

一、课程设计题目及内容

- (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。

一、功能模块的设计分析及算法描述

➤ 学生管理系统

· 登录对话框

程序开始界面为登录界面, 主要由两个编辑框(账号、密码)、一个登录按钮组成。点击登录按钮, 程序获取输入账号与密码进行判断, 正确则创建管理主界面的模态对话框, 并将登录对话框进行隐藏。关闭管理主界面后也将关闭登录对话框。

· 管理主界面

管理主界面由八个按钮、八个编辑框、八个单选框和一个 list control 控件组成。按钮有“排序”、“全选”、“反选”、“删除”、“修改”、“打开”、“保存”、“添加”; 编辑框包括“学号输入框”、“姓名输入框”、“班级输入框”、“高数及面向对象成绩输入框”以及“人数统计框”; 单选框有 排序依据、性别、年级 三组。

在 Student 类中包括了声明为 public 的学号、年级、班级、姓名、性别 五个 Cstring 类型的变量和高等数学成绩、面向对象成绩等两个 double 变量。

在 Management 类中 public 定义了三个 Student 指针变量, 作为三个容器用于存储添加的数据 (Pstu)、按高数排序的数据 (SortM)、按面向对象排序的数

据 (SortC)。private 声明了 Count (已经添加的人数)、Size (容器大小) 两个 int 型变量记录容器的特性。同时 Management 中还提供了对容器数据的增加 (AddData)、删除 (DeleteData)、修改 (EditData)、按高数排序 (SortDataM)、按面向对象排序 (SortDataM) 等功能。同时还提供了对数据有效性判断的函数 (IsSame、IsCorrect)

- ✧ Management 构造函数将使用 Pstu 指针开辟 100 人大小的空间作为容器，将 Count 设置为 0，Size 设置为 100，SortM SortC 指向 NULL。析构函数将三个容器清空，将 Count Size 设置为 0。
- ✧ AddData (Student obj) 该函数传入一个 Student 类型的变量，容器先对 Count 进行++操作，并判断如果容器内人数已经大于容器大小，容器自动进行扩容增加 50 个位置。最后将 Pstu[Count-1]=obj 将数据拷贝进容器。
- ✧ EditData (int n) 该函数提供删除操作，传入参数为位置。函数将该位置后面的所有数据前移一位，并对 Count 进行--。
- ✧ EditData(Student obj, int n) 该函数将传入位置的数据替换为传入的 obj。
- ✧ SortDataM() 与 SortDataC，两者都使用自身容器的指针 SortM/SortC 新建一个 Count 大小的 Student 类型数组，将 Pstu 数据复制进自身容器，然后使用冒泡排序进行排序。
- ✧ IsSame() IsCorrect() 对数据进行有效性判断，判断有无雷同学号或负数成绩。

➤ 直方图和折线图

主要在 view 中添加 OnDraw 函数，使用画刷进行直方图和折线图绘制。使用 OnOne() OnTwo() 两个函数进行读取高数成绩以及面向对象成绩。

二、程序中使用的数据及主要符号说明

※ Student类中

Num; //学号
Grade; //年级
Class; //班级
Name; //姓名
Sex; //性别
maths; //高等数学的成绩
CPP; //面向对象程序设计

※ Management 类中

Pstu; //指向无序数据数组的指针
SortM; //指向按高数成绩排序后数组的指针
SortC; //指向按面向对象成绩排序后数组的指针
Count; //用于统计容器内已经添加的人数
Size; //容器的总容量

※ 在管理系统的主对话框中（MFC 实现）

全局变量

ManagerSystem; // Management的对象
STU //用于记录list control中项目的数量，与Management类中Count保持一致
T_mathd, T_matht; //记录高数成绩输入框，平时成绩与期末考试成绩
T_cppd, T_cppt; //记录面向对象成绩输入框，平时成绩与期末考试成绩
T_sex //记录性别
T_G //记录年级

三、部分关键程序的源代码

✧ Student.h

```
//----- Student.h -----  
  
#pragma once  
#include "afxdialogex.h"  
  
//----- Student类声明 -----  
class Student  
{  
public:  
    Student();  
    CString Num; //学号  
    CString Grade;  
    CString Class;  
    CString Name;  
    CString Sex;  
    double maths; //高等数学的成绩  
    double CPP; //面向对象程序设计  
};  
  
Student::Student()  
{  
    Num = '\0';  
    Grade = '\0';  
    Sex = '\0';  
    Class = '\0';  
    Name = '\0';  
    maths = 0;  
    CPP = 0;  
}
```

✧ Management.h

```
//----- Management.h -----  
  
#pragma once  
#include "afxdialogex.h"  
#include "Student.h"  
#include <fstream>  
#include <string>  
#include "stdafx.h"  
  
using namespace std;
```

```

class Management
{
public:
    Management(); //构造
    ~Management(); //析构

    int GetCount(); //返回容器内已经添加的人数

    //-----对容器的操作-----
    void AddData(Student); //学生成绩数据的增加
    void DeleteData(int n); //学生成绩数据的删除
    void EditData(Student obj, int n); //学生成绩数据的修改
    void SortDataM(); //学生成绩数据的排序(按高数)
    void SortDataC(); //学生成绩数据的排序(按面向对象)

    //-----对数据的有效性判断-----
    BOOL IsSame(Student obj); //判断容器内是否已经存在相同学号的人
    BOOL IsCorrect(Student obj); //判断输入数据的有效性

    //-----容器数据成员-----
public:
    Student* Pstu; //指向无序数据数组的指针
    Student* SortM; //指向按高数成绩排序后数组的指针
    Student* SortC; //指向按面向对象成绩排序后数组的指针
private:
    int Count; //用于统计容器内已经添加的人数
    int Size; //容器的总容量
};

```

✧ Management.cpp

```

//-----Management.cpp-----
#include "Management.h"
Management::Management()
{
    Size = 100;
    Pstu = new Student[Size];
    SortM = NULL;
    SortC = NULL;
    Count = 0;
}

```

```

Management::~Management()
{
    delete[] Pstu;
    delete[] SortC;
    delete[] SortM;
    Size = 0;
    Count = 0;
}

int Management::GetCount()
{
    return Count;
}

//-----对容器的操作-----
void Management::AddData(Student obj)
{
    Count++;
    if (Count > Size)//判断容器空间是否足够
    {
        Student* temp = new Student[Size + 50];
        for (int i = 0; i < Size; i++)
            temp[i] = Pstu[i];
        Size += 50;
        delete[] Pstu;
        Pstu = temp;
    }
    Pstu[Count - 1] = obj;
}

void Management::DeleteData(int n)
{
    for (int i = n; i < Count; i++)
        Pstu[i] = Pstu[i + 1];
    Count--;
}

void Management::EditData(Student obj, int n)
{
    Pstu[n] = obj;
}

```

```

void Management::SortDataM()
{
    //为按数学成绩排序的容器开辟空间，并复制原容器数据
    SortM = new Student[Count];
    for (int i = 0; i < Count; i++)
        SortM[i] = Pstu[i];

    //-----使用冒泡排序-----
    for (int i = 0; i < Count - 1; i++)
    {
        int work = 1;    //监测点，若数据本已经有序直接结束程序
        for (int k = 0; k < Count - i - 1; k++)
        {
            if (SortM[k].maths < SortM[k + 1].maths)
            {
                swap(SortM[k], SortM[k + 1]);
                work = 0;
            }
        }
        if (work)
            break;
    }
}

```

```

void Management::SortDataC()
{
    //为按C++成绩排序的容器开辟空间，并复制原容器数据
    SortC = new Student[Count];
    for (int i = 0; i < Count; i++)
        SortC[i] = Pstu[i];

    //-----使用冒泡排序-----
    for (int i = 0; i < Count - 1; i++)
    {
        int work = 1;    //监测点，若数据本已经有序直接结束程序
        for (int k = 0; k < Count - i - 1; k++)
        {
            if (SortC[k].C++ < SortC[k + 1].C++)
            {
                swap(SortC[k], SortC[k + 1]);
                work = 0;
            }
        }
        if (work)

```



```

        break;
    }
}

//-----对数据的有效性判断-----
BOOL Management::IsSame(Student obj)
{
    for (int i = 0; i < Count; i++)
    {
        if (obj.Num == Pstu[i].Num)
            return 0;
    }
    return 1;
}

BOOL Management::IsCorrect(Student obj)
{
    if (!IsSame(obj))           //对学号有效性判断
        return 0;
    if (obj.maths < 0 || obj.CPP < 0) //对成绩有效性判断
        return 0;
    else
        return 1;
}

```

管理系统 MFC 实现

➤ 登录对话框

✧ STUSMDlg.cpp//登录对话框部分主要实现

```

//-----STUSMDlg.cpp-----

void CSTUSMDlg::OnPaint()//实现登录框背景图
{
    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
    {
        //-----设置背景图片-----
        CPaintDC dc(this);
        CRect rect;
        GetClientRect(&rect);
        CDC dcMem;
        dcMem.CreateCompatibleDC(&dc);
        CBitmap bmpBackground;
        bmpBackground.LoadBitmap(IDB_BITMAP_loginbg); //对话框的背景图片

        BITMAP bitmap;
        bmpBackground.GetBitmap(&bitmap);
        CBitmap *pbmpOld = dcMem.SelectObject(&bmpBackground);
        dc.StretchBlt(0, 0, rect.Width(), rect.Height(), &dcMem, 0, 0, bitmap.bmWidth,
        bitmap.bmHeight, SRCCOPY);
    }
}

//输入账号、密码部分判断
//Account
CString ACC = _T("32116160100");
//Password
CString PAS = _T("gzhuacs");
//错误次数
int COUNT = 0;
void CSTUSMDlg::OnBnClickedMfcbuttonLogin()//点击登录按钮时
{
    CString account, password;
    GetDlgItemText(IDC_EDIT_Account, account);
    GetDlgItemText(IDC_EDIT_Password, password);

    //-----登录流程判断逻辑-----
    if (COUNT > 3)
    {
        MessageBox(_T("错误次数超过三次，程序退出！"), _T("提示"), MB_ICONWARNING);
        (CDialog*)AfxGetMainWnd()->DestroyWindow();
    }
}

```

```

    }
    if (account == ACC&&password == PAS)
    {
        ShowWindow(SW_HIDE);
        MainDlg main; //密码正确进入系统
        main.DoModal();
    }
    else if (account == "" || password == "")
        MessageBox(_T("账号或密码不能为空"), _T("提示"), MB_ICONWARNING);
    else
    {
        MessageBox(_T("账号或密码错误!"), _T("提醒"), MB_ICONWARNING);
        COUNT++;
    }
}
}

```

➤ 学生管理系统主界面对话框

✧ MainDlg.cpp//管理系统主界面

```

//-----MainDlg.cpp-----

#include "stdafx.h"
#include "STUSM.h"
#include "MainDlg.h"
#include "afxdialogex.h"
#include "Student.h"
#include "Management.h"

//声明全局变量
Management ManagerSystem;
CString STU = _T("0");
CString T_mathtd, T_matht;
CString T_cpptd, T_cppt;

void MainDlg::OnClose()//关闭事件
{
    //-----实现子窗口关闭时将父窗口关闭-----
    CDialog* pdlg = (CDialog*)AfxGetMainWnd();
    pdlg->DestroyWindow();
    //-----
}

```

```

        CDialogEx::OnClose();
    }

    BOOL MainDlg::OnInitDialog()//窗口初始化
    {
        CDialogEx::OnInitDialog();

        //-----设置窗口标题-----
        CString Title = _T("STUSM 学生成绩管理系统-当前用户: 计科212 冯韦铭 32116160100");
        SetWindowText(Title);

        //-----列表(表头)初始化-----
        m_List.InsertColumn(0, _T("学号"), LVCFMT_CENTER, 150);
        m_List.InsertColumn(1, _T("年级"), LVCFMT_CENTER, 80);
        m_List.InsertColumn(2, _T("班级"), LVCFMT_CENTER, 80);
        m_List.InsertColumn(3, _T("姓名"), LVCFMT_CENTER, 80);
        m_List.InsertColumn(4, _T("性别"), LVCFMT_CENTER, 80);
        m_List.InsertColumn(5, _T("高等数学"), LVCFMT_CENTER, 100);
        m_List.InsertColumn(6, _T("面向对象程序设计"), LVCFMT_CENTER, 135);

        //-----
        m_List.SetExtendedStyle(LVS_EX_FULLROWSELECT | LVS_EX_CHECKBOXES);

        SetDlgItemText(IDC_EDIT_STU, STU);
        //更新学生人数
        STU.Format(_T("%d"), m_List.GetItemCount());
        SetDlgItemText(IDC_EDIT_STU, STU);

        return TRUE; // return TRUE unless you set the focus to a control
        // 异常: OCX 属性页应返回 FALSE
    }

    void MainDlg::OnBnClickedButtonAll()//点击“全选”按钮
    {
        // TODO: 在此添加控件通知处理程序代码
        for (int i = 0; i < m_List.GetItemCount(); i++)
        {
            m_List.SetCheck(i, TRUE);
        }
    }

```

```

void MainDlg::OnBnClickedButtonOthers()//点击“反选”按钮
{
    for (int i = 0; i < m_List.GetItemCount(); i++)
    {
        m_List.SetCheck(i, !m_List.GetCheck(i));
    }
}

```

```

CString T_G;//表示年级
//点击年级组的单选框
void MainDlg::OnBnClickedRadioGra1()
{
    T_G = _T("大一");
}
void MainDlg::OnBnClickedRadioGra2()
{
    T_G = _T("大二");
}
void MainDlg::OnBnClickedRadioGra3()
{
    T_G = _T("大三");
}
void MainDlg::OnBnClickedRadioGra4()
{
    T_G = _T("大四");
}

```

```

//点击性别组的单选框
CString T_sex;
void MainDlg::OnBnClickedRadioMan()
{
    T_sex = _T("男");
}
void MainDlg::OnBnClickedRadioWoman()
{
    T_sex = _T("女");
}

```

```

void MainDlg::OnBnClickedMfcbuttonAdd()///点击“添加”按钮
{
    CString T_name, T_num, T_class, T_temp;
    Student temp;

    //-----从编辑框获取数据-----
    GetDlgItemText(IDC_EDIT_Name, T_name);
    GetDlgItemText(IDC_EDIT_Num, T_num);
    GetDlgItemText(IDC_EDIT_Class, T_class);

    GetDlgItemText(IDC_EDIT_Mathsd, T_mathd);
    GetDlgItemText(IDC_EDIT_Mathst, T_matht);
    double maths = 0.4* _tcsd(T_mathd, NULL) + 0.6*_tcsd(T_matht, NULL);

    GetDlgItemText(IDC_EDIT_Cppd, T_cppd);
    GetDlgItemText(IDC_EDIT_Cppt, T_cppt);
    double cpp = 0.4* _tcsd(T_cppd, NULL) + 0.6*_tcsd(T_cppt, NULL);

    //-----判断是否获取了空数据-----
    if (T_name == "" && T_num && T_cppd && T_cppt && T_mathd && T_matht)
    {
        MessageBox(_T("数据不能为空！请检查输入"), _T("提示"), MB_ICONWARNING);
        return;
    }

    temp.Num = T_num;
    temp.Grade = T_G;
    temp.Name = T_name;
    temp.Sex = T_sex;
    temp.Class = T_class;
    temp.CPP = cpp;
    temp.maths = maths;

    //将数据检查合理性后放进Managerment的对象ManagerSystem内
    if (ManagerSystem.IsCorrect(temp))
    {
        //在容器内添加数据
        ManagerSystem.AddData(temp);
        int nCount = m_List.GetItemCount();

        //在界面列表控件中添加数据
        m_List.InsertItem(nCount, T_num);
        m_List.SetItemText(nCount, 1, T_G);
        m_List.SetItemText(nCount, 2, T_class);
    }
}

```

```

        m_List.SetItemText(nCount, 3, T_name);
        m_List.SetItemText(nCount, 4, T_sex);
        //将浮点型转为cstring型
        T_temp.Format(_T("%.1f"), maths);
        m_List.SetItemText(nCount, 5, T_temp);
        T_temp.Format(_T("%.1f"), cpp);
        m_List.SetItemText(nCount, 6, T_temp);

        //更新学生人数
        STU.Format(_T("%d"), nCount+1);
        SetDlgItemText(IDC_EDIT_STU, STU);
    }
    else
    {
        MessageBox(_T("学号雷同或成绩输入错误！请检查输入"), _T("提示"),
        MB_ICONWARNING);
    }
}

void MainDlg::OnBnClickedButtonEdit()//点击“编辑”按钮
{
    Student temp;
    int i;//选中位置
    for (i = 0; i < m_List.GetItemCount(); i++)//获取选中位置
    {
        if (m_List.GetCheck(i))
            break;
    }
    if (i == m_List.GetItemCount())//若无选中位置直接结束
    {
        return;
    }
    CString T_name, T_num, T_class, T_temp;
    //-----从编辑框获取数据-----
    GetDlgItemText(IDC_EDIT_Name, T_name);
    GetDlgItemText(IDC_EDIT_Num, T_num);
    GetDlgItemText(IDC_EDIT_Class, T_class);

    GetDlgItemText(IDC_EDIT_Mathsd, T_mathd);
    GetDlgItemText(IDC_EDIT_Matht, T_matht);
    double maths = 0.4 * _tctod(T_mathd, NULL) + 0.6 * _tctod(T_matht, NULL);
    GetDlgItemText(IDC_EDIT_Cppd, T_cppd);

```



```

GetDlgItemText(IDC_EDIT_Cppt, T_cppt);
double cpp = 0.4 * _tctod(T_cppt, NULL) + 0.6 * _tctod(T_cppt, NULL);

//-----判断是否获取了空数据-----
if (T_name == "" && T_num && T_cppt && T_cppt && T_matht && T_matht)
{
    MessageBox(_T("数据不能为空！请检查输入"), _T("提示"), MB_ICONWARNING);
    return;
}

//获取数据放入对象
temp.Num = T_num;
temp.Grade = T_G;
temp.Name = T_name;
temp.Sex = T_sex;
temp.Class = T_class;
temp.CPP = cpp;
temp.maths = maths;

//调用ManagerSystem成员函数放入容器
ManagerSystem.EditData(temp, i);

m_List.DeleteItem(i);
m_List.InsertItem(i, T_num);
m_List.SetItemText(i, 1, T_G);
m_List.SetItemText(i, 2, T_class);
m_List.SetItemText(i, 3, T_name);
m_List.SetItemText(i, 4, T_sex);
//将浮点型转为cstring型
T_temp.Format(_T("%.1f"), maths);
m_List.SetItemText(i, 5, T_temp);
T_temp.Format(_T("%.1f"), cpp);
m_List.SetItemText(i, 6, T_temp);
}

```

```

void MainDlg::OnBnClickedButtonDelete()//点击“删除”按钮
{
    for (int i = 0; i < m_List.GetItemCount(); i++)
    {
        if (m_List.GetCheck(i))
        {
            ManagerSystem.DeleteData(i);
            m_List.DeleteItem(i);
            i--;
        }
    }
}

```



```

    }
}
//更新学生人数
int nCount = m_List.GetItemCount();
STU.Format(_T("%d"), nCount);
SetDlgItemText(IDC_EDIT_STU, STU);
}

int s; //排序依据单选框
void MainDlg::OnBnClickedRadiomaths()
{
    s = 0;
}

void MainDlg::OnBnClickedRadioCpp()
{
    s = 1;
}

void MainDlg::OnBnClickedButtonSort() //点击“排序”按钮
{
    m_List.DeleteAllItems();
    int nCount = ManagerSystem.GetCount();
    CString temp;
    switch (s)
    {
    case 0:
        ManagerSystem.SortDataM(); //调用ManagerSystem成员函数排序

        for (int i = 0; i < nCount; i++) //排序完成后刷新control list
        {
            m_List.InsertItem(i, ManagerSystem.SortM[i].Num);
            m_List.SetItemText(i, 1, ManagerSystem.SortM[i].Grade);
            m_List.SetItemText(i, 2, ManagerSystem.SortM[i].Class);
            m_List.SetItemText(i, 3, ManagerSystem.SortM[i].Name);
            m_List.SetItemText(i, 4, ManagerSystem.SortM[i].Sex);
            temp.Format(_T("%.1f"), ManagerSystem.SortM[i].maths);
            m_List.SetItemText(i, 5, temp);
            temp.Format(_T("%.1f"), ManagerSystem.SortM[i].CPP);
            m_List.SetItemText(i, 6, temp);
        }
        //更新学生人数
        STU.Format(_T("%d"), m_List.GetItemCount());
        SetDlgItemText(IDC_EDIT_STU, STU);
        break;

```

```

case 1:
    ManagerSystem.SortDataC(); //调用ManagerSystem成员函数排序

    for (int i = 0; i < nCount; i++) //排序完成后刷新control list
    {
        m_List.InsertItem(i, ManagerSystem.SortC[i].Num);
        m_List.SetItemText(i, 1, ManagerSystem.SortC[i].Grade);
        m_List.SetItemText(i, 2, ManagerSystem.SortC[i].Class);
        m_List.SetItemText(i, 3, ManagerSystem.SortC[i].Name);
        m_List.SetItemText(i, 4, ManagerSystem.SortC[i].Sex);
        temp.Format(_T("%.1f"), ManagerSystem.SortC[i].maths);
        m_List.SetItemText(i, 5, temp);
        temp.Format(_T("%.1f"), ManagerSystem.SortC[i].CPP);
        m_List.SetItemText(i, 6, temp);
    }
    //更新学生人数
    STU.Format(_T("%d"), m_List.GetItemCount());
    SetDlgItemText(IDC_EDIT_STU, STU);
    break;
default:
    break;
}
}

void MainDlg::OnBnClickedButtonSave() //点击“保存”按钮
{
    CFileDialog fDlg(FALSE, ".dat", "save", OFN_OVERWRITEPROMPT, "数据文件
    (*.dat)|*.dat", NULL);
    if (fDlg.DoModal() == IDOK)
    {
        CString sPath = fDlg.GetPathName();
        //创建文件对象
        CStdioFile file(sPath, CFile::modeWrite | CFile::modeCreate);
        CString temp;
        for (int i = 0; i < m_List.GetItemCount(); i++) //将control list内容全部写入
        {
            temp += m_List.GetItemText(i, 0) + "#";
            temp += m_List.GetItemText(i, 1) + "#";
            temp += m_List.GetItemText(i, 2) + "#";
            temp += m_List.GetItemText(i, 3) + "#";
            temp += m_List.GetItemText(i, 4) + "#";
            temp += m_List.GetItemText(i, 5) + "#";
            temp += m_List.GetItemText(i, 6) + "#\n";
        }
    }
}

```

```

        file.WriteString(temp);
    }

    file.Close();
}

}

void MainDlg::OnBnClickedButtonOpen()//点击“打开”按钮
{
    CFileDialog fdlg(TRUE, ".dat", "save", OFN_OVERWRITEPROMPT, "数据文件
(*.dat)|*.dat", NULL);
    if (fdlg.DoModal() == IDOK)
    {
        CStdioFile file(fdlg.GetPathName(), CFile::modeRead);
        CString temp;
        CString T_name, T_num, T_class, T_temp, T_maths, T_cpp;
        Student Stemp;
        int i = 0;
        while (file.ReadString(temp)) //不断读取，直到尾部
        {
            //查找#位置
            int x = temp.Find("#");
            int x2 = temp.Find("#", x + 1);
            int x3 = temp.Find("#", x2 + 1);
            int x4 = temp.Find("#", x3 + 1);
            int x5 = temp.Find("#", x4 + 1);
            int x6 = temp.Find("#", x5 + 1);
            int x7 = temp.Find("#", x6 + 1);
            //写入临时变量Stemp
            Stemp.Num = temp.Left(x);
            Stemp.Grade = temp.Mid(x + 1, x2 - x - 1);
            Stemp.Class = temp.Mid(x2 + 1, x3 - x2 - 1);
            Stemp.Name = temp.Mid(x3 + 1, x4 - x3 - 1);
            Stemp.Sex = temp.Mid(x4 + 1, x5 - x4 - 1);
            Stemp.maths = _tcsdtoa(temp.Mid(x5 + 1, x6 - x5 - 1), NULL);
            Stemp.CPP = _tcsdtoa(temp.Mid(x6 + 1, x7 - x6 - 1), NULL);
            //调用ManagerSystem成员函数放入容器
            ManagerSystem.AddData(Stemp);
            //刷新contorl list控件
            m_List.InsertItem(i, Stemp.Num);
            m_List.SetItemText(i, 1, Stemp.Grade);
            m_List.SetItemText(i, 2, Stemp.Class);

```

```

        m_List.SetItemText(i, 3, Stemp.Name);
        m_List.SetItemText(i, 4, Stemp.Sex);
        CString Format_Gra;
        Format_Gra.Format("%.1f", Stemp.maths);
        m_List.SetItemText(i, 5, Format_Gra);
        Format_Gra.Format("%.1f", Stemp.CPP);
        m_List.SetItemText(i, 6, Format_Gra);

        i++;
    }
}

//更新学生人数
int nCount = m_List.GetItemCount();
STU.Format(T("%d"), nCount);
SetDlgItemText(IDC_EDIT_STU, STU);
}

```

直方、折线图MFC 实现

✧ MytestView.cpp

```

void CMytestView::OnOne()//读文件
{
    a = b = c = d = e = 0;

    //打开文件并且操作
    CFileDialog fdlg(TRUE, ".dat", "save", OFN_OVERWRITEPROMPT, "数据文件(*.txt)|*.txt|
所有文件(*.*)|*.*|", NULL);

    if (fdlg.DoModal() == IDOK)
    {
        CStdioFile file(fdlg.GetPathName(), CFile::modeRead);
        CString cs;

        int i = 0;
        while (file.ReadString(cs))
        {
            //还能读取到行
            if (i<5)
            {
                //转换为数值再判断大小

```

```

        int j=cs.GetLength();
        cs = cs.Right(10);
        cs = cs.Left(4);
        long long q = _ttoi(cs);
        if (q < 60) a++;
        else if (q >= 60 && q <= 70) b++;
        else if (q > 70 && q <= 80) c++;
        else if (q > 80 && q <= 90) d++;
        else if (q > 90) e++;
    }
    i++;
}
}
InvalidateRect(NULL);
}

void CMytestView::Ontwo()//读文件
{
    a = b = c = d = e = 0;

    //打开文件并且操作
    CFileDialog fdlg(TRUE, ".dat", "save", OFN_OVERWRITEPROMPT, "数据文件(*.txt)|*.txt|
所有文件(*.*)|*.*|", NULL);

    //如果成功读取文件
    if (fdlg.DoModal() == IDOK)
    {
        CStdioFile file(fdlg.GetPathName(), CFile::modeRead);
        CString cs;

        int i = 0;
        while (file.ReadString(cs))
        {
            //还能读取到行
            if (i < 5)
            {
                //转换为数值再判断大小
                int j = cs.GetLength();
                cs = cs.Right(5);
                cs = cs.Left(4);
                long long q = _ttoi(cs);
                if (q < 60) a++;
                else if (q >= 60 && q <= 70) b++;
            }
        }
    }
}

```

```

        else if (q > 70 && q <= 80) c++;
        else if (q > 80 && q <= 90) d++;
        else if (q > 90) e++;
    }
    i++;
}
}
InvalidateRect(NULL);
}

```

```

void CMytestView::OnDraw(CDC* pDC) //绘图
{
    // CMytestDoc* pDoc = GetDocument();
    // ASSERT_VALID(pDoc);
    // if (!pDoc)
    //     return;

```

//矩阵高度

//在OnDraw函数中添加以下代码:

```

pDC->SetMapMode(MM_ISOTROPIC);
pDC->SetViewportOrg(200, 450); //设置原点
pDC->MoveTo(100, 0); //从100, 0画到1300, 0
pDC->LineTo(1300, 0);

```

//设置画笔参数

CBrush brush;

//画笔颜色

```

brush.CreateHatchBrush(HS_BDIAGONAL, RGB(150, 100, 45));
pDC->SelectObject (brush);

```

int i = 1;

//绘制矩形, 分别为四个点坐标

```

pDC->Rectangle(200, 0, 400, 50 * a);
pDC->Rectangle(400, 0, 600, 50 * b);
pDC->Rectangle(600, 0, 800, 50 * c);
pDC->Rectangle(800, 0, 1000, 50 * d);
pDC->Rectangle(1000, 0, 1200, 50 * e);

```

//设置分段文本坐标

```

pDC->TextOut (250, 0, "<60");
pDC->TextOut (450, 0, "60-70");
pDC->TextOut (650, 0, "70-80");
pDC->TextOut (850, 0, "80-90");

```

```

pDC->TextOut(1050, 0, "90-100");

//设置文字高度（多少人）
CString y1, y2, y3, y4, y5;
y1.Format("%d", a);
y2.Format("%d", b);
y3.Format("%d", c);
y4.Format("%d", d);
y5.Format("%d", e);

//x轴从250-1050
pDC->MoveTo(250, 50 * a + 100);
pDC->LineTo(450, 50 * b + 100);
pDC->LineTo(650, 50 * c + 100);
pDC->LineTo(850, 50 * d + 100);
pDC->LineTo(1050, 50 * e + 100);

//设置文本位置
if (a != 0)
    pDC->TextOut(250, 27 * a, y1 + "人");
if (b != 0)
    pDC->TextOut(450, 27 * b, y2 + "人");
if (c != 0)
    pDC->TextOut(650, 27 * c, y3 + "人");
if (d != 0)
    pDC->TextOut(850, 27 * d, y4 + "人");
if (e != 0)
    pDC->TextOut(1050, 27 * e, y5 + "人");
}

```


四、程序运行时的效果图

➤ 登录对话框



STUSM 统一身份认证

广州大学
GUANGZHOU UNIVERSITY

统一身份认证

账号: 32116160100

密码: *****

test

登录

☐ 记住账号密码

➤ 管理系统主对话框



STUSM 学生成绩管理系统-当前用户: 计科212 冯韦铭 32116160100

排序依据: ☒ 高数 ☐ CPP

排序 全选 反选 删除 修改

学号	年级	班级	姓名	性别	高等数学	面向对象程序设计
<input type="checkbox"/> 32116160100	大一	计科212	冯韦铭	男	96.2	94.2
<input type="checkbox"/> 32116160199	大四	计科214	林佳芬	女	98.0	98.4
<input type="checkbox"/> 32116160087	大一	计科211	刘尊文	男	86.0	96.4
<input type="checkbox"/> 32116160085	大一	计科211	辛勇	男	98.8	87.6
<input type="checkbox"/> 32116160082	大一	智能211	李治	男	95.2	89.0
<input type="checkbox"/> 32116160084	大一	智能211	李如	男	78.8	79.0

学生人数: 6 打开 保存

学号: 32116160084

姓名: 李如

班级: 智能211

性别: ☒ 男 ☐ 女

年级: ☒ 大一 ☐ 大二 ☐ 大三 ☐ 大四

平时成绩: 高数 50 + 98 CPP 61 + 91

期末考试: 高数 98 CPP 91

添加

➤ 按 CPP 成绩排序

STUSM 学生成绩管理系统-当前用户: 计科212 冯韦铭 32116160100

排序依据: ☒ 高数 ☐ CPP

学生人数: 6

学号	年级	班级	姓名	性别	高等数学	面向对象程序设计
<input type="checkbox"/> 32116160199	大四	计科214	林佳芬	女	98.0	98.4
<input type="checkbox"/> 32116160087	大一	计科211	刘馨文	男	86.0	96.4
<input type="checkbox"/> 32116160100	大一	计科212	冯韦铭	男	96.2	94.2
<input type="checkbox"/> 32116160082	大一	智能211	李治	男	95.2	89.0
<input type="checkbox"/> 32116160085	大一	计科211	幸勇	男	98.8	87.6
<input type="checkbox"/> 32116160084	大一	智能211	李如	男	78.8	79.0

学号: 32116160084

姓名: 李如

班级: 智能211

性别: ☒ 男 ☐ 女

年级: ☒ 大一 ☐ 大二 ☐ 大三 ☐ 大四

平时成绩: 高数 50 + 期末考试 98

CPP 61 + 91

➤ 动图演示如下

STUSM 学生成绩管理系统-当前用户: 计科212 冯韦铭 32116160100

排序依据: ☒ 高数 ☐ CPP

学生人数: 6

学号	年级	班级	姓名	性别	高等数学	面向对象程序设计
<input type="checkbox"/> 32116160100	大一	计科212	冯韦铭	男	96.2	94.2
<input type="checkbox"/> 32116160199	大四	计科214	林佳芬	女	98.0	98.4
<input type="checkbox"/> 32116160087	大一	计科211	刘馨文	男	86.0	96.4
<input type="checkbox"/> 32116160085	大一	计科211	幸勇	男	98.8	87.6
<input type="checkbox"/> 32116160082	大一	智能211	李治	男	95.2	89.0
<input type="checkbox"/> 32116160084	大一	智能211	李如	男	78.8	79.0

学号:

姓名:

班级:

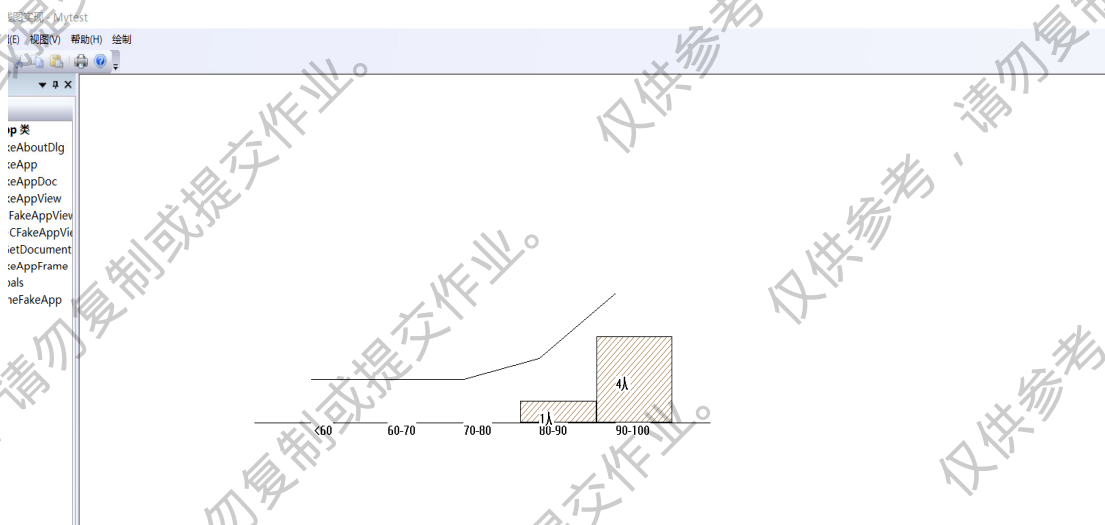
性别: ☐ 男 ☐ 女

年级: ☐ 大一 ☐ 大二 ☐ 大三 ☐ 大四

平时成绩: 高数 + 期末考试

CPP +

➤ 单文档绘图



五、实验结果分析，实验收获和体会

➤ 实验结果分析

这次的 mfc 课设是对这学期学习的 CPP 知识的综合应用，并且结合了 VC 的单文档与对话框的应用。

实验运用单文档和对话框实现了要求，并且在要求以外，我根据实际情况和自己的想法在对话框程序中添加了登录对话框、多选 反选按钮、年级 性别的单选框等功能。程序实现添加、修改、删除、排序等功能，并且运行良好。

➤ 实验收获和体会

在初写程序的时候我对 mfc 是完全一头雾水，可以说是 0 基础开发。但在完成整个课设后，我已经开始熟悉了 mfc 的一些函数 类库 控件等等。

第一个难倒我的是对控件的添加处理事件，后来通过在网络上的学习我了解了类向导的使用，控件变量的添加。并且我在开发中时常会碰到因为控件过多需要频繁查看控件 ID 的情况，我认识到了良好的控件 ID 命名的重要性，使用 IDC_EDIT_Name 这种格式的命名方式可以有效的提高写代码效率也可以使代码可读性提高。

第二个困扰了我整整一天的 bug 是在写 managerment 类的 AddData 函数时，程序运行到 } 时会发生内存越界的错误，但是我花了非常多时间排查了所有的代码后也没有找到问题所在，当时心态接近崩溃。最后发现居然是在 managerment

的构造函数中，有两句话的顺序发生了颠倒。但是我曾无数次排查到这个地方却浑然不知，我们要用计算机执行程序机械性去阅读、写代码，不能以阅读文学作品那样加以自己的脑补。

MFC 是一个非常庞大的类库，我学习到的只是冰山一角，在这次的课程设计中时间紧迫，我只能不断地上网查找资料、现学现卖，这极大地提高了我的学习能力和抗压能力。锻炼了作为一名计算机学生的基本素养。

这次的课程设计是对我编程能力的极大考验，在有限的时间内完成这样一个我来说庞大的程序，使我终身难忘没日没夜的 debug，也由衷向编写各类应用程序的程序员表示敬佩，是他们的辛勤工作使我们用上了简单易用的程序。最后感谢老师的指导与帮助。

六、 自评成绩

90 分