< 윈도우프로그래밍 9주차 과제 >

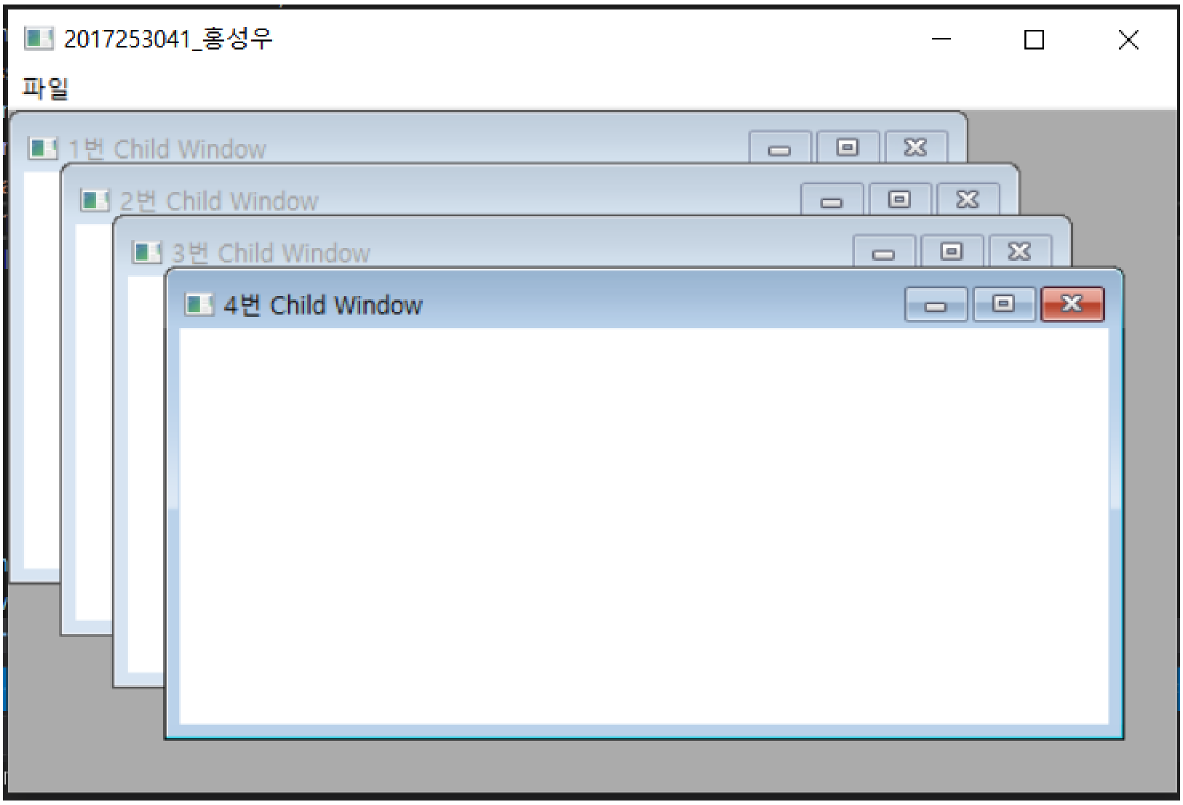
2017253041_홍성우

#1

```
#include <windows.h>
#include <TCHAR.H>
#include "resource.h"
LRESULT CALLBACK FrameWndProc(HWND hwnd, UINT iMsg,
        WPARAM wParam, LPARAM lParam);
LRESULT CALLBACK ChildWndProc(HWND hDlg, UINT iMsg,
        WPARAM wParam, LPARAM lParam);
HINSTANCE hInst;

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
        LPSTR lpszCmdLine, int nCmdShow)
{
        HWND     hwnd;
        MSG      msg;
        WNDCLASS WndClass;
        hInst = hInstance;
        WndClass.style = CS_HREDRAW | CS_VREDRAW;
        WndClass.lpfnWndProc = FrameWndProc;
        WndClass.cbClsExtra = 0;
        WndClass.cbWndExtra = 0;
        WndClass.hInstance = hInstance;
        WndClass.hIcon = LoadIcon(NULL, IDI_APPLICATION);
        WndClass.hCursor = LoadCursor(NULL, IDC_ARROW);
        WndClass.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);
        WndClass.lpszMenuName = MAKEINTRESOURCE(IDR_MENU7);
        WndClass.lpszClassName = _T("Window Class Name");
        RegisterClass(&WndClass);
        WndClass.lpfnWndProc = ChildWndProc;
        WndClass.lpszMenuName = NULL;
        WndClass.lpszClassName = _T("Child Window Class Name");
        RegisterClass(&WndClass);
        hwnd = CreateWindow(
                _T("Window Class Name"),
                _T("2017253041_홍성우"),
                WS_OVERLAPPEDWINDOW,
                400, 300,
                600, 400,
                NULL,
                NULL,
                hInstance,
                NULL
        );
        ShowWindow(hwnd, nCmdShow);
        UpdateWindow(hwnd);
        while (GetMessage(&msg, NULL, 0, 0))
        {
                TranslateMessage(&msg);
                DispatchMessage(&msg);
```

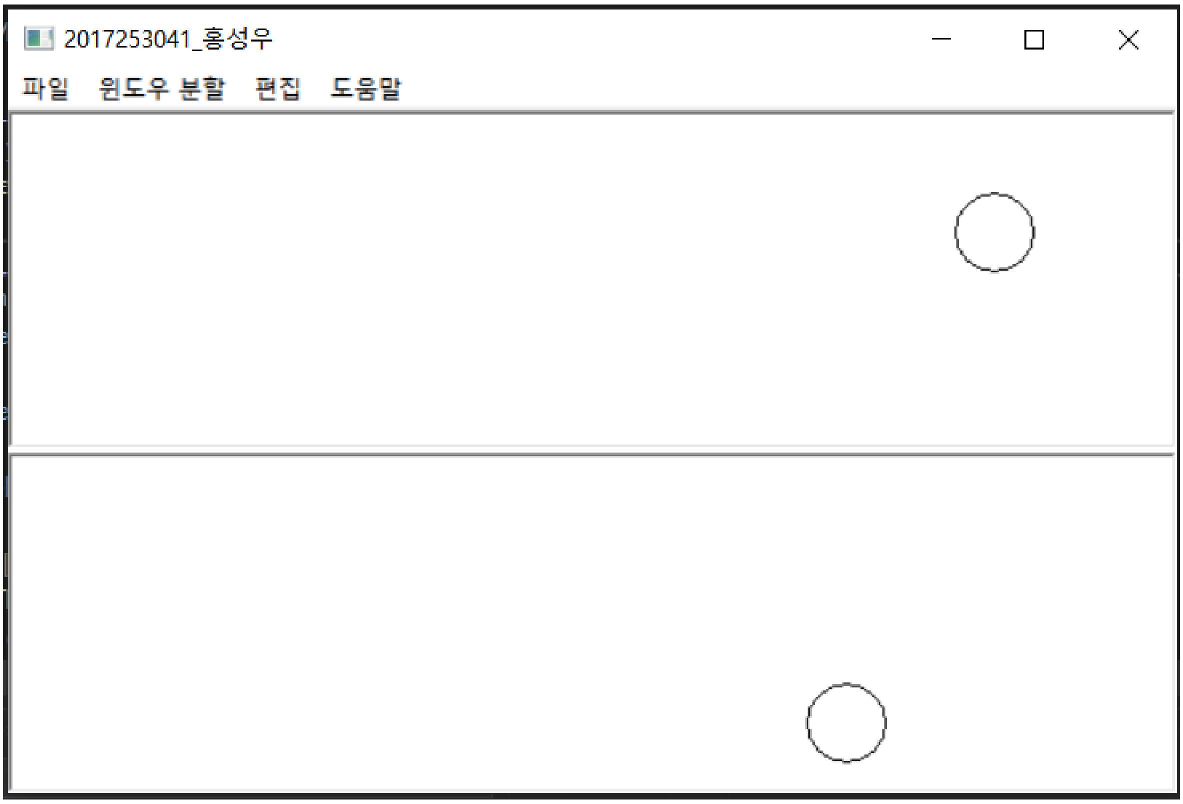
```
	}
	return (int)msg.wParam;
}

LRESULT CALLBACK FrameWndProc(HWND hwnd, UINT iMsg,
	WPARAM wParam, LPARAM lParam)
{
	static HWND hwndClient;
	CLIENTCREATESTRUCT clientcreate;
	MDICREATESTRUCT mdicreate;
	HWND hwndChild;

	TCHAR str[30];
	static int childnum = 1;

	switch (iMsg)
	{
	case WM_CREATE:
		clientcreate.hWindowMenu = GetSubMenu(GetMenu(hwnd), 0);
		clientcreate.idFirstChild = 100;
		hwndClient = CreateWindow(
			_T("MDICLIENT"), NULL,
			WS_CHILD | WS_CLIPCHILDREN | WS_VISIBLE,
			0, 0, 0, 0,
			hwnd,
			NULL,
			hInst,
			(LPSTR)&clientcreate
		);
		ShowWindow(hwndClient, SW_SHOW);
		return 0;
	case WM_COMMAND:
		switch (LOWORD(wParam))
		{
		case ID_FILENEW:
			mdicreate.szClass = _T("Child Window Class Name");
			_stprintf_s(str, 30, _T("%d번 Child Window"), childnum);
			childnum++;
			mdicreate.szTitle = str;
			mdicreate.hOwner = hInst;
			mdicreate.x = CW_USEDEFAULT;
			mdicreate.y = CW_USEDEFAULT;
			mdicreate.cx = CW_USEDEFAULT;
			mdicreate.cy = CW_USEDEFAULT;
			mdicreate.style = 0;
			mdicreate.lParam = 0;
			hwndChild = (HWND)SendMessage(hwndClient,
				WM_MDICREATE, 0, (LPARAM)(LPMDICREATESTRUCT)&mdicreate);
			return 0;
		}
		break;
	case WM_DESTROY:
		PostQuitMessage(0);
		return 0;
	}
```

```
        return DefFrameProc(hwnd, hwndClient, iMsg, wParam, lParam);
}

LRESULT CALLBACK ChildWndProc(HWND hwnd, UINT iMsg,
        WPARAM wParam, LPARAM lParam)
{
        switch (iMsg)
        {
        case WM_CREATE:
                break;
        case WM_DESTROY:
                return 0;
        }
        return DefMDIChildProc(hwnd, iMsg, wParam, lParam);
}
```



#2

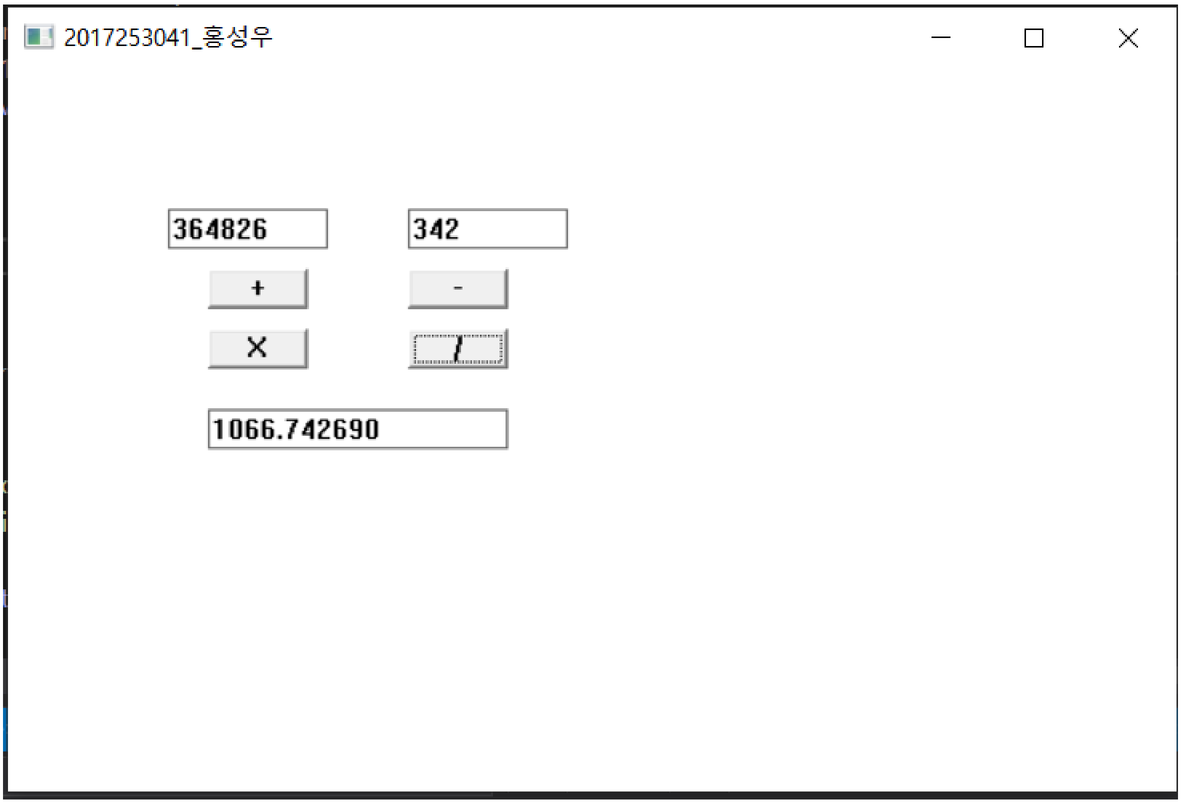
```
#include <Windows.h>
#include<tchar.h>
#include"resource.h"
LRESULT CALLBACK FrameWndProc(HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam);
LRESULT CALLBACK ChildWndProc(HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam);
HINSTANCE hInst;

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpszCmdLine, int
nCmdShow)
{
    HWND hwnd;
    MSG msg;
    WNDCLASS WndClass;

    hInst = hInstance;
    WndClass.style = CS_HREDRAW | CS_VREDRAW;
    WndClass.lpfnWndProc = FrameWndProc;
    WndClass.cbClsExtra = 0;
    WndClass.cbWndExtra = 0;
    WndClass.hInstance = hInstance;
    WndClass.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    WndClass.hCursor = LoadCursor(NULL, IDC_ARROW);
    WndClass.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);
    WndClass.lpszMenuName = MAKEINTRESOURCE(IDR_MENU7);
    WndClass.lpszClassName = _T("Window Class Name");
    RegisterClass(&WndClass);

    WndClass.lpszMenuName = NULL;
    WndClass.lpfnWndProc = ChildWndProc;
    WndClass.lpszClassName = _T("Child Window Class Name");
    RegisterClass(&WndClass);

    hwnd = CreateWindow(
        _T("Window Class Name"),
        _T("2017253041_홍성우"),
        WS_OVERLAPPEDWINDOW,
        400, 300,
        600, 400,
        NULL,
        NULL,
        hInstance,
        NULL);

    ShowWindow(hwnd, nCmdShow);
    UpdateWindow(hwnd);

    while (GetMessage(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return (int)msg.wParam;
}

HWND hwndChild[2];
LRESULT CALLBACK FrameWndProc(HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam)
{
    HDC hdc;
    PAINTSTRUCT ps;
    RECT rect;

    switch (iMsg)
    {
    case WM_CREATE:
        break;
    case WM_COMMAND:
        switch (LOWORD(wParam))
        {
        case ID_SPLIT_2X1:
            GetClientRect(hwnd, &rect);
            hwndChild[0] = CreateWindowEx
            (WS_EX_CLIENTEDGE,
                _T("Child Window Class Name"),
                NULL,
                WS_CHILD | WS_VISIBLE,
                0,
                0,
                rect.right,
                rect.bottom / 2 - 1,
                hwnd,
                NULL,
                hInst,
                NULL
            );
            hwndChild[1] = CreateWindowEx
            (WS_EX_CLIENTEDGE,
                _T("Child Window Class Name"),
                NULL,
                WS_CHILD | WS_VISIBLE,
                0,
                rect.bottom / 2 + 1,
                rect.right,
                rect.bottom / 2,
                hwnd,
                NULL,
                hInst,
                NULL
            );
            break;
        }
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
    }
    return DefWindowProc(hwnd, iMsg, wParam, lParam);
}
LRESULT CALLBACK ChildWndProc(HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam)
{
    static RECT rect;
    static HDC hdc;
    PAINTSTRUCT ps;

    static bool dirX[2] = { true,true }, dirY[2] = { true,true };
    static int x[2], y[2];
    static int selection;
    static bool flag[2] = { false,false };

    switch (iMsg)
    {
    case WM_CREATE:
        x[0] = y[0] = x[1] = y[1] = 50;
        GetClientRect(hwnd, &rect);
        break;
    case WM_LBUTTONDOWN:
        if (hwnd == hwndChild[0])
            selection = 0;
        else
            selection = 1;

        flag[selection] = flag[selection] ? false : true;

        if (flag[selection])
            SetTimer(hwndChild[selection], selection, 10, NULL);
        else if (!flag[selection])
            KillTimer(hwndChild[selection], selection);
        break;
    case WM_TIMER:
        if (x[wParam]<20 || x[wParam]>rect.right - 20) dirX[wParam] = dirX[wParam] ? false :
true;
        if (y[wParam]<20 || y[wParam]>rect.bottom - 20) dirY[wParam] = dirY[wParam] ? false :
true;

        if (dirX[wParam])
            x[wParam]++;
        else
            x[wParam]--;
        if (dirY[wParam])
            y[wParam]++;
        else
            y[wParam]--;
        InvalidateRgn(hwndChild[wParam], NULL, TRUE);
        break;
    case WM_PAINT:
        hdc = BeginPaint(hwndChild[0], &ps);
        Ellipse(hdc, x[0] - 20, y[0] - 20, x[0] + 20, y[0] + 20);
        EndPaint(hwndChild[0], &ps);

        hdc = BeginPaint(hwndChild[1], &ps);
        Ellipse(hdc, x[1] - 20, y[1] - 20, x[1] + 20, y[1] + 20);
        EndPaint(hwndChild[1], &ps);
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
    }

    return DefMDIChildProc(hwnd, iMsg, wParam, lParam);
}
```



#3

```
#include <Windows.h>
#include<tchar.h>
#include<string.h>
#include<math.h>
LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam);
HINSTANCE hInst;

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpszCmdLine, int
nCmdShow)
{
    HWND hwnd;
    MSG msg;
    WNDCLASS WndClass;

    hInst = hInstance;
    WndClass.style = CS_HREDRAW | CS_VREDRAW;
    WndClass.lpfnWndProc = WndProc;
    WndClass.cbClsExtra = 0;
    WndClass.cbWndExtra = 0;
    WndClass.hInstance = hInstance;
    WndClass.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    WndClass.hCursor = LoadCursor(NULL, IDC_ARROW);
    WndClass.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);
    WndClass.lpszMenuName = NULL;
    WndClass.lpszClassName = _T("Window Class Name");

    RegisterClass(&WndClass);
    hwnd = CreateWindow(
        _T("Window Class Name"),
        _T("2017253041_홍성우"),
        WS_OVERLAPPEDWINDOW,
        400,
        300,
        600,
        400,
        NULL,
        NULL,
        hInstance,
        NULL);

    ShowWindow(hwnd, nCmdShow);
    UpdateWindow(hwnd);

    while (GetMessage(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return (int)msg.wParam;
}

#define IDC_EDIT1 100
#define IDC_EDIT2 101
#define IDC_EDIT3 102
#define IDC_BUTTON1 103
#define IDC_BUTTON2 104
#define IDC_BUTTON3 105
#define IDC_BUTTON4 106
LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam)
{
    HDC hdc;
    static HWND hEdit1, hEdit2, hEdit3, hButton1, hButton2, hButton3, hButton4;
    double num1, num2, result;
    TCHAR str[20];
    switch (iMsg)
    {
    case WM_CREATE:
        hEdit1 = (HWND)CreateWindow(_T("edit"), NULL, WS_CHILD | WS_VISIBLE | WS_BORDER, 80,
70, 80, 20, hwnd, (HMENU)IDC_EDIT1, hInst, NULL);
        hEdit2 = (HWND)CreateWindow(_T("edit"), NULL, WS_CHILD | WS_VISIBLE | WS_BORDER, 200,
70, 80, 20, hwnd, (HMENU)IDC_EDIT2, hInst, NULL);
        hEdit3 = (HWND)CreateWindow(_T("edit"), NULL, WS_CHILD | WS_VISIBLE | WS_BORDER, 100,
170, 150, 20, hwnd, (HMENU)IDC_EDIT3, hInst, NULL);
        hButton1 = (HWND)CreateWindow(_T("button"), _T("+"), WS_CHILD | WS_VISIBLE |
BS_PUSHBUTTON, 100, 100, 50, 20, hwnd, (HMENU)IDC_BUTTON1, hInst, NULL);
        hButton2 = (HWND)CreateWindow(_T("button"), _T("-"), WS_CHILD | WS_VISIBLE |
BS_PUSHBUTTON, 200, 100, 50, 20, hwnd, (HMENU)IDC_BUTTON2, hInst, NULL);
        hButton3 = (HWND)CreateWindow(_T("button"), _T("X"), WS_CHILD | WS_VISIBLE |
BS_PUSHBUTTON, 100, 130, 50, 20, hwnd, (HMENU)IDC_BUTTON3, hInst, NULL);
        hButton4 = (HWND)CreateWindow(_T("button"), _T("/"), WS_CHILD | WS_VISIBLE |
BS_PUSHBUTTON, 200, 130, 50, 20, hwnd, (HMENU)IDC_BUTTON4, hInst, NULL);
```

```
        break;
    case WM_COMMAND:
        switch (LOWORD(wParam))
        {
        case IDC_BUTTON1:
            num1 = GetDlgItemInt(hwnd, IDC_EDIT1, NULL, TRUE);
            num2 = GetDlgItemInt(hwnd, IDC_EDIT2, NULL, TRUE);
            result = num1 + num2;
            SetDlgItemInt(hwnd, IDC_EDIT3, result, TRUE);
            break;
        case IDC_BUTTON2:
            num1 = GetDlgItemInt(hwnd, IDC_EDIT1, NULL, TRUE);
            num2 = GetDlgItemInt(hwnd, IDC_EDIT2, NULL, TRUE);
            result = num1 - num2;
            SetDlgItemInt(hwnd, IDC_EDIT3, result, TRUE);
            break;
        case IDC_BUTTON3:
            num1 = GetDlgItemInt(hwnd, IDC_EDIT1, NULL, TRUE);
            num2 = GetDlgItemInt(hwnd, IDC_EDIT2, NULL, TRUE);
            result = num1 * num2;
            SetDlgItemInt(hwnd, IDC_EDIT3, result, TRUE);
            break;
        case IDC_BUTTON4:
            num1 = GetDlgItemInt(hwnd, IDC_EDIT1, NULL, TRUE);
            num2 = GetDlgItemInt(hwnd, IDC_EDIT2, NULL, TRUE);
            result = num1 / num2;
            _stprintf_s(str, 20, _T("%lf"), result);
            SetDlgItemText(hwnd, IDC_EDIT3, str);
            break;
        }
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
        break;

    }
    return DefWindowProc(hwnd, iMsg, wParam, lParam);
}
```



2017253041_홍성우

364826

342

1066.742690