**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
«ООП»

Виконала:

студентка групи ІМ-21
Рабійчук Дар’я
номер у списку групи: 19

Перевірив:

Порєв В. М.

Київ 2023

**Мета:** отримати перші навички створення програм для Windows на основі проєктів для Visual C++ (Visual Studio) з використанням Windows API (WPF) і навчитися модульному програмуванню на C++ (C#)

**Завдання:**
1. Створити у середовищі Visual Studio  проєкт з ім'ям Lab1.
2. Написати вихідний текст програми згідно варіанту завдання.
3. Скомпілювати вихідний текст і отримати виконуваний файл програми.
4. Перевірити роботу програми. Налагодити програму.
5. Проаналізувати та прокоментувати результати та вихідний текст програми.

**Варіант 0:**
Вікно діалогу для вводу тексту, яке має стрічку вводу (Edit Control) та дві кнопки: [Так] і [Відміна]. Якщо ввести рядок тексту і натиснути [Так], то у головному вікні повинен відображатися текст, що був введений.

**Варіант 3:**
Вікно діалогу з елементом списку (List Box) та двома кнопками: [Так] і [Відміна]. У список автоматично записуються назви груп нашого факультету. Якщо вибрати потрібний рядок списку і натиснути [Так], то у головному вікні повинен відображатися текст вибраного рядка списку.

**Текст програми Lab1.cpp:**

```
#include "framework.h"
#include "Lab1.h"
#include "module1.h"
#include "module2.h"
#include <tchar.h>
#include <string>

#define MAX_LOADSTRING 100

HINSTANCE hInst;                                // текущий экземпляр
WCHAR szTitle[MAX_LOADSTRING];                  // Текст строки заголовка
WCHAR szWindowClass[MAX_LOADSTRING];            // имя класса главного окна

ATOM                MyRegisterClass(HINSTANCE hInstance);
BOOL                InitInstance(HINSTANCE, int);
LRESULT CALLBACK    WndProc(HWND, UINT, WPARAM, LPARAM);
INT_PTR CALLBACK    About(HWND, UINT, WPARAM, LPARAM);
void OnWork1(HWND hWnd);
void OnWork2(HWND hWnd);
LPCWSTR res1 = L"";
LPCWSTR res2 = L"";


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.

    LoadStringW(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadStringW(hInstance, IDC_LAB1, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);

    if (!InitInstance(hInstance, nCmdShow))
    {
        return FALSE;
    }

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

    MSG msg;

    while (GetMessage(&msg, nullptr, 0, 0))
    {
        if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }

    return (int)msg.wParam;
}

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);
}

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;
}

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
    case WM_COMMAND:
    {
        int wmId = LOWORD(wParam);
        switch (wmId)
        {
        case ID_ACTIONS_WORK1:
            OnWork1(hWnd);
            break;
        case ID_ACTIONS_WORK2:
            OnWork2(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);
        if (lstrlen(res1) != 0) {
            TextOut(hdc, 50, 50, L"Module 1 res: ", lstrlen(L"Module 1 res: "));
            TextOut(hdc, 50, 70, res1, lstrlen(res1));
        }
        else if (lstrlen(res2) != 0) {
            TextOut(hdc, 50, 50, L"Module 2 res: ", lstrlen(L"Module 2 res: "));
            TextOut(hdc, 50, 70, res2, lstrlen(res2));
        }

        EndPaint(hWnd, &ps);
    }
    break;
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
    default:
        return DefWindowProc(hWnd, message, wParam, lParam);
    }
    return 0;
}

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 OnWork1(HWND hWnd) {
    res1 = Func_MOD1(hWnd, hInst);
    res2 = L"";
    InvalidateRect(hWnd, NULL, TRUE);
}

void OnWork2(HWND hWnd) {
    res2 = Func_MOD2(hWnd, hInst);
    res1 = L"";
    InvalidateRect(hWnd, NULL, TRUE);
}
```

## Текст програми module1.cpp

```
#include "framework.h"
#include <string>
#include "resource.h"

LPCWSTR str;
static INT_PTR CALLBACK Work1Dlg(HWND hDlg, UINT message, WPARAM wParam, LPARAM
lParam)
{
    std::wstring groupStr;
    UNREFERENCED_PARAMETER(lParam);
    switch (message)
    {
    case WM_INITDIALOG:
        for (int index = 1; index <= 4; index++) {
            groupStr = L"IM-2" + std::to_wstring(index);
            SendDlgItemMessage(hDlg, IDC_LIST1, LB_ADDSTRING, 0,
(LPARAM)groupStr.c_str());
        }
        return (INT_PTR)TRUE;

    case WM_COMMAND:
        if (LOWORD(wParam) == IDOK) {
            int indx = SendDlgItemMessage(hDlg, IDC_LIST1, LB_GETCURSEL, 0, 0);
            int length = SendDlgItemMessage(hDlg, IDC_LIST1, LB_GETTEXTLEN, indx,
0);
            TCHAR* buff = new TCHAR[length++];
            SendDlgItemMessage(hDlg, IDC_LIST1, LB_GETTEXT, indx, (LPARAM)buff);
            str = buff;

            EndDialog(hDlg, LOWORD(wParam));
            return (INT_PTR)TRUE;

        }
        else if (LOWORD(wParam) == IDCANCEL)
        {
            EndDialog(hDlg, LOWORD(wParam));
            return (INT_PTR)FALSE;
        }
        break;
```

```
    }
    return (INT_PTR)FALSE;
}

LPCWSTR Func_MOD1(HWND hWnd, HINSTANCE hInst) {
    DialogBox(hInst, MAKEINTRESOURCE(IDD_DIALOG_LIST), hWnd, Work1Dlg);
    return str;
}
```

## Текст програми module2.cpp

```
#include "framework.h"
#include <string>
#include "resource.h"

LPTSTR str;
LPTSTR buff;
static INT_PTR CALLBACK Work2Dlg(HWND hDlg, UINT message, WPARAM wParam, LPARAM
lParam)
{
    std::wstring groupStr;
    UNREFERENCED_PARAMETER(lParam);
    switch (message)
    {

    case WM_INITDIALOG:
        return (INT_PTR)TRUE;

    case WM_COMMAND:
        if (LOWORD(wParam) == IDOK) {
            int length = 20;
            buff = new TCHAR[20];
            GetDlgItemText(hDlg, IDC_INPUT_EDIT, buff, length);
            str = buff;

            EndDialog(hDlg, LOWORD(wParam));
            return (INT_PTR)TRUE;

        }
        else if (LOWORD(wParam) == IDCANCEL)
        {
            EndDialog(hDlg, LOWORD(wParam));
            return (INT_PTR)FALSE;
        }
        break;
    }
    return (INT_PTR)FALSE;
}

LPCWSTR Func_MOD2(HWND hWnd, HINSTANCE hInst) {
    DialogBox(hInst, MAKEINTRESOURCE(IDD_INPUT_DIALOG), hWnd, Work2Dlg);
    return str;
}
```

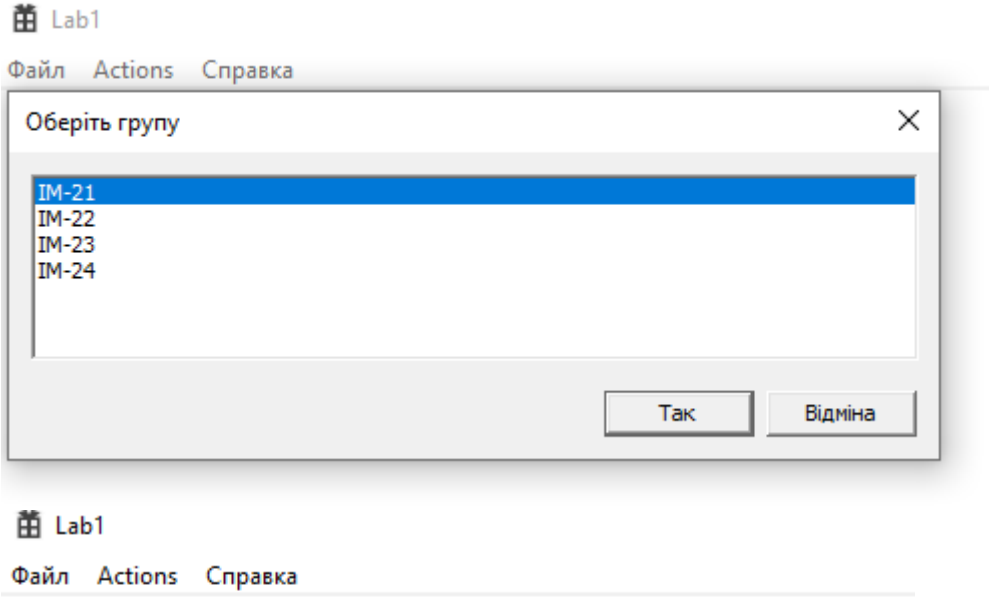
## Текст програми module1.h:

```
#pragma once
extern LPCWSTR Func_MOD1(HWND hWnd, HINSTANCE hInst);
```

## Текст програми module2.h:

```
#pragma once
extern LPCWSTR Func_MOD2(HWND hWnd, HINSTANCE hInst);
```

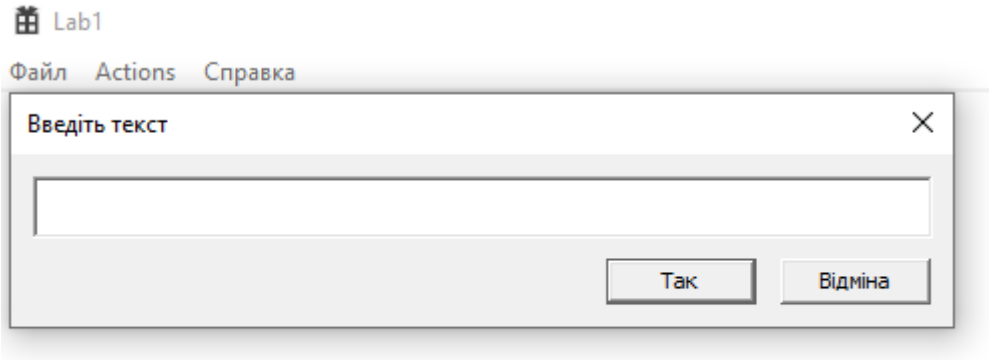
**Результат роботи програми:**







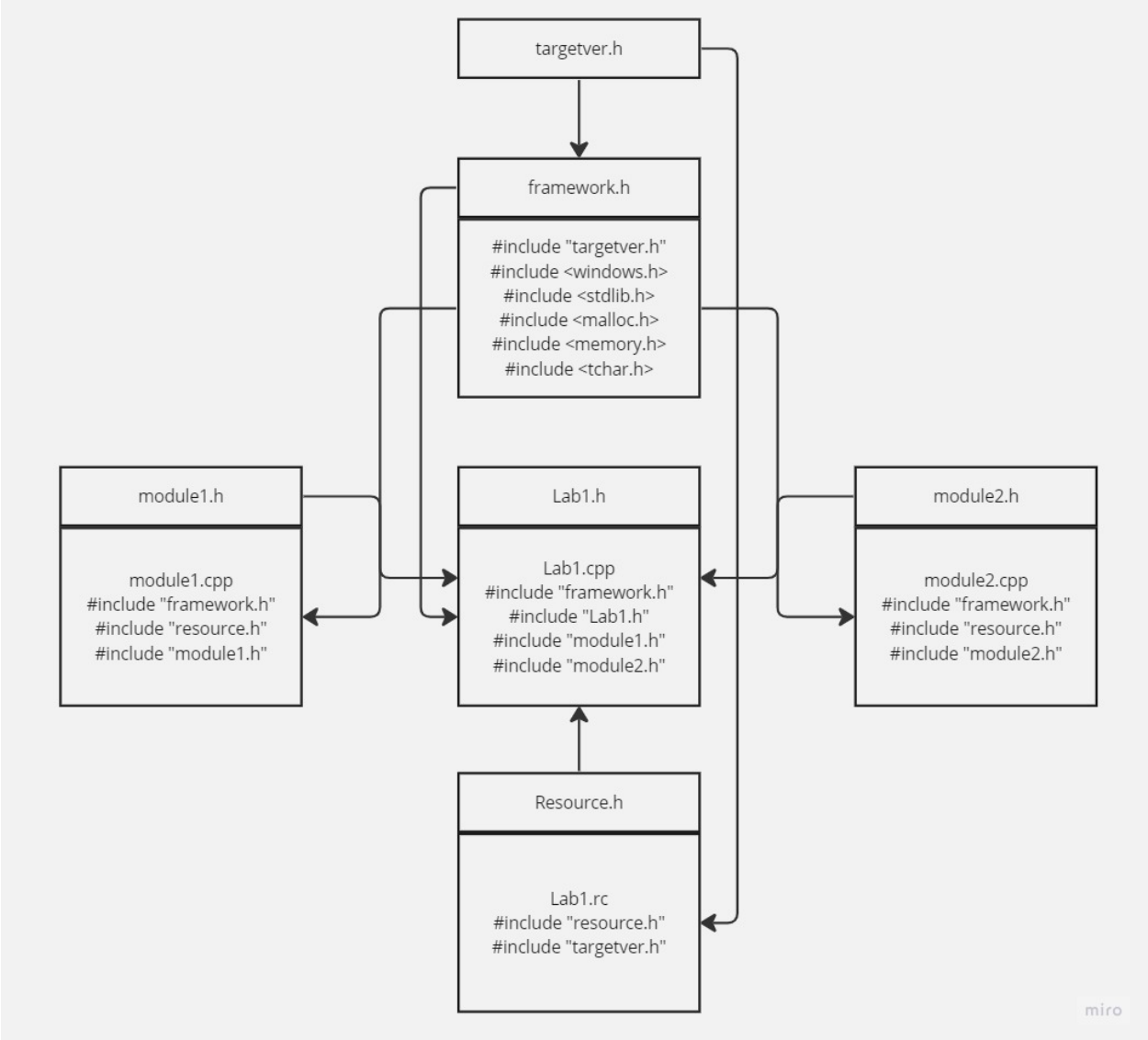
**Module 1 res:**
**IM-21**





**Module 2 res:**
**Hello world!**

**#include-залежності файлів проекту:**




**Висновок:**
Ми навчились працювати з Visual Studio з використанням Windows API (WPF) і вивчили основи модульного програмування.