

课 程 设 计 报 告

课程名称 计算机程序设计基础 2

班 级 无 26

学 号 2022010645

姓 名 李卓航

2023 年 7 月 7 日

目 录

一、	系统需求分析	3
二、	总体设计	3
三、	详细设计	4
1.	类的设计	6
2.	功能的实现	7
3.	图形界面的设计	11
四、	调试过程	11
五、	最终结果功能及使用说明	13
六、	总结	23
附录一：	源程序清单	24
附录二：	大作业任务书	48
1.	课程设计目的	48
2.	课题题目	48
3.	文档设计要求	48
4.	程序设计的基本要求	48
5.	进度安排	49
附录三：	评分表	50

一、 系统需求分析

学生成绩管理系统基于本校的成绩考核记录机制，分为等级制和记 pf 制两种记录方法，按照本校对应百分比成绩与等级制成绩的换算关系折算为相应的成绩并对应相应的绩点。成绩里记录了指定学生指定成绩的信息，它包括：课程名称、pf 属性（用 0 表示等级制，1 表示 pf 制）、学分、绩点、成绩与百分制成绩。同时系统里还记录了学生的一些个人信息，例如年级、班号、专业、性别、学号，为此一个基本的成绩管理系统应实现以下功能：

（1）录入学生的个人信息以及成绩：需要事先录入学生的个人信息，再依照学号录入指定学期的课程的各项属性并依照百分制成绩按照对应的规则折合成对应的成绩与绩点，并计算出学生这一学期的总学分绩和总学分和 GPA。

（2）查询学生的成绩：需要在确定查询的学期后读取系统存储的数据显示该学生的个人信息，按照成绩表的格式显示该学期所有课程的成绩。

（3）显示学生的排名与总 GPA：要求系统访问指定学生所有学期的成绩，将所有学分绩与学分分别相加计算得到总的 GPA，并按照相同年级，同年级同专业，同班三个由大到小的维度进行 GPA 排序得到级排名、系排名和班排名并分别显示出来，并显示总 GPA。

（4）修改学生的成绩：在录入、改卷等过程致使课程对应属性或成绩出现问题时，可以及时访问系统并加以修正，得到新的成绩信息、GPA 与排名。

（5）储存录入的数据：在录入信息后存储信息到指定的文件中便于后续直接打开。

（6）登录界面设定：依照实际情况设定为管理员端口与学生端口，管理员端口具有以上所有功能，学生端口只能查看成绩和排名。

二、 总体设计

学生成绩管理系统包含四个大的功能，分别是：录入学生信息和成绩、修改学生成绩、查询某个学生某学期成绩、计算某一学生 GPA 和排名。总体成绩换算机制，GPA 计算方法等详细成绩规则以本校规则为模板制定。

为此用户可以根据第五部分的使用说明与菜单上的相关提示来进行操作，总体功能设计思路可见下图：

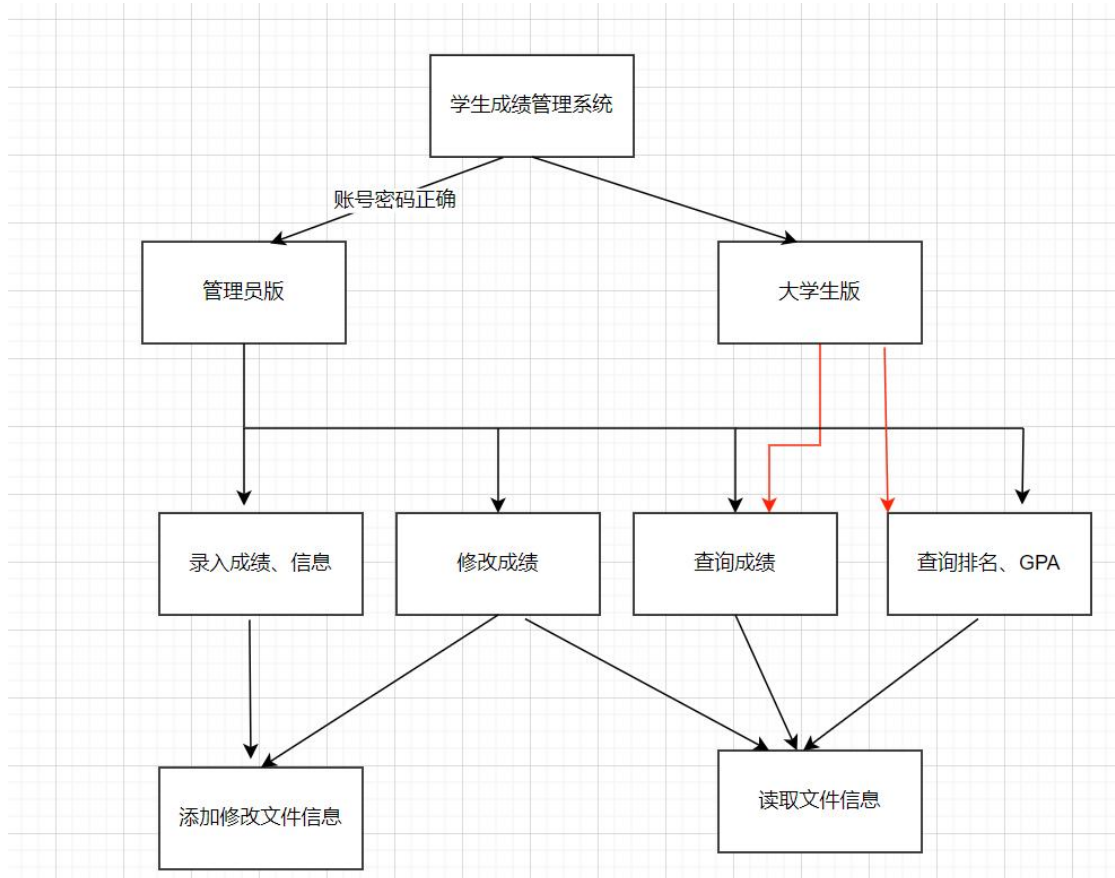


图 1 总体功能设计框架

三、 详细设计

设计类的 UML 图包含了所有类的属性及方法，如下图所示：

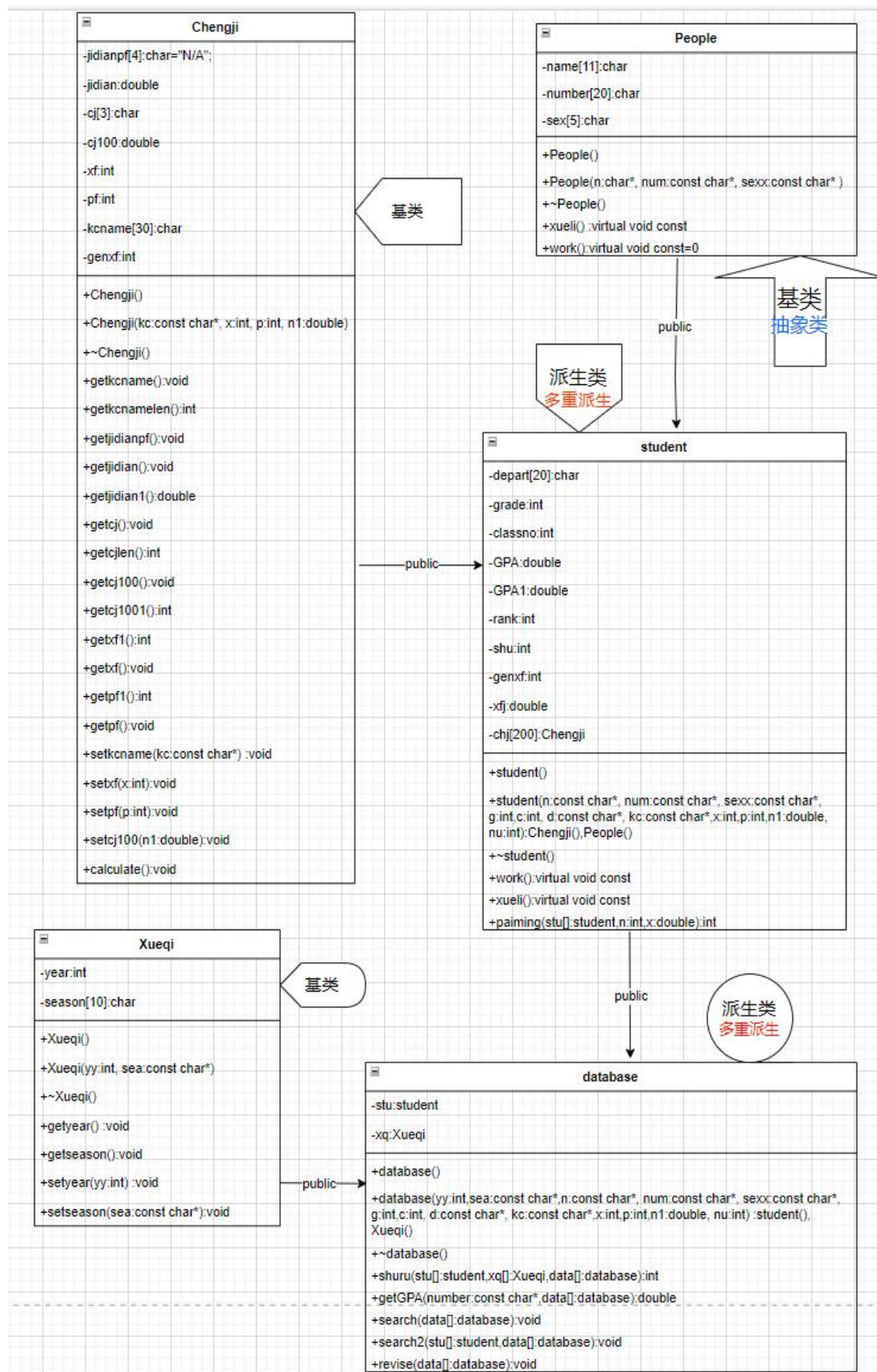


图 2 类的 UML 图

1. 类的设计

按照功能的需求设计了图 2 所示的五个类和一个名为 `adminis` 的结构体，如图：

```
struct adminis
{
    char name[20];
    int gonghao;
    char userid[30];
    char password[50];
}; //定义管理员结构体
```

图 3 `adminis` 结构体

该结构体主要包含了 `name`（姓名）、`gonghao`（工号）、`userid`（用户名）、`password`（密码）四个属性，用以登录管理员界面。

在这五个类中，首先有三个互不相干的基类：

`People` 类主要作为 `student` 的基类和一个抽象类，含有最基本的 `name`、`number`、`sex` 属性并构造了虚函数 `xueli()`（调用时与 `student` 类的 `xueli()` 相区分）和 `work()`（纯虚函数，供派生类具体化）以及自身带有的构造函数与析构函数。

`Xueqi` 类含有两个属性：`year` 和 `season` 用以确定指定的学期（如 2023 学年春季学期），除了自身带有的构造函数与析构函数还含有四个函数用以访问、设定该类中的 `protected` 成员：`season` 和 `year`。

`Chengji` 类含有八个属性，但其中只有 `kcname`（课程名）、`xf`（学分）、`pf`（`pf` 属性）、`cj100`（百分制成绩）是自行确定的，其余的 `jidian`、`jidianpf`、`cj`、`genxf` 是通过 `calculate` 函数基于自行确定的四个属性按照指定规则来设定的，不能自由更改；其次就是构造析构函数，读取更改 `protected` 成员的函数。

`student` 类由 `People` 和 `Chengji` 以公有方式派生而来，在 `People` 基础上增加了 `depart`、`grade`、`classno`、`GPA(1)`、`rank`、`shu`、`genxf`、`xfj` 学生特有的属性并引用 `Chengji` 类 `chj[]` 作为对象数组存储学生的成绩信息，在含有构造析构函数外实例化了抽象类的纯虚函数和虚函数，并在 `paiming` 函数中被引用。该类主要是用来存储总 `GPA` 并在所有学生中进行排名。

`database` 类由 `student` 和 `Xueqi` 以公有方式派生而来，实则包含了两者的共同信息，即指定学期下的学生的成绩信息，由二者共同成员组成，在其对象数组被引用访问时可以遍历该数组中所有属性，按照指定的学期和学生信息找到对应的成员，进而确定在指定学期下指定学生的成绩信息，除了构造析构函数，该类还会再 `shuru`、`getGPA`、`search`、`search2`、`revise` 这几个大函数被引用。该类主要是被用来访问和录入、修改指定学期指定学生的成绩。

2. 功能的实现

总体功能结构如下图所示：

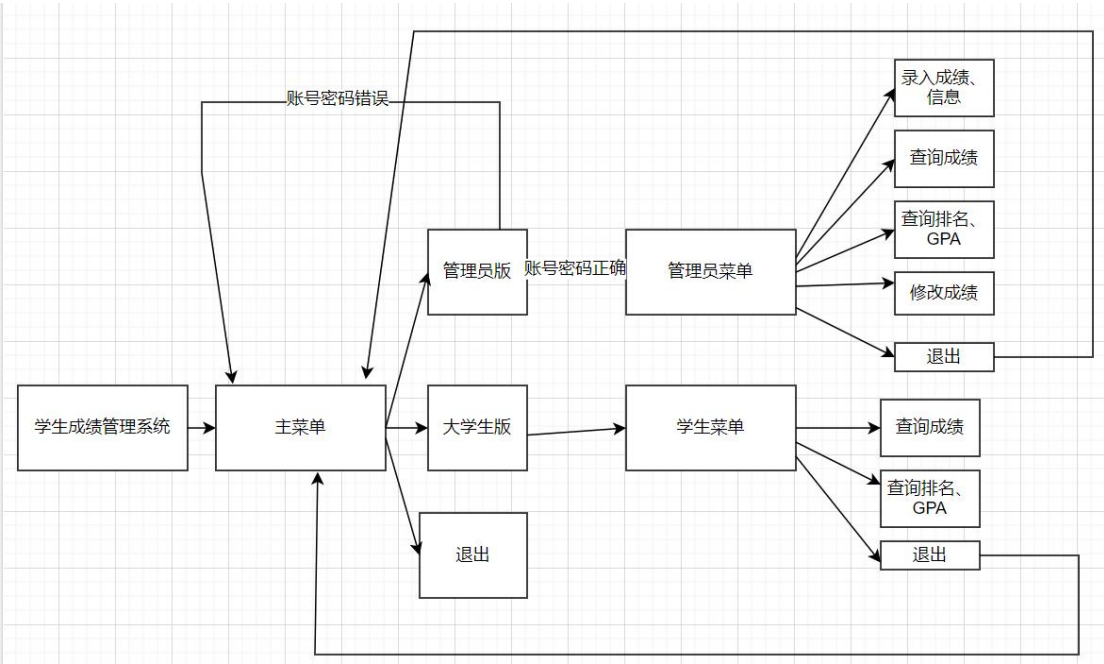


图 4 功能结构

输入模块功能具体实现方法如下图所示：

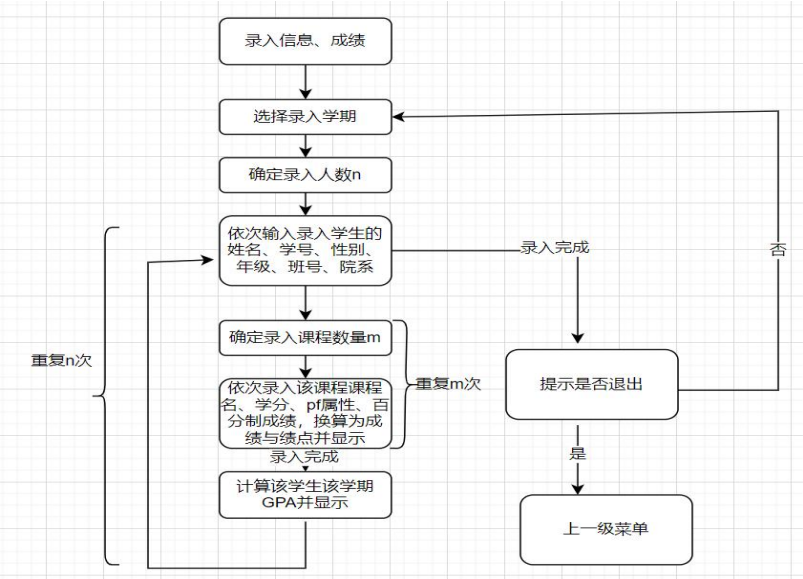


图 5 输入功能模块

在信息录入前，系统会自动打开存储信息的二进制数据文件，录入结束后会将这些对象数组成员以及所有属性信息以二进制形式写入到文件中储存，以实现信息的保存管理。

查询信息模块具体实现如下图所示：

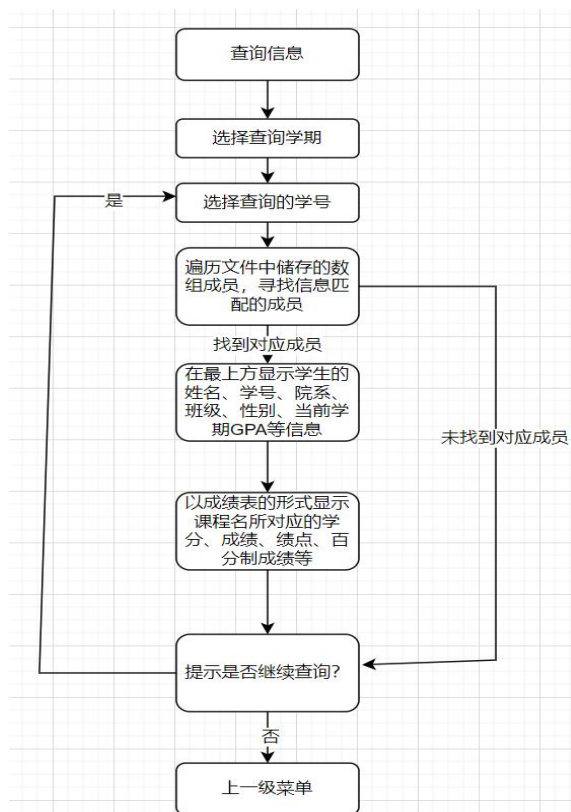


图 6 查询功能模块

在进入查询功能时，系统会以只读方式打开存储信息的数据文件，为数组分配新的内存空间后进行赋值，输入查找的相关信息后遍历数组成员，找到与信息相匹配的对象。在完成查询后关闭文件，释放内存空间，以确保数据文件信息不会丢失或被改动。

计算与查询排名、GPA 具体实现如下图所示：

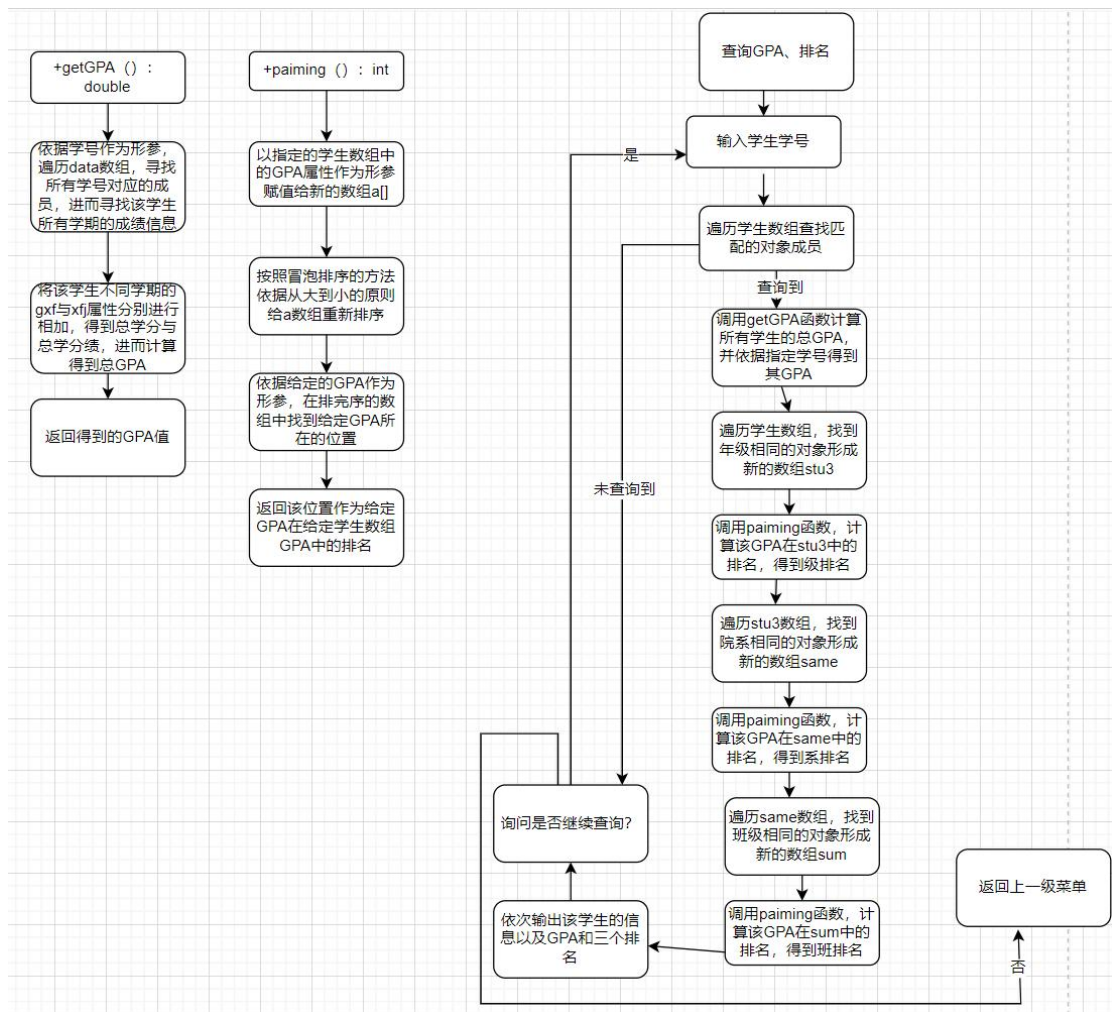


图 7 查询排名与 GPA 功能模块

在进入该功能时，系统会以只读方式打开存储信息的数据文件，本功能需要同时引用两种类型的对象数组，因此需要为两个数组分别分配新的内存空间后进行赋值。在完成查询后关闭文件，释放内存空间，以确保数据文件信息不会丢失或被改动。

修改成绩功能实现如下图所示：

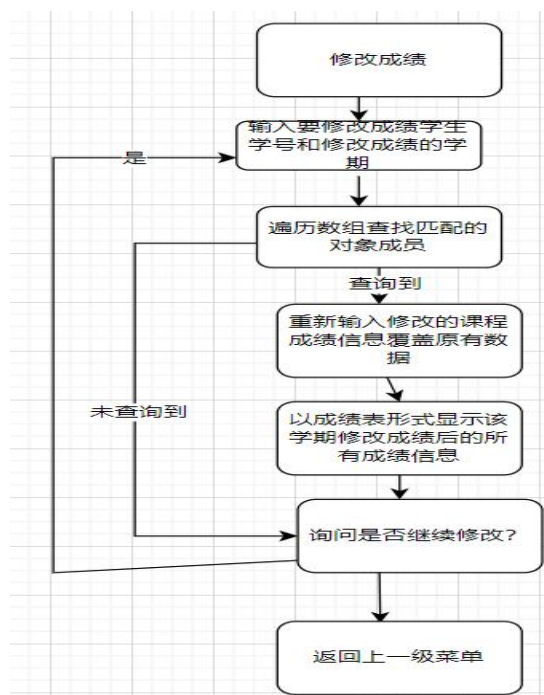


图 8 修改成绩功能模块

在进入该功能时，系统会以写入方式打开存储信息的数据文件，为数组分配新的内存空间后进行赋值，并根据输入的信息查询到指定的对象，在对信息做出修改后会同时通过文件指针操作找到修改的成员的位置进行重新的信息写入，覆盖掉文件中的原本信息，以实现信息的修改，并在以后查询等其他功能中更新现有的信息。在完成修改后关闭文件，释放内存空间，以确保数据文件信息不会丢失或被改动。

对于文件存储、查看与修改、录入信息的功能已在上述模块中进行了简要介绍，实现手段涉及到一些基本的文件流操作，在此不进行过多赘述。而对于登录功能的设置，我实现构造了一个结构体用来存储管理员的相关信息，包括用户名和密码等，并设置了一个新的二进制文件用来存储这些信息，在主函数中可以以重载输入运算符的方式实现简洁地输入管理员的信息并写入文件，同时可以在后台文件中删除和修改信息，以实现管理员账户的管控，并保证其安全性与封装性。在登陆页面中如果选择管理员版则需要输入用户名和密码，若与文件中的某一成员相匹配，则进入管理员页面，同时显示出该管理员的信息，表现为“欢迎回来（姓名） No.（工号）”，如果没有匹配的对象则显示用户名或密码错误，返回到主页面进行重新登陆；在大学生页面不需要任何操作，但只能查询成绩，不能进行录入和修改。而且考虑到实际情况的需要可能存在重名的情况，而参考本校，每名学生个体的学号都是独一无二的，因而在查询修改等操作时都是直接通过学号查询，这样可以精准定位到学生个体。

3. 图形界面的设计

对于图形界面的设计并没有太过华丽，只是采用了一些 I/O 流操作函数以实现输出格式的控制，这些主要体现在对菜单的制作以及成绩查询时的成绩表对齐问题上，其中用到了左对齐、控制输出精度、控制域宽等函数来实现功能，具体实现情况可参见第五部分“最终结果功能及使用说明”的相关图片展示。菜单中设置了多个选项可实现不同的功能，为此在主函数中运用了 switch 结构来进行选择，进入对应的页面实现不同功能。

四、 调试过程

.vs	2023/6/19 1:03	文件夹	
x64	2023/7/3 12:33	文件夹	
administrator.bin	2023/8/25 22:39	BIN 文件	0 KB
grade management system.bin	2023/8/26 14:02	BIN 文件	95 KB
学生成绩管理系统.sln	2023/6/19 1:03	Visual Studio Soluti...	2 KB
学生成绩管理系统.vcxproj	2023/7/14 22:16	VCXPROJ 文件	7 KB
学生成绩管理系统.vcxproj.filters	2023/7/14 22:16	VC++ Project Filter...	2 KB
学生成绩管理系统.vcxproj.user	2023/6/19 1:03	Per-User Project O...	1 KB
源.cpp	2023/8/26 12:03	C++ 源文件	31 KB
源1.cpp	2023/7/14 21:47	C++ 源文件	23 KB

图 9 存储成绩信息与管理员信息的两个二进制文件

如图，我将成绩信息以二进制形式储存在同一目录下名为 `grade management system.bin` 的二进制文件中，管理员信息储存在名为 `administrator.bin` 的文件中，测试所用的数据可参见第五部分图 14。

在调试过程中出现了不少问题，我接下来列举几个比较典型的问题：

（1）首先是文件的读取与写入方面，开始我并未采用二进制进行操作，只设定为普通的 `txt` 文件以便于查看信息，在输入的过程中可能是由于汉字不适配编译器的问题总会在文件中输出乱码，也可能与字符串操作不当有关，而且在构建文件对象进行写入操作时中直接引用的是对象数组，在后续打开的过程中并未记录录入的信息，于是我便在文件对象采取了读取数组地址进行指针操作，成功将读取到的信息存入到了文件当中；并将文件改为了二进制输入输出，实现了信息的及时读取与修改。

（2）在输入过程中，有时会碰到一些不小心输错格式的情况，如果未及时发现继续操作可能产生一些奇怪的 `bug`，为此我认真分析了每个输入的情况，新加入了输错格式的提示语，提醒退出或者重新返回上一输入界面，一定程度增加了容错率。

（3）在后续调用类中信息时，我有时会因为过程太匆忙而忽略类成员属性，经常会无法直接读取类中的私有（保护）类成员，为此只能在前面类中公有部分通过一些函数访问这

些成员。

(4) 在主函数中因为未及时释放内存造成了一些栈溢出，因而使主程序无法运行，开始我不知道什么意思，为此我思索了很久，意识到问题所在后采用 `new` 和 `delete` 函数动态分配和释放内存空间，将信息及时导入文件，有效解决了栈溢出的问题。

(5) 编写查看学生排名的过程中，开始想直接调用学生数组，根据其中 GPA 属性进行排序处理，可在运行过程中发现同时具体访问到这些属性后再返回处理学生数组成员时不仅使结构体系复杂化，同时也会打乱原有的默认顺序，可能在文件操作查找修改对象信息时产生重叠甚至删除，于是又另外编写了一个函数，仅仅将学生数组信息作为形参不改变其内容，将它们的 GPA 值赋值给一个新的数组 `a[]`，再对 `a` 数组进行排序，根据指定 GPA 找到对应位置并返回，这样可以有效实现排序得到排名同时不改变主函数中的顺序。

(6) 在计算学期总 GPA 时，由于 `pf` 课程属性的绩点“N/A”为字符串类型，无法直接参与计算，所以调试过程中单独遍历课程数组，将所有 `pf` 属性为 0 的课程对象单独组成一个数组，调用这些课程的学分与绩点计算 GPA，但很快发现这样做也有问题，可能的原因就是对象与对象之间赋值处理时产生了某些问题，具体是什么也没想明白，于是我取消了设置新的对象数组，而是在循环过程中碰到 `pf` 属性为 0 的直接累加学分与学分绩，有效避免了上述问题；考虑到一学期课程可能均为记 `pf` 的课，于是加入了异常处理选项：总学分如果为 0，即课程均为 `pf`，则直接认定绩点为“N/A”，计算总 GPA 时也是同理，避免了算绩点课程总学分作为分母为 0 的情况。

总而言之，调试测试的过程是一个艰难的过程，我也遇到了很多问题，后来也是通过仔细琢磨，复习课件后逐步探索、尝试后才摸索出来的道路。在 `debug` 的过程中我发现其实有很多问题都是由于自己粗心大意造成的，就比如直接像给整数赋值一样用“=”而不是用 `strcpy` 专门处理字符串的函数等等，真正技术上的问题也不算太多，但也在经历漫长的探索中才逐步有了头绪。有许多技术上的问题还是由于课上没完全理解透，不懂如何正确运用造成的，在此我深刻做出自我检讨，但在此过程中也通过回顾以往知识，求助同学，自己额外写一些包含相似情景的简单的程序尝试可能的解决办法，以此类推到复杂的大工程项目中，探寻解决的办法。

在调试过程中也使得系统越来越接近自己想要的结果，最终都能基本实现想要的功能。

五、 最终结果功能及使用说明

如图为进入页面时的主页面以及登录管理员页面的效果展示：

```
*****欢迎进入学生成绩管理系统！*****
*****
          学生成绩管理系统
*****
+++++
          请选择版本
          1.管理员版
          2.大学生版
          3.退出系统
          请选择序号（1-3）
*****
1
请输入用户名
wxqtsinghua.edu.cn
请输入密码
234755687sddvcEFCTd!dfgf
```

图 10 主页面及管理员登录页面

在进入管理员页面后会显示管理员的信息和以下的五个选项，首先尝试功能 1，在端口输入 1，进入第一个选项录入成绩

```
*****
          学生成绩管理系统（管理员版）
欢迎回来，wxq No.1
*****
+++++
          主菜单
          1.成绩录入
          2.成绩查询
          3.排名查询
          4.成绩修改
          5.退出系统
          请选择序号（1-5）
*****
1
请输入考试年份：
```

图 11 管理员页面

在进入第一个选项后会依次提示输入年份和学期（学期以数字代替并进行转换），依次通过键盘按提示输入学生姓名、学号、性别（以数字代替）、年级、班级、院系，在之后输入录入成绩的数量，并按照课程名、课程学分、pf 属性、百分制成绩的顺序输入信息，经过换算得到成绩与绩点并显示出来，如图 12 所示。

输入完该学生的所有成绩后会自动计算出其该学期的 GPA 并显示出来，如图 13 所示。

接下来会提示是否继续输入，键入 y 会继续回到输入学期的状态重新输入，n 则会退出当前界面返回管理员页面进行后续操作。

```
请输入考试年份: 2023
请输入考试学期 (春季输入1, 夏季输入2, 秋季输入3): 1
请输入学生姓名: 张三
请输入学号: 202201
请输入性别 (男性输入1, 女性输入0): 1
请输入年级: 1
请输入班级: 1
请输入院系: 电子系
请输入要输入的课程成绩数量:
5
  请输入课程名:
  高等微积分
  请输入课程学分:
  5
  请输入该课程属性 (是否记P/F)? (是输入1, 否输入0):
  0
  请输入该课程百分制成绩
  96
  该课程的成绩为A
  该课程的绩点为: 4.0
  该课程的百分制成绩为: 96
    请输入课程名:
    线性代数
    请输入课程学分:
    4
    请输入该课程属性 (是否记P/F)? (是输入1, 否输入0):
    0
    请输入该课程百分制成绩
    78
    该课程的成绩为B-
    该课程的绩点为: 3.0
    该课程的百分制成绩为: 78
      请输入课程名:
      大学物理
      请输入课程学分:
      4
      请输入该课程属性 (是否记P/F)? (是输入1, 否输入0):
      0
      请输入该课程百分制成绩
      88
      该课程的成绩为B+
      该课程的绩点为: 3.6
      该课程的百分制成绩为: 88
        请输入课程名:
        形势与政策
        请输入课程学分:
        1
        请输入该课程属性 (是否记P/F)? (是输入1, 否输入0):
        1
```

图 12 成绩录入页面

```
请输入该课程属性（是否记P/F）？（是输入1，否输入0）：
0
请输入该课程百分制成绩
78
该课程的成绩为B-
该课程的绩点为：3.0
该课程的百分制成绩为：78
  请输入课程名：
大学物理
请输入课程学分：
4
请输入该课程属性（是否记P/F）？（是输入1，否输入0）：
0
请输入该课程百分制成绩
88
该课程的成绩为B+
该课程的绩点为：3.6
该课程的百分制成绩为：88
  请输入课程名：
形势与政策
请输入课程学分：
1
请输入该课程属性（是否记P/F）？（是输入1，否输入0）：
1
请输入该课程百分制成绩
90
该课程的成绩为P
  该课程的绩点为：N/A
该课程的百分制成绩为：90
  请输入课程名：
电电
请输入课程学分：
2
请输入该课程属性（是否记P/F）？（是输入1，否输入0）：
0
请输入该课程百分制成绩
81
该课程的成绩为B
该课程的绩点为：3.3
该课程的百分制成绩为：81
该学生在2023学年春季学期的GPA为：3.533
是否还要继续输入：(y/n)
|
```

图 13 成绩录入完成页面

重新回到管理员页面后可以键入 2 选择第二个选项来查询学生的成绩。首先输入要查询的学期，接着输入查询学生的学号，如果查有此人则会按照图 14 所示的格式来输出对应的成绩信息，输出完后询问是否继续查询？键入 y 会重新进入学期选择阶段，键入 n 会退出当前界面返回管理员页面。查无此人会询问是否继续查询，按照提示输入对应信息操作即可。

```
*****
学生成绩管理系统（管理员版）
欢迎回来， wxq No.1
*****
+++++
主菜单
1.成绩录入
2.成绩查询
3.排名查询
4.成绩修改
5.退出系统
请选择序号（1-5）
*****
2
输入你要查询的学期（依次输入学年和学期（春季输入1，夏季输入2，秋季输入3））：
2023
1
输入你要查询的学生学号：
202201
姓名：张三 学号：202201 性别：男 学期的GPA为：3.53 年级：1 班号：1 院系：电子系
| 课程名 | 学分 | 成绩 | 百分制成绩 | 绩点 |
| 高等微积分 | 5 | A | 096 | 4.0 |
| 线性代数 | 4 | B- | 078 | 3.0 |
| 大学物理 | 4 | B+ | 088 | 3.6 |
| 形势与政策 | 1 | P | 090 | N/A |
| 电电 | 2 | B | 081 | 3.3 |
是否继续查询(y/n)继续请按 y 退出按 n
```

```
输入你要查询的学期（依次输入学年和学期（春季输入1，夏季输入2，秋季输入3））：
2023
1
输入你要查询的学生学号：
202202
姓名：李四 学号：202202 性别：女 学期的GPA为：2.77 年级：1 班号：1 院系：电子系
| 课程名 | 学分 | 成绩 | 百分制成绩 | 绩点 |
| 高等微积分 | 5 | C | 076 | 2.7 |
| 线性代数 | 4 | A+ | 100 | 4.0 |
| 大学物理 | 4 | B- | 077 | 3.0 |
| 形势与政策 | 1 | P | 098 | N/A |
| 电电 | 2 | F | 058 | 0.0 |
是否继续查询(y/n)继续请按 y 退出按 n
```

```
输入你要查询的学期（依次输入学年和学期（春季输入1，夏季输入2，秋季输入3））：
2023
1
输入你要查询的学生学号：
202203
姓名：王五 学号：202203 性别：男 学期的GPA为：4 年级：1 班号：2 院系：电子系
| 课程名 | 学分 | 成绩 | 百分制成绩 | 绩点 |
| 高等微积分 | 5 | A- | 090 | 4.0 |
| 线性代数 | 4 | A- | 090 | 4.0 |
| 大学物理 | 4 | A- | 091 | 4.0 |
| 形势与政策 | 1 | P | 087 | N/A |
| 电电 | 2 | A- | 090 | 4.0 |
是否继续查询(y/n)继续请按 y 退出按 n
```

```
输入你要查询的学期（依次输入学年和学期（春季输入1，夏季输入2，秋季输入3））：
2023
1
输入你要查询的学生学号：
202204
姓名：胡桃 学号：202204 性别：女 学期的GPA为：2.98 年级：1 班号：1 院系：计算机系
| 课程名 | 学分 | 成绩 | 百分制成绩 | 绩点 |
| 微积分A | 5 | C- | 068 | 2.0 |
| 高等线性代数选讲 | 3 | A | 096 | 4.0 |
| 数据结构 | 3 | B+ | 087 | 3.6 |
是否继续查询(y/n)继续请按 y 退出按 n
```



```

输入你要查询的学期（依次输入学年和学期（春季输入1，夏季输入2，秋季输入3））：
2023
1
输入你要查询的学生学号：
202101
姓名：赵六 学号：202101 性别：男 学期的GPA为：3.54 年级：2 班号：1 院系：电子系
|      课程名      | 学分 | 成绩 | 百分制成绩 | 绩点 |
|      信号与系统    | 4    | B+   | 087        | 3.6   |
|      数据与算法    | 3    | A    | 097        | 4.0   |
|      电磁场与波    | 3    | B-   | 077        | 3.0   |
是否继续查询(y/n)继续请按 y 退出按 n

```

图 14 成绩查询页面

查询完成绩后退出管理员页面，选择大学生页面，因为事先已将录入成绩写入文件存储所以在此界面可直接调取查看成绩，与管理员端格式基本相同，只是少了管理员的信息提示，排名查询也与管理员端口相同。

```

学生成绩管理系统（管理员版）
欢迎回来，wxq No.1
*****
+++++
主菜单
1.成绩录入
2.成绩查询
3.排名查询
4.成绩修改
5.退出系统
请选择序号（1-5）
*****
5
系统已关闭
*****欢迎进入学生成绩管理系统！*****
*****
学生成绩管理系统
*****
+++++
请选择版本
1.管理员版
2.大学生版
3.退出系统
请选择序号（1-3）
*****
2
*****
学生成绩管理系统（学生版）
*****
+++++
主菜单
1.成绩查询
2.排名查询
3.退出系统
请选择序号（1-3）
*****
1
输入你要查询的学期（依次输入学年和学期（春季输入1，夏季输入2，秋季输入3））：
2023
1
输入你要查询的学生学号：
202201
姓名：张三 学号：202201 性别：男 学期的GPA为：3.53 年级：1 班号：1 院系：电子系
|      课程名      | 学分 | 成绩 | 百分制成绩 | 绩点 |
|      高等微积分    | 5    | A    | 096        | 4.0   |
|      线性代数      | 4    | B-   | 078        | 3.0   |
|      大学物理      | 4    | B+   | 088        | 3.6   |
|      形势与政策    | 1    | P    | 090        | N/A   |
|      电电          | 2    | B    | 081        | 3.3   |
是否继续查询(y/n)继续请按 y 退出按 n

```

图 15 大学生端成绩查询页面

重新进入管理员页面后可以键入 3 进入第三个选项，查看学生的排名和 GPA，先输入查询的学号，如果查无此人会进行信息提示，继续查询可继续输入学号，不查询可退出页面。查询到后会显示该学生的基本信息、GPA、三大排名，之后可选择继续查询或退出，退出则会返回上一级菜单页面。

```
*****
      学生成绩管理系统（管理员版）
欢迎回来， wxq  No.1
*****
+++++
      主菜单
      1.成绩录入
      2.成绩查询
      3.排名查询
      4.成绩修改
      5.退出系统
      请选择序号（1-5）
*****
3
输入你要查询的学生学号：
202201
姓名：张三 学号：202201 性别：男 年级：1 班号：1 院系：电子系
级排名为：2
系排名为：2
班排名为：1
总体GPA为：3.533
是否继续查询(y/n)继续请按 y 退出按 n
y
输入你要查询的学生学号：
202202
姓名：李四 学号：202202 性别：女 年级：1 班号：1 院系：电子系
级排名为：4
系排名为：3
班排名为：2
总体GPA为：2.767
是否继续查询(y/n)继续请按 y 退出按 n
y
输入你要查询的学生学号：
202203
姓名：王五 学号：202203 性别：男 年级：1 班号：2 院系：电子系
级排名为：1
系排名为：1
班排名为：1
总体GPA为：4.000
是否继续查询(y/n)继续请按 y 退出按 n
y
输入你要查询的学生学号：
202204
姓名：胡桃 学号：202204 性别：女 年级：1 班号：1 院系：计算机系
级排名为：3
系排名为：1
班排名为：1
总体GPA为：2.982
是否继续查询(y/n)继续请按 y 退出按 n
y
输入你要查询的学生学号：
202101
姓名：赵六 学号：202101 性别：男 年级：2 班号：1 院系：电子系
级排名为：1
系排名为：1
班排名为：1
总体GPA为：3.540
是否继续查询(y/n)继续请按 y 退出按 n
```

图 16 排名、GPA 查询页面

返回管理员页面后键入 4 可进入第四个选项进行学生成绩的修改，按提示输入学期和学号，接着按照输入的课程名找到该对象，进而对该课程的学分、pf 属性、百分制成绩进行修改，覆盖原有数据，最后显示该学期修改完后的成绩，之后可按提示选择继续修改或退出，返回上一级菜单页面。

```
*****
      学生成绩管理系统（管理员版）
欢迎回来, wxq No.1
*****
+++++
      主菜单
      1.成绩录入
      2.成绩查询
      3.排名查询
      4.成绩修改
      5.退出系统
      请选择序号（1-5）
*****
4
-----输入开始！-----
请输入考试年份：2023
请输入考试学期（春季输入1，夏季输入2，秋季输入3）：1
请输入你要修改成绩的学生学号
202201
请输入要修改的课程名：
线性代数
请重新输入课程学分：
4
请重新输入该课程属性（是否记P/F）？（是输入1，否输入0）：
0
请重新输入该课程百分制成绩
90
该课程的成绩为A-
该课程的百分制成绩为：90
该课程的绩点为：4.0
该课程的百分制成绩为：90
|      课程名      | 学分 | 成绩 | 百分制成绩 | 绩点 |
|      高等微积分   | 5    | A    | 096        | 4.0  |
|      线性代数     | 4    | A-   | 090        | 4.0  |
|      大学物理     | 4    | B+   | 088        | 3.6  |
|      形势与政策   | 1    | P    | 090        | N/A  |
|      电电        | 2    | B    | 081        | 3.3  |
还要继续吗?退出请按 0,继续请按1:
|
```

图 17 成绩修改页面

在成绩修改完成后文件的相应信息数据也会被重新写入更新，再次查询成绩后可以发现成绩（如图 18 所示）和排名（如图 19 所示）均已更新到修改后的状态。

```

*****
      学生成绩管理系统（管理员版）
欢迎回来, wxq No.1
*****
+++++
      主菜单
      1.成绩录入
      2.成绩查询
      3.排名查询
      4.成绩修改
      5.退出系统
      请选择序号（1-5）
*****
2
输入你要查询的学期（依次输入学年和学期（春季输入1，夏季输入2，秋季输入3））：
2023
1
输入你要查询的学生学号：
202201
姓名：张三 学号：202201 性别：男 学期的GPA为：3.8 年级：1 班号：1 院系：电子系
|      课程名      | 学分 | 成绩 | 百分制成绩 | 绩点 |
|      高等微积分  |    5 |   A  |    096    |  4.0 |
|      线性代数    |    4 |  A-  |    090    |  4.0 |
|      大学物理    |    4 |  B+  |    088    |  3.6 |
|      形势与政策  |    2 |   P  |    090    |  N/A |
|      电电        |    2 |   B  |    081    |  3.3 |
是否继续查询(y/n)继续请按 y 退出按 n

```

图 18 修改更新后成绩查询页面

```

输入你要查询的学生学号：
202201
姓名：张三 学号：202201 性别：男 年级：1 班号：1 院系：电子系
级排名为：2
系排名为：2
班排名为：1
总体GPA为：3.800
是否继续查询(y/n)继续请按 y 退出按 n
y
输入你要查询的学生学号：
202202
姓名：李四 学号：202202 性别：女 年级：1 班号：1 院系：电子系
级排名为：4
系排名为：3
班排名为：2
总体GPA为：2.767
是否继续查询(y/n)继续请按 y 退出按 n
y
输入你要查询的学生学号：
202203
姓名：王五 学号：202203 性别：男 年级：1 班号：2 院系：电子系
级排名为：1
系排名为：1
班排名为：1
总体GPA为：4.000
是否继续查询(y/n)继续请按 y 退出按 n
y
输入你要查询的学生学号：
202204
姓名：胡桃 学号：202204 性别：女 年级：1 班号：1 院系：计算机系
级排名为：3
系排名为：1
班排名为：1
总体GPA为：2.982
是否继续查询(y/n)继续请按 y 退出按 n
y
输入你要查询的学生学号：
202101
姓名：赵六 学号：202101 性别：男 年级：2 班号：1 院系：电子系
级排名为：1
系排名为：1
班排名为：1
总体GPA为：3.540
是否继续查询(y/n)继续请按 y 退出按 n

```

图 19 修改更新后排名查询页面

返回到管理员页面,如果想继续录入添加已有学生的成绩可以直接进入选项 1 录入新的成绩,如图 20 所示。

```
*****
学生成绩管理系统（管理员版）
欢迎回来, wxq No.1
*****
+++++
      主菜单
      1.成绩录入
      2.成绩查询
      3.排名查询
      4.成绩修改
      5.退出系统
      请选择序号（1-5）
*****
2
输入你要查询的学期（依次输入学年和学期（春季输入1，夏季输入2，秋季输入3））：
2022
3
输入你要查询的学生学号：
202201
姓名：张三 学号：202201 性别：男 学期的GPA为：3.6 年级：1 班号：1 院系：电子系
|          课程名          | 学分 | 成绩 | 百分制成绩 | 绩点 |
|          军事理论          | 2    | B+   | 088        | 3.6   |
|          军事技能          | 2    | B+   | 088        | 3.6   |
是否继续查询(y/n)继续请按 y 退出按 n
n
```

图 20 录入已有学生新的成绩

在录入完新的成绩后,文件中的信息也会被更新, data 数组会被更新,进而使 GPA 和排名更新为最新状态,即统计所有有记录成绩的总 GPA 和对应的排名,如图 21 所示可以发现统计了另一学期成绩后, GPA 发生了相应改变,根据第三部分的设计机制可知排名也会随着该 GPA 被更新。

```
*****
学生成绩管理系统（管理员版）
欢迎回来, wxq No.1
*****
+++++
      主菜单
      1.成绩录入
      2.成绩查询
      3.排名查询
      4.成绩修改
      5.退出系统
      请选择序号（1-5）
*****
3
输入你要查询的学生学号：
202201
姓名：张三 学号：202201 性别：男 年级：1 班号：1 院系：电子系
级排名为：2
系排名为：2
班排名为：1
总体GPA为：3.758
是否继续查询(y/n)继续请按 y 退出按 n
```

图 21 录入已有学生新的成绩后 GPA 与排名更新

在退出管理系统后由于文件存档可以使之之后的调用中直接查询到该学生最新版的成绩，进而实现成绩的管理功能，如图 23 所示。

```
*****
      学生成绩管理系统（管理员版）
欢迎回来，wxq No.1
*****
+++++
      主菜单
      1.成绩录入
      2.成绩查询
      3.排名查询
      4.成绩修改
      5.退出系统
      请选择序号（1-5）
*****
5
系统已关闭
*****欢迎进入学生成绩管理系统！*****
*****
      学生成绩管理系统
*****
+++++
      请选择版本
      1.管理员版
      2.大学生版
      3.退出系统
      请选择序号（1-3）
*****
3

F:\程设\学生成绩管理系统\学生成绩管理系统\x64\Debug\学生成绩管理系统.exe（进程 16068）已退出，代码为 0。
按任意键关闭此窗口。 . .|
```

图 22 退出管理系统页面

```
*****欢迎进入学生成绩管理系统！*****
*****
      学生成绩管理系统
*****
+++++
      请选择版本
      1.管理员版
      2.大学生版
      3.退出系统
      请选择序号（1-3）
*****
2
*****
      学生成绩管理系统（学生版）
*****
+++++
      主菜单
      1.成绩查询
      2.排名查询
      3.退出系统
      请选择序号（1-3）
*****
1
输入你要查询的学期（依次输入学年和学期（春季输入1，夏季输入2，秋季输入3））：
2023
1
输入你要查询的学生学号：
202201
姓名：张三 学号：202201 性别：男 学期的GPA为：3.8 年级：1 班号：1 院系：电子系
|      课程名      | 学分 | 成绩 | 百分制成绩 | 绩点 |
|      高等微积分      | 5 | A | 096 | 4.0 |
|      线性代数      | 4 | A- | 090 | 4.0 |
|      大学物理      | 4 | B+ | 088 | 3.6 |
|      形势与政策      | 1 | P | 090 | N/A |
|      电电      | 2 | B | 081 | 3.3 |
是否继续查询(y/n)继续请按 y 退出按 n
```

图 23 退出管理系统直接查询最新的成绩

本部分主要以管理员页面的操作为例，大学生版基本与管理员版操作流程一致，除了少了一个管理员信息提示页面与成绩录入修改功能，使用者可根据需求按照提示操作。

六、 总结

总而言之，这次大作业让我有了很多收获，我从中不仅重新复习了以往的知识，也能通过这次大作业更好的锻炼自己写代码、运用所学知识解决实际情境问题的能力，这对于以后工科课程的学习也是有举足轻重的意义的。

在阅读完任务书后我首先对系统需求按照自己的理解进行分析，构想出几种应当实现的功能，又从自己的学习环境入手，引入了一些特殊的机制，如 pf 课程的处理，GPA 的计算，如何调用同一学生不同学期的成绩，构建类储存多个学期的成绩等，这些都是应当考虑的全新问题。最终在纸上对类的关系图及其函数进行多次分析修改后确定了基本的结构方向，接着就是规划时间来实现功能。所以我体会到在完成大型工作时首先要纵观全局，对工作有一个基本的了解并进行大致规划，不能盲目地走一步看一步完全没有方向，想到啥写到哪里这样子，应当对事物有一个全局观。

然而这个系统距离真正成为完善、可以投入应用还有很长的路需要走，比如录入成绩时需要自己手动录入，而没有根据给定的文件直接进行信息写入的功能，这一定程度上增大了工作量，这也是我在大局已成之后才想到的；同时需要进一步优化算法、增加新的功能，这也启示我在编写之前要进一步全面考虑需求与可行性、必要性。

此外本次大作业也更加开拓了我的思维，让我熟悉了对象操作的同时进一步改进了原有的一些处理方法和结构，虽然在本次作业中意义不算太大，但在之后更大数据量、计算量项目的编写中却有着举足轻重的作用，以此来指导我应当改进优化结构算法，让处理起来更加高效，精益求精；同时此次作业让我对面向对象的编程有了更进一步地理解，通过对类的封装可以更好增加大项目的可操作性，只需要在主函数中调用，处理类中的属性和方法即可，而不是像上学期面向过程编程一样在主函数中重新写大量的过程，这样对类的处理极大地增加了主函数的简便性与可操作性，使层次更加清晰一目了然，也使 debug 变得更加容易，只需要根据问题找到对应模块进行分析处理而不用把全部函数都检查一遍。

在这次实践过程中，我逐步摸索到了面向对象编程的精髓所在，也体会到了处理大项目常用的切入方法。同时因为自己的心急，只追求高技术而忽视了最基本的处理，由此产生了一些小儿科的问题，为此我深刻体会到做学问不应该马虎，对于所学一定要有深入的理解，应当脚踏实地的学习，学一步理解一步，以此为基础逐步攀登，而非急于求成，只靠突击临阵磨枪，这样掌握的终究不够扎实，不能自由灵活运用。这也是我这次课设得到的启示。

总而言之，通过这次大作业，让我在掌握技术、巩固知识同时也收获了很多思想，正所谓实践出真知，只有在亲身经历操作后才能得到最真挚的感悟，这只靠死学知识是学不来的。

附录一：源程序清单

main.cpp

```
#include<iostream>
#include<string>
#include<iomanip>
#include<fstream>
#pragma warning(disable: 4996)
using namespace std;
int q = -1; int b1 = 0; int shu = 0; int g = 1; int year, xf, pf;
char season[20];
char kcname[30];
char number[20];
double cj100;
class People//定义抽象类
{
public:
    char name[11];
    char number[20];
    char sex[5];//定义三个基本属性
    People() {}
    People(const char* n, const char* num, const char* sexx )
    {
        strcpy(name, n);
        strcpy(number, num);
        strcpy(sex, sexx);
    }//定义构造函数
    ~People() {};//定义析构函数
    virtual void xueli()const { cout << "未知"; }//查看学历
    virtual void work()const = 0;//查看职业（无函数体）
};
class Xueqi//定义学期类
{
public:
    int year;
    char season[10];//定义学年和季节两个基本属性
    Xueqi() {}
    Xueqi(int yy, const char* sea)
    {
        year = yy;
        strcpy(season, sea);
    }//构造函数
    ~Xueqi() {};//析构函数
```



```

    void getyear() {
        cout << year;
    }
    void getseason()
    {
        cout << season;
    }
    void setyear(int yy) {
        year = yy;
    }
    void setseason(const char* sea)
    {
        strcpy(season, sea);
    }

};

class Chengji//定义成绩类
{
protected:
    char jidianpf[4] = "N/A";
    double jidian;
    char cj[3];
    double cj100;
    int xf;
    int pf;//定义成绩的基本属性（封装保护部分）
public:
    char kcname[30];
    int genxf;//定义成绩的基本属性（公共公开部分）
    Chengji() {}
    Chengji(const char* kc, int x, int p, double n1)
    {
        strcpy(kcname, kc);
        xf = x;
        pf = p;
        cj100 = n1;
    }//构造函数
    ~Chengji() {};//析构函数
    void getkcname() {cout << kcname;}
    int getkcnamelen(){return strlen(kcname);}
    void getjidianpf(){cout << jidianpf;}
    void getjidian(){cout << jidian;}
    double getjidian1() {return jidian;}
    void getcj(){cout << cj;}
    int getcjlen(){return strlen(cj);}

```

```

void getcj100(){cout << cj100;}
int getcj1001(){return cj100;}
int getxf1(){return xf;}
void getxf(){cout << xf;}
int getpf1(){return pf;}
void getpf(){cout << pf;}//获取保护类型的属性
void setkcname(const char* kc) {strcpy(kcname, kc);}
void setxf(int x){xf = x;}
void setpf(int p){pf = p;}
void setcj100(double n1){cj100 = n1;}//输入保护类型的属性
void calculate() {
    if (pf == 1)
    {
        if (cj100 >= 60)
        {
            strcpy(cj, "P");
        }
        else
        {
            strcpy(cj, "F");
        }
    }
    else
    {
        if (cj100 < 60)
        {
            strcpy(cj, "F");
            jidian = 0.0;
        }
        else if (cj100 >= 60 && cj100 < 65)
        {
            strcpy(cj, "D");
            jidian = 1.6;
        }
        else if (cj100 >= 65 && cj100 < 70)
        {
            strcpy(cj, "C-");
            jidian = 2.0;
        }
        else if (cj100 >= 70 && cj100 < 73)
        {
            strcpy(cj, "C");
            jidian = 2.4;
        }
    }
}

```

```

else if (cj100 >= 73 && cj100 < 77)
{
    strcpy(cj, "C");
    jidian = 2.7;
}
else if (cj100 >= 77 && cj100 < 80)
{
    strcpy(cj, "B-");
    jidian = 3.0;
}
else if (cj100 >= 80 && cj100 < 85)
{
    strcpy(cj, "B");
    jidian = 3.3;
}
else if (cj100 >= 85 && cj100 < 90)
{
    strcpy(cj, "B+");
    jidian = 3.6;
}
else if (cj100 >= 90 && cj100 < 95)
{
    strcpy(cj, "A-");
    jidian = 4.0;
}
else if (cj100 >= 95 && cj100 < 100)
{
    strcpy(cj, "A");
    jidian = 4.0;
}
else
{
    strcpy(cj, "A+");
    jidian = 4.0;
}
}

```

}//按照制定的规则根据百分制成绩换算成对应的等级制成绩与绩点或计算 PF 成绩

```

};

class student : public Chengji,public People//由成绩与 People 两个类共同派生出学生类
{
public:
    char depart[20];
    int grade;

```

```

    int classno;
    double GPA;
    double GPA1;
    int rank;
    int shu;
    int genxf;
    double xfj;//定义学生类基本属性
    Chengji chj[200];//定义学生类包含的各课程成绩组成的对象数组
    student() {}
    student(const char* n, const char* num, const char* sexx, int g, int c, const char*
d, const char* kc, int x, int p, double n1, int nu) :Chengji(),People()
    {
        grade = g;
        classno = c;
        strcpy(depart, d);
    };//构造函数
    ~student() {};//析构函数
    virtual void work()const { cout << "大学生"; }//虚函数，输出学生职业
    virtual void xueli()const { cout << "高中及以上"; }//虚函数，将抽象类实例化，输出学
生学历

    int paiming(student stu[], int n, double x) {};//在全体学生中计算指定学号的学生的排
名函数

};

class database :public student, public Xueqi//由学生与具体某学期双重属性构成的数据类
{
public:
    student stu;
    Xueqi xq;//定义学期与学生两个组成部分（属性）
    database() {};
    database(int yy, const char* sea, const char* n, const char* num, const char* sexx,
int g, int c, const char* d, const char* kc, int x, int p, double n1, int nu) :student(),
Xueqi()
    {};//构造函数
    ~database() {};//析构函数
    int shuru(student stu[], Xueqi xq[], database data[]) {};//输入学生成绩
    double getGPA(const char* number, database data[]) {};//计算指定学号的学生的 GPA 函
数

    void search(database data[]) {};//查询学生指定学期成绩的函数
    void search2(student stu[], database data[]) {};//查询学生排名和 GPA 的函数
    void revise(database data[]) {};//修改学生指定学期成绩的函数
};

struct adminis
{

```

```

    char name[20];
    int gonghao;
    char userid[30];
    char password[50];
}; //定义管理员结构体

istream& operator>>(istream& input, adminis& ad)
{
    input >> ad.name >> ad.gonghao >> ad.userid >> ad.password;
    return input;
}; //重载输入管理员信息的流运算符

void input(adminis ad[])
{
    ifstream in;
    in.open("administrator.txt", ios::in);
    for (int i = 0; i < 4; i++)
    {
        in >> ad[i];
    }
    in.close();
}; //将管理员的信息导入指定文件中

int shuru(student stu[], Xueqi xq[], database data[]) //输入学生成绩
{
    char a;
    int c, d;
    cout << "请输入考试年份: ";
    cin >> year;
    cout << "请输入考试学期 (春季输入 1, 夏季输入 2, 秋季输入 3): ";
    cin >> c; //提示输入信息, 输入指定学期的属性来锁定该学期
    if (c == 1) { strcpy(season, "春季"); }
    else if (c == 2) { strcpy(season, "夏季"); }
    else if (c == 3) { strcpy(season, "秋季"); }
    else {
        do {
            cout << "输入不合规范, 请重新输入: ";
            cin >> c;
        } while (c != 1 && c != 2 && c != 3);
    } //判断对应的季节
    if ((year % 4 == 0))
    {
        if (strcmp(season, "秋季") == 0) b1 = 0;
        else if (strcmp(season, "春季") == 0) b1 = 1;
        else if (strcmp(season, "夏季") == 0) b1 = 2;
    }
    else if ((year % 4 == 1))

```

```

{
    if (strcmp(season, "秋季") == 0) b1 = 3;
    else if (strcmp(season, "春季") == 0) b1 = 4;
    else if (strcmp(season, "夏季") == 0) b1 = 5;
}
else if ((year % 4 == 2))
{
    if (strcmp(season, "秋季") == 0) b1 = 6;
    else if (strcmp(season, "春季") == 0) b1 = 7;
    else if (strcmp(season, "夏季") == 0) b1 = 8;
}
else if ((year % 4 == 3))
{
    if (strcmp(season, "秋季") == 0) b1 = 9;
    else if (strcmp(season, "春季") == 0) b1 = 10;
    else if (strcmp(season, "夏季") == 0) b1 = 11;
} //依据年份与季节来判断该学期在总的学期数组中的位置
xq[b1].setyear(year);
xq[b1].setseason(season);
do {
    q++;
    cout << "请输入学生姓名:";
    cin >> stu[q].name;
    cout << "请输入学号:";
    cin >> stu[q].number;
    cout << "请输入性别（男性输入 1，女性输入 0）:";
    cin >> d;
    if (d == 0)
    {
        strcpy(stu[q].sex, "女");
    }
    else if (d == 1) strcpy(stu[q].sex, "男");
    else {
        do {
            cout << "输入不合规范，请重新输入: ";
            cin >> d;
        } while (d != 0 && d != 1);
    } //判断对应的性别
    cout << "请输入年级:";
    cin >> stu[q].grade;
    cout << "请输入班级:";
    cin >> stu[q].classno;
    cout << "请输入院系:";
    cin >> stu[q].depart; //输入学生基本信息
}

```

```

cout << " 请输入要输入的课程成绩数量:" << endl;
cin >> shu;//输入课程数量
stu[q].shu = shu;
for (int i = 0; i < shu; i++)
{
    cout << " 请输入课程名:" << endl;
    cin >> kcname;
    stu[q].chj[i].setkcname(kcname);
    cout << "请输入课程学分: " << endl;
    cin >> xf;
    stu[q].chj[i].setxf(xf);
    cout << "请输入该课程属性（是否记 P/F）？（是输入 1，否输入 0）: " << endl;
    cin >> pf;
    stu[q].chj[i].setpf(pf);
    cout << "请输入该课程百分制成绩 " << endl;
    cin >> cj100;
    stu[q].chj[i].setcj100(cj100);
    stu[q].chj[i].calculate();
    cout << "该课程的成绩为"; stu[q].chj[i].getcj(); cout << endl;
    if (stu[q].chj[i].getpf1() == 1) { cout << " 该课程的绩点为: ";
stu[q].chj[i].getjidianpf(); cout << endl; }//如果该课程记 pf，直接换算成 pf 成绩
    else { cout << "该课程的绩点为: "; cout.setf(ios::fixed); cout <<
setprecision(1); stu[q].chj[i].getjidian(); cout << setiosflags(ios_base::right) <<
setprecision(0); cout << endl; }
        cout << "该课程的百分制成绩为: "; stu[q].chj[i].getcj100(); cout << endl;//
如果该课程不记 pf，则换算成等级制成绩与绩点
    }
    int j = 0, m = 0, i = 0;
    stu[q].genxf = 0;
    stu[q].xfj = 0;
    do
    {
        if (stu[q].chj[i].getpf1() == 0)
        {
            stu[q].genxf += stu[q].chj[i].getxf1();
            stu[q].xfj += (stu[q].chj[i].getxf1()) *
(stu[q].chj[i].getjidian1());
        }
        i++;
    } while (i < shu);//计算不记 pf 的课程的总学分与总学分绩
    m = j - 1;
    if (stu[q].genxf == 0) { stu[q].GPA = 0; }
    else { stu[q].GPA = stu[q].xfj / stu[q].genxf; }//计算该学生本学期的 GPA
    cout << "该学生在";

```

```

        xq[b1].getyear();
        cout << "学年";
        xq[b1].getseason();
        if (stu[q].genxf != 0) { cout.setf(ios::fixed); cout << "学期的 GPA 为: " <<
setprecision(3) << stu[q].GPA; cout.unsetf(ios::fixed); }
        else { cout << "学期的 GPA 为: " << "N/A"; }
        cout << endl; //输出对应信息
        cout << "是否还要继续输入:(y/n)" << endl;
        cin >> a;
        data[b1 * 20 + q].xq = xq[b1];
        data[b1 * 20 + q].stu = stu[q]; //将该学期该学生的信息导入到数据的对象数组中
    } while (a == 'y');
    return q + 1; //返回输入学生的数量
}

double getGPA(const char* number, database data[]) //计算指定学号的学生的 GPA 函数
{
    int i1 = 0; int j1 = 0;
    int m1 = j1 - 1;
    int gef = 0;
    double xj = 0;
    double p = 0;
    char r[10];
    for (i1 = 0; i1 < 240; i1++)
    {
        if ((strcmp(number, data[i1].stu.number) == 0)) //查找数据数组中所有有关该学生的
对象
        {
            gef += data[i1].stu.genxf; //计算该学生所有学期的总学分
            xj += data[i1].stu.xfj; //计算该学生所有学期的总学分
        }
    }
    if (gef == 0) p == 0;
    else { p = xj / gef; } //计算该学生所有学期总 GPA
    return p; //返回计算得到的 GPA
}

void search(database data[]) //查询学生指定学期成绩的函数
{
    char n = 'y';
    int z;
    while (n == 'y')
    {
        int j = -1, c;
        char n1[20];

```



```

        cout << "输入你要查询的学期(依次输入学年和学期(春季输入 1, 夏季输入 2, 秋季输入 3)):"
<< endl;
        cin >> year;
        cin >> c;
        if (c == 1) { strcpy(season, "春季"); }
        else if (c == 2) { strcpy(season, "夏季"); }
        else if (c == 3) { strcpy(season, "秋季"); } //输入查找的学期信息
        cout << "输入你要查询的学生学号:" << endl;
        cin >> number; //输入查找学生的学号
        for (int i = b1; i < b1*20+g; i++)
        {
            if ((strcmp(number, data[i].stu.number) == 0) && (strcmp(season,
data[i].xq.season) == 0) && (data[i].xq.year == year)) //查询数据数组中三种属性同时满足
的对象
            {
                j = i;
            }
        }
        if (j == -1)
        {
            cout << "没有你所要的学生!";
            cout << "是否退出?退出请按 n\n";
            cin >> n;
        } //没有找到对应的对象并输出提示
        else {
            cout << "姓名: " << data[j].stu.name << " ";
            cout << "学号: " << data[j].stu.number << " ";
            cout << "性别: " << data[j].stu.sex << " ";
            if (data[j].stu.genxf != 0) { cout << "学期的 GPA 为: " << data[j].stu.GPA
<< " "; }
            else { cout << "学期的 GPA 为: " << "N/A "; }
            cout << "年级: " << data[j].stu.grade << " ";
            cout << "班号: " << data[j].stu.classno << " ";
            cout << "院系: " << data[j].stu.depart << " " << endl; //输出该学期该学生基
本成绩信息
            cout << "|          课程名          |" << " 学分 |" << " 成绩 |" << " 百分
制成绩 |" << " 绩点 |" << endl; //设定具体成绩输出格式
            for (int a = 0; a < data[j].stu.shu; a++)
            {
                int len = data[j].stu.chj[a].getknamelen();
                int len1 = floor((30 - len) / 2);
                cout << "|";
                cout << setfill(' ') << setw(len1 + len);
                data[j].stu.chj[a].getkname(); //依照指定的总宽度按照格式输出课程名

```

```

        cout << setfill(' ') << setw(30 - len1 - len);
        cout << "| ";
        if (data[j].stu.chj[a].getxf1() < 10) { data[j].stu.chj[a].getxf();
cout << " | "; }
        else if (data[j].stu.chj[a].getxf1() >= 10)
{ data[j].stu.chj[a].getxf(); cout << " | "; } //依照指定的总宽度按照格式输出课程学分
        int len2 = data[j].stu.chj[a].getcjlent();
        int len3 = (7 - len2) / 2;
        cout << setfill(' ') << setw(len3 - 1);
        data[j].stu.chj[a].getcj(); //依照指定的总宽度按照格式输出课程成绩
        cout << setfill(' ') << setw(7 - len2 - len3) << " | ";
        cout.setf(ios::fixed);
        cout << setiosflags(ios_base::right) << setprecision(0);
        if (data[j].stu.chj[a].getcj1001() < 10) { cout <<
setiosflags(ios_base::right) << setprecision(0); cout << "00";
data[j].stu.chj[a].getcj100(); cout << " "; }
        else if (data[j].stu.chj[a].getcj1001() >= 10 &&
data[j].stu.chj[a].getcj1001() < 100) { cout << setiosflags(ios_base::right) <<
setprecision(0); cout << "0"; data[j].stu.chj[a].getcj100(); cout << " "; }
        else {
            cout << setiosflags(ios_base::right) << setprecision(0);
            data[j].stu.chj[a].getcj100(); cout << " ";
        } //依照指定的总宽度按照格式输出课程百分制成绩
        cout << " | ";
        if (data[j].stu.chj[a].getpf1() == 0)
        {
            cout.setf(ios::fixed);
            cout << setprecision(1) << data[j].stu.chj[a].getjidian1();
            cout << setiosflags(ios_base::right) << setprecision(0);
        }
        else
        {
            cout << "N/A";
        } //依照指定的总宽度按照格式输出课程绩点
        cout << " | " << endl;
    }
    cout << "是否继续查询(y/n)继续请按 y 退出按 n" << endl;
    cin >> n;
}
}

void search2(student stu[], database data[]) //查询学生排名和 GPA 的函数
{
    int paiming(student stu[], int n, double x); //调用查询学生排名的函数

```

```

int j = -1;
char n = 'y';
int g1 = g;
int p = 0;
int q = 0;
char num[20];
while (n == 'y')
{
    cout << "输入你要查询的学生学号:" << endl;
    cin >> num;//输入指定学生学号
    for (int i = 0; i < g1; i++)
    {
        if ((strcmp(num, stu[i].number) == 0))//在学生数组里找到满足条件的学生对象
        {
            j = i;
        }
    }
    if (j == -1)
    {
        cout << "没有你所要的学生!";
        cout << "是否退出?退出请按 n\n";
        cin >> n;
    }//没有找到并提示
    else {
        for (int i = 0; i < g1; i++)
        {
            double GPA2 = getGPA(stu[i].number, data);
            stu[i].GPA1 = GPA2;//计算出所有学生的 GPA
        }double x = stu[j].GPA1;//对指定学生 GPA 进行赋值处理
        int k1 = 0; int k2 = 0; int k3 = 0;
        int h = 0; int s = 0; int f = 0; int d = 0; int s1 = 0; int s2 = 0;
        char dep[20];
        cout << "姓名: " << stu[j].name << " ";
        cout << "学号: " << stu[j].number << " ";
        cout << "性别: " << stu[j].sex << " ";
        cout << "年级: " << stu[j].grade << " ";
        cout << "班号: " << stu[j].classno << " ";
        cout << "院系: " << stu[j].depart << " " << endl;//输出该学生基本信息
        student* stu3 = new student[20];
        student* same = new student[200];
        student* sum = new student[200];//为三个对象数组申请动态内存空间
        strcpy(dep, stu[j].depart);
        p = stu[j].grade;
        q = stu[j].classno;
    }
}

```

```

do
{
    if (stu[s].grade == p)
    {
        stu3[d] = stu[s];
        d++;
    }
    s++;
} while (s < g1); //将所有年级相同的学生合并为新的对象数组
do
{
    if (strcmp(dep, stu3[s1].depart) == 0)
    {
        same[h] = stu3[s1];
        h++;
    }
    s1++;
} while (s1 < d); //将上述新数组中系一样的学生合并为新对象数组
do
{
    if (same[s2].classno == q)
    {
        sum[f] = same[s2];
        f++;
    }
    s2++;
} while (s2 < h); //将上述新数组中班级一样的学生合并为新对象数组
cout << "级排名为: " << paiming(stu3, d, x) << endl; //计算该学生年级排名
cout << "系排名为: " << paiming(same, h, x) << endl; //计算该学生系排名
cout << "班排名为: " << paiming(sum, f, x) << endl; //计算该学生班级排名
cout.setf(ios::fixed); //指定输出格式
cout << "总体 GPA 为: " << setprecision(3) << x << endl; //输出该学生总 GPA
cout.unsetf(ios::fixed);
delete[] stu3;
delete[] same;
delete[] sum; //释放内存空间
}
cout << "是否继续查询(y/n)继续请按 y 退出按 n" << endl;
cin >> n;
}
}

int paiming(student stu[], int n, double x) //计算在指定学生数组中某一学生的排名
{
    int k1 = -2;

```

```

double a[20];
for (int i = 0; i < n; i++)
{
    a[i] = stu[i].GPA1;//将学生的 GPA 赋值到 a 数组中
}
if (n == 1)
{
    k1 = 1;//只有一个学生则排名为 1
}
else if (n == 2)
{
    if (a[0] < a[1])//对两个学生 GPA 进行从高到低的排序
    {
        double t = a[0]; a[0] = a[1]; a[1] = t;
    }
    for (int i = 0; i < 2; i++)
    {
        if (a[i] == x)k1 = i + 1;//在新排完序数组中查找指定学生 GPA 得到其在新数组中的位置进而得到排名
    }
}
else
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j1 = 0; j1 < n - 1 - i; j1++)
        {
            if (a[j1] < a[j1 + 1])
            {
                double t = a[j1]; a[j1] = a[j1 + 1]; a[j1 + 1] = t;//采用冒泡排序对多个学生 GPA 进行从高到低的排序
            }
        }
    }
    for (int i = 0; i < n; i++)
    {
        if (a[i] == x)k1 = i + 1;//在新排完序数组中查找指定学生 GPA 得到其在新数组中的位置进而得到排名
    }
}
return k1;//返回得到的排名
}

void revise(database data[], Xueqi xq[], student stu[])//修改学生指定学期成绩的函数
{

```

```

int k = 1, j = -1, c, a[10], b1;
int flgh;
int h;
char n = 'y';
int d;
cout << "-----输入开始! ----" << endl;
cout << "请输入考试年份: ";
cin >> year;
cout << "请输入考试学期 (春季输入 1, 夏季输入 2, 秋季输入 3): ";
cin >> c;
if (c == 1) { strcpy(season, "春季"); }
else if (c == 2) { strcpy(season, "夏季"); }
else if (c == 3) { strcpy(season, "秋季"); }
else {
    do {
        cout << "输入不合规范, 请重新输入: ";
        cin >> c;
    } while (c != 1 && c != 2 && c != 3);
} //进行提示, 输入相关信息
while (k == 1)
{
    cout << "请输入你要修改成绩的学生学号" << endl;
    cin >> number;
    flgh = 0;
    for (int i = 0; i < 240; i++)
    {
        if ((strcmp(number, data[i].stu.number) == 0) && (strcmp(season,
data[i].xq.season) == 0) && data[i].xq.year == year) //查找指定学期指定学号的具体对象
        {
            j = i;
        }
    }
    if (j == -1)
    {
        cout << "没有你所要的学生!";
        cout << "是否退出?退出请按 n\n";
        cin >> n; //提示没有找到的信息
    }
    else {
        char kcn[30];
        cout << "请输入要修改的课程名: " << endl;
        cin >> kcn; //输入修改的课程名
        int b = data[j].stu.shu;
        for (int i = 0; i < b; i++)

```

```

        {
            if (strcmp(data[j].stu.chj[i].kcname, kcn) == 0)//对查到的数据对象中的
            学生成绩数组进行遍历查找，找到指定课程名对应的成绩对象
            {
                data[j].stu.genxf = data[j].stu.genxf -
data[j].stu.chj[i].getxf1();
                data[j].stu.xfj = data[j].stu.xfj - (data[j].stu.chj[i].getxf1())
* (data[j].stu.chj[i].getjidian1());
                cout << "请重新输入课程学分: " << endl;
                cin >> xf;
                data[j].stu.chj[i].setxf(xf);
                data[j].stu.genxf = data[j].stu.genxf +
data[j].stu.chj[i].getxf1();
                cout << "请重新输入该课程属性（是否记 P/F）？（是输入 1，否输入 0）: "
<< endl;

                cin >> pf;
                data[j].stu.chj[i].setpf(pf);
                cout << "请重新输入该课程百分制成绩 " << endl;
                cin >> cj100;
                data[j].stu.chj[i].setcj100(cj100);
                data[j].stu.chj[i].calculate();
                cout << "该课程的成绩为"; data[j].stu.chj[i].getcj(); cout <<
endl;

                cout << "该课程的百分制成绩为: "; data[j].stu.chj[i].getcj100();
cout << endl;

                if (data[j].stu.chj[i].getpf1() == 1) { cout << " 该课程的绩点为:
"; data[j].stu.chj[i].getjidianpf(); cout << endl; }
                else { cout << "该课程的绩点为: "; cout.setf(ios::fixed); cout <<
setprecision(1); data[j].stu.chj[i].getjidian(); cout << setiosflags(ios_base::right)
<< setprecision(0); cout << endl; }
                cout << "该课程的百分制成绩为: "; data[j].stu.chj[i].getcj100();
cout << endl;

                data[j].stu.xfj = data[j].stu.xfj + (data[j].stu.chj[i].getxf1())
* (data[j].stu.chj[i].getjidian1());
                if (data[j].stu.genxf == 0) { data[j].stu.GPA = 0; }
                else { data[j].stu.GPA = data[j].stu.xfj / data[j].stu.genxf; }//
显示修正以后的成绩各属性
            }
        }
        cout << "|          课程名          |" << " 学分 |" << " 成绩 |" << " 百分
制成绩 |" << " 绩点 |" << endl;
        for (int a = 0; a < data[j].stu.shu; a++)
        {
            int len = data[j].stu.chj[a].getkcnamelen();

```

```

        int len1 = floor((30 - len) / 2);
        cout << "|";
        cout << setfill(' ') << setw(len1 + len);
        data[j].stu.chj[a].getkcname();
        cout << setfill(' ') << setw(30 - len1 - len);
        cout << "| ";
        if (data[j].stu.chj[a].getxf1() < 10) { data[j].stu.chj[a].getxf();
cout << " | "; }
            else if (data[j].stu.chj[a].getxf1() >= 10)
{ data[j].stu.chj[a].getxf(); cout << " | "; }
            int len2 = data[j].stu.chj[a].getcjlent();
            int len3 = (7 - len2) / 2;
            cout << setfill(' ') << setw(len3 - 1);
            data[j].stu.chj[a].getcj();
            cout << setfill(' ') << setw(7 - len2 - len3) << " | ";
            cout.setf(ios::fixed);
            cout << setiosflags(ios_base::right) << setprecision(0);
            if (data[j].stu.chj[a].getcj1001() < 10) { cout <<
setiosflags(ios_base::right) << setprecision(0); cout << "00";
data[j].stu.chj[a].getcj100(); cout << " "; }
            else if (data[j].stu.chj[a].getcj1001() >= 10 &&
data[j].stu.chj[a].getcj1001() < 100) { cout << setiosflags(ios_base::right) <<
setprecision(0); cout << "0"; data[j].stu.chj[a].getcj100(); cout << " "; }
            else {
                cout << setiosflags(ios_base::right) << setprecision(0);
                data[j].stu.chj[a].getcj100(); cout << " ";
            }
            cout << " | ";
            if (data[j].stu.chj[a].getpf1() == 0)
            {
                cout.setf(ios::fixed);
                cout << setprecision(1) << data[j].stu.chj[a].getjidian1();
                cout << setiosflags(ios_base::right) << setprecision(0);
            }
            else
            {
                cout << "N/A";
            }
            cout << " | " << endl;
        } //同前面的查询函数，指定格式对该学期修正后的所有成绩进行输出展示
    }
    cout << "还要继续吗?退出请按 0,继续请按 1: " << endl;
    cin >> k;
}

```



```

}
void caidan()//显示开始的总菜单
{
    string a;
    getline(cin, a);
    int b = 8 + a.size();
    cout << "*****欢迎进入学生成绩管理系统!*****" << endl;
    cout << setfill('*') << setw(40) << "" << endl;
    cout.fill(' ');
    cout.width(b);
    cout.setf(ios::right);//设定格式
    cout << a << "学生成绩管理系统" << endl;
    cout << setfill('*') << setw(40) << "" << endl;
    cout << setfill('+') << setw(40) << "" << endl;
    cout.fill(' ');
    cout.width(20);
    cout.setf(ios::right);//设定格式
    cout << "请选择版本" << endl;
    cout.width(22);
    cout.setf(ios::right);
    cout << "1.管理员版" << endl;
    cout.width(22);
    cout.setf(ios::right);
    cout << "2.大学生版" << endl;
    cout.width(22);
    cout.setf(ios::right);
    cout << "3.退出系统" << endl;
    cout.width(22);
    cout.setf(ios::right);
    cout << "请选择序号（1-3）" << endl;//设定三个选项并提示进行选择
    cout << setfill('*') << setw(40) << "" << endl;
}

void caidan2(adminis ad)//选择选项 1 并输入用户名和密码后进入成绩管理系统的管理员端口
{
    string a;
    getline(cin, a);
    int b = 8 + a.size();
    cout << setfill('*') << setw(40) << "" << endl;
    cout.fill(' ');
    cout.width(b);
    cout.setf(ios::right);
    cout << a << "学生成绩管理系统（管理员版）" << endl;
}

```

```
    cout << a << "欢迎回来, " << ad.name << " No." << ad.gonghao << endl; //根据用户名  
    显示管理员的相关信息
```

```
    cout << setfill('*') << setw(40) << "" << endl;  
    cout << setfill('+') << setw(40) << "" << endl;  
    cout.fill(' ');  
    cout.width(20);  
    cout.setf(ios::right);  
    cout << "主菜单" << endl;  
    cout.width(22);  
    cout.setf(ios::right);  
    cout << "1.成绩录入" << endl;  
    cout.width(22);  
    cout.setf(ios::right);  
    cout << "2.成绩查询" << endl;  
    cout.width(22);  
    cout.setf(ios::right);  
    cout << "3.排名查询" << endl;  
    cout.width(22);  
    cout.setf(ios::right);  
    cout << "4.成绩修改" << endl;  
    cout.width(22);  
    cout.setf(ios::right);  
    cout << "5.退出系统" << endl;  
    cout.width(26);  
    cout.setf(ios::right);  
    cout << "请选择序号 (1-5) " << endl; //设置五个选项并提示输入选项  
    cout << setfill('*') << setw(40) << "" << endl; //设定格式  
}  
void caidan3() //选择选项 2 后进入成绩管理系统的学生端口  
{
```

```
    string a;  
    getline(cin, a);  
    int b = 8 + a.size();  
    cout << setfill('*') << setw(40) << "" << endl;  
    cout.fill(' ');  
    cout.width(b);  
    cout.setf(ios::right);  
    cout << a << "学生成绩管理系统 (学生版) " << endl;  
    cout << setfill('*') << setw(40) << "" << endl;  
    cout << setfill('+') << setw(40) << "" << endl;  
    cout.fill(' ');  
    cout.width(20);  
    cout.setf(ios::right);  
    cout << "主菜单" << endl;
```

```

        cout.width(22);
        cout.setf(ios::right);
        cout << "1.成绩查询" << endl;
        cout.width(22);
        cout.setf(ios::right);
        cout << "2.排名查询" << endl;
        cout.width(22);
        cout.setf(ios::right);
        cout << "3.退出系统" << endl;
        cout.width(26);
        cout.setf(ios::right);
        cout << "请选择序号（1-3）" << endl;//设置三个选项并提示输入选项
        cout << setfill('*') << setw(40) << "" << endl;//设定格式
    }
int main()//主函数
{
    student* stu = new student[20];
    Xueqi* xq = new Xueqi[12];
    database* data = new database[240];//定义三个基本对象数组并为其分配动态内存空间
    adminis ad[5] =
    {
        "wxq",1,"wxqtsinghua.edu.cn","234755687sddvcEFCTd!dfgf",
        "zxd",2,"zxdtsinghua.edu.cn","8237rtghdsfgRTDFYR%TREF^t",
        "shc",3,"shctsinghua.edu.cn","iwue3hrbu887ydsfUV&^&RR$#4",
        "sls",4,"slstsinghua.edu.cn","vytr567rfut5r5E%$EDYCRTY",
        "zzy",5,"zzytsinghua.edu.cn","Gugy7u65tf7vytjFDRFY%^jb67"
    };//为结构体数组进行赋值
    void input(adminis ad);//将结构体数组信息写入指定文件
    int ch;
    while (1)//进入选择界面
    {
        B:
        string a;
        int fg;
        char useri[30];
        char passwo[50];
        ifstream in;
        ofstream ofile;
        ifstream ifile;//定义基本文件输入输出对象
        caidan();//显示总菜单
        if (!ofile || !ifile || !in)
        {
            cerr << "文件打开错误" << endl;
            abort();
        }
    }
}

```

```

} //判断文件是否打开成功
cin >> fg; //输入总菜单提示的选项值
if (fg == 1) //选择进入管理员端口
{
    char n = 'y';
    int j = -1;
    in.open("administrator.bin", ios::in); //打开存储管理员信息的文件
    for (int i = 0; i < 5; i++)
    {
        in.read((char*)&ad[i], sizeof(ad[i])); //读入管理员的相关信息
    }
    in.close();
    do {
        cout << "请输入用户名" << endl;
        cin >> useri;
        cout << "请输入密码" << endl;
        cin >> passwo; //输入管理员相关信息
        for (int i = 0; i < 5; i++)
        {
            if ((strcmp(useri, ad[i].userid) == 0) && (strcmp(passwo,
ad[i].password) == 0)) //判断管理员信息输入是否正确
            {
                j = i;
            }
        }
        if (j == -1)
        {
            cout << "用户名或密码输入错误, 请退出重试" << endl << endl;
            goto B; //提示输入错误并返回总菜单
        }
        else {
            A:
            caidan2(ad[j]); //打开管理系统(管理员版)菜单
            cin >> ch; //输入管理员菜单提示的选项
            switch (ch) //对输入的选项值进行处理
            {
                case 1: ofile.open("grade management system.bin", ios::out |
ios::binary); //打开文件并进行写入操作, 指定文件操作对象
                    g = shuru(stu, xq, data); //输入学生成绩
                    for (int i = 0; i < g; i++)
                    {
                        ofile.write(reinterpret_cast<char*>(&stu[i]),
sizeof(stu[i]));
                    } //向文件写入学生数组信息

```

```

        for (int i = b1 * 20; i < b1 * 20 + g; i++)
        {
            ofile.write(reinterpret_cast<char*>(&data[i]),
sizeof(data[i]));

        }//向文件写入数据数组信息
        ofile.close();//关闭文件
        goto A;//返回管理系统（管理员版）菜单
    case 2: ifile.open("grade management system.bin", ios::in |
ios::binary);//打开文件并进行读出操作，指定文件操作对象
        for (int i = 0; i < g; i++)
        {
            ifile.read(reinterpret_cast<char*>(&stu[i]),
sizeof(stu[i]));

        }//读入学生数组信息
        for (int i = b1 * 20; i < b1 * 20 + g; i++)
        {
            ifile.read(reinterpret_cast<char*>(&data[i]),
sizeof(data[i]));

        }
        for (int i = 0; i < 240; i++)
        {
            ifile.read(reinterpret_cast<char*>(&data[i]),
sizeof(data[i]));

        }//读入数据数组信息
        search(data);//在数据数组中查询学生成绩
        ifile.close();//关闭文件
        goto A;//返回管理系统（管理员版）菜单
    case 3: ifile.open("grade management system.bin", ios::in |
ios::binary);//打开文件并进行读出操作，指定文件操作对象
        for (int i = 0; i < g; i++)
        {
            ifile.read(reinterpret_cast<char*>(&stu[i]),
sizeof(stu[i]));

        }//读入学生数组信息
        for (int i = b1 * 20; i < b1 * 20 + g; i++)
        {
            ifile.read(reinterpret_cast<char*>(&data[i]),
sizeof(data[i]));

        }
        for (int i = 0; i < 240; i++)
        {
            ifile.read(reinterpret_cast<char*>(&data[i]),
sizeof(data[i]));

        }//读入数据数组信息

```

```

        search2(stu, data); // 查询学生 GPA 和排名
        ifile.close(); // 关闭文件
        goto A; // 返回管理系统（管理员版）菜单
        case 4: ofile.open("grade management system.bin", ios::out |
ios::binary); // 打开文件并进行写入操作，指定文件操作对象
                ifile.open("grade management system.bin", ios::in |
ios::binary); // 打开文件并进行读出操作，指定文件操作对象
                for (int i = 0; i < g; i++)
                {
                    ifile.read((char*)&stu[i], sizeof(stu[i]));
                } // 读入学生数组信息
                for (int i = 0; i < 240; i++)
                {
                    ifile.read((char*)&data[i], sizeof(data[i]));
                } // 读入数据数组信息
                revise(data, xq, stu); // 修改学生成绩
                for (int i = b1 * 20; i < b1 * 20 + g; i++)
                {
                    ofile.write(reinterpret_cast<char*>(&data[i]),
sizeof(data[i]));

                } // 向文件写入数据数组信息
                ifile.close();
                ofile.close(); // 关闭文件
                goto A; // 返回管理系统（管理员版）菜单
        case 5: cout << "系统已关闭" << endl; // 提示退出信息
                goto B; // 返回总菜单
            }
        }
    } while (n == 'y');
}
else if (fg == 2) // 选择进入学生端口
{
    C:
        caidan3(); // 打开管理系统（学生版）菜单
        cin >> ch; // 输入学生菜单提示的选项
        switch (ch) // 对输入的选项值进行处理
        {
            case 1: ifile.open("grade management system.bin", ios::in | ios::binary); //
打开文件并进行读出操作，指定文件操作对象
                    for (int i = 0; i < g; i++)
                    {
                        ifile.read(reinterpret_cast<char*>(&stu[i]), sizeof(stu[i]));
                    } // 读入学生数组信息
                    for (int i = b1 * 20; i < b1 * 20 + g; i++)

```

```

        {
            ifile.read(reinterpret_cast<char*>(&data[i]), sizeof(data[i]));
        }
        for (int i = 0; i < 240; i++)
        {
            ifile.read(reinterpret_cast<char*>(&data[i]), sizeof(data[i]));
        } //读入数据数组信息
        search(data); //在数据数组中查询学生成绩
        ifile.close(); //关闭文件
        goto C; //返回管理系统（学生版）菜单
    case 2: ifile.open("grade management system.bin", ios::in | ios::binary); //
打开文件并进行读出操作，指定文件操作对象
        for (int i = 0; i < g; i++)
        {
            ifile.read(reinterpret_cast<char*>(&stu[i]), sizeof(stu[i]));
        } //读入学生数组信息
        for (int i = b1 * 20; i < b1 * 20 + g; i++)
        {
            ifile.read(reinterpret_cast<char*>(&data[i]), sizeof(data[i]));
        }
        for (int i = 0; i < 240; i++)
        {
            ifile.read(reinterpret_cast<char*>(&data[i]), sizeof(data[i]));
        } //读入数据数组信息
        search2(stu, data); //查询学生 GPA 和排名
        ifile.close(); //关闭文件
        goto C; //返回管理系统（学生版）菜单
    case 3: cout << "系统已关闭" << endl; //提示退出信息
        goto B; //返回总菜单
    }
}
else exit(0); //退出系统
}
delete[] data;
delete[] stu;
delete[] xq; //释放内存空间
return 0;
}

```

附录二：大作业任务书

1. 课程设计目的

面向对象程序设计课程设计是集中实践性环节之一，是学习完《计算机程序设计基础 2》C++面向对象程序设计课程后进行的一次全面的综合练习。要求学生达到熟练掌握 C++语言的基本知识和技能；基本掌握面向对象程序设计的思想和方法；能够利用所学的基本知识和技能，解决简单的面向对象程序设计问题，从而提高动手编程解决实际问题的能力。**尤其重视创新思维培养。**

2. 课题题目

学生成绩管理系统（或公司人事管理系统）

3. 文档设计要求

3.1 设计课题题目：每个同学都单独完成 1 道课题。后面有范题，仅供同学们参考，不列入本次课程设计的课题。

3.2 对于程设题目，按照范题的格式。自行虚构软件需求。并按照第 4 点要求，编写设计文档。基本要求系统中设计的类的数目不少于 4 个，每个类中要有各自的属性（多于 3 个）和方法（多于 3 个）；需要定义一个抽象类，采用继承方式派生这些类。并设计一个多重继承的派生类。在程序设计中，引入虚函数的多态性、运算符重载等机制。

4. 程序设计的基本要求

- （1）要求利用面向对象的方法以及 C++的编程思想来完成系统的设计；
- （2）要求在设计的过程中，建立清晰的类层次；
- （3）根据课题完成以下主要工作：①完成系统需求分析：包括系统设计目的与意义；系统功能需求（系统流程图）；输入输出的要求。②完成系统总体设计：包括系统功能分析；系统功能模块划分与设计（系统功能模块图）。③完成系统详细设计：数据文件；类层次图；界面设计与各功能模块实现。④系统调试：调试出现的主要问题，编译语法错误及修改，重点是运行逻辑问题修改和调整。⑤使用说明书及编程体会：说明如何使用你编写的程序，详细列出每一步的操作步骤。⑥关键源程序（带注释）。
- （4）自己设计测试数据，将测试数据存在文件中，通过文件进行数据读写来获

得测试结果。

(5) 按规定格式完成课程设计报告，并在网络学堂上按时提交。

(6) 不得抄袭他人程序、课程设计报告，每个人应独立完成，在程序和设计报告中体现自己的个性设计。

5. 进度安排

小学期 第2周	节次	时间	计划安排
	1	7月3日 11:30-15:00	进行初步规划及抽象类和类以及结构体的设计，完成输入函数设计
	2	7月4日 8:00-11:30	在此基础上进一步思索功能布局，设计三种菜单以及管理员登录机制，完成查询成绩的功能
	3	7月5日 18:30-22:00	实现读取不同学期成绩的功能，并设计计算 GPA 与学生排名的函数
	4	7月6日 15:00-18:30	完成修改学生成绩功能的设计，并对所有功能进行整合、封装，完成 main 函数设计，同时实现文件写入与读取功能
	5	7月7日 11:30-15:00	进行整体程序调试，进行适度调整、改正与补充和最终完善

附录三：评分表

项 目	评 价	
设计方案的合理性与创新性	6	
设计与调试结果	8	
设计说明书的质量	2	
程序基本要求涵盖情况	8	
程序代码编写素养情况	4	
课程设计周表现情况	2	
综合成绩	30	