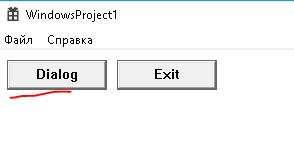
**ЛАБОРАТОРНАЯ РАБОТА №5.** СОЗДАНИЕ ОКОННЫХ ПРИЛОЖЕНИЙ. ОКНО С КНОПКОЙ

**Цель:** ознакомиться с созданием оконных приложений и получить основные навыки создания оконных приложений.

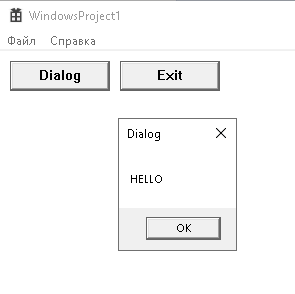
**Ход работы**

**Задание.** Написать программу, в главном окне которой размещены 2 кнопки. Первая кнопка для запуска диалога пользователя. Вторая кнопка для закрытия приложения.

**Результат** работы приложения.



При нажатии на кнопку **«Dialog»** открывается окно с сообщением «Hello».



При нажатии на кнопку **«Exit»** происходит выход из приложения.

Листинг программного кода представлен ниже:

#include "stdafx.h"

#include "WindowsProject1.h"

#include "framework.h"

#define MAX\_LOADSTRING 100

// Глобальные переменные:

HINSTANCE hInst; // текущий экземпляр

WCHAR szTitle[MAX\_LOADSTRING]; // Текст строки заголовка

WCHAR szWindowClass[MAX\_LOADSTRING]; // имя класса главного окна

// Отправить объявления функций, включенных в этот модуль кода:

ATOM MyRegisterClass(HINSTANCE hInstance);

BOOL InitInstance(HINSTANCE, int);

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

INT\_PTR CALLBACK About(HWND, UINT, WPARAM, LPARAM);

HWND hWndExit;

HWND hWndDialog;

int APIENTRY wWinMain(\_In\_ HINSTANCE hInstance,

\_In\_opt\_ HINSTANCE hPrevInstance,

\_In\_ LPWSTR lpCmdLine,

\_In\_ int nCmdShow)

{

UNREFERENCED\_PARAMETER(hPrevInstance);

UNREFERENCED\_PARAMETER(lpCmdLine);

// TODO: Разместите код здесь.

// Инициализация глобальных строк

LoadStringW(hInstance, IDS\_APP\_TITLE, szTitle, MAX\_LOADSTRING);

LoadStringW(hInstance, IDC\_WINDOWSPROJECT1, szWindowClass, MAX\_LOADSTRING);

MyRegisterClass(hInstance);

// Выполнить инициализацию приложения:

if (!InitInstance(hInstance, nCmdShow))

{

return FALSE;

}

HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC\_WINDOWSPROJECT1));

MSG msg;

// Цикл основного сообщения:

while (GetMessage(&msg, nullptr, 0, 0))

{

if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

}

return (int)msg.wParam;

}

// ФУНКЦИЯ: MyRegisterClass()

// ЦЕЛЬ: Регистрирует класс окна.

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\_WINDOWSPROJECT1));

wcex.hCursor = LoadCursor(nullptr, IDC\_ARROW);

wcex.hbrBackground = (HBRUSH)(COLOR\_WINDOW + 1);

wcex.lpszMenuName = MAKEINTRESOURCEW(IDC\_WINDOWSPROJECT1);

wcex.lpszClassName = szWindowClass;

wcex.hIconSm = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI\_SMALL));

return RegisterClassExW(&wcex);

}

// ФУНКЦИЯ: InitInstance(HINSTANCE, int)

// ЦЕЛЬ: Сохраняет маркер экземпляра и создает главное окно

// КОММЕНТАРИИ:

BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)

{

hInst = hInstance; // Сохранить маркер экземпляра в глобальной переменной

HWND hWnd = CreateWindowW(szWindowClass, szTitle, WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT, 0, CW\_USEDEFAULT, 0, nullptr, nullptr, hInstance, nullptr);

if (!hWnd)

{

return FALSE;

}

ShowWindow(hWnd, nCmdShow);

UpdateWindow(hWnd);

return TRUE;

}

// ФУНКЦИЯ: WndProc(HWND, UINT, WPARAM, LPARAM)

// ЦЕЛЬ: Обрабатывает сообщения в главном окне.

// WM\_COMMAND - обработать меню приложения

// WM\_PAINT - Отрисовка главного окна

// WM\_DESTROY - отправить сообщение о выходе и вернуться

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)

{

int wmId, wmEvent;

switch (message)

{

case WM\_CREATE:

hWndExit = CreateWindowEx(NULL,

L"BUTTON",

L"Exit",

WS\_TABSTOP | WS\_VISIBLE |

WS\_CHILD | BS\_DEFPUSHBUTTON,

120,

10,

100,

30,

hWnd,

HMENU(IDC\_MyButtonExit),

hInst,

NULL);

hWndDialog = CreateWindowEx(NULL,

L"BUTTON", L"Dialog",

WS\_TABSTOP | WS\_VISIBLE |

WS\_CHILD | BS\_DEFPUSHBUTTON,

10, 10,

100, 30,

hWnd,

HMENU(IDC\_MyButtonDialog),

hInst, NULL);

case WM\_COMMAND:

{

int wmId = LOWORD(wParam);

wmEvent = HIWORD(wParam);

// Разобрать выбор в меню:

switch (wmId)

{

case IDM\_ABOUT:

DialogBox(hInst, MAKEINTRESOURCE(IDD\_ABOUTBOX), hWnd, About);

break;

case IDM\_EXIT:

DestroyWindow(hWnd);

break;

}

if (LOWORD(wParam) == IDC\_MyButtonExit &&

HIWORD(wParam) == BN\_CLICKED &&

(HWND)lParam == hWndExit)

{

DestroyWindow(hWnd);

}

if (LOWORD(wParam) == IDC\_MyButtonDialog &&

HIWORD(wParam) == BN\_CLICKED &&

(HWND)lParam == hWndDialog)

{

MessageBoxEx(NULL, TEXT("HELLO"), TEXT("Dialog"), MB\_OK, NULL);

}

return 0;

}

break;

case WM\_PAINT:

{

PAINTSTRUCT ps;

HDC hdc = BeginPaint(hWnd, &ps);

// TODO: Добавьте сюда любой код прорисовки, использующий HDC...

EndPaint(hWnd, &ps);

}

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

return DefWindowProc(hWnd, message, wParam, lParam);

}

return 0;

}

// Обработчик сообщений для окна "О программе".

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;

}

**Вывод**: в ходе работы были приобретены навыки создания оконных приложений.