

< 윈도우프로그래밍 3주차 과제 >

2017253041_홍성우

#1

```
#include <windows.h>
#include <TCHAR.H>
LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg,
    WPARAM wParam, LPARAM lParam);
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpszCmdLine, int nCmdShow)
{
    HWND    hwnd;
    MSG      msg;
    WNDCLASS WndClass;
    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("홍성우_2017253041"),
        WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        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 WndProc(HWND hwnd, UINT iMsg,
    WPARAM wParam, LPARAM lParam)
{
    HDC hdc;
    PAINTSTRUCT ps;
```

```

RECT rect;

switch (iMsg)
{
case WM_CREATE:
    break;
case WM_PAINT:
    hdc = BeginPaint(hwnd, &ps);
    rect.left = 0;
    rect.top = 0;
    rect.right = 200;
    rect.bottom = 200;
    DrawText(hdc, _T("i love you"), _tcslen(_T("i love you")), &rect,
        DT_SINGLELINE | DT_CENTER | DT_VCENTER);
    EndPaint(hwnd, &ps);
    break;
case WM_DESTROY:
    PostQuitMessage(0);
    break;
}
return DefWindowProc(hwnd, iMsg, wParam, lParam);
}

```



#2

```

#include <windows.h>
#include <TCHAR.H>
LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg,
    WPARAM wParam, LPARAM lParam);
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpszCmdLine, int nCmdShow)
{
    HWND    hwnd;
    MSG     msg;
    WNDCLASS WndClass;

```

```

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("홍성우_2017253041"),
    WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    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 WndProc(HWND hwnd, UINT iMsg,
    WPARAM wParam, LPARAM lParam)
{
    HDC hdc;
    PAINTSTRUCT ps;
    static TCHAR str[10][100];
    static int count, line, i;

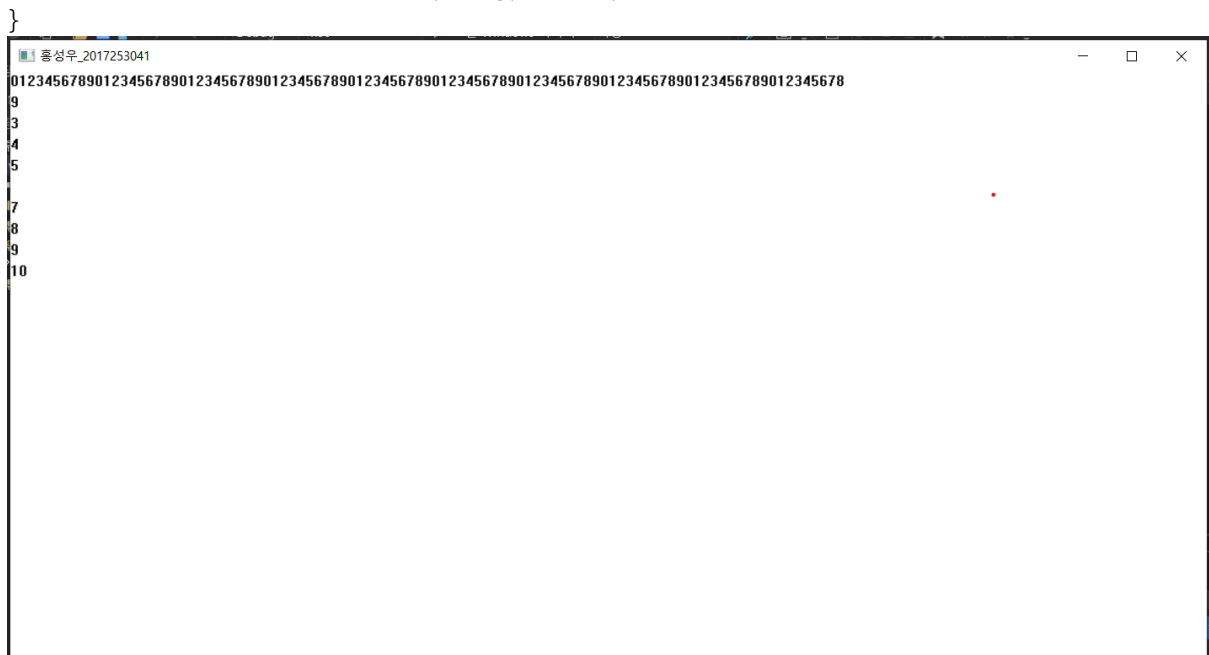
    switch (iMsg)
    {
    case WM_CREATE:
        count=0;
        line = 0;
        break;
    case WM_PAINT:
        hdc = BeginPaint(hwnd, &ps);
        for (i = 0; i < 10; i++) {
            TextOut(hdc, 0, i*20, str[i], _tcslen(str[i]));
        }
        EndPaint(hwnd, &ps);
        break;
    case WM_CHAR:

```

```

        if (wParam == VK_BACK) {
            if (count > 0) {
                count--;
            }
            else if (count == 0) {
                if (line > 0) {
                    line--;
                    count = _tcslen(str[line]);
                }
            }
        }
        else if (wParam == VK_RETURN) {
            if (line < 9) {
                line++;
                count = 0;
            }
        }
        else{
            if (count >= 99) {
                line++;
                count = 0;
            }
            str[line][count] = wParam;
            count++;
        }
        str[line][count] = NULL;
        InvalidateRgn(hwnd, NULL, TRUE);
        break;
case WM_DESTROY:
    PostQuitMessage(0);
    break;
}
return DefWindowProc(hwnd, iMsg, wParam, lParam);
}

```



```

#include <windows.h>
#include <TCHAR.H>
LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg,
    WPARAM wParam, LPARAM lParam);
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpszCmdLine, int nCmdShow)
{
    HWND    hwnd;
    MSG      msg;
    WNDCLASS WndClass;
    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("홍성우_2017253041"),
        WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        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 WndProc(HWND hwnd, UINT iMsg,
    WPARAM wParam, LPARAM lParam)
{
    HDC hdc;
    PAINTSTRUCT ps;
    HPEN hPen, oldPen;
    HBRUSH hBrush, oldBrush;

    switch (iMsg)
    {
    case WM_CREATE:
        break;

```

```

    case WM_PAINT:
        hdc = BeginPaint(hwnd, &ps);
        hPen = CreatePen(PS_DASH, 1, RGB(255, 255, 0));
        oldPen = (HPEN)SelectObject(hdc, hPen);
        hBrush = CreateSolidBrush(RGB(255, 0, 255));
        oldBrush = (HBRUSH)SelectObject(hdc, hBrush);
        Ellipse(hdc, 180, 100, 220, 140);
        SelectObject(hdc, oldPen);
        DeleteObject(hPen);
        SelectObject(hdc, oldBrush);
        DeleteObject(hBrush);
        EndPaint(hwnd, &ps);
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
}
return DefWindowProc(hwnd, iMsg, wParam, lParam);
}

```



#4

```

#include <windows.h>
#include <TCHAR.H>
LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg,
    WPARAM wParam, LPARAM lParam);
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpszCmdLine, int nCmdShow)
{
    HWND    hwnd;
    MSG      msg;
    WNDCLASS WndClass;
    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("홍성우_2017253041"),
    WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    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 WndProc(HWND hwnd, UINT iMsg,
    WPARAM wParam, LPARAM lParam)
{
    HDC hdc;
    PAINTSTRUCT ps;
    static int i;
    HBRUSH hBrush, oldBrush;

    RECT rectTop;
    RECT rectLeft;
    RECT rectRight;
    RECT rectBottom;

    switch (iMsg)
    {
    case WM_CREATE:
        i = 0;
        break;
    case WM_PAINT:
        hdc = BeginPaint(hwnd, &ps);
        rectTop.left = 100;
        rectTop.top = 0;
        rectTop.right = 150;
        rectTop.bottom = 100;
        Rectangle(hdc, 100, 0, 150, 100);
        DrawText(hdc, _T("위쪽"), 2, &rectTop, DT_SINGLELINE | DT_CENTER |

```

```

DT_VCENTER);
    rectLeft.left = 50;
    rectLeft.top = 100;
    rectLeft.right = 100;
    rectLeft.bottom = 200;
    Rectangle(hdc, 50, 100, 100, 200);
    DrawText(hdc, _T("왼쪽"), 2, &rectLeft, DT_SINGLELINE | DT_CENTER |
DT_VCENTER);
    rectRight.left = 150;
    rectRight.top = 100;
    rectRight.right = 200;
    rectRight.bottom = 200;
    Rectangle(hdc, 150, 100, 200, 200);
    DrawText(hdc, _T("오른쪽"), 3, &rectRight, DT_SINGLELINE | DT_CENTER |
DT_VCENTER);
    rectBottom.left = 100;
    rectBottom.top = 200;
    rectBottom.right = 150;
    rectBottom.bottom = 300;
    Rectangle(hdc, 100, 200, 150, 300);
    DrawText(hdc, _T("아래쪽"), 3, &rectBottom, DT_SINGLELINE | DT_CENTER |
DT_VCENTER);
    hBrush = CreateSolidBrush(RGB(255, 0, 0));
    oldBrush = (HBRUSH)SelectObject(hdc, hBrush);
    switch (i) {
    case 1:
        Rectangle(hdc, 100, 0, 150, 100);
        break;
    case 2:
        Rectangle(hdc, 50, 100, 100, 200);
        break;
    case 3:
        Rectangle(hdc, 150, 100, 200, 200);
        break;
    case 4:
        Rectangle(hdc, 100, 200, 150, 300);
        break;
    default:
        break;
    }
    SelectObject(hdc, oldBrush);
    DeleteObject(hBrush);
    EndPaint(hwnd, &ps);
    break;
case WM_KEYDOWN:
    if (wParam == VK_UP) i = 1;
    else if (wParam == VK_LEFT) i = 2;
    else if (wParam == VK_RIGHT) i = 3;
    else if (wParam == VK_DOWN) i = 4;
    InvalidateRgn(hwnd, NULL, TRUE);
    break;
case WM_KEYUP:
    i = 0;
    InvalidateRgn(hwnd, NULL, TRUE);
    break;

```



```

        case WM_DESTROY:
            PostQuitMessage(0);
            break;
    }
    return DefWindowProc(hwnd, iMsg, wParam, lParam);
}

```



#5

```

#include <windows.h>
#include <TCHAR.H>
LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg,
    WPARAM wParam, LPARAM lParam);
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpszCmdLine, int nCmdShow)
{
    HWND    hwnd;
    MSG      msg;
    WNDCLASS WndClass;
    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("홍성우_2017253041"),
        WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT,
        CW_USEDEFAULT,

```

```

        CW_USEDEFAULT,
        CW_USEDEFAULT,
        NULL,
        NULL,
        hInstance,
        NULL
    );
    ShowWindow(hwnd, nCmdShow);
    UpdateWindow(hwnd);
    while (GetMessage(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return (int)msg.wParam;
}

#define LEFT 0
#define TOP 500
#define RIGHT 300
#define BOTTOM TOP+20
#define LINE 1
#define ELLIPSE 2
#define RECTANGLE 3
int ObjectKind;
int x1, y1, x2, y2;
void Parsing(LPTSTR str)
{
    TCHAR cmd[20];
    int i;
    i = 0;
    while (*str != ' ' && *str != '(')
        cmd[i++] = *str++;
    cmd[i] = NULL;
    _stscanf_s(str, _T("(%d, %d, %d, %d)", &x1, &y1, &x2, &y2);
    if (!_tcscmp(cmd, _T("Line")))
        ObjectKind = LINE;
    else if (!_tcscmp(cmd, _T("Ellipse")))
        ObjectKind = ELLIPSE;
    else if (!_tcscmp(cmd, _T("Rectangle")))
        ObjectKind = RECTANGLE;
}

LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg,
    WPARAM wParam, LPARAM lParam)
{
    HDC hdc;
    PAINTSTRUCT ps;
    static TCHAR str[100];
    static int count;
    static SIZE size;

    switch (iMsg)
    {
        case WM_CREATE:

```

```

        CreateCaret(hwnd, NULL, 5, 15);
        ShowCaret(hwnd);
        count = 0;
        break;
    case WM_PAINT:
        hdc = BeginPaint(hwnd, &ps);
        Rectangle(hdc, 0, 500, 300, 520);
        GetTextExtentPoint(hdc, str, _tcslen(str), &size);
        TextOut(hdc, 0, 500, str, _tcslen(str));
        switch (ObjectKind) {
            case LINE:
                MoveToEx(hdc, x1, y1, NULL);
                LineTo(hdc, x2, y2);
                break;
            case ELLIPSE:
                Ellipse(hdc, x1, y1, x2, y2);
                break;
            case RECTANGLE:
                Rectangle(hdc, x1, y1, x2, y2);
                break;
        }
        SetCaretPos(size.cx, 500);
        EndPaint(hwnd, &ps);
        break;
    case WM_CHAR:
        str[count++] = wParam;
        str[count] = NULL;
        if (wParam == VK_RETURN) {
            Parsing(str);
        }
        InvalidateRgn(hwnd, NULL, TRUE);
        break;
    case WM_DESTROY:
        HideCaret(hwnd);
        DestroyCaret();
        PostQuitMessage(0);
        break;
    }
    return DefWindowProc(hwnd, iMsg, wParam, lParam);
}

```

