

课程设计报告书

**题目：学生信息管理系统**

**学 院 计算机科学与工程学院**

**专 业 20计算机类二班**

**学生姓名 罗俊龙**

**学生学号 202030444506**

**指导教师 沃 焱**

**课程编号 045101571**

**课程学分 2**

**起始日期 2021年3月至2021年6月**

|  |  |
| --- | --- |
| 教  师  评  语 | 该生完成了《学生信息管理系统》的开发与设计，该系统包括学生模块，教师模块，管理员模块  其中  学生模块完成了信息查询删改，成绩和班级功能  教师模块完成了添加成绩，管理班级信息功能  管理员模块完成了学期管理、班级管理，和整个数据库的修改与查看功能  代码测试正确，程序运行基本正常、设计合理、功能完整、ppt展示思路基本清晰，能正确回答问题    教师签名：沃焱  日期：2021年6月22日 |
| 成  绩  评  定 | 87 |
| 备  注 |  |

**学生信息管理系统**

**一、选题背景**

对于每个高校，都会有一套自己的学生信息管理系统供全体师生使用。在实际的高校信息管理中，需要保存和管理的数据很多。最基础的内容包括每个学生的个人信息，学期表现，成绩，班级等信息，并需要向学生和对应的教师提供查看和修改的窗口，进行实时的数据管理。

学生信息管理系统的好坏会一定程度上关系到师生对学校的满意度，也关系到一所高校行政管理效率的高低。因此，设计一个执行效率高，功能齐全的信息管理系统是十分有必要的。在本次课程设计中本人将运用所学的C++知识，设计出一个具有简单基础功能的学生信息管理系统。

在本次设计的系统中，内含三大模块：学生模块、教师模块、管理员模块，每个模块拥有不同的权限和功能，供给3种不同身份的人使用。系统需要实现的功能包括以下几点：

1.登录系统：账号注册、登录、删改密码功能

2.信息系统：存储每个学生和教师拥有姓名、ID、电话号码等信息、可以实现对自己信息的修改和对他人信息的查看。

3.班级系统：每个学生可以选择加入不同班级，查看班级名单和同班同学信息。教师以班级为单位对学生数据进行管理

4.成绩&学期系统：可细分到不同学期的不同科目。科目拥有科目名、绩点，学分，分数等数据。每个学期具有不同科目数据。由教师对班级进行添加和删除科目。可以实时计算某学期内所有学生的GPA和加权平均分。提供不同类型的成绩排序功能。

5.管理员&教师系统：管理可以查看该系统内所有数据。拥有查看全体名单，发放教师ID,添加和删除班级、学期等功能。教师可以修改本班各学生成绩。

6.文件存储:文件存储和读取功能，保存系统内数据

**二、方案论证(设计理念)**

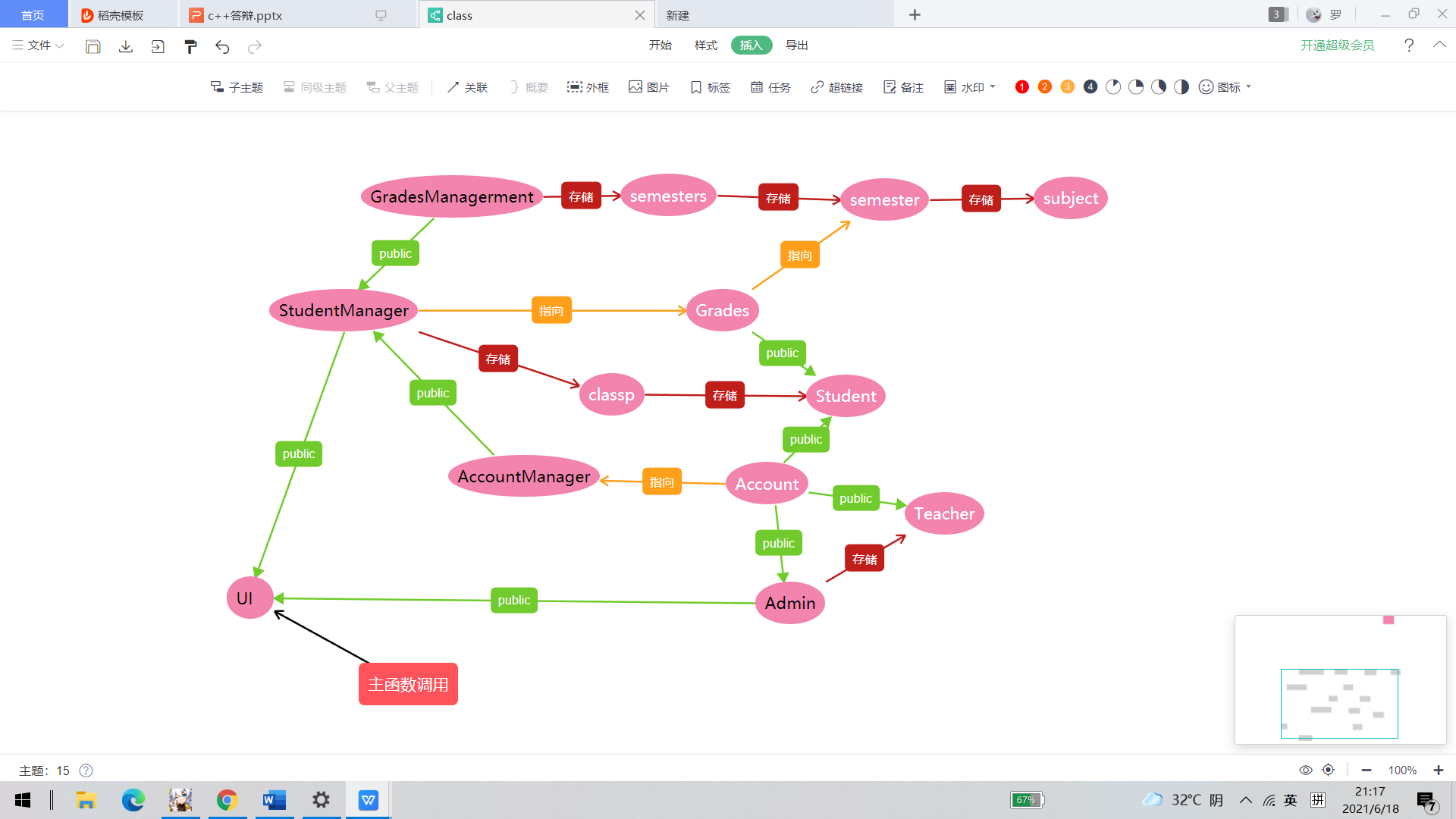
本程序内类的架构如图一：

图 1 程序内类的设计结构

图中粉底椭圆框为类的名字，箭头表示类之间的关系。其中绿色箭头代表类的派生关系，红色箭头表示类的存储关系，黄色箭头表示类的内部数据成员有指针指向另一个类。

可以通过不同类的功能，将类归纳为以下类型：

**数据类：**（图中以粉底白字表示，负责存储基本信息和提供修改函数接口）

Account:账号类，存储账号和密码，其中密码类型为ull\*，指向AccountManager类内的数据

Subject:学科类，拥有学科名字，所占学分，分数，绩点的数据

Semester:学期类，Map存储一个学生一个学期的所有科目，拥有计算GPA,加权平均分的功能。

Semesters:总学期类，拥有学期名的数据，Map存储所有学生在这学期的semester类。

Grades:成绩类，存储指向同一学生所有Semester类的指针。

Student:学生类，由Account，Grades类派生，拥有姓名，ID,电话号码，班级名等数据。

Classp:班级类，拥有班级名的数据，Map存储所有student类。

Teacher:教师类，由Account类pulibc派生，拥有姓名，ID,电话号码，班级名等数据。

**管理类：**（图中以粉底黑字表示，负责存储和管理数据类的数据，并提供函数接口给界面类）

GradesManagerment：成绩管理类，Map存储所有Semesters类，负责数据文件*grades.dat*的写入的读取

AccountManager:账号管理类，Map<string,ull>存储所有账号ID值和密码对应的字符串哈希值，负责数据文件*account.dat*的二进制写入和读取。

StudentManager:学生管理类，由AccountManager，GradesManagermen类public派生，Map存储所有classp类，负责数据文件*student.dat*的写入和读取。

Admin:管理员类，Map存储所有Teacher类，负责数据文件*teacher.dat*的写入和读取。

**界面类：**（图中以粉底黑字表示）

UI：界面类，由StudentManager,Admin类public派生，由主函数直接调用，负责对屏幕的cout输出，cin读取屏幕数据，并将指令下放到对应的管理类的数据接口中。

本程序使用了面对对象的方式来完成，运用到多文件，继承，多态等机制，优化了代码结构，并体验了不同层级之间的数据关系。比如为了实现不同班级，不同学期之间的管理，需要多个不同classp，semesters类，那么需要向上一层级的GradesManagerment，StudentManager类来存储这些类，这些类可以存储下一层级的Student，Semester类。通过类的层级设计来实验管理类和不同数据类之间的交互。

选择类内数据存储容器类型也进行一番对比。对于程序需要实现ID查找某一学生的功能，对于List容器需要逐个遍历每个班的学生，当数据量比较大时效率较低。对于Vector容器，由于程序部分类中使用了指针，Vector自身的特性当删除添加数据时，将会重新申请内存空间，导致其他类指针指向的地址失效。横向对比其他容器综合考虑后，本程序采用了STL Map容器来存储类内的数据成员，通过Map内部自建的红黑树和关键词查找，可以以更高的效率实现设计的功能，且写出来的更加直观易懂。

**三、过程论述**

由于使用了map容器存储数据，在遍历时只需拿到对应map的寄存器iterator，并通过iterator=map.begin();while(iterator!=map.end()) iterator++;即可实现遍历所有数据。删除数据时使用map.erase()函数。为保证类的封装性，大多数数据在类中为私有成员，因此都写了对应的函数接口。

下面细分各个系统的实现。

**登录系统： 头文件<LoginSystem.h> 数据文件：account.dat**

相关类：Account，AccountManager

Account是student,teacher,Admin的基类，存储账号的ID（char[20]），密码(unsigned longlong\*)数据。为保护密码的数据安全，密码以字符串哈希值的方式存储。

AccountManager用于管理所有账号数据，拥有map<string, ull>(key值为ID)的私有，每个Account类都将通过搜索ID方式将密码指针指向map内的数据。在写入*account.dat*二进制文件时，遍历map里的数据并二进制写入。

**班级系统：头文件<StudentSystem.h> 数据文件：student.dat**

相关类：Classp，StudentManager，student

Student:学生类，有ID，电话，名字等数据，并继承了grades，account类

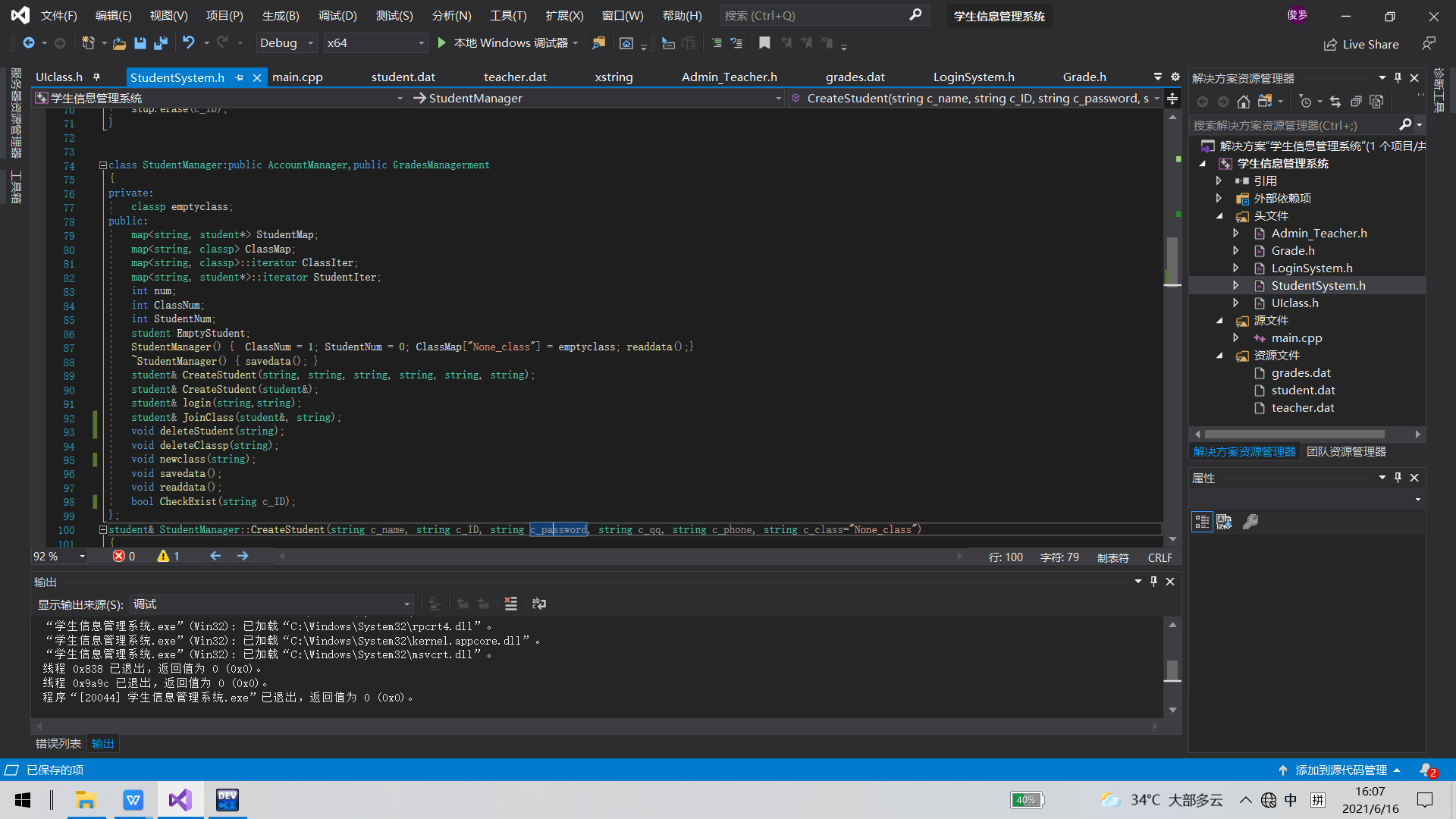
Classp类中拥有班级名(char[20])，用map<string,student>(key值为ID)存储该班所有student类。

StudentManagre类public继承了GradesManagerment，Accountmanagement。使用map<string, classp>存储所有的classp类，可以通过班级名找到该班的classp内。此外还拥有map<string, student\*>(key值为班级名)通过学号直接查找学生，可以省去先遍历班级的步骤提高效率。

新建和删除学生各类信息等都是UI类调用StudentManagement类中的函数，函数中根据需求把对应的新增、删除命令发送到GradesManagerment，AccountManager的管理类中。

存储数据时先遍历StudentManagement内所有班级Classp，再遍历该Classp所有学生，二重循环写入文件。

图二 student.dat数据



图三 StudentManager数据成员与函数

**成绩&学期系统：头文件<Grade.h> 数据文件：grades.dat**

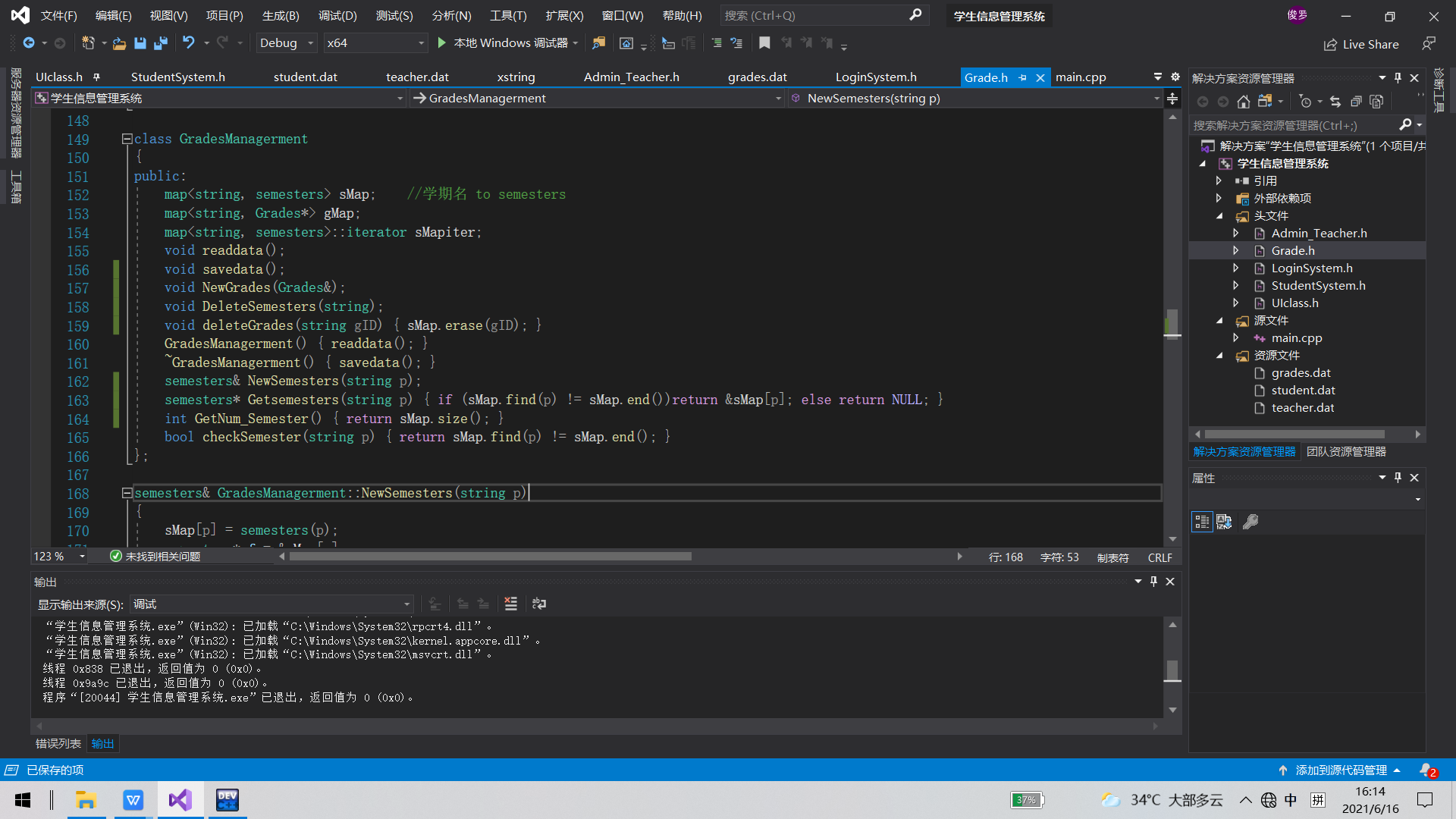
相关类：Subject，Semester，Semesters，Grades，GradesManagerment

Subject：科目名(string)，学分(double)，分数(double)，绩点(double)

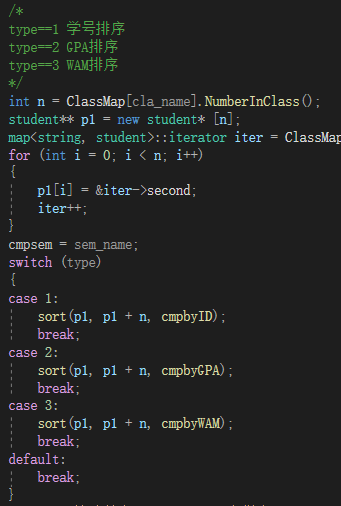
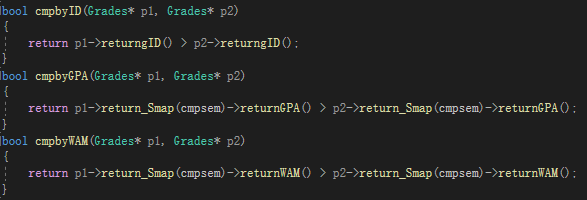
Semester：一个学生某一学期的数据。拥有数据成员学期名(string)，GPA(double)，总学分(double)，加权平均分(double)，map<string, subject>(key值为学科名)存储所有科目，有计算函数实时更新分数数据。

Semesters：某一学期所有学生的数据。有学期名(string), map<string, semester>(key值为学生ID)存储在该学期下所有学生这一学期的数据。

Grades：student的基类，成员有map<string, semester\*>(key值为学期名)，通过semester的指针指向该学生的所有学期，结构与Account类似。

GradesManagerment：管理类，通过map<string, semesters>(key值为学期名)以学期名字为key值存储所有学期数据。通过map<string, Grades\*>(key值为学生ID)指向所有学生成绩类。接收来自Admin，studentmanagerment关于修改学生分数、学期添加删除，新增科目的命令。

图四 GradesManagerment类的数据成员与函数

对于同班同学的成绩排序，使用了基类指针指向派生类对象的静态联编思想，运用了algorithm库里的快排函数sort，通过写基于Grades\*的比较器函数cmp，遍历classp内的学生并用指针数组（student\*）记录进行快排。

图五（左） 图六（上）

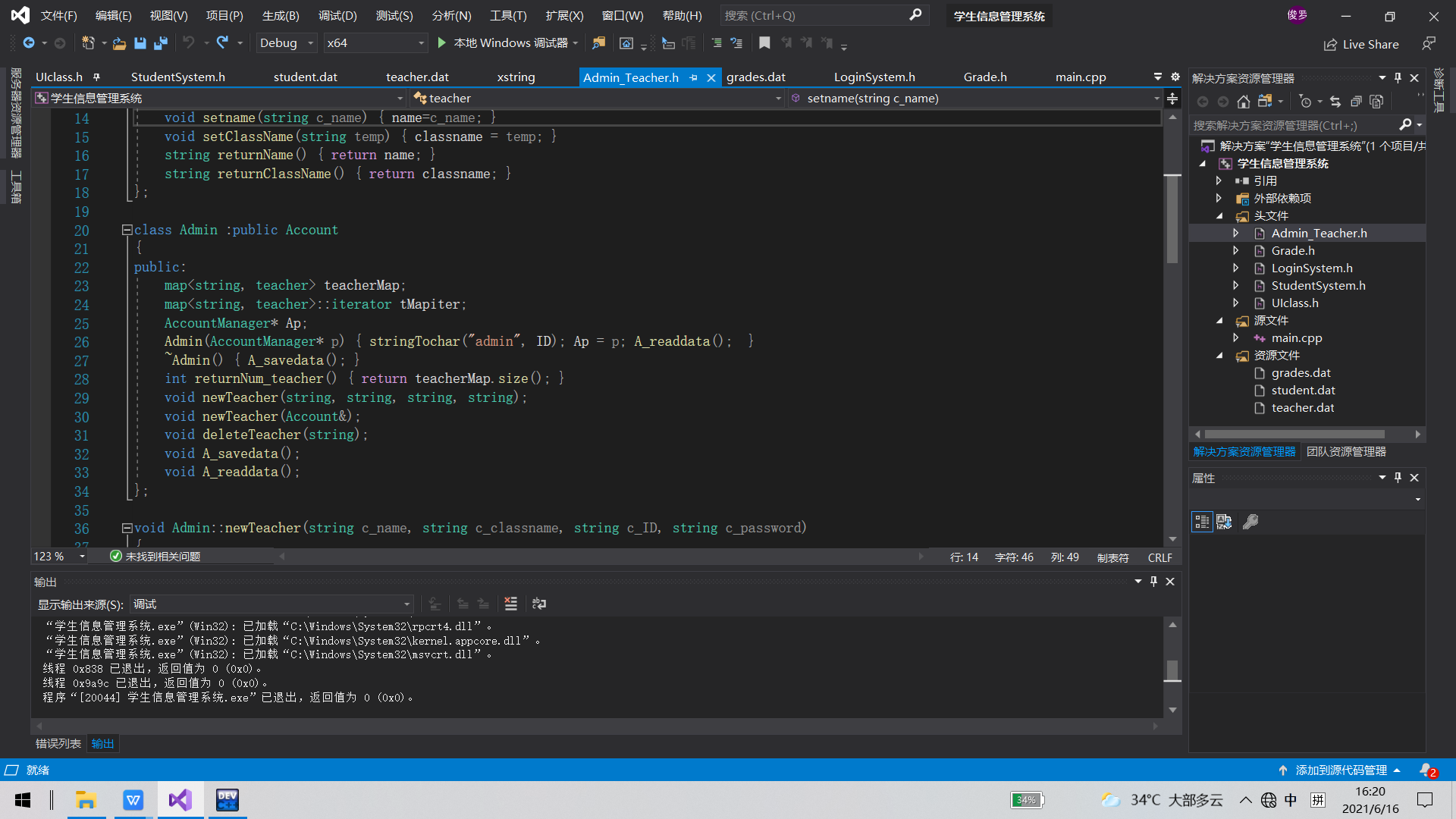
同班级的成绩排序代码图

存储数据时，先遍历GradesManagerment里所有学期，再遍历Semesters里所有学生的学期，最后遍历Semester里所有科目，使用了三重循环写入数据。

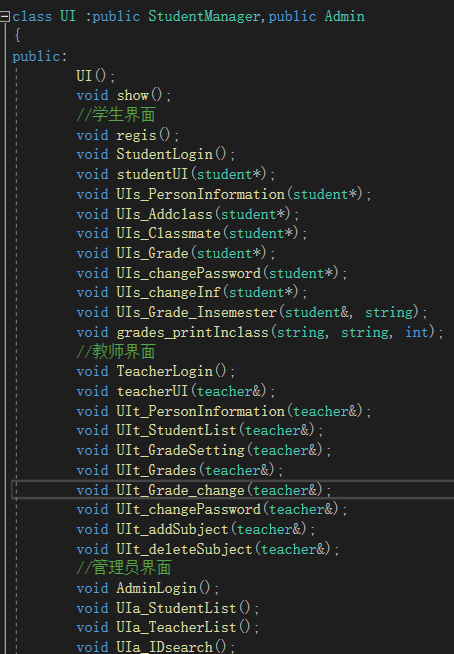
**管理员&教师系统：头文件<Admin\_Teacher.h> 数据文件：**

相关类：Teacher，Admin

Teacher：存储ID(string)名字(string)管理班级(string)信息。

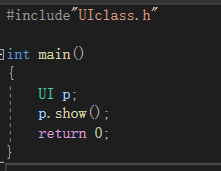
Admin：本身public继承了account，存储了管理员的账号和密码。Map<string，teacher>(key值为ID)存储所有教师信息，结构与StudentManagement类型。

图七 Admin类的数据成员与函数

**界面类：头文件<UIclass.h>**

相关类：UI

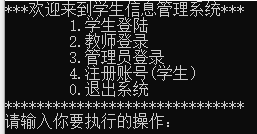
main函数的直接调用对象。实现屏幕的数据与各个管理类进行数据对接



图七 main函数代码图 图八 UI类的成员函数名单

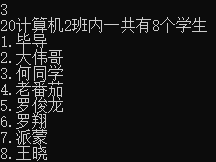
**四、结果分析**

最终程序实现了设计的全部功能。以下为部分主要功能的展示：



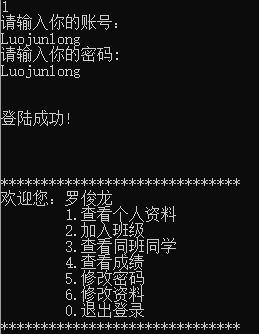
登录界面

**学生模块：**

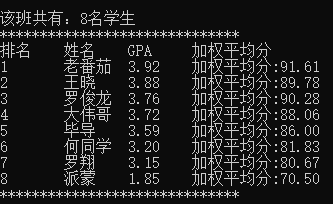
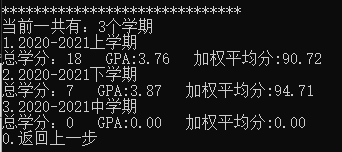


班级名单界面

查看个人信息界面



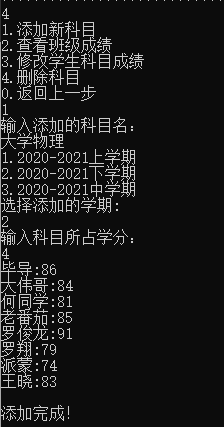
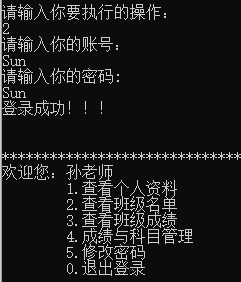
学生登录界面



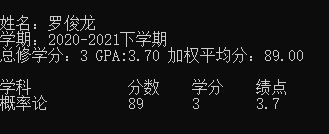
学期\_成绩界面

班级内成绩排序

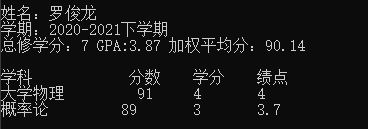
**教师模块：**



教师界面 成绩管理界面\_添加科目操作

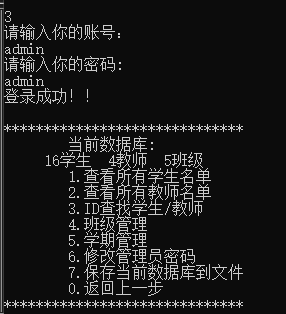


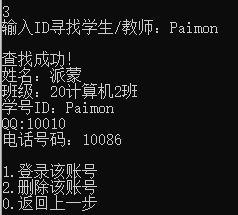
成绩界面变化（添加科目前）



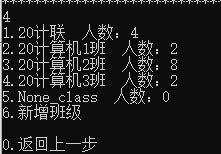
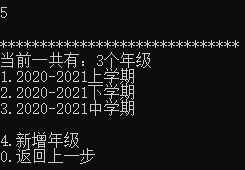
成绩界面变化（添加科目后）

**管理员模块：**



管理员登录界面 查看学生名单 查看教师名单

ID查找功能

学期管理 班级管理

**五、课程设计总结**

对于本次大作业的设计，最终圆满的完成了最初设计的全部功能，从最初的设计到成果的完成耗时近40小时，代码量也写到了1800行，算是一个不小的工程了，自己对结果还是非常满意的。

这次大作业的完成过程中还是有不少困难的，写代码中途也出现不少的阻碍。有时为了调整一个函数的实现需要反复debug几个小时。在我看来，本次设计过程中困难主要体验在类的设计和交互方面。因为本程序的层次比较多，涉及到不同班级，不同学期的管理，因此即使是已经提前设计好了不同类的结构和负责的功能，但在实际写到那部分时还是有很多设计时没有想到的，造成了不少问题。其中令我影响最深刻的是写到“学生加入班级”部分的功能时，设计目标要把这个student类从原classp类内的map中删除，再加入到新的classp类中，但实际写的时候，在删除操作后，有一些指向原来student的指针便会失效，还要找到对应的指针进行修改，这个问题因为最初设计时没有想到，最终花费了较长时间设计和修改。写到UI类时，也会发现很多功能管理类的函数没法实现，最后还要去修改已经本来写好的数据类，造成了很多麻烦。

从上述困难中我得到了教训，在正式敲代码前一定要设计好程序结构，在最初的设计阶段多去思考一些可能存在的问题，并用合理的方式解决它，即“谋定而后动”，不然最后就会出现反复修改的内耗。并且出现bug时多用调试功能，VS的调试功能还是很强大的，帮助我解决不少麻烦。

总之，通过这次大作业设计我从中学到了很多，也是我第一次体验到类的封装，面对对象编程等设计过程，加深了我对c++的理解，让我受益匪浅。

**参考文献**

［1］《C++程序设计基础（上）》——周霭如 林伟健 编著