

第二部分：实验过程记录（可加页）（包括实验原始数据记录，实验现象记录，实验过程发现的问题等）

1. 建立基于对话框的图形界面

（1）界面展示

（2）对“List Control”控件进行初始化

```
152 list1.SetExtendedStyle(LVS_EX_GRIDLINES | LVS_EX_HEADERDRAGDROP | LVS_EX_FULLROWSELECT);
153 //拓展list control的格式
154 //绘制网格线，表头可以拖拽，可以选择整行或者整列
155 CString col[8] = { _T("姓名"), _T("学号"), _T("专业"), _T("模电"), _T("概率论"), _T("C++"), _T("总分"), _T("平均分") };
156 RECT r; //rect对象用来存储一个矩形框的左上角坐标、宽度和高度。
157 list1.GetClientRect(&r); //显示矩形框调用函数要加取地址符号
158 for (int i = 0; i < 8; i++)
159 {
160     list1.InsertColumn(i, col[i], LVCFMT_CENTER, (r.right - r.left) / 8);
161 }
162 list1.SetBkColor(RGB(199, 237, 233)); //设置列表框背景颜色
163 list1.SetTextColor(RGB(18, 53, 85)); //字体颜色
164 list1.SetTextBkColor(RGB(175, 215, 237)); //文字背景色
165 return TRUE; // 除非将焦点设置到控件，否则返回 TRUE
166 }
```

2. 密码登录功能

（1）在主程序中初始化弹出登陆界面

```
149 Login_1 Login;
150 Login.DoModal();
```

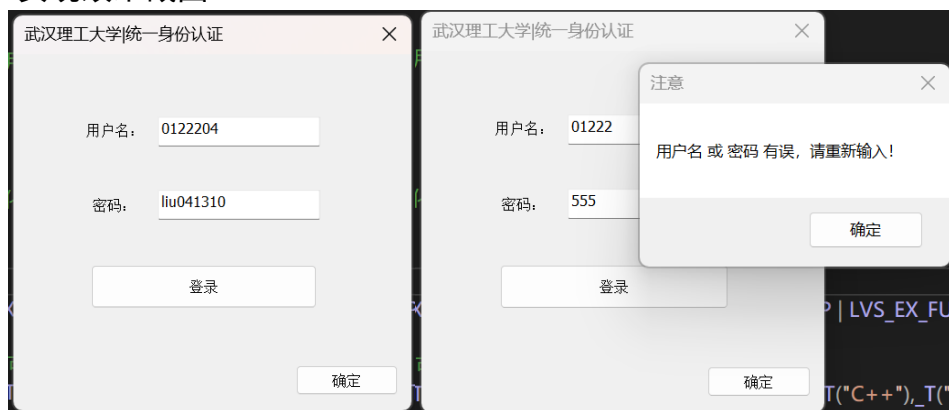
（2）在 Login_1.cpp 中对“登录”（Button1）控件进行定义

```

43 void Login_1::OnBnClickedButton1()
44 {
45     // TODO: 在此添加控件通知处理程序代码
46
47     GetDlgItemText(IDC_EDIT1, m_strName);
48     GetDlgItemText(IDC_EDIT2, m_password);
49     CString username = L"0122204"; // 用户名
50     CString password = L"liu041310"; // 密码
51     if (m_strName == L"" || m_password == L"")
52     {
53         MessageBox(L"用户名 或 密码 不能为空! ", L"注意");
54         GetDlgItem(IDC_EDIT1)->SetFocus();
55     }
56     else if (m_strName.Compare(username) != 0 || m_password.Compare(password) != 0)
57     {
58         MessageBox(L"用户名 或 密码 有误, 请重新输入! ", L"注意");
59         SetDlgItemText(IDC_EDIT1, L"");
60         SetDlgItemText(IDC_EDIT2, L"");
61         GetDlgItem(IDC_EDIT1)->SetFocus();
62     }
63     else if (m_strName.Compare(username) == 0 || m_password.Compare(password) == 0)
64     {
65         CDialogEx::OnOK();
66     }
67 }

```

(3) 实现效果截图



3. 录入学生信息功能

3.1 录入界面展示

3.2 在主程序中对“录入”（Button1）控件进行定义，弹出录入窗口

```
308 void CMFCApplication6experiment3Dlg::OnBnClickedButton1()
309 {
310     AddIn a;
311     a.DoModal();
312 }
```

3.3 建立 AddIn.h 文件

- (1) 在 AddIn.h 中定义 student 基类；
- (2) 定义 CString n[8]用于保存学生各项信息；
- (3) 创建 CMFCApplication6experiment3Dlg*Dlg1，保存父级对话框指针；

```
22 class student
23 {
24 public:
25     CString n[8];
26     CMFCApplication6experiment3Dlg* Dlg1 = NULL; //创建父级对话框指针
27     student(CMFCApplication6experiment3Dlg* dlg);
28     void set();
29 };
30
31 class AddIn : public CDialogEx
32 {
33     DECLARE_DYNAMIC(AddIn)
34
35 public:
36     AddIn(CWnd* pParent = nullptr); // 标准构造函数
37     virtual ~AddIn();
38     //AddIn(CMFCApplication6experiment3Dlg*dlg);
39     CMFCApplication6experiment3Dlg* Dlg = NULL;
```

3.4 在 AddIn.c 文件中

- (1) 构造 student()函数，将父级对话框指针传入；
- (2) 构造 void set()成员函数，将 n[8]插入到父级对话框的 CListCtrl 控件。

```
11 // AddIn 对话框
12
13 student::student(CMFCApplication6experiment3Dlg* dlg)
14 {
15     Dlg1 = dlg;
16 }
17
18 void student::set()
19 {
20     for (int b = 0; b < 8; b++)
21     {
22         Dlg1->list1.SetItemText(Dlg1->list1.GetItemText() - 1, b, n[b]);
23         //添加每行中的每个元素
24         //从子窗口读出的东西，传到父窗口控件上
25     }
26 }
```

3.5 在 AddIn.c 中对“确认录入”（Button1）控件进行定义

```
62 void AddIn::OnBnClickedButton1() {
63     student stu(Dlg);
64     a_Name.GetWindowTextW(stu.n[0]);
65     a_ID.GetWindowTextW(stu.n[1]);
66     a_Major.GetWindowTextW(stu.n[2]);
67     a_Elc.GetWindowTextW(stu.n[3]);
68     a_Prob.GetWindowTextW(stu.n[4]);
69     a_Cpp.GetWindowTextW(stu.n[5]);
70
71     for (int g = 0; g < 6; g++) {
72         if (stu.n[g].IsEmpty()) {
73             MessageBox(L"请输入完整信息! ", L"Error!");
74             return;
75         }
76     }
77
78     for (int i = 0; i < Dlg->list1.GetItemCount(); i++)
79     {
80         if (Dlg->list1.GetItemText(i, 1) == stu.n[1])
81         {
82             CString str("学号为");
83             str += stu.n[1]; str += "的学生信息已录入! ";
84             MessageBox(str, L"提示!");
85             AddIn::OnOK();
86             return;
87         }
88     }
89     Dlg->list1.InsertItem(Dlg->list1.GetItemCount(), L"");
90     // 从子窗口获取内容传到父类控件上
91     double x = _ttof(stu.n[3]) + _ttof(stu.n[4]) + _ttof(stu.n[5]);
92     double y = x / 3;
93     stu.n[6].Format(L"%f", x); stu.n[7].Format(L"%f", y);
94     stu.set();
95     MessageBox(L"录入成功", L"提示!");
96     Dlg->save = false;
97     AddIn::OnOK();
98 }
99
100 BOOL AddIn::OnInitDialog()
101 {
102     CDialogEx::OnInitDialog();
103     Dlg = (CMFCApplication6experiment3Dlg*)AfxGetMainWnd();
104     // 换取主窗口指针, 不加会冲突
105     return TRUE;
106 }
107
108 BOOL AddIn::PreTranslateMessage(MSG* pMsg)
109 {
110     // TODO: 在此添加专用代码和/或调用基类
111     if (pMsg->message == WM_KEYDOWN)
112     {
113         switch (pMsg->wParam)
114         {
115             case VK_RETURN: // 按回车表示确定
116                 OnBnClickedButton1();
117                 return TRUE;
118         }
119     }
120     return CDialog::PreTranslateMessage(pMsg);
121 }
```

3.6 实现效果截图



4. 删除学生信息功能

(1) 在主函数中对“删除”（Button2）控件进行定义

```
315 void CMFCApplication6experiment3Dlg::OnBnClickedButton2()
316 {
317     // TODO: 在此添加控件通知处理程序代码
318     int count = list1.GetSelectedCount();
319     POSITION pos = list1.GetFirstSelectedItemPosition();
320     if (pos)
321     {
322         int p = 0;
323         while (pos)
324         {
325             int k = list1.GetNextSelectedItem(pos);
326             list1.DeleteItem(k - p); p++;
327         }
328         save = false;
329     }
330     else
331         MessageBox(L"请选择需要删除的信息!", L"Error!");
332 }
```

(2) 实现效果截图



5. 修改学生信息功能

(1) 在主函数中对创建函数，将 list1 列表被选中的内容传递到右侧的“修改”编辑框的“Edit control”控件中

```
336 void CMFCApplication6experiment3Dlg::OnNMClickList1(NMHDR* pNMHDR, LRESULT* pResult)//显示
337 {
338     LPNMITEMACTIVATE pNMItemActivate = reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);
339     // TODO: 在此添加控件通知处理程序代码
340     *pResult = 0;
341     int nIndex = list1.GetSelectionMark();//选中行
342     //获取索引指定内容
343     CString name = list1.GetItemText(nIndex, 0);
344     CString number = list1.GetItemText(nIndex, 1);
345     CString major = list1.GetItemText(nIndex, 2);
346     CString Elc = list1.GetItemText(nIndex, 3);
347     CString Prob = list1.GetItemText(nIndex, 4);
348     CString Cpp = list1.GetItemText(nIndex, 5);
349     Vmain_name = name; //把获得的内容传到编辑框
350     Vmain_ID = number;
351     Vmain_major = major;
352     Vmain_Elc = Elc;
353     Vmain_Prob = Prob;
354     Vmain_Cpp = Cpp;
355     UpdateData(FALSE); //变量更新到编辑框
356 }
```

(2) 在主函数中对“确认修改”（Button4）控件进行定义

```
362 void CMFCApplication6experiment3Dlg::OnBnClickedButton4()//更改
363 {
364     // TODO: 在此添加控件通知处理程序代码
365     if (MessageBox(L"确定修改吗! ", L"提示", MB_YESNO) == IDNO)
366         //消息弹出框，点击“否”则退出
367         return;
368     else//如果点击“是”则进行程序
369     {
370         UpdateData(TRUE); //编辑框到变量
371         int nIndex = list1.GetSelectionMark();//获取整行
372         list1.SetItemText(nIndex, 0, Vmain_name); //设置字符到编辑框
373         list1.SetItemText(nIndex, 1, Vmain_ID);
374         list1.SetItemText(nIndex, 2, Vmain_major);
375         list1.SetItemText(nIndex, 3, Vmain_Elc);
376         list1.SetItemText(nIndex, 4, Vmain_Prob);
377         list1.SetItemText(nIndex, 5, Vmain_Cpp);
378         float elc = _ttoi(Vmain_Elc); //cstring转float
379         float prob = _ttoi(Vmain_Prob);
380         float cpp = _ttoi(Vmain_Cpp);
381         float sum = elc + prob + cpp; //用来计算
382         Vmain_summ.Format(_T("%f"), sum); //float转cstring
383         list1.SetItemText(nIndex, 6, Vmain_summ);
384         float aver = sum / 3.0;
385         Vmain_average.Format(_T("%f"), aver);
386         list1.SetItemText(nIndex, 7, Vmain_average);
387     }
388 }
```

(3) 实现效果截图



6. 查找学生信息功能

(1) 在主函数中对“查找”（Button3）控件进行定义

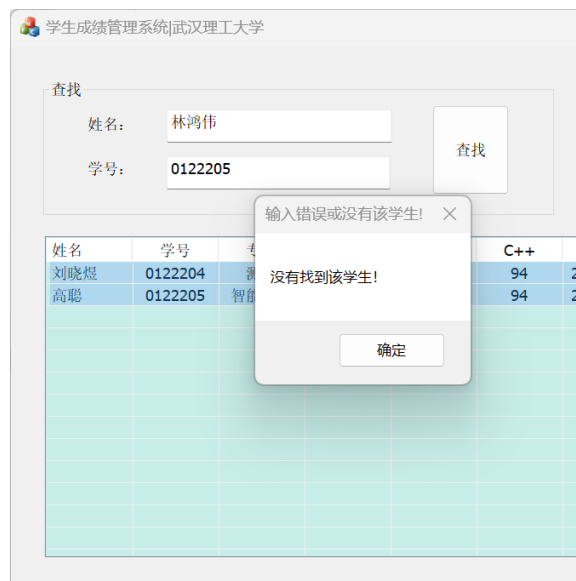
```
228 void CMFCApplcation6experiment3Dlg::OnBnClickedButton3()    //查找
229 {
230     CString fname, fnumber; int index = list1.GetItemCount();
231     int q = 0; //用来做flag
232
233     for (int i = 0; i < index; i++)//初始化各行信息的选择状态，设置为未选择。
234     {
235         list1.SetItemState(i, 0, LVIS_SELECTED | LVIS_FOCUSED);
236     }
237     Fname.GetWindowTextW(fname); Fnumber.GetWindowTextW(fnumber);
238     //获取需要搜索的姓名和学号，传入fname和fnumber。
239     CString temp;
240     //判断fname和fnumber是否为空，并做出对应操作。用int q的值判断是否找到符合信息的学生。
241     if (fname != "" && fnumber != "")
242     {
243         for (int i = 0; i < index; i++)
244         {
245             temp = list1.GetItemText(i, 0);
246             temp.TrimLeft(); //去掉左面空格//防止非定长函数对结果进行印象
247             if (temp == fname && list1.GetItemText(i, 1) == fnumber)
248             {
249                 list1.SetFocus();
250                 list1.SetItemState(i, LVIS_SELECTED | LVIS_FOCUSED, LVIS_SELECTED | LVIS_FOCUSED);
251                 list1.EnsureVisible(i, true); q = 1;
252                 CString name, number, major, Elc, Prob, Cpp;
253                 name = list1.GetItemText(i, 0); number = list1.GetItemText(i, 1); major = list1.GetItemText(i, 2);
254                 Elc = list1.GetItemText(i, 3); Prob = list1.GetItemText(i, 4); Cpp = list1.GetItemText(i, 5);
255                 Vmain_name = name; Vmain_ID = number; Vmain_major = major;
256                 Vmain_Elc = Elc; Vmain_Prob = Prob; Vmain_Cpp = Cpp;
257                 UpdateData(false);
258             }
259         }
260     }
```

```

261     else if (fname != "" && fnumber == "")
262     {
263         for (int i = 0; i < index; i++)
264         {
265             temp = list1.GetItemText(i, 0);
266             temp.TrimLeft(); //去掉左面空格
267             if (temp == fname)
268             {
269                 list1.SetFocus(); //获得焦点
270                 list1.SetItemState(i, LVIS_SELECTED | LVIS_FOCUSED, LVIS_SELECTED | LVIS_FOCUSED);
271                 //选中第一行, 当鼠标点击列表光标悬浮在上面
272                 list1.EnsureVisible(i, true); q = 1; //确保可视化
273                 CString name, number, major, Elc, Prob, Cpp;
274                 name = list1.GetItemText(i, 0); number = list1.GetItemText(i, 1); major = list1.GetItemText(i, 2);
275                 Elc = list1.GetItemText(i, 3); Prob = list1.GetItemText(i, 4); Cpp = list1.GetItemText(i, 5);
276                 Vmain_name = name; Vmain_ID = number; Vmain_major = major;
277                 Vmain_Elc = Elc; Vmain_Prob = Prob; Vmain_Cpp = Cpp;
278                 UpdateData(false);
279             }
280         }
281     }
282     else if (fname == "" && fnumber != "")
283     {
284         for (int i = 0; i < index; i++)
285         {
286             if (list1.GetItemText(i, 1) == fnumber)
287             {
288                 list1.SetFocus();
289                 list1.SetItemState(i, LVIS_SELECTED | LVIS_FOCUSED, LVIS_SELECTED | LVIS_FOCUSED);
290                 list1.EnsureVisible(i, true); q = 1;
291                 CString name, number, major, Elc, Prob, Cpp;
292                 name = list1.GetItemText(i, 0); number = list1.GetItemText(i, 1); major = list1.GetItemText(i, 2);
293                 Elc = list1.GetItemText(i, 3); Prob = list1.GetItemText(i, 4); Cpp = list1.GetItemText(i, 5);
294                 Vmain_name = name; Vmain_ID = number; Vmain_major = major;
295                 Vmain_Elc = Elc; Vmain_Prob = Prob; Vmain_Cpp = Cpp;
296                 UpdateData(false);
297             }
298         }
299     }
300     else if (fname == "" && fnumber == "")
301     {
302         MessageBox(L"请输入查找信息! ", L"输入错误");
303         return;
304     }
305     if (q == 0)
306         MessageBox(L"没有找到该学生! ", L"输入错误或没有该学生!");
307 }

```

(2) 实现效果截图



7. 保存文件功能

(1) 在主函数中对“另存为”（Button5）控件进行定义，使文件保存在 D 盘，并命名为“学生成绩记录”

```
430 void CMFCApplication6experiment3Dlg::OnBnClickedButton5()    //另存为
431 {
432     CFile s(L"D://学生成绩记录.txt", CFile::modeCreate | CFile::modeReadWrite);
433     //如果文件不存在，创建一个新文件。 如果该文件已存在，它将被覆盖，并最初设置为长度为零
434     //以读写方式打开文件
435     int index = list1.GetItemCount();
436     for (int i = 0; i < index; i++)
437     {
438         for (int j = 0; j < 8; j++)
439         {
440             CString a;
441             if (j != 7)
442                 a = list1.GetItemText(i, j) + '\t'; //从控件获取信息
443             if (j == 7 && i != index - 1)
444             {
445                 a = list1.GetItemText(i, j) + '\t'; //回退一格
446                 a += "\r\n"; //回车+换行
447             }
448             if (j == 7 && i == index - 1)
449                 a = list1.GetItemText(i, j) + '\t';
450             s.Write(a, a.GetLength()); //统计cstring类的长度
451         }
452     }
453     save = true;
454     MessageBox(L"已保存!", L"提示");
```

(2) 实现效果截图

