

山西工程技术学院

《面向对象程序设计》课程设计说明书

学 生 姓 名 : _____

专 业 班 级 : _____

学 号 : _____

指 导 教 师 : _____

所属系（部） : 大数据系 _____

2023 年 12 月

山西工程技术学院

课程设计评阅书

题目：

学生成绩管理系统面向对象程序设计课程设计报告

大数据系 计算机科学与技术专业 姓名

设计时间：2023 年 11 月 20 日~2023 年 11 月 26 日

评阅意见：

成绩：

指导教师：_____（签字）

职 务：_____

2023 年____月____日

目 录

目 录.....	3
摘 要.....	4
第 1 章 开发背景.....	1
1.1 设计学生成绩管理系统的目的和意义	1
1.2 设计内容	1
第 2 章 设计学生成绩管理系统的开发环境.....	2
第 3 章 总体设计.....	3
第 4 章 学生成绩管理系统结构图和流程图.....	5
第 5 章 调试结果.....	6
5.1 管理系统界面	6
5.2 管理人员控制	6
5.3 操作	6
第 6 章 设计总结.....	7
附录：源程序代码.....	8

摘 要

随着时代发展，高等教育也逐步普及开来，越来越多学生步入高等教育的门槛，进入高等院校进行深造，据不完全统计 2023 年进入本科院校进行学习的学生约有 500 万人，如此之多的学生进入高等院校，必然需要有效的学生管理系统才能对学生的成绩进行有效且科学管理。

传统上，学生成绩管理采用手工方式，这种方式存在许多不足，例如：效率低下、容易出错、数据难以维护和更新等。因此，开发一个自动化的学生成绩管理系统成为一种迫切的需求。通过学生成绩管理系统，教师可以更加方便地查看学生的学习情况，及时发现学生的学习问题，并采取相应的措施进行辅导和帮助，从而提高教学质量。学生和家长可以通过学生成绩管理系统，及时查看学生的成绩和其他相关信息，方便快捷地了解学生的学习情况，更好地参与学生的学业管理，随着教育改革的深入，对于学生的各项信息管理也提出了更高的要求。学生成绩管理系统可以满足教育改革对于学生信息管理的需求，提供更加全面、准确的学生成绩信息，方便学校、老师、学生和家长的使使用。近年来，计算机技术和网络技术得到了飞速的发展，这为开发学生成绩管理系统提供了可能。通过使用计算机和网络技术，可以实现学生成绩的自动化管理和维护，提高管理效率和准确性。

我设计的学生信息管理系统，可以对学生的信息进行“插入，查找，修改，展示，排序，删除”的功能，可以方便快捷地对学生成绩进行有效的管理和操作，方便高校的相关负责老师的操作，且计算机对成绩进行操作避免了计算失误和误操作，同时也加强了学生的自我管理能力，提升了学校的信息化建设水平，自动化管理系统还可以节约大量人力资源。

关键词：C++；学生成绩管理系统；多功能；自动化管理；

第1章 开发背景

1.1 设计学生成绩管理系统的目的和意义

1.提高管理效率：传统的学生成绩管理方式通常需要人工记录和整理，不仅容易出错，而且效率低下。通过设计一个自动化的学生成绩管理系统，可以大大提高管理效率，减少错误率，并且可以节省大量的人力物力。

2.方便学生和教师查询：学生和教师可以随时通过学生成绩管理系统查询自己的成绩，无需到教务处或者教学科进行查询，方便快捷。

3.方便统计分析：学生成绩管理系统可以方便地进行统计分析，例如计算平均分、及格率、优秀率等指标，为教学评估和改进提供数据支持。

4.保证数据安全：学生成绩管理系统可以设置不同的权限，确保只有相关人员才能访问和修改成绩数据，保证数据的安全性和保密性。

5.加强学生自我管理：学生可以通过系统查询自己的成绩，了解自己的学习情况，加强自我管理。

6.节省人力物力：通过自动化的管理系统，可以减少学校在人工成绩管理方面的投入，节省人力物力。

7.促进学校信息化建设：学生成绩管理系统是学校信息化建设的重要组成部分，可以提高学校信息化水平

综上所述，设计学生成绩管理系统具有重要的意义和目的，可以提高工作效率、方便快捷的查询、确保数据准确性、统计分析功能、提高教学质量、加强学生自我管理、节省人力物力、促进学校信息化建设、保密性和安全性等。

1.2 设计内容

1.学生信息管理：包括学生基本信息（如姓名、性别、出生年月、等）的录入、修改、查询等功能。

2.成绩管理：包括学生各科成绩的录入、修改、查询等功能，以及成绩分析、统计等数据处理功能

3.设计成绩的录入与修改功能，包括单个成绩的录入与修改和批量成绩的录入与修改。

第 2 章 设计学生成绩管理系统的开发环境

Visual Studio Code（简称“VS Code”）是 Microsoft 在 2015 年 4 月 30 日 Build 开发者大会上正式宣布一个运行于 Mac OS X、Windows 和 Linux 之上的，针对于编写现代 Web 和云应用的跨平台源代码编辑器，可在桌面上运行，并且可用于 Windows，macOS 和 Linux。它具有对 JavaScript，TypeScript 和 Node.js 的内置支持，并具有丰富的其他语言（例如 C++，C#，Java，Python，PHP，Go）和运行时（例如 .NET 和 Unity）扩展的生态系统。

这标志着微软公司第一次向开发者们提供了一款真正的跨平台编辑器。虽然完整版的 Visual Studio 仍然是只能运行在 Windows 和 macOS（Mac OS X）之上，但是这一次的声明展示了微软公司对于支持其他计算机平台的承诺

该编辑器集成了所有一款现代编辑器所应该具备的特性，包括语法高亮（syntax highlighting），可定制的热键绑定（customizable keyboard bindings），括号匹配（bracket matching）以及代码片段收集（snippets）。Somasegar 也告诉笔者这款编辑器也拥有对 Git 的开箱即用的支持。Microsoft Docs（微软文档）提供了相应的学习教程帮助用户在 Visual Studio Code 中登陆 GitHub。

Visual Studio Code 提供了丰富的快捷键。用户可通过快捷键 [Ctrl] + [K] + [S]（按住 Ctrl 键不放，再按字母 K 键和 S 键）调出快捷键面板，查看全部的快捷键定义。也可在面板中双击任一快捷键，为某项功能指定新的快捷键。一些预定义的常用快捷键包括：格式化文档（整理当前视图中的全部代码），[Shift] + [Alt] + [F]；格式化选定内容（整理当前视图被选定部分代码），[Ctrl] + [K] + [F]；放大视图，[Ctrl] + [Shift] + [=]；缩小视图，[Ctrl] + [Shift] + [-]；打开新的外部终端（打开新的命令行提示符）：[Ctrl] + [Shift] + [C]。

该编辑器支持多种语言和文件格式的编写，截止 2019 年 9 月，已经支持了如下 37 种语言或文件：F#、HandleBars、Markdown、Python、Java、PHP、Haxe、Ruby、Sass、Rust、PowerShell、Groovy、R、Makefile、HTML、JSON、TypeScript、Batch、Visual Basic、Swift、Less、SQL、XML、Lua、Go、C++、Ini、Razor、Clojure、C#、Objective-C、CSS、JavaScript、Perl、Coffee Script、Dockerfile、Dart。

第3章 总体设计

根据功能需要，程序分为简单的图形显示，管理员控制和后台操作三个模块，以下是每个模块的需求。

简单图形显示：程序开始运行时，会有简单的显示界面引导，管理员和学生需要根据指引进行操作，输入相应的数字即可开始相关的操作（“输入学生成绩”“显示统计成绩”“查找学生成绩”“修改学生成绩”“删除学生成绩”“插入学生成绩”“按平均分排序”“显示出学生成绩”“退出管理系统”），操作进行时，也会有相应的文字提醒出现在屏幕上，操作人员需要根据提示正确输入相对应的字符。

管理员操作：根据图形显示的结果，管理员根据成绩单等文件正确输入相关的字符，最后确认无误后退出。

后台操作：后台中定义了每个学生的类，其中包含学号（num）姓名（name[8]）班级（class_0[20]）电子技术成绩（elec）C语言成绩（C_program）英语成绩（english）数学成绩（math）多媒体成绩（media）体育成绩（sport）政治（polity）平均分（average）

其中还定义了友元函数（friend）输入 Input（Student stu[]）统计数据 Statistic(Student stu[])查找 Lookup（Student stu[]）修改成绩 Modify(Students stu[])删除成绩 Delete（Student stu[]）输出 Output（Student stu[]）增添 Insert（Student stu[]）排序 Sort(Student stu[])写 write(Student stu[],int n)读 Read（Student stu[]）

在排序函数中使用了冒泡排序法（图 3-1）

```
for (i = 0; i < n-1; i++) {  
    for (j = 0; j < n - 1; j++) {  
        if (stu[j].average < stu[j + 1].average) {}  
    }  
}
```

图 3-1 排序函数中的冒泡排序

当输入排序函数时，会调用冒泡排序法对“名字”“学号”“平均分”“各科成绩”进行一个排名，冒泡排序法的空间复杂度小，平均时间复杂度为 $O(n^2)$ ，而且非常稳定，故可以采取冒泡排序法。

在友元函数中使用了“fstream”“myFile.open”“myFile.close”等文件操作函数，可以方便对文件进行操作。

在各个函数中，也写入了检查程序，用来检查输入的数字和字符的合法性和合理性，避免了报错和逻辑错误，保证了数据的合法性和安全性。

```
if (stu[i].num == stu[i - c].num) {  
    cout << "您输入的学号已存在！请重新输入。" << endl;  
    goto loop;  
}
```

图 3-2 判断是否有重重复数据是否合法

第 4 章 学生成绩管理系统结构图和流程图

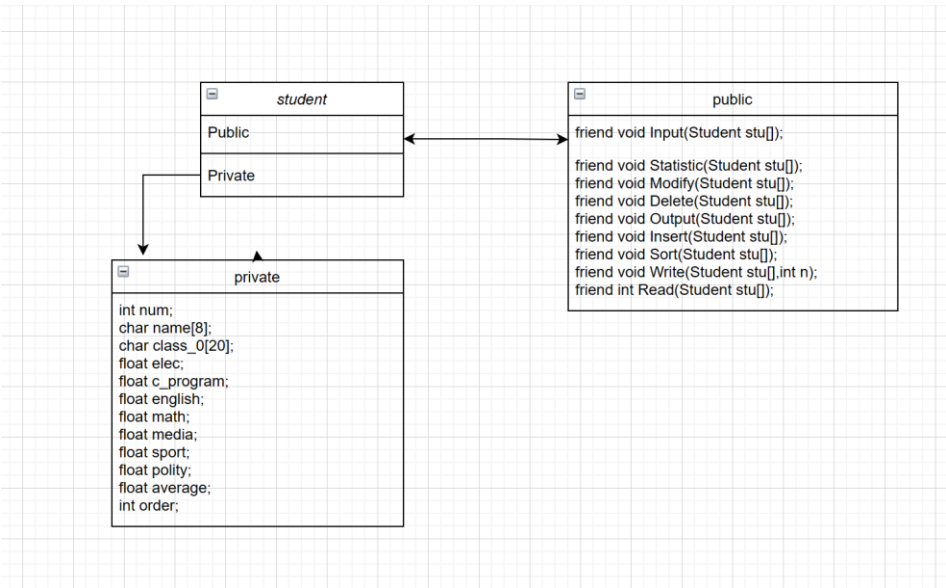


图 4-1 类的定义

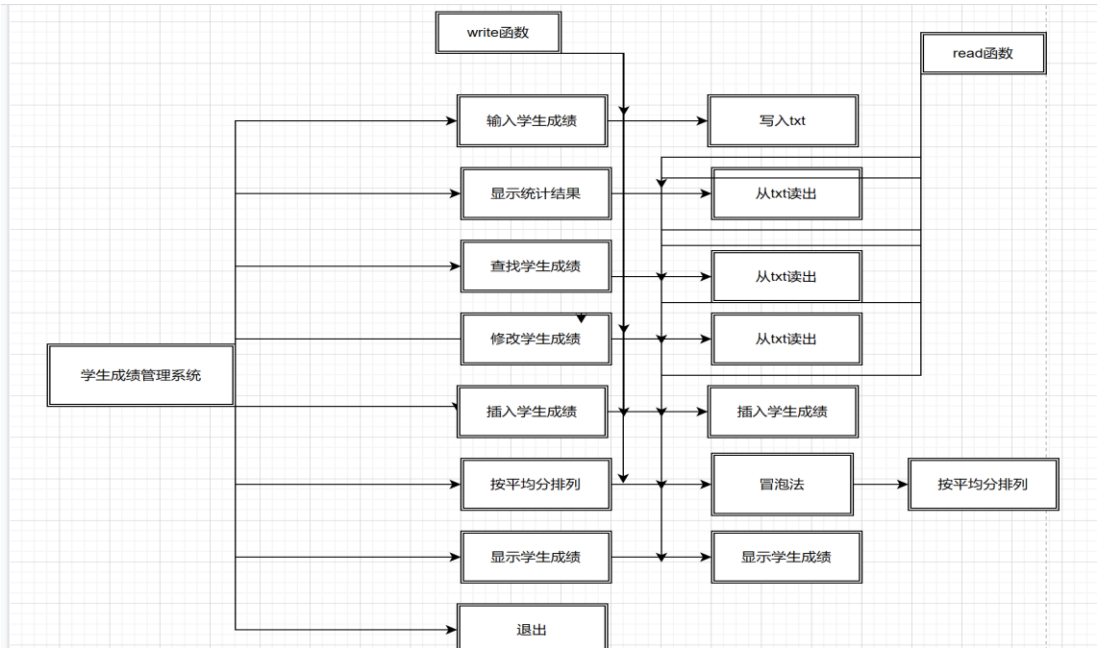


图 4-2 流程图

第5章 调试结果

5.1 管理系统界面

```
*****
-----欢迎使用学生成绩管理系统-----
*      【1】 输入学生成绩      *
*      【2】 显示统计数据      *
*      【3】 查找学生成绩      *
*      【4】 修改学生成绩      *
*      【5】 删除学生成绩      *
*      【6】 插入学生成绩      *
*      【7】 按平均分排列      *
*      【8】 显示学生成绩      *
*      【0】 退出管理系统      *
*****
请选择您的操作（0-8）：
|
```

图 5-1 学生成绩管理系统引导界面

5.2 管理人员控制

根据提示输入数字（1.2.3.4.5.6.7.8.0）来进行操作（输入学生成绩，显示统计数据，查找学生成绩，修改学生成绩，删除学生成绩，插入学生成绩，按平均分排列，显示学生成绩，退出管理系统）

5.3 操作

根据提示进行输入后就会调取相应的友元函数进行输入/显示统计数据/查找学生成绩/修改学生成绩/删除学生成绩/插入学生成绩/按平均分排列，显示学生成绩等的计算和输出。

也调取了 score.txt 文件进行保存。

第6章 设计总结

我设计的学生成绩管理系统，可以对学生的成绩进行“输入，显示统计，查找成绩，修改学生成绩，删除成绩，的功能，可以方便快捷地对学生成绩进行有效的管理和操作，方便高校的相关负责老师的操作，且计算机对成绩进行操作避免了计算失误和误操作，同时也加强了学生的自我管理能力，提升了学校的信息化建设水平，自动化管理系统还可以节约大量人力资源。

通过这次的面向对象程序设计的课程设计我学会了使用封装和类，在定义类中使用了 `private` 中放置了学生的“学号”“姓名”“班级”“其他课程的成绩”在使用中可以进行调用，在 `public` 部分放置了相关操作的友元函数，以便在后续调用中使用。

传统上，学生成绩管理采用手工方式，这种方式存在许多不足，例如：效率低下、容易出错、数据难以维护和更新等。因此，开发一个自动化的学生成绩管理系统成为一种迫切的需求。通过学生成绩管理系统，教师可以更加方便地查看学生的学习情况，及时发现学生的学习问题，并采取相应的措施进行辅导和帮助，从而提高教学质量。近年来，计算机技术和网络技术得到了飞速的发展，这为开发学生成绩管理系统提供了可能。通过使用计算机和网络技术，可以实现学生成绩的自动化管理和维护，提高管理效率和准确性。

在友元函数中使用了“`fstream`” “`myFile.open`”“ `myFile.close` ”等文件操作，在排列函数中使用了冒泡法进行排序，对元素进行了排序后然后输出。

在函数中也进行了违规和错误信息判别，对数据合法性的判别是很重要的，输入违法字符可能会导致程序报错以及输入信息出错，在程序中必须确保输入的数据和字符的合法性

在这次的课程设计中我对面向对象程序设计有了深刻的体会和认识，学会了类以及相关的友元函数，以及文件操函数的使用和定义，学会了如何使用更加方便的函数和操作。

附录：源程序代码

```
#if defined(_MSC_VER) && !defined(_CRT_SECURE_NO_DEPRECATED)
#define _CRT_SECURE_NO_DEPRECATED
#endif
#include <bits/stdc++.h>
using namespace std;
class Student{
public:
    friend void Input(Student stu[]);
    friend void Statistic(Student stu[]);
    friend void Lookup(Student stu[]);
    friend void Modify(Student stu[]);
    friend void Delete(Student stu[]);
    friend void Output(Student stu[]);
    friend void Insert(Student stu[]);
    friend void Sort(Student stu[]);
    friend void Write(Student stu[],int n);
    friend int Read(Student stu[]);
private:
    int num;
    char name[8];
    char class_0[20];
    float elec;
    float c_program;
    float english;
    float math;
    float media;
    float sport;
    float polity;
    float average;
    int order;
} stu[100];

void Write(Student stu[], int n) {
    fstream myFile;
    myFile.open("score.txt", ios::out | ios::binary);
    if (!myFile) {
        cout << "score.txt can't open!" << endl;
        abort();
    }
    int count = n;
    myFile << count << endl<<endl;
```

```

    for (int i = 0; i < count; i++) {
        myFile << stu[i].class_0 << "\t"
            << stu[i].num << "\t"
            << stu[i].name << "\t"
            << stu[i].elec << "\t"
            << stu[i].c_program<< "\t"
            << stu[i].media << "\t"
            << stu[i].english << "\t"
            << stu[i].math << "\t"
            << stu[i].sport << "\t"
            << stu[i].polity << "\t"
            << stu[i].average << endl;
    }
    myFile.close();
}

int Read(Student stu[]) {
    fstream myFile;
    myFile.open("score.txt", ios::in | ios::binary);
    if (!myFile) {
        cout << "score.txt can't open!" << endl;
        abort();
    }
    int count;
    myFile.seekg(0);
    myFile >> count;
    for (int i = 0; i <= count; i++) {
        myFile >> stu[i].class_0 >> stu[i].num >> stu[i].name
            >> stu[i].elec >> stu[i].c_program >> stu[i].media
            >> stu[i].english >> stu[i].math >> stu[i].sport
            >> stu[i].polity >> stu[i].average;
    }
    myFile.close();
    return count;
}

void Input(Student stu[]) {
    system("cls");
    int i = 0;
    int flag;
    char sign = '0';
    cout << endl<<"====>> 请输入学生成绩 <<====<<endl;
    while (sign != 'n' && sign != 'N') {
        cout << "班级: ";

```

```

    cin >> stu[i].class_0;
loop:
    cout << "学号: ";
    cin >> stu[i].num;
    int c = 0;
    while (c < i) {
        c++;
        if (stu[i].num == stu[i - c].num) {
            cout << "您输入的学号已存在! 请重新输入." << endl;
            goto loop;
        }
    }
    cout << "姓名: ";
    cin >> stu[i].name;
    do {
        flag = 0;
        cout << "电子技术成绩: ";
        cin >> stu[i].elec;
        if (stu[i].elec > 100 || stu[i].elec < 1) {
            cout << " 对不起, 请输入 1-100 之间的数字!! \n";
        }
        else {
            flag = 1;
        }
    } while (flag == 0);
    do {
        flag = 0;
        cout << "C++程序设计成绩: ";
        cin >> stu[i].c_program;
        if (stu[i].c_program > 100 || stu[i].c_program < 1) {
            cout << " 对不起, 请输入 1-100 之间的数字!! \n";
        }
        else {
            flag = 1;
        }
    } while (flag == 0);
    do {
        flag = 0;
        cout << "多媒体技术成绩: ";
        cin >> stu[i].media;
        if (stu[i].media > 100 || stu[i].media < 1) {
            cout << " 对不起, 请输入 1-100 之间的数字!! \n";
        }
        else {

```

```

        flag = 1;
    }
} while (flag == 0);
do {
    flag = 0;
    cout << "大学英语成绩: ";
    cin >> stu[i].english;
    if (stu[i].english > 100 || stu[i].english < 1) {
        cout << " 对不起, 请输入 1-100 之间的数字!! \n";
    }
    else {
        flag = 1;
    }
} while (flag == 0);
do {
    flag = 0;
    cout << "高等数学成绩: ";
    cin >> stu[i].math;
    if (stu[i].math > 100 || stu[i].math < 1) {
        cout << " 对不起, 请输入 1-100 之间的数字!! \n";
    }
    else {
        flag = 1;
    }
} while (flag == 0);
do {
    flag = 0;
    cout << "大学体育成绩: ";
    cin >> stu[i].sport;
    if (stu[i].sport > 100 || stu[i].sport < 1) {
        cout << " 对不起, 请输入 1-100 之间的数字!! \n";
    }
    else {
        flag = 1;
    }
} while (flag == 0);
do {
    flag = 0;
    cout << "马克思主义基本原理成绩: ";
    cin >> stu[i].polity;
    if (stu[i].polity > 100 || stu[i].polity < 1) {
        cout << " 对不起, 请输入 1-100 之间的数字!! \n";
    }
    else {

```

```

        flag = 1;
    }
} while (flag == 0);
stu[i].average = (stu[i].elec + stu[i].c_program + stu[i].media + stu[i].english + stu[i].math +
    stu[i].sport + stu[i].polity) / 7;
cout << " 平均分为: " << stu[i].average<<endl;
cout << "====>>    提示: 是否继续写入学生成绩 ?(y/n)";
cin >> sign;
i++;
}
Write(stu, i);
}

```

```

void Statistic(Student stu[]) {
    system("cls");
    int n = Read(stu);
    cout << endl << "====>>    输出学生统计数据    <<=====\n" << endl;
    cout << "-----" << endl;
    cout << "班级" << "\t" << "学号" << "\t" << "姓名" << "\t" << "平均分" << endl;
    cout << "-----" << endl;
    for (int i = 0; i < n; i++)
        cout << stu[i].class_0 << "\t" << stu[i].num << "\t" << stu[i].name << "\t" << stu[i].average <<
endl;
    cout << "-----" << endl;
    system("pause");
}

```

```

void Lookup(Student stu[]) {
    system("cls");
    int n = Read(stu);
    int s;
    int i = 0;
    cout << endl << "====>>    查找学生成绩    <<=====" << endl;
    cout << "请输入要查找学生的学号: ";
    cin >> s;
    while ((stu[i].num - s) != 0 && i < n)i++;
    if (i == n) {
        cout << "====>>    对不起, 无法找到该学生.....    <<=====" << endl;
    }
    else {
        cout << "-----" << endl;
        cout << "班级: " << stu[i].class_0 << endl;
        cout << "学号: " << stu[i].num << endl;
        cout << "姓名: " << stu[i].name << endl;
    }
}

```



```

        cout << "电子技术: " << stu[i].elec << endl;
        cout << "C++程序设计: " << stu[i].c_program << endl;
        cout << "多媒体技术: " << stu[i].media << endl;
        cout << "大学英语: " << stu[i].english << endl;
        cout << "高等数学: " << stu[i].math << endl;
        cout << "大学体育: " << stu[i].sport << endl;
        cout << "马克思主义基本原理: " << stu[i].polity << endl;
        cout << "平均分: " << stu[i].average << endl;
    }
}

void Modify(Student stu[]) {
    system("cls");
    int n = Read(stu);
    int s;
    int i = 0;
    cout << endl << "===== 修改学生成绩 <<=====" << endl;
    cout << "请输入要修改成绩学生的学号: ";
    cin >> s;
    while ((stu[i].num - s) != 0 && i < n)i++;
    if (i == n) {
        cout << "===== 对不起, 无法找到该学生..... <<=====" << endl;
    }
    else {
        cout << "-----" << endl;
        cout << "班级" << "\t" << "学号" << "\t" << "姓名" << "\t"
            << "电子" << "\t" << "C++" << "\t" << "多媒体" << "\t"
            << "英语" << "\t" << "数学" << "\t" << "体育" << "\t"
            << "政治" << "\t" << "平均分" << endl;
        cout << "-----" << endl;
        cout << stu[i].class_0 << "\t" << stu[i].num << "\t" << stu[i].name << "\t"
            << stu[i].elec << "\t" << stu[i].c_program << "\t" << stu[i].media << "\t"
            << stu[i].english << "\t" << stu[i].math << "\t" << stu[i].sport << "\t"
            << stu[i].polity << "\t" << stu[i].average << endl;
        cout << endl << "请重新输入该学生成绩: " << endl;
        cout << "电子技术成绩: ";
        cin >> stu[i].elec;
        cout << "C++成绩: ";
        cin >> stu[i].c_program;
        cout << "多媒体技术成绩: ";
        cin >> stu[i].media;
        cout << "大学英语成绩: ";
        cin >> stu[i].english;
        cout << "高等数学成绩: ";
    }
}

```

```

    cin >> stu[i].math;
    cout << "大学体育成绩: ";
    cin >> stu[i].sport;
    cout << "马克思主义基本原理成绩: ";
    cin >> stu[i].polity;
    stu[i].average = (stu[i].elec + stu[i].c_program + stu[i].media +
        stu[i].english + stu[i].math + stu[i].sport + stu[i].polity) / 7;
    cout << "平均分: " << stu[i].average << endl;

    char c;
    cout << "====>>    是否保存数据 ? (y/n)";
    cin >> c;
    if (c != 'n' && c != 'N')
        Write(stu, n);
}
}

void Delete(Student stu[]) {
    system("cls");
    int n = Read(stu);
    int s;
    int i = 0, j;
    cout << endl << "====>>    删除学生成绩    <<====" << endl;
    cout << "请输入要删除的学生的学号: ";
    cin >> s;
    while ((stu[i].num - s) != 0 && i < n) i++;
    if (i == n) {
        cout << "====>>    对不起, 无法找到该学生.....    <<====" << endl;
    }
    else {
        for (j = i; j < n - 1; j++) {
            strcpy(stu[j].class_0, stu[j + 1].class_0);
            stu[j].num = stu[j + 1].num;
            strcpy(stu[j].name, stu[j + 1].name);
            stu[j].elec = stu[j + 1].elec;
            stu[j].c_program = stu[j + 1].c_program;
            stu[j].media = stu[j + 1].media;
            stu[j].english = stu[j + 1].english;
            stu[j].math = stu[j + 1].math;
            stu[j].sport = stu[j + 1].sport;
            stu[j].polity = stu[j + 1].polity;
            stu[j].average = stu[j + 1].average;
        }
        cout << "====>>    提示: 已成功删除! " << endl;
    }
}

```

```

    }
    Write(stu, n - 1);
}

void Insert(Student stu[]) {
    system("cls");
    int n = Read(stu);
    char s='0';
    cout << endl << "===== >>      增加学生成绩      <<===== " << endl;
    while (s != 'n' && s != 'N') {
        cout << "班级: ";
        cin >> stu[n].class_0;
        cout << "学号: ";
        cin >> stu[n].num;
        cout << "姓名: ";
        cin >> stu[n].name;
        cout << "电子技术成绩: ";
        cin >> stu[n].elec;
        cout << "C++成绩: ";
        cin >> stu[n].c_program;
        cout << "多媒体技术成绩: ";
        cin >> stu[n].media;
        cout << "大学英语成绩: ";
        cin >> stu[n].english;
        cout << "高等数学成绩: ";
        cin >> stu[n].math;
        cout << "大学体育成绩: ";
        cin >> stu[n].sport;
        cout << "马克思主义基本原理成绩: ";
        cin >> stu[n].polity;
        stu[n].average = (stu[n].elec + stu[n].c_program + stu[n].media +
            stu[n].english + stu[n].math + stu[n].sport + stu[n].polity) / 7;
        cout << "平均分: " << stu[n].average << endl;
        n++;
        cout << "===== >>      是否继续插入(y/n)";
        cin >> s;
    }
    Write(stu, n);
}

void Sort(Student stu[]) {
    system("cls");
    int i, j, k;
    float s;

```

```

char t[20];
cout << endl << "===== >>    降序排列    <<===== " << endl;
int n = Read(stu);
for (i = 0; i < n-1; i++) {
    for (j = 0; j < n - 1; j++) {
        if (stu[j].average < stu[j + 1].average) {

            strcpy(t, stu[j + 1].class_0);
            strcpy(stu[j + 1].class_0, stu[j].class_0);
            strcpy(stu[j].class_0, t);
            //num
            k = stu[j + 1].num;
            stu[j + 1].num = stu[j].num;
            stu[j].num = k;

            //name
            strcpy(t, stu[j + 1].name);
            strcpy(stu[j + 1].name, stu[j].name);
            strcpy(stu[j].name, t);
            //elec
            s = stu[j + 1].elec;
            stu[j + 1].elec = stu[j].elec;
            stu[j].elec = s;
            //c_program
            s = stu[j + 1].c_program;
            stu[j + 1].c_program = stu[j].c_program;
            stu[j].c_program = s;
            //media
            s = stu[j + 1].media;
            stu[j + 1].media = stu[j].media;
            stu[j].media = s;
            //english
            s = stu[j + 1].english;
            stu[j + 1].english = stu[j].english;
            stu[j].english = s;
            //math
            s = stu[j + 1].math;
            stu[j + 1].math = stu[j].math;
            stu[j].math = s;
            //sport
            s = stu[j + 1].sport;
            stu[j + 1].sport = stu[j].sport;
            stu[j].sport = s;
            //polity

```

```

        s = stu[j + 1].polity;
        stu[j + 1].polity = stu[j].polity;
        stu[j].polity = s;
        //average
        s = stu[j + 1].average;
        stu[j + 1].average = stu[j].average;
        stu[j].average = s;
    }
}
}
cout << "-----" << endl;
cout << "班级" << "\t" << "学号" << "\t" << "姓名" << "\t"
    << "电子" << "\t" << "C++" << "\t" << "多媒体" << "\t"
    << "英语" << "\t" << "数学" << "\t" << "体育" << "\t"
    << "政治" << "\t" << "平均分" << endl;
cout << "-----" << endl;
for (int i = 0; i < n; i++) {
    stu[i].order = i + 1;
    cout << stu[i].class_0 << "\t" << stu[i].num        << "\t" << stu[i].name << "\t"
        << stu[i].elec    << "\t" << stu[i].c_program << "\t" << stu[i].media << "\t"
        << stu[i].english << "\t" << stu[i].math        << "\t" << stu[i].sport << "\t"
        << stu[i].polity  << "\t" << stu[i].average    << endl;
}
Write(stu, n);
}

void Output(Student stu[]) {
    system("cls");
    int n = Read(stu);
    cout << endl << "=====>>    显示全部学生成绩    <<=====" << endl;
    if (!stu) {
        cout << "没有记录";
    }
    else {
        cout << "-----" << endl;
        cout << "班级" << "\t" << "学号"  << "\t" << "姓名"  << "\t"
            << "电子" << "\t" << "C++"  << "\t" << "多媒体" << "\t"
            << "英语" << "\t" << "数学"  << "\t" << "体育"   << "\t"
            << "政治" << "\t" << "平均分" << endl;
        cout << "-----" << endl;
        for (int i = 0; i < n; i++) {
            cout << stu[i].class_0 << "\t" << stu[i].num        << "\t" << stu[i].name  << "\t"
                << stu[i].elec    << "\t" << stu[i].c_program << "\t" << stu[i].media << "\t"
                << stu[i].english << "\t" << stu[i].math        << "\t" << stu[i].sport << "\t"

```

```

                << stu[i].polity << "\t" << stu[i].average << endl;
            }
            cout << "-----" << endl;
        }
    }
}

```

```

int menu() {
    char c;
    do {
        system("cls");
        cout << "*****" << endl;
        cout << "-----欢迎使用学生成绩管理系统-----" << endl;
        cout << "      *           【1】 输入学生成绩           *      " << endl;
        cout << "      *           【2】 显示统计数据           *      " << endl;
        cout << "      *           【3】 查找学生成绩           *      " << endl;
        cout << "      *           【4】 修改学生成绩           *      " << endl;
        cout << "      *           【5】 删除学生成绩           *      " << endl;
        cout << "      *           【6】 插入学生成绩           *      " << endl;
        cout << "      *           【7】 按平均分排列           *      " << endl;
        cout << "      *           【8】 显示学生成绩           *      " << endl;
        cout << "      *           【0】 退出管理系统           *      " << endl;
        cout << "*****" << endl;
        cout << "请选择您的操作 (0-8):" << endl;
        c = getchar();
    } while (c < '0' || c > '8');
    return (c - '0');
}

```

```

int main() {
    for (;;) {
        switch (menu()) {
            case 1:
                Input(stu);
                break;
            case 2:
                Statistic(stu);
                break;
            case 3:
                Lookup(stu);
                system("pause");
                break;
            case 4:
                Modify(stu);
                system("pause");

```

```

        break;
    case 5:
        Delete(stu);
        system("pause");
        break;
    case 6:
        Insert(stu);
        system("pause");
        break;
    case 7:
        Sort(stu);
        system("pause");
        break;
    case 8:
        Output(stu);
        system("pause");
        break;
    case 0:
        cout << endl << "===== 感谢您使用学生成绩管理系统
=====\\n" << endl;
        exit(0);
    }
}
return 0;
}

```