

Системне програмування. Семестр 2

Лабораторна робота №1

Ознайомлення з написанням програм з графічним інтерфейсом

Діалогові вікна користувача

1. Вступ

Для написання Win32-програми рекомендовано створити **папку** для нового проекту.

Проект для програми з графічним інтерфейсом складається, як правило, з таких файлів:

- 1) файл вихідного коду (*.asm файл);
- 2) ресурси – готові іконки, рисунки, звуки, і взагалі довільні заготовки;
- 3) файл-скрипт ресурсів. В ньому міститься список та опис всіх ресурсів (розширення *.rc);
- 4) інші допоміжні файли (*.inc, ...).

2. Побудова ехе-програми

Для побудови ехе-програми, на відміну від консольної програми, потрібно додатково компілювати ресурси.

Якщо проект складається з трьох файлів, напр.

```
Lab_01.asm  
Sharp.ico  
Res.rc
```

то BAT-файл для створення виконавчого модуля (ехе-програми) буде десь таким:

```
Build.bat  
C:\masm32\bin\ml /c /coff Lab_01.asm  
  
C:\masm32\bin\rc Res.rc  
C:\masm32\bin\cvtres /MACHINE:ix86 Res.res  
  
C:\masm32\bin\link /subsystem:windows Lab_01.obj Res.obj  
  
pause
```

3. Структура вихідного коду

Головний код програми поміщається у функцію WinMain(••••).

Тому (майже) кожна програма, що ми будемо писати, має таку структуру:

```
WinMain PROTO :DWORD, :DWORD, :DWORD, :DWORD
```

```
.data  
hInstance dd 0  
CommandLine dd 0  
; ..... інші змінні .....
```

```
start:
```

```
invoke GetModuleHandle, NULL  
mov hInstance, eax
```

```
invoke GetCommandLine  
mov CommandLine, eax
```

```
invoke WinMain, hInstance, NULL, CommandLine, SW_SHOWDEFAULT  
invoke ExitProcess, eax
```

```
; -----  
WinMain proc hInst:DWORD, hPrevInst:DWORD, CmdLine:DWORD, CmdShow:DWORD
```

```
; ... текст процедури ...
```

```
WinMain endp
```

```
end start
```

4. Задання параметрів діалогового вікна

Шаблон діалогового вікна задається у файлі-скрипті ресурсів найчастіше у такому форматі:

```
діалогID DIALOGEX x, y, cx, cy
[CAPTION "Назва заголовку"]
[CLASS класID]
[STYLE стилі]
[EXSTYLE розширені_стилі]
[MENU ідентифікатор_меню]
[FONT розмір, стиль [, ширина, нахил]]
BEGIN
    CONTROL      "Назва", ID, назва_класу, стиль, x, y, cx, cy [, ex-style]
    staticClass   "Назва", ID, x, y, cx, cy [, style [, ex-style]]
    buttonClass   "Назва", ID, x, y, cx, cy [, style [, ex-style]]
    editClass     ID, x, y, cx, cy [, style [, ex-style]]
    ICON         іконкаID, ID, x, y, cx, cy
END
```

де діалогID – ідентифікатор: 16-бітове число від 1 до 65535, або унікальне ім'я (без лапок).

CLASS класID – ідентифікатор класу: число від 1 до 65535, або рядок, поміщений в лапки;

STYLE стилі – комбінація DS_ та WS_ стилів.

EXSTYLE розширені_стилі – комбінація WS_EX_ стилів.

(CONTROL) назва_класу: назва свого класу (в лапках), або один з перелічених стандартних класів:

BUTTON, STATIC, EDIT, LISTBOX, SCROLLBAR, COMBOBOX.

(CONTROL) стиль: WS_ стиль та, в залежності від типу класу, BS_, SS_, ES_, LBS_, SBS_, CBS_ стилі.

staticClass – один з керуючих елементів:

LTEXT, RTEXT, CTEXT.

buttonClass – один з керуючих елементів:

PUSHBUTTON, DEFPUSHBUTTON, CHECKBOX, AUTOCHECKBOX,
RADIOBUTTON, AUTORADIOBUTTON, PUSHBOX, LISTBOX, GROUPBOX.

editClass – один з керуючих елементів:

EDITTEXT, BEDIT, HEDIT, IEDIT.

5. Функції

INT **WinMain** (hInstance, hPrevInstance, lpCmdLine, nCmdShow).

Дія: головна функція Win32-програми (назва може бути й іншою).

Параметри:

hInstance – ідентифікатор поточного екземпляру програми;

hPrevInstance – ідентифікатор попереднього екземпляру програми (для Win32-програм = NULL);

lpCmdLine – вказівник на командний рядок;

nCmdShow – тип початкового вигляду (поведінки) вікна (як правило, = SW_SHOWDEFAULT).

INT **MessageBox** (hWnd, lpText, lpCaption, uType).

Дія: виведення панелі з повідомленням.

Параметри:

hWnd – handle вікна, якому належить повідомлення (може бути NULL – "нічیه повідомлення");

lpText – текст повідомлення;

lpCaption – заголовок панелі повідомлення,

uType – тип повідомлення:

MB_OK, MB_OKCANCEL, MB_ABORTRETRYIGNORE, MB_YESNOCANCEL, MB_YESNO, ...;

іх можна комбінувати (напр., оператором додавання) з додатковими типами:

MB_ICONSTOP, MB_ICONQUESTION, MB_ICONEXCLAMATION, MB_ICONINFORMATION, MB_ICONASTERISK;

MB_DEFBUTTON1, MB_DEFBUTTON2, MB_DEFBUTTON3.

INT **DialogBoxParam**(hInstance, lpTemplateName, hWndParent, lpDialogProc, dwInitParam)

– створення діалогового вікна;

BOOL CALLBACK **DialogProc** (hWnd, uMsg, wParam, lParam)

– віконна функція діалогового вікна (назва може бути й іншою).

Містить опис поведінки діалогового вікна.

HWND	GetDlgItem (hDlg, nIDDlgItem) – отримання дескриптора віконця елемента діалогу;
HICON	LoadIcon (hInstance, lpIconName) – отримання дескриптора іконки із завантаженого в пам'ять екземпляру програми;
LRESULT	SendMessage (hWnd, Msg, wParam, lParam) – посилення повідомлення вікну (напр., діалоговому вікну);
INT	GetWindowText (hWnd, lpString, nMaxCount) – зчитати текст з вікна.
BOOL	SetWindowText (hWnd, lpString) – вивести текст у вікно.

6. Завдання

1. Написати на основі діалогового вікна програму, яка знаходить значення виразу

1-й варіант: $x + y$;

2-й варіант: $x - y$;

3-й варіант: $x * y$;

4-й варіант: $x \div y$ (якщо $y \neq 0$); (якщо $y = 0$, то видати `MessageBox(, , MB_OK | MB_ICONSTOP)`);

5-й варіант: остачу від ділення x на y (якщо $y \neq 0$) (якщо $y = 0$, то так само, як в 4-му варіанті).

Аргументи x та y вводяться з Edit-елементів керування. Результат – в доступному тільки для читання Edit-віконці.

2. Дописати програму так, щоб при закритті вікна виводилось повідомлення з текстом "Вже покидаєте, навіть чаю не випивши?"

3. Додати кнопку About, яка викликає MessageBox-повідомлення з інформацією про автора.

Під інформацією мається на увазі:

- Прізвище, ім'я та по-батькові
- Дата народження.
- Посада.
- Список рекомендацій.
- Детальна біографія.
- Детальна біографія всіх родичів до 7-го покоління (щоб скучно не було).

7. Приклад проекту

Проект складається з трьох файлів:

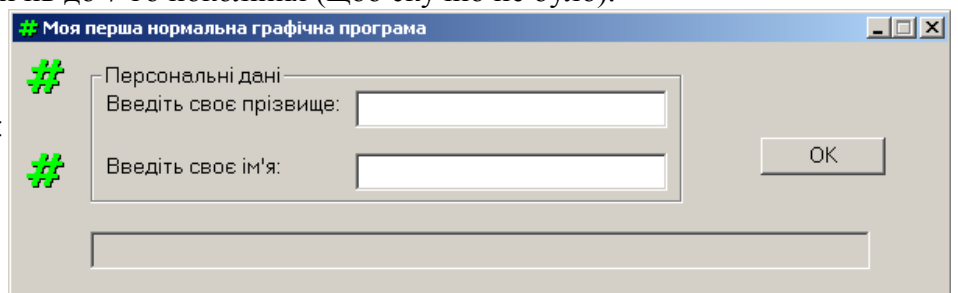
Sharp.ico
.....

Res.rc

```
#include "C:\masm32\include\resource.h"

500 ICON "Sharp.ICO"

700 DIALOG 100, 100, 300, 80
STYLE WS_SYSMENU | WS_MINIMIZEBOX
CAPTION "Моя перша нормальна графічна програма"
FONT 10, "MS Sans Serif"
BEGIN
    LTEXT "Введіть своє прізвище:", 101, 30, 15, 100, 12
    LTEXT "Введіть своє ім'я:", 102, 30, 35, 100, 12
    EDITTEXT 201, 110, 15, 100, 12
    EDITTEXT 202, 110, 35, 100, 12
    EDITTEXT 203, 25, 60, 250, 12, ES_READONLY
    PUSHBUTTON "OK", 301, 240, 30, 40, 12
    ICON 500, 401, 2, 2, 36, 36
    ICON 500, 402, 2, 32, 36, 36
    GROUPBOX "Персональні дані", 501, 25, 6, 190, 45
END
```



Example_01.asm	
<pre> . . . return macro arg mov eax, arg ret endm let macro x, func: vararg invoke func mov x, eax endm szText macro Name, Text:VARARG jmp @F Name db Text,0 @@: endm </pre>	<pre> ; (стандартна шапка) ;===== Означення макросів ===== </pre>
<pre> WinMain PROTO :DWORD,:DWORD,:DWORD,:DWORD DialogProc PROTO :DWORD,:DWORD,:DWORD,:DWORD </pre>	<pre> ;===== Прототипи функцій ===== </pre>
<pre> .data hInstance dd 0 CommandLine dd 0 hEdit1 dd 0 hEdit2 dd 0 hEdit3 dd 0 szName1 db 50 dup (0) szName2 db 50 dup (0) szBuffer db 256 dup (0) </pre>	<pre> ;===== опис сегменту даних ===== </pre>
<pre> .code start: let hInstance, GetModuleHandle, NULL let CommandLine, GetCommandLine invoke WinMain, hInstance, NULL, CommandLine, SW_SHOWDEFAULT invoke ExitProcess, eax WinMain proc hInst:DWORD, hPrew:DWORD, CmdLine:DWORD, CmdShow:DWORD invoke DialogBoxParam, hInst, 700, 0, addr DialogProc, 0 return 0 WinMain endp DialogProc proc hWnd:DWORD, uMsg:DWORD, wParam:DWORD, lParam:DWORD .if uMsg == WM_CLOSE invoke EndDialog, hWnd, 0 .elseif uMsg == WM_INITDIALOG szText szMsg, "Стартує моя перша діалогова програма!!!" szText szMsgTitle, "Повідомлення" invoke MessageBox, hWnd, addr szMsg, addr szMsgTitle, MB_OK+MB_ICONINFORMATION invoke LoadIcon, hInstance, 500 invoke SendMessage, hWnd, WM_SETICON, 1, eax let hEdit1, GetDlgItem, hWnd, 201 let hEdit2, GetDlgItem, hWnd, 202 let hEdit3, GetDlgItem, hWnd, 203 .elseif uMsg == WM_COMMAND .if wParam == 301 .endif .endif return 0 DialogProc endp end start </pre>	<pre> ;===== опис сегменту коду ===== ; 0) ----- реакція на закриття програми ; 1) ----- видача MessageBox-повідомлення ; 2) ----- зміна іконки вікна ; 3) ----- отримання дескрипторів малих віконць ; а) -- зчитування текстів з Edit-віконць ; б) -- видача нового тексту в MessageBox-повідомлення ; в) -- видача нового тексту в Edit-віконць </pre>