

计算机高级语言程序设计 (C++)

课程设计报告

班级：191181

学号：20181001095

姓名：常文瀚

日期：2019 年 1 月 17 日

报告正文部分

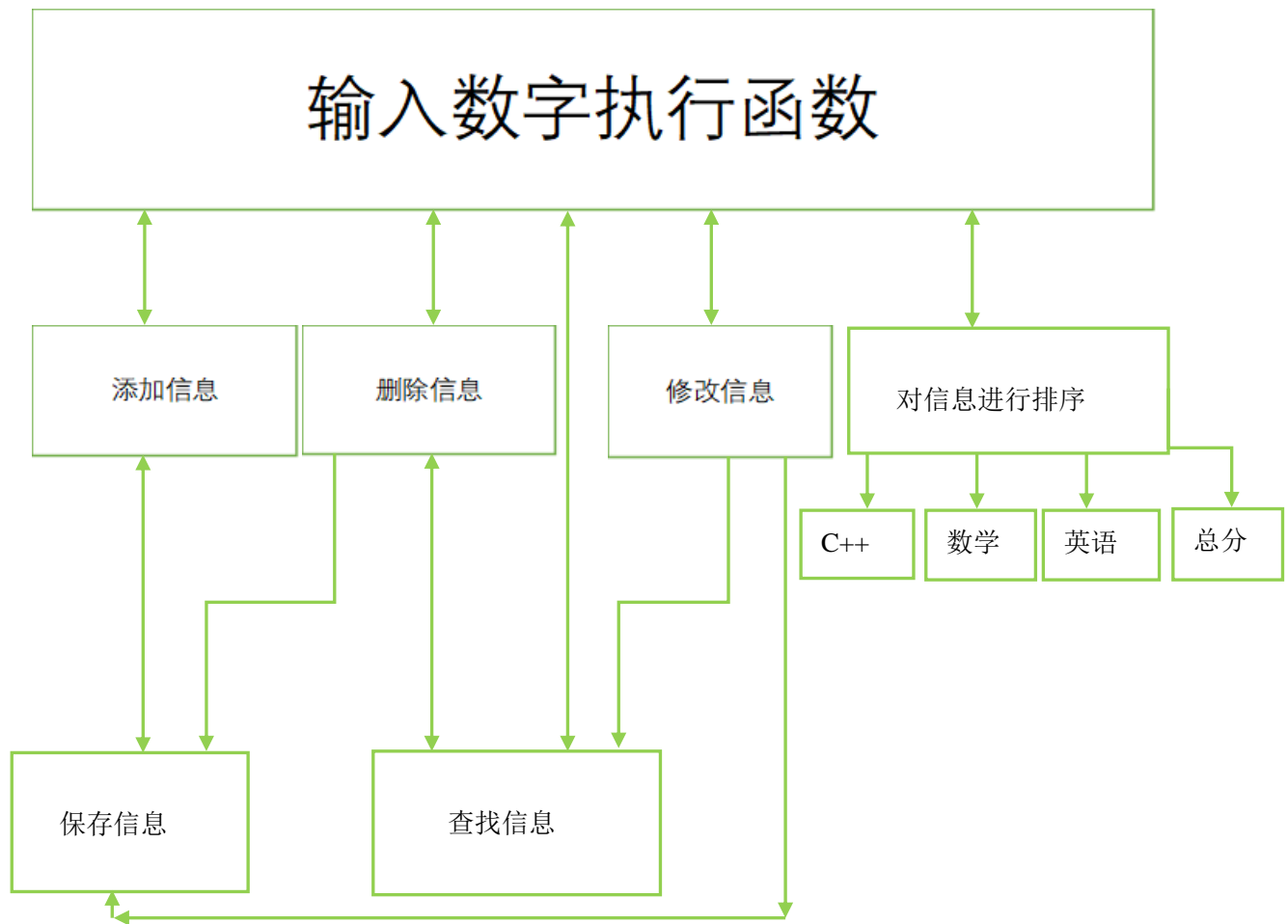
一 课程设计题目与要求

1. 课程设计题目：《学生成绩信息管理系统》
2. 课程设计内容
 - 1) 学生信息的录入：录入一名新学生的信息；
 - 2) 学生信息的添加：添加一名新学生的信息；
 - 3) 学生信息的显示：现实所添加的学生的成绩信息；
 - 4) 学生信息的读取：从文件中读取录入的学生信息；
 - 5) 学生信息的查找：查找所录入学生的成绩信息；
 - 6) 学生信息的删除：删除所录入的学生信息；
 - 7) 学生信息的修改：修改录入错误的学生信息；
 - 8) 学生信息的排序：按照使用者所希望的项目对已录入的信息进行排序；
3. 功能描述
 - 1) 信息的添加
该系统将提示用户输入将要添加的学生的信息，并录入输入的信息；
 - 2) 信息的删除
该系统将提示用户选择学号或姓名查找将要删除的学生的信息，然后调用删除函数，删除学生的信息；
 - 3) 信息的查找
该系统将提示用户选择学号或姓名查找其所要查询的学生信息，并调用查找函数对信息进行查找；
 - 4) 信息的修改
该系统将提示用户选择学号或姓名查找其所要查询的学生信息，并在用户选择并输入后提示用户重新输入该学生的成绩信息；
4. 要求
用 C++ 面向对象的方式实现，类名和成员函数名需要使用自己的名字缩写为前缀。

二 需求分析

学生成绩信息管理系统是学校教务管理的的重要组成部分，学生信息管理系统使教务对学生信息的统计更加方便快捷，信息的统计涵盖了学生的姓名、学号等一系列学生各门科目的成绩。

三 概要设计



四 详细设计

1. 类的函数成员：

```
class Student
{
public:
    char name[27];
    int Id;
    int Cnum;
    int Enum;
    int Mnum;
    int sum;

    Student*Next;
    Student*Head;
    Student*End;

    void CWHAddStu()
```

```

{
    static int j = 0;
    cout << "请输入学生学号"; cin >> Id; CWHstu[j].Id = Id;
    j++;
loop:    int a = j - 2;
    do
    {
        if (Id == CWHstu[a].Id)
        {
            cout << "对不起，该学号的同学的成绩已录入过，请重新输入学号！\n";
            cin >> Id;
            CWHstu[j].Id = Id;
            goto loop;
        }
        a--;
    } while (a >= 0);

    cout << "请输入学生姓名"; cin >> name;
    cout << "请输入C++课程的成绩"; cin >> Cnum;
    cout << "请输入数学课程的成绩"; cin >> Mnum;
    cout << "请输入英语课程的成绩"; cin >> Enum;
    sum = Cnum + Enum + Mnum;
}

void ReadFile(istream&in)
{
    in >> name >> Id >> Cnum >> Mnum >> Enum >> sum;
}

void Show()
{
    cout << "    \t    " << Id << "    \t    " << name << "    \t    " << Cnum <<
    "    \t    " << Mnum << "    \t    " << Enum << "    \t" << sum << endl;
}

}CWHstu[27];

```

3. 数据成员设计：

数据成员包括：学生学号、姓名 C++课程成绩、数学成绩、英语成绩、总分、指向类成员的指针。

4. 界面设计：

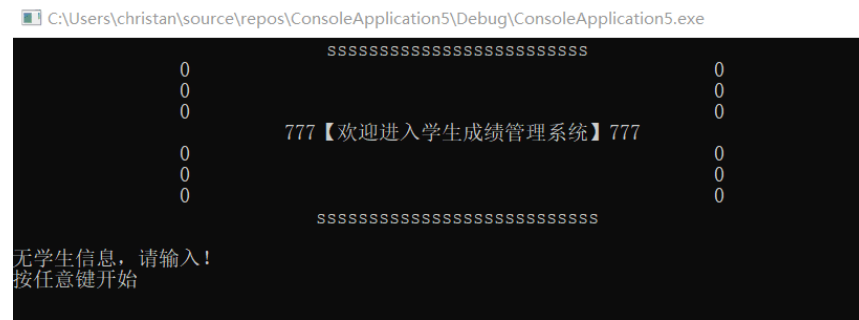
通过输出文字和字符以及<windows.h>头文件里面包含的函数和转义字符来实现简单的可视化。

5. 其它模块设计与实现：

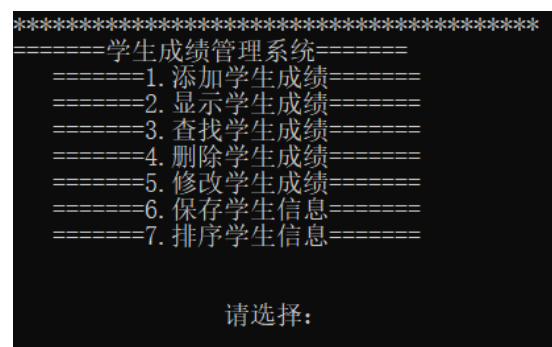
部分由手动调整和文本进行行列排序。

五 测试

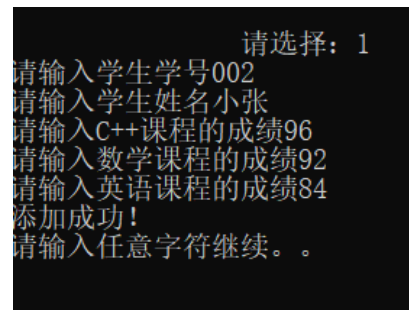
运行界面



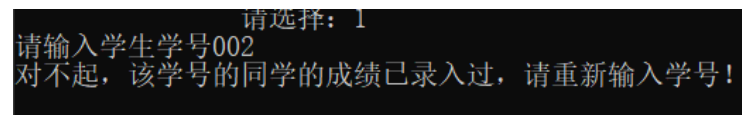
主菜单



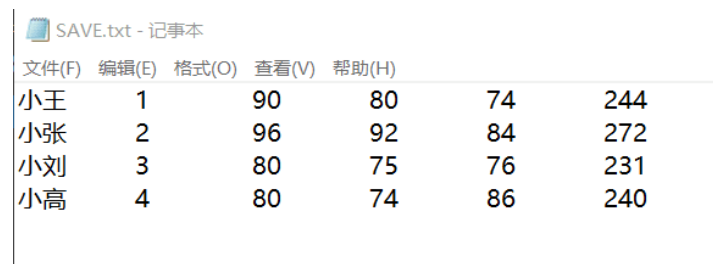
添加成绩



学号重复时自动提醒



信息保存



对不同科目进行排序

ID:	1	姓名:	小王	C++:	95
ID:	2	姓名:	小刘	C++:	75
ID:	3	姓名:	小张	C++:	74
ID:	4	姓名:	小高	C++:	73

数学:	85
数学:	84
数学:	84
数学:	83

英语:	82
英语:	75
英语:	73
英语:	72

修改信息:

ID: 239	1	姓名:	小张	C++:	74	数学:	83	英语:	82	总分:
ID: 232	2	姓名:	小高	C++:	73	数学:	84	英语:	75	总分:
ID: 232	3	姓名:	小刘	C++:	75	数学:	84	英语:	73	总分:
ID: 230	88	姓名:	小刚	C++:	90	数学:	70	英语:	70	总分:

将小刚及其信息改为小红

```

请选择: 5
请选择 1. 按姓名查询 2. 按学号查询1
请输入要修改信息的学生的姓名: 小刚
已找到学生信息, 请输入新的信息请输入学生学号004
对不起, 该学号的学生的成绩已录入过, 请重新输入学号!
088
对不起, 该学号的学生的成绩已录入过, 请重新输入学号!
777
请输入学生姓名小红
请输入C++课程的成绩78
请输入数学课程的成绩79
请输入英语课程的成绩80
修改成功
输入任意字符才可继续~~~~

```

[illegible]

保存后



SAVE.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

小红	777	78	79	80	237
小张	2	96	92	84	272
小刘	3	80	75	76	231
小高	4	80	74	86	240

六 结论

本次学生成绩信息管理系统使我对较大程序的设计与开发有了较为初步的认识。因为是第一次独自进行设计，难免会在一开始遇到很多困难，也会有手足无措的情况，但在通过书籍，网络查阅相关资料并参考后成功解决，但就目前来看代码仍然不够简洁，且信息保存需要手动保存，而且在应对一些问题时可以看出，我应该更多的去看 C++ 程序的实例并针对自己较为薄弱的知识点进行练习，加强在进行编程时的思维连贯性，能够做到提前对自己的程序有一个清晰的构架，本次编写学生信息管理系统时问题主要出现在了函数的重载和指针与数组的使用，行列间距的控制，以及系统各部分的连接上，有时也会对一些基本概念出现遗忘的情况，今后需要巩固基础，加强对程序中算法的设计。

七 附录

```
#include <iostream>
#include <fstream>
#include <string.h>
#include <conio.h>
#pragma warning(disable:4996) #查询自CSDN
using namespace std;

class Student
{
public:
    char name[27];
    int Id;
    int Cnum;
    int Enum;
    int Mnum;
    int sum;

    Student*Next;
    Student*Head;
    Student*End;

    void CWHAddStu()
    {
        static int j = 0;
```

```

        cout << "请输入学生学号"; cin >> Id; CWHstu[j].Id = Id;
        j++;
loop:    int a = j - 2;
        do
        {
            if (Id == CWHstu[a].Id)
            {

                cout << "对不起，该学号的同学的成绩已录入过，请重新输入学号! \n";
                cin >> Id;
                CWHstu[j].Id = Id;
                goto loop;
            }
            a--;
        } while (a >= 0);

        cout << "请输入学生姓名"; cin >> name;
        cout << "请输入C++课程的成绩"; cin >> Cnum;
        cout << "请输入数学课程的成绩"; cin >> Mnum;
        cout << "请输入英语课程的成绩"; cin >> Enum;
        sum = Cnum + Enum + Mnum;
    }

    void ReadFile(istream&in)
    {
        in >> name >> Id >> Cnum >> Mnum >> Enum >> sum;
    }
    void Show()
    {
        cout << "    \t    " << Id << "    \t    " << name << "    \t    " << Cnum
<< "    \t    " << Mnum << "    \t    " << Enum << "    \t" << sum << endl;
    }
}CWHstu[27];

```

```

//CWHStudentmessage类
class CWHStudentmessage
{
public:
    CWHStudentmessage();
    ~CWHStudentmessage();
    int CWHlistcount();

```



```

void CWHShowMenu();
void CWHFind();
void CWHSave();
void CWHModifyItem();
void CWHRemoveItem();
void CWHSwap(Student*, Student*);
void CWHsort1();
void CWHsort2();
void CWHsort3();
void CWHsort4();

void CWHDisplay()
{
    cout << "          -----成绩单-----"
<< endl;
    cout << "      \t      学号\t      姓名\t      C++\t      数学\t      英语\t
总成绩 " << endl;
    for (Student*p = Head->Next; p != End; p = p->Next)
        p->Show();
    cout << "请输入任何字符，继续。。。";
    getch();
}

void AddItem()
{
    End->CWHAddStu();
    End->Next = new Student;
    End = End->Next;
    cout << "添加成功!" << endl;
    cout << "请输入任意字符继续。。";
    getch();
}

private:
    Student*Head, *End;
    ifstream in;
    ofstream out;

    Student*FindItem(char*name)
    {
        for (Student*p = Head; p->Next != End; p = p->Next) //匹配成功则返回上一个指针，
不成功则返回空。
            if (!strcmp(p->Next->name, name))return p;
        return NULL;
    }

    Student*FindID(int *Id)

```

```

    {
        for (Student*p = Head; p->Next != End; p = p->Next) //匹配成功则返回上一个指针，
        不成功则返回空。
            if (p->Next->Id == *Id) return p;

        return NULL;
    }
};

```

//构造函数*****

```

CWHStudentmessage::CWHStudentmessage()
{
    Head = new Student;
    Head->Next = new Student;
    End = Head->Next;
    in.open("sort.txt");
    if (!in)
        cout << "无学生信息，请输入！" << endl;
    else
    {
        while (!in.eof())
        {
            End->ReadFile(in);
            if (End->name[0] == '\0') break;
            End->Next = new Student;
            End = End->Next;
        }
        in.close();
        cout << "\t\t成功读取学生信息" << endl;
    }
}

```

//*****析构函数*****

```

CWHStudentmessage::~CWHStudentmessage()
{
    CWHSave();
    for (Student*temp; Head->Head != End; )
    {
        temp = Head->Next;
        Head->Next = Head->Next->Next;
        delete temp;
    }
    delete Head, End;
}

```

//菜单menu

void CWHStudentmessage::CWHShowMenu()

```
{
    cout << "*****" << endl;
    cout << "=====学生成绩管理系统=====" << endl;
    cout << "    =====1. 添加学生成绩=====" << endl;
    cout << "    =====2. 显示学生成绩=====" << endl;
    cout << "    =====3. 查找学生成绩=====" << endl;
    cout << "    =====4. 删除学生成绩=====" << endl;
    cout << "    =====5. 修改学生成绩=====" << endl;
    cout << "    =====6. 保存学生信息=====" << endl;
    cout << "    =====7. 排序学生信息=====" << endl;
    cout << "\n\t\t\n\t\t请选择: ";
}
```

//查找函数search

void CWHStudentmessage::CWHFind()

```
{
    char name[27], Id[27];
    int x;
    Student*p = NULL;
    cout << "\t\t* 1. 按学生姓名查找\n\t\t* 2. 按学生学号查找";
    cout << "\n\t\t*****\n请选择: ";
    cin >> x;
    switch (x)
    {
    case 1:
    {
        cout << "请输入要查找学生的姓名: ";
        cin >> name;
        if (p = FindItem(name))
        {
            p->Next->Show();
            cout << "随意输入才能继续";
            getch();
        }
    }break;
    case 2:
    {
        cout << "请输入要查找学生的学号: ";
        cin >> Id;
        if (p = FindItem(Id))
        {
```

```

        p->Next->Show();
        cout << "输入任意符号才能继续";
        getch();
    }
    else
    {
        cout << "\t\t没有找到该学号的学生。。";
        cout << "输入任意字符才能继续";
        getch();
    }
}break;
}
}

//修改信息*****
void CWHStudentmessage::CWHModifyItem()
{
    int x;
    char name[27], Id[27];
    Student*p = NULL;
    cout << "请选择 1. 按姓名查询 2. 按学号查询";
    cin >> x;
    switch (x)
    {
    case 1:
    {
        cout << "\t\t请输入要修改信息的学生的姓名: ";
        cin >> name;
        if (p = FindItem(name))
        {
            cout << "已找到学生信息, 请输入新的信息";
            p->Next->CWHAddStu();
            cout << "修改成功" << endl;
            cout << "输入任意字符才可继续~~~";
            getch();
        }
    }break;
    case 2:
    {
        cout << "\t\t请输入要修改信息的学生的学号: ";
        cin >> Id;
        if (p = FindItem(Id))
        {
            cout << "已找到学生信息, 请输入新的信息";

```

```

        p->Next->CWHAddStu();
        cout << "修改成功" << endl;
        cout << "输入任意字符才可继续" << endl;
        getch();
    }
    else
    {
        cout << "未找到该学生信息";
        cout << "输入任意字符继续";
        getch();
    }
}
}

//删除信息*****
void CWHStudentmessage::CWHRemoveItem()
{
    int x = 0;
    char name[27], Id[27];
    Student*p = NULL, *temp = NULL;
    cout << "请选择通过1. 学生姓名 2. 学生学号查找学生";
    cin >> name;
    switch (x)
    {
    case 1:
        cout << "请输入要删除学生的姓名";
        cin >> name;
        if (p = FindItem(name))
        {
            cout << "已找到所查找的学生!" << endl;
            temp = p->Next;
            p->Next = p->Next->Next;
            delete temp;
            cout << "删除成功";
            cout << "输入任意字符继续。。。。";

            }break;
        case 2:
            cout << "请输入要删除学生的学号";
            cin >> Id;
            if (p = FindItem(Id))
            {
                cout << "已找到所查找的学生!" << endl;

```

```

        temp = p->Next;
        p->Next = p->Next->Next;
        delete temp;
        cout << "删除成功";
        cout << "输入任意字符继续。。。。";
    }
    else
    {
        cout << "没有找到该学生! ";
        cout << "输入任意字符继续";
        getch();
    }
}

}

void CWHStudentmessage::CWHSwap(Student*p1, Student*p2)
{
    Student*temp = new Student;
    strcpy(temp->name, p1->name);
    temp->Id, p1->Id;
    temp->Cnum = p1->Cnum;
    temp->Mnum = p1->Mnum;
    temp->Enum = p1->Enum;
    temp->sum = p1->sum;

    strcpy(p1->name, p2->name);
    p1->Id, p2->Id;
    p1->Cnum = p2->Cnum;
    p1->Mnum = p2->Mnum;
    p1->Enum = p2->Enum;
    p1->sum = p2->sum;

    strcpy(p2->name, temp->name);
    p2->Id, temp->Id;
    p2->Cnum = temp->Cnum;
    p2->Mnum = temp->Mnum;
    p2->Enum = temp->Enum;
    p2->sum = temp->sum;
}

//SORT*****
//
int CWHStudentmessage::CWHlistcount()

```

```

{
    if (!Head)
        return 0;
    int n = 0;
    for (Student*p = Head->Next; p != End; p = p->Next)
    {
        n++;
    }
    return n;
}

```

```

void CWHStudentmessage::CWHsort1()
{
    cout << "CWH-排序" << endl;
    Student*p = NULL, *k = NULL;
    for (p = Head->Next; p != End; p = p->Next)
    {
        for (k = Head->Next; k != End; k = k->Next)
        {
            if (p->Cnum > k->Cnum)
            {
                CWHStudentmessage::CWHSwap(p, k);
            }
        }
    }
    cout << "排序完成~" << endl;
    getch();
}

```

```

void CWHStudentmessage::CWHsort2()
{
    cout << "CWH-排序" << endl;
    Student*p = NULL, *k = NULL;
    for (p = Head->Next; p != End; p = p->Next)
    {
        for (k = Head->Next; k != End; k = k->Next)
        {
            if (p->Mnum > k->Mnum)
            {
                CWHStudentmessage::CWHSwap(p, k);
            }
        }
    }
    cout << "排序完成~" << endl;
    getch();
}

```

```

}
void CWHStudentmessage::CWHSort3()
{
    cout << "CWH-排序" << endl;
    Student*p = NULL, *k = NULL;
    for (p = Head->Next; p != End; p = p->Next)
    {
        for (k = Head->Next; k != End; k = k->Next)
        {
            if (p->Enum > k->Enum)
            {
                CWHStudentmessage::CWHSwap(p, k);
            }
        }
    }
    cout << "排序完成~" << endl;
    getch();
}
void CWHStudentmessage::CWHSort4()
{
    cout << "CWH-排序" << endl;
    Student*p = NULL, *k = NULL;
    for (p = Head->Next; p != End; p = p->Next)
    {
        for (k = Head->Next; k != End; k = k->Next)
        {
            if (p->sum > k->sum)
            {
                CWHStudentmessage::CWHSwap(p, k);
            }
        }
    }
    cout << "排序完成~" << endl;
    getch();
}
//保存函数*****
void CWHStudentmessage::CWHSave()
{
    ofstream out("SAVE.txt", ios::out);
    for (Student*p = Head->Next; p != End; p = p->Next)
        out << p->name << "\t" << p->Id << "\t" << p->Cnum << "\t" << p->Mnum << "\t" <<
        p->Enum << "\t" << p->sum << "\n";
    out.close();
}

```


[illegible]

```
        Grade.CWHsort3();
        break;
    case 4:
        Grade.CWHsort4();
        break;
    }
}
}
return 0;
}
```

报告附页

原创性声明：

本人声明报告者中的内容和程序为本人独立完成，引用他人的文献、数据、图件、资料均已明确标注出。除标注内容外，不包含任何形式的他人成果，无侵权和抄袭行为，并愿意承担由此而产生后果。

作者签字： 时间：

指导教师评语：

课程成绩：

程序界面 (10 分)	程序结构和功能 (40 分)	程序安全和效率 (40 分)	报告 (10 分)	总分

指导教师签字：

时间