

**Выполнила: Белоусова Е., ИП-911**

## Задача

**Цель:** приобрести навыки разработки Windows приложений с использованием низкоуровневого интерфейса прикладного программирования — Windows API; освоиться с некоторыми механизмами взаимодействия ОС MS Windows с выполняемыми пользовательскими приложениями.

**Задание 1:** протестировать программы, разобранные на лекции 5.

## Описание работы программы

Протестируем шаблон создания оконного приложения Windows.

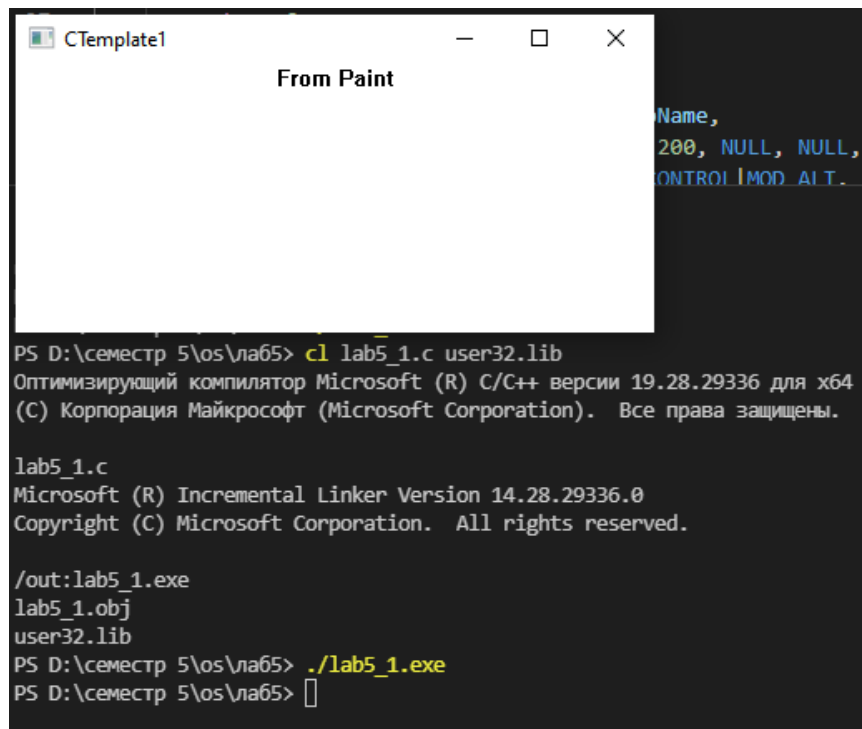


Рисунок 1 Оконное приложение

Попробуем добавить какие-нибудь другие элементы, например, горизонтальную полосу прокрутки.

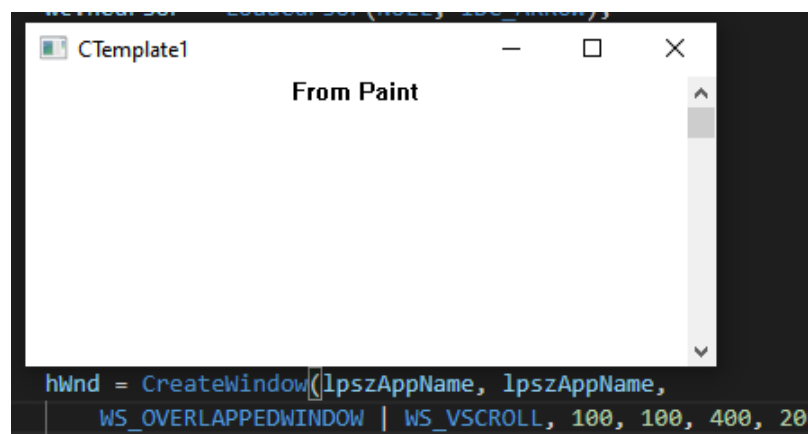


Рисунок 2 Добавили scroll bar

Попробуем поменять иконку курсора.

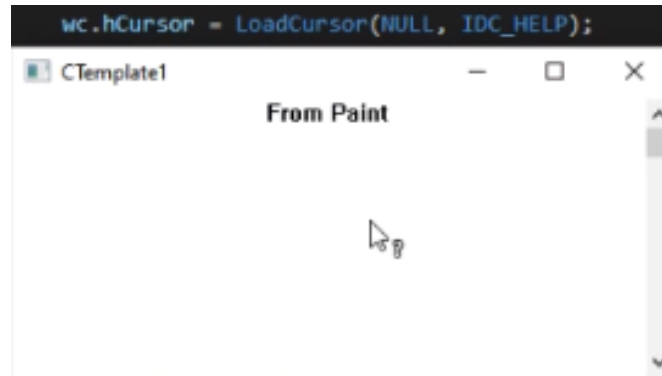


Рисунок 3 Поменяли иконку приложения

Попробуем поменять цвет окна.

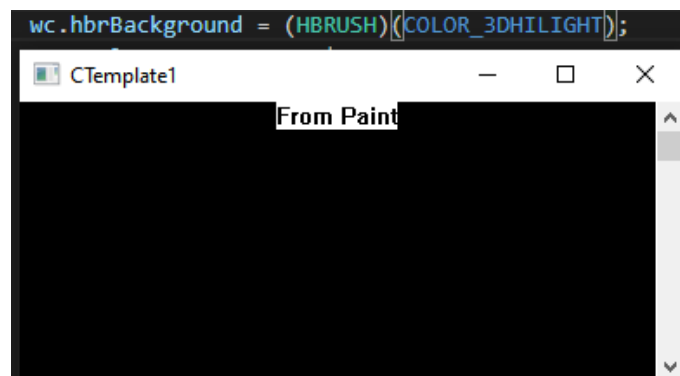


Рисунок 4 Поменяли цвет

Протестируем окно predefined styles.

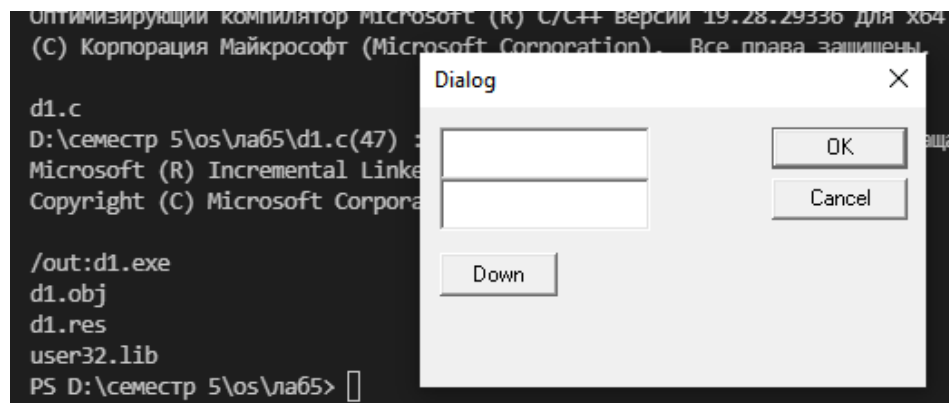


Рисунок 5 Окно predefined styles

Добавим какой-нибудь элемент, например, еще одно поле.

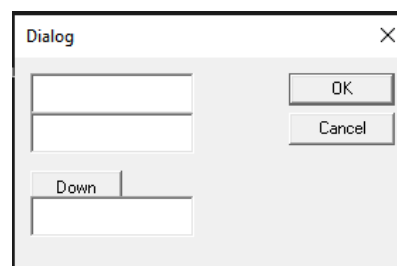


Рисунок 6 Добавили поле

## Листинг

```
//lab5_1.cpp
```

```
//#pragma comment(linker, "/SUBSYSTEM:windows /ENTRY:mainCRTStartup")
#include <windows.h>
```

```
LRESULT CALLBACK MyWndProc(HWND, UINT, WPARAM, LPARAM);
HDC hdc;
RECT rt;
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpCmdLine, int nCmdShow)
{
    HWND hWnd;
    MSG msg;
    WNDCLASS wc;
    BOOL ret;
    LPCSTR lpszAppName = "CTemplate1";
    wc.lpszClassName = lpszAppName;
    wc.hInstance = hInstance;
    wc.lpfnWndProc = (WNDPROC)MyWndProc;
    wc.hIcon = 0;
    wc.hCursor = LoadCursor(NULL, IDC_HELP);
    wc.lpszMenuName = 0;
    wc.hbrBackground = (HBRUSH)(COLOR_WINDOW);
    wc.style = CS_HREDRAW | CS_VREDRAW;
    wc.cbClsExtra = 0;
    wc.cbWndExtra = 0;
    if(!RegisterClass(&wc))
    {
        return 0;
    }

    hWnd = CreateWindow(lpszAppName, lpszAppName,
        WS_OVERLAPPEDWINDOW | WS_VSCROLL, 100, 100, 400, 200, NULL, NULL, hInstance, NULL);
    ret = RegisterHotKey(hWnd, 0xB001, MOD_CONTROL | MOD_ALT, 'W');
    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);
    while (GetMessage(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return msg.wParam;
}

LRESULT CALLBACK MyWndProc(HWND hWnd, UINT Message, WPARAM wparam, LPARAM lparam)
{
    HBRUSH hBrush;
    PAINTSTRUCT ps;
    switch (Message)
    {
        case WM_DESTROY:
            PostQuitMessage(0);
    }
}
```

```

        break;
    case WM_HOTKEY:
        ShowWindow(hWnd, SW_SHOWNORMAL);
        break;
    case WM_PAINT:
        hdc = BeginPaint(hWnd, &ps);
        GetClientRect(hWnd, &rt);
        DrawText(hdc, "From Paint", strlen("From Paint"), &rt, DT_CENTER);
        EndPaint(hWnd, &ps);
        break;
    case WM_COPYDATA:
        GetClientRect(hWnd, &rt);
        hdc = GetDC(hWnd);
        DrawText(hdc, (char*)((COPYDATASTRUCT*)lparam)->lpData, ((COPYDATASTRUCT*)lparam)-
>cbData, &rt, DT_LEFT);
        ReleaseDC(hWnd, hdc);
        break;

    default:
        return DefWindowProc(hWnd, Message, wParam, lParam);
    }
    return 0;
}

```

```

//d1.cpp
#include <windows.h>
#include "resource.h"

```

```

LRESULT CALLBACK DlgProc(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam);
int WINAPI WinMain(HINSTANCE hInstance,
    HINSTANCE hPrevInstance,
    LPSTR lpCmdLine,
    int nCmdShow)
{
    MSG msg;
    DialogBox( hInstance, (LPCTSTR)IDD_DLGTEST, NULL, (DLGPROC)DlgProc);
    while (GetMessage(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return msg.wParam;
}

```

```

LRESULT CALLBACK DlgProc( HWND hDlg, UINT message, WPARAM wParam,
    LPARAM lParam )
{
    char strText[100];
    switch (message)
    {
        case WM_INITDIALOG:
            return TRUE;
        case WM_COMMAND:
            switch (LOWORD(wParam))

```

```

        {
            case IDOK:
                PostQuitMessage(0);
                return TRUE;
            case IDCANCEL:
                PostQuitMessage(0);
                return TRUE;
            case IDC_BTN1:
                GetDlgItemText(hDlg, IDC_EDIT1, strText, 100);
                SetDlgItemText(hDlg, IDC_EDIT2, strText);
                SetDlgItemText(hDlg, IDC_EDIT1, "");
                break;
        }
        break;
    default:
        return FALSE;
}
}

//d1.rc
#include <windows.h>
#include "resource.h"
IDD_DLGTEST DIALOG DISCARDABLE 0, 0, 186, 95
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION |
WS_SYSMENU
CAPTION "Dialog"
FONT 8, "MS Sans Serif"
BEGIN
    DEFPUSHBUTTON "OK", IDOK, 129, 7, 50, 14
    PUSHBUTTON "Cancel", IDCANCEL, 129, 24, 50, 14
    PUSHBUTTON "Down", IDC_BTN1, 7, 49, 43, 15
    EDITTEXT IDC_EDIT1,7,7,77,18, ES_AUTOHSCROLL
    EDITTEXT IDC_EDIT2,7,24,77,18,ES_AUTOHSCROLL
    EDITTEXT IDC_EDIT3,7,60,77,18,ES_AUTOHSCROLL
END

//resource.h

#define IDD_DLGTEST 101
#define IDC_EDIT1 1000
#define IDC_EDIT2 1001
#define IDC_EDIT3 1003
#define IDC_BTN1 1002

//lab5_2.cpp
#include <windows.h>

LRESULT CALLBACK MyWndProc(HWND, UINT, WPARAM, LPARAM);
HDC hdc;
RECT rt;
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpCmdLine, int nCmdShow)
{

```

```

HWND hWnd;
MSG msg;
WNDCLASS wc;
BOOL ret;
LPCSTR lpszAppName = "CTemplate1";
//memset(&wc, 0, sizeof(WNDCLASS));
wc.lpszClassName = lpszAppName;
wc.hInstance = hInstance;
wc.lpfnWndProc = (WNDPROC)MyWndProc;
wc.hIcon = 0;
wc.hCursor = LoadCursor(NULL, IDC_ARROW);
wc.lpszMenuName = 0;
wc.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1);
wc.style = CS_HREDRAW | CS_VREDRAW;
wc.cbClsExtra = 0;
wc.cbWndExtra = 0;
if(!RegisterClass(&wc))
{
    return 0;
}

hWnd = CreateWindow(lpszAppName, lpszAppName,
    WS_OVERLAPPEDWINDOW, 100, 100, 400, 200, NULL, NULL, hInstance, NULL);
ret = RegisterHotKey(hWnd, 0xB001, MOD_CONTROL|MOD_ALT, 'W');
ShowWindow(hWnd, nCmdShow);
UpdateWindow(hWnd);
while (GetMessage(&msg, NULL, 0, 0))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
return msg.wParam;
}

```

```

LRESULT CALLBACK MyWndProc(HWND hWnd, UINT Message, WPARAM wparam, LPARAM lparam)
{
    PAINTSTRUCT ps;
    switch (Message)
    {
        case WM_DESTROY:
            PostQuitMessage(0);
            break;
        case WM_HOTKEY:
            ShowWindow(hWnd, SW_SHOWNORMAL);
            break;
        case WM_PAINT:
            hdc = BeginPaint(hWnd, &ps);
            GetClientRect(hWnd, &rt);
            DrawText(hdc, "From Paint", strlen("From Paint"), &rt, DT_CENTER);
            EndPaint(hWnd, &ps);
            break;
        default:
            return DefWindowProc(hWnd, Message, wparam, lparam);
    }
}

```

```
        return 0;
    }
    int main()
    {
        HWND hWnd;
        COPYDATASTRUCT data;
        char str[80];
        hWnd = FindWindow("CTemplate1", "CTemplate1");
        do
        {
            gets(str);
            data.cbData = strlen(str);
            data.lpData = str;
            SendMessage(hWnd, WM_COPYDATA, (WPARAM)GetFocus(), (LPARAM)&data);
        } while(strcmp(str, "quit"));
        return 0;
    }
```