



Window API Programming- 입문 1

afewhee@gmail.com





- 1. 윈도우 생성
- 2. 클래스 전환
- 3. 문자열 출력
- 4. 실습





- 간단한 윈도우 프로그래밍

//팝업창 띄우기

```
#include <windows.h>
```

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    MessageBox(NULL, “My First Message”,  
    “Title”, MB_OK);
```

```
}
```





- 필수 라이브러리:

- ◆ user32.lib: 윈도우 메시지, 기본 함수 등에 대한 라이브러리
- ◆ gdi32.lib: 그래픽 관련 라이브러리
- ◆ kernel32.lib: 프로세스 관련 라이브러리

- 필수 함수:

- ◆ WinMain(HINSTANCE, HINSTANCE, LPSTR, int):
프로그램 진입함수. Console의 main() 함수와 동일한 기능
- ◆ GetModuleHandle(); 프로그램의 Instance을 얻는 함수
- ◆ RegisterClass(); 사용자가 설정한 값을 WNDCLASS 구조체 변수로 설정하고 등록
- ◆ CreateWindow(); 윈도우 생성 함수
- ◆ PostQuitMessage(); 윈도우 종료 함수
- ◆ MessagePocedure(); 사용자가 정의한 메시지 처리 함수. WNDCLASS 구조체 변수의 함수 포인터 값에 이 함수를 연결함





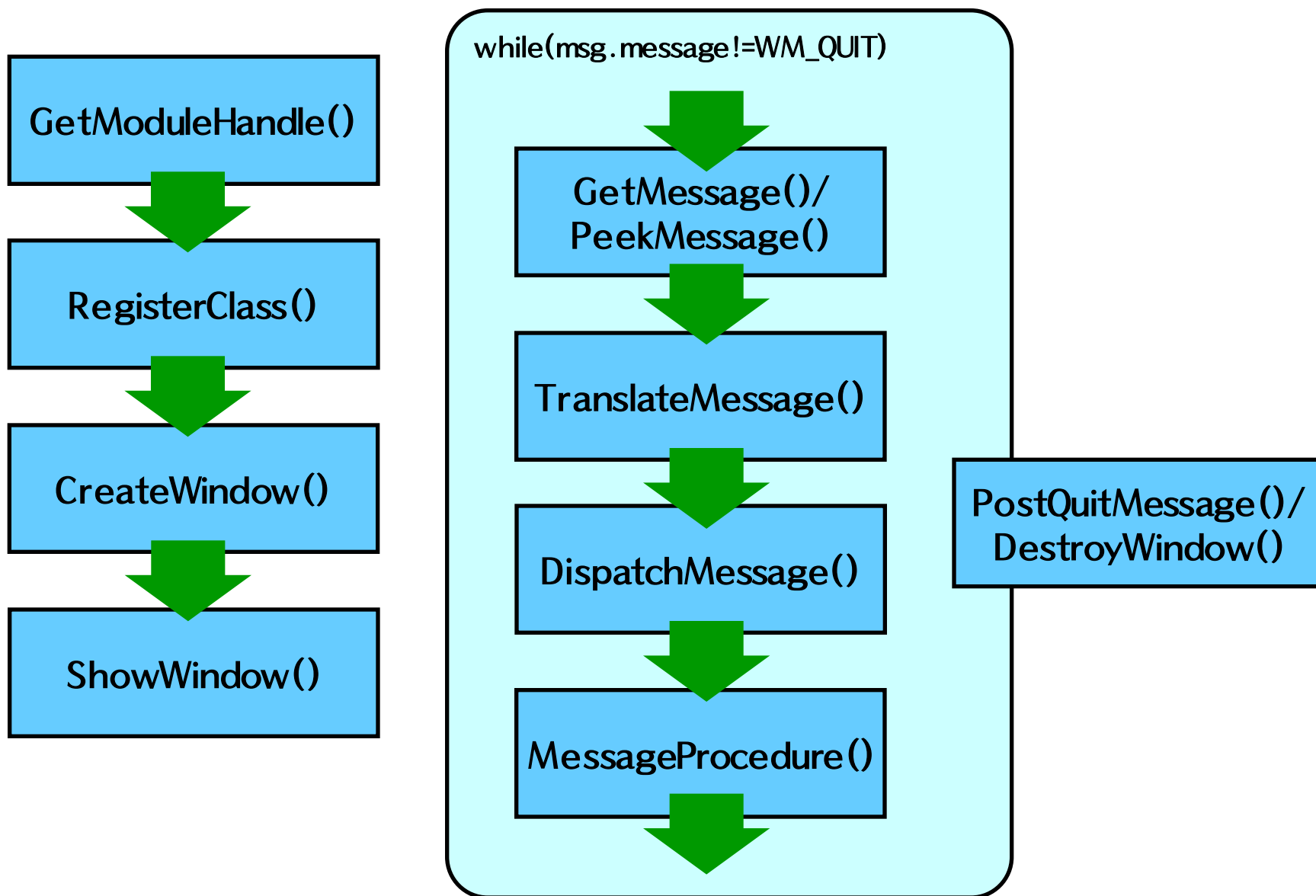
● 필수 함수:

- ◆ GetMessage(), PeekMessage(): 운영 체제의 메시지 큐에서 해당 프로그램의 이벤트를 가져옴
- ◆ TranslateMessage(): 가상 키(Virtual Key) 메시지를 문자형 메시지로 변환해줌
- ◆ RegisterClass(): 사용자가 설정한 값을 WNDCLASS 구조체 변수로 설정하고 등록
- ◆ CreateWindow(): 윈도우 생성 함수
- ◆ ShowWindow(): 윈도우의 show/hide 설정
- ◆ PostQuitMessage(): 윈도우 종료 함수
- ◆ Message Pocedure(): 사용자가 정의한 메시지 처리 함수. WNDCLASS 구조체 변수의 함수 포인터 값에 이 함수를 연결함
- ◆ SendMessage(): 해당 윈도우에 메시지 전송





1. 윈도우 생성





- CApplication:
 - ◆ 윈도우 생성/ 소멸에 관한 클래스
- 주요 멤버 함수
 - ◆ Create(): 윈도우 생성 함수
 - ◆ Run(): 실행 함수
 - ◆ Cleanup(): 프로그램 종료 시 자원을 반납하는 함수
 - ◆ MsgProc(): 메시지 처리 함수
- static 변수, 함수
 - ◆ WINAPI 형식의 함수를 클래스에서 멤버로 갖기 위해서 static 으로 지정
 - ◆ static 변수: static 함수를 호출하기 위한 인스턴스
➔ CApplication 인스턴스와 동일





2. 클래스 전환 - 예

```
class CApplication
{
public:
    char                m_sCls[128]    ;
    HINSTANCE           m_hInst        ;
    HWND               m_hWnd          ;
    DWORD              m_dWinStyle     ;
    DWORD              m_dScnX         ;           // Screen Width
    DWORD              m_dScnY         ;           // Screen Height

    bool                m_bShowCursor ;           // Show Cursor

public:
    CApplication();
    virtual ~CApplication();

    //Window
    INT                Create(HINSTANCE hInst);
    INT                Run();
    void               Cleanup();

public:
    // WndProc에서필요한static 변수
    static CApplication* g_pApp;
    static CApplication* GetAppInstance();
    static LRESULT WINAPI WndProc(HWND, UINT, WPARAM, LPARAM);

    // 메시지처리함수
    virtual LRESULT MsgProc(HWND, UINT, WPARAM, LPARAM);
};
```





- GDI (Graphic Device Interface): 윈도우는 문자열을 포함한 모든 출력을 GDI를 이용
- GDI를 이용한 문자열 출력
 - ◆ 윈도우로 부터 DC를 가져옴
 - BeginPaint()/EndPaint()
 - GetDC()/ReleaseDC()
 - ◆ 문자열 출력
 - TextOut(): 단일 문자열 출력. 문자열 바꿈이 허용 안됨
 - DrawText(): 여러 문자열 출력. 문자열 바꿈이 허용
 - ◆ 문자열 출력 관련 함수
 - SetTextColor(): 문자열 색상 변경
 - SetTextAlign(): 문자열 Align 변경
 - SetBkMode(): OPAQUE, TRANSPARENT





3. 문자열 출력 - 예

```
// WM_PAINT 에서 처리 예
LRESULT MsgProc( HWND hWnd, UINT msg, WPARAM wParam, LPARAM lParam )
...
case WM_PAINT:
{
    char str[] = "출력 문자열 ";

    HDC hDC;
    PAINTSTRUCT ps;

    hDC = BeginPaint(hWnd, &ps);

        TextOut(hDC, 100, 50, str, strlen(str));
    EndPaint(hWnd, &ps);
    break;
}

// WM_PAINT 이외에서 처리 예 ( GetDC() / ReleaseDC() )

...

HDC hDC = GetDC(hWnd);
RECT rt={100, 100, 400, 300};

char str[] = "출력 문자열 ";
TextOut(hDC, 100, 50, str, strlen(str));
ReleaseDC(hWnd, hDC);
```





- C 형식으로 윈도우 생성과 문자열 출력을 연습해 본다.
- 위의 C 형식으로 작성한 프로그램을 C++로 전환해 본다.
- 파일에서 문자열을 읽어와 화면에 출력해 본다.
 - ◆ 크기, 색상, alignment가 지정되어 있어야 한다.

