МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

КАФЕДРА ИИТ

ОТЧЁТ

по лабораторной работе №2

**«Создание программы с графическим  
пользовательским интерфейсом в ОС Windows»**

Выполнил:

студент 3 курса

группы ПО-9

Мисиюк Алексей Сергеевич

Проверил:

Козик И. Д.

**Цель работы**: Научиться создавать простейшие приложения с графическим пользовательским интерфейсом в ОС Windows.

**Вариант №3**

Создать приложение с тремя кнопками и полем для отображения текста. По нажатию на кнопку выводить на экран «победу» или «поражение» в зависимости от нажатой кнопки. Определять, какая кнопка будет отвечать за это случайным образом при старте приложения и после нажатия на кнопку.

**Код программы:**

/\* labrab2.cpp

\* ---------------------------------------------------------------------------------------------------

\* Variant #3

\* Создать приложение с тремя кнопками и полем для отображения текста.

\* По нажатию на кнопку выводить на экран «победу» или «поражение» в зависимости от нажатой кнопки.

\* Определять, какая кнопка будет отвечать за это случайным образом при старте приложения и после нажатия на кнопку.

\*/

#include "framework.h"

#include "labrab2.h"

#define MAX\_LOADSTRING 100

// Global Variables:

HINSTANCE hInst; // current instance

WCHAR szTitle[MAX\_LOADSTRING]; // The title bar text

WCHAR szWindowClass[MAX\_LOADSTRING]; // the main window class name

// Forward declarations of functions included in this code module:

ATOM MyRegisterClass(HINSTANCE hInstance);

BOOL InitInstance(HINSTANCE, int);

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

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

int FindWinner();

int APIENTRY wWinMain(\_In\_ HINSTANCE hInstance,

\_In\_opt\_ HINSTANCE hPrevInstance,

\_In\_ LPWSTR lpCmdLine,

\_In\_ int nCmdShow)

{

UNREFERENCED\_PARAMETER(hPrevInstance);

UNREFERENCED\_PARAMETER(lpCmdLine);

// TODO: Place code here.

// Initialize global strings

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

LoadStringW(hInstance, IDC\_LABRAB2, szWindowClass, MAX\_LOADSTRING);

MyRegisterClass(hInstance);

// Perform application initialization:

if (!InitInstance(hInstance, nCmdShow))

{

return FALSE;

}

HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC\_LABRAB2));

MSG msg;

// Main message loop:

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

{

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

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

}

return (int)msg.wParam;

}

//

// FUNCTION: MyRegisterClass()

//

// PURPOSE: Registers the window class.

//

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\_LABRAB2));

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

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

wcex.lpszMenuName = MAKEINTRESOURCEW(IDC\_LABRAB2);

wcex.lpszClassName = szWindowClass;

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

return RegisterClassExW(&wcex);

}

//

// FUNCTION: InitInstance(HINSTANCE, int)

//

// PURPOSE: Saves instance handle and creates main window

//

// COMMENTS:

//

// In this function, we save the instance handle in a global variable and

// create and display the main program window.

//

BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)

{

hInst = hInstance; // Store instance handle in our global variable

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;

}

//

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

//

// PURPOSE: Processes messages for the main window.

//

// WM\_COMMAND - process the application menu

// WM\_PAINT - Paint the main window

// WM\_DESTROY - post a quit message and return

//

//

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

{

switch (message)

{

case WM\_CREATE:

{

// btn1: button

CreateWindowW(TEXT("button"),

TEXT("btn1"),

WS\_VISIBLE | WS\_CHILD,

10, 10, 100, 30,

hWnd, (HMENU)BTN1, NULL, NULL

);

// btn2: button

CreateWindowW(TEXT("button"),

TEXT("btn2"),

WS\_VISIBLE | WS\_CHILD,

120, 10, 100, 30,

hWnd, (HMENU)BTN2, NULL, NULL

);

// btn3: button

CreateWindowW(TEXT("button"),

TEXT("btn3"),

WS\_VISIBLE | WS\_CHILD,

230, 10, 100, 30,

hWnd, (HMENU)BTN3, NULL, NULL

);

// Text lable1: static

hLabel1 = CreateWindowW(TEXT("static"),

TEXT("Text"),

WS\_VISIBLE | WS\_CHILD,

10, 50, 320, 30,

hWnd, (HMENU)LABEL1, NULL, NULL

);

}

break;

case WM\_COMMAND:

{

int wmId = LOWORD(wParam);

int winner;

switch (wmId)

{

case BTN1:

winner = FindWinner();

if (winner == BTN1) {

SetWindowText(hLabel1, TEXT("You won!"));

}

else {

SetWindowText(hLabel1, TEXT("You lost!"));

}

break;

case BTN2:

winner = FindWinner();

if (winner == BTN2) {

SetWindowText(hLabel1, TEXT("You won!"));

}

else {

SetWindowText(hLabel1, TEXT("You lost!"));

}

break;

case BTN3:

winner = FindWinner();

if (winner == BTN3) {

SetWindowText(hLabel1, TEXT("You won!"));

}

else {

SetWindowText(hLabel1, TEXT("You lost!"));

}

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: Add any drawing code that uses hdc here...

EndPaint(hWnd, &ps);

}

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

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

}

return 0;

}

// Message handler for about box.

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;

}

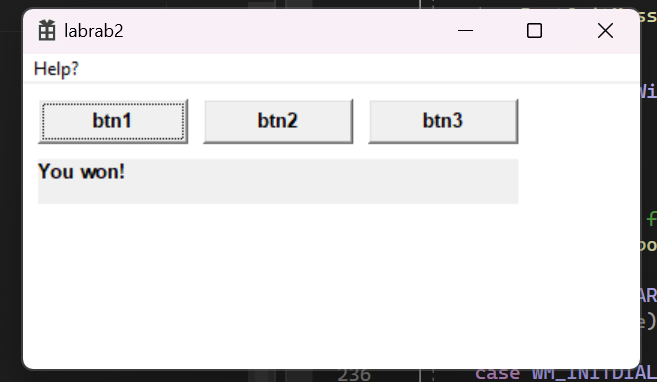
int FindWinner() {

// BTN1 = 201, BTN2 = 202, BTN3 = 203

return rand() % 3 + 200;

}

**Пример работы программы:**

****

