

(2021.6.13)

```
/* 版本: 3.0
 * 时间: 2021.6.13
 * 作者: orall
 * 完善: 1.增加了查询成绩时候的排序
 *       2.增加了修改学号不存在的提示
 */

// MFCAplication3Dlg.cpp: 实现文件
//

#include "pch.h"
#include "framework.h"
#include "MFCAplication3.h"
#include "MFCAplication3Dlg.h"
#include "afxdialogex.h"

#include <iostream>
#include <windows.h>
#include <algorithm>
#include <cstdlib>
#include <string>
#include <ctime>
#include <vector>
#include <unordered_map>
#include <sstream>
using namespace std;

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

typedef struct str_stu {
    CString stu_num;    //学号
    CString stu_name;   //姓名
    int usual_exam;    //平时成绩
    int final_exam;    //期末成绩
}Str_stu;

class Student {
```

```

public:
    CString stu_name;        //姓名
    double usual_exam;      //平时成绩
    double final_exam;      //期末成绩
    CString stu_num;         //学号
    double total_exam;       //总评成绩
    Student(CString num, CString name, double us_ex, double fi_ex) :stu_name(name),
stu_num(num), usual_exam(us_ex), final_exam(fi_ex)
    {
        total_exam = usual_exam * 0.4 + final_exam * 0.6;
    }
    void modify(CString num, CString name, double us_ex, double fi_ex)
    {
        stu_name = name;
        stu_num = num;
        usual_exam = us_ex;
        final_exam = fi_ex;
        total_exam = int((double)usual_exam * 0.4 + (double)final_exam * 0.6);
    }
    void show() {

        //cout << "★" << endl;
    }
};

vector<Student> stu;

// 用于应用程序“关于”菜单项的 CAboutDlg 对话框

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

    // 对话框数据
#ifdef AFX_DESIGN_TIME
        enum { IDD = IDD_ABOUTBOX };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

    // 实现
protected:
    DECLARE_MESSAGE_MAP()

```

```

};

CAboutDlg::CAboutDlg() : CDialogEx(IDD_ABOUTBOX)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

// CMFCApplication3Dlg 对话框

CMFCApplication3Dlg::CMFCApplication3Dlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_MFCAPPLICATION3_DIALOG, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CMFCApplication3Dlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CMFCApplication3Dlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_EN_CHANGE(IDC_EDIT5, &CMFCApplication3Dlg::OnEnChangeEdit5)
    ON_BN_CLICKED(IDC_BUTTON1, &CMFCApplication3Dlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON4, &CMFCApplication3Dlg::OnBnClickedButton4)
    ON_BN_CLICKED(IDC_BUTTON5, &CMFCApplication3Dlg::OnBnClickedButton5)
    ON_BN_CLICKED(IDC_BUTTON7, &CMFCApplication3Dlg::OnBnClickedButton7)
    ON_BN_CLICKED(IDC_BUTTON8, &CMFCApplication3Dlg::OnBnClickedButton8)
    ON_BN_CLICKED(IDC_BUTTON6, &CMFCApplication3Dlg::OnBnClickedButton6)
    ON_BN_CLICKED(IDC_BUTTON9, &CMFCApplication3Dlg::OnBnClickedButton9)
END_MESSAGE_MAP()

// CMFCApplication3Dlg 消息处理程序

```

```

BOOL CMFCApplication3Dlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // 将“关于...”菜单项添加到系统菜单中。

    // IDM_ABOUTBOX 必须在系统命令范围内。
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != nullptr)
    {
        BOOL bNameValid;
        CString strAboutMenu;
        bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // 设置此对话框的图标。 当应用程序主窗口不是对话框时，框架将自动
    // 执行此操作
    SetIcon(m_hIcon, TRUE);      // 设置大图标
    SetIcon(m_hIcon, FALSE);    // 设置小图标

    // TODO: 在此添加额外的初始化代码

    return TRUE;  // 除非将焦点设置到控件，否则返回 TRUE
}

void CMFCApplication3Dlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

```

```

}

// 如果向对话框添加最小化按钮，则需要下面的代码
// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，
// 这将由框架自动完成。

void CMFCApplication3Dlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 用于绘制的设备上下文

        SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // 使图标在工作区矩形中居中
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // 绘制图标
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialogEx::OnPaint();
    }
}

//当用户拖动最小化窗口时系统调用此函数取得光标
//显示。
HCURSOR CMFCApplication3Dlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

void CMFCApplication3Dlg::OnEnChangeEdit5()
{
    // TODO: 如果该控件是 RICHEDIT 控件，它将不
    // 发送此通知，除非重写 CDialogEx::OnInitDialog()
    // 函数并调用 CRichEditCtrl().SetEventMask(),
    // 同时将 ENM_CHANGE 标志“或”运算到掩码中。

```

```

// TODO: 在此添加控件通知处理程序代码
}

void CMFCApplication3Dlg::OnBnClickedButton1() //导入成绩
{
    // TODO: 在此添加控件通知处理程序代码
    CString stu_name;
    GetDlgItemText(IDC_EDIT2, stu_name); //获取编辑框内容

    CString stu_num;
    GetDlgItemText(IDC_EDIT3, stu_num); //获取编辑框内容
    CString usual_exam;
    GetDlgItemText(IDC_EDIT4, usual_exam); //获取编辑框内容
    CString final_exam;
    GetDlgItemText(IDC_EDIT5, final_exam); //获取编辑框内容

    if (stu_name == "" && stu_num == "" && usual_exam == "" && final_exam == "") {
//边界判断
        MessageBox(TEXT("■ 输入框内容未填写完毕!"), TEXT("提示"), MB_OK);
        return;
    }

    //注意如何将CSting转为float
    Student tmp_stu(stu_num, stu_name, atof(CT2A(usual_exam.GetBuffer()))),
    atof(CT2A(final_exam.GetBuffer())));
    stu.push_back(tmp_stu);

    MessageBox(TEXT("导入成功! "), TEXT("提示"), MB_OK); //弹出信息框

    //清空编辑框
    CString text3;
    text3 = "";
    GetDlgItem(IDC_EDIT2)->SetWindowText(text3); //设置编辑框内容
    GetDlgItem(IDC_EDIT3)->SetWindowText(text3); //设置编辑框内容
    GetDlgItem(IDC_EDIT4)->SetWindowText(text3); //设置编辑框内容
    GetDlgItem(IDC_EDIT5)->SetWindowText(text3); //设置编辑框内容
}

void CMFCApplication3Dlg::OnBnClickedButton4()
{
    // TODO: 在此添加控件通知处理程序代码
    CString text;

```

```

GetDlgItemText(IDC_EDIT6, text);    //获取编辑框内容

if (text == "" ) {                //边界判断
    MessageBox(TEXT("■ 还未填写要查询人的学号!"), TEXT("提示"), MB_OK);
    return;
}

int isfind = 0;    //用来记录是否找到
for (int i = 0; i < stu.size(); i++) {
    if (stu[i].stu_num == text ) {
        CString tmp;
        //注意这里加个_T
        tmp.Format(_T("⊕姓名:%s\n⊕学号:%s\n⊕平时成绩:%.2lf\n⊕期末成绩:%.2lf\n⊕总评成绩:%.2lf"), stu[i].stu_name, stu[i].stu_num, stu[i].usual_exam, stu[i].final_exam, stu[i].total_exam);
        MessageBox(tmp, TEXT("成绩单"), MB_OK);
        isfind = 1;
        return;
    }
}

if (isfind == 0) {
    MessageBox(TEXT("■ 未在成绩表中找到该学号!"), TEXT("提示"), MB_OK);
}

return;
}

void CMFCApplication3Dlg::OnBnClickedButton5()    //删除
{
    // TODO: 在此添加控件通知处理程序代码

    CString text;
    GetDlgItemText(IDC_EDIT6, text);    //获取编辑框内容
    if (text == "") {                //边界判断
        MessageBox(TEXT("■ 还未填写要删除人的学号!"), TEXT("提示"), MB_OK);
        return;
    }

    int isfind = 0;
    if (stu.size() == 0) {                //边界判断
        MessageBox(TEXT("■ 当前成绩单为空,无内容可以删除!"), TEXT("提示"), MB_OK);
        return;
    }
    for (int i = 0; i < stu.size(); i++) {
        if ( stu[i].stu_num == text) {
            isfind = 1;

```

```

        if ( MessageBox(TEXT("● 请确认是否删除? "), TEXT("提示"), MB_YESNO) == IDYES ) {
            stu.erase(stu.begin() + i);
        }
        else {
            MessageBox(TEXT("■ 删除失败! "), TEXT("提示"), MB_OK);
        }
    }
}

if (isfind == 0) {
    cout << "■ 未在成绩表中找到该学号! " << endl;
}

return;
}

bool compare(Student x, Student y) {
    string a = CT2A(x.stu_num.GetBuffer());
    string b = CT2A(y.stu_num.GetBuffer());
    return atoi(a.c_str()) < atoi(b.c_str());
}

void CMFCApplication3Dlg::OnBnClickedButton7()           // 查看成绩
{
    // TODO: 在此添加控件通知处理程序代码
    if (stu.size() == 0) {                               // 边界判断
        MessageBox(TEXT("■ 当前成绩单为空,无内容可以查看!"), TEXT("提示"), MB_OK);
        return;
    }

    // 注意这里的排序
    sort(stu.begin(), stu.end(), compare);

    CString tmp;
    for (int i = 0; i < stu.size(); i++) {
        // 注意这里加个_T
        tmp.Format(tmp+_T("-----\n⊕姓名:%s\n⊕学号:%s\n⊕平时成绩:%.2lf\n⊕期末成绩:%.2lf\n⊕总评成绩:%.2lf\n"), stu[i].stu_name, stu[i].stu_num, stu[i].usual_exam, stu[i].final_exam, stu[i].total_exam);
    }
    MessageBox(tmp, TEXT("成绩单"), MB_OK);
}

void CMFCApplication3Dlg::OnBnClickedButton8()           // 查看成绩分析

```



```

{
    // TODO: 在此添加控件通知处理程序代码
    if (stu.size() == 0) { //边界判断
        MessageBox(TEXT("■ 当前成绩单为空,无内容可以分析!"), TEXT("提示"), MB_OK);
        return;
    }

    double all = 0;
    unordered_map<int, int> freq; //各个成绩段的人数和比例
    for (int i = 0; i < stu.size(); i++) {
        double tmp = stu[i].total_exam;
        freq[int(tmp / 10)]++;
        all += tmp;
    }

    CString tmp;
    for (int i = 0; i < stu.size(); i++) {
        //注意这里加个_T
        tmp.Format(tmp + _T("-----\n姓名:%s\n学号:%s\n平时成绩:%.2lf\n期末成绩:%.2lf\n总评成绩:%.2lf\n"), stu[i].stu_name, stu[i].stu_num, stu[i].usual_exam, stu[i].final_exam, stu[i].total_exam);
    }

    tmp.Format(_T("★ 各分数段的人数及比例\n"));
    tmp.Format(tmp + _T("⊕ 90 - 100分:%d人, 比例是%.2lf\n"), freq[10] + freq[9], (freq[10] + freq[9]) / double(stu.size()));
    tmp.Format(tmp + _T("⊕ 80 - 89分 :%d人, 比例是%.2lf\n"), freq[8], (freq[8]) / double(stu.size()));
    tmp.Format(tmp + _T("⊕ 70 - 79分 :%d人, 比例是%.2lf\n"), freq[7], (freq[7]) / double(stu.size()));
    tmp.Format(tmp + _T("⊕ 60 - 69分 :%d人, 比例是%.2lf\n"), freq[6], (freq[6]) / double(stu.size()));
    tmp.Format(tmp + _T("⊕ 60分以下 :%d人, 比例是%.2lf\n"), freq[5] + freq[4] + freq[3] + freq[2] + freq[1] + freq[0], (freq[5] + freq[4] + freq[3] + freq[2] + freq[1] + freq[0]) / double(stu.size()));
    tmp.Format(tmp + _T("★ 该课程的平均成绩: %.2lf\n"), all / stu.size());

    MessageBox(tmp, TEXT("成绩分析"), MB_OK);

    return;
}

void CMFCApplication3Dlg::OnBnClickedButton6() //进入修改
{
    // TODO: 在此添加控件通知处理程序代码

```

```

CString text;
GetDlgItemText(IDC_EDIT7, text);    //获取编辑框内容
if (text == "") {                  //边界判断
    MessageBox(TEXT("■ 还未填写要修改人的学号!"), TEXT("提示"), MB_OK);
    return;
}

int isfind = 0;
if (stu.size() == 0) {             //边界判断
    MessageBox(TEXT("■ 当前成绩单为空,无内容可以修改!"), TEXT("提示"), MB_OK);
    return;
}
for (int i = 0; i < stu.size(); i++) {
    if (stu[i].stu_num == text) {
        isfind = 1;
        //启用下面的控件
        CWnd* pWnd = GetDlgItem(IDC_EDIT10);
        pWnd->EnableWindow(TRUE);
        pWnd = GetDlgItem(IDC_EDIT11);
        pWnd->EnableWindow(TRUE);
        pWnd = GetDlgItem(IDC_EDIT12);
        pWnd->EnableWindow(TRUE);
        pWnd = GetDlgItem(IDC_BUTTON9);
        pWnd->EnableWindow(TRUE);
        break;
    }
}
if (isfind == 0) {
    MessageBox(TEXT("■ 该学号不存在,无法进行修改!"), TEXT("提示"), MB_OK);
}
return;
}

void CMFCApplication3Dlg::OnBnClickedButton9()    //确认修改
{
    // TODO: 在此添加控件通知处理程序代码
    CString text;
    GetDlgItemText(IDC_EDIT7, text);    //获取编辑框内容

    int isfind = 0;
    if (stu.size() == 0) {             //边界判断
        MessageBox(TEXT("■ 当前成绩单为空,无内容可以修改!"), TEXT("提示"), MB_OK);
    }
}

```

```

        return;
    }
    CString stu_name;
    GetDlgItemText(IDC_EDIT10, stu_name);    //获取编辑框内容
    CString stu_num;
    GetDlgItemText(IDC_EDIT7, stu_num);      //获取编辑框内容
    CString usual_exam;
    GetDlgItemText(IDC_EDIT11, usual_exam); //获取编辑框内容
    CString final_exam;
    GetDlgItemText(IDC_EDIT12, final_exam); //获取编辑框内容
    if (stu_name == "" && stu_num == "" && usual_exam == "" && final_exam == "" ) {
//边界判断
        MessageBox(TEXT("❌ 输入框内容未填写完毕!"), TEXT("提示"), MB_OK);
        return;
    }
    for (int i = 0; i < stu.size(); i++) {
        if (stu[i].stu_num == text) {

            //注意如何将CString转为float
            stu[i].modify(stu_num, stu_name, atof(CT2A(usual_exam.GetBuffer())) ,
atof(CT2A(final_exam.GetBuffer())));

            MessageBox(TEXT("修改成功! "), TEXT("提示"), MB_OK);                //弹出信息框

            //清空编辑框
            CString text3;
            text3 = "";
            GetDlgItem(IDC_EDIT7)->SetWindowText(text3); //设置编辑框内容
            GetDlgItem(IDC_EDIT10)->SetWindowText(text3); //设置编辑框内容
            GetDlgItem(IDC_EDIT11)->SetWindowText(text3); //设置编辑框内容
            GetDlgItem(IDC_EDIT12)->SetWindowText(text3); //设置编辑框内容

            //禁用下面的控件
            CWnd* pWnd = GetDlgItem(IDC_EDIT10);
            pWnd->EnableWindow(FALSE);
            pWnd = GetDlgItem(IDC_EDIT11);
            pWnd->EnableWindow(FALSE);
            pWnd = GetDlgItem(IDC_EDIT12);
            pWnd->EnableWindow(FALSE);
            pWnd = GetDlgItem(IDC_BUTTON9);
            pWnd->EnableWindow(FALSE);
        }
    }
}

```

```
}  
}
```

直接将黑窗口的关键代码移植到MFC中，即可  
关键：

```
1.CString tmp;  
//注意这里加个_T  
tmp.Format(_T("  
2.
```

(2021.6.13)

```
/* 版本： 2.0  
 * 时间： 2021.6.13  
 * 作者： orall  
 * 完善： 写了gui界面  
 */  
  
// MFCApplication3Dlg.cpp：实现文件  
//  
  
#include "pch.h"  
#include "framework.h"  
#include "MFCApplication3.h"  
#include "MFCApplication3Dlg.h"  
#include "afxdialogex.h"  
  
#include <iostream>  
#include <windows.h>  
#include <algorithm>  
#include <cstdlib>  
#include <string>  
#include <ctime>  
#include <vector>  
#include <unordered_map>  
#include <sstream>  
using namespace std;  
  
#ifdef _DEBUG  
#define new DEBUG_NEW  
#endif  
  
typedef struct str_stu {
```

```

        CString stu_num;        //学号
        CString stu_name;       //姓名
        int usual_exam;        //平时成绩
        int final_exam;        //期末成绩
    }Str_stu;

class Student {
public:
    CString stu_name;           //姓名
    double usual_exam;         //平时成绩
    double final_exam;         //期末成绩
    CString stu_num;            //学号
    double total_exam;          //总评成绩
    Student(CString num, CString name, double us_ex, double fi_ex) :stu_name(name),
stu_num(num), usual_exam(us_ex), final_exam(fi_ex)
    {
        total_exam = usual_exam * 0.4 + final_exam * 0.6;
    }
    void modify(CString num, CString name, double us_ex, double fi_ex)
    {
        stu_name = name;
        stu_num = num;
        usual_exam = us_ex;
        final_exam = fi_ex;
        total_exam = int((double)usual_exam * 0.4 + (double)final_exam * 0.6);
    }
    void show() {

        //cout << "★" << endl;
    }
};

vector<Student> stu;

// 用于应用程序“关于”菜单项的 CAboutDlg 对话框

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

    // 对话框数据
#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_ABOUTBOX };
#endif

```

```

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(IDD_ABOUTBOX)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

// CMFCApplication3Dlg 对话框

CMFCApplication3Dlg::CMFCApplication3Dlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_MFCAPPLICATION3_DIALOG, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CMFCApplication3Dlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CMFCApplication3Dlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_EN_CHANGE(IDC_EDIT5, &CMFCApplication3Dlg::OnEnChangeEdit5)
    ON_BN_CLICKED(IDC_BUTTON1, &CMFCApplication3Dlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON4, &CMFCApplication3Dlg::OnBnClickedButton4)
    ON_BN_CLICKED(IDC_BUTTON5, &CMFCApplication3Dlg::OnBnClickedButton5)
    ON_BN_CLICKED(IDC_BUTTON7, &CMFCApplication3Dlg::OnBnClickedButton7)

```

```

        ON_BN_CLICKED(IDC_BUTTON8, &CMFCApplication3Dlg::OnBnClickedButton8)
        ON_BN_CLICKED(IDC_BUTTON6, &CMFCApplication3Dlg::OnBnClickedButton6)
        ON_BN_CLICKED(IDC_BUTTON9, &CMFCApplication3Dlg::OnBnClickedButton9)
    END_MESSAGE_MAP()

// CMFCApplication3Dlg 消息处理程序

BOOL CMFCApplication3Dlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // 将“关于...”菜单项添加到系统菜单中。

    // IDM_ABOUTBOX 必须在系统命令范围内。
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != nullptr)
    {
        BOOL bNameValid;
        CString strAboutMenu;
        bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // 设置此对话框的图标。 当应用程序主窗口不是对话框时，框架将自动
    // 执行此操作
    SetIcon(m_hIcon, TRUE);        // 设置大图标
    SetIcon(m_hIcon, FALSE);       // 设置小图标

    // TODO: 在此添加额外的初始化代码

    return TRUE;  // 除非将焦点设置到控件，否则返回 TRUE
}

void CMFCApplication3Dlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {

```

```

        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

// 如果向对话框添加最小化按钮，则需要下面的代码
// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，
// 这将由框架自动完成。

void CMFCApplication3Dlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 用于绘制的设备上下文

        SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // 使图标在工作区矩形中居中
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // 绘制图标
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialogEx::OnPaint();
    }
}

//当用户拖动最小化窗口时系统调用此函数取得光标
//显示。
HCURSOR CMFCApplication3Dlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

```



```

void CMFCApplication3Dlg::OnEnChangeEdit5()
{
    // TODO: 如果该控件是 RICHEDIT 控件，它将不
    // 发送此通知，除非重写 CDialogEx::OnInitDialog()
    // 函数并调用 CRichEditCtrl().SetEventMask(),
    // 同时将 ENM_CHANGE 标志“或”运算到掩码中。

    // TODO: 在此添加控件通知处理程序代码
}

void CMFCApplication3Dlg::OnBnClickedButton1() // 导入成绩
{
    // TODO: 在此添加控件通知处理程序代码
    CString stu_name;
    GetDlgItemText(IDC_EDIT2, stu_name); // 获取编辑框内容

    CString stu_num;
    GetDlgItemText(IDC_EDIT3, stu_num); // 获取编辑框内容
    CString usual_exam;
    GetDlgItemText(IDC_EDIT4, usual_exam); // 获取编辑框内容
    CString final_exam;
    GetDlgItemText(IDC_EDIT5, final_exam); // 获取编辑框内容

    if (stu_name == "" && stu_num == "" && usual_exam == "" && final_exam == "") {
// 边界判断
        MessageBox(TEXT("❌ 输入框内容未填写完毕!"), TEXT("提示"), MB_OK);
        return;
    }

    // 注意如何将CString转为float
    Student tmp_stu(stu_num, stu_name, atof(CT2A(usual_exam.GetBuffer()))),
    atof(CT2A(final_exam.GetBuffer())));
    stu.push_back(tmp_stu);

    MessageBox(TEXT("导入成功! "), TEXT("提示"), MB_OK); // 弹出信息框

    // 清空编辑框
    CString text3;
    text3 = "";
    GetDlgItem(IDC_EDIT2)->SetWindowText(text3); // 设置编辑框内容
    GetDlgItem(IDC_EDIT3)->SetWindowText(text3); // 设置编辑框内容
    GetDlgItem(IDC_EDIT4)->SetWindowText(text3); // 设置编辑框内容
    GetDlgItem(IDC_EDIT5)->SetWindowText(text3); // 设置编辑框内容
}

```

```

}

void CMFCApplication3Dlg::OnBnClickedButton4()
{
    // TODO: 在此添加控件通知处理程序代码
    CString text;
    GetDlgItemText(IDC_EDIT6, text);    //获取编辑框内容

    if (text == "" ) {                //边界判断
        MessageBox(TEXT("■ 还未填写要查询人的学号!"), TEXT("提示"), MB_OK);
        return;
    }

    int isfind = 0;    //用来记录是否找到
    for (int i = 0; i < stu.size(); i++) {
        if (stu[i].stu_num == text ) {
            CString tmp;
            //注意这里加个_T
            tmp.Format(_T("⊕姓名:%s\n⊕学号:%s\n⊕平时成绩:%.2lf\n⊕期末成绩:%.2lf\n⊕总评成绩:%.2lf"), stu[i].stu_name, stu[i].stu_num, stu[i].usual_exam, stu[i].final_exam, stu[i].total_exam);
            MessageBox(tmp, TEXT("成绩单"), MB_OK);
            isfind = 1;
            return;
        }
    }

    if (isfind == 0) {
        MessageBox(TEXT("■ 未在成绩表中找到该学号! "), TEXT("提示"), MB_OK);
    }
    return;
}

void CMFCApplication3Dlg::OnBnClickedButton5()    //删除
{
    // TODO: 在此添加控件通知处理程序代码

    CString text;
    GetDlgItemText(IDC_EDIT6, text);    //获取编辑框内容
    if (text == "") {                //边界判断
        MessageBox(TEXT("■ 还未填写要删除人的学号!"), TEXT("提示"), MB_OK);
        return;
    }

    int isfind = 0;
    if (stu.size() == 0) {            //边界判断

```

```

        MessageBox(TEXT("■ 当前成绩单为空,无内容可以删除!"), TEXT("提示"), MB_OK);
        return;
    }
    for (int i = 0; i < stu.size(); i++) {
        if ( stu[i].stu_num == text) {
            isfind = 1;

            if ( MessageBox(TEXT("● 请确认是否删除? "), TEXT("提示"), MB_YESNO) == IDYES ) {
                stu.erase(stu.begin() + i);
            }
            else {
                MessageBox(TEXT("■ 删除失败! "), TEXT("提示"), MB_OK);
            }
        }
    }
}

if (isfind == 0) {
    cout << "■ 未在成绩表中找到该学号! " << endl;
}

return;
}

void CMFCApplication3Dlg::OnBnClickedButton7() //查看成绩
{
    // TODO: 在此添加控件通知处理程序代码
    if (stu.size() == 0) { //边界判断
        MessageBox(TEXT("■ 当前成绩单为空,无内容可以查看!"), TEXT("提示"), MB_OK);
        return;
    }
    CString tmp;
    for (int i = 0; i < stu.size(); i++) {
        //注意这里加个_T
        tmp.Format(tmp+_T("-----\n⊕姓名:%s\n⊕学号:%s\n⊕平时成绩:%.2lf\n⊕期末成绩:%.2lf\n⊕总评成绩:%.2lf\n"), stu[i].stu_name, stu[i].stu_num, stu[i].usual_exam, stu[i].final_exam, stu[i].total_exam);
    }
    MessageBox(tmp, TEXT("成绩单"), MB_OK);
}

void CMFCApplication3Dlg::OnBnClickedButton8() //查看成绩分析
{
    // TODO: 在此添加控件通知处理程序代码

```

```

        if (stu.size() == 0) { //边界判断
            MessageBox(TEXT("❌ 当前成绩单为空,无内容可以分析!"), TEXT("提示"), MB_OK);
            return;
        }

        double all = 0;
        unordered_map<int, int> freq; //各个成绩段的人数和比例
        for (int i = 0; i < stu.size(); i++) {
            double tmp = stu[i].total_exam;
            freq[int(tmp / 10)]++;
            all += tmp;
        }

        CString tmp;
        for (int i = 0; i < stu.size(); i++) {
            //注意这里加个_T
            tmp.Format(tmp + _T("-----\n⊕姓名:%s\n⊕学号:%s\n⊕平时成绩:%.2lf\n⊕期末成绩:%.2lf\n⊕总评成绩:%.2lf\n"), stu[i].stu_name, stu[i].stu_num, stu[i].usual_exam, stu[i].final_exam, stu[i].total_exam);
        }

        tmp.Format(_T("★ 各分数段的人数及比例\n"));
        tmp.Format(tmp + _T("⊕ 90 - 100分:%d人, 比例是%.2lf\n"), freq[10] + freq[9], (freq[10] + freq[9]) / double(stu.size()));
        tmp.Format(tmp + _T("⊕ 80 - 89分 :%d人, 比例是%.2lf\n"), freq[8], (freq[8]) / double(stu.size()));
        tmp.Format(tmp + _T("⊕ 70 - 79分 :%d人, 比例是%.2lf\n"), freq[7], (freq[7]) / double(stu.size()));
        tmp.Format(tmp + _T("⊕ 60 - 69分 :%d人, 比例是%.2lf\n"), freq[6], (freq[6]) / double(stu.size()));
        tmp.Format(tmp + _T("⊕ 60分以下 :%d人, 比例是%.2lf\n"), freq[5] + freq[4] + freq[3] + freq[2] + freq[1] + freq[0], (freq[5] + freq[4] + freq[3] + freq[2] + freq[1] + freq[0]) / double(stu.size()));
        tmp.Format(tmp + _T("★ 该课程的平均成绩: %.2lf\n"), all / stu.size());

        MessageBox(tmp, TEXT("成绩分析"), MB_OK);

        return;
    }

void CMFCApplication3Dlg::OnBnClickedButton6() //进入修改
{
    // TODO: 在此添加控件通知处理程序代码

    CString text;

```

```

GetDlgItemText(IDC_EDIT7, text);    //获取编辑框内容
if (text == "") {                  //边界判断
    MessageBox(TEXT("■ 还未填写要修改人的学号!"), TEXT("提示"), MB_OK);
    return;
}

int isfind = 0;
if (stu.size() == 0) {             //边界判断
    MessageBox(TEXT("■ 当前成绩单为空,无内容可以修改!"), TEXT("提示"), MB_OK);
    return;
}
for (int i = 0; i < stu.size(); i++) {
    if (stu[i].stu_num == text) {
        isfind = 1;
        //启用下面的控件
        CWnd* pWnd = GetDlgItem(IDC_EDIT10);
        pWnd->EnableWindow(TRUE);
        pWnd = GetDlgItem(IDC_EDIT11);
        pWnd->EnableWindow(TRUE);
        pWnd = GetDlgItem(IDC_EDIT12);
        pWnd->EnableWindow(TRUE);
        pWnd = GetDlgItem(IDC_BUTTON9);
        pWnd->EnableWindow(TRUE);
        break;
    }
}
if (isfind == 0) {
    cout << "■ 未在成绩表中找到该学号! " << endl;
}
return;
}

void CMFCApplication3Dlg::OnBnClickedButton9()    //确认修改
{
    // TODO: 在此添加控件通知处理程序代码
    CString text;
    GetDlgItemText(IDC_EDIT7, text);    //获取编辑框内容

    int isfind = 0;
    if (stu.size() == 0) {             //边界判断
        MessageBox(TEXT("■ 当前成绩单为空,无内容可以修改!"), TEXT("提示"), MB_OK);
        return;
    }

```

```

        CString stu_name;
        GetDlgItemText(IDC_EDIT10, stu_name);        //获取编辑框内容
        CString stu_num;
        GetDlgItemText(IDC_EDIT7, stu_num);          //获取编辑框内容
        CString usual_exam;
        GetDlgItemText(IDC_EDIT11, usual_exam);     //获取编辑框内容
        CString final_exam;
        GetDlgItemText(IDC_EDIT12, final_exam);     //获取编辑框内容
        if (stu_name == "" && stu_num == "" && usual_exam == "" && final_exam == "" ) {
//边界判断
            MessageBox(TEXT("❌ 输入框内容未填写完毕!"), TEXT("提示"), MB_OK);
            return;
        }
        for (int i = 0; i < stu.size(); i++) {
            if (stu[i].stu_num == text) {

                //注意如何将CString转为float
                stu[i].modify(stu_num, stu_name, atof(CT2A(usual_exam.GetBuffer()))),
                atof(CT2A(final_exam.GetBuffer())));

                MessageBox(TEXT("修改成功! "), TEXT("提示"), MB_OK);                //弹出信息框

                //清空编辑框
                CString text3;
                text3 = "";
                GetDlgItem(IDC_EDIT7)->SetWindowText(text3); //设置编辑框内容
                GetDlgItem(IDC_EDIT10)->SetWindowText(text3); //设置编辑框内容
                GetDlgItem(IDC_EDIT11)->SetWindowText(text3); //设置编辑框内容
                GetDlgItem(IDC_EDIT12)->SetWindowText(text3); //设置编辑框内容

                //禁用下面的控件
                CWnd* pWnd = GetDlgItem(IDC_EDIT10);
                pWnd->EnableWindow(FALSE);
                pWnd = GetDlgItem(IDC_EDIT11);
                pWnd->EnableWindow(FALSE);
                pWnd = GetDlgItem(IDC_EDIT12);
                pWnd->EnableWindow(FALSE);
                pWnd = GetDlgItem(IDC_BUTTON9);
                pWnd->EnableWindow(FALSE);
            }
        }
    }
}

```

黑窗口是根据前面写的学生通讯录改编，使用vector容器可以大大简化代码

关键点为：

1.删除某一元素的代码：`stu.erase(stu.begin() + i);`

(2021.6.13)

```
/* 版本：学生成绩单管理系统1.0
 * 时间：2021.6.13
 * 作者：orall
 * */

#include <iostream>
#include <windows.h>
#include <algorithm>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <unordered_map>

using namespace std;

typedef struct str_stu {
    string stu_num;    //学号
    string stu_name;   //姓名
    int usual_exam;   //平时成绩
    int final_exam;   //期末成绩
}Str_stu;

class Student {
private:
    string stu_name;    //姓名
    double usual_exam; //平时成绩
    double final_exam; //期末成绩

public:
    string stu_num;    //学号
    double total_exam; //总评成绩
    Student(string num, string name, double us_ex, double fi_ex) :stu_name(name),
stu_num(num), usual_exam(us_ex), final_exam(fi_ex)
    {
        total_exam = usual_exam * 0.4 +final_exam * 0.6;
    }
    void modify(string num, string name, double us_ex, double fi_ex)
    {
        stu_name = name;
        stu_num = num;
        usual_exam = us_ex;
```

```

        final_exam = fi_ex;
        total_exam = int((double)usual_exam * 0.4 + (double)final_exam * 0.6);
    }
    void show() {
        cout << "★" << endl;
        cout << "⊕姓名 " << stu_name << "\n⊕学号 " << stu_num << "\n⊕平时成绩 " << usual_exam
<< "\n⊕期末成绩 " << final_exam << "\n⊕总评成绩 " << total_exam << endl << endl;
    }
};

vector<Student> stu;           //等价于链表

void mycreat();               //创建成绩单
void mydelete(string stu_num);
void mydisplay();            //显示成绩单
void statistics();
void mysearch(string num);    //查找
void mymodify(string num);    //修改成绩单
void myprint();              //显示主菜单

int main() {
    // system("chcp 936");
    // system("chcp 936");           //避免乱码(自己打开的是65001,别人的是
936)
    system("title 华师学生成绩单管理系统   ");           //设置标题
    system("mode con: cols=35 lines=25");           //设置窗口大小
    system("color 79");           //设置颜色和大小
    while (1) {
        myprint();
    }
}

void mycreat() {
    // ★■□○▬⊕▲▼●
    Str_stu tmp;           //结构体作为变量
    cout << "======" << endl;
    cout << "● 姓名是: ";
    cin >> tmp.stu_name;
    cout << "● 学号是: ";
    cin >> tmp.stu_num;
    cout << "● 平时成绩是: ";
    cin >> tmp.usual_exam;
    cout << "● 期末成绩是: ";
    cin >> tmp.final_exam;
    cout << "======" << endl;
    Student tmp_stu(tmp.stu_num,tmp.stu_name,tmp.usual_exam,tmp.final_exam);
}

```



```

        stu.push_back(tmp_stu);
        return;
    }
//添加通讯录
void myadd() {
    if ( stu.size() == 0 ) {                //边界判断
        cout << "❌ 当前成绩单为空，无法进行添加！" << endl;
        return;
    }
    mycreat();
}

//删除通讯录
void mydelete(string num) {
    int isfind = 0;
    if ( stu.size() == 0 ) {                //边界判断
        cout << "❌ 当前成绩单为空,无内容可以删除!" << endl;
        return;
    }
    for (int i = 0; i < stu.size(); i++) {
        if (stu[i].stu_num == num ) {
            isfind = 1;
            int tmp;
            cout << "● 请确认是否删除【y/n】:";
            cin >> tmp;
            if ( tmp == 'y' ) {
                stu.erase(stu.begin() + i);
                cout << "◆ 正在删除.....";
                Sleep(2000);
            }
            else {
                cout << "❌ 删除失败! ";
            }
        }
    }

    if (isfind == 0) {
        cout << "❌ 未在成绩表中找到该学号!" << endl;
    }
    return;
}

bool compare(Student x, Student y) { return atoi(x.stu_num.c_str()) <
atoi(y.stu_num.c_str()); }

//显示

```

```

void mydisplay() {
    if (stu.size() == 0) {
        cout << "■ 当前成绩表为空" << endl;
        return;
    }

    //注意这里的排序
    sort(stu.begin(), stu.end(),compare);

    for (int i = 0; i < stu.size(); i++) {
        stu[i].show();
    }
}

//根据学号查找
void mysearch(string num) {
    int isfind = 0;    //用来记录是否找到
    for (int i = 0; i < stu.size(); i++) {
        if (stu[i].stu_num == num ) {
            stu[i].show();
            isfind = 1;
            return;
        }
    }

    if (isfind == 0) {
        cout << "■ 未在成绩表中找到该学号!" << endl;
    }
    return;
}

//修改成绩单
void mymodify(string num) {
    int isfind = 0;    //用来记录是否找到
    for (int i = 0; i < stu.size(); i++) {
        if (stu[i].stu_num == num ) {
            stu[i].show();
            isfind = 1;

            Str_stu tmp;    //结构体作为变量
            tmp.stu_num = stu[i].stu_num;
            cout << "*****请输入对应修改的内容*****" << endl;

            cout << "● 姓名是: ";
            cin >> tmp.stu_name;
            cout << "● 平时成绩是: ";

```

```

        cin >> tmp.usual_exam;
        cout << "● 期末成绩是: ";
        cin >> tmp.final_exam;
        cout << "======" << endl;
        stu[i].modify(tmp.stu_num,tmp.stu_name,tmp.usual_exam,tmp.final_exam);
        return;
    }

}

}

void statistics() {
    double all = 0 ;
    unordered_map<int, int> freq;           //各个成绩段的人数和比例
    for (int i = 0; i < stu.size(); i++) {
        double tmp = stu[i].total_exam;
        freq[int(tmp / 10)]++;
        all += tmp;
    }
    cout << "★ 各分数段的人数及比例" << endl;
    cout << "⊕ 90-100分:" << freq[10] + freq[9] << "人, 比例是" << (freq[10] + freq[9]) /
double(stu.size())<<endl;
    cout << "⊕ 80-89分:" << freq[8] << "人, 比例是" << (freq[8]) / double(stu.size()) <<
endl;
    cout << "⊕ 70-79分:" << freq[7] << "人, 比例是" << (freq[7]) / double(stu.size()) <<
endl;
    cout << "⊕ 60-69分:" << freq[6] << "人, 比例是" << (freq[6]) / double(stu.size()) <<
endl;
    cout << "⊕ 60分以下:" << freq[5]+ freq[4]+ freq[3]+ freq[2] + freq[1] + freq[0] << "人,
比例是" << (freq[5] + freq[4] + freq[3] + freq[2] + freq[1] + freq[0]) / double(stu.size())
<< endl;
    cout << "★ 该课程的平均成绩: " << all/stu.size() << endl;
    return;
}

//主菜单
void myprint() {
    cin.clear();           //清空缓冲区, 避免闪屏
    cout << "=====学生成绩单管理系统=====\\n"
        "          1. 录入学生信息\\n"
        "          2. 添加学生信息\\n"
        "          3. 查询学生成绩\\n"
        "          4. 修改学生信息\\n"
        "          5. 删除学生信息\\n"

```

```

        "        6. 显示学生成绩单\n"
        "        7. 统计和分析成绩\n"
        "        8. 退出\n\n";

int choose, num;
cout << "■ 请选择:";
cin >> choose;
system("cls");          //清屏
cout << "=====学生成绩单管理系统=====\n";
string tmp;
switch (choose) {
case 1:                  // 1. 录入学生信息
    cout << "● 正在创建成绩单，请输入需要录入的人数: ";
    cin >> num;
    for (int i = 0; i < num; i++) {
        mycreat();
    }
    cout << "★ 创建完毕，正在保存.....";
    Sleep(2000);
    break;
case 2:                  // 2. 添加学生信息
    mycreat();
    system("pause");
    break;
case 3:                  //3. 查询学生成绩
    cout << "● 请输入要查询的学号:";
    cin >> tmp;
    mysearch(tmp);
    system("pause");
    break;
case 4:                  //4. 修改学生信息
    cout << "● 请输入要修改信息的学生学号:";
    cin >> tmp;
    mymodify(tmp);
    cout << "★ 修改完毕，正在保存.....";
    Sleep(2000);
    break;
case 5:                  //5. 删除学生信息
    cout << "● 请输入要删除信息的学生学号:";
    cin >> tmp;
    mydelete(tmp);

    break;

case 6:                  //6. 显示学生成绩单
    mydisplay();
    system("pause");

```

```
        break;

    case 7:                //7. 统计和分析成绩
        statistics();
        system("pause");
        break;
    case 8:                //退出
        exit(0);
        break;
    default:
        cout << "❌ 输入错误，请重新输入！ " << endl;
        system("pause");
        break;
}
system("cls");            //退出前清屏
// cin.clear();           //清空缓冲区，避免闪屏
return;
}
```