

Windows 程序设计

实验报告



姓 名
班 级
实 验 名 称
开 设 学 期
开 设 时 间
报 告 日 期
评 定 成 绩

东北大学软件学院

实验一 基于 Win32 应用的性格测试分析系统

一、实验内容：

使用 visual studio 2019 的 windows 桌面向导在老师的 win32 程序基础上，实现创新，构建一个属于自己的 win32 桌面应用程序。我开发了一个简单的性格测试调查，用户可以在正确输入答案后，显示自己的性格特点。并且查看所有分数对应的性格特点。

二、实验环境：

1. 实验类型：设计型。

2. 实验环境：

操作系统：windows10

调试软件名称及版本号：Microsoft Visual Studio Community 2019 版本 16.11.4

编程语言及版本号：C++/C

上机地点：班级未集中进行实验课

机器台号：华硕飞行堡垒 8

三、实验步骤及实验分析：

1. 头文件的引入及宏定义

```
#include "framework.h"
#include "WindowsProject2.h"
#include <string>
#include<iostream>
#include<vector>
#include<Windows.h>
#include <strstream>
#include <sstream>
using namespace std;
#define MAX_LOADSTRING 100
```

2. 数据结构的声明

(1)类 Question

```
class Question {
private:
    wstring question;
    wstring optionA;
    wstring optionB;
    wstring optionC;
    wstring optionD;
    wstring optionE;
public:
    Question(wstring question, wstring optionA, wstring optionB, wstring optionC, wstring optionD, wstring optionE) {
        this->question = question;
        this->optionA = optionA;
        this->optionB = optionB;
        this->optionC = optionC;
        this->optionD = optionD;
        this->optionE = optionE;
    }
}
```

(2)存储类 Question 的容器

```
vector<Question> questions;
vector<wstring> problems;
vector<wstring> choices_a;
vector<wstring> choices_b;
vector<wstring> choices_c;
vector<wstring> choices_d;
vector<wstring> choices_e;
wstring question;
wstring choice_a;
wstring choice_b;
wstring choice_c;
wstring choice_d;
wstring choice_e;
```

(3)windowsAPI 的全局变量

```
// 全局变量:
HINSTANCE hInst; // 当前实例
WCHAR szTitle[MAX_LOADSTRING]; // 标题栏文本
WCHAR szWindowClass[MAX_LOADSTRING]; // 主窗口类名

// 此代码模块中包含的函数的前向声明:
ATOM MyRegisterClass(HINSTANCE hInstance);
BOOL InitInstance(HINSTANCE, int);
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
INT_PTR CALLBACK About(HWND, UINT, WPARAM, LPARAM);
```

3. 代码逻辑

(1) 对类 Question 的封装

```
class Question {
private:
    wstring question;
    wstring optionA;
    wstring optionB;
    wstring optionC;
    wstring optionD;
    wstring optionE;
public:
    Question(wstring question, wstring optionA, wstring optionB, wstring optionC, wstring optionD, wstring optionE)
        : question(question), optionA(optionA), optionB(optionB), optionC(optionC), optionD(optionD), optionE(optionE) {}
    wstring to_string() {
        return (L"question: " + this->question + L"\n" +
            L"option A: " + this->optionA + L"\n" +
            L"option B: " + this->optionB + L"\n" +
            L"option C: " + this->optionC + L"\n" +
            L"option D: " + this->optionD + L"\n" +
            L"option E: " + this->optionE + L"\n");
    }
    wstring get_question() {
        return this->question;
    }
    wstring get_optionA() {
        return this->optionA;
    }
    wstring get_optionB() {
        return this->optionB;
    }
}
```

(2) 对类 Question 相关数据的初始化

分别调用函数 `void init_data()` 和 `void init_questions()`:

```
void init_questions() {
    for (int i = 0; i < problems.size(); i++) {
        Question question(problems[i], choices_a[i], choices_b[i], choices_c[i], choices_d[i], choices_e[i]);
        questions.push_back(question);
    }
}

void init_data() {
    problems.push_back(L"1、你更喜欢吃那种水果？");
    problems.push_back(L"2、你平时休闲经常去的地方？");
    problems.push_back(L"3、你认为容易吸引你的人是？");
    problems.push_back(L"4、如果你可以成为一种动物，你希望自己是哪种？");
    problems.push_back(L"5、天气很热，你更愿意选择什么方式解暑？");
    problems.push_back(L"6、如果必须与一个你讨厌的动物或昆虫在一起生活，你能容忍哪一个？");
    problems.push_back(L"7、你喜欢看哪类电影、电视剧？");
    problems.push_back(L"8、以下哪个是你身边必带的物品？");
    problems.push_back(L"9、你出行时喜欢坐什么交通工具？");
    problems.push_back(L"10、以下颜色你更喜欢哪种？");
    problems.push_back(L"11、下列运动中挑选一个你最喜欢的（不一定擅长）？");
    problems.push_back(L"12、如果你拥有一座别墅，你认为它应当建立在哪里？");
    problems.push_back(L"13、你更喜欢以下哪种天气现象？");
    problems.push_back(L"14、你希望自己的窗口在一座30层大楼的第几层？");
    problems.push_back(L"15、你认为自己更喜欢在以下哪一个城市中生活？");

    choices_a.push_back(L"草莓");
    choices_b.push_back(L"苹果");
    choices_c.push_back(L"西瓜");
    choices_d.push_back(L"菠萝");
    choices_e.push_back(L"橘子");
```

(3)对工具类函数的封装

```
wstring IntToString(unsigned int i)
{
    wstringstream ss;
    ss << i;
    return ss.str();
}

string TCHAR_to_string(TCHAR* STR)
{
    int iLen = WideCharToMultiByte(CP_ACP, 0, STR, -1, NULL, 0, NULL, NULL);
    char* chRtn = new char[iLen * sizeof(char)];
    WideCharToMultiByte(CP_ACP, 0, STR, -1, chRtn, iLen, NULL, NULL);
    std::string str(chRtn);
    delete chRtn;
    return str;
}
```

(4)window 库函数的声明

```
// 此代码模块中包含的函数的前向声明:
ATOM           MyRegisterClass(HINSTANCE hInstance);
BOOL           InitInstance(HINSTANCE, int);
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
INT_PTR CALLBACK About(HWND, UINT, WPARAM, LPARAM);
```

①winmain 函数的声明

```
int APIENTRY wWinMain(_In_ HINSTANCE hInstance,
                      _In_opt_ HINSTANCE hPrevInstance,
                      _In_ LPWSTR    lpCmdLine,
                      _In_ int       nCmdShow)
{
    UNREFERENCED_PARAMETER(hPrevInstance);
    UNREFERENCED_PARAMETER(lpCmdLine);

    // TODO: 在此处放置代码。
    init_data();
    init_questions();
    // 初始化全局字符串
    LoadStringW(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadStringW(hInstance, IDC_WINDOWSPROJECT2, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);
    // 执行应用程序初始化:
    if (!InitInstance (hInstance, nCmdShow))
    {
        return FALSE;
    }
    HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC_WINDOWSPROJECT2));
    MSG msg;
    // 主消息循环:
    while (GetMessage(&msg, nullptr, 0, 0))
    {
        if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }

    return (int) msg.wParam;
}
```

②注册窗口类函数的声明

```
ATOM MyRegisterClass(HINSTANCE hInstance)
{
    WNDCLASSEXW wcex;

    wcex.cbSize = sizeof(WNDCLASSEX);

    wcex.style          = CS_HREDRAW | CS_VREDRAW;
    wcex.lpfnWndProc   = WndProc;
    wcex.cbClsExtra     = 0;
    wcex.cbWndExtra     = 0;
    wcex.hInstance      = hInstance;
    wcex.hIcon          = LoadIcon(hInstance, MAKEINTRESOURCE(IDI_WINDOWSPROJECT2));
    wcex.hCursor         = LoadCursor(nullptr, IDC_ARROW);
    wcex.hbrBackground  = (HBRUSH)(COLOR_WINDOW+1);
    wcex.lpszMenuName   = MAKEINTRESOURCEW(IDC_WINDOWSPROJECT2);
    wcex.lpszClassName   = szWindowClass;
    wcex.hIconSm         = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI_SMALL));

    return RegisterClassExW(&wcex);
}
```

③窗口句柄的初始化

```
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
    hInst = hInstance; // 将实例句柄存储在全局变量中

    HWND hwnd = CreateWindowW(szWindowClass, L"Northeastern University SoftwareCollege", WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, nullptr, nullptr, hInstance, nullptr);

    /* HWND hwnd;
    hwnd = CreateWindow(L"SoftwareCollege", L"软件学院", WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, CW_USEDEFAULT, 600, 400, NULL, NULL, hInstance, NULL); */

    if (!hwnd)
    {
        return FALSE;
    }

    ShowWindow(hwnd, SW_SHOWNORMAL);
    UpdateWindow(hwnd);

    return TRUE;
}
```

④回调函数的声明

```
HRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    const WCHAR* s;
    wstring ss = L"是否真的结束？";
    HWND edit{};
    HWND button;
    int wmid;
    /* LPTSTR ans="fff"; */
    TCHAR ans[15] = _T("aa");
    string res;
    int score = 0;
    switch (message)

    // “关于”框的消息处理程序。
    INT_PTR CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)
    {
        UNREFERENCED_PARAMETER(lParam);
        switch (message)
        {
            case WM_INITDIALOG:
                return (INT_PTR)TRUE;

            case WM_COMMAND:
                if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)
                {
                    EndDialog(hDlg, LOWORD(wParam));
                    return (INT_PTR)TRUE;
                }
                break;
        }
        return (INT_PTR)FALSE;
    }
}
```

(5)为系统回调函数添加消息处理

①窗口初始化时

```
case WM_CREATE:  
    edit=CreateWindow(TEXT("edit"), TEXT("请输入测试答案: "), WS_CHILD | WS_VISIBLE | ES_MULTILINE,  
        450, 505, 350, 500,  
        hWnd, (HMENU)0, ((LPCREATESTRUCT)lParam)->hInstance, NULL);  
    SendMessage(edit, WM_SETFONT, (WPARAM)GetStockObject(25), 0);  
  
    button= CreateWindow(TEXT("button"), //必须为: button  
        TEXT("提交答案"), //按钮上显示的字符  
        WS_CHILD | WS_VISIBLE,  
        800, 505, 150, 20, //x, y, 宽, 高  
        hWnd, (HMENU)1, //绑定按钮ID  
        ((LPCREATESTRUCT)lParam)->hInstance, NULL);  
  
    break;
```

新建一个文本框和按钮，文本编辑框存储性格测试题目的答案，按钮为确定提交按钮，用户在点击确定提交后，即可看见自己的分数。

②鼠标右键按下

```
case WM_RBUTTONDOWN:  
    InvalidateRect(hWnd, NULL, true);  
  
    break;
```

刷新页面，重画窗口。

③鼠标右键松开

```
case WM_RBUTTONUP:  
    HDC hdc0;  
    hdc0 = GetDC(hWnd);  
    TextOut(hdc0, 0, 115, tips1_1.c_str(), wcslen(tips1_1.c_str()));  
    TextOut(hdc0, 0, 145, tips1_2.c_str(), wcslen(tips1_2.c_str()));  
    TextOut(hdc0, 0, 175, tips1_3.c_str(), wcslen(tips1_3.c_str()));  
  
    TextOut(hdc0, 0, 205, tips2_1.c_str(), wcslen(tips2_1.c_str()));  
    TextOut(hdc0, 0, 235, tips2_2.c_str(), wcslen(tips2_2.c_str()));  
    TextOut(hdc0, 0, 265, tips2_3.c_str(), wcslen(tips2_3.c_str()));  
  
    TextOut(hdc0, 0, 295, tips3_1.c_str(), wcslen(tips3_1.c_str()));  
    TextOut(hdc0, 0, 325, tips3_2.c_str(), wcslen(tips3_2.c_str()));  
    TextOut(hdc0, 0, 355, tips3_3.c_str(), wcslen(tips3_3.c_str()));  
  
    TextOut(hdc0, 0, 385, tips4_1.c_str(), wcslen(tips4_1.c_str()));  
    TextOut(hdc0, 0, 415, tips4_2.c_str(), wcslen(tips4_2.c_str()));  
    TextOut(hdc0, 0, 445, tips4_3.c_str(), wcslen(tips4_3.c_str()));  
  
    TextOut(hdc0, 0, 475, tips5_1.c_str(), wcslen(tips5_1.c_str()));  
    TextOut(hdc0, 0, 505, tips5_2.c_str(), wcslen(tips5_2.c_str()));  
    TextOut(hdc0, 0, 535, tips5_3.c_str(), wcslen(tips5_3.c_str()));  
  
    TextOut(hdc0, 0, 565, tips6_1.c_str(), wcslen(tips6_1.c_str()));  
    TextOut(hdc0, 0, 595, tips6_2.c_str(), wcslen(tips6_2.c_str()));  
    TextOut(hdc0, 0, 625, tips6_3.c_str(), wcslen(tips6_3.c_str()));  
  
    ReleaseDC(hWnd, hdc0);  
    break;
```

显示性格测试的所有情况，按照线性布局依次呈现在窗口上。此消息 WM_RBUTTONUP 是和 WM_RBUTTONDOWN 结合在一起使用的。当用户鼠标右键点击屏幕后会刷新窗口的文字，当用户松开鼠标右键后，则会执行显示性格测试所有情况的程序。

④关闭窗口

```
case WM_CLOSE:  
    s = L"是否真的结束？";  
    if (IDYES == MessageBox(hWnd, ss.c_str(), L"退出警告", MB_YESNO))  
    {  
        DestroyWindow(hWnd);  
    }  
  
    break;  
  
case WM_DESTROY:  
    PostQuitMessage(0);  
    break;
```

添加询问，用户点击确定后关闭应用程序

⑤提前加载测试题目的说明文字

```
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: 在此处添加使用 hdc 的任何绘图代码...
    wstring content1= L"本性格测评是由中国现代心理研究所著名的美国兰德公司拟制的一套经典心理测试题为蓝本，根据中国人心理特点加以适当改造后形成的心理测试题。";
    wstring content2 = L"这套性格测试题目前已被一些著名大公司，如联想、长虹、海尔等公司作为对员工性格测评的重要辅助试卷，据说效果很好。";
    wstring content3 = L"(Tips:现在已经有人建议将来作为对公务员的必选辅助心理测试推广使用。点击屏幕任意地方可以打开测试题目！)";
    wstring title = L"东北大学在线性格测试系统";
    TextOut(hdc, 450, 5, title.c_str(), wcslen(title.c_str()));
    TextOut(hdc, 10, 35, content1.c_str(), wcslen(content1.c_str()));
    TextOut(hdc, 10, 65, content2.c_str(), wcslen(content2.c_str()));
    TextOut(hdc, 250, 95, content3.c_str(), wcslen(content3.c_str()));
    EndPaint(hWnd, &ps);
}
break;
```

⑥对于其他重要命令的响应

消息处理为：

```
case WM_COMMAND:
{
    wmid = LOWORD(wParam);
    // 分析菜单选择:
    switch (wmId)
```

首先获取传进来的消息 id。

然后根据消息 id 添加消息处理：

```
switch (wmId)
{
case 1:
    if (IDYES == MessageBox(hWnd, L"你确定要提交了吗?", L"通知", MB_YESNO | MB_ICONINFORMATION)) {
        GetWindowText(GetDlgItem(hWnd, 0), ans, 15);
        /*SendMessage(edit, WM_GETTEXT, 15, (LPARAM)ans);*/
        res = TCHAR_to_string(ans);
        for (int i = 0; i < res.size(); i++) {
            if (res.at(i) == 'a' || res.at(i) == 'A') {
                score = score + 2;
            }
            if (res.at(i) == 'b' || res.at(i) == 'B') {
                score = score + 3;
            }
            if (res.at(i) == 'c' || res.at(i) == 'C') {
                score = score + 5;
            }
            if (res.at(i) == 'd' || res.at(i) == 'D') {
                score = score + 10;
            }
            if (res.at(i) == 'e' || res.at(i) == 'E') {
                score = score + 15;
            }
        }
        wstring notify = L"你的最终得分为" + IntToString(score);
        MessageBox(hWnd, notify.c_str(), L"通知", NULL);
    }
}
break;
```

当确定按钮被点击后，系统提示是否确定提交，若取消则系统没有响应的相关程序。否则，从文本编辑框里面获取至多 15 个字符。（因为一共只有 15 道题目）然后根据获取的答案字符串来计算总得分。最后将总得分显示出来，呈现在窗体中以提示用户。

其他消息处理：

```
case IDM_ABOUT:
    DialogBox(hInst, MAKEINTRESOURCE(IDD_ABOUTBOX), hWnd, About);
    break;
case IDM_EXIT:
    DestroyWindow(hWnd);
    break;
default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}
```

四、实验结果：

主函数：

```
int APIENTRY wWinMain(_In_ HINSTANCE hInstance,
                      _In_opt_ HINSTANCE hPrevInstance,
                      _In_ LPWSTR    lpCmdLine,
                      _In_ int       nCmdShow)
{
    UNREFERENCED_PARAMETER(hPrevInstance);
    UNREFERENCED_PARAMETER(lpCmdLine);

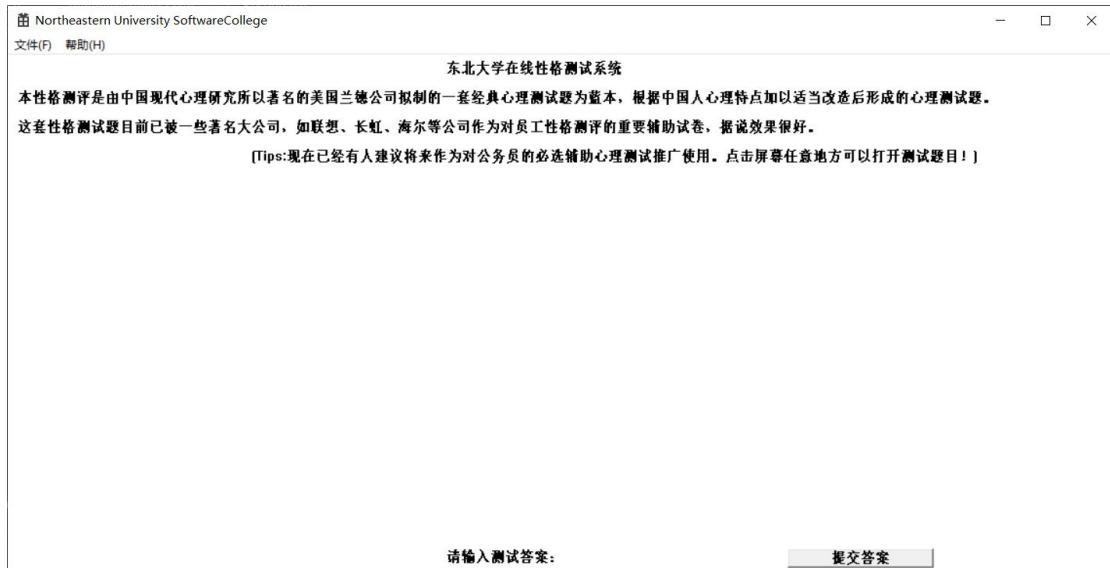
    // TODO: 在此处放置代码。
    init_data();
    init_questions();
    // 初始化全局字符串
    LoadStringW(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadStringW(hInstance, IDC_WINDOWSPROJECT2, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);
    // 执行应用程序初始化:
    if (!InitInstance (hInstance, nCmdShow))
    {
        return FALSE;
    }
    HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC_WINDOWSPROJECT2));
    MSG msg;
    // 主消息循环:
    while (GetMessage(&msg, nullptr, 0, 0))
    {
        if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }

    return (int) msg.wParam;
}
```

运行实验结果：

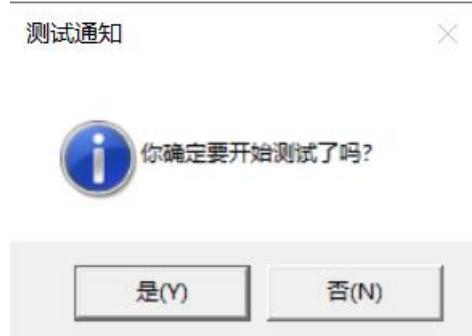
(1)进入程序主页面

此时进入到本程序的主页面。本程序是一个基于 win32 的性格测试分析系统。首先是调用消息处理 WM_PAINT 画出如下的文字。



(2)鼠标左键点击屏幕上任意位置

此时屏幕上为“东北大学在线测试系统”的使用说明和题目背景。调用消息处理 WM_LBUTTONDOWN，系统会出现弹窗，提示用户是否想要开始测试。若选择是，系统会显示所有性格测试题目；选择否，则不会显示题目：



我们点击“是”，系统会显示所有性格测试题目：

东北大学在线性格测试系统

本性格测评是由中国现代心理研究所以著名的美国兰德公司拟制的一套经典心理测试题为蓝本，根据中国人心理特点加以适当改造后形成的心理测试题。

这套性格测试题目前已被一些著名大公司，如联想、长虹、海尔等公司作为对员工性格测评的重要辅助试卷，据说效果很好。

[Tips:现在已经有人建议将来作为对公务员的必选辅助心理测试推广使用。点击屏幕任意地方可以打开测试题目！]

question: 1、你更喜欢吃哪种水果？ option A: 草莓 | option B: 苹果 | option C: 西瓜 | option D: 菠萝 | option E: 瓪子 |

question: 2、你平时休闲经常去的地方 option A: 郊外 | option B: 电影院 | option C: 公园 | option D: 商场 | option E: 酒吧 |

question: 3、你认为容易吸引你的人是？ option A: 有才气的人 | option B: 依赖你的人 | option C: 优雅的人 | option D: 善良的人 | option E: 性情豪放的人 |

question: 4、如果你可以成为一种动物，你希望自己是哪种？ option A: 猫 | option B: 马 | option C: 大象 | option D: 狐子 | option E: 狗 |

question: 5、天气很热，你更愿意选择什么方式解暑？ option A: 游泳 | option B: 喝冷饮 | option C: 开空调 | option D: 睡觉 | option E: 洗澡 |

question: 6、如果必须与一个你讨厌的动物或昆虫在一起生活，你能容忍哪一个？ option A: 蛇 | option B: 猪 | option C: 老鼠 | option D: 苍蝇 | option E: 蜘蛛 |

question: 7、你喜欢看哪类电影、电视剧？ option A: 悬疑推理类 | option B: 童话神话类 | option C: 自然科学类 | option D: 伦理道德类 | option E: 战争枪战类 |

question: 8、以下哪个是你身边必备的物品？ option A: 打火机 | option B: 口红 | option C: 记事本 | option D: 纸巾 | option E: 手机 |

question: 9、你出行时喜欢坐什么交通工具？ option A: 火车 | option B: 自行车 | option C: 汽车 | option D: 飞机 | option E: 步行 |

question: 10、以下颜色你更喜欢哪种？ option A: 黑或白 | option B: 紫 | option C: 蓝 | option D: 黄 | option E: 红 |

question: 11、下列运动中挑选一个你最喜欢的（不一定擅长）？ option A: 雪雕 | option B: 自行车 | option C: 乒乓球 | option D: 拳击 | option E: 足球或篮球 |

question: 12、如果你拥有一座别墅，你认为它应当建立在哪里？ option A: 湖边 | option B: 草原 | option C: 海边 | option D: 森林 | option E: 城中区 |

question: 13、你更喜欢以下哪种天气现象？ option A: 雪 | option B: 风 | option C: 雨 | option D: 雾 | option E: 雷电 |

question: 14、你希望自己的窗口在一座30层大楼的第几层？ option A: 七层 | option B: 一层 | option C: 二十三层 | option D: 十八层 | option E: 三十层 |

question: 15、你认为自己更喜欢在以下哪一个城市中生活？ option A: 南京 | option B: 沈阳 | option C: 北京 | option D: 上海 | option E: 成都 |

请输入测试答案:

提交答案

在下方的文本编辑框输入答案：

请输入测试答案：abcdeabcdeabcde|

点击提交按钮，系统提示我们是否确定提交，我们点击是：



系统弹窗会提示分数：



调用消息处理 WM_RBUTTONDOWN，点击鼠标右键，实现题目的清除：



调用消息处理 WM_RBUTTONDOWN，释放鼠标右键，出现测试结果：

180分以上：意志力强，头脑冷静，有较强的领导欲，事业心强，不达目的不罢休。
外表和善，内心自傲，对有利于自己的人际关系比较看重，有时显得性格急躁，咄咄逼人，得理不饶人，不利于自己时顽强抗争，不轻易认输。

思维理性，对爱情和婚姻的看法很现实，对金钱的欲望一般。思维理性，对爱情和婚姻的看法很现实，对金钱的欲望一般。

140分至179分：聪明，性格活泼，人缘好，善于交朋友，心机较深。

事业心强，渴望成功。思维较理性，崇尚爱情，但当爱情与婚姻发生冲突时会选择有利于自己的婚姻。

金钱欲望强烈。

100分至139分：爱幻想，思维较感性，以是否与自己投缘为标准来选择朋友。

性格显得较孤傲，有时较急躁，有时优柔寡断。事业心较强，喜欢有创造性的工作，不喜欢按常规办事。

性格倔强，言语犀利，不善于妥协。崇尚浪漫的爱情，但想法往往不切合实际。金钱欲望一般。

70分至99分：好奇心强，喜欢冒险，人缘较好。事业心一般，对待工作，随遇而安，善于妥协。

善于发现有趣的事情，但耐心较差，敢于冒险，但有时较胆小。

渴望浪漫的爱情，但对婚姻的要求比较现实。不善理财。

40分至69分：性情温良，重友谊，性格塌实稳重，但有时也比较狡黠。

事业心一般，对本职工作能认真对待，但对自己专业以外事物没有太大兴趣。:edabced

提交答案

喜欢有规律的工作和生活，不喜欢冒险，家庭观念强，比较善于理财。

40分以下：散漫，爱玩，富于幻想。聪明机灵，待人热情，爱交朋友，但对朋友没有严格的选择标准。

事业心较差，更善于享受生活，意志力和耐心都较差，我行我素。

有较好的异性缘，但对爱情不够坚持认真，容易妥协。没有财产观念。

至此，东北大学在线性格测试系统的功能就演示完毕了。我的测试结果为 105 分，对应于以下的性格特点：

100 分至 139 分：爱幻想，思维较感性，以是否与自己投缘为标准来选择朋友。性格显得较孤傲，有时较急躁，有时优柔寡断。事业心较强，喜欢有创造性的工作，不喜欢按常规办事。性格倔强，言语犀利，不善于妥协。崇尚浪漫的爱情，但想法往往不切合实际。金钱欲望一般。

五、实验总结：

在本次操作过程中我们学到了 windows32 API 编程的基本程序框架与编程方法，并且掌握了几个主要 API 函数的功能，以及如何利用 visual studio2019 的编译环境编写 C 语言下的 windows 程序的步骤和利用其资源管理和编译器建立 windows 资源的方法和 API 编程对 windows 消息的处理有利于 MFC 下的 windows 编程的理解。虽然刚开始的时候并没有真真

正正的做出来，后来通过老师课堂上的又一次讲解，我们还是顺利的做了出来。通过此次实验我明白了，学习不是一蹴而就的事情。通过这次 win32 编程，我领悟到很多思想方法。总之，受益匪浅！

六、附录

实验参考资料：

- 1.<https://blog.csdn.net/misterliwei/article/details/2027882?spm=1001.2014.3001.5506>
- 2.<https://blog.csdn.net/bcbobo21cn/article/details/106340762?spm=1001.2014.3001.5506>

实验源代码：

```
// WindowsProject2.cpp : 定义应用程序的入口点。
#include "framework.h"
#include "WindowsProject2.h"
#include <string>
#include<iostream>
#include<vector>
#include<Windows.h>
#include <strstream>
#include <sstream>
using namespace std;
#define MAX_LOADSTRING 100
class Question {
private:
    wstring question;
    wstring optionA;
    wstring optionB;
    wstring optionC;
    wstring optionD;
    wstring optionE;
public:
    Question(wstring question, wstring optionA, wstring optionB, wstring optionC, wstring
optionD,wstring optionE) {
        this->question = question;
        this->optionA = optionA;
        this->optionB = optionB;
        this->optionC = optionC;
        this->optionD = optionD;
        this->optionE = optionE;
    }
    wstring to_string() {
        return (L"question: " + this->question + L" \n " +
L"option A: " + this->optionA + L" \n " +
L"option B: " + this->optionB + L" \n " +
```

```

        L"option C: " + this->optionC + L" \n " +
        L"option D: " + this->optionD + L" \n " +
        L"option E: " + this->optionE + L" \n "
    );
}

wstring get_question() {
    return this->question;
}

wstring get_optionA() {
    return this->optionA;
}

wstring get_optionB() {
    return this->optionB;
}

wstring get_optionC() {
    return this->optionC;
}

wstring get_optionD() {
    return this->optionD;
}

wstring get_optionE() {
    return this->optionE;
}

};

vector<Question> questions;
vector<wstring> problems;
vector<wstring> choices_a;
vector<wstring> choices_b;
vector<wstring> choices_c;
vector<wstring> choices_d;
vector<wstring> choices_e;
wstring question;
wstring choice_a;
wstring choice_b;
wstring choice_c;
wstring choice_d;
wstring choice_e;
wstring tips1_1 = L"180 分以上：意志力强，头脑冷静，有较强的领导欲，事业心强，不达目的不罢休。";
wstring tips1_2 = L"外表和善，内心自傲，对有利于自己的人际关系比较看重，有时显得性格急躁，咄咄逼人，得理不饶人，不利于自己时顽强抗争，不轻易认输。";
wstring tips1_3 = L"思维理性，对爱情和婚姻的看法很现实，对金钱的欲望一般。思维理性，对爱情和婚姻的看法很现实，对金钱的欲望一般。";
wstring tips2_1 = L"140 分至 179 分：聪明，性格活泼，人缘好，善于交朋友，心机较深。";

```

```
wstring tips2_2 = L"事业心强，渴望成功。思维较理性，崇尚爱情，但当爱情与婚姻发生冲突时会选择有利于自己的婚姻。";  
wstring tips2_3 = L"金钱欲望强烈。";  
wstring tips3_1 = L"100 分至 139 分：爱幻想，思维较感性，以是否与自己投缘为标准来选择朋友。";  
wstring tips3_2 = L"性格显得较孤傲，有时较急躁，有时优柔寡断。事业心较强，喜欢有创造性的工作，不喜欢按常规办事。";  
wstring tips3_3 = L"性格倔强，言语犀利，不善于妥协。崇尚浪漫的爱情，但想法往往不切合实际。金钱欲望一般。";  
wstring tips4_1 = L"70 分至 99 分：好奇心强，喜欢冒险，人缘较好。事业心一般，对待工作，随遇而安，善于妥协。";  
wstring tips4_2 = L"善于发现有趣的事情，但耐心较差，敢于冒险，但有时较胆小。";  
wstring tips4_3 = L"渴望浪漫的爱情，但对婚姻的要求比较现实。不善理财。";  
wstring tips5_1 = L"40 分至 69 分：性情温良，重友谊，性格塌实稳重，但有时也比较狡黠。";  
wstring tips5_2 = L"事业心一般，对本职工作能认真对待，但对自己专业以外事物没有太大兴趣。";  
wstring tips5_3 = L"喜欢有规律的工作和生活，不喜欢冒险，家庭观念强，比较善于理财。";  
wstring tips6_1 = L"40 分以下：散漫，爱玩，富于幻想。聪明机灵，待人热情，爱交朋友，但对朋友没有严格的选择标准。";  
wstring tips6_2 = L"事业心较差，更善于享受生活，意志力和耐心都较差，我行我素。";  
wstring tips6_3 = L"有较好的异性缘，但对爱情不够坚持认真，容易妥协。没有财产观念。";  
void init_data() {  
    problems.push_back(L"1、你更喜欢吃那种水果？");  
    problems.push_back(L"2、你平时休闲经常去的地方");  
    problems.push_back(L"3、你认为容易吸引你的人是？");  
    problems.push_back(L"4、如果你可以成为一种动物，你希望自己是哪种？");  
    problems.push_back(L"5、天气很热，你更愿意选择什么方式解暑？");  
    problems.push_back(L"6、如果必须与一个你讨厌的动物或昆虫在一起生活，你能容忍哪一个？");  
    problems.push_back(L"7、你喜欢看哪类电影、电视剧？");  
    problems.push_back(L"8、以下哪个是你身边必带的物品？");  
    problems.push_back(L"9、你出行时喜欢坐什么交通工具？");  
    problems.push_back(L"10、以下颜色你更喜欢哪种？");  
    problems.push_back(L"11、下列运动中挑选一个你最喜欢的（不一定擅长）？");  
    problems.push_back(L"12、如果你拥有一座别墅，你认为它应当建立在哪里？");  
    problems.push_back(L"13、你更喜欢以下哪种天气现象？");  
    problems.push_back(L"14、你希望自己的窗口在一座 30 层大楼的第几层？");  
    problems.push_back(L"15、你认为自己更喜欢在以下哪一个城市中生活？");  
    choices_a.push_back(L"草莓");  
    choices_b.push_back(L"苹果");  
    choices_c.push_back(L"西瓜");  
    choices_d.push_back(L"菠萝");  
    choices_e.push_back(L"橘子");  
    choices_a.push_back(L"郊外");  
    choices_b.push_back(L"电影院");  
    choices_c.push_back(L"公园");  
    choices_d.push_back(L"商场");
```

```
choices_e.push_back(L"酒吧");
choices_a.push_back(L"有才气的人");
choices_b.push_back(L"依赖你的人");
choices_c.push_back(L"优雅的人");
choices_d.push_back(L"善良的人");
choices_e.push_back(L"性情豪放的人");
choices_a.push_back(L"猫");
choices_b.push_back(L"马");
choices_c.push_back(L"大象");
choices_d.push_back(L"猴子");
choices_e.push_back(L"狗");
choices_a.push_back(L"游泳");
choices_b.push_back(L"喝冷饮");
choices_c.push_back(L"开空调");
choices_d.push_back(L"睡觉");
choices_e.push_back(L"洗澡");
choices_a.push_back(L"蛇");
choices_b.push_back(L"猪");
choices_c.push_back(L"老鼠");
choices_d.push_back(L"苍蝇");
choices_e.push_back(L"蜘蛛");
choices_a.push_back(L"悬疑推理类");
choices_b.push_back(L"童话神话类");
choices_c.push_back(L"自然科学类");
choices_d.push_back(L"伦理道德类");
choices_e.push_back(L"战争枪战类");
choices_a.push_back(L"打火机");
choices_b.push_back(L"口红");
choices_c.push_back(L"记事本");
choices_d.push_back(L"纸巾");
choices_e.push_back(L"手机");
choices_a.push_back(L"火车");
choices_b.push_back(L"自行车");
choices_c.push_back(L"汽车");
choices_d.push_back(L"飞机");
choices_e.push_back(L"步行");
choices_a.push_back(L"黑或白");
choices_b.push_back(L"紫");
choices_c.push_back(L"蓝");
choices_d.push_back(L"黄");
choices_e.push_back(L"红");
choices_a.push_back(L"瑜伽");
choices_b.push_back(L"自行车");
choices_c.push_back(L"乒乓球");
```

```
    choices_d.push_back(L"拳击");
    choices_e.push_back(L"足球或篮球");
    choices_a.push_back(L"湖边");
    choices_b.push_back(L"草原");
    choices_c.push_back(L"海边");
    choices_d.push_back(L"森林");
    choices_e.push_back(L"城中区");
    choices_a.push_back(L"雪");
    choices_b.push_back(L"风");
    choices_c.push_back(L"雨");
    choices_d.push_back(L"雾");
    choices_e.push_back(L"雷电");
    choices_a.push_back(L"七层");
    choices_b.push_back(L"一层");
    choices_c.push_back(L"二十三层");
    choices_d.push_back(L"十八层");
    choices_e.push_back(L"三十层");
    choices_a.push_back(L"南京");
    choices_b.push_back(L"沈阳");
    choices_c.push_back(L"北京");
    choices_d.push_back(L"上海");
    choices_e.push_back(L"成都");

}

void init_questions() {
    for (int i = 0; i < problems.size(); i++) {
        Question question(problems[i], choices_a[i], choices_b[i], choices_c[i], choices_d[i],
                           choices_e[i]);
        questions.push_back(question);
    }
}

wstring IntToWstring(unsigned int i)
{
    wstringstream ss;
    ss << i;
    return ss.str();
}

string TCHAR_to_string(TCHAR* STR)
{
    int iLen = WideCharToMultiByte(CP_ACP, 0, STR, -1, NULL, 0, NULL, NULL);
    char* chRtn = new char[iLen * sizeof(char)];
    WideCharToMultiByte(CP_ACP, 0, STR, -1, chRtn, iLen, NULL, NULL);
    std::string str(chRtn);
    delete chRtn;
}
```

```
    return str;
}

// 全局变量:
HINSTANCE hInst;                                // 当前实例
WCHAR szTitle[MAX_LOADSTRING];                  // 标题栏文本
WCHAR szWindowClass[MAX_LOADSTRING];             // 主窗口类名

// 此代码模块中包含的函数的前向声明:
ATOM           MyRegisterClass(HINSTANCE hInstance);
BOOL          InitInstance(HINSTANCE, int);
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
INT_PTR CALLBACK About(HWND, UINT, WPARAM, LPARAM);

int APIENTRY wWinMain(_In_ HINSTANCE hInstance,
                      _In_opt_ HINSTANCE hPrevInstance,
                      _In_ LPWSTR    lpCmdLine,
                      _In_ int       nCmdShow)

{
    UNREFERENCED_PARAMETER(hPrevInstance);
    UNREFERENCED_PARAMETER(lpCmdLine);

    // TODO: 在此处放置代码。

    init_data();
    init_questions();
    // 初始化全局字符串
    LoadStringW(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadStringW(hInstance, IDC_WINDOWSPROJECT2, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);
    // 执行应用程序初始化:
    if (!InitInstance (hInstance, nCmdShow))
    {
        return FALSE;
    }
    HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC_WINDOWSPROJECT2));
    MSG msg;

    // 主消息循环:
    while (GetMessage(&msg, nullptr, 0, 0))
    {
        if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }
    return (int) msg.wParam;
}
//
```

```
// 函数: MyRegisterClass()
//
// 目标: 注册窗口类。
//
ATOM MyRegisterClass(HINSTANCE hInstance)
{
    WNDCLASSEXW wcex;
    wcex.cbSize = sizeof(WNDCLASSEX);
    wcex.style          = CS_HREDRAW | CS_VREDRAW;
    wcex.lpfnWndProc   = WndProc;
    wcex.cbClsExtra     = 0;
    wcex.cbWndExtra     = 0;
    wcex.hInstance      = hInstance;
    wcex.hIcon          = LoadIcon(hInstance, MAKEINTRESOURCE(IDI_WINDOWSPROJECT2));
    wcex.hCursor         = LoadCursor(nullptr, IDC_ARROW);
    wcex.hbrBackground  = (HBRUSH)(COLOR_WINDOW+1);
    wcex.lpszMenuName   = MAKEINTRESOURCEW(IDC_WINDOWSPROJECT2);
    wcex.lpszClassName  = szWindowClass;
    wcex.hIconSm         = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI_SMALL));
    return RegisterClassExW(&wcex);
}

//
// 函数: InitInstance(HINSTANCE, int)
//
// 目标: 保存实例句柄并创建主窗口
//
// 注释:
//
// 在此函数中, 我们在全局变量中保存实例句柄并
// 创建和显示主程序窗口。
//
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
    hInst = hInstance; // 将实例句柄存储在全局变量中
    HWND hwnd = CreateWindowW(szWindowClass, L"Northeastern University SoftwareCollege",
    WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, nullptr, nullptr, hInstance, nullptr);
    /* HWND hwnd;
    hwnd = CreateWindow(L"SoftwareCollege", L"软件学院", WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT, CW_USEDEFAULT, 600, 400, NULL, NULL, hInstance, NULL);*/
    if (!hwnd)
    {
        return FALSE;
    }
}
```

```
    ShowWindow(hwnd, SW_SHOWNORMAL);
    UpdateWindow(hwnd);
    return TRUE;
}

// 函数: WndProc(HWND, UINT, WPARAM, LPARAM)
//
// 目标: 处理主窗口的消息。
//
// WM_COMMAND - 处理应用程序菜单
// WM_PAINT - 绘制主窗口
// WM_DESTROY - 发送退出消息并返回
//


LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    const WCHAR* s;
    wstring ss = L"是否真的结束? ";
    HWND edit{};
    HWND button;
    int wmid;

    /* LPTSTR ans="ffff"; */
    TCHAR ans[16] = _T("aa");
    string res;
    int score = 0;
    switch (message)
    {
        case WM_CREATE:
            edit=CreateWindow(TEXT("edit"), TEXT("(请在此处输入测试答案)"), WS_CHILD | WS_VISIBLE | ES_MULTILINE,
                450, 505, 350, 500,
                hWnd, (HMENU)0, ((LPCREATESTRUCT)lParam)->hInstance, NULL);
            SendMessage(edit, WM_SETFONT, (WPARAM)GetStockObject(25), 0);
            button= CreateWindow(TEXT("button"), //必须为: button
                TEXT("提交答案"), //按钮上显示的字符
                WS_CHILD | WS_VISIBLE,
                800, 505, 150, 20, //x,y,宽,高
                hWnd, (HMENU)1, //绑定按钮 ID
                ((LPCREATESTRUCT)lParam)->hInstance, NULL);
            break;
        case WM_CHAR:
            WCHAR wszChar[20];
            swprintf(wszChar,20, L"char is %c", wParam);
    }
}
```

```
    MessageBox(hWnd, wszChar, L"东北大学", MB_YESNO | MB_ICONINFORMATION);
    break;

case WM_RBUTTONDOWN:
    InvalidateRect(hWnd, NULL, true);
    break;

case WM_RBUTTONUP:
    HDC hdc0;
    hdc0 = GetDC(hWnd);
    TextOut(hdc0, 0, 115, tips1_1.c_str(), wcslen(tips1_1.c_str()));
    TextOut(hdc0, 0, 145, tips1_2.c_str(), wcslen(tips1_2.c_str()));
    TextOut(hdc0, 0, 175, tips1_3.c_str(), wcslen(tips1_3.c_str()));
    TextOut(hdc0, 0, 205, tips2_1.c_str(), wcslen(tips2_1.c_str()));
    TextOut(hdc0, 0, 235, tips2_2.c_str(), wcslen(tips2_2.c_str()));
    TextOut(hdc0, 0, 265, tips2_3.c_str(), wcslen(tips2_3.c_str()));
    TextOut(hdc0, 0, 295, tips3_1.c_str(), wcslen(tips3_1.c_str()));
    TextOut(hdc0, 0, 325, tips3_2.c_str(), wcslen(tips3_2.c_str()));
    TextOut(hdc0, 0, 355, tips3_3.c_str(), wcslen(tips3_3.c_str()));
    TextOut(hdc0, 0, 385, tips4_1.c_str(), wcslen(tips4_1.c_str()));
    TextOut(hdc0, 0, 415, tips4_2.c_str(), wcslen(tips4_2.c_str()));
    TextOut(hdc0, 0, 445, tips4_3.c_str(), wcslen(tips4_3.c_str()));
    TextOut(hdc0, 0, 475, tips5_1.c_str(), wcslen(tips5_1.c_str()));
    TextOut(hdc0, 0, 505, tips5_2.c_str(), wcslen(tips5_2.c_str()));
    TextOut(hdc0, 0, 535, tips5_3.c_str(), wcslen(tips5_3.c_str()));
    TextOut(hdc0, 0, 565, tips6_1.c_str(), wcslen(tips6_1.c_str()));
    TextOut(hdc0, 0, 595, tips6_2.c_str(), wcslen(tips6_2.c_str()));
    TextOut(hdc0, 0, 625, tips6_3.c_str(), wcslen(tips6_3.c_str()));
    ReleaseDC(hWnd, hdc0);
    break;

case WM_LBUTTONDOWN:
    InvalidateRect(hWnd, NULL, true);
    if (IDYES != MessageBox(hWnd, L"你确定要开始测试了吗?", L"测试通知", MB_YESNO | MB_ICONINFORMATION)) {
        break;
    }
    HDC hdc;
    hdc = GetDC(hWnd);
    for (int i = 0; i < questions.size(); i++) {
        TextOut(hdc, 0, 115 + 25 * i, (questions[i].to_string() + L"\n").c_str(),
        wcslen(questions[i].to_string().c_str()));
        /*TextOut(hdc, 0, 5+20*i, problems[i].c_str(), wcslen(problems[i].c_str()));
        TextOut(hdc, 0, 5 + 40* i, choices_a[i].c_str(), wcslen(choices_a[i].c_str()));
        TextOut(hdc, 0, 5 + 60* i, choices_b[i].c_str(), wcslen(choices_b[i].c_str()));
        TextOut(hdc, 0, 5 + 80 * i, choices_c[i].c_str(), wcslen(choices_c[i].c_str()));
        TextOut(hdc, 0, 5 +100 * i, choices_d[i].c_str(), wcslen(choices_d[i].c_str()));
```

```
    TextOut(hdc, 0, 5 + 120 * i, choices_e[i].c_str(), wcslen(choices_e[i].c_str())) ; */
}

ReleaseDC(hWnd, hdc);
break;

case WM_CLOSE:
    s = L"是否真的结束？";
    if (IDYES == MessageBox(hWnd, ss.c_str(), L"退出警告", MB_YESNO))
    {
        DestroyWindow(hWnd);
    }

    break;

case WM_COMMAND:
{
    wmid = LOWORD(wParam);
    // 分析菜单选择:
    switch (wmid)
    {
    case 1:
        if (IDYES == MessageBox(hWnd, L"你确定要提交了吗？", L"通知", MB_YESNO | MB_ICONINFORMATION)) {
            GetWindowText(GetDlgItem(hWnd, 0), ans, 16);
            /*SendMessage(edit, WM_GETTEXT, 15, (LPARAM)ans);*/
            res = TCHAR_to_string(ans);
            for (int i = 0; i < res.size(); i++) {
                if (res.at(i) == 'a' || res.at(i) == 'A') {
                    score = score + 2;
                }
                if (res.at(i) == 'b' || res.at(i) == 'B') {
                    score = score + 3;
                }
                if (res.at(i) == 'c' || res.at(i) == 'C') {
                    score = score + 5;
                }
                if (res.at(i) == 'd' || res.at(i) == 'D') {
                    score = score + 10;
                }
                if (res.at(i) == 'e' || res.at(i) == 'E') {
                    score = score + 15;
                }
            }
            wstring notify = L"你的最终得分为" + IntToString(score);
            MessageBox(hWnd, notify.c_str(), L"通知", NULL);
        }
    }
}
```

```
        break;

    case IDM_ABOUT:
        DialogBox(hInst, MAKEINTRESOURCE(IDD_ABOUTBOX), hWnd, About);
        break;

    case IDM_EXIT:
        DestroyWindow(hWnd);
        break;

    default:
        return DefWindowProc(hWnd, message, wParam, lParam);
    }
}

break;

case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: 在此处添加使用 hdc 的任何绘图代码...
    wstring content1= L"本性格测评是由中国现代心理研究所以著名的美国兰德公司拟制的一套经典心理
测试题为蓝本，根据中国人心理特点加以适当改造后形成的心理测试题。";
    wstring content2 = L"这套性格测试题目目前已被一些著名大公司，如联想、长虹、海尔等公司作为对
员工性格测评的重要辅助试卷，据说效果很好。";
    wstring content3 = L"(Tips:现在已经有人建议将来作为对公务员的必选辅助心理测试推广使用。点
击屏幕任意地方可以打开测试题目！)";

    wstring title = L"东北大学在线性格测试系统";
    TextOut(hdc, 450, 5, title.c_str(), wcslen(title.c_str()));
    TextOut(hdc, 10, 35, content1.c_str(), wcslen(content1.c_str()));
    TextOut(hdc, 10, 65, content2.c_str(), wcslen(content2.c_str()));
    TextOut(hdc, 250, 95, content3.c_str(), wcslen(content3.c_str()));
    EndPaint(hWnd, &ps);
}

break;

case WM_DESTROY:
    PostQuitMessage(0);
    break;

default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}

return 0;
}

// “关于”框的消息处理程序。
INT_PTR CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)
{
    UNREFERENCED_PARAMETER(lParam);
    switch (message)
```

```
{  
    case WM_INITDIALOG:  
        return (INT_PTR)TRUE;  
    case WM_COMMAND:  
        if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)  
        {  
            EndDialog(hDlg, LOWORD(wParam));  
            return (INT_PTR)TRUE;  
        }  
        break;  
    }  
    return (INT_PTR)FALSE;  
}
```

实验二 MFC 可视化程序设计

一、实验内容：

- (1) 模态对话框
- (2) 无模态对话框
- (3) 文件对话框
- (4) 消息对话框
- (5) 对话框类与控件交换
- (6) 动态创建按钮
- (7) 单项选择/多项选择
- (8) 滚动条/滑块
- (9) 另外，在结合上述基本内容的学习基础之上，我还开发了一个简易便携版的 MBTI 性格测试分析系统，基于 MFC 可视化程序设计，实现一定的创新。注意，此部分和实验一的基于 Win32 应用的性格测试分析系统都是性格测试分析系统，但是原理和实现方法不相同噢。具体内容会在后续的实验报告部分详细说明。

二、实验环境：

1. 实验类型：设计型。

2. 实验环境：

操作系统：windows10

调试软件名称及版本号：Microsoft Visual Studio Community 2019 版本 16.11.4

编程语言及版本号：C++/C

上机地点：班级未集中进行实验课

机器台号：华硕飞行堡垒 8

三、实验步骤及实验结果：

(1) 模态对话框

① 步骤：创建工程→插入对话框→添加类→为对话框添加成员函数→编辑菜单项→子菜单添加事件处理程序

② 关键代码：

MainFrm.cpp:

```
void CMainFrame::OnModalDlg0()
{
    CFirstDlg dlg;
    dlg.DoModal();
    // TODO: 在此添加命令处理程序代码
}

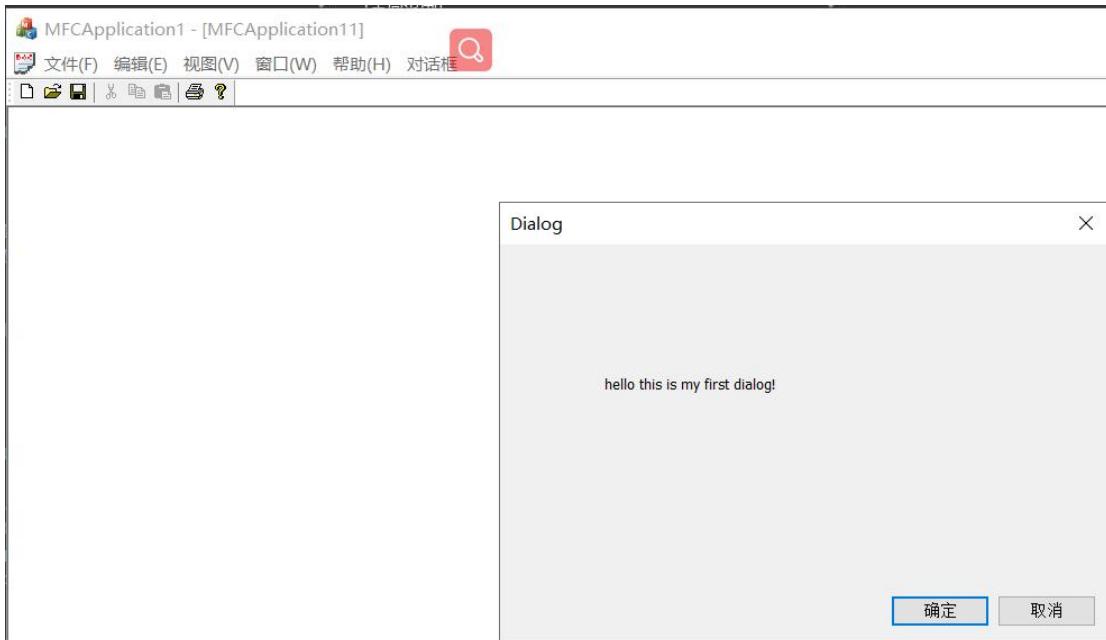
CFirstDlg.cpp:
BOOL CFirstDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();
}
```

```

// TODO: 在此添加额外的初始化
CStatic* pwnd = (CStatic*)GetDlgItem(IDC_STATIC_1);
pwnd->SetWindowText(_T("hello this is my first dialog!"));
return TRUE; // return TRUE unless you set the focus to a control
             // 异常: OCX 属性页应返回 FALSE
}

```

③实验截图



(2) 非模态对话框

①步骤：插入对话框→添加类→为对话框添加成员函数→编辑菜单项→子菜单添加事件处理程序

②关键代码：

MainFrm.cpp:

```

void CMainFrame::OnModelessDlg()
{
    // TODO: 在此添加命令处理程序代码
    CSecondDlg* pdlg = new CSecondDlg;
    pdlg->Create(IDD_DIALOG2);
    pdlg->ShowWindow(SW_SHOWNORMAL);
}

```

CSecond.cpp:

```

// CSecondDlg 消息处理程序
void CSecondDlg::OnOK()
{
    // TODO: 在此添加专用代码和/或调用基类
}

```

```
/*CDialogEx::OnOK();*/
```

```
DestroyWindow();
```

```

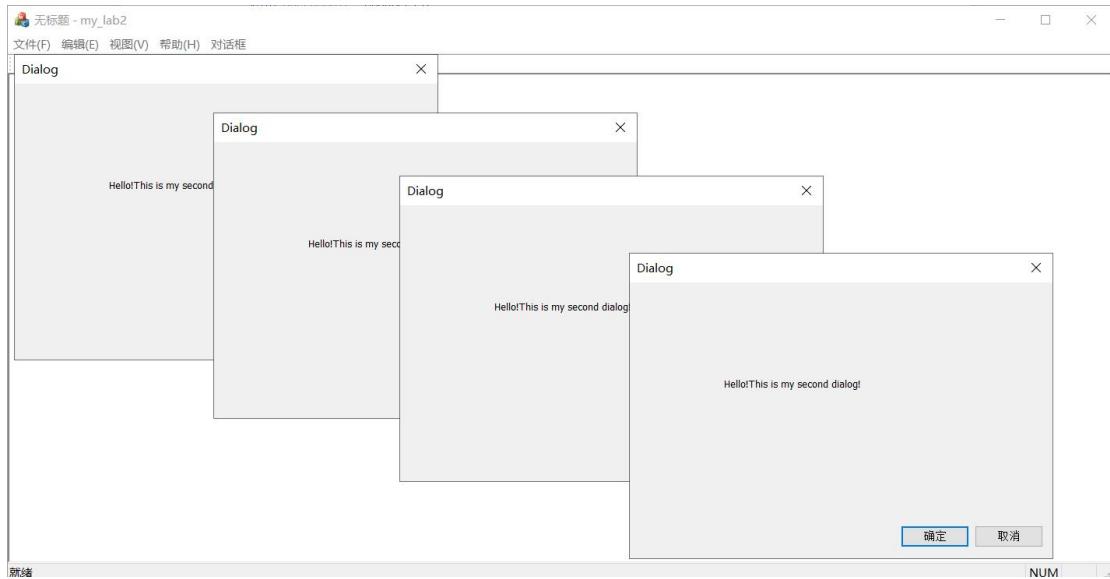
    delete this;
}

void CSecondDlg::OnCancel()
{
    // TODO: 在此添加专用代码和/或调用基类

/*CDialogEx::OnCancel()*/
DestroyWindow();
delete this;
}

```

③实验截图



(3)文件对话框

①步骤：添加子菜单->为子菜单添加事件处理程序

②关键代码：

MainFrm.cpp:

实现文件的单选：

```

void CMainFrame::OnFileDialog()
{
    // TODO: 在此添加命令处理程序代码
    CString filter;
    filter = "文本文件 (*.txt)|*.txt|C++文件 (*.h, *.cpp)|*.h;*.cpp||";
    CFileDialog dlg(TRUE, NULL, NULL, OFN_HIDEREADONLY, filter);
    if (dlg.DoModal() == IDOK) {
        CString str;
        str = dlg.GetPathName();
        AfxMessageBox(str);
    }
}

```

实现文件的多选：

```

void CMainFrame::OnFileDialog()

```

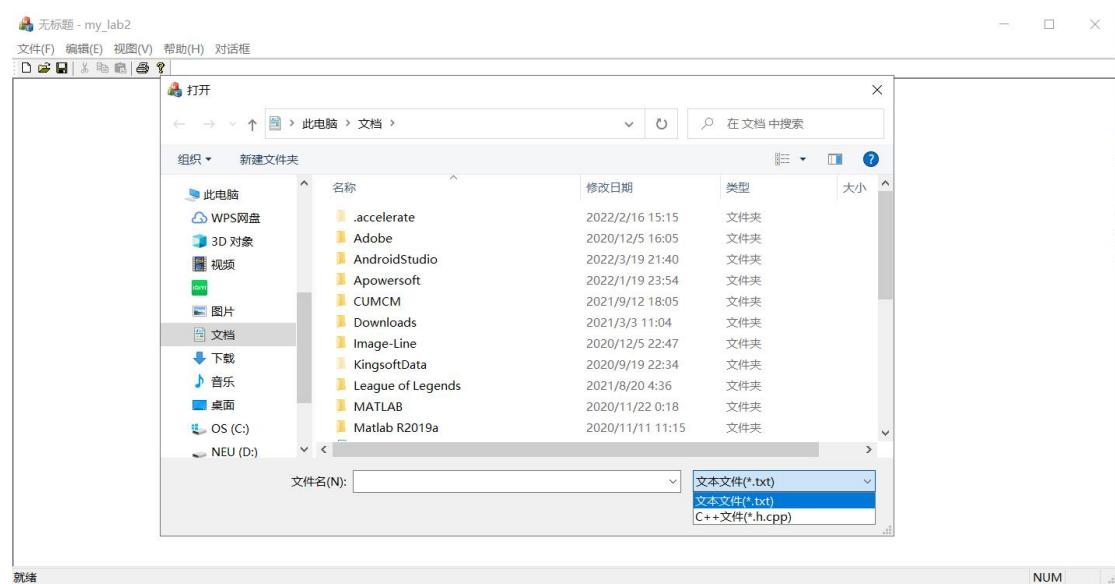
```

{
    CString filter;
    filter = "文本文件 (*.txt)|*.txt|C++文件 (*.h.cpp)|*.h;*.cpp||";
    CFileDialog dlg(TRUE, NULL, NULL, OFN_HIDEREADONLY|OFN_ALLOWMULTISELECT, filter);
    if (dlg.DoModal() == IDOK) {
        CString str;
        POSITION pos;
        pos = dlg.GetStartPosition();
        while (pos != NULL) {
            str = dlg.GetNextPathName(pos);
            AfxMessageBox(str);
        }
    }
}

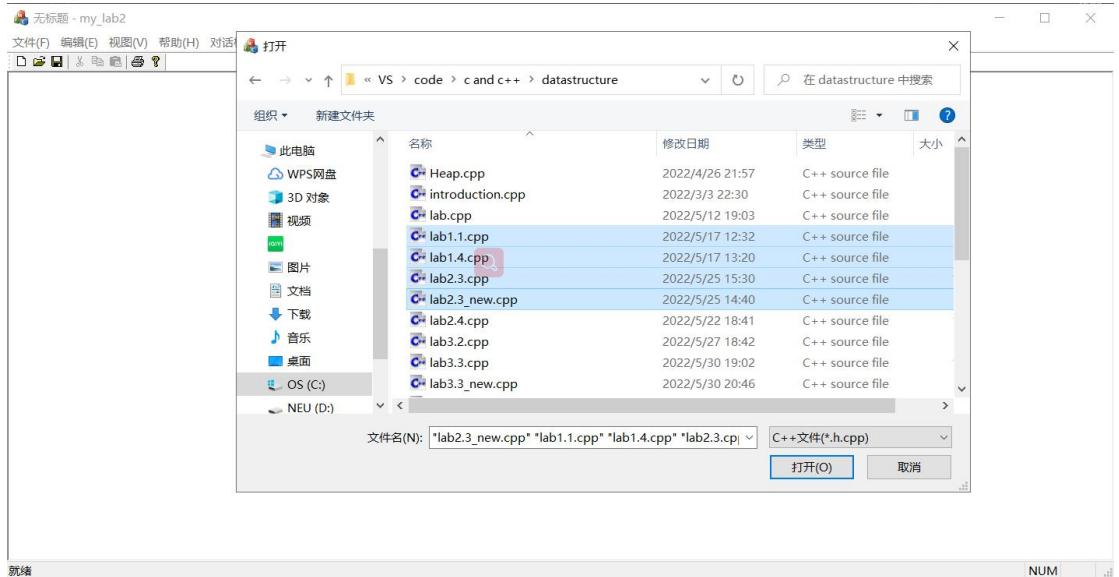
```

③实验截图

实现文件的单选:



实现文件的多选:



(4) 消息对话框

①步骤：添加子菜单->为子菜单添加事件处理程序

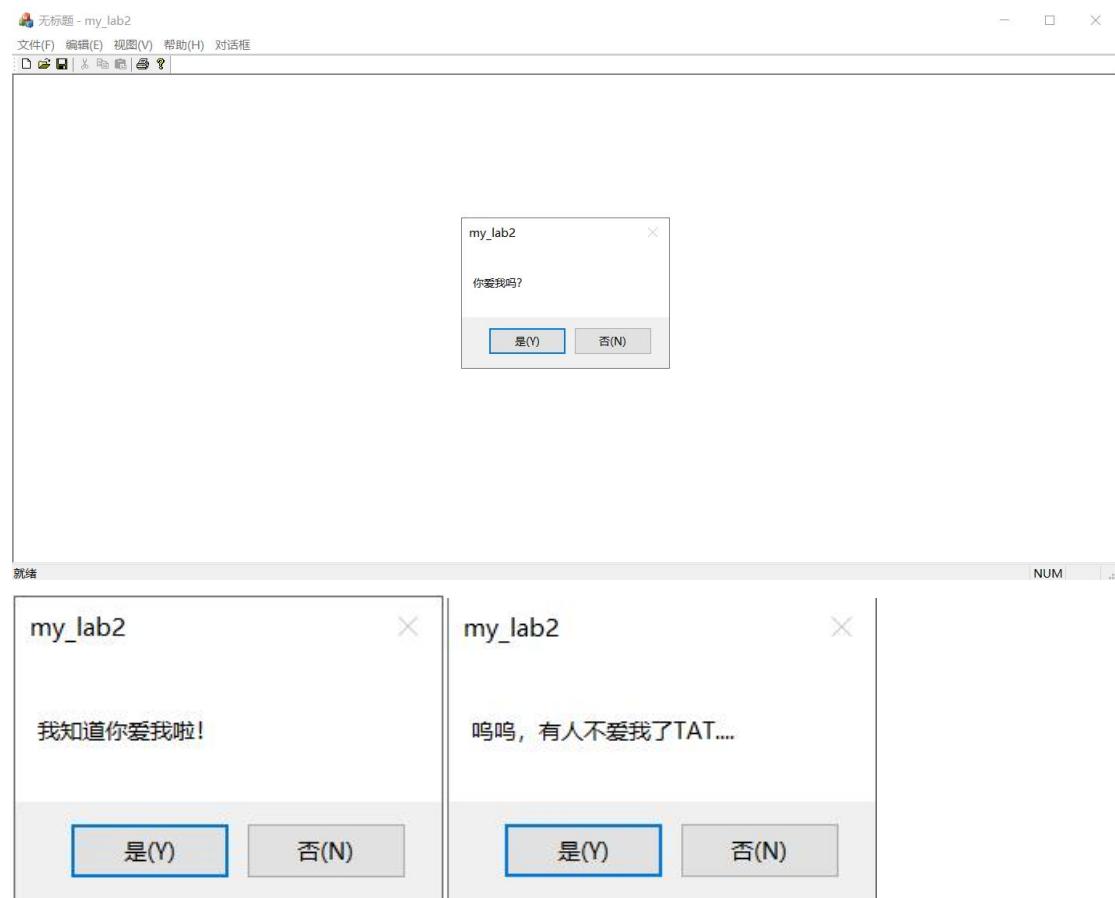
②关键代码：

MainFrm.cpp:

```
void CMainFrame::OnMessageDlg()
{
    // TODO: 在此添加命令处理程序代码
    while(MessageBox(_T("你爱我吗？"), NULL, MB_YESNO) == IDYES) {
        MessageBox(_T("我知道你爱我啦！"), NULL, MB_YESNO);
    }

    while(MessageBox(_T("你爱我吗？"), NULL, MB_YESNO) == IDNO) {
        MessageBox(_T("呜呜， 有人不爱我了 TAT...."), NULL, MB_YESNO);
    }
}
```

③实验截图



(5)对话框类与控件交换

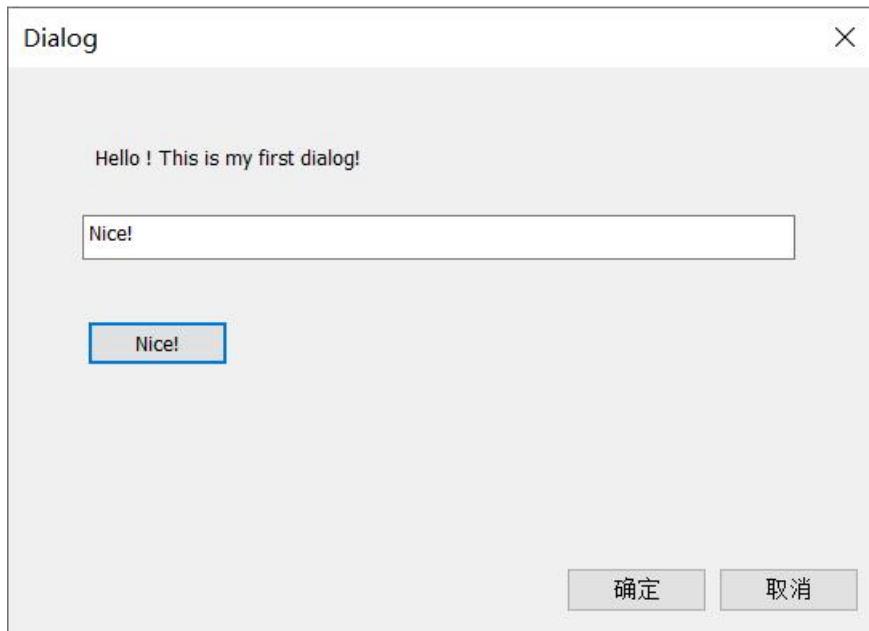
①步骤：添加子菜单->为子菜单添加事件处理程序

②关键代码：

CFirstDlg.cpp:

```
void CFirstDlg::OnBnClickedButton1()
{
    // TODO: 在此添加控件通知处理程序代码
    UpdateData(TRUE);
    m_RelBtn.SetWindowText(m_strEdit);
}
```

③实验截图



(6) 动态添加按钮

①步骤：插入对话框->添加类->为对话框添加成员函数->编辑菜单项->子菜单添加事件处理程序

②关键代码：

```
void CDlgBtnCreate::OnOK()
{
    // TODO: 在此添加专用代码和/或调用基类

    /*CDialogEx::OnOK();*/
    if (m_btnMine.m_hWnd == NULL) {

        m_btnMine.Create(L"动态创建的按钮噢(○o○)！", WS_CHILD | WS_VISIBLE |
BS_PUSHBUTTON, CRect(20, 20, 300, 100), this, 2000);
        GetDlgItem(IDOK)->SetWindowText(L"删除动态创建的按钮噢(○o○)");
    }
    else {
        m_btnMine.DestroyWindow();
        GetDlgItem(IDOK)->SetWindowText(L"运行时创建的按钮噢(○o○)");
    }
}

BOOL CDlgBtnCreate::OnCommand(WPARAM wParam, LPARAM lParam)
{
    // TODO: 在此添加专用代码和/或调用基类
    WORD nCode = HIWORD(wParam);
    WORD nID = LOWORD(wParam);
    if (nID == 2000 && nCode == BN_CLICKED) {
```

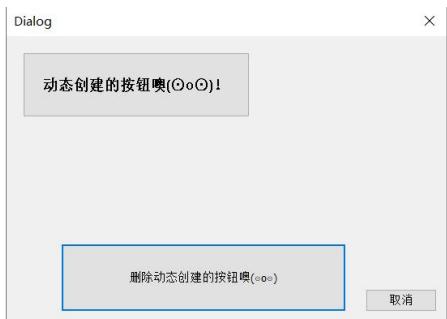
```
    MessageBox(L"你按下了动态创建的按钮(○o○)!" );  
}  
return CDialogEx::OnCommand(wParam, lParam);  
}
```

③实验截图

初始加载时：



点击确定按钮：



点击新创建的按钮：



再点击同一个按钮，实现动态创建的按钮的删除：



(7) 单项选择/多项选择

①步骤：插入对话框->添加类->添加八个 radio buttons->添加四个 check boxes->添加 WM_INITDIALOG message 并编辑代码->添加子菜单->为子菜单添加事件处理程序。

②关键代码：

```
BOOL CDlgNetQuestionary::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // TODO: 在此添加额外的初始化
    CheckRadioButton(IDC AGE L18, IDC AGE M38, IDC AGE 18T27);
    CheckRadioButton(IDC DO POP, IDC DO NEWS, IDC DO POP);
    CButton* pbtn = (CButton*)GetDlgItem(IDC CM_FTTL);
    pbtn->SetCheck(1);

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

void CDlgNetQuestionary::OnOK()
{
    // TODO: 在此添加专用代码和/或调用基类
    CString str, strCtrl;
    str = L"Your age: ";
    UINT nID = GetCheckedRadioButton(IDC AGE L18, IDC AGE M38);
    GetDlgItemText(nID, strCtrl);
    str = str + strCtrl;
    str = str + L"\nwhat you often do online are(is):";
    UINT nID0= GetCheckedRadioButton(IDC DO POP, IDC DO OTHER);
    GetDlgItemText(nID0, strCtrl);
    str = str + strCtrl;
    str = str + L"\n the way you connect to the net are(is):\n";
    UINT ncheckIDs[4] = { IDC_CM_FTTL, IDC_CM_LAN, IDC_CM_56K, IDC_CM_OTHER };
    CButton* pbtn;
    for (int i = 0; i < 4; i++) {
        pbtn = (CButton*)GetDlgItem(ncheckIDs[i]);
        if (pbtn->GetCheck()) {
            pbtn->GetWindowText(strCtrl);
            str = str + strCtrl+L" ";
        }
    }
    MessageBox(str);
    CDialogEx::OnOK();
}
```

③实验截图

上网问卷调查 ×

Age:

<18 18-27 27-38 >38

what often to do when online:

email learning gaming other

The way to surf online:

FTTL OR ADSL LAN 56K OTHER

确定 取消

上网问卷调查 ×

Age:

<18 my_lab2 ×

what often to do when online:

email

The way to surf online:

FTTL OR ADSL

Your age: 18-27
what you often do online are(is):email
the way you connect to the net are(is):
FTTL OR ADSL

确定 取消



(8) 滚动条/滑块

①步骤: a. 步骤: 插入对话框-添加类-添加 Scroll bar 和 Slider 并分别为其添加变量-为 CBkColorDlg 添加两个私有成员-添加消息映射并编辑代码-添加子菜单-为子菜单添加事件处理程序

②关键代码:

```

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

    // TODO: 在此添加额外的初始化
    m_scrollRed.SetScrollRange(0, 255);
    m_sliderBlue.SetRange(0, 255);
    m_sliderGreen.SetRange(0, 255);
    m_nBlue = m_nGreen = m_nRedValue = 192;
    UpdateData(FALSE);
    m_scrollRed.SetScrollPos(m_nRedValue);

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

void CBkColorDlg::OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    int nID = pScrollBar->GetDlgCtrlID();
    if (nID == IDC_SCROLLBAR_RED) {
        switch (nSBCode) {
            case SB_LINELEFT:
                m_nRedValue--;
                break;
        }
    }
}

```

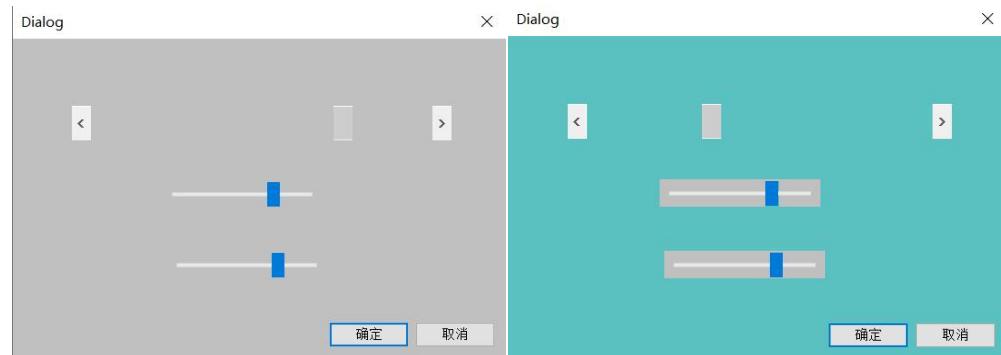
```

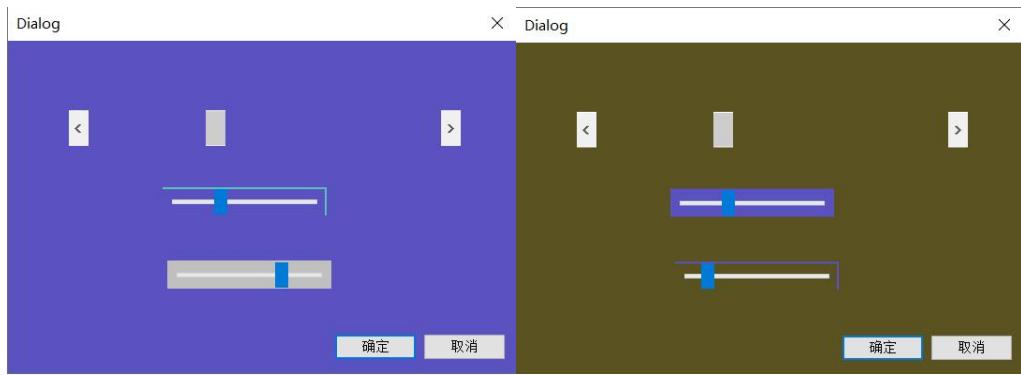
        case SB_LINERIGHT:
            m_nRedValue++;
            break;
        case SB_PAGELEFT:
            m_nRedValue -= 10;
            break;
        case SB_PAGERIGHT:
            m_nRedValue += 10;
            break;
        case SB_THUMBTRACK:
            m_nRedValue = nPos;
            break;
    }
    if (m_nRedValue < 0) {
        m_nRedValue = 0;
    }
    if (m_nRedValue > 255) {
        m_nRedValue = 255;
    }
    m_scrollRed.SetScrollPos(m_nRedValue);
}
Invalidate();
CDialogEx::OnHScroll(nSBCode, nPos, pScrollBar);
}

HBRUSH CBkColorDlg::OnCtlColor(CDC* pDC, CWnd* pWnd, UINT nCtlColor)
{
    UpdateData(TRUE);
    COLORREF color = RGB(m_nRedValue, m_nGreen, m_nBlue);
    m_Brush.Detach();
    m_Brush.CreateSolidBrush(color);
    pDC->SetBkColor(color);
    return (HBRUSH)m_Brush;
}

```

③实验截图



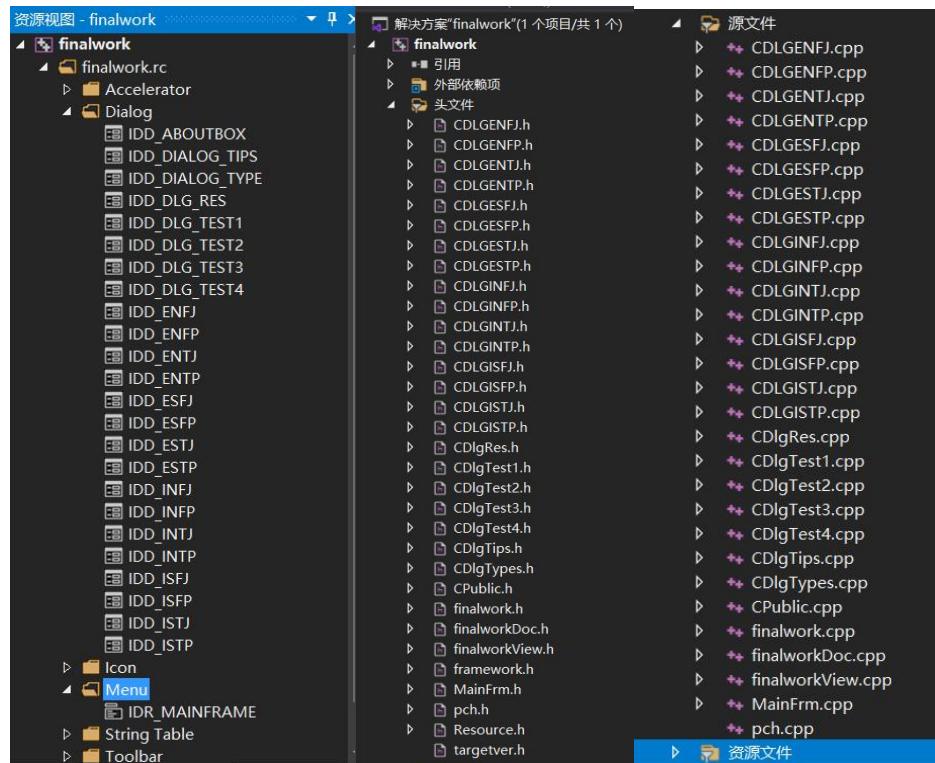


点击三个滑块可以自由调整对话框背景的颜色。

(9) 基于 MFC 可视化程序设计的简易便携版 MBTI 性格测试分析系统

基于 MFC 可视化程序设计的简易便携版 MBTI 性格测试分析系统

① 整体框架



②特殊数据结构说明

新建 CPublic 类，用于存储全局变量：

```
#pragma once
class CPublic
{
public:
    CPublic();
    virtual ~CPublic();

public:
    static int e; //全局变量
    static int i; //全局变量
    static int s; //全局变量
    static int n; //全局变量
    static int t; //全局变量
    static int f; //全局变量
    static int j; //全局变量
    static int p; //全局变量
};

#include "pch.h"
#include "CPublic.h"

int CPublic::e = 0;
int CPublic::i = 0;
int CPublic::s = 0;
int CPublic::n = 0;
int CPublic::t = 0;
int CPublic::f = 0;
int CPublic::j = 0;
int CPublic::p = 0;

CPublic::CPublic()
{
}

CPublic::~CPublic()
{
}
```

③功能分析

本程序是基于 MFC 可视化程序设计的简易便携版 MBTI 性格测试分析系统。主要功能基本上涵盖了 MBTI 性格测试的主要功能。用户可以实现在线答题，答题后查看性格分析报告，也可以查看所以人格的性格特点、性格分析和性格建议等等。

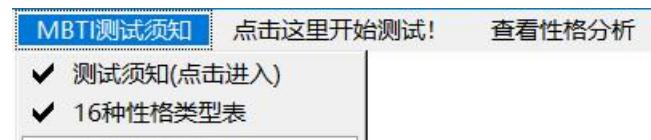
④系统功能截图

1) 进入程序主页面



新增任务栏：MBTI 测试须知，点击这里开始测试，查看性格分析

2) 点击 MBTI 测试须知



I. 首先进入测试须知

东北大学MBTI测试前须知

X

东北大学MBTI测试前须知

系统制作者：软信2001 赖骏鸿

- 1、参加测试的人员请务必诚实、独立地回答问题，只有如此，才能得到有效的结果。
 - 2、《性格分析报告》展示的是你的性格倾向，而不是你的知识、技能、经验。
 - 3、MBTI提供的性格类型描述仅供测试者确定自己的性格类型之用，性格类型没有好坏，只有不同。每一种性格特征都有其价值和优点，也有缺点和需要注意的地方。清楚地了解自己的性格优劣势，有利于更好地发挥自己的特长，而尽可能的在为人处事中避免自己性格中的劣势，更好地和他人相处，更好地作重要的决策。
 - 4、本测试分为四部分，共93题，需时约18分钟。所有题目没有对错之分，请根据自己的实际情况选择。将你选择的A或B所在的○涂黑，例如：●
- 温馨提示：只要你是认真、真实地填写了测试问卷，那么通常情况下你都能得到一个确实和你的性格相匹配的类型。希望你能从中或多或少地获得一些有益的信息。

我已知晓并且同意测试

取消

上面有MBTI性格测试的详细要求与对于测试者的一些温馨提示。点击取消直接关闭窗口。
点击“我已知晓并且同意测试”：

东北大学MBTI测试前须知

X

东北大学MBTI测试前须知

系统制作者：软信2001 赖骏鸿

- 1、参加测试的人员请务必诚实、独立地回答问题，只有如此，才能得到有效的结果。
 - 2、《性格分析报告》展示的是你的性格倾向。
 - 3、MBTI提供的性格类型描述仅供测试者确定自己的性格类型之用，性格类型没有好坏，只有不同。每一种性格特征都有其价值和优点，也有缺点和需要注意的地方。清楚地了解自己的性格优劣势，有利于更好地发挥自己的特长，而尽可能的在为人处事中避免自己性格中的劣势，更好地和他人相处，更好地作重要的决策。
 - 4、本测试分为四部分，共93题，需时约18分钟。所有题目没有对错之分，请根据自己的实际情况选择。将你选择的A或B所在的○涂黑，例如：●
- 温馨提示：只要你是认真、真实地填写了测试问卷，那么通常情况下你都能得到一个确实和你的性格相匹配的类型。希望你能从中或多或少地获得一些有益的信息。

是(Y)

否(N)

我已知晓并且同意测试

取消

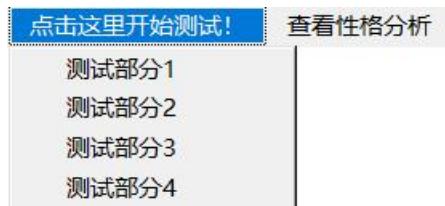
系统提醒用户点击任务栏的右上角开始测试。

II. 查看 16 种性格类型表格



此表格主要列举了基于 MBTI 性格测试的 16 种不同的人格类型。

3) 点击点击这里开始测试



有四个测试部分，系统目前实现前两个。

I. 进入测试一

Dialog

测试部分一

哪一个答案最能贴切的描绘你一般的感受或行为？

1、当你要外出一整天时，你会	<input checked="" type="radio"/> 有计划	<input type="radio"/> 随性随心
2、你认为你自己是一个怎样的人	<input checked="" type="radio"/> 随性而至	<input type="radio"/> 较有调理
3、假如你是老师，你会选教什么课程	<input checked="" type="radio"/> 事实为主	<input type="radio"/> 涉及理论
4、你通常	<input checked="" type="radio"/> 容易自来熟	<input type="radio"/> 沉静和矜持
5、一般来说，你和哪些人比较合得来	<input checked="" type="radio"/> 富于想象力	<input type="radio"/> 现实主义

在下列每一对词语中，哪一個词语更合你心意？请仔细想想这些词语的意义，而不要理会他们的字形或读音。

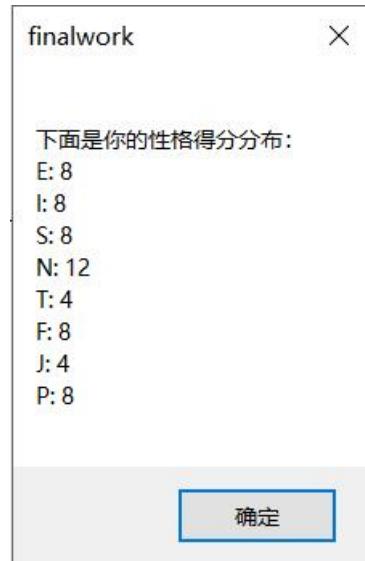
1、	<input checked="" type="radio"/> 注重隐私	<input type="radio"/> 坦率开放
2、	<input checked="" type="radio"/> 抽象	<input type="radio"/> 具体
3、	<input checked="" type="radio"/> 温柔	<input type="radio"/> 坚定
4、	<input checked="" type="radio"/> 思考	<input type="radio"/> 感受
5、	<input checked="" type="radio"/> 冲动	<input type="radio"/> 决定

哪一个答案最能贴切地描绘你一般的感受或行为。

1、要做许多人也做的事情，你比较喜欢	<input checked="" type="radio"/> 按照一般认可的方法	<input type="radio"/> 构想一个独特的想法
2、你刚认识的朋友能否说出你的兴趣	<input checked="" type="radio"/> 可以	<input type="radio"/> 需要一段时间相处
3、哪个是对你来说较高的赞誉	<input checked="" type="radio"/> 一贯感性	<input type="radio"/> 一贯理性
4、和一群人在一起，你通常会	<input checked="" type="radio"/> 只和最熟悉的人交流	<input type="radio"/> 参与大伙交流
5、为乐趣而阅读时，你会	<input checked="" type="radio"/> 喜欢创新新颖	<input type="radio"/> 喜欢开门见山

提交 退出

系统在初始化所有题目时，会默认为所有题目选择 A 选项，以保证用户不会出现漏选的情况。上述为测试一的测试题目，分别从不同维度测试用户的心理。点击提交按钮，系统会提示在本测验各项指标的得分情况：



并且此窗口支持垂直的滑动条：

Dialog

3、假如你正在工作，你会选择以下什么状态

4、你通常

5、一般来说，你和哪些人比较合得来

在下列每一对词语中，哪一個词语更合你心意？请仔细想想这些词语的意义，而不要理会他们的字形或读音。

1、

2、

3、

4、

5、

哪一个答案最能贴切地描绘你一般的感受或行为.

1、要做许多人也做的事情，你比较喜欢

2、你刚认识的朋友能否说出你的兴趣

3、哪个是对你来说较高的赞誉

4、和一群人在一起，你通常会

5、为乐趣而阅读时，你会

事半功倍 涉及理论

容易自来熟 沉静和矜持

富于想象力 现实主义

注重隐私 坦率开放

抽象 具体

温柔 坚定

思考 感受

冲动 决定

按照一般认可的方法 构想一个独特的想法

可以 需要一段时间相处

一贯感性 一贯理性

只和最熟悉的人交流 参与大伙交流

喜欢创新新颖 喜欢开门见山

II. 进入测试二

Dialog

测试部分二

在下列每一对词语中，哪一個词语更合你心意？

选项A	选项B
<input checked="" type="radio"/> 想象的	<input type="radio"/> 真实的
<input checked="" type="radio"/> 仁慈慷慨的	<input type="radio"/> 意志坚定的
<input checked="" type="radio"/> 公正的	<input type="radio"/> 同情的
<input checked="" type="radio"/> 制作	<input type="radio"/> 设计
<input checked="" type="radio"/> 可能性	<input type="radio"/> 必然性
<input checked="" type="radio"/> 温柔	<input type="radio"/> 力量
<input checked="" type="radio"/> 现实主义	<input type="radio"/> 多愁善感
<input checked="" type="radio"/> 制造	<input type="radio"/> 创造
<input checked="" type="radio"/> 新颖的	<input type="radio"/> 已知的
<input checked="" type="radio"/> 同情	<input type="radio"/> 分析
<input checked="" type="radio"/> 坚持己见	<input type="radio"/> 温柔善良
<input checked="" type="radio"/> 全心投入	<input type="radio"/> 决心坚毅
<input checked="" type="radio"/> 具体的	<input type="radio"/> 抽象的
<input checked="" type="radio"/> 能干	<input type="radio"/> 仁慈
<input checked="" type="radio"/> 实际	<input type="radio"/> 创新

系统在初始化所有题目时，会默认为所有题目选择 A 选项，以保证用户不会出现漏选的情况。上述为测试二的测试题目，分别从不同维度测试用户的心理。点击提交按钮，系统会提示在本测验各项指标的得分情况，此分数会累计测试一的分数：

finalwork ×

下面是你的性格得分分布：

E: 8
I: 8
S: 12
N: 15
T: 8
F: 12
J: 4
P: 8

确定

目前，性格测试的答题部分就结束了。

4) 点击查看性格分析

查看性格分析 请在此处键入

✓ 点击查看属于你的性格分析报告---->

- ISTJ
- ISFJ
- INFJ
- INTJ
- ISTP
- ISFP
- INFP
- INTP
- ESTP
- ESFP
- ENFP
- ENTP
- ESTJ
- ESFJ
- ENFJ
- ENTJ

此部分用户可以首先进入查看属于用户的性格分析报告，然后根据性格分析报告所呈现的综合分析，用户可以去选择对应的性格人格，来查看更加详细的性格分析。

I .进入查看性格分析报告



此部分为可视化的性格分析报告，主要利用了滑块的合理使用，滑块的对应的值表示的是相应指标得分占总题目的百分比。系统显示，我的性格为 INFP。

II. 查看 INFP 的详细性格分析

Dialog

性格类型：INFP INFP把内在的和谐视为高于其他一切。他们敏感、理想化、忠诚，对于个人价值具有一种强烈的荣誉感。他们个人信仰坚定，有为自认为有价值的事业献身的精神。INFP型的人对于已知事物之外的可能性很感兴趣，精力集中于他们的梦想和想象。他们思维开阔、有好奇心和洞察力，常常具有出色的长远眼光。在日常事务中，他们通常灵活多变、具有忍耐力和适应性，但是他们非常坚定地对待内心的忠诚，为自己设定了事实上几乎是不可能的标准。INFP型的人具有许多使他们忙碌的理想和忠诚。他们十分坚定地完成自己所选择的事情，

适合领域：创作性、艺术类教育、研究、咨询类

适合职业：各类艺术家、插图画家、诗人、小说家、建筑师、设计师、文学编辑、艺术指导、记者等大学老师（人文类）、心理学工作者、心理辅导和咨询人员、社科类研究人员、社会工作者、教育顾问、图书管理者、翻译家等。

确定 取消

此部分为 INFP 人格的详细性格分析报告。

III. 其他性格类型视图的呈现

Dialog

ISTJ型的人是严肃的、有责任心的和谨慎处理的社会变革分子。他们值得信赖，他们尊重承诺。对他们来说，言语就是严肃的监督，ISTJ型的人工作勤奋，讲求实际，很有头脑也很现实。他们具有很强的集中力、条理性和准确性。无论他们做什么，都相当有条理和可靠。他们真有坚定不移、深思熟虑的思想。一旦他们着手自己相信最好的行动方式时，就很准时地完成得很快。ISTJ型的人特别安静和勤奋，对于细节有很强的记忆和判断。他们能够引证准确的事实支持自己的观点，把过去的经验运用到现在的决策中。他们重视和利用符合逻辑、客观的分析，以

适合领域：工商业领域、政府机构 金融银行业、政府机构技术领域商务领域

适合职业：审计师、会计、财务经理、办公室行政管理、后勤和出产管理、中层经理、公务（法律、税务）执行人员等银行职员、成本会计师、保险精算师、销售人员、证券交易员等机械、工具操作、计算机程序员、地理、气象学家、法律研究员、律师等外科医生、药剂师、实验室技术人员、牙科医生、医学研究员等。

确定 取消

Dialog

ISFJ型的人忠诚、有奉献精神和同情心，理解别人的需要。他们直言清醒而有责任心，乐于为人所需。ISFJ型的人十分努力，他们喜欢平和谦逊的人。他们喜欢利用大量的事实情况，对于细节则有很强的记忆力。他们耐心地对待任务的整个阶段，喜欢事情能够清晰明确。ISFJ型的人具有强烈的职业道德，所以他们如果知道自己的行为真正有用时，会对需要完成之事项承担责任。他们准确系统地完成任务。他们具有传统的价值观，十分保守。他们利用符合实际的判断标准做决定，通过出色的注重实际的态度增加了稳定性。ISFJ型的人平和且虔诚、勤奋严谨。

适合领域：无明显领域特征 医护领域 消费类商业、服务业领域

适合职业：行政管理人员、总经理助理、秘书、人事管理者、项目经理、物流经理、律师助手等外科医生及其它各类医生、家庭医生、牙科医生、护士、药剂师、医学专家、营养学专家、顾问等零售店、精品店店主、大型商场、酒店管理人员、室内设计师等。

确定 取消

Dialog

INTP型的人生活在思想的世界里。他们是独立的、有独创性的思想家，具有强烈的感情、坚定的原则和正直的人格。即使面对怀疑，INTP型的人仍相信自己是正确的。他们对自己有极高的要求，希望看到一切，包括流行观点和存在的权威。这种内在的观念支撑着他们的积极性，通常INTP型的人有本初的洞察力，能够看到事物更深层的含义。即使他人无法分享他们的热情，但灵感对于他们重要且令人信服。

INFP型的人忠诚、坚定、富有理想。他们论理正直，十分坚定以至达到僵硬的地步。因为他们的说服能力，以及对于什么对公共利益最有利有清楚的看法，所以INFP型的人会成为伟大的领导者。由于他们的贡献，他们通常会受到尊重或敬佩。因为珍惜友谊和和平，INFP型的人喜欢说服别人，使之相信他们的观点是正确的。通过运用嘉许和鼓励，而不是争执和挑剔，他们赢得了他人的合作。他们愿意毫无保留地提供帮助，避免争吵。通常INFP型的人是深思熟虑的决策者，他们觉得问题使人兴奋，在行动之前他们通常要仔细地考虑。他们喜欢首次全貌贯注。

适合领域：咨询、教育、科研等领域 文化、艺术、设计等领域

适合职业：心理咨询工作者、心理治疗师、职业指导顾问、大学教师（人文学科、艺术类）、心理学、教育学、社会学、哲学及其它领域的研究人员等作家、诗人、剧作家、电影编剧、电影导演、画家、雕塑家、音乐家、艺术顾问、建筑师、设计师等。

确定 取消

Dialog

INTJ型的人是完美主义者。他们强烈地要求个人自由和能力，同时在他们独创的思想中，不可动摇的信仰促使他们达到目标。INTJ型的人思维严谨、有逻辑性、足智多谋，他们能够看到新计划实行后的结果。他们对自己和别人都很苛求，往往几乎同样强硬地逼迫别人和自己。他们并不十分受冷漠与批评的干扰，作为所有性格类型中最独立的，INTJ型的人更喜欢以自己的方式行事。面对相反意见，他们通常持怀疑态度，十分坚定和坚决。权威本身不能强制他们，只有他们认为这些规则对自己的更重要的目标有用时，才会去遵守。INTJ型的人是天生的阴谋家。

适合领域：科研、科技应用 技术咨询、管理咨询金融、投资领域创造性行业

适合职业：各类科学家、研究所研究人员、设计工程师、系统分析员、计算机程序员、研究开发部经理等各类技术顾问、技术专家、企业管理顾问、投资专家、法律顾问、医学专家、精神性分析学家、经济学家、营销研究人员、证券投资和金融分析师、投资银行家、财务计划人、企业并购专家等各类发明家、建筑师、社论作家、设计师、艺术家等。

确定 取消

<p>Dialog</p> <p>ISFP型的人坦率、诚实、讲求实效，他们喜欢行动而非漫谈。他们很谦逊，对于完成工作的方法有很好的理解力。ISFP型的人擅长分析，所以他们对客观含蓄的原则很有兴趣。他们对于技巧性的事物有天生的理解力，通常善于使用工具和进行手工劳动。他们往往做出有条理而保密的决定。他们善于按照自己的意愿和想象的、合乎逻辑的、有条理的途径去处理问题，但常常是慢条斯理、有目无神。只有理性、可预测的实践能给他们信心。他们重视观察，简而言之就是他们深知知识的宝库。他们是现实主义者，所以能够很好地利用可获得的资源，同时他们善于把握</p> <p>适合领域：技术领域 证券、金融业贸易、商业领域户外、运动、艺术等领域</p> <p>适合职业：机械、电气、电子工程师、各类技术专家和技师、计算机硬件、系统集成专业人员等。证券分析师、金融、财务顾问、经济学研究者等。贸易商、商品经销商、产品代理商（有形产品为主）等。警察、侦探、体育工作者、赛车手、飞行员、雕塑家、手工制作、画家等。</p> <p style="text-align: center;">确定 取消</p>	<p>Dialog</p> <p>ISFP型的人平和、敦厚，他们保持着许多强烈的个人理想和自己的价值观念。他们更多地是通过行为而不是语言表达自己深沉的情感。ISFP型的人谦虚而缄默，但实际上他们是具有巨大的爱受和热情之人。但是除了与他们相知和信赖的人在一起外，他们不经常表现出自我的另一面。因为ISFP型的人不喜欢直接地自我表达，所以常常被误解。ISFP型的人耐心、灵活，很容易与他人相处，很少支配或控制别人。他们很客观，以一种相当实事求是的方式接受他人的行为。他们善于观察周围的人和物，却寻求发现动机和含义。ISFP型的人完全生活在现在，</p> <p>适合领域：手工艺、艺术领域 医疗领域 商业、服务业领域</p> <p>适合职业：时装、首饰设计师、装潢、园艺设计师、陶器、乐器、卡通、漫画制作者、素描画家、舞蹈演员、画家等门诊医生、门诊护士、理疗师、牙科医生、个人健康和运动教练等。餐饮业、娱乐业业主、旅行社销售人员、体育用品、个人护理用品销售员等。</p> <p style="text-align: center;">确定 取消</p>
--	---

<p>Dialog</p> <p>性格类型：INFP INFP把内在的和谐视为高于其他一切。他们敏感、理想化、忠诚，对于个人价值具有一种强烈的荣誉感。他们个人的坚定，有为自己认为有价值的事业献身的精神。INFP型的人对于未知事物之力的可能性深感兴趣，精力集中于他们的理想和想象。他们思维开阔，有好奇心和洞察力，常具有出色的长远眼光。在日常事务中，他们通常灵活多变、具有忍耐力和适应性。但是他们非常坚定地对内心忠诚，为自己设定了事实上几乎是不可能的标准。INFP型的人具有许多使他们忙碌的理想和忠诚。他们十分坚定地完成自己所选择的事情。</p> <p>适合领域：创作性、艺术类 教育、研究、咨询类</p> <p>适合职业：各类艺术家、插图画家、诗人、小说家、建筑师、设计师、文学编辑、艺术指导、记者等文人学者（人文类）、心理学工作者、心理辅导和咨询人员、社科类研究人员、社会工作者、教育顾问、图书管理、图书馆员、翻译家等。</p> <p style="text-align: center;">确定 取消</p>	<p>Dialog</p> <p>性格类型：INTP INT型的人是解决理性问题者。他们很有才智和条理性，以及创造才华的突出表现。INTP型的人外表平静、沉默寡言，内心世界丰富，善于分析和解决问题。他们拘谨、细心、怀旧，他们努力寻找和运用复杂的理解多面性。他们喜欢有组织的活动，但不喜欢有目的的活动。他们为了了解一些无关紧要的细节，不惜花费大量的时间。INTP型的人是逻辑学家，善于推理论证，是富有见解的思考者。他们重视才智，对个人能力有强烈的欲望，有能力也很感兴趣向他人挑战。INTP型的人最主要的兴趣在于理解明显的事物之外的可能性。</p> <p>适合领域：计算机技术 理论研究、学术领域 专业领域创造性领域</p> <p>适合职业：软件设计员、系统分析师、计算机程序员、数据库管理、故障排除专家等大学教授、科研机构研究人员、数学家、物理学家、经济学家、考古学家、历史学家等证券分析师、金融投资顾问、律师、法律顾问、财务专家、侦探等各类型发明家、作家、设计师、音乐家、艺术家、艺术品鉴赏家等。</p> <p style="text-align: center;">确定 取消</p>
---	---

<p>Dialog</p> <p>ESTP型的人不会焦虑，因为他们是快乐的。ESTP型的人活跃、随遇而安、天真直率。他们乐于享受现在的一切而不是为将来计划什么。ESTP型的人很现实，他们信任和依赖于自己对这个世界的感受。他们是好奇而热心的观察者。因为他们接受现在的一切，所以他们思维开阔，能够容忍自己和他人。ESTP型的人喜欢处理、分解与恢复原状的真实事物。ESTP型的人喜欢行动而不是假设，当问题出现时，他们乐于去处理。他们在适当的解决方法上，这是因为它们能够掌握必要的事实情况，然后找到行之有效的明智的解决方案，而无需浪费大量的努力或精力。</p> <p>适合领域：贸易、商业、某些特殊领域 服务业 金融证券业娱乐、体育、艺术领域</p> <p>适合职业：各类贸易商、批发商、中间商、零售商、房地产经纪人、保险经纪人、汽车销售员、私人管家、警察、餐饮、娱乐及其他各类服务业的业主、主管、特许经营权者、自由职业者等。股票经纪人、证券分析师、理财顾问、个人投资者等。娱乐节目主持人、体育节目评论、脱口秀、音乐、音乐、舞蹈表演者、健身教练、体育工作者等。</p> <p style="text-align: center;">确定 取消</p>	<p>Dialog</p> <p>ESTP型的人乐章与人相处，有一种真正的生活热情。他们顽皮活泼，通过真诚和玩笑使别人感到事情更加有趣。ESTP型的人脾气随和、热情奔放，热爱生活和慷慨大方。他们举止粗犷、交际，常常是别人的“沾沾自喜”。他们热情而乐于在各地参加各种活动和节目，而且通常立刻能应对各种活动。ESTP型的人是现实的观察者，他们按照事物的本身去对待并接受它们。他们往往相信自己能够听到、闻到、触摸和看到的事物，而不是依赖于理论上的解释。因为他们喜欢具体的事物，对于细节有很好的记忆力，所以他们能从亲身的经历中学到最好的东西。共同的热爱经验</p> <p>适合领域：消费类商业、服务业领域 广告业、娱乐业领域旅游业、社区服务等其他领域</p> <p>适合职业：精品店、商场销售人员、娱乐、餐饮业客户经理、房地产销售人员、汽车销售人员、市场营销人员（消费品类）等广告企业中的设计师、创意人员、客户经理、时装设计和表演人员、摄影师、节目主持人、脱口秀演员等旅游业中的销售、服务员、导游、社区工作人员、自愿工作者、公共关系专家、健身和运动教练、医护人员等。</p> <p style="text-align: center;">确定 取消</p>
---	--

<p>Dialog</p> <p>ENFP型的人充满热情和新思想。他们乐观、自然、富于创造性和自信，具有独创性的思想和对可能性的强烈渴望。对于ENFP型的人来说，生活是激动人生的戏剧。ENFP型的人对可能的热情和渴望，所以他们对所有事物中的深邃意义（他们具有这种能力），是热情的、有想象力的、有独创性的。ENFP型的人对事物的理解而不是分析。ENFP型的人员具有想象力、适应性和可塑性。他们沉醉于高超的、富于独创性的、富于想象的发明。ENFP型的人不墨守成规，善于发现新事物的新方法，为思想或行为开辟新道路，并保持它们的开放。在完成新想法时</p> <p>适合领域：广告创意、广告撰稿人，市场营销和宣传策划、市场调研人员、艺术指导、公关专家、公司对外发言人等。</p> <p>适合职业：儿童教育老师、大学老师（人文类）、心理学工作者、心理辅导和咨询人员、职业规划顾问、社会工作者、人力资源专家、培训师、演讲家等记者（访谈类）、节目策划和主持人、专栏作家、剧作家、艺术指导、设计师、卡通制作者、电影、电视制片人等。</p> <p style="text-align: center;">确定 取消</p>	<p>Dialog</p> <p>ENTP型的人喜欢兴奋与挑战。他们热情开放、足智多谋、健谈而聪明，擅长于许多事情，不断追求增加能力和个人权力。ENTP型的人天生富有想象力，他们深深地喜欢新思想，留心一切可能性。他们有很强的首创精神，擅于运用创造冲动。ENTP型的人视灵感高于其他的一切，力求使他们的新颖想法转变为现实。他们才华横溢、才多艺、适应性强，在解决挑战性问题时善于随机应变。ENTP型的人员机智而睿智，能够轻易地看到任何情况中的缺点，从而找出需要解决的问题的某方面。他们有极好的分析能力，是出色的策略规划者。他们几乎一直能够为他</p> <p>适合领域：投资项目策划、投资银行、自我创业市场营销、创造性领域 公共关系政治</p> <p>适合职业：投资项目（房地产、金融、贸易、商业等）、各类项目的策划人和发起者、投资银行家、风险投资人、企业业主（新兴产业）等市场营销人员、各类产品销售经理、广告创意、艺术总监、访谈类节目主持人、制片人等公共关系专家、公司对外发言人、社团负责人、政治家等。</p> <p style="text-align: center;">确定 取消</p>
---	--

Dialog

ESTJ 高效率地工作，自我负责，监督他人工作，合理分配和处置资源，主次分明，并井有条；能制定和遵守规则，多喜欢在制度健全、等级分明、比较稳定的企业工作；倾向于选择较为安全的业务，以有形产品为主；喜欢工作中带有和人接触、交流的成分，但不以态度取胜；不特别强调工作的行业或兴趣，多以职业角度看待每一份工作。**ESTJ**型的人很善于完成任务；他们喜欢摸拟局势和促使事情发生；他们具有责任感，信守他们的承诺。他们喜欢理性和能记住和组织安排许多细节。他们及时和尽可能高效率地、系统地开始达到目标。**ESTJ**型的人被激励

适合领域：无明显领域特征

适合职业：大、中型外资企业员工、业务经理、中层经理（多分布在财务、营运、物流采购、销售管理、项目管理、工厂管理、人事行政部门）、职业经理人、各类中小型企业主管和业主。

确定 **取消**

Dialog

ESFJ型的人通过直接的行动和合作积极地以真实、实际的方法帮助别人。他们友好、富有同情心和责任感。**ESFJ**型的人把他们与别人的关系放在十分重要的位置，所以他们往往具有和睦的人际关系，并且通过很大的努力以获得和维持这种关系。事实上，他们常常理想化自己欣赏的人或物。**ESFJ**型的人往往对自己以及自己的成绩十分欣赏，因而他们对于批评或者别人的漠视很敏感。通常他们很果断，表达自己的坚定的主张，乐于事情能很快得到解决。**ESFJ**型的人很现实，他们讲求实际、实事求是和安排有序。他们参与并能记住重要的事情和细节，乐于

适合领域：无明显领域特征

适合职业：办公室行政或管理人员、秘书、总经理助理、项目经理、客户服务部人员、采购和物流管理人员等内科医生及其它各类医生、牙科医生、护士、健康护理指导师、饮食学、营养学专家、小学教师（班主任）、学校管理者、零售业、酒店、大型企业客户服务代表、客户经理、公共关系部主任、商场经理、餐饮业业主和管理人员等。

确定 **取消**

Dialog

ENFJ型的人热爱人类，他们认为人的感情是最重要的。而且他们很自然地关心别人，以热情的态度对待生命，感受与个人相关的所有事物。由于他们很理想化，按照自己的价值观生活，因此**ENFJ**型的人对于他们所尊重和敬佩的人、事业和机构非常忠诚。他们精力充沛、充满热情、富有责任感、勤勤恳恳、锲而不舍。**ENFJ**型的人具有自我批评的自然倾向。然而，他们对他人的情感具有责任心，所以**ENFJ**型的人很少在公共场合批评别人。他们敏锐地意识到什么是（或不是）合适的行为。他们彬彬有礼、富有魅力、讨人喜欢、深谙社会。**ENFJ**型的人具有

适合领域：培训、咨询、教育、新闻传播、公共关系、文化艺术

适合职业：人力资源培训主任、销售、沟通、团队培训员、职业指导顾问、心理咨询工作者、大学教师（人文学科类）、教育学、心理学研究人员等记者、撰稿人、节目主持人（新闻、采访类）、公共关系专家、社会活动家、文艺工作者、平面设计师、画家、音乐家等。

确定 **取消**

Dialog

ENTJ型的人是伟大的领导者和决策人。他们是头脑灵活的思想家和伟大的长远规划者。因为**ENTJ**型的人很有条理和分析能力，所以他们通常对要求推理和才智的任何事情都很容易。为了在完成工作中称职，他们通常会很自然地看出所处情况中可能存在的缺陷，并且立刻知道如何改进。他们力求精通整个体系，而不是简单地把它们视为现有的接受而已。**ENTJ**型的人乐于完成一些需要解决的复杂问题，他们大胆地力求掌握使他们感兴趣的任何事情。**ENTJ**型的人把事实看得高于

适合领域 工商业、政界、金融和投资领域、管理咨询、培训专业性领域

适合职业 各类企业的高级主管、总经理、企业主、社会团体负责人、政治家等投资银行家、风险投资者、股票经纪人、公司财务经理、财务顾问、经济学家、企业管理顾问、企业战略顾问、项目顾问、专项培训师等律师、法官、知识产权专家、大学教师、科技专家等。

确定 **取消**

5) MBTI 性格测试原理简介

人们的性格塑成期通常在青春期，在这之后人们具有了较为稳定的个性类型，并在随后的岁月中动态发展与完善。通常我们觉得随着一个人年龄的增长，他的性格也发生了变化。按照荣格的理论，人的性格一旦成型，就很难发生变化，之所以展现不同的表现方式，正是由于环境、经历等因素的变化，性格在动态地发展，之前不太使用的功能也得到了相应的发挥。如果用左手和右手来做一个比喻的话，一个人的MBTI 倾向就是他最熟悉使用的那只手，随着阅历的增加，他也开始练习使用另外一只手。MBTI 有四个子量表，分别是：

外倾——内倾（E—I），感觉——直觉（S—N），思维——情感（T—F），判断——知觉（J—P）。

第一个维度：根据个人的能量更集中地指向哪里来区分，分为内倾与外倾两种类型（I—E）。

外倾型E	内倾型I
与他人相处精力充沛	独自度过时光精力充沛
希望成为注意的焦点	避免成为注意的焦点
行动，之后思考	思考，之后行动
喜欢边想边说出声	在心中思考问题不善于表露
易于“读”和了解；随意地分享个人信息	相对封闭，更愿意在经挑选的小群体中分享个人的信息
说的比听的多	听的比说的多
高度热情地社交	不把热情表现出来
反应快，喜欢快节奏	仔细考虑后，才有所反应，喜欢慢节奏
重于广度而不是深度	喜欢深度而不是广度

第二个维度：根据个人收集信息的方式不同分为感觉与直觉两种类型（S—N）。

感觉型S	直觉型N
相信确定和有型的事物	相信灵感和推断
喜欢新想法—它们必须有实际意义	喜欢新思想和概念必须符合自己的意愿
重视现实性和常识性	重视想象力和独创力
喜欢使用和琢磨已知的技能	喜欢学习新技能，但掌握之后很容易就厌倦了
留心具体的和特殊的；进行细节描述	留心普遍的和有象征性的；使用隐喻和类比
循序渐进地讲述有关情况	跳跃性地展现事实以一种绕圈子的方式
着眼于现实或现在	着眼于未来

第三个维度：根据个人做决定的方式的不同，可分为思维与情感两种类型（T—F）。

思维型T	情感型F
退后一步思考，对问题进行客观的分析	超前思考，考虑行为对他人的影响
重视符合逻辑、公正、公平的价值；一视同仁	重视同情与和睦；重视准则的例外性
容易发现缺点，有吹毛求疵的倾向，倾向于批评	给人快乐，容易理解别人
被认为冷酷、麻木、漠不关心	被认为感情过多，缺少逻辑性，软弱
认为圆通比坦率更重要	认为圆通与坦率同样重要
只有情感符合逻辑时，才是正确的，才可取	无论是否有意义，认为任何感情都可取
渴望成就而激励	为了获得欣赏而激励

第四个维度：根据个人最感到舒适的生活方式，可分为判断与知觉两种类型（J—P）。

判断型J	知觉型P
做了决定后感到快乐	当各种选择都存在时，感到快乐
有“工作原则”：先工作再玩（如果有时间的话）	“玩的原则”：先玩再完成工作（如果有时间的话）
建立目标，并准时地完成	随着新信息的获取，不断改变目标
愿意知道它们将面对的情况	喜欢适应新情况
着重结果（重点在于完成任务）	着重过程（重点在于如何完成工作）
满足感来源于完成计划	满足感来源于计划的开始
把时间看作有限的资源，认真地对待最后期限	认为时间是可更新的资源，而最后期限也是有收缩的

四个维度、八个端点可组合成下表的 16 种性格类型，你必然属于其中的一种。根据每一项倾向性的判定，最终可以得出十六种组合，如下表所示。这其中的每一种都将是独特的类型，没有哪一种类型最好，也没有哪一种类型不好，要记住的是，这就是你，具有独特风格的你。

十六种性格类型表：

内倾感觉思维判断 (ISTJ)	内倾感觉情感判断 (ISFJ)	内倾直觉情感判断 (INFJ)	内倾直觉思维判断 (INTJ)
内倾感觉思维知觉 (ISTP)	内倾感觉情感知觉 (ISFP)	内倾直觉情感知觉 (INFP)	内倾直觉思维知觉 (INTP)
外倾感觉思维知觉 (ESTP)	外倾感觉情感知觉 (ESFP)	外倾直觉情感知觉 (ENFP)	外倾直觉思维知觉 (ENTP)
外倾感觉思维判断 (ESTJ)	外倾感觉情感判断 (ESFJ)	外倾直觉情感判断 (ENFJ)	外倾直觉思维判断 (ENTJ)

MBTI 通过四项二元轴来测量人在性格和行为方面的喜好和差异。这四项轴分别为：

- 人的注意力集中所在和精力的来源：外向和内向 (Extraversion vs. Introversion)
- 人获取信息的方式：感知和直觉 (Sensing vs. Intuition)
- 人作决策的方式：思考和感觉 (Thinking vs. Feeling)
- 人对待外界和处世的方式：计划性和情绪型 (Judging vs. Perceiving)

四、实验总结：

通过 windows 程序设计的学习，我了解并掌握了 win32 程序的基本原理和 MFC 可视化编程的基本原理。通过学习，我了解到 MFC 编程的实现是通过建立消息映

射表来实现各个函数的调用，知道了一些类的基本使用规则。每种对话框的操作要通过先建立相应的 CDialog 类来具体进行。每个消息通过相关类的函数调用与映射来实现逻辑。通过本次实验，使我更具体的了解了其中的各种类、函数和工具，让我初步接触了如何通过 MFC 实现基本的可视化编程。

本实验的主要创新点是，基于基于 Win32 应用的性格测试分析系统和基于 MFC 可视化程序设计的简易便携版 MBTI 性格测试分析系统。而这虽然都是性格分析系统，但是实现原理和功能不尽相同。在开发中也学习到了很多心理学的知识。

并且在运用 MFC 实现个人开发的程序时，融入了自身比较新颖的想法和观点，在程序构建过程中学到了很多。在此期间也积极和马毅老师沟通，感谢马毅老师对于我疑难的解答。总之，获益匪浅！

五、附录

实验参考资料：

<https://zhuanlan.zhihu.com/p/374121222>

实验源代码：

相应的工程文件打包在对应的文件夹中。

教师评语或评价表格：（任课教师可根据实际情况，做适当调整）

评语及评价表格的字体颜色为红色

评价表格示例：（考核标准与教学大纲中的实验考核标准一致）

考核标准	得分
(1) 遵守课堂秩序，不无故缺席 (20%)	
(2) 实验结果准确，具有一定的创新性 (40%)	
(3) 实验报告规范 (40%)	
总分	