!"));
        return;
    }
    if (daywrong(m_changeyear, m_changemonth, m_changeday))
    {

```

```

        MessageBox(TEXT("日期输入错误!"));
        return;
    }
    CStringA ntmp5;
    ntmp5 = m_changecoursename;
    int index = m_combo3.GetCurSel();
    CString str;
    m_combo4.GetLBText(index, str);
    if (str == TEXT("是"))
    {

        sil.FindNode(sil.ShowTemp())->SearchScore(ntmp4.GetBuffer())->SetScore(ntmp5.GetBuffer(),
m_changecoursescore, m_changecredit, m_changeyear, m_changemonth, m_changeday, 0);
    }
    else
    {

        sil.FindNode(sil.ShowTemp())->SearchScore(ntmp4.GetBuffer())->SetScore(ntmp5.GetBuffer(),
m_changecoursescore, m_changecredit, m_changeyear, m_changemonth, m_changeday, 1);
    }
    m_changecoursename.Empty();
    m_changecoursescore = 0;
    m_changecredit = 0;
    m_changeday = 1;
    m_changemonth = 1;
    m_changeyear = 2019;
    m_combo4.SetCurSel(0);
    m_showcoursename =
CString(sil.FindNode(sil.ShowTemp())->SearchScore(ntmp5.GetBuffer())->ShowSubject());
    m_showcredit = sil.FindNode(sil.ShowTemp())->SearchScore(ntmp5.GetBuffer())->ShowCredit();
    m_showcoursescore =
sil.FindNode(sil.ShowTemp())->SearchScore(ntmp5.GetBuffer())->ShowScore();
    m_showyear = sil.FindNode(sil.ShowTemp())->SearchScore(ntmp5.GetBuffer())->ShowYear();
    m_showmonth = sil.FindNode(sil.ShowTemp())->SearchScore(ntmp5.GetBuffer())->ShowMonth();
    m_showday = sil.FindNode(sil.ShowTemp())->SearchScore(ntmp5.GetBuffer())->ShowDay();
    if (sil.FindNode(sil.ShowTemp())->SearchScore(ntmp5.GetBuffer())->ShowPF() == 1)
    {
        m_showpf = CString("否");
    }
    else
    {
        m_showpf = CString("是");
    }
    m_searchcourse.Empty();
    UpdateData(FALSE);

```



```

        MessageBox(TEXT("成绩修改成功!"));
    }

```

### 头文件: SysTotal.h

```

#pragma once
#include "afxcmn.h"
// CSysTotal 窗体视图

class CSysTotal : public CFormView
{
    DECLARE_DYNCREATE(CSysTotal)

protected:
    CSysTotal();           // 动态创建所使用的受保护的构造函数
    virtual ~CSysTotal();

public:
    enum { IDD = IDD_Total };
#ifdef _DEBUG
        virtual void AssertValid() const;
#endif
#ifdef _WIN32_WCE
        virtual void Dump(CDumpContext& dc) const;
#endif
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

    DECLARE_MESSAGE_MAP()
private:
    CListCtrl m_tlist;
public:
    virtual void OnInitialUpdate();
    afx_msg void OnLvnItemchangedList1(NMHDR *pNMHDR, LRESULT *pResult);
private:
    CString m_totalshow;
};

```

### 源文件: SysTotal.cpp

```

// SysTotal.cpp : 实现文件
//

#include "stdafx.h"
#include "StudentScoreSystem.h"
#include "SysTotal.h"
#include "head.h"

```

```

#include "InfoFile.h"

// CSysTotal
extern StuInfoList sil;
IMPLEMENT_DYNCREATE(CSysTotal, CFormView)

CSysTotal::CSysTotal()
    : CFormView(CSysTotal::IDD)
    , m_totalshow(_T(""))
{
}

CSysTotal::~~CSysTotal()
{
}

void CSysTotal::DoDataExchange(CDataExchange* pDX)
{
    CFormView::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_LIST1, m_tlist);
    DDX_Text(pDX, IDC_EDIT1, m_totalshow);
}

BEGIN_MESSAGE_MAP(CSysTotal, CFormView)
END_MESSAGE_MAP()

// CSysTotal 诊断

#ifdef _DEBUG
void CSysTotal::AssertValid() const
{
    CFormView::AssertValid();
}

#endif

#ifdef _WIN32_WCE
void CSysTotal::Dump(CDumpContext& dc) const
{
    CFormView::Dump(dc);
}
#endif

#endif // _DEBUG

// CSysTotal 消息处理程序

```

```

void CSysTotal::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    //设置风格
    m_tlist.SetExtendedStyle(m_tlist.GetExtendedStyle() | LVS_EX_FULLROWSELECT |
LVS_EX_GRIDLINES);
    //设置表头
    CString column[] = { TEXT("姓名"), TEXT("学号"), TEXT("性别"), TEXT("均绩"), TEXT("完成学分
数"), TEXT("完成课程数") };
    m_tlist.InsertColumn(0, column[0], LVCFMT_CENTER, 100);
    m_tlist.InsertColumn(1, column[1], LVCFMT_CENTER, 150);
    m_tlist.InsertColumn(2, column[2], LVCFMT_CENTER, 50);
    m_tlist.InsertColumn(3, column[3], LVCFMT_CENTER, 100);
    m_tlist.InsertColumn(4, column[4], LVCFMT_CENTER, 100);
    m_tlist.InsertColumn(5, column[5], LVCFMT_CENTER, 100);
    if (sil.IsEmpty())
    {
        MessageBox(TEXT("信息库中没有任何学生信息!"));
        return;
    }
    for (int number=0; number < sil.showtotalnumber(); number++)
    {
        m_tlist.InsertItem(number, CString(sil.FindNode(number)->ShowName()));
        m_tlist.SetItemText(number, 1, CString(sil.FindNode(number)->ShowStuID()));
        m_tlist.SetItemText(number, 2, CString(sil.FindNode(number)->ShowGender()));
        CString t13, t14, t15;
        if (sil.FindNode(number)->IsEmpty())
        {
            t13.Format(TEXT("NULL"));
            m_tlist.SetItemText(number, 3, t13);
            t14.Format(TEXT("NULL"));
            m_tlist.SetItemText(number, 4, t14);
            t15.Format(TEXT("NULL"));
            m_tlist.SetItemText(number, 5, t15);
        }
        else
        {
            if (sil.FindNode(number)->personGPA() == -1)
            {
                t13.Format(TEXT("NULL"));
            }
            else
            {
                t13.Format(TEXT("%.2f"), sil.FindNode(number)->personGPA());
            }
        }
    }
}

```

```

        m_tlist.SetItemText(number, 3, t13);
        t14.Format(TEXT("%d"), sil.FindNode(number)->showscorelist().PassCredit());
        m_tlist.SetItemText(number, 4, t14);
        t15.Format(TEXT("%d"), sil.FindNode(number)->showscorelist().PassCourse());
        m_tlist.SetItemText(number, 5, t15);
    }

}

CString ts;
if (sil.averageGPA() == -1)
{
    ts.Format(TEXT("学生成绩管理系统中总计有%d名学生，全体学生的平均绩点为NULL"),
sil.showtotalnumber());
}
else
{
    ts.Format(TEXT("学生成绩管理系统中总计有%d名学生，全体学生的平均绩点
为%.2f"), sil.showtotalnumber(), sil.averageGPA());
}
m_totalshow = ts;
UpdateData(FALSE);
}

```

## 头文件: UserDlg.h

```

#pragma once
// CUserDlg 窗体视图

class CUserDlg : public CFormView
{
    DECLARE_DYNCREATE(CUserDlg)

protected:
    CUserDlg();           // 动态创建所使用的受保护的构造函数
    virtual ~CUserDlg();

public:
    enum { IDD = IDD_USER };
#ifdef _DEBUG
    virtual void AssertValid() const;
#endif
#ifdef _WIN32_WCE
    virtual void Dump(CDumpContext& dc) const;
#endif
#ifdef _DEBUG
    virtual void Dump(CDumpContext& dc) const;
#endif

protected:

```

```

        virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

        DECLARE_MESSAGE_MAP()
private:
        CString m_username;
        CString m_newpwd;
        CString m_confpwd;
public:
        virtual void OnInitUpdate();
        afx_msg void OnBnClickedButton1();
        afx_msg void OnBnClickedButton2();
};

```

### 源文件: UserDlg.cpp

```

// UserDlg.cpp : 实现文件
#include "stdafx.h"
#include "StudentScoreSystem.h"
#include "UserDlg.h"
#include "InfoFile.h"
#include "head.h"

// CUserDlg
extern StuInfoList sil;
IMPLEMENT_DYNCREATE(CUserDlg, CFormView)

CUserDlg::CUserDlg()
    : CFormView(CUserDlg::IDD)
    , m_username(_T(""))
    , m_newpwd(_T(""))
    , m_confpwd(_T(""))
{
}

CUserDlg::~CUserDlg()
{
}

void CUserDlg::DoDataExchange(CDataExchange* pDX)
{
    CFormView::DoDataExchange(pDX);
    DDX_Text(pDX, IDC_EDIT1, m_username);
    DDX_Text(pDX, IDC_EDIT3, m_newpwd);
    DDX_Text(pDX, IDC_EDIT4, m_confpwd);
}

BEGIN_MESSAGE_MAP(CUserDlg, CFormView)
    ON_BN_CLICKED(IDC_BUTTON1, &CUserDlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON2, &CUserDlg::OnBnClickedButton2)

```

```

END_MESSAGE_MAP()
// CUserDlg 诊断
#ifdef _DEBUG
void CUserDlg::AssertValid() const
{
    CFormView::AssertValid();
}
#endif
#ifdef _WIN32_WCE
void CUserDlg::Dump(CDumpContext& dc) const
{
    CFormView::Dump(dc);
}
#endif
// CUserDlg 消息处理程序
void CUserDlg::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();

    CInfoFile fp;
    CString f_name, f_pwd;
    fp.ReadLogin(f_name, f_pwd);
    m_username = f_name;
    UpdateData(FALSE);
}

void CUserDlg::OnBnClickedButton1()
{
    UpdateData(TRUE);
    CInfoFile fp;
    CString f_name, f_pwd;
    fp.ReadLogin(f_name, f_pwd);
    m_username = f_name;
    if (m_newpwd.IsEmpty() || m_confpwd.IsEmpty())
    {
        MessageBox(TEXT("新的密码或确认密码不能为空!"));
        return;
    }
    else if (m_newpwd != m_confpwd)
    {
        MessageBox(TEXT("新的密码与确认密码不一致!"));
        return;
    }
    else if (m_newpwd == f_pwd)
    {
        MessageBox(TEXT("新的密码需与旧密码不一样!"));
    }
}

```

```

        return;
    }
    //写入新密码，需要转化
    CStringA temp1, temp2;
    temp1 = f_name;
    temp2 = m_confpwd;
    fp.WritePw(temp1.GetBuffer(), temp2.GetBuffer());
    MessageBox(TEXT("密码修改成功!"));
    //清空内容
    m_newpwd.Empty();
    m_confpwd.Empty();
    UpdateData(FALSE);
}

void CUserDlg::OnBnClickedButton2()
{
    m_newpwd.Empty();
    m_confpwd.Empty();
    UpdateData(FALSE);
}

```

## 第 1 题评分表：

项 目	评 价	
设计方案的合理性与创新性	3	
设计与调试结果	4	
设计说明书的质量	1	
程序基本要求涵盖情况	4	
程序代码编写素养情况	2	
课程设计周表现情况	1	
综合成绩	15	

助教签名：\_\_\_\_\_

日 期：\_\_\_\_\_