

《C++面向对象程序设计》

课程设计报告

**专业年级：** 20级计科4班

**学生姓名:**  曹侃 **学 号:** 20201542

**课设题目：** 学生成绩管理

**指导教师:**  蒋湘涛

**日 期：** 2022年5月8日

1. 课程设计题目及要求

问题描述：

设学生信息包括：学号、姓名、学期、每门课程的成绩(每学期的课程门数是不一样的) ，对学生的成绩信息进行管理。

实现要求：

实现：学生信息的录入；修改；删除和查询，按学期、学号、成绩不及格等查询。

⑴输入学生的成绩信息，包含学号、姓名、性别等基本信息和各课成绩

⑵显示全部学生各科成绩信息；

⑶对各科成绩统计分析（总分、平均分、最高分、最低分、及格率等）；

⑷统计各科各分数段人数；

⑸按学号或姓名查找并显示某个学生的各科成绩；

⑹按课程成绩或总分由高到低排序显示；

⑺更新某个学生的基本信息或课程成绩；

⑻设计一个字符菜单界面，具有上述规定的操作要求、退出系统等最基本的功能。

1. 总体分析与设计

该程序包括2个头文件和1个源文件和3个文本文件。3个头文件名分别是：operator.h、GradeManagement.h；源文件名为Test.cpp，

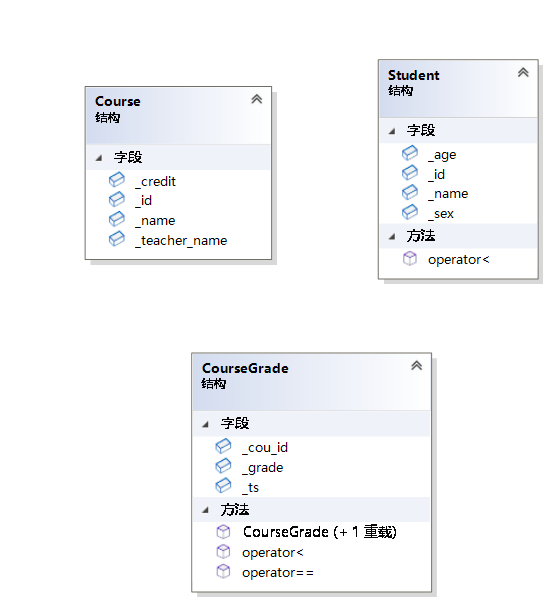
3个文本文件名分别是stu\_info.txt、cou\_intfo.txt、stu\_grade.txt。

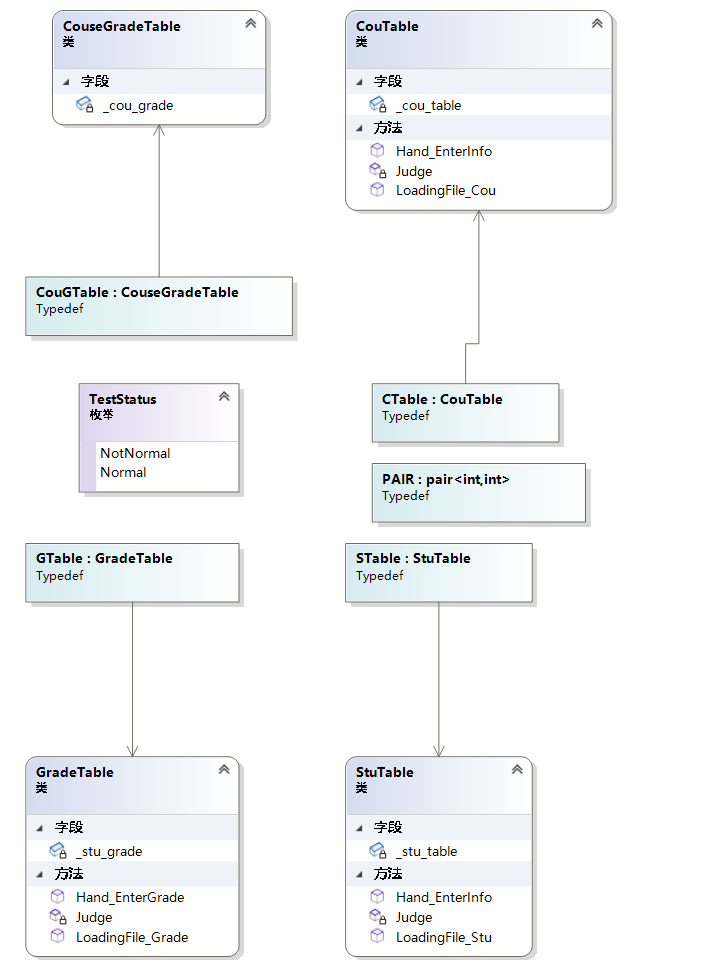
其具体作用如下：

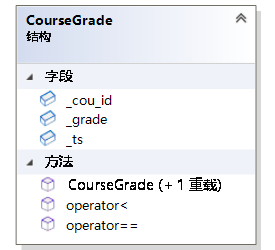
|  |  |
| --- | --- |
| 文件名 | 作用 |
| operator.h | 程序功能函数的具体实现 |
| GradeManagement.h | 头文件和自定义类型的声明 |
| Test.cpp | 启动程序 |
| stu\_info.txt | 学生信息文件 |
| cou\_intfo.txt | 课程信息文件 |
| stu\_grade.txt | 学生成绩文件 |

1. 详细设计

本程序有多个类，类中具体内容如下图所示：







1. 具体设计思路

由设计需求可知，增删查改，用的最频繁的功能会是查询，同时学号与学生之间，课程与课程号之间也存在映射关系，自然选用STL库中的map和set作为容器,因为其底层是红黑树实现，效率很高，但是由于学生与成绩多对多，故容易出现数据冗余，结合数据库范式相关知识决定将学号作为学生的唯一标识，将课程号作为课程的唯一标识。

获得信息的方式都是通过映射，成绩只存储一份在总成绩单中，从总成绩单中获得单科成绩。

通过学号获得学生的具体信息，通过课程号获得课程的具体信息。

排序选用algorithm中的sort函数，底层是快排，效率也十分可观，同时利用记录等方式避免多次重复排序。

**核心类**

Couse类

存储着课程的基本信息



Student类

存储着学生的基本信息



StuTable

学生信息表，可通过文件或者手动录入的方式录入数据



CouTable

课程信息表，可通过文件或者手动录入的方式录入数据



CourseGrade类

CourseGrade存储着课程号、成绩和考试状态（缺考，作弊等定义为不正常，成绩置为0）。通过课程号可以获取课程的具体信息。



GradeTable类

GradeTable是核心类，通过学号可以找到这个学生所有的课程成绩。学号可以找到一个set容器，容器里放的是CourseGrade类型。



CourseGradeTab类

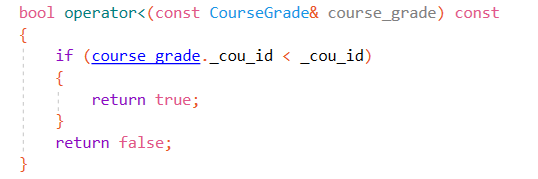
课程成绩单，不存储具体信息，只存储选了这门课的学生学号，通过学号再去获取这门课的成绩和学生的具体信息。



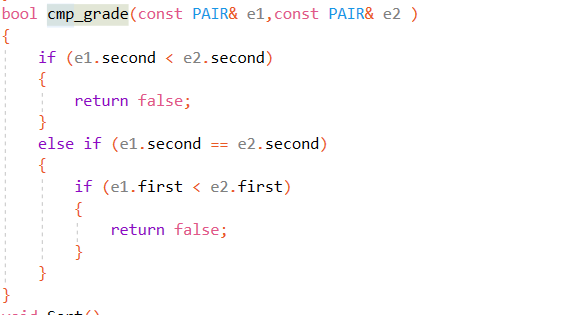
**算法**

下面算法均只举一个例子，代码实现中可能实现多个，但是实现思路大体相同。

map和set排序自定义类型，可以通过仿函数和重载运算符的方法，这里采用后一种，重载了小于号（bool operator<(...)）

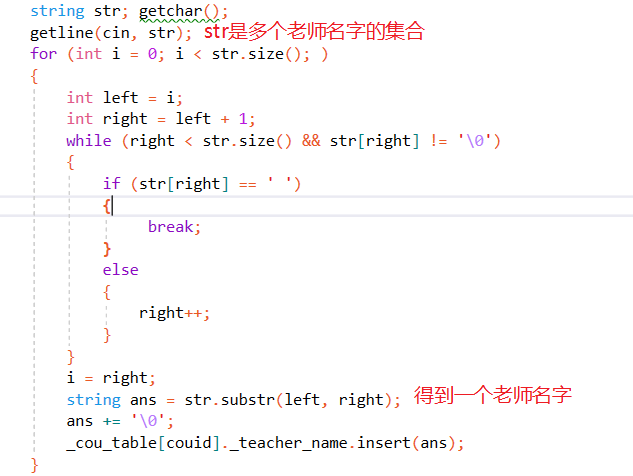


map为Key-Value结构，默认以Key排序，排序成绩时需要通过Value。所以这里根据value对map进行排序,写一个比较函数，再利用库函数sort进行自定义排序





切割字符串，以空格为间隔符切割字符串提取出授课老师的姓名



**数据结构**

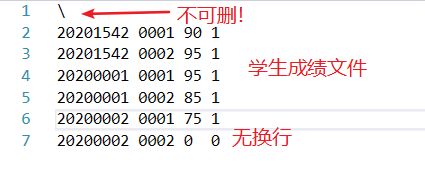
数据结构采用红黑树、顺序表和字符串，表现为map,set,vector和string。利用合理的数据结构和准确的映射（哈希）可以大大提高程序运行的效率

**输入与输出**

输入采用文件读取和手动输入两种方式，文件读取用了ifstream流,手动输入和输出通过外部设备（显示器）完成。

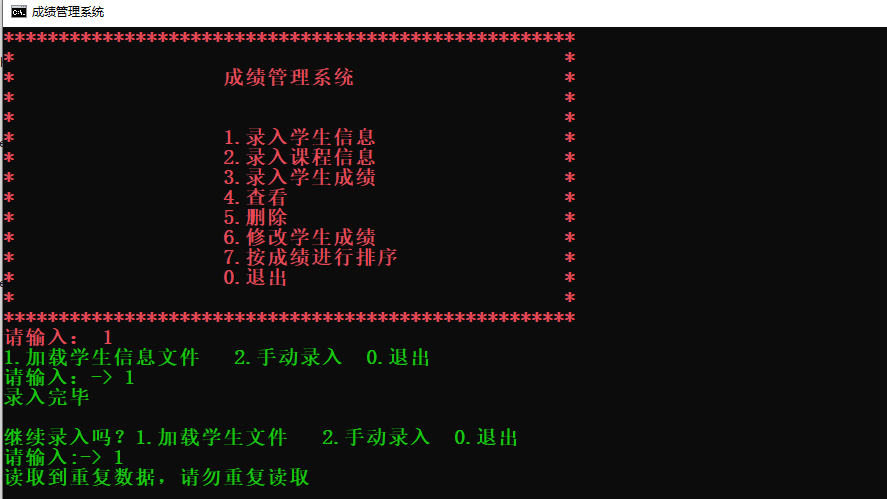
此外，程序功能过多，使用起来界面难以区分，利用部分Windows编程修改了控制台的标签，字体，颜色等，功能的颜色也不尽相同，便于用户操作。

文件读取时由于需要判断文件是否为空和文件是否存在，所以文件首行加了字符’/’作为标志，若去掉会导致读取错误，如信息遗漏，此外为了防止末行数据重复读取，文件末尾并没有设置换行！

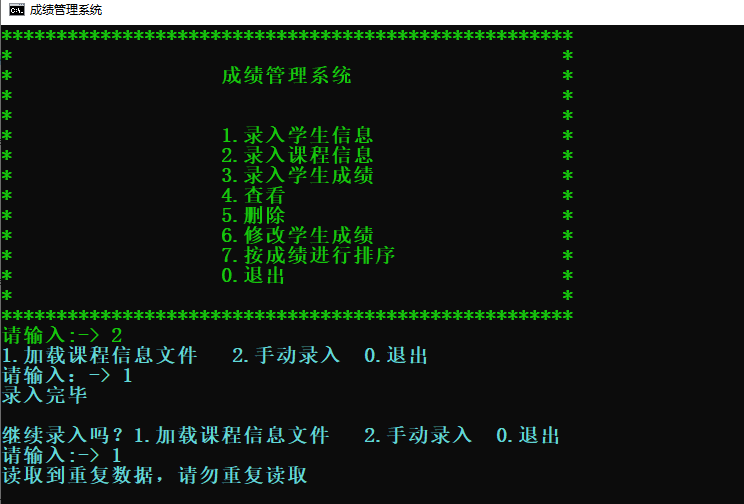


1. 调试与测试

加载学生文件



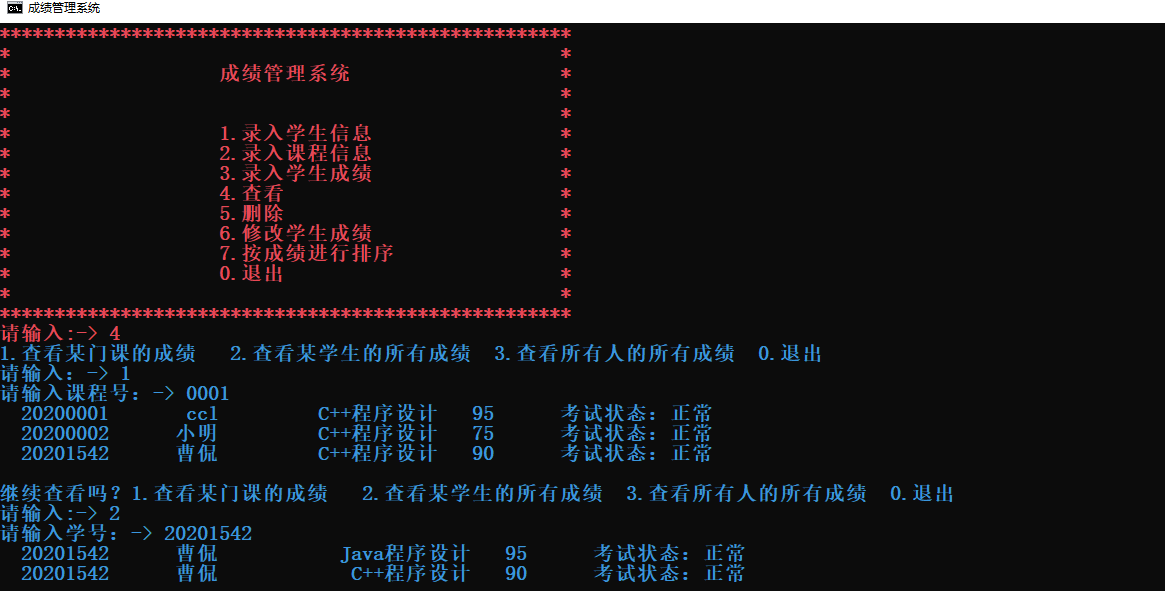
加载课程文件



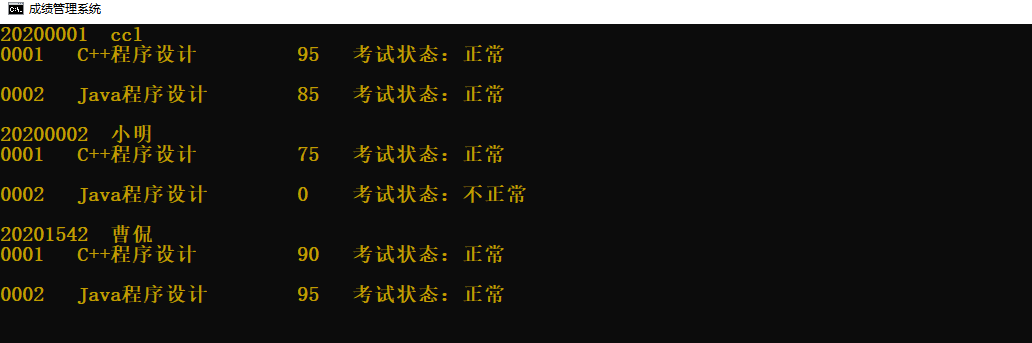
加载学生成绩文件



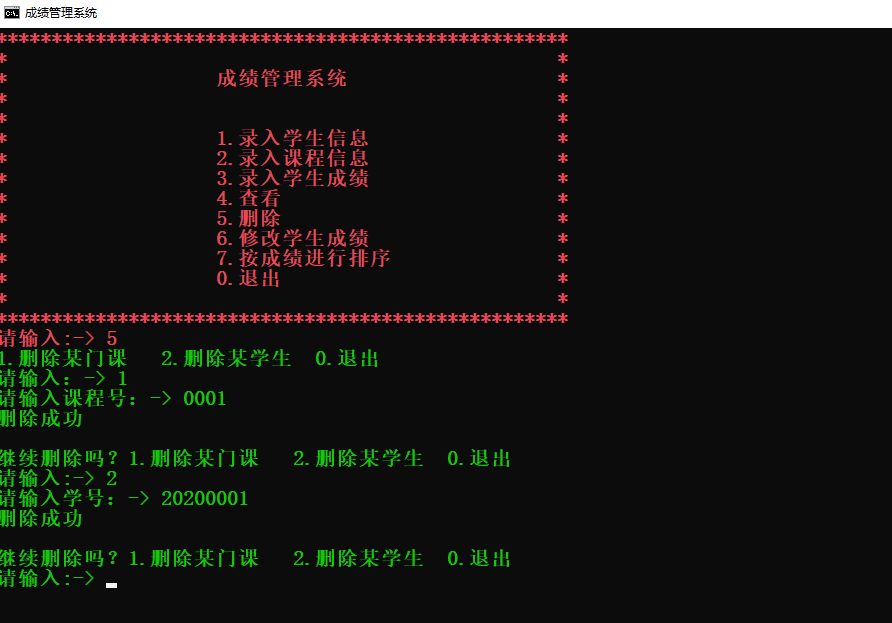
查看成绩



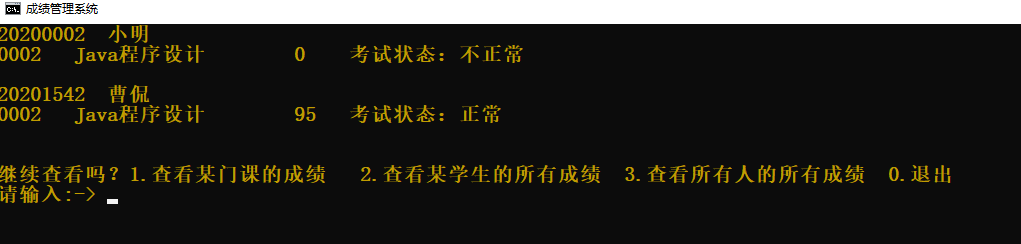
查看所有人成绩（自动清屏）



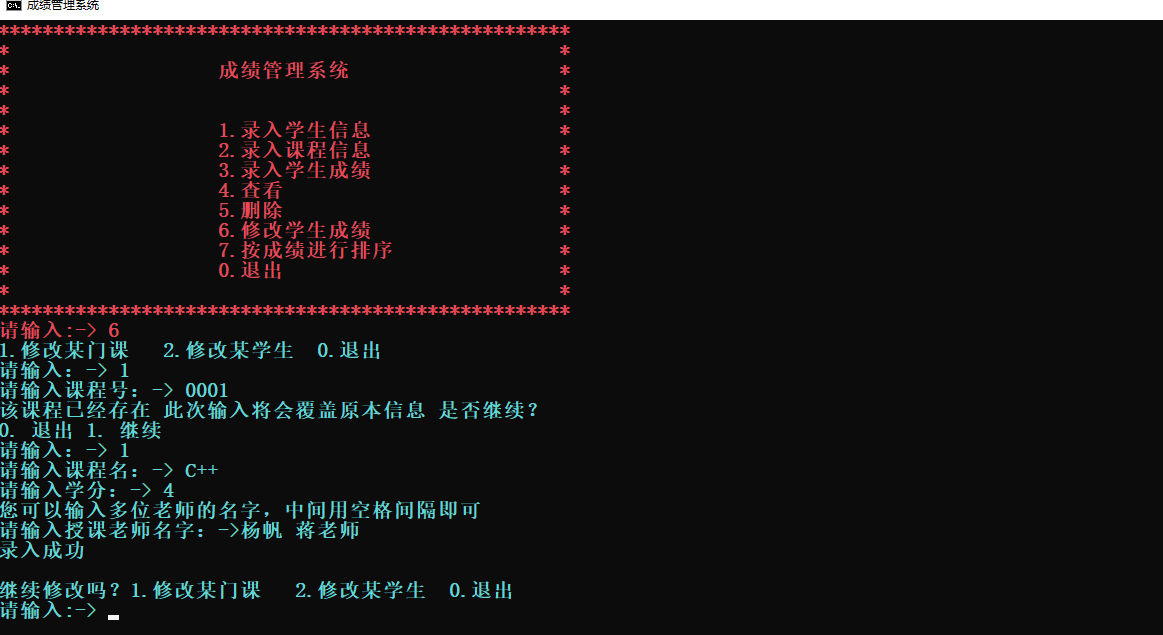
删除



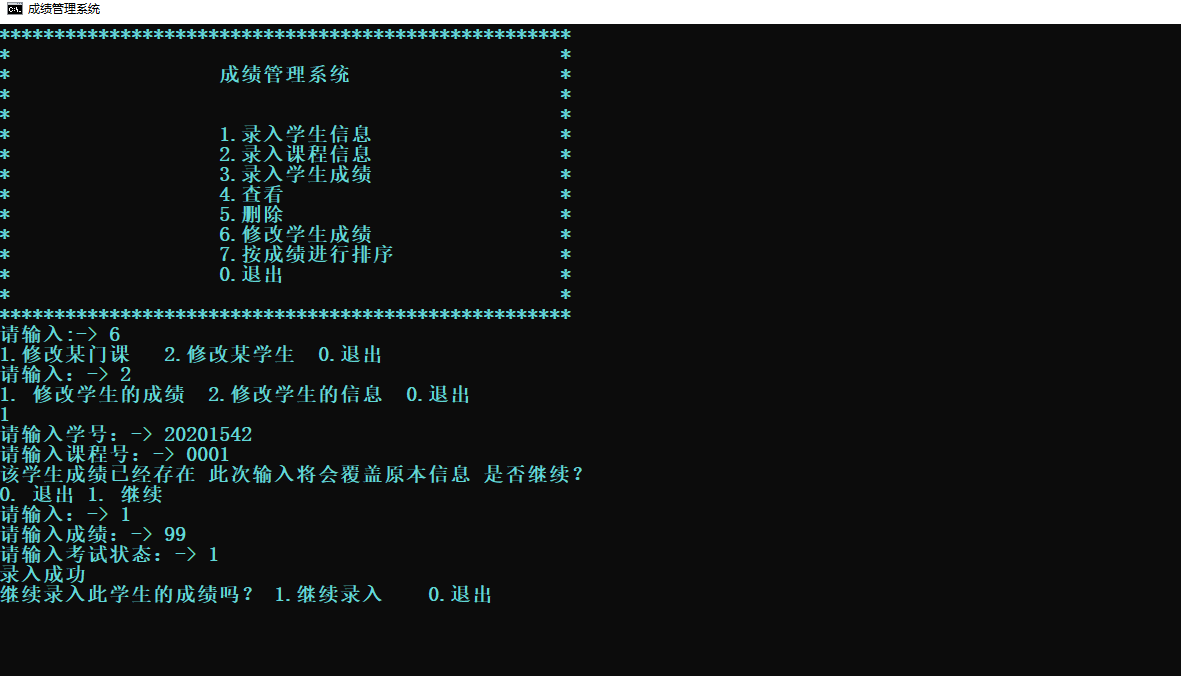
删除后查看所有人成绩



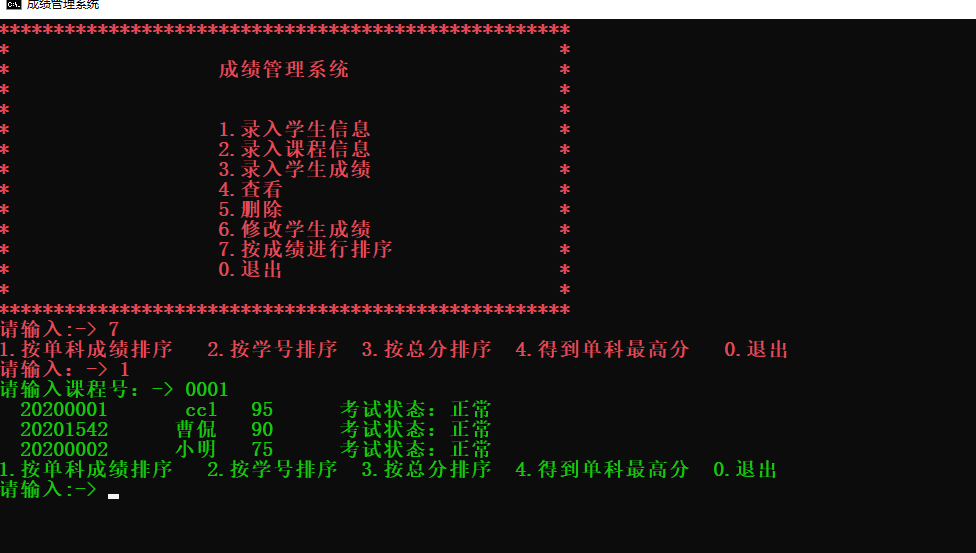
修改学生成绩

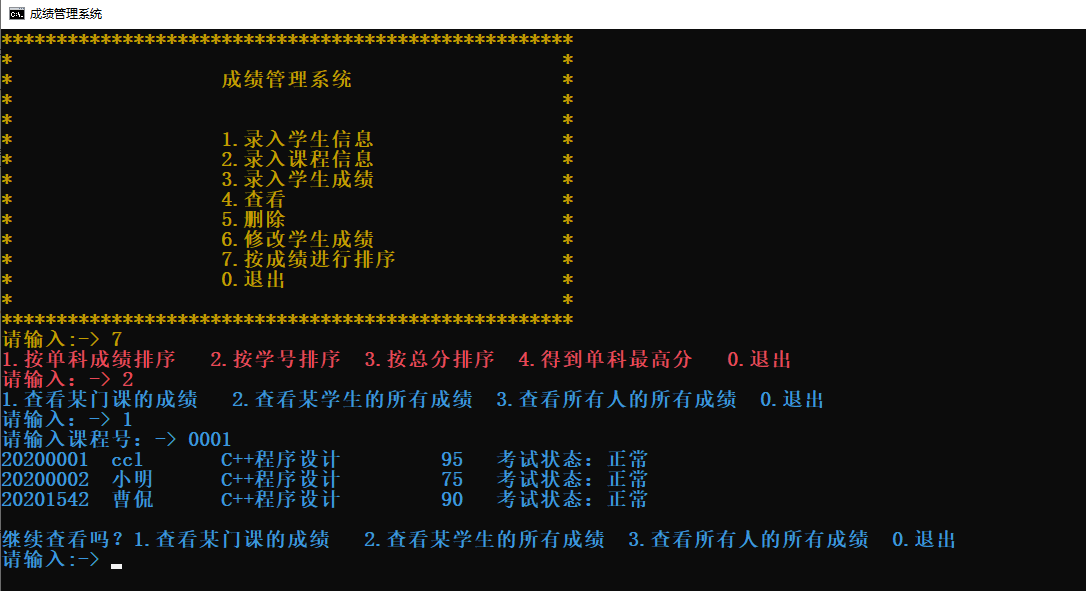


更改学生信息



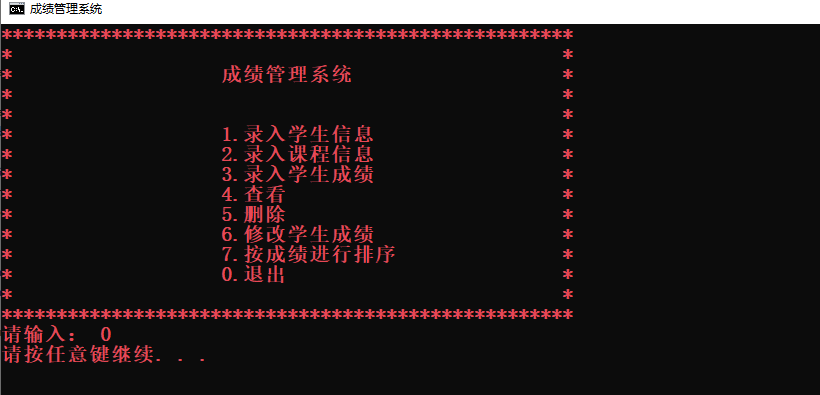
对学生成绩进行排序和分类







退出程序



1. 总结

1.数据冗余的处理是一件很麻烦的事情，要做好数据冗余的处理需要设计出合理的结构

2.这次课设的实现中既有亮点也有不足。

亮点在于自定义排序map和哈希的思想去访问数据，利用部分Windows编程美化控制台，对一些非法情况进行了处理，提高了代码的健壮性。在程序中加入了io流，使其具有了记忆性（其中不足在于没能把手动录入的数据写入原先文件）

不足在于结构设计方面不够优秀导致实现起来较为麻烦，代码实现中出现了重复代码，由于设计的时间原因未能很好的将一些重复的代码进行封装。简而言之，结构和代码实现可以再进行优化。有些许遗憾的是有些能用上模板的地方未能用上模板。

3.复习了C++的基础知识，再度理解封装的概念，对类的设计方面有了更深的理解，同时设计过程中也在思考数据的一致性和如何减少数据的冗余。

总而言之，这次课设对我个人的编程能力是一次很好的锻炼，同时也开拓了我的知识面。