Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №1

з дисципліни:
«Об’єктно-орієнтоване програмування»

Виконав:
Баран Павло Юрійович
студент групи ІП-03
Номер у списку групи: 1

Перевірив:
Порєв В. М.

Київ 2021

Варіанти:

1	Діалогове вікно з повзуном горизонтального скролінгу (Horizontal scroll Bar) та дві кнопки: [Так] і [Відміна]. Рухаючи повзунок скролінгу користувач вводить число у діапазоні від 1 до 100. Після натискування кнопки [Так] вибране число буде відображатися у головному вікні.	Обробка руху повзуна скролінгу – обробка повідомлення **WM_HSCROLL** у віконній функції діалогового вікна. Вивести число у головному вікні – функція **TextOut** в обробнику повідомлення WM_PAINT. Щоб число вивести як текст – функція **itoa** або **ltoa** перетворює ціле число у рядок тексту.
2	Два діалогових вікна. Спочатку з'являється перше, яке має дві кнопки: [Далі >] і [Відміна]. Якщо натиснути кнопку [Далі >], то з'явиться друге длг вікно, яке має три кнопки: [< Назад], [Так] і [Відміна]. Якщо натиснути кнопку [< Назад], то перехід до першого длг вікна.	Аналізувати результат роботи длг вікна можна по значенню, яке повертає DialogBox. Значення визначається другим аргументом функції EndDialog, яка розташовується у віконнній функції. Можна, наприклад: - для кнопки [Далі >] робити виклик функції EndDialog(hDlg, 1) - для кнопки [Відміна] – EndDialog(hDlg, 0) - для кнопки [< Назад] – EndDialog(hDlg, -1)

Програмний код:
Lab1.cpp

```
#include "framework.h"
#include "Lab1.h"
#include "module1.h"
#include "module2_1.h"
#include "module2_2.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

WCHAR textbuf[256];

// 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);

void DoWork1(HWND hWnd);
void DoWork2(HWND hWnd);

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_LAB1, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);

    // Выполнить инициализацию приложения:
    if (!InitInstance (hInstance, nCmdShow))
    {
        return FALSE;
    }

    HACCEL hAccelTable = LoadAccelerators(hInstance,
MAKEINTRESOURCE(IDC_LAB1));

    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_LAB1));
    wcex.hCursor        = LoadCursor(nullptr, IDC_ARROW);
    wcex.hbrBackground  = (HBRUSH)(COLOR_WINDOW+1);
    wcex.lpszMenuName   = MAKEINTRESOURCEW(IDC_LAB1);
    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:
//
//        В этой функции маркер экземпляра сохраняется в глобальной
переменной, а также
//        создается и выводится главное окно программы.
//
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)
{
   switch (message)
   {
   case WM_COMMAND:
      {
         int wmId = LOWORD(wParam);
         // Разобрать выбор в меню:
         switch (wmId)
         {
         case IDM_WORK_1:
            DoWork1(hWnd);
            break;
         case IDM_WORK_2:
            DoWork2(hWnd);
            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);
        TextOutW(hdc, 200, 100, textbuf, 4);
        // 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;
}

void DoWork1(HWND hWnd)
{
    Func_Work1(hInst, hWnd, textbuf);
    InvalidateRect(hWnd, NULL, TRUE);
}

void DoWork2(HWND hWnd)
{
    while (Func_Work2_1(hInst, hWnd))
```

```
    {
        if (!Func_Work2_2(hInst, hWnd)) break;
    }
}
```

module1.cpp

```
#include "framework.h"
#include "resource1.h"

static WCHAR buf[256];
static int pos = 1;

static INT_PTR CALLBACK Work1(HWND hDlg, UINT message, WPARAM
wParam, LPARAM lParam)
{
	HWND hWndScroll = GetDlgItem(hDlg, IDC_SCROLLBAR1);
	UNREFERENCED_PARAMETER(lParam);
	switch (message)
	{
	case WM_INITDIALOG:
		SetScrollRange(hWndScroll, SB_CTL, 1, 100, TRUE);
		return (INT_PTR)TRUE;
	case WM_COMMAND:
		if (LOWORD(wParam) == IDOK)
		{
			EndDialog(hDlg, 1);
			return (INT_PTR)TRUE;
		}
		if (LOWORD(wParam) == IDCANCEL)
		{
			EndDialog(hDlg, 0);
			return (INT_PTR)FALSE;
		}
		break;
	case WM_HSCROLL:
		pos = GetScrollPos(hWndScroll, SB_CTL);
		switch (LOWORD(wParam))
		{
		case SB_LINELEFT:
			pos--;
			break;
		case SB_LINERIGHT:
			pos++;
```

```
                break;
            case SB_THUMBPOSITION:
            case SB_THUMBTRACK:
                pos = HIWORD(wParam);
                break;
            default: break;
            }
            SetScrollPos(hWndScroll, SB_CTL, pos, TRUE);
                break;
        }
        return (INT_PTR)FALSE;
}
int Func_Work1(HINSTANCE hInst, HWND hWnd, WCHAR* dest)
{
        if (DialogBox(hInst, MAKEINTRESOURCE(IDD_DIALOG1), hWnd, Work1))
        {
            _itow_s(pos, buf, 10);
            lstrcpyW(dest, buf);
        }
        return 1;
}
```

module2_1.cpp

```
#include "framework.h"
#include "resource2_1.h"

static INT_PTR CALLBACK Work2_1(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)
            {
                EndDialog(hDlg, 1);
                return (INT_PTR)TRUE;
            }
            if (LOWORD(wParam) == IDCANCEL)
            {
                EndDialog(hDlg, 0);
```

```
                return (INT_PTR)FALSE;
            }
            break;
        }
        return (INT_PTR)FALSE;
}
int Func_Work2_1(HINSTANCE hInst, HWND hWnd)
{
        return DialogBox(hInst, MAKEINTRESOURCE(IDD_DIALOG2_1), hWnd,
Work2_1);
}
```

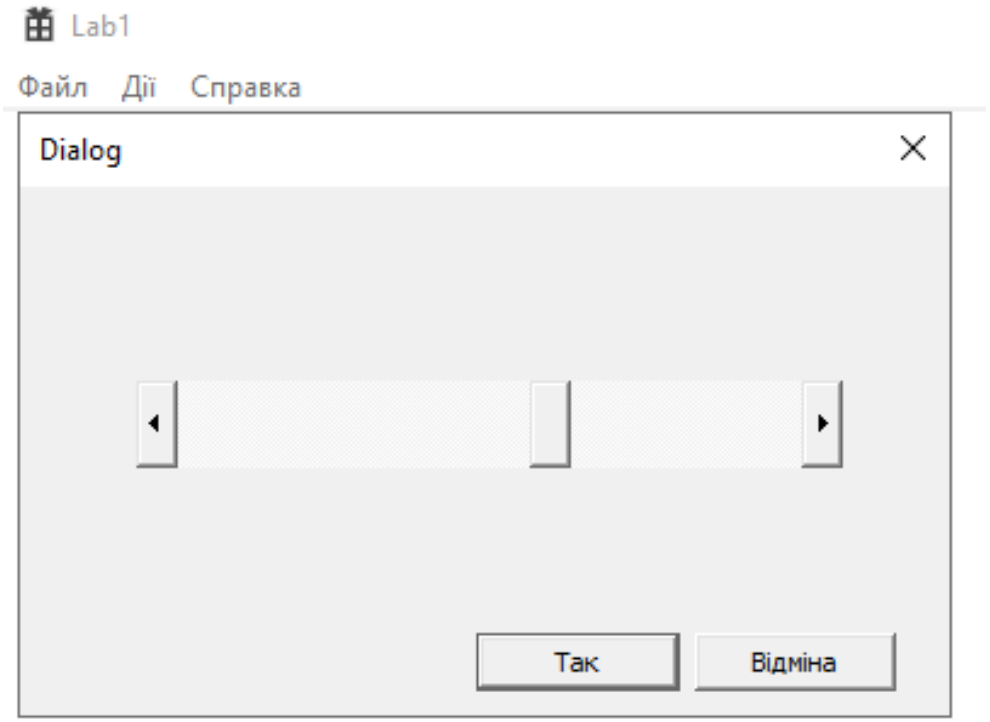
module2_2.cpp

```
#include "framework.h"
#include "resource2_2.h"

static INT_PTR CALLBACK Work2_2(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)
            {
                EndDialog(hDlg, 0);
                return (INT_PTR)FALSE;
            }
            if (LOWORD(wParam) == IDCANCEL)
            {
                EndDialog(hDlg, 0);
                return (INT_PTR)FALSE;
            }
            if (LOWORD(wParam) == IDBACK)
            {
                EndDialog(hDlg, -1);
                return (INT_PTR)FALSE;
            }
            break;
        }
        return (INT_PTR)FALSE;
```

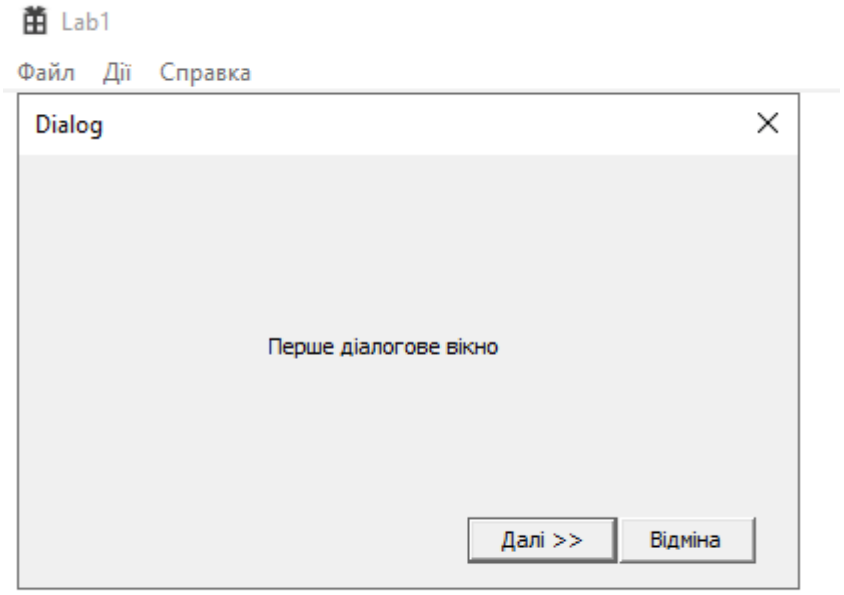
```
}
int Func_Work2_2(HINSTANCE hInst, HWND hWnd)
{
	return DialogBox(hInst, MAKEINTRESOURCE(IDD_DIALOG2_2), hWnd,
Work2_2);
}
```

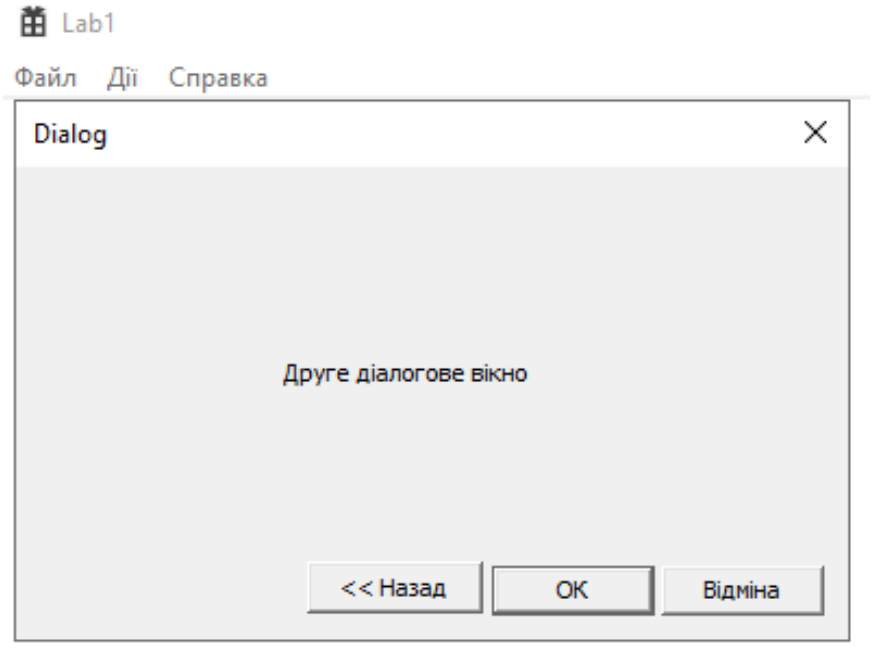
Результати тестування програми:




Lab1
Файл Дії Справка

61


Lab1
Файл Дії Справка
Dialog
Перше діалогове вікно
Далі >>
Відміна


Lab1
Файл Дії Справка
Dialog
Друге діалогове вікно
<< Назад
OK
Відміна