**程序设计课程设计实验报告**

****

**学院： 计算机科学与教育软件学院**

**专业班级： 软件工程 185**

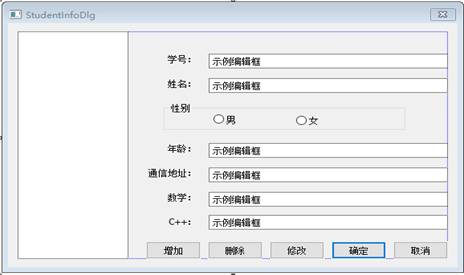
**姓名： 曾祥毅**

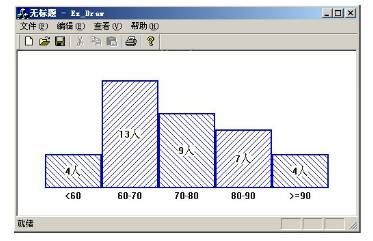
**学号： 1806300153**

**2019.06.21**

**一、课程设计题目**

应用MFC应用程序实现学生信息管理系统

（1）、设计一个学生类Student,包括数据成员：姓名、学号、二门课程(面向对象程序设计、高等数学)的成绩。（2）、创建一个管理学生的类Management，包括实现学生的数据的增加、删除、修改、按课程成绩排序、保存学生数据到文件及加载文件中的数据等功能。（3）、创建一个基于对话框的MFC应用程序，程序窗口的标题上有你姓名、学号和应用程序名称。使用（1）和（2）中的类，实现对学生信息和成绩的输入和管理。

（4）、创建一个单文档的MFC应用程序，读取（3）中保存的文件中的学生成绩，分别用直方图和折线方式显示所有学生某课程的成绩分布图。

**二、程序设计基本步骤**

1.分析需求，了解程序所需的功能；

2.根据所需功能构思、设计相应算法；

3.创建工程，添加所需类、变量、函数等；

4.编写程序；

5.调试、测试；

6.生成、导出程序；

**三、自定义类及函数功能概括**

// CCourceDesignDlg 对话框

class CCourceDesignDlg : public CDialogEx

{

// 构造

public:

CCourceDesignDlg(CWnd\* pParent = nullptr); // 标准构造函数

// 对话框数据

#ifdef AFX\_DESIGN\_TIME

enum { IDD = IDD\_COURCEDESIGN\_DIALOG };

#endif

protected:

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

// 实现

protected:

HICON m\_hIcon;

// 生成的消息映射函数

virtual BOOL OnInitDialog();

afx\_msg void OnSysCommand(UINT nID, LPARAM lParam);

afx\_msg void OnPaint();

afx\_msg HCURSOR OnQueryDragIcon();

DECLARE\_MESSAGE\_MAP()

public:

afx\_msg void OnBnClickedButton1();//修改按钮

CString NUM;//学号输入框

CString NAME;//名字输入框

CString AGE;//年龄输入框

CString ADDRESS;//通讯地址输入框

CString MATH;//数学成绩输入框

CString CPP;//C++成绩输入框

CListBox LIST;//列表控件

CString SEX;//存储界面中选中性别的变量，1男，0女，2未选中

afx\_msg void OnBnClickedButton2();//增加按钮

afx\_msg void OnBnClickedRadio2();//选择男

afx\_msg void OnBnClickedRadio3();//选择女

afx\_msg void OnBnClickedButton5();//清空输入框按钮

afx\_msg void OnLbnSelchangeList1();//选中列表项的事件函数

CButton MALE;//男选择框控件变量

CButton FEMALE;//女选择框控件变量

afx\_msg void OnBnClickedOk();//另存为按钮

afx\_msg void OnBnClickedButton3();//删除按钮

afx\_msg void OnBnClickedButton6();//打开文件按钮

afx\_msg void OnBnClickedCancel();//退出按钮

afx\_msg void OnBnClickedButton4();

afx\_msg void OnBnClickedButton7();//保存按钮

afx\_msg void OnBnClickedButton8();//新建空白表按钮

};

//学生类

class student

{

public:

CString NUMM;//学号

CString NAMEE;//名字

CString AGEE;//年龄

CString ADDRESSS;//通讯地址

CString MATHH;//数学

CString CPPP;//C++成绩

CString SEXX;//性别

};

// AnalysisDoc.h: CAnalysisDoc 类的接口

//

#pragma once

class CAnalysisDoc : public CDocument

{

protected: // 仅从序列化创建

CAnalysisDoc() noexcept;

DECLARE\_DYNCREATE(CAnalysisDoc)

// 特性

public:

// 操作

public:

float\* CPP;//C++成绩的数据接口

float\* MATH;//数学成绩的数据接口

int len;//数组长度

// 重写

public:

virtual BOOL OnNewDocument();

virtual void Serialize(CArchive& ar);

#ifdef SHARED\_HANDLERS

virtual void InitializeSearchContent();

virtual void OnDrawThumbnail(CDC& dc, LPRECT lprcBounds);

#endif // SHARED\_HANDLERS

// 实现

public:

virtual ~CAnalysisDoc();

#ifdef \_DEBUG

virtual void AssertValid() const;

virtual void Dump(CDumpContext& dc) const;

#endif

protected:

// 生成的消息映射函数

protected:

DECLARE\_MESSAGE\_MAP()

#ifdef SHARED\_HANDLERS

// 用于为搜索处理程序设置搜索内容的 Helper 函数

void SetSearchContent(const CString& value);

#endif // SHARED\_HANDLERS

};

// AnalysisView.h: CAnalysisView 类的接口

//

#pragma once

class CAnalysisView : public CView

{

protected: // 仅从序列化创建

CAnalysisView() noexcept;

DECLARE\_DYNCREATE(CAnalysisView)

// 特性

public:

CAnalysisDoc\* GetDocument() const;

// 操作

public:

void DrawScore(CDC \*pDC,float \*fScore,int nNum);

// 重写

public:

virtual void OnDraw(CDC\* pDC); // 重写以绘制该视图

virtual BOOL PreCreateWindow(CREATESTRUCT& cs);

protected:

virtual BOOL OnPreparePrinting(CPrintInfo\* pInfo);

virtual void OnBeginPrinting(CDC\* pDC, CPrintInfo\* pInfo);

virtual void OnEndPrinting(CDC\* pDC, CPrintInfo\* pInfo);

// 实现

public:

virtual ~CAnalysisView();

#ifdef \_DEBUG

virtual void AssertValid() const;

virtual void Dump(CDumpContext& dc) const;

#endif

protected:

// 生成的消息映射函数

protected:

DECLARE\_MESSAGE\_MAP()

public:

afx\_msg void On32771();//绘制C++成绩直方图

afx\_msg void On32772();//绘制数学成绩直方图

};

1. **实验代码**

// CourceDesignDlg.cpp: 实现文件

//

#include "pch.h"

#include "framework.h"

#include "CourceDesign.h"

#include "CourceDesignDlg.h"

#include "afxdialogex.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#endif

student\* stu = new student[100];

int index = 0;

bool changed = FALSE;//列表变动标记

bool saved = FALSE;//列表文件关联标记

CString PathName;

// 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()

// CCourceDesignDlg 对话框

CCourceDesignDlg::CCourceDesignDlg(CWnd\* pParent /\*=nullptr\*/)

: CDialogEx(IDD\_COURCEDESIGN\_DIALOG, pParent)

, NUM(\_T(""))

, NAME(\_T(""))

, AGE(\_T(""))

, ADDRESS(\_T(""))

, MATH(\_T(""))

, CPP(\_T(""))

, SEX(\_T("2"))

{

m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);

}

void CCourceDesignDlg::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

DDX\_Text(pDX, IDC\_EDIT5, NUM);

DDX\_Text(pDX, IDC\_EDIT6, NAME);

DDX\_Text(pDX, IDC\_EDIT7, AGE);

DDX\_Text(pDX, IDC\_EDIT8, ADDRESS);

DDX\_Text(pDX, IDC\_EDIT9, MATH);

DDX\_Text(pDX, IDC\_EDIT10, CPP);

DDX\_Control(pDX, IDC\_LIST1, LIST);

DDX\_Control(pDX, IDC\_RADIO2, MALE);

DDX\_Control(pDX, IDC\_RADIO3, FEMALE);

}

BEGIN\_MESSAGE\_MAP(CCourceDesignDlg, CDialogEx)

ON\_WM\_SYSCOMMAND()

ON\_WM\_PAINT()

ON\_WM\_QUERYDRAGICON()

ON\_BN\_CLICKED(IDC\_BUTTON1, &CCourceDesignDlg::OnBnClickedButton1)

ON\_BN\_CLICKED(IDC\_BUTTON2, &CCourceDesignDlg::OnBnClickedButton2)

ON\_BN\_CLICKED(IDC\_RADIO2, &CCourceDesignDlg::OnBnClickedRadio2)

ON\_BN\_CLICKED(IDC\_RADIO3, &CCourceDesignDlg::OnBnClickedRadio3)

ON\_BN\_CLICKED(IDC\_BUTTON5, &CCourceDesignDlg::OnBnClickedButton5)

ON\_LBN\_SELCHANGE(IDC\_LIST1, &CCourceDesignDlg::OnLbnSelchangeList1)

ON\_BN\_CLICKED(IDOK, &CCourceDesignDlg::OnBnClickedOk)

ON\_BN\_CLICKED(IDC\_BUTTON3, &CCourceDesignDlg::OnBnClickedButton3)

ON\_BN\_CLICKED(IDC\_BUTTON6, &CCourceDesignDlg::OnBnClickedButton6)

ON\_BN\_CLICKED(IDCANCEL, &CCourceDesignDlg::OnBnClickedCancel)

ON\_BN\_CLICKED(IDC\_BUTTON4, &CCourceDesignDlg::OnBnClickedButton4)

ON\_BN\_CLICKED(IDC\_BUTTON7, &CCourceDesignDlg::OnBnClickedButton7)

ON\_BN\_CLICKED(IDC\_BUTTON8, &CCourceDesignDlg::OnBnClickedButton8)

END\_MESSAGE\_MAP()

// CCourceDesignDlg 消息处理程序

BOOL CCourceDesignDlg::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 CCourceDesignDlg::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 CCourceDesignDlg::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 CCourceDesignDlg::OnQueryDragIcon()

{

return static\_cast<HCURSOR>(m\_hIcon);

}

bool check(student s)//查重

{

for (int i = 0; i < 100; i++)

{

if (s.NUMM == stu[i].NUMM)

{

return FALSE;

}

else

{

continue;

}

}

return TRUE;

}

bool checkchange(student s, int indexx)//检查是否有变动

{

if (s.ADDRESSS == stu[indexx].ADDRESSS && s.AGEE == stu[indexx].AGEE && s.CPPP == stu[indexx].CPPP && s.MATHH == stu[indexx].MATHH && s.NAMEE == stu[indexx].NAMEE && s.NUMM == stu[indexx].NUMM && s.SEXX == stu[indexx].SEXX)

{

return FALSE;

}

else

{

return TRUE;

}

}

void CCourceDesignDlg::OnBnClickedButton1()//修改按钮

{

// TODO: 在此添加控件通知处理程序代码

if (LIST.GetCurSel() != LB\_ERR)

{

int c = LIST.GetCurSel();

UpdateData();

if (NUM == "" || NAME == "" || SEX == "2" || AGE == "" || ADDRESS == "" || MATH == "" || CPP == "")

{

MessageBox(TEXT("请输入完整的信息！"), TEXT("错误"), MB\_ICONWARNING);

}

else

{

student temp;

temp.ADDRESSS = ADDRESS;

temp.AGEE = AGE;

temp.CPPP = CPP;

temp.MATHH = MATH;

temp.NAMEE = NAME;

temp.NUMM = NUM;

temp.SEXX = SEX;

if (checkchange(temp,c))

{

LIST.DeleteString(c);

LIST.InsertString(c, NAME + NUM);

stu[c] = temp;

UpdateData(FALSE);

MessageBox(TEXT("修改成功！"), TEXT("消息"), MB\_ICONINFORMATION);

changed = TRUE;

}

}

UpdateData(FALSE);

}

else

{

MessageBox(TEXT("请选中要修改的项！"), TEXT("错误"), MB\_ICONWARNING);

}

}

void CCourceDesignDlg::OnBnClickedButton2()//增加按钮

{

// TODO: 在此添加控件通知处理程序代码

UpdateData();

if (NUM == "" || NAME == "" || SEX == "2" || AGE == "" || ADDRESS == "" || MATH == "" || CPP == "")

{

MessageBox(TEXT("请输入完整的信息！"), TEXT("错误"), MB\_ICONWARNING);

}

else

{

student temp;

temp.ADDRESSS = ADDRESS;

temp.AGEE = AGE;

temp.CPPP = CPP;

temp.MATHH = MATH;

temp.NAMEE = NAME;

temp.NUMM = NUM;

temp.SEXX = SEX;

if (check(temp))

{

LIST.AddString(NAME + NUM);

stu[index] = temp;

ADDRESS = "";

AGE = "";

CPP = "";

MATH = "";

NAME = "";

NUM = "";

SEX = "2";

MALE.SetCheck(0);

FEMALE.SetCheck(0);

UpdateData(FALSE);

index++;

changed = TRUE;

}

else

{

MessageBox(TEXT("已有该学生信息！"), TEXT("错误"), MB\_ICONWARNING);

}

}

UpdateData(FALSE);

}

void CCourceDesignDlg::OnBnClickedRadio2()

{

// TODO: 在此添加控件通知处理程序代码

SEX = "1";

}

void CCourceDesignDlg::OnBnClickedRadio3()

{

// TODO: 在此添加控件通知处理程序代码

SEX = "0";

}

void CCourceDesignDlg::OnBnClickedButton5()//清空输入框按钮

{

// TODO: 在此添加控件通知处理程序代码

ADDRESS = "";

AGE = "";

CPP = "";

MATH = "";

NAME = "";

NUM = "";

SEX = "2";

MALE.SetCheck(0);

FEMALE.SetCheck(0);

UpdateData(FALSE);

}

void CCourceDesignDlg::OnLbnSelchangeList1()

{

if (LIST.GetCurSel() != LB\_ERR)

{

int a = LIST.GetCurSel();

ADDRESS = stu[a].ADDRESSS;

AGE = stu[a].AGEE;

CPP = stu[a].CPPP;

MATH = stu[a].MATHH;

NAME = stu[a].NAMEE;

NUM = stu[a].NUMM;

if (stu[a].SEXX == "0")

{

SEX = "0";

FEMALE.SetCheck(1);

MALE.SetCheck(0);

UpdateData(FALSE);

}

else if (stu[a].SEXX == "1")

{

SEX = "1";

FEMALE.SetCheck(0);

MALE.SetCheck(1);

UpdateData(FALSE);

}

UpdateData(FALSE);

}

// TODO: 在此添加控件通知处理程序代码

}

void CCourceDesignDlg::OnBnClickedOk()//另存为按钮

{

// TODO: 在此添加控件通知处理程序代码

if (index != 0)

{

CFileDialog dlg(FALSE, TEXT("list"), TEXT("学生列表"), OFN\_OVERWRITEPROMPT, TEXT("列表数据文件(\*.list)|\*.list||"));

if (dlg.DoModal() == IDOK)

{

CString L = "|";

CString end = "$";

PathName = dlg.GetPathName();

CFile oFile;

oFile.Open(PathName, CFile::modeCreate | CFile::modeWrite);

for (int i = 0; i < index; i++)

{

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].ADDRESSS.GetBuffer(), stu[i].ADDRESSS.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].AGEE.GetBuffer(), stu[i].AGEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].CPPP.GetBuffer(), stu[i].CPPP.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].MATHH.GetBuffer(), stu[i].MATHH.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NAMEE.GetBuffer(), stu[i].NAMEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NUMM.GetBuffer(), stu[i].NUMM.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].SEXX.GetBuffer(), stu[i].SEXX.GetLength());

}

oFile.Write(end.GetBuffer(), end.GetLength());

UpdateData(FALSE);

oFile.Close();

saved = TRUE;

}

}

else

{

MessageBox(TEXT("请不要保存空表！"), TEXT("错误"), MB\_ICONWARNING);

}

}

void CCourceDesignDlg::OnBnClickedButton3()//删除按钮

{

// TODO: 在此添加控件通知处理程序代码

if (LIST.GetCurSel() != LB\_ERR)

{

int b = LIST.GetCurSel();

LIST.DeleteString(b);

for (int i = b; i < 100; i++)

{

stu[b] = stu[b + 1];

}

index--;

changed = TRUE;

ADDRESS = "";

AGE = "";

CPP = "";

MATH = "";

NAME = "";

NUM = "";

SEX = "2";

MALE.SetCheck(0);

FEMALE.SetCheck(0);

UpdateData(FALSE);

MessageBox(TEXT("已删除"), TEXT("消息"), MB\_ICONINFORMATION);

}

else

{

MessageBox(TEXT("请选中要删除的项！"), TEXT("错误"), MB\_ICONWARNING);

}

}

void CCourceDesignDlg::OnBnClickedButton6()//打开文件按钮

{

// TODO: 在此添加控件通知处理程序代码

if (changed)

{

int choice = MessageBox(TEXT("您尚未保存！是否保存？"), TEXT("警告"), MB\_YESNOCANCEL | MB\_ICONQUESTION);

if (choice == IDYES)

{

if (saved)

{

CString L = "|";

CString end = "$";

CFile oFile;

oFile.Open(PathName, CFile::modeCreate | CFile::modeWrite);

for (int i = 0; i < index; i++)

{

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].ADDRESSS.GetBuffer(), stu[i].ADDRESSS.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].AGEE.GetBuffer(), stu[i].AGEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].CPPP.GetBuffer(), stu[i].CPPP.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].MATHH.GetBuffer(), stu[i].MATHH.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NAMEE.GetBuffer(), stu[i].NAMEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NUMM.GetBuffer(), stu[i].NUMM.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].SEXX.GetBuffer(), stu[i].SEXX.GetLength());

}

oFile.Write(end.GetBuffer(), end.GetLength());

oFile.Close();

CFileDialog dlg(TRUE, TEXT("list"), TEXT("学生列表"), OFN\_HIDEREADONLY, TEXT("列表数据文件(\*.list)|\*.list||"));

if (dlg.DoModal() == IDOK)

{

PathName = dlg.GetPathName();

CFile file;

file.Open(PathName, CFile::modeRead);

char\* buf;

int len = file.GetLength();

buf = new char[len + 1];

file.Read(buf, len);

buf[len] = 0;

CString\* r = new CString[len];

int dex = -2;

char\* temp = new char[len + 1];

temp[len] = 0;

int h = 0;

int j = 0;

for (int i = 0; i < len; i++)

{

if (buf[i] == '|')

{

dex++;

if (dex != -1)

{

r[dex] = temp;

}

for (int k = 0; k < len + 1; k++)

{

temp[k] = NULL;

}

j = 0;

}

else if (buf[i] == '$')

{

dex++;

r[dex] = temp;

break;

}

else

{

temp[j] = buf[i];

j++;

}

}

int de = 0;

dex++;

index = 0;

LIST.ResetContent();

stu = new student[100];

for (int i = 0; i < dex / 7; i++)

{

stu[i].ADDRESSS = r[de];

de++;

stu[i].AGEE = r[de];

de++;

stu[i].CPPP = r[de];

de++;

stu[i].MATHH = r[de];

de++;

stu[i].NAMEE = r[de];

de++;

stu[i].NUMM = r[de];

de++;

stu[i].SEXX = r[de];

de++;

LIST.AddString(stu[i].NAMEE + stu[i].NUMM);

index++;

}

file.Close();

ADDRESS = "";

AGE = "";

CPP = "";

MATH = "";

NAME = "";

NUM = "";

SEX = "2";

MALE.SetCheck(0);

FEMALE.SetCheck(0);

saved = TRUE;

UpdateData(FALSE);

}

}

else

{

CFileDialog dlg1(FALSE, TEXT("list"), TEXT("学生列表"), OFN\_OVERWRITEPROMPT, TEXT("列表数据文件(\*.list)|\*.list||"));

if (dlg1.DoModal() == IDOK)

{

CString L = "|";

CString end = "$";

PathName = dlg1.GetPathName();

CFile oFile;

oFile.Open(PathName, CFile::modeCreate | CFile::modeWrite);

for (int i = 0; i < index; i++)

{

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].ADDRESSS.GetBuffer(), stu[i].ADDRESSS.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].AGEE.GetBuffer(), stu[i].AGEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].CPPP.GetBuffer(), stu[i].CPPP.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].MATHH.GetBuffer(), stu[i].MATHH.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NAMEE.GetBuffer(), stu[i].NAMEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NUMM.GetBuffer(), stu[i].NUMM.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].SEXX.GetBuffer(), stu[i].SEXX.GetLength());

}

oFile.Write(end.GetBuffer(), end.GetLength());

oFile.Close();

UpdateData(FALSE);

}

CFileDialog dlg2(TRUE, TEXT("list"), TEXT("学生列表"), OFN\_HIDEREADONLY, TEXT("列表数据文件(\*.list)|\*.list||"));

if (dlg2.DoModal() == IDOK)

{

PathName = dlg2.GetPathName();

CFile file;

file.Open(PathName, CFile::modeRead);

char\* buf;

int len = file.GetLength();

buf = new char[len + 1];

file.Read(buf, len);

buf[len] = 0;

CString\* r = new CString[len];

int dex = -2;

char\* temp = new char[len + 1];

temp[len] = 0;

int h = 0;

int j = 0;

for (int i = 0; i < len; i++)

{

if (buf[i] == '|')

{

dex++;

if (dex != -1)

{

r[dex] = temp;

}

for (int k = 0; k < len + 1; k++)

{

temp[k] = NULL;

}

j = 0;

}

else if (buf[i] == '$')

{

dex++;

r[dex] = temp;

break;

}

else

{

temp[j] = buf[i];

j++;

}

}

int de = 0;

dex++;

index = 0;

LIST.ResetContent();

stu = new student[100];

for (int i = 0; i < dex / 7; i++)

{

stu[i].ADDRESSS = r[de];

de++;

stu[i].AGEE = r[de];

de++;

stu[i].CPPP = r[de];

de++;

stu[i].MATHH = r[de];

de++;

stu[i].NAMEE = r[de];

de++;

stu[i].NUMM = r[de];

de++;

stu[i].SEXX = r[de];

de++;

LIST.AddString(stu[i].NAMEE + stu[i].NUMM);

index++;

}

file.Close();

ADDRESS = "";

AGE = "";

CPP = "";

MATH = "";

NAME = "";

NUM = "";

SEX = "2";

MALE.SetCheck(0);

FEMALE.SetCheck(0);

saved = TRUE;

UpdateData(FALSE);

}

}

changed = FALSE;

}

else if (choice == IDNO)

{

CFileDialog dlg(TRUE, TEXT("list"), TEXT("学生列表"), OFN\_HIDEREADONLY, TEXT("列表数据文件(\*.list)|\*.list||"));

if (dlg.DoModal() == IDOK)

{

PathName = dlg.GetPathName();

CFile file;

file.Open(PathName, CFile::modeRead);

char\* buf;

int len = file.GetLength();

buf = new char[len + 1];

file.Read(buf, len);

buf[len] = 0;

CString\* r = new CString[len];

int dex = -2;

char\* temp = new char[len + 1];

temp[len] = 0;

int h = 0;

int j = 0;

for (int i = 0; i < len; i++)

{

if (buf[i] == '|')

{

dex++;

if (dex != -1)

{

r[dex] = temp;

}

for (int k = 0; k < len + 1; k++)

{

temp[k] = NULL;

}

j = 0;

}

else if (buf[i] == '$')

{

dex++;

r[dex] = temp;

break;

}

else

{

temp[j] = buf[i];

j++;

}

}

int de = 0;

dex++;

index = 0;

LIST.ResetContent();

stu = new student[100];

for (int i = 0; i < dex / 7; i++)

{

stu[i].ADDRESSS = r[de];

de++;

stu[i].AGEE = r[de];

de++;

stu[i].CPPP = r[de];

de++;

stu[i].MATHH = r[de];

de++;

stu[i].NAMEE = r[de];

de++;

stu[i].NUMM = r[de];

de++;

stu[i].SEXX = r[de];

de++;

LIST.AddString(stu[i].NAMEE + stu[i].NUMM);

index++;

}

file.Close();

ADDRESS = "";

AGE = "";

CPP = "";

MATH = "";

NAME = "";

NUM = "";

SEX = "2";

MALE.SetCheck(0);

FEMALE.SetCheck(0);

saved = TRUE;

changed = FALSE;

UpdateData(FALSE);

}

}

}

else

{

CFileDialog dlg(TRUE, TEXT("list"), TEXT("学生列表"), OFN\_HIDEREADONLY, TEXT("列表数据文件(\*.list)|\*.list||"));

if (dlg.DoModal() == IDOK)

{

PathName = dlg.GetPathName();

CFile file;

file.Open(PathName, CFile::modeRead);

char\* buf;

int len = file.GetLength();

buf = new char[len + 1];

file.Read(buf, len);

buf[len] = 0;

CString\* r = new CString[len];

int dex = -2;

char\* temp = new char[len + 1];

temp[len] = 0;

int h = 0;

int j = 0;

for (int i = 0; i < len; i++)

{

if (buf[i] == '|')

{

dex++;

if (dex != -1)

{

r[dex] = temp;

}

for (int k = 0; k < len + 1; k++)

{

temp[k] = NULL;

}

j = 0;

}

else if (buf[i] == '$')

{

dex++;

r[dex] = temp;

break;

}

else

{

temp[j] = buf[i];

j++;

}

}

int de = 0;

dex++;

index = 0;

LIST.ResetContent();

stu = new student[100];

for (int i = 0; i < dex / 7; i++)

{

stu[i].ADDRESSS = r[de];

de++;

stu[i].AGEE = r[de];

de++;

stu[i].CPPP = r[de];

de++;

stu[i].MATHH = r[de];

de++;

stu[i].NAMEE = r[de];

de++;

stu[i].NUMM = r[de];

de++;

stu[i].SEXX = r[de];

de++;

LIST.AddString(stu[i].NAMEE + stu[i].NUMM);

index++;

}

file.Close();

ADDRESS = "";

AGE = "";

CPP = "";

MATH = "";

NAME = "";

NUM = "";

SEX = "2";

MALE.SetCheck(0);

FEMALE.SetCheck(0);

saved = TRUE;

changed = FALSE;

UpdateData(FALSE);

}

}

}

void CCourceDesignDlg::OnBnClickedCancel()//退出按钮

{

// TODO: 在此添加控件通知处理程序代码

if (changed)

{

int choice = MessageBox(TEXT("您尚未保存！是否保存？"), TEXT("警告"), MB\_YESNOCANCEL | MB\_ICONQUESTION);

if (choice == IDYES)

{

if (saved)

{

CString L = "|";

CString end = "$";

CFile oFile;

oFile.Open(PathName, CFile::modeCreate | CFile::modeWrite);

for (int i = 0; i < index; i++)

{

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].ADDRESSS.GetBuffer(), stu[i].ADDRESSS.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].AGEE.GetBuffer(), stu[i].AGEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].CPPP.GetBuffer(), stu[i].CPPP.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].MATHH.GetBuffer(), stu[i].MATHH.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NAMEE.GetBuffer(), stu[i].NAMEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NUMM.GetBuffer(), stu[i].NUMM.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].SEXX.GetBuffer(), stu[i].SEXX.GetLength());

}

oFile.Write(end.GetBuffer(), end.GetLength());

oFile.Close();

UpdateData(FALSE);

CDialogEx::OnCancel();

}

else

{

CFileDialog dlg(FALSE, TEXT("list"), TEXT("学生列表"), OFN\_OVERWRITEPROMPT, TEXT("列表数据文件(\*.list)|\*.list||"));

if (dlg.DoModal() == IDOK)

{

CString L = "|";

CString end = "$";

PathName = dlg.GetPathName();

CFile oFile;

oFile.Open(PathName, CFile::modeCreate | CFile::modeWrite);

for (int i = 0; i < index; i++)

{

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].ADDRESSS.GetBuffer(), stu[i].ADDRESSS.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].AGEE.GetBuffer(), stu[i].AGEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].CPPP.GetBuffer(), stu[i].CPPP.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].MATHH.GetBuffer(), stu[i].MATHH.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NAMEE.GetBuffer(), stu[i].NAMEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NUMM.GetBuffer(), stu[i].NUMM.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].SEXX.GetBuffer(), stu[i].SEXX.GetLength());

}

oFile.Write(end.GetBuffer(), end.GetLength());

oFile.Close();

UpdateData(FALSE);

CDialogEx::OnCancel();

}

}

}

else if (choice == IDNO)

{

CDialogEx::OnCancel();

}

else if (choice == IDCANCEL)

{

;

}

}

else

{

CDialogEx::OnCancel();

}

}

void CCourceDesignDlg::OnBnClickedButton4()

{

// TODO: 在此添加控件通知处理程序代码

}

void CCourceDesignDlg::OnBnClickedButton7()//保存按钮

{

// TODO: 在此添加控件通知处理程序代码

if (index != 0)

{

if (saved)

{

CString L = "|";

CString end = "$";

CFile oFile;

oFile.Open(PathName, CFile::modeCreate | CFile::modeWrite);

for (int i = 0; i < index; i++)

{

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].ADDRESSS.GetBuffer(), stu[i].ADDRESSS.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].AGEE.GetBuffer(), stu[i].AGEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].CPPP.GetBuffer(), stu[i].CPPP.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].MATHH.GetBuffer(), stu[i].MATHH.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NAMEE.GetBuffer(), stu[i].NAMEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NUMM.GetBuffer(), stu[i].NUMM.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].SEXX.GetBuffer(), stu[i].SEXX.GetLength());

}

oFile.Write(end.GetBuffer(), end.GetLength());

UpdateData(FALSE);

oFile.Close();

saved = TRUE;

changed = FALSE;

MessageBox(TEXT("已保存"), TEXT("消息"), MB\_ICONINFORMATION);

}

else

{

CFileDialog dlg(FALSE, TEXT("list"), TEXT("学生列表"), OFN\_OVERWRITEPROMPT, TEXT("列表数据文件(\*.list)|\*.list||"));

if (dlg.DoModal() == IDOK)

{

CString L = "|";

CString end = "$";

PathName = dlg.GetPathName();

CFile oFile;

oFile.Open(PathName, CFile::modeCreate | CFile::modeWrite);

for (int i = 0; i < index; i++)

{

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].ADDRESSS.GetBuffer(), stu[i].ADDRESSS.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].AGEE.GetBuffer(), stu[i].AGEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].CPPP.GetBuffer(), stu[i].CPPP.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].MATHH.GetBuffer(), stu[i].MATHH.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NAMEE.GetBuffer(), stu[i].NAMEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NUMM.GetBuffer(), stu[i].NUMM.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].SEXX.GetBuffer(), stu[i].SEXX.GetLength());

}

oFile.Write(end.GetBuffer(), end.GetLength());

UpdateData(FALSE);

oFile.Close();

changed = FALSE;

saved = TRUE;

}

}

}

else

{

MessageBox(TEXT("请不要保存空表！"), TEXT("错误"), MB\_ICONWARNING);

}

}

void CCourceDesignDlg::OnBnClickedButton8()//新建空白表按钮

{

// TODO: 在此添加控件通知处理程序代码

if (changed)

{

int choice = MessageBox(TEXT("您尚未保存！是否保存？"), TEXT("警告"), MB\_YESNOCANCEL | MB\_ICONQUESTION);

if (choice == IDOK)

{

if (saved)

{

CString L = "|";

CString end = "$";

CFile oFile;

oFile.Open(PathName, CFile::modeCreate | CFile::modeWrite);

for (int i = 0; i < index; i++)

{

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].ADDRESSS.GetBuffer(), stu[i].ADDRESSS.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].AGEE.GetBuffer(), stu[i].AGEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].CPPP.GetBuffer(), stu[i].CPPP.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].MATHH.GetBuffer(), stu[i].MATHH.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NAMEE.GetBuffer(), stu[i].NAMEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NUMM.GetBuffer(), stu[i].NUMM.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].SEXX.GetBuffer(), stu[i].SEXX.GetLength());

}

oFile.Write(end.GetBuffer(), end.GetLength());

UpdateData(FALSE);

oFile.Close();

saved = FALSE;

ADDRESS = "";

AGE = "";

CPP = "";

MATH = "";

NAME = "";

NUM = "";

SEX = "2";

MALE.SetCheck(0);

FEMALE.SetCheck(0);

UpdateData(FALSE);

}

else

{

CFileDialog dlg(FALSE, TEXT("list"), TEXT("学生列表"), OFN\_OVERWRITEPROMPT, TEXT("列表数据文件(\*.list)|\*.list||"));

if (dlg.DoModal() == IDOK)

{

CString L = "|";

CString end = "$";

PathName = dlg.GetPathName();

CFile oFile;

oFile.Open(PathName, CFile::modeCreate | CFile::modeWrite);

for (int i = 0; i < index; i++)

{

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].ADDRESSS.GetBuffer(), stu[i].ADDRESSS.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].AGEE.GetBuffer(), stu[i].AGEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].CPPP.GetBuffer(), stu[i].CPPP.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].MATHH.GetBuffer(), stu[i].MATHH.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NAMEE.GetBuffer(), stu[i].NAMEE.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].NUMM.GetBuffer(), stu[i].NUMM.GetLength());

oFile.Write(L.GetBuffer(), L.GetLength());

oFile.Write(stu[i].SEXX.GetBuffer(), stu[i].SEXX.GetLength());

}

oFile.Write(end.GetBuffer(), end.GetLength());

UpdateData(FALSE);

oFile.Close();

saved = FALSE;

ADDRESS = "";

AGE = "";

CPP = "";

MATH = "";

NAME = "";

NUM = "";

SEX = "2";

MALE.SetCheck(0);

FEMALE.SetCheck(0);

UpdateData(FALSE);

}

}

}

else if (choice == IDNO)

{

index = 0;

LIST.ResetContent();

stu = new student[100];

ADDRESS = "";

AGE = "";

CPP = "";

MATH = "";

NAME = "";

NUM = "";

SEX = "2";

MALE.SetCheck(0);

FEMALE.SetCheck(0);

saved = FALSE;

UpdateData(FALSE);

}

else if (choice == IDCANCEL)

{

;

}

}

else

{

index = 0;

LIST.ResetContent();

stu = new student[100];

ADDRESS = "";

AGE = "";

CPP = "";

MATH = "";

NAME = "";

NUM = "";

SEX = "2";

MALE.SetCheck(0);

FEMALE.SetCheck(0);

saved = FALSE;

UpdateData(FALSE);

}

changed = FALSE;

}

// AnalysisView.cpp: CAnalysisView 类的实现

//

#include "pch.h"

#include "framework.h"

// SHARED\_HANDLERS 可以在实现预览、缩略图和搜索筛选器句柄的

// ATL 项目中进行定义，并允许与该项目共享文档代码。

#ifndef SHARED\_HANDLERS

#include "Analysis.h"

#endif

#include "AnalysisDoc.h"

#include "AnalysisView.h"

#include "MainFrm.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#endif

// CAnalysisView

IMPLEMENT\_DYNCREATE(CAnalysisView, CView)

BEGIN\_MESSAGE\_MAP(CAnalysisView, CView)

// 标准打印命令

ON\_COMMAND(ID\_FILE\_PRINT, &CView::OnFilePrint)

ON\_COMMAND(ID\_FILE\_PRINT\_DIRECT, &CView::OnFilePrint)

ON\_COMMAND(ID\_FILE\_PRINT\_PREVIEW, &CView::OnFilePrintPreview)

ON\_COMMAND(ID\_32771, &CAnalysisView::On32771)

ON\_COMMAND(ID\_32772, &CAnalysisView::On32772)

END\_MESSAGE\_MAP()

// CAnalysisView 构造/析构

CAnalysisView::CAnalysisView() noexcept

{

// TODO: 在此处添加构造代码

}

CAnalysisView::~CAnalysisView()

{

}

BOOL CAnalysisView::PreCreateWindow(CREATESTRUCT& cs)

{

// TODO: 在此处通过修改

// CREATESTRUCT cs 来修改窗口类或样式

return CView::PreCreateWindow(cs);

}

// CAnalysisView 绘图

void CAnalysisView::DrawScore(CDC\* pDC, float\* fScore, int nNum)//绘制直方图

{

int nScoreNum[] = { 0,0,0,0,0 };//各成绩段的人数初始值

for (int i = 0; i < nNum; i++)

{

int nSeg = (int)(fScore[i]) / 10;//取十位上的值

if (nSeg < 6)nSeg = 5;//<60分

if (nSeg == 10)nSeg = 9;//当为100分，算为>90分数段

nScoreNum[nSeg - 5]++;//各分数段计数

}

int nSegNum = sizeof(nScoreNum) / sizeof(int);//计算有多少个分数段

//求分数段上最大的人数

int nNumMax = nScoreNum[0];

for (int i = 1; i < nSegNum; i++)

{

if (nNumMax < nScoreNum[i])nNumMax = nScoreNum[i];

}

CRect rc;

GetClientRect(rc);

rc.DeflateRect(40, 40);//缩小矩形大小

int nSegWidth = rc.Width() / nSegNum;//计算每段的宽度

int nSegHeight = rc.Height() / nNumMax;//计算每段的单位高度

COLORREF crSeg = RGB(0, 0, 192);//定义一个颜色变量

CBrush brush1(HS\_FDIAGONAL, crSeg);

CBrush brush2(HS\_BDIAGONAL, crSeg);

CPen pen(PS\_INSIDEFRAME, 2, crSeg);

CBrush\* oldBrush = pDC->SelectObject(&brush1);//将brush1选入设备环境

CPen\* oldPen = pDC->SelectObject(&pen);//将pen选入设备环境

CRect rcSeg(rc);

rcSeg.right = rcSeg.left + nSegWidth;//使每段的矩形宽度等于nSegWidth

CString strSeg[] = {TEXT("<60"),TEXT("60-70"),TEXT("70-80"),TEXT("80-90"),TEXT(">=90")};

CRect rcStr;

for (int i = 0; i < nSegNum; i++)

{

//保证相邻矩形填充样式不相同

if (i % 2)

pDC->SelectObject(&brush2);

else

pDC->SelectObject(&brush1);

rcSeg.top = rcSeg.bottom - nScoreNum[i] \* nSegHeight - 2;//计算每段矩形的高度

pDC->Rectangle(rcSeg);

if (nScoreNum[i] > 0)

{

CString str;

str.Format(TEXT("%d人"), nScoreNum[i]);

pDC->DrawText(str, rcSeg, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

}

rcStr = rcSeg;

rcStr.top = rcStr.bottom + 2;

rcStr.bottom += 20;

pDC->DrawText(strSeg[i], rcStr, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

rcSeg.OffsetRect(nSegWidth, 0);//右移矩形

}

pDC->SelectObject(oldBrush);//恢复到原来的画刷属性

pDC->SelectObject(oldPen);//恢复到原来的画笔属性

}

void CAnalysisView::OnDraw(CDC\* pDC)

{

CAnalysisDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

if (!pDoc)

return;

// TODO: 在此处为本机数据添加绘制代码

}

// CAnalysisView 打印

BOOL CAnalysisView::OnPreparePrinting(CPrintInfo\* pInfo)

{

// 默认准备

return DoPreparePrinting(pInfo);

}

void CAnalysisView::OnBeginPrinting(CDC\* /\*pDC\*/, CPrintInfo\* /\*pInfo\*/)

{

// TODO: 添加额外的打印前进行的初始化过程

}

void CAnalysisView::OnEndPrinting(CDC\* /\*pDC\*/, CPrintInfo\* /\*pInfo\*/)

{

// TODO: 添加打印后进行的清理过程

}

// CAnalysisView 诊断

#ifdef \_DEBUG

void CAnalysisView::AssertValid() const

{

CView::AssertValid();

}

void CAnalysisView::Dump(CDumpContext& dc) const

{

CView::Dump(dc);

}

CAnalysisDoc\* CAnalysisView::GetDocument() const // 非调试版本是内联的

{

ASSERT(m\_pDocument->IsKindOf(RUNTIME\_CLASS(CAnalysisDoc)));

return (CAnalysisDoc\*)m\_pDocument;

}

#endif //\_DEBUG

void CAnalysisView::On32771()//绘制C++成绩直方图

{

CAnalysisDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

if (!pDoc)

return;

// TODO: 在此添加命令处理程序代码

CDC\* pDC = GetDC();

float\* Data = pDoc->CPP;

int count = pDoc->len;

Invalidate();//使窗口无效化

UpdateWindow();//重绘窗口

DrawScore(pDC, Data, count);

}

void CAnalysisView::On32772()//绘制数学成绩直方图

{

CAnalysisDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

if (!pDoc)

return;

// TODO: 在此添加命令处理程序代码

CDC\* pDC = GetDC();

float\* Data = pDoc->MATH;

int count = pDoc->len;

Invalidate();//使窗口无效化

UpdateWindow();//重绘窗口

DrawScore(pDC, Data, count);

}

// AnalysisDoc.cpp: CAnalysisDoc 类的实现

//

#include "pch.h"

#include "framework.h"

// SHARED\_HANDLERS 可以在实现预览、缩略图和搜索筛选器句柄的

// ATL 项目中进行定义，并允许与该项目共享文档代码。

#ifndef SHARED\_HANDLERS

#include "Analysis.h"

#endif

#include "AnalysisDoc.h"

#include "AnalysisView.h"

#include <propkey.h>

#ifdef \_DEBUG

#define new DEBUG\_NEW

#endif

// CAnalysisDoc

IMPLEMENT\_DYNCREATE(CAnalysisDoc, CDocument)

BEGIN\_MESSAGE\_MAP(CAnalysisDoc, CDocument)

END\_MESSAGE\_MAP()

// CAnalysisDoc 构造/析构

CAnalysisDoc::CAnalysisDoc() noexcept

{

// TODO: 在此添加一次性构造代码

}

CAnalysisDoc::~CAnalysisDoc()

{

}

BOOL CAnalysisDoc::OnNewDocument()

{

if (!CDocument::OnNewDocument())

return FALSE;

// TODO: 在此添加重新初始化代码

// (SDI 文档将重用该文档)

return TRUE;

}

// CAnalysisDoc 序列化

void CAnalysisDoc::Serialize(CArchive& ar)

{

if (ar.IsStoring())

{

// TODO: 在此添加存储代码

}

else

{

// TODO: 在此添加加载代码

char buf[700];

CString r[700];

int dex = -2;

char temp[700];

int h = 0;

int j = 0;

for (int i = 0;; i++)

{

ar >> buf[i];

if (buf[i] == '|')

{

dex++;

if (dex != -1)

{

r[dex] = temp;

}

for (int k = 0; k < 700; k++)

{

temp[k] = NULL;

}

j = 0;

}

else if (buf[i] == '$')

{

dex++;

r[dex] = temp;

break;

}

else

{

temp[j] = buf[i];

j++;

}

}

int de = 0;

dex++;

float\* data = new float[dex];

for (int i = 3; i < dex; i += 7)

{

data[de] = \_ttof(r[i]);

de++;

}

len = de;

CPP = data;

de = 0;

data = new float[dex];

for (int i = 4; i < dex; i += 7)

{

data[de] = \_ttof(r[i]);

de++;

}

MATH = data;

}

}

#ifdef SHARED\_HANDLERS

// 缩略图的支持

void CAnalysisDoc::OnDrawThumbnail(CDC& dc, LPRECT lprcBounds)

{

// 修改此代码以绘制文档数据

dc.FillSolidRect(lprcBounds, RGB(255, 255, 255));

CString strText = \_T("TODO: implement thumbnail drawing here");

LOGFONT lf;

CFont\* pDefaultGUIFont = CFont::FromHandle((HFONT) GetStockObject(DEFAULT\_GUI\_FONT));

pDefaultGUIFont->GetLogFont(&lf);

lf.lfHeight = 36;

CFont fontDraw;

fontDraw.CreateFontIndirect(&lf);

CFont\* pOldFont = dc.SelectObject(&fontDraw);

dc.DrawText(strText, lprcBounds, DT\_CENTER | DT\_WORDBREAK);

dc.SelectObject(pOldFont);

}

// 搜索处理程序的支持

void CAnalysisDoc::InitializeSearchContent()

{

CString strSearchContent;

// 从文档数据设置搜索内容。

// 内容部分应由“;”分隔

// 例如: strSearchContent = \_T("point;rectangle;circle;ole object;")；

SetSearchContent(strSearchContent);

}

void CAnalysisDoc::SetSearchContent(const CString& value)

{

if (value.IsEmpty())

{

RemoveChunk(PKEY\_Search\_Contents.fmtid, PKEY\_Search\_Contents.pid);

}

else

{

CMFCFilterChunkValueImpl \*pChunk = nullptr;

ATLTRY(pChunk = new CMFCFilterChunkValueImpl);

if (pChunk != nullptr)

{

pChunk->SetTextValue(PKEY\_Search\_Contents, value, CHUNK\_TEXT);

SetChunkValue(pChunk);

}

}

}

#endif // SHARED\_HANDLERS

// CAnalysisDoc 诊断

#ifdef \_DEBUG

void CAnalysisDoc::AssertValid() const

{

CDocument::AssertValid();

}

void CAnalysisDoc::Dump(CDumpContext& dc) const

{

CDocument::Dump(dc);

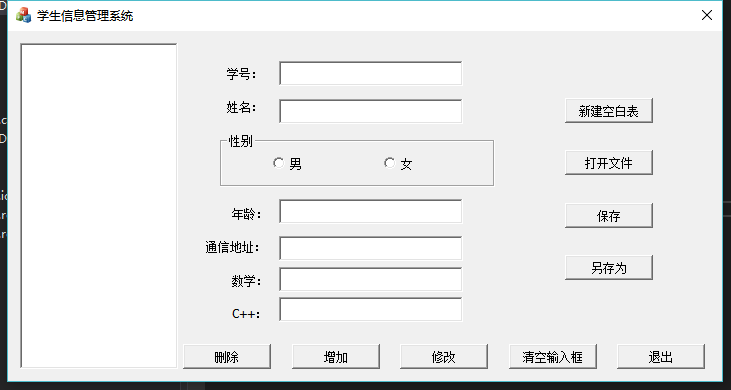
}

#endif //\_DEBUG

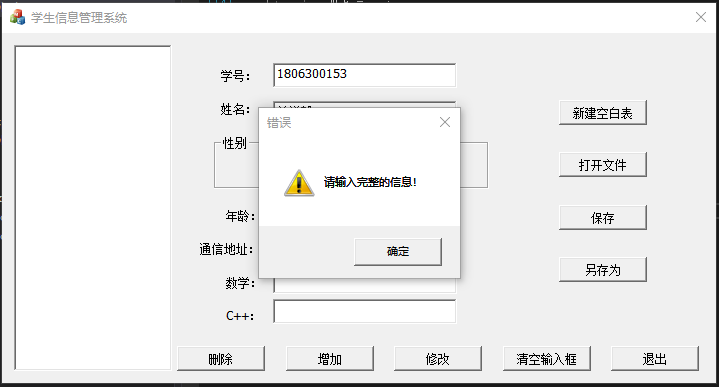
// CAnalysisDoc 命令

1. **实验结果**

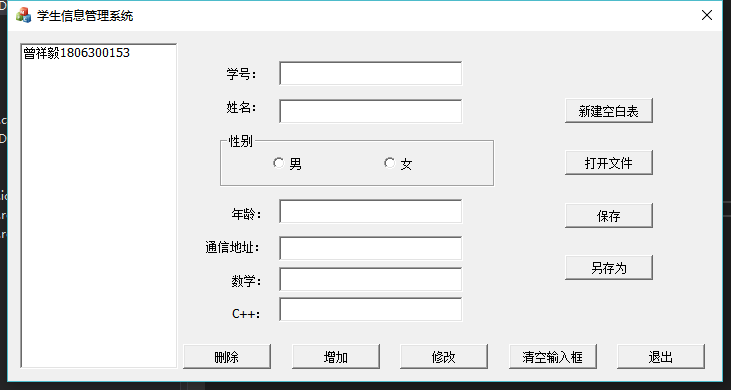
**界面：**



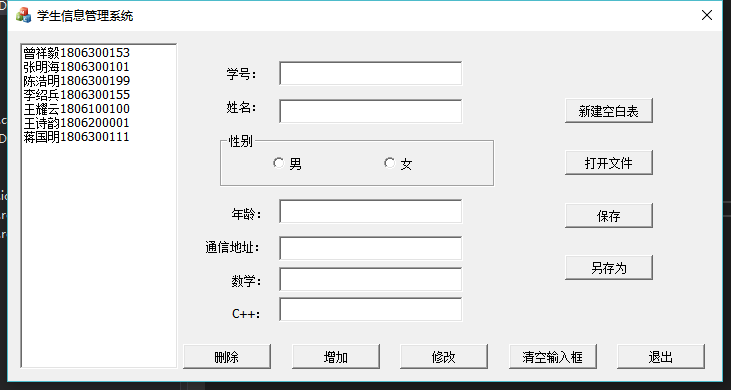
当输入信息不完整时有提示：



添加一个学生：



添加多名学生：



选中列表项来查看某位学生的信息：



清空输入框：



将“王诗韵”的C++成绩修改为83分：



在没有任何改动的情况下点击修改按钮是没有反应的：

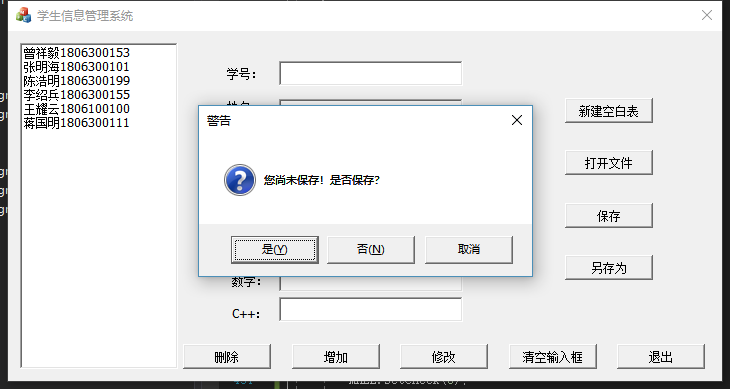


删除“王诗韵”的信息：

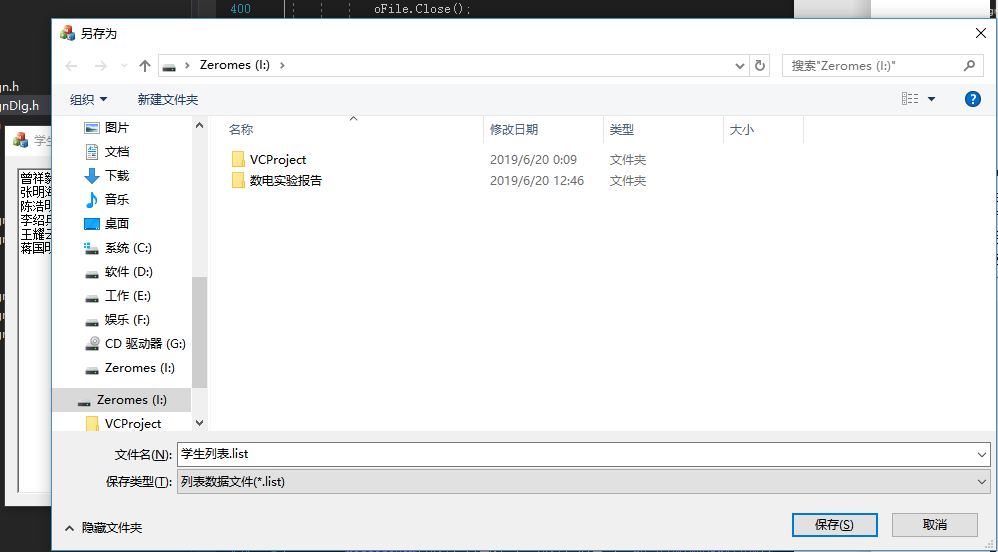


当点击“新建空白表”、“打开文件”、“退出”按钮时，如果为保存则会提醒保存，如果保存而且关联了文件，程序会自动保存。如果尚未关联文件（是新表）则会让用户选择保存的路径。

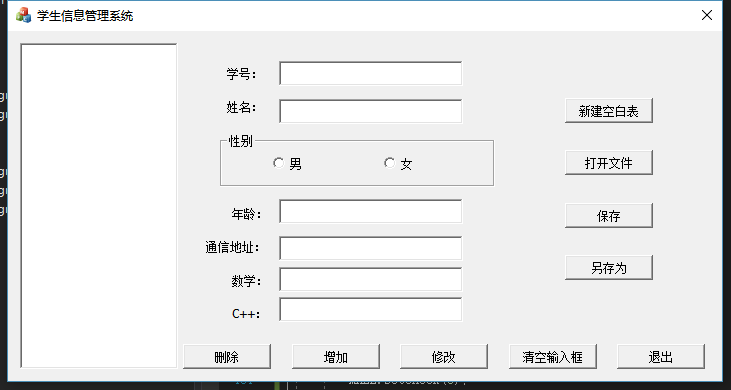
选择新建空白表：



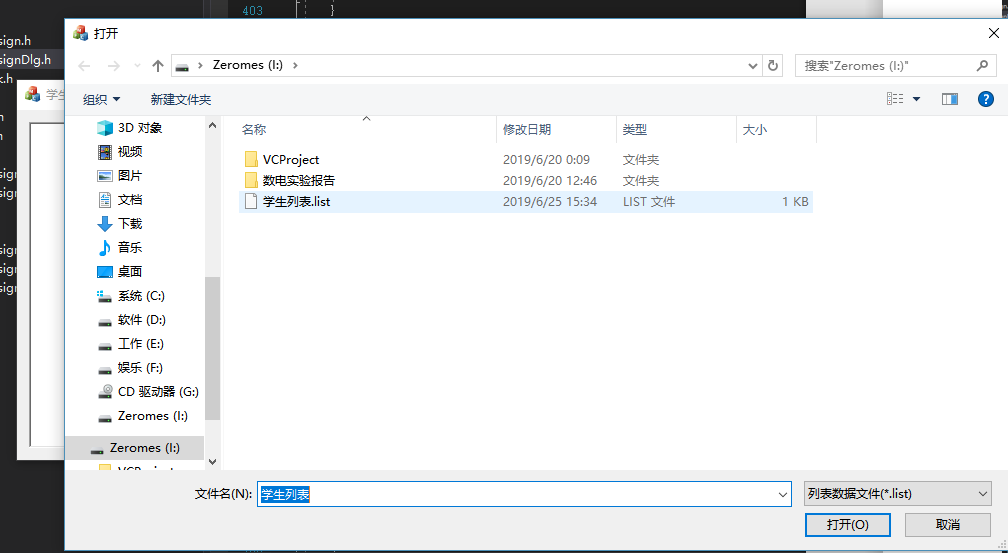
点击“是”。这里由于是新表，会让用户选择保存路径：



新建空白表完成：



点击打开文件：



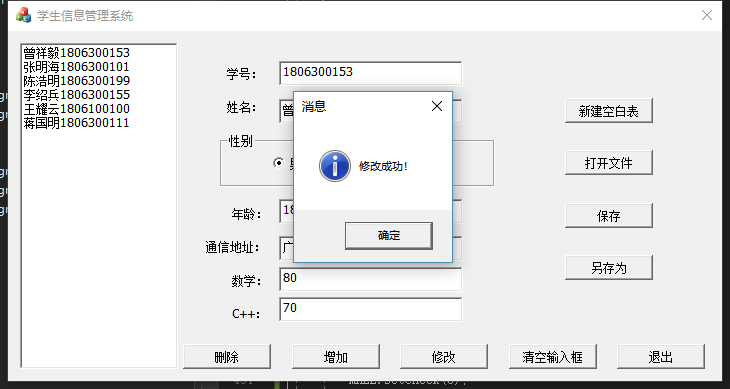
打开“学生列表.list”：



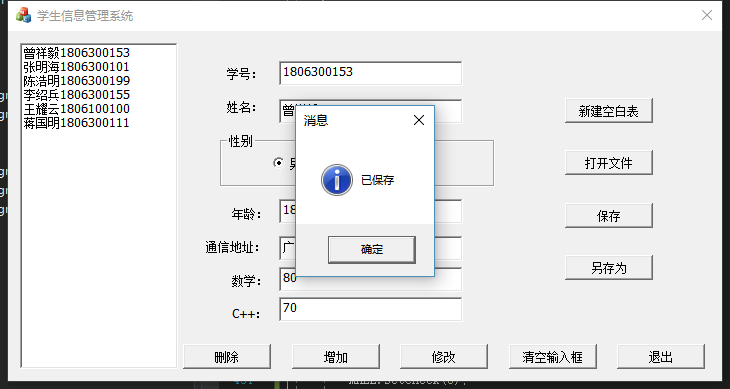
可以正常查看各学生的信息：



将“曾祥毅”的数学成绩改为80分：

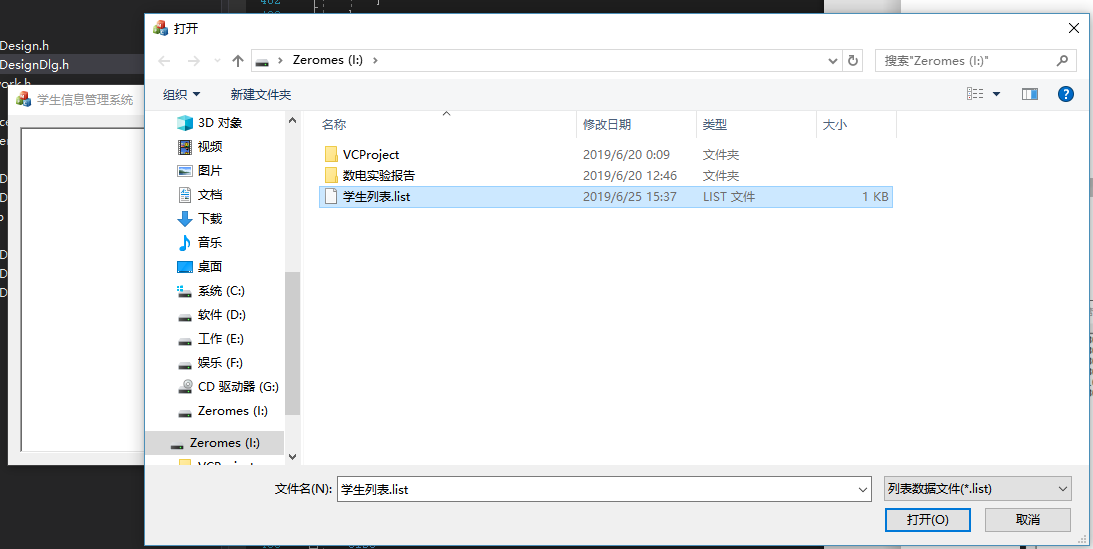


点击保存，由于是打开的文件，已经关联了本地的文件，所以不会让用户选择路径而是直接保存：



退出，因为已经保存，所以没有提示是否保存。

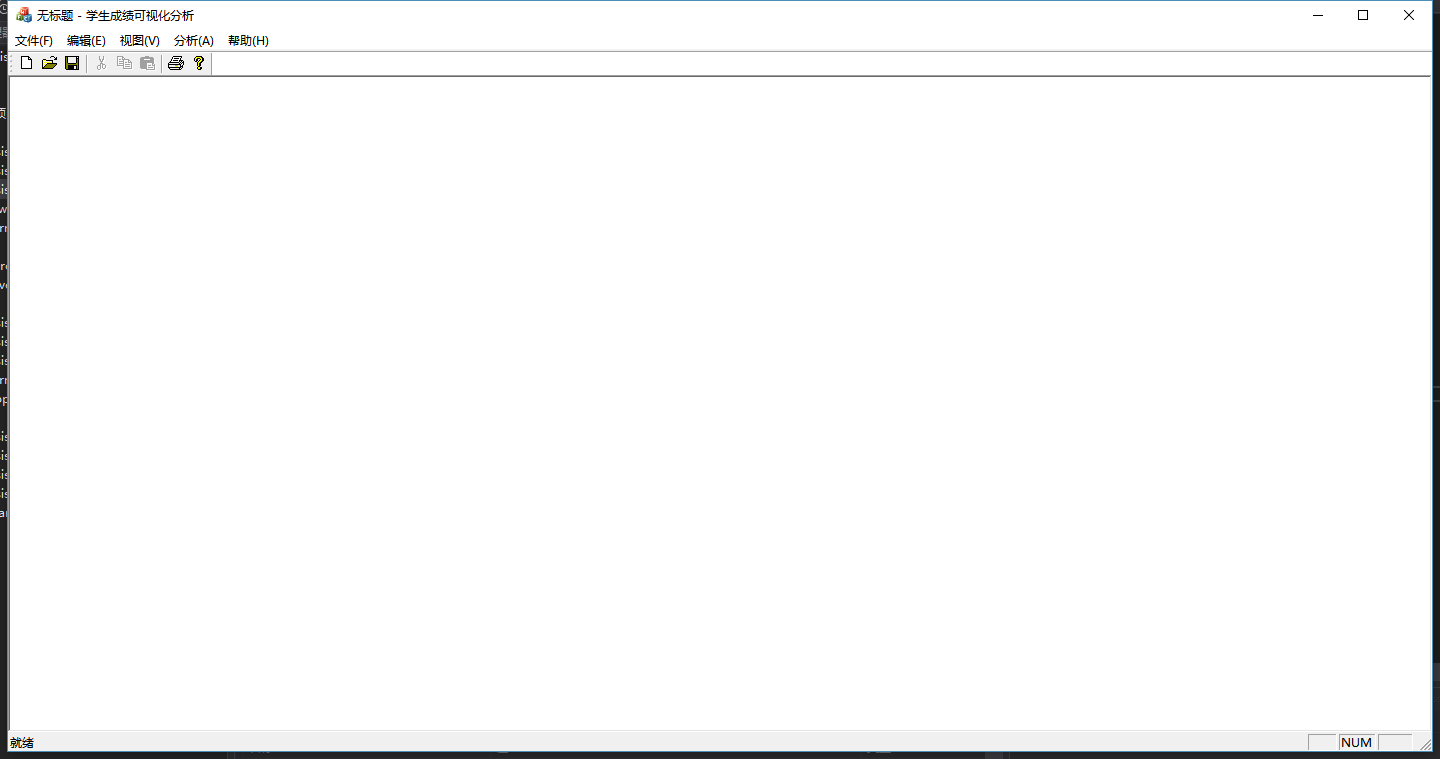
再次打开，打开文件“学生列表.list”：



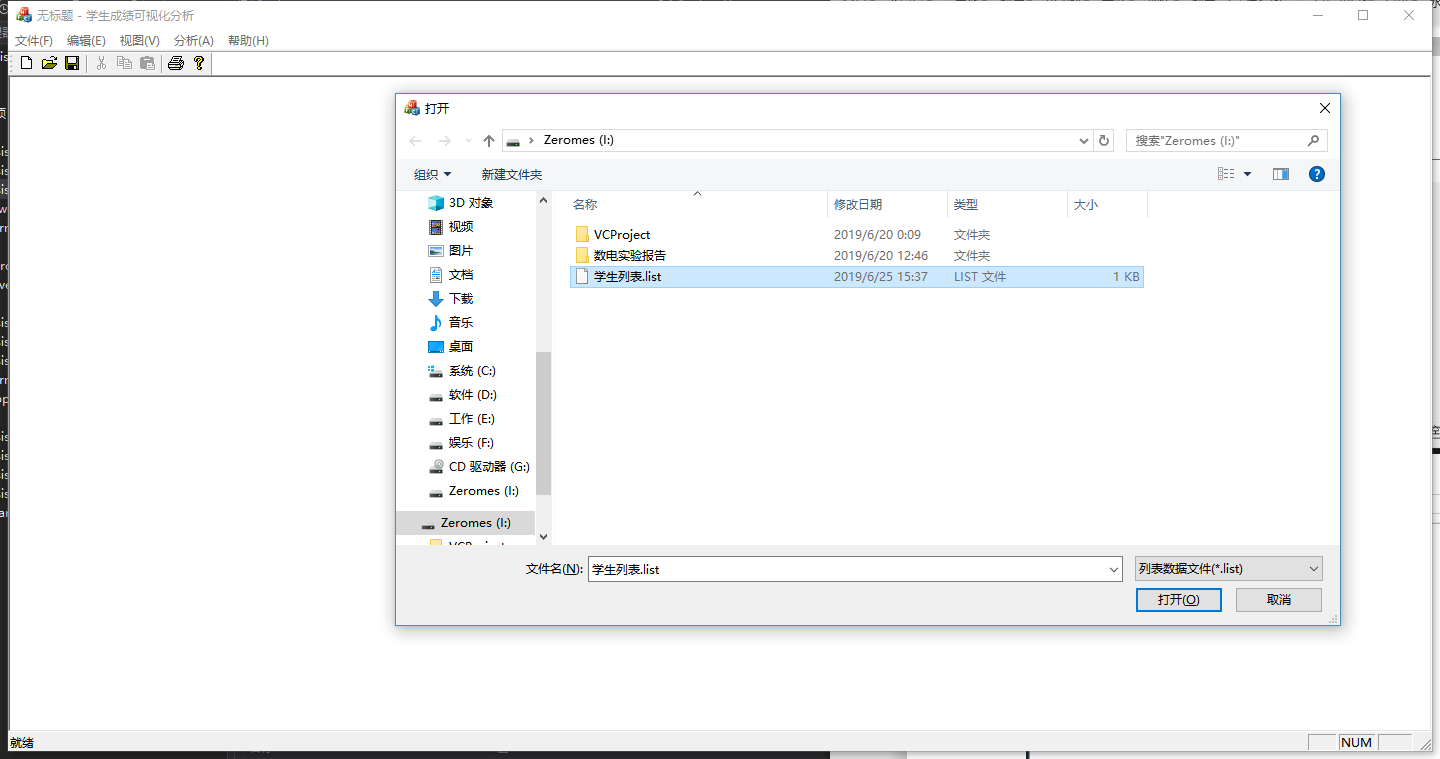
可以看到上次的修改被成功保存：



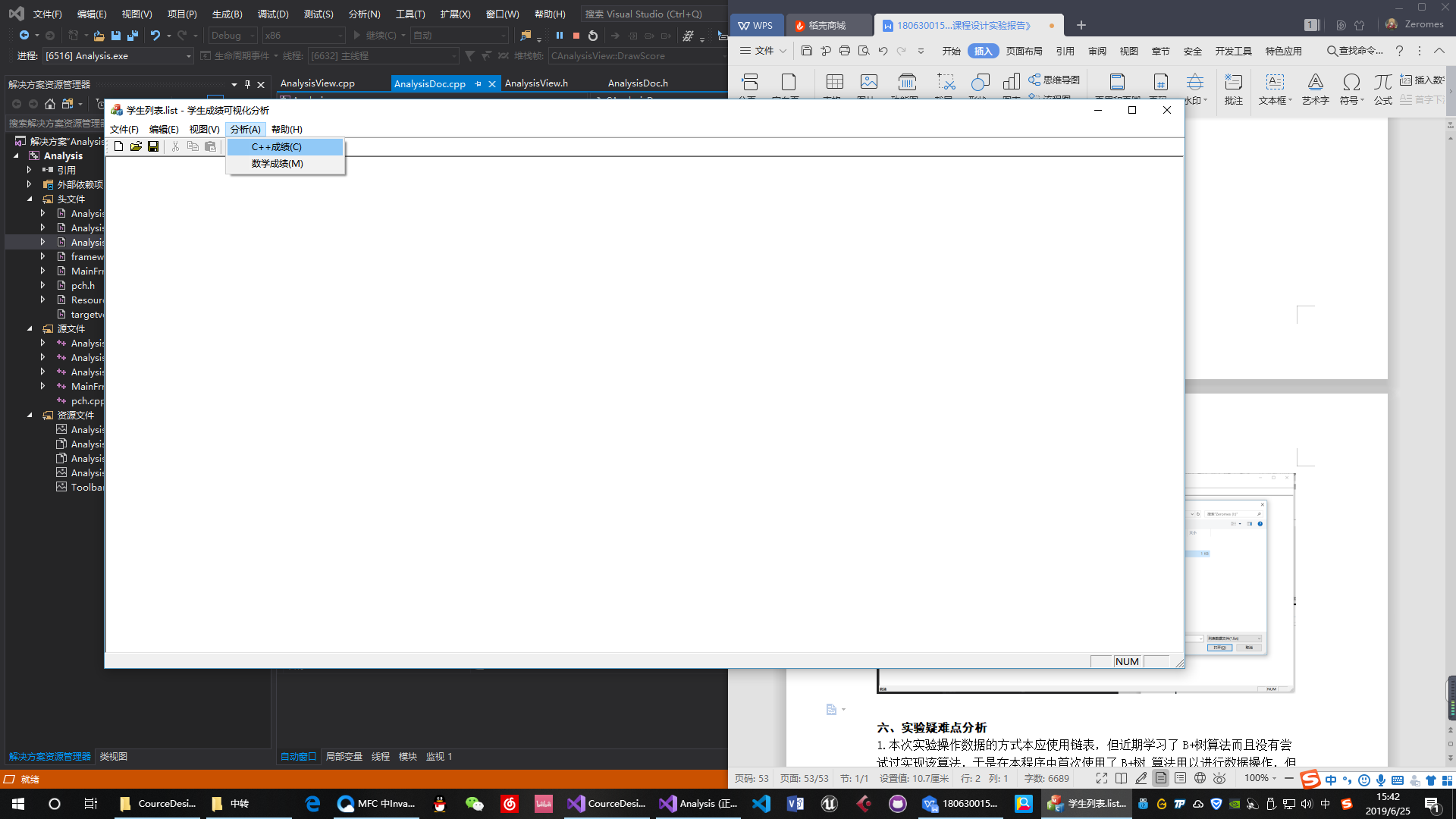
打开绘制直方图程序：



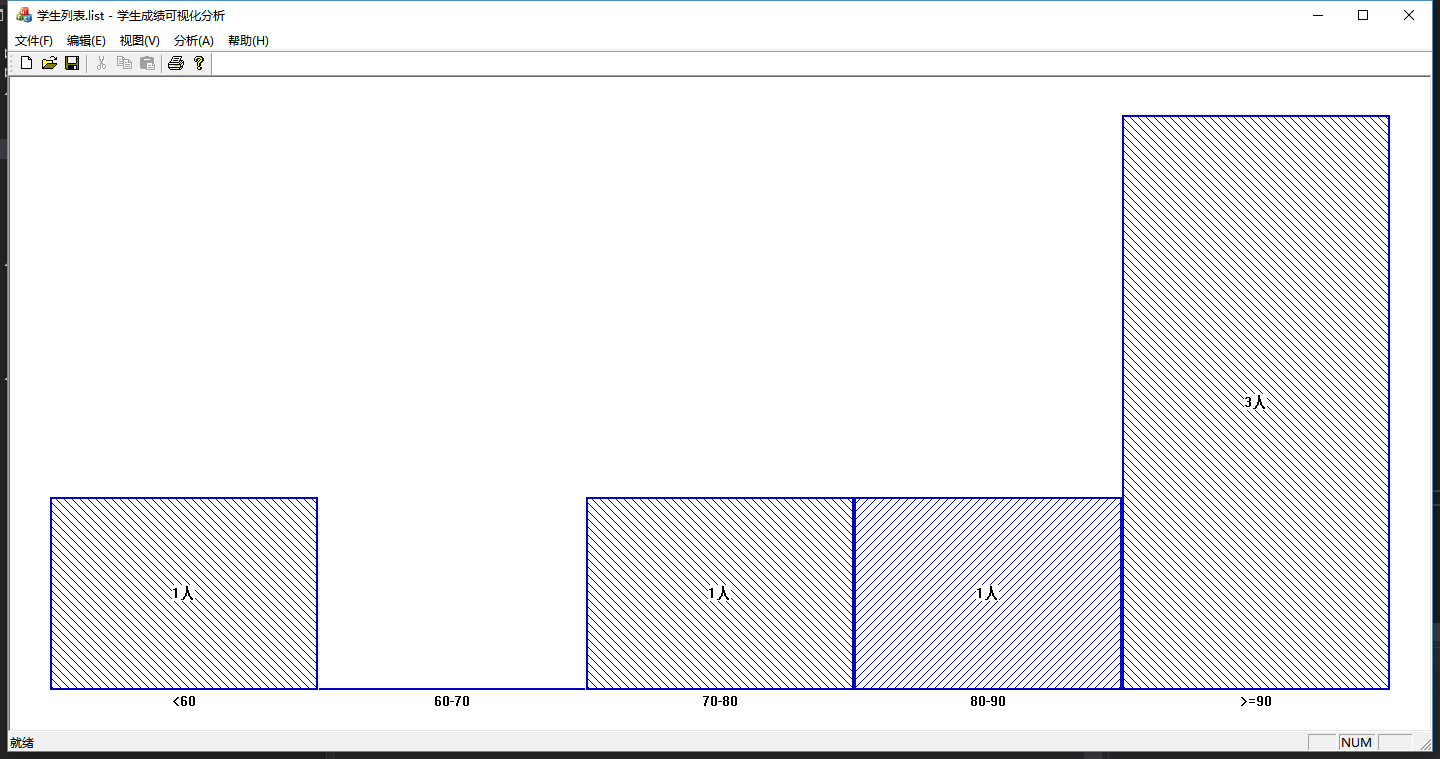
打开“学生列表.list”：



在菜单栏中找到分析->C++成绩



可以根据刚才输入的C++成绩绘制直方图：



选择数学成绩则绘制数学成绩的图形：

