

# 北航 2022 秋 c++程序设计报告

陈子涵

## 一、程序设计名称

学生成绩管理系统

## 二、程序设计目的

本程序设计是对前面多学 c++ 知识的总结，通过设计一个功能比较完整的成绩管理系统，将所学的 c++ 知识连贯起来，进一步掌握 c++ 知识，同时能更加熟练的运用开发工具。通过本次系统设计，也能更好掌握如何运用面向对象的设计。

## 三、基本知识点

- 1、类和对象的定义、对象的初始化和使用
- 2、面向对象的继承机制
- 3、虚函数与多态性
- 4、运算符重载
- 5、文件的使用
- 6、算法的使用

## 四、程序内容

### 4.1 登录界面

#### 4.1.1 功能介绍

登录界面主要由两个输入框（Edit control）和一个按钮（Button）控件构成，在账户和输入密码框处输入程序设定好的固定账户密码点击登录即可进入系统管理界面，输入数字在输入密码框处是不可见的。

#### 4.1.2 登录界面图片



### 4.1.3 部分代码说明

```
void CmanagementSystemDlg::OnBnClickedOk()
{
    //在此添加控件通知处理程序代码
    UpdateData(TRUE);
    /*当你使用了ClassWizard建立了控件和变量之间的联系后：当你修改了变量的值，而希望对话框控件更新显示，就应该在修改变量后调用 UpdateData (FALSE)；如果你希望知道用户在对话框中到底输入了什么，就应该在访问变量前调用UpdateData(TRUE)。 用处：主要控制ddx_data 把控件中的值和变量进行交换，比如你有个编辑框，输入了一些值，你的程序怎么得到呢；必须先调用这个（当然是用了ddx的情况）反之控件的成员变量中的值也可以通过该函数自动显示到控件里面，根据传进来的是true或者false如果你有个编辑框，输入了一些值，程序怎么得到呢，必须先调用这个情况，反之控件的成员变量中的值也可以通过该函数自动显示到控件里面，根据传进来的是true或者false */
    CString zhang = zhanghao;           //Csrtring是mfc中常见的类之一，用于封装字符串数据结构
    CString mi = mima;
    if (zhang == "youni" && mi == "123") //确认密码值
    {
        ShowWindow(SW_HIDE);          //关闭当前窗口
        mymenu dlgs;                //创建新的类
        dlgs.DoModal();             //DoModal()弹出对话框 这句话的意思就是弹出一个mymenu类中的对话框
    }
    else if (zhang == "" || mi == "") //不能为空
    {
        MessageBox("账号或密码不能为空", "提醒");
        return;
    }
    else                         //别乱输密码捏
    {
        MessageBox("账号或密码错误，还有三次机会~", "提醒");
        return;
    }
    UpdateData(FALSE);            //更新编辑窗口的显示值
    CDialogEx::OnOK();
}

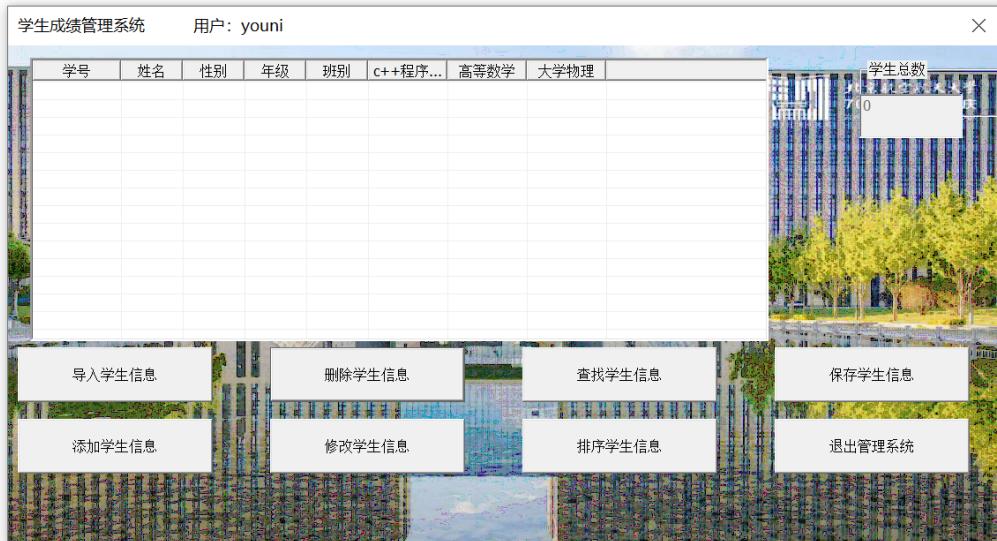
BOOL CmanagementSystemDlg::PreTranslateMessage(MSG* pMsg)           //重载它来处理键盘和鼠标消息，这段代码主要用于点击编辑框，会把默认文本清除
{
    CEdit* pEdit = (CEdit*)GetDlgItem(IDC_EDIT1);           //获取该控件指针
    CEdit* pEdit2 = (CEdit*)GetDlgItem(IDC_EDIT2);
    CString s = (CString)"";
    if (pMsg->message == WM_LBUTTONDOWN)                  //WM_LBUTTONDOWN是一个windows消息，该消息当用户在window客户区域点击鼠标左键的时候发送
    {
        UpdateData(TRUE);
        if (pMsg->hwnd == pEdit->GetSafeHwnd())
        {
            //SetDlgItemText(IDC_EDIT1,"要显示的内容")
            CEdit* kd = (CEdit*)GetDlgItem(IDC_EDIT1);
            kd->SetPasswordChar('*');                      //将密码输入显示为*
            pEdit->SetWindowText(_T(""));
            if (zhanghao == s)                            //将该输入框变为空，_T表示储存时用unicode而不用系统默认的如ansi
            {
                pEdit2->SetWindowText("输入账户名");
            }
        }
        else if (pMsg->hwnd == pEdit2->GetSafeHwnd())      //如果指向的是第二个编辑框
        {
            pEdit2->SetWindowText("");
            if (mima == s)
            {
                CEdit* kd = (CEdit*)GetDlgItem(IDC_EDIT1);
                kd->SetPasswordChar(NULL);
                pEdit->SetWindowText("输入密码");
            }
        }
    }
    return CDialogEx::PreTranslateMessage(pMsg);
}
```

## 4.2 系统管理界面

### 4.2.1 功能介绍

系统管理界面主要由一个表格列表（List Control）、一个组合框（Group Box）和八个按钮（Button）控件组成。表格列表显示学生的学号、姓名、性别、年级、班级、c++程序设计、高等数学和大学物理信息。八个按钮对应的是八个功能，分别是添加学生信息、删除学生信息、保存学生信息、查找学生信息、导入学生信息、修改学生信息、排序学生信息和退出管理系统。

### 4.2.2 系统管理界面图片



### 4.2.3 部分代码说明

```
BOOL mymenu::OnInitDialog()
{
    CDialogEx::OnInitDialog();
    //添加额外的初始化
    CString str = "学生成绩管理系统";
    SetWindowText(str);

    //设置表格的风格
    CRect rect; //Rect结构的成员(left,top,right和bottom),也是CRect的可访问数据成员
    //Rect成员:left:指定矩形左上角的X坐标 top:指定矩形左上角的Y坐标 right:指定矩形框右下角的X坐标 bottom:指定矩形框右下角的Y坐标Rect。
    CListCtrl* pmyListCtrl = (CListCtrl*)GetDlgItem(IDC_LIST2);
    //设置表格为报表模式
    LONG dwStyle = GetWindowLong(pmyListCtrl->m_hWnd, GWL_STYLE);
    SetWindowLong(pmyListCtrl->m_hWnd, GWL_STYLE, dwStyle | LVS_REPORT);
    //设置表格可以整行选择和网格条纹
    LONG styles = pmyListCtrl->GetExtendedStyle();
    pmyListCtrl->SetExtendedStyle(styles | LVS_EX_FULLROWSELECT | LVS_EX_GRIDLINES | LVS_NOSCROLL);
    pmyListCtrl->GetWindowRect(&rect);
    //InsertColumn(0,"学号", LVCFT_CENTER, 100) 第一个参数是插入的列 第二个参数是列名
    //第三个参数是LVCFT_CENTER是字体居中 但是存在第一列始终居左的问题
    //解决办法：把第一列当一个虚列，从第二列开始插入列及数据，最后删除第一列
    //最后一个参数决定列宽

    m_list.InsertColumn(0, "学号", LVCFT_CENTER, 100); //list control第一列会不在中间，但通过此方法可以很好的达到中间的目的
    m_list.InsertColumn(1, "学号", LVCFT_CENTER, 100);
    m_list.InsertColumn(2, "姓名", LVCFT_CENTER, 70);
    m_list.InsertColumn(3, "性别", LVCFT_CENTER, 70);
    m_list.InsertColumn(4, "年级", LVCFT_CENTER, 70);
    m_list.InsertColumn(5, "班级", LVCFT_CENTER, 70);
    m_list.InsertColumn(6, "c++程序设计", LVCFT_CENTER, 90);
    m_list.InsertColumn(7, "高等数学", LVCFT_CENTER, 90);
    m_list.InsertColumn(8, "大学物理", LVCFT_CENTER, 90);
    m_list.DeleteColumn(0); //删除第一列

    import();
    //文件导入函数
    m_stusum = m_line;
    UpdateData(FALSE); //UpdateData(true)用于将屏幕上控件中的数据交换到变量中 (false) 用于将数据在屏幕上对应控件中显示出来
    GetDlgItem(IDC_EDIT1)->EnableWindow(FALSE);

    m_edit.ShowWindow(SW_HIDE);
    // TODO: 在此添加额外的初始化，用于隐藏窗口
    return TRUE; // return TRUE unless you set the focus to a control
    // 异常: OCX 属性页应返回 FALSE
}
```

## 4.3 添加学生信息功能

### 4.3.1 功能介绍

添加学生信息界面主要由七个输入框（Edit control），一个勾选框（Radio Button）和两个按钮（Button）控件组成。该窗口是系统管理界面窗口的子窗口之一，主要作用是将学生的成绩信息等填入系统。当有输入框没有填入时，点击添加按钮，会弹出没有添加对应框信息的警告。

### 4.3.2 添加学生信息功能图片



### 4.3.3 部分代码说明

```
void mymenu::OnBnClickedButton1()//将数据加载至列表框
{
    // TODO: 在此添加控件通知处理程序代码
    while (1)
    {
        adds.xing = -1;

        if (adds.DoModal() == IDOK && empty() && adds.FA == FALSE && checkgrade())
        {
            char c[15];

            if (checknumber(c) && checkxingbie())
            {
                char c1[5], c2[5], c3[5];
                _itoa_s(adds.program, c1, 10);           //将adds.class1的数值转换为字符串放到数组c1中，以10进制转换
                _itoa_s(adds.math, c2, 10);
                _itoa_s(adds.lishan, c3, 10);
                m_list.InsertItem(0, c);                  //每次都是把新的信息加到第一行
                m_list.SetItemText(0, ++m_index, adds.name);
                m_list.SetItemText(0, ++m_index, adds.m_set);
                m_list.SetItemText(0, ++m_index, adds.grade);
                m_list.SetItemText(0, ++m_index, adds.classss);
                m_list.SetItemText(0, ++m_index, c1);
                m_list.SetItemText(0, ++m_index, c2);
                m_list.SetItemText(0, ++m_index, c3);
                //SetItemText这个函数一共有三个参数，第一个是要设置文本的项目号，第二个是子项目号，第三个是要设置的文本
                //要先用InsertItem插入一个item，再用SetItemText才有效
                m_line++;

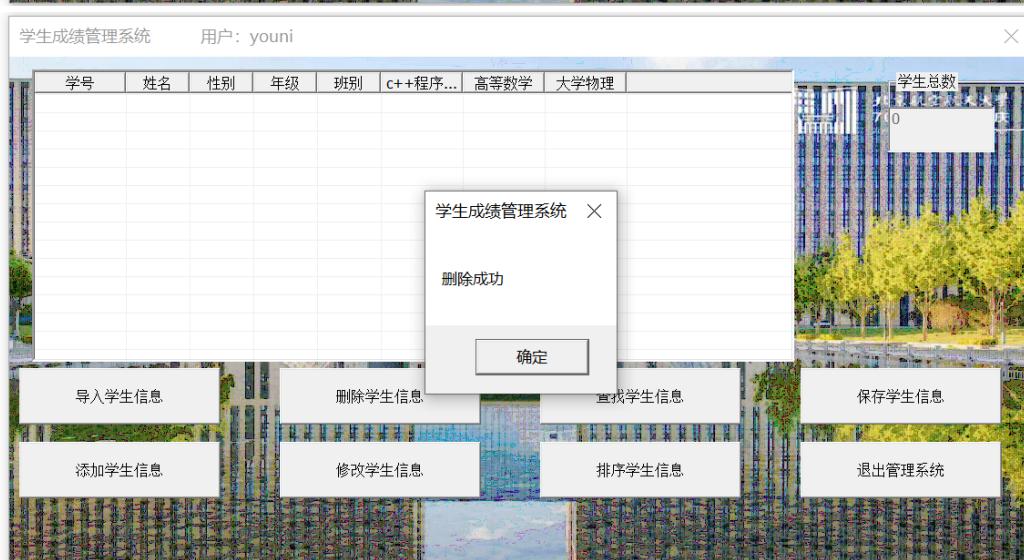
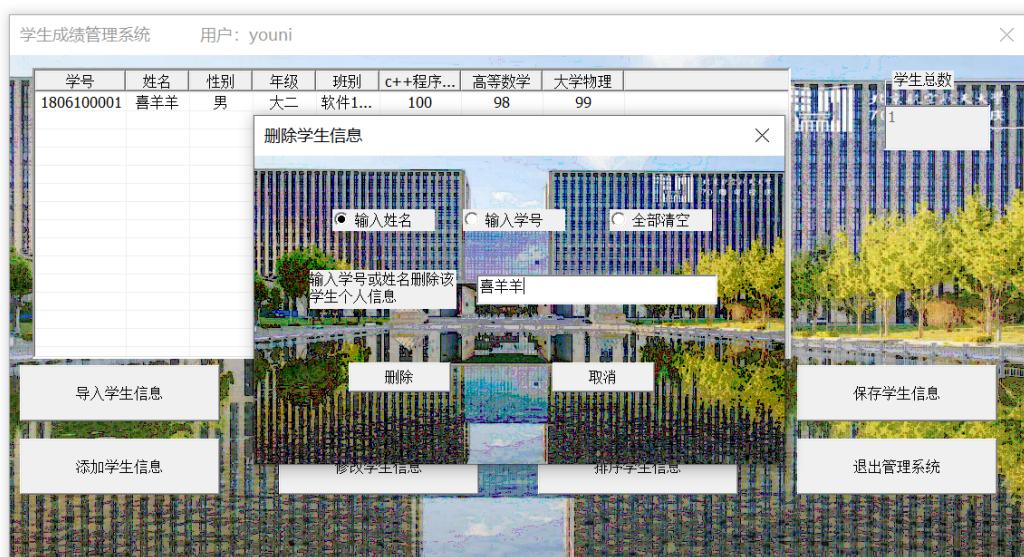
                m_stusum = m_line;                      //用于显示
                UpdateData(FALSE);
                m_index = 0;
                check = FALSE;
                adds.name = "";
                adds.grade = "";
                adds.classss = "";
                adds.number = 0;
                adds.program = 0;
                adds.math = 0;
                adds.lishan = 0;
            }
        }
    }
}
```

## 4.4 删除学生信息功能

### 4.4.1 功能介绍

删除学生信息界面主要由三个勾选框（Radio Button）和一个输入框（Edit control）组成。该窗口是系统管理界面窗口的子窗口之一，主要作用是将学生的成绩信息从系统中删除。删除可以选择通过姓名删除、通过学号删除和全部清空。将要删除的信息填好后，点击删除，系统会弹出删除成功的消息框，同时系统中对应学生的信息会被删除。

### 4.4.2 删除学生信息功能图片



#### 4.4.3 部分代码说明

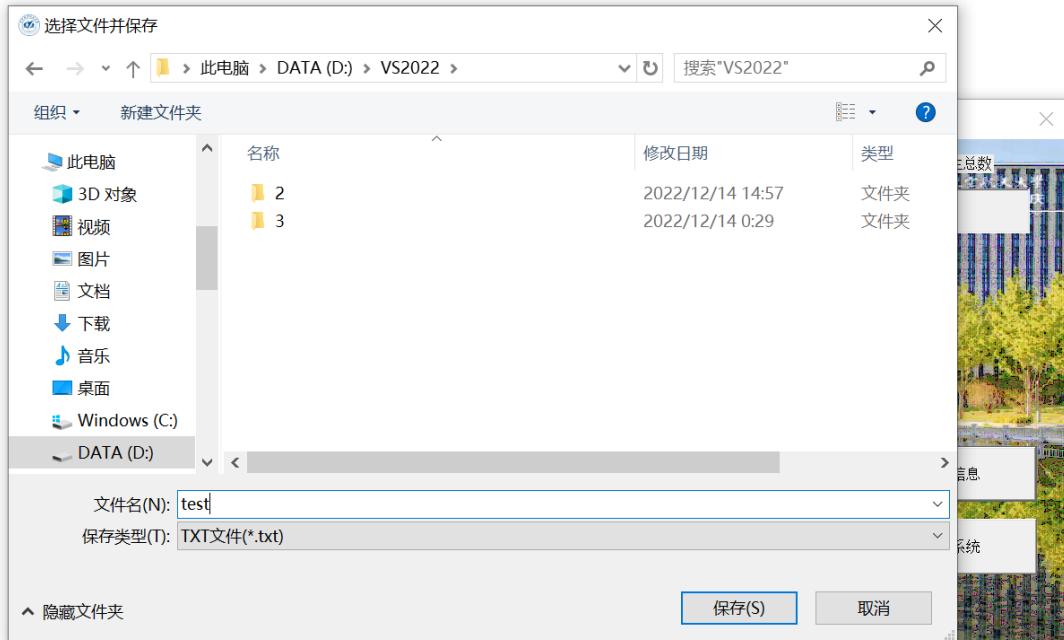
```
//-----删除-----\n\nvoid mymenu::deletenull() //删除列表框的所有数据\n{\n    m_list.DeleteAllItems();\n    m_line = 0;\n    m_stusum = m_line;\n    UpdateData(FALSE);\n}\n\nvoid mymenu::delete_q() //删除单个数据+调用上面函数删除所有数据\n{\n    //a=0实现姓名查找\n    if (deletes.a == 0 && deletes.b != "")\n    {\n        for (UINT i = 0; i < m_line; i++)\n        {\n            CString s = m_list.GetItemText(i, 1); //获取数据 获得第i行第1列的数据\n            if (deletes.b == s)\n            {\n                m_list.DeleteItem(i); //删除第i行\n                m_line--; //减少统计的行数\n                deletes.b = ""; //清零查找学号\n                judge = TRUE; //判断开关\n                m_stusum = m_line;\n                UpdateData(FALSE);\n                MessageBox("删除成功");\n                break;\n            }\n        }\n        if (judge == FALSE)\n            MessageBox("没有此名字的学生");\n    }\n\n    //a=1实现学号查找\n    else if (deletes.a == 1 && deletes.b != "")\n    {\n        for (UINT i = 0; i < m_line; i++)\n        {\n            CString s = m_list.GetItemText(i, 0); //获取数据 获得第i行第0列的数据\n            if (deletes.b == s)\n            {\n                m_list.DeleteItem(i);\n                m_line--;\n                deletes.b = "";\n                judge = TRUE;\n                m_stusum = m_line;\n                UpdateData(FALSE);\n                MessageBox("删除成功");\n                break;\n            }\n        }\n        if (judge == FALSE)\n            MessageBox("没有此学号的学生");\n    }\n\n    //a=2全部删除\n    else if (deletes.a == 2)\n    {\n        deletenull();\n    }\n    else\n    {\n        MessageBox("请输入一项数据删除");\n    }\n}
```

## 4.5 保存学生信息功能

### 4.5.1 功能介绍

点击保存学生信息按钮，会打开文件保存窗口，输入保存文件名和选择保存类型即可将系统中的学生信息以规定文件形式保存到用户电脑中。

### 4.5.2 保存学生信息功能图片



### 4.5.3 部分代码说明

```
void mymenu::OnBnClickedButton3()
{
    //在此添加控件通知处理程序代码
    BOOL jdg = FALSE;

    mysave saves;
    CFileDialog filesaves(FALSE);      //true表示创建一个打开文件的对话框 false表示创建一个另存为对话框
    filesaves.m_ofn.lpstrTitle = "选择文件并保存";
    filesaves.m_ofn.lpstrFilter = "TXT文件 (*.txt)|*.txt|Word文档 (*.doc)|*.doc|所有文件 (*.*)|*.*|";
    if (filesaves.DoModal() == IDOK)    //弹出的保存文件对话如果按下保存按钮
    {
        filename = filesaves.GetPathName(); //得到文件路径名称
        finds.filenaem = filename;       //将文件路径保存到系统，系统关闭后再打开可以自动初始化添加该数据
    }
    if (saves.DoModal() == IDOK && nodata())
    {
        if (filename == "")MessageBox("请选择保存的文件名");
        else
        {
            CFile file;
            file.Open(filename, file.modeWrite | file.modeCreate); //通过写入形式访问filename路径的文件 open函数打开文件
            /*文件操作使用的是文件指针，通过文件指针实现对指向的文件的各种操作。
            这些文件操作函数都是最终还是调用了操作系统的API函数或者处理过程与之类似
            Windows系统的API函数除了ReadFile，还有CreateFile、WriteFile函数。而MFC基于面向对象的思想，将这些Windows函数封装到了CFile类中，实现对文件的打开、关闭、读、写、获取文件信息等操作。使用CFile类对文件进行操作非常便捷。
            CFile类的成员函数
            CFile():
            CFile (HANDLE hFile):
            CFile ((LPCWSTR lpszFileName, UINT nOpenFlags) :
            以上三个成员函数都是CFile的构造函数，用于构造CFile对象。参数hFile为要关联到CFile对象的文件的句柄。参数lpszFileName为要关联到CFile对象的文件的相对路径或者绝对路径；参数nOpenFlags为文件访问选项的组合，通过各选项的按位或运算实现组合。
            下面的文件访问模式进项表中只能选择一个进行组合，默认取CFile::modeRead
            CFile::modeRead          只读方式访问文件
            CFile::modeWrite         写入方式访问文件
            CFile::modeReadWrite     读写方式访问文件
            CFile::modeCreate        如果文件不存在则创建文件，而如果存在则将它关联到此CFile对象并格长度截取为0
            CFile::modeNoTruncate   如果文件不存在则创建文件，而如果存在则将它关联到此CFile对象而不进行截取 */
    }
}
```

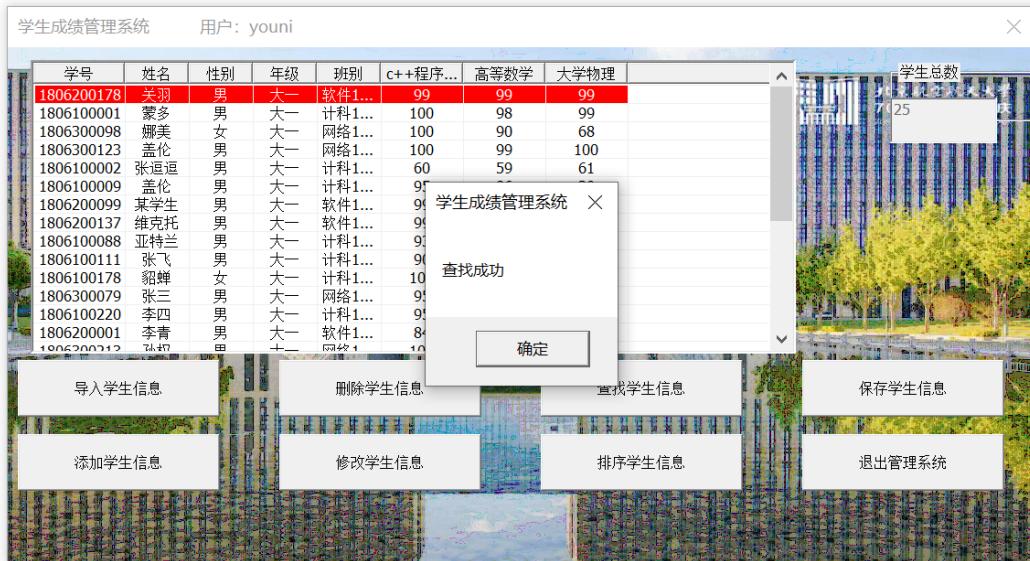
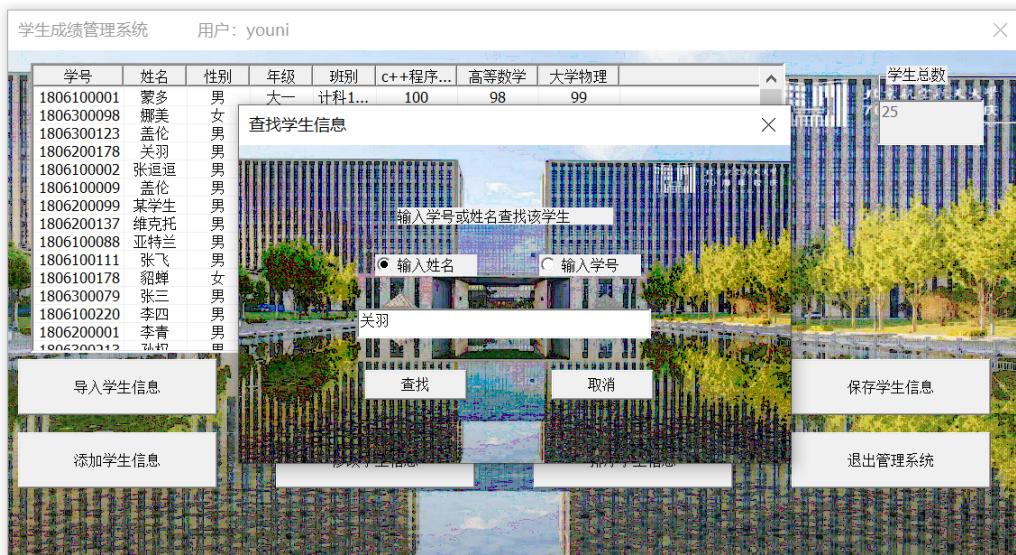
## 4.6 查找学生信息功能

### 4.6.1 功能介绍

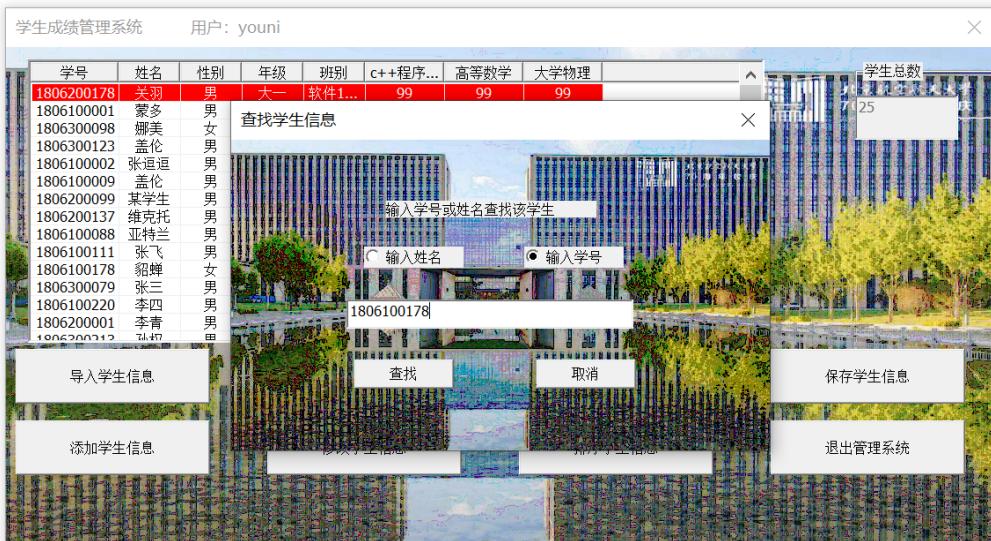
查找学生信息界面主要由两个勾选框（Radio Button）、一个输入框（Edit control）和两个按钮（Button）控件组成。该窗口是系统管理界面窗口的子窗口之一，主要作用是通过勾选选择输入姓名还是学号，在输入框中输入要查找的信息进行查找。

### 4.6.2 查找学生信息功能图片

#### (1) 输入姓名查找



## (2) 输入学号查找



### 4.6.3 部分代码说明

```
void mymenu::reset(UINT i) //把查找的同学原来的行数传过来，将其插入第一行，达到查找效果
{
    CString s1 = m_list.GetItemText(i, 0);
    CString s2 = m_list.GetItemText(i, 1);
    CString s3 = m_list.GetItemText(i, 2);
    CString s4 = m_list.GetItemText(i, 3);
    CString s5 = m_list.GetItemText(i, 4);
    CString s6 = m_list.GetItemText(i, 5);
    CString s7 = m_list.GetItemText(i, 6);
    CString s8 = m_list.GetItemText(i, 7);
    m_list.DeleteItem(i);
    m_list.InsertItem(0, s1);
    m_list.SetItemText(0, 1, s2);
    m_list.SetItemText(0, 2, s3);
    m_list.SetItemText(0, 3, s4);
    m_list.SetItemText(0, 4, s5);
    m_list.SetItemText(0, 5, s6);
    m_list.SetItemText(0, 6, s7);
    m_list.SetItemText(0, 7, s8);
    m_list.SetItemData(0, COLOR_RED); //给该行设置背景颜色和字体颜色使得更加突出
}
```

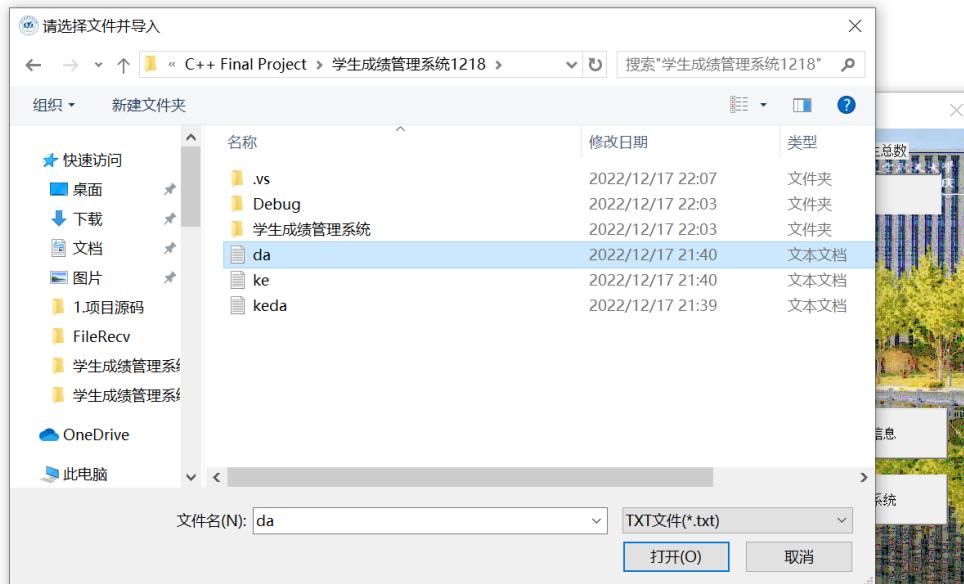
```
void mymenu::find_q() //查找某学号或某名字的学生的信息
{
    if (finds.a == 0 && finds.b != "")
    {
        for (UINT i = 0; i < m_line; i++)
        {
            CString s = m_list.GetItemText(i, 1);
            if (finds.b == s)
            {
                reset(i);
                finds.b = "";
                judge = TRUE;
                MessageBox("查找成功");
                break;
            }
        }
        if (judge == FALSE)
            MessageBox("没有此名字的学生");
    }
    else if (finds.a == 1 && finds.b != "")
    {
        for (UINT i = 0; i < m_line; i++)
        {
            CString s = m_list.GetItemText(i, 0);
            if (finds.b == s)
            {
                reset(i);
                finds.b = "";
                judge = TRUE;
                MessageBox("查找成功");
                break;
            }
        }
        if (judge == FALSE)
            MessageBox("没有此学号的学生");
    }
    else
    {
        MessageBox("请输入一项数据删除");
    }
}
```

## 4.7 导入学生信息功能

### 4.7.1 功能介绍

导入学生信息界面主要由三个按钮（Button）控件组成。该窗口是系统管理界面窗口的子窗口之一，主要作用是将电脑文件内的学生信息导入系统。在该窗口下可以选择清除当前系统数据导入新数据和保留当前数据导入新数据两种选项。

### 4.7.2 导入学生信息功能图片



### 4.7.3 部分代码说明

```
void mymenu::importtt()
{
    MessageBox("请输入学生信息!");
    CFileDialog filedlg(TRUE);
    filedlg.m_ofn.lpszTitle = "请选择文件并导入";
    filedlg.m_ofn.lpszFilter = "TXT文件 (*.txt)\0*.txt\0doc文档 (*.doc)\0*.doc\0所有文件 (*.*)\0*.*\0\0";
    if (IDOK == filedlg.DoModal())
    {
        finds.filenaem = filedlg.GetPathName();
        UINT num = importtt();
        if (num != 0)
            MessageBox("导入信息成功");
    }
}
```

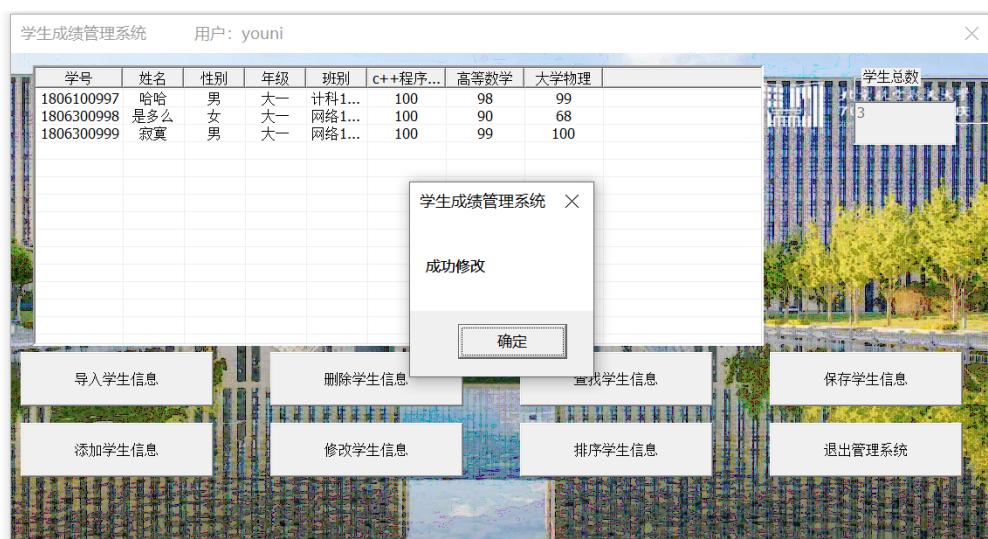
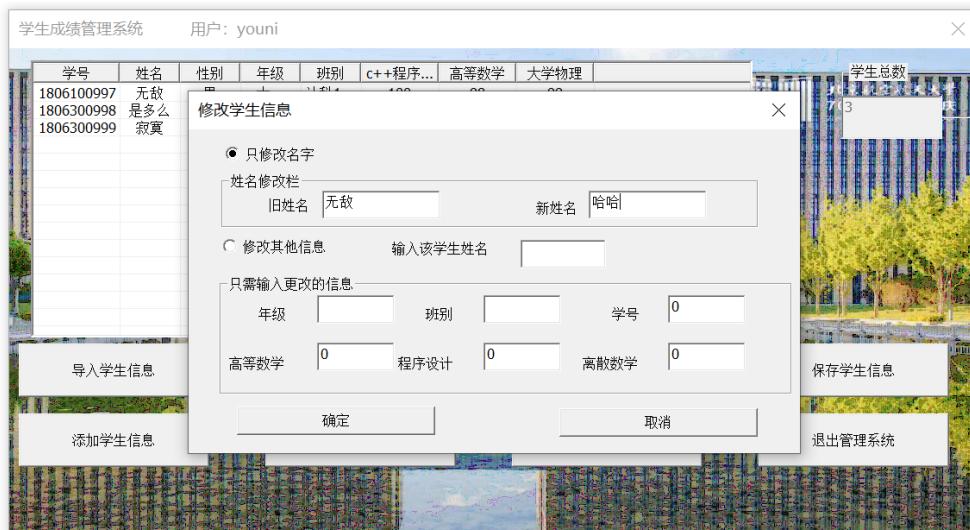
## 4.8 修改学生信息功能

### 4.8.1 功能介绍

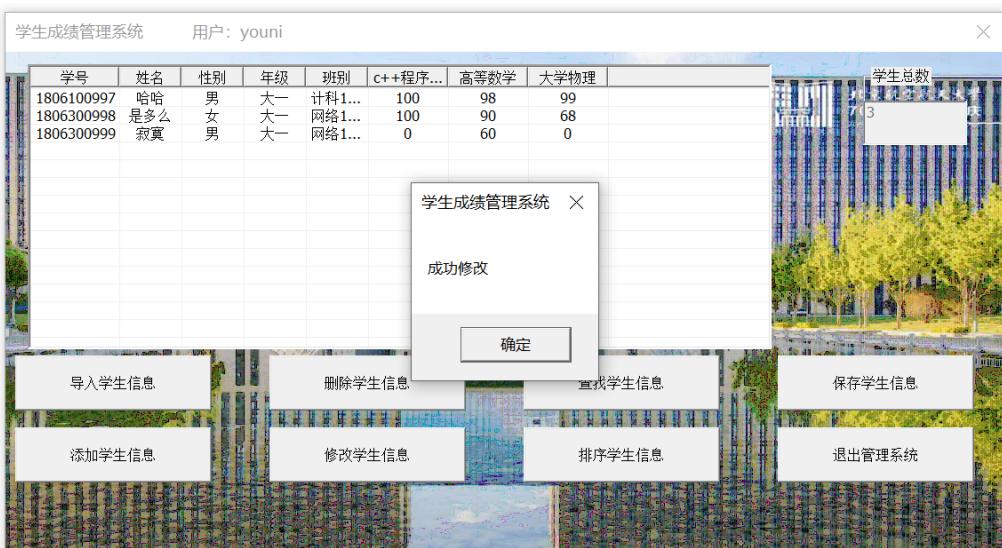
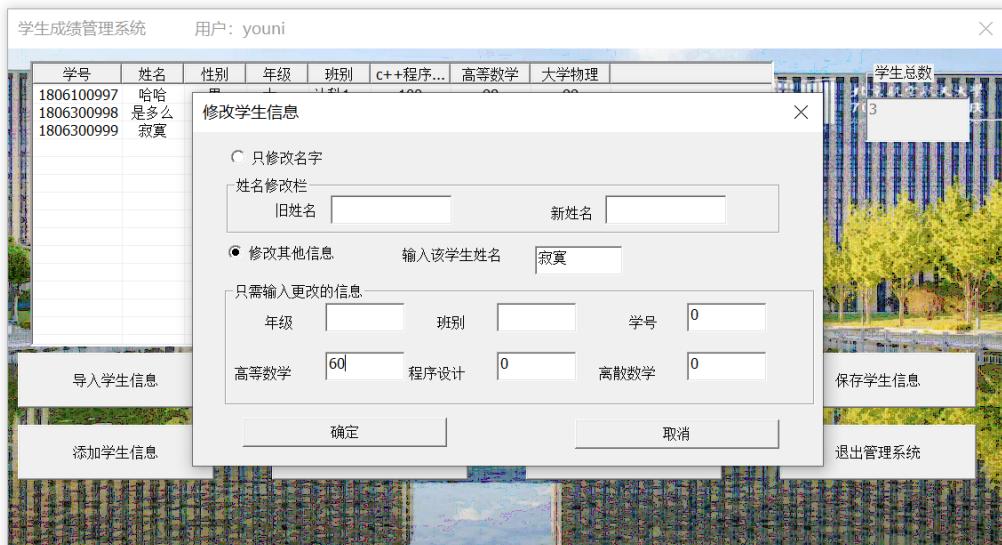
修改学生信息界面主要由两个勾选框(Radio Button)和九个输入框(Edit control)组成。该窗口是系统管理界面窗口的子窗口之一，主要作用是修改成绩系统中的学生信息。勾选框可以选择只修改名字或者修改除名字以外的其他信息。

### 4.8.2 修改学生信息功能图片

#### (1) 只修改名字



## (2) 修改其他信息



### 4.8.3 部分代码说明

```
void mymenu::findcheck()           //个人信息修改函数
{
    BOOL f = FALSE;
    if (changes.name != "") {
        for (UINT i = 0; i < m_line; i++) {
            if (changes.name == m_list.GetItemText(i, 1)) {
                settext(i);
                f = TRUE;
            }
            if (f) {
                MessageBox("成功修改");
                break;
            }
        }
        if (!f) {
            MessageBox("查找的人名字可能不符,请检查");
        }
    } else {
        MessageBox("请输入修改同学的新旧姓名");
    }
}

void mymenu::changename()          //只修改名字
{
    BOOL f = FALSE;
    for (UINT i = 0; i < m_line; i++) {
        if (changes.old_name == m_list.GetItemText(i, 1)) {
            m_list.SetItemText(i, 1, changes.new_name);
            f = TRUE;
        }
        if (f) {
            MessageBox("成功修改");
            break;
        }
    }
}

void mymenu::OnBnClickedButton6()
{
    // TODO: 在此添加控件通知处理程序代码

    while (!nodata())
    {
        changes.old_name = "";
        changes.new_name = "";
        changes.new_number = 0;
        changes.new_program = 0;
        changes.new_math = 0;
        changes.new_lishan = 0;
        changes.new_class = "";
        changes.new_grade = "";

        if (changes.DoModal() == IDOK)
        {
            UpdateData(TRUE);
            if (changes.cao == 0 && changes.old_name != "") {
                changename();
            }
            else if (changes.cao == 1 && changes.name != "") {
                findcheck();
            }
            else {
                MessageBox("请输入该学生的姓名");
            }
        }
        if (changes.FA == TRUE)
        {
            deletes.FA = FALSE;
            break;
        }
    }
    if (!nodata())
    {
        MessageBox("信息为空,请先添加信息");
    }
}
```

## 4.9 排序学生信息功能

### 4.9.1 功能介绍

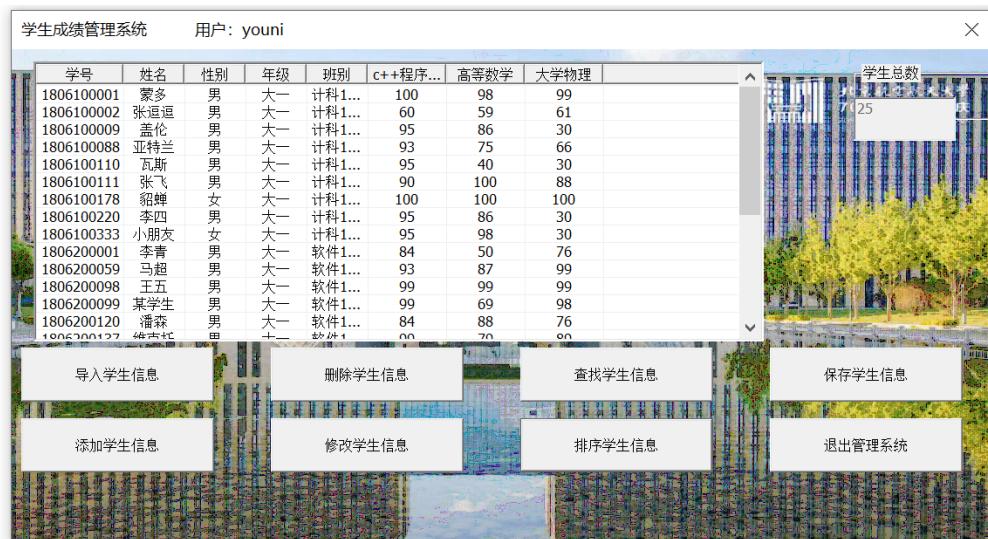
点击排序学生信息按钮后，系统中的学生信息会按照学号排序。

### 4.9.2 排序学生信息功能图片

(1) 未排序时的系统界面



(2) 排序后的系统界面



#### 4.9.3 部分代码说明

```
void mymenu::resort() //排序函数
{
    point* keda = new point[m_line]; //建立动态数组 结构体数组

    for (UINT i = 0; i < m_line; i++)
    {
        CString a = m_list.GetItemText(i, 0); //宏定义 实际是将字符串转换为整数型
        UINT AA = _toi(a);
        keda[i].x = i; //存放行数
        keda[i].y = AA; //存放该行第0列的数据
    }
    sort(keda, keda + m_line, cmp); //快排

    for (int i = 0; i < m_line; i++)
    {
        CString a = m_list.GetItemText(keda[i].x, 0); //定义字符串a=第i行第0列
        m_list.InsertItem(i + m_line, a); //给第i+m_line行0列放入a的值
        for (UINT j = 1; j < 8; j++)
        {
            CString a = m_list.GetItemText(keda[i].x, j); //获得第i行第1-7列的数据
            m_list.SetItemText(i + m_line, j, a); //将第i行第1-7列的数据放入第i+m_line行对应位置
        }
    }
    for (int i = 0; i < m_line; i++) //删除前i行数据
    {
        m_list.DeleteItem(0); //删除第0行
    }
    delete[] keda; //防止内存泄露
}
```

## 五、程序设计总结

通过这次 c++实验设计设计，用 c++语言实现一个可视化的程序界面，加深了我对 c++之前所学知识的理解，更加熟练了对 c++类的使用、类的继承、c++中 fopen 打开读写文件等知识的运用，在写代码的过程中也遇到了许多的问题，例如：

- (1) c++自带排序算法，直接用 sort 函数对数组进行排序即可。
- (2) 用 listcontrol 控件的时候，一般不会居中对齐，可以先设置两个一样的一二列，然后通过删除第一列的方法达到表格的居中对齐。
- (3) 数组过大时候要通过 new 在堆上建立数组，用完后要使用 delete 及时删除，不然会导致程序占用内存过大，最后造成电脑的死机崩溃，或者内存泄露。