

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий  
Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_

Преподаватель \_\_\_\_\_  
(подпись)

« \_\_\_\_ » \_\_\_\_\_ 2022 г.

Отчет  
По Лабораторной работе №6

**«Простое оконное приложение»**

по дисциплине «Машинно-зависимые языки программирования - 4 семестр»

Студент группы ПИ-02 Чередов Р.А.

Преподаватель доцент, к.т.н. Боровцов Е.Г.

Барнаул 2022

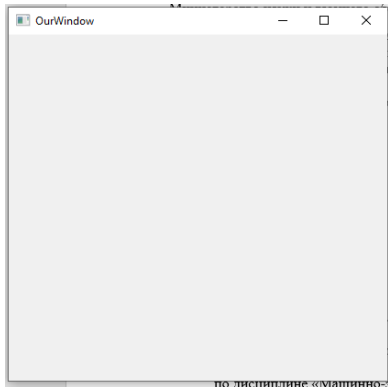
## Задача:

Создать простое оконное приложение в соответствии с индивидуальным заданием, размещенном в личном кабинете

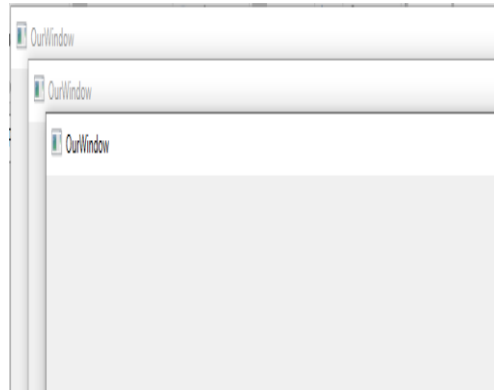
## Индивидуальное задание:

21. Создать три окна, одно из них показать. Двойной щелчок правой кнопкой в неклиентской области окна приводит к выводу на экран остальных окон, если они еще не показывались. Вывод созданных, но не показанных окон разрешен только в случае, если не нажата клавиша Enter.

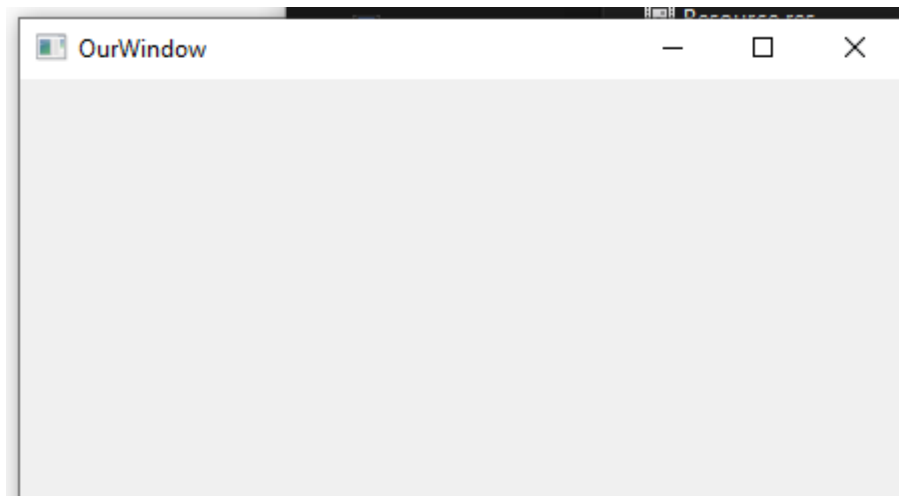
## Результат работы программы:



*Изначальная программа*



*Программа после 2 кликов пкм*



*Программа при нажатии клавиши enter и нажатии правой кнопкой мыши*

## Код файла Assembly1.asm:

```
.386 ; Enable 80386+ instruction set

.model flat, stdcall ; Flat, 32-bit memory model (not used in 64-bit)

option casemap: none ; Case sensitive syntax

; *****

; Our Window procedure prototype

; *****

WndProc proto :DWORD,:DWORD,:DWORD,:DWORD

; *****

; Our main application procedure prototype

; *****

WinMain proto :DWORD,:DWORD,:DWORD,:DWORD

; *****

include \masm32\include\windows.inc

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

include \masm32\include\gdi32.inc

; *****

includelib \masm32\lib\user32.lib

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\gdi32.lib

; *****

; Our initialized data section. Here we declare our strings.

; *****

.data

    ClassName db "OurWindow",0

    WindowTitle db "OurWindow",0

    window_1dd ?

    window_2dd ?

    window_3dd ?

    Flag          db 0

; *****

; Our un-initialized data section. We don't care about the values assigned.

; *****

.data?

hInstance HINSTANCE ?

CommandLine LPSTR ?
```

.code

start:

invoke GetModuleHandle, NULL ; Handle экземпляра

mov hInstance,eax ; приложения

invoke GetCommandLine ; указатель на командную строку

mov CommandLine,eax ;

invoke WinMain, hInstance,NULL,

CommandLine, SW\_SHOWDEFAULT

invoke ExitProcess,eax

WinMain proc hInst:HINSTANCE,hPrevInst:HINSTANCE,CmdLine:LPSTR,CmdShow:DWORD

LOCAL wc:WNDCLASSEX

LOCAL msg:MSG

LOCAL hWin:HWND

mov wc.cbSize,SIZEOF WNDCLASSEX

mov wc.style, CS\_HREDRAW or CS\_VREDRAW or CS\_DBLCLKS

mov wc.lpfnWndProc, OFFSET WndProc

mov wc.cbClsExtra,NULL

mov wc.cbWndExtra,NULL

push hInst

pop wc.hInstance

mov wc.hbrBackground,COLOR\_BTNFACE+1

mov wc.lpszMenuName,NULL

mov wc.lpszClassName,OFFSET ClassName

invoke LoadIcon,NULL,IDI\_APPLICATION

mov wc.hIcon,eax

mov wc.hIconSm,eax

invoke LoadCursor,NULL,IDC\_ARROW

mov wc.hCursor,eax

invoke RegisterClassEx, addr wc

invoke CreateWindowEx,NULL, ADDR ClassName, ADDR WindowTitle, WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT,0, CW\_USEDEFAULT,0, NULL,NULL,hInstance,NULL

mov window\_1,eax

invoke CreateWindowEx,NULL, ADDR ClassName, ADDR WindowTitle, WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT,0, CW\_USEDEFAULT,0, NULL,NULL,hInstance,NULL

mov window\_2,eax

invoke CreateWindowEx,NULL, ADDR ClassName, ADDR WindowTitle, WS\_OVERLAPPEDWINDOW,

```

CW_USEDEFAULT,0, CW_USEDEFAULT,0, NULL,NULL,hInstance,NULL

mov window_3,eax

invoke ShowWindow, window_1,SW_SHOWNORMAL

invoke UpdateWindow, window_1

.WHILE TRUE

    invoke GetMessage, ADDR msg,NULL,0,0

    .BREAK .IF (!eax)

    invoke TranslateMessage, ADDR msg

    invoke DispatchMessage, ADDR msg

.ENDW

mov eax,msg.wParam

ret

WinMain endp

WndProc proc hWnd:DWORD, uMsg:DWORD, wParam:DWORD,
    lParam:DWORD

LOCAL hDC :DWORD

LOCAL Ps:PAINTSTRUCT

.IF uMsg==WM_DESTROY

    invoke PostQuitMessage,NULL

.ELSEIF uMsg==WM_KEYDOWN

    .IF wParam == VK_RETURN

        mov Flag,1

    .ENDIF

.ELSEIF uMsg==WM_KEYUP

    .IF wParam == VK_RETURN

        mov Flag,0

    .ENDIF

.ELSEIF uMsg==WM_NCRBUTTONDBLCLK

    .IF Flag == 0

        invoke ShowWindow, window_1,SW_SHOWNORMAL

        invoke ShowWindow, window_2,SW_SHOWNORMAL

        invoke ShowWindow, window_3,SW_SHOWNORMAL

    .ENDIF

.ELSE

    invoke DefWindowProc,hWnd,uMsg,wParam,lParam

ret

.ENDIF

```

xor eax,eax

ret

WndProc endp

end start

