

MFC 시작하기

- MFC(Microsoft Foundation Class) 개요
- Visual Studio 2017 프로그램 설치하기
- 마우스 클릭 프로그램 작성
- 그림 그리기 프로그램 작성
- 마우스 좌표값 구하기 프로그램 작성

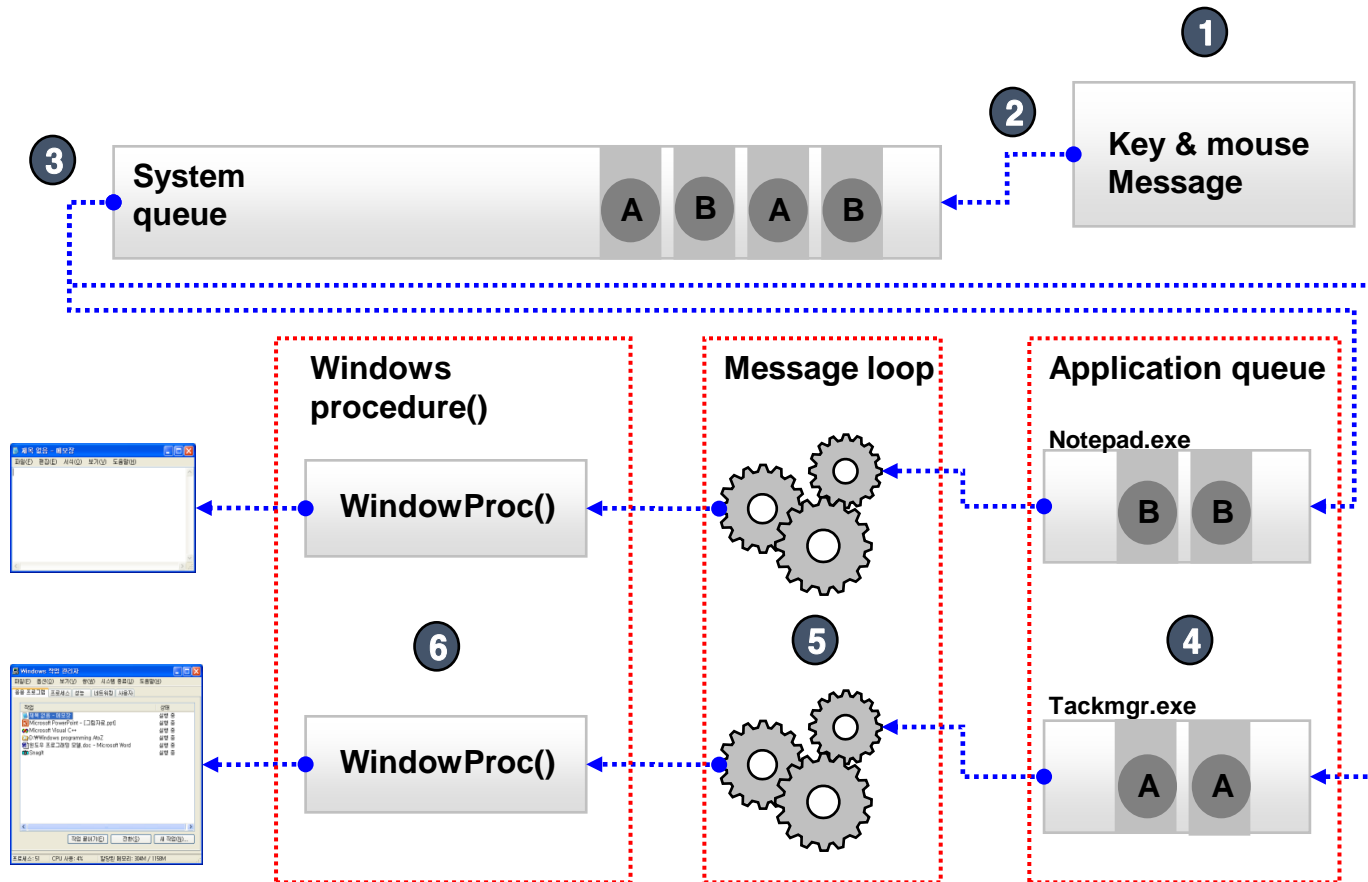
C++ 언어

- 객체지향 언어(Object-Oriented Programming, OOP)
- **C언어의 장점 + 객체지향 속성**
- 객체지향의 기본 구성요소
 - (1) Class : attribute + behavior 를 정의
 - (2) Object : class를 구체화 한 형태
- 객체지향 언어의 특징
 - (1) **추상화**: 불필요한 정보는 숨기고 중요한 정보만 표현
-> 프로그램 간단화
 - (2) **상속성**: 기존 클래스의 자료와 연산을 이용하는 새로운 클래스 생성 가능
 - (3) **캡슐화**: 데이터+함수로써 독립적으로 활용가능
 - (4) **다형성**: 한 요소에 여러 개념을 넣은 것
 - (5) **정보은닉**: 세부 구현 정보를 노출시키지 않음
-> 내부 구현 방법을 독립적으로 수정 가능.

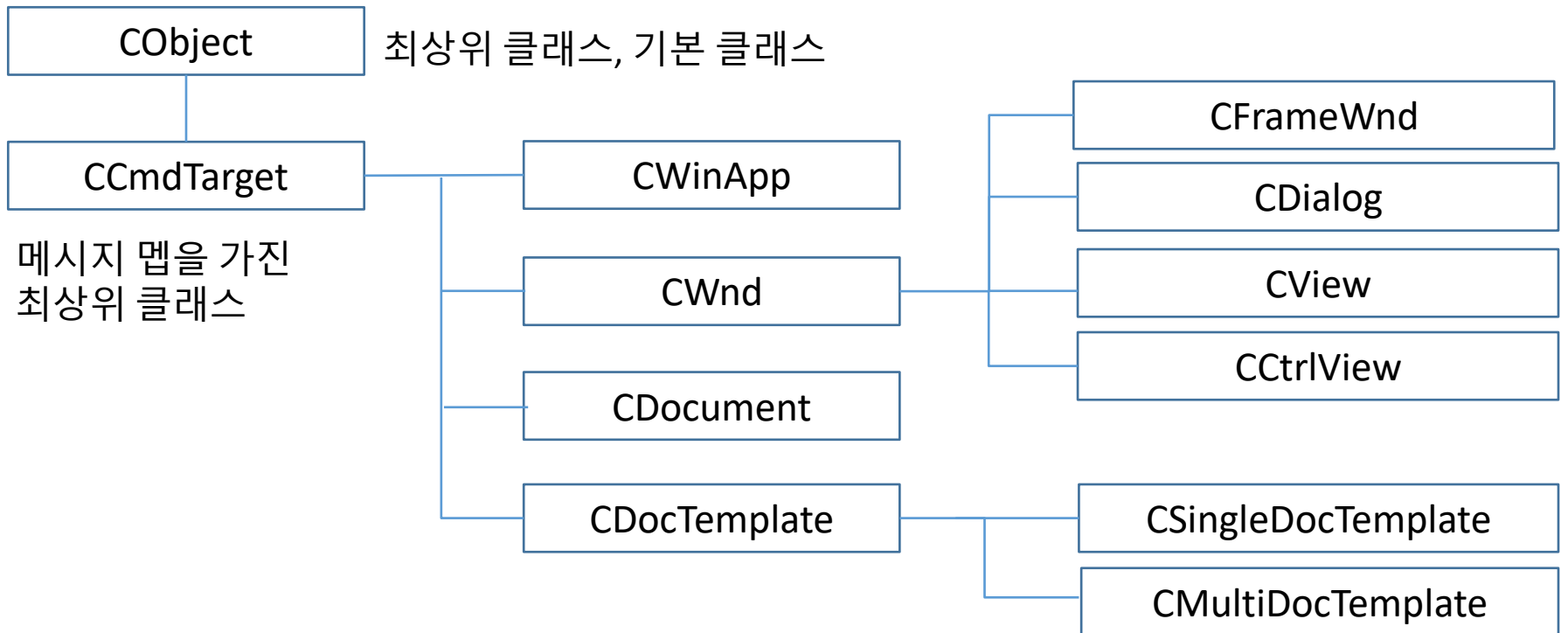
윈도우 프로그래밍

- 윈도우 프로그램의 특징
 - **GUI**(Graphic User Interface: 그래픽 사용자 인터페이스) 기반
 - **Multi-Tasking**
 - **Multi-Threading**
- **API** (Application Programming Interface)
 - 윈도우 내에 포함된 내부 코드를 프로그램 개발에 이용
 - 프로그램을 직접 설계/제어 가능
 - 필요한 부분만을 코딩하여 사용가능 - 코딩 양 감소
- **MFC** (Microsoft Foundation Class)
 - C++ 클래스 라이브러리
 - API보다 편리하게 프로그래밍 가능
 - 프로그램 용량이 커짐

윈도우 프로그래밍 모델 (message-driven 방식)



MFC Class 구성도



Win32 프로젝트

- Win32 프로젝트 생성
 - “파일” - “프로젝트” - “생성”
 - “Visual C++” - “Windows 데스크톱” - “Windows 데스크톱 응용 프로그램”
- 빌드(Control+Shift+B) -> 실행(Control+F5)
- 디렉토리 구조

실행파일 위치

Debug

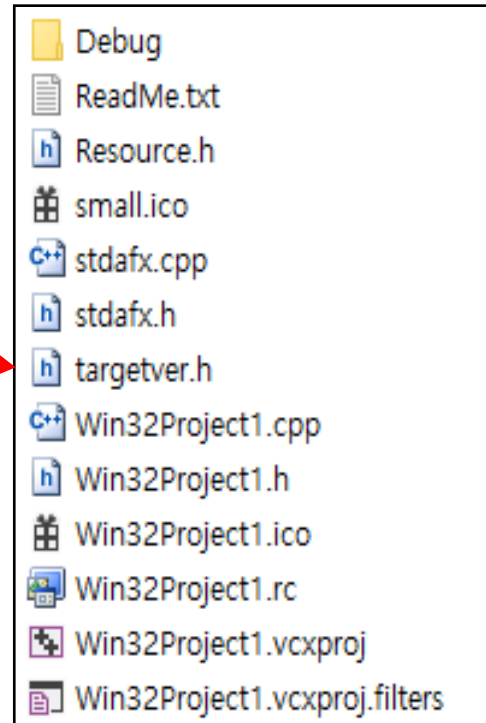
소스파일 위치

Win32Project1

솔루션파일

Win32Project1.sln

Win32Project1.VC.db



Win32 프로젝트

```
int APIENTRY wWinMain(_In_ HINSTANCE hInstance,  
                    _In_opt_ HINSTANCE hPrevInstance,  
                    _In_ LPWSTR lpCmdLine,  
                    _In_ int nCmdShow) {
```

.....

```
LoadStringW(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);  
LoadStringW(hInstance, IDC_WIN32PROJECT1, szWindowClass, MAX_LOADSTRING);
```

```
MyRegisterClass(hInstance);
```

프로그램 정보 등록.

(예: 자신의 WindowProc() callback 함수를 등록)

```
if (!InitInstance (hInstance, nCmdShow)) {  
    return FALSE;  
}
```

윈도우 생성

.....

```
HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC_WIN32PROJECT1));
```

```
MSG msg;
```

메시지 큐에서 메시지 가져오기.
WM_QUIT 일 때 : 0을 return.

Accelerator key를
메뉴입력으로 바꾸기.

```
// 기본 메시지 루프입니다
```

```
while (GetMessage(&msg, nullptr, 0, 0)) {  
    if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg)) {  
        TranslateMessage(&msg);  
        DispatchMessage(&msg);  
    }  
}
```

메시지 해석

메시지 처리. (WindowProc() 에서 처리하게 함.)

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

```
}
```

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam) {
```

```
switch (message) {
```

```
case WM_COMMAND:
```

메뉴나 버튼선택에 의해 발생

```
{
```

```
int wml = LOWORD(wParam);  
// 메뉴 선택을 구문 분석합니다.
```

```
switch (wml) {
```

```
case IDM_ABOUT:
```

```
DialogBox(hInst, MAKEINTRESOURCE(IDD_ABOUTBOX), hWnd, About);  
break;
```

```
case IDM_EXIT:
```

```
DestroyWindow(hWnd);  
break;
```

```
default:
```

```
return DefWindowProc(hWnd, message, wParam, lParam);
```

```
}
```

```
}
```

```
break;
```

```
case WM_PAINT:
```

윈도우를 다시 그려야 할 때 발생

```
{
```

```
PAINTSTRUCT ps;  
HDC hdc = BeginPaint(hWnd, &ps);  
// TODO: 여기에 hdc를 사용하는 그리기 코드를 추가합니다.  
EndPaint(hWnd, &ps);
```

```
}
```

```
break;
```

```
case WM_DESTROY:
```

WM_QUIT 전송

```
PostQuitMessage(0);
```

```
break;
```

```
default:
```

처리하지 않은 메세지들은 윈도우 운영체제에 맡김.

```
return DefWindowProc(hWnd, message, wParam, lParam);
```

```
}
```

```
return 0;
```

윈도우 프로시저 함수 (WindowProc)

[구글]

msdn mfc classes

로 찾아 msdn site 들어가서 어떤 class들이 있는지 훑어보기.

[과제]

msdn mfc classes

에서 아래 메시지 내용 읽고 정리해 오기.

WM_LBUTTONDOWN...,
WM_KEY...,
WM_MOUSE...

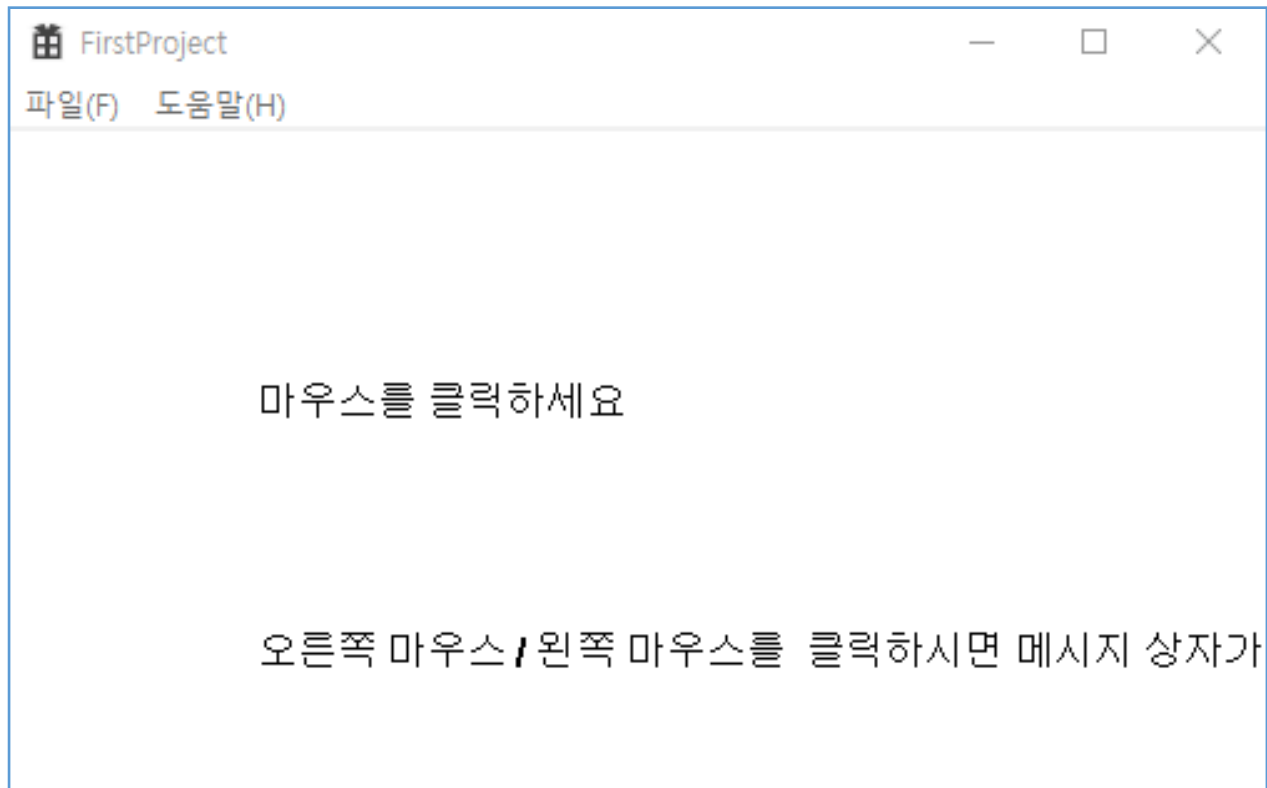
마우스 클릭 프로그램 (FirstProject)

(1) 새 프로젝트 생성

(“파일”-“새로만들기”-“프로젝트”

-> “Visual C++”-“윈도우 데스크톱 응용프로그램”)

-> 이름: **FirstProject**



마우스 클릭 프로그램 (FirstProject)

(2) FirstProject.cpp 수정

```
LRESULT CALLBACK WndProc(HWND hWnd,UINT message,WPARAM wParam,LPARAM lParam){
```

```
.....
```

```
case WM_LBUTTONDOWN:
```

```
    MessageBox(hWnd, TEXT("왼쪽 마우스를 클릭하셨습니다."),  
                TEXT("왼쪽 마우스 클릭!"), MB_OK);
```

```
    return 0;
```

```
case WM_RBUTTONDOWN:
```

```
    MessageBox(hWnd, TEXT("오른쪽 마우스를 클릭하셨습니다."),  
                TEXT("오른쪽 마우스 클릭!"), MB_OK);
```

```
    return 0;
```

```
case WM_PAINT: {
```

```
    PAINTSTRUCT ps;
```

```
    HDC hdc = BeginPaint(hWnd, &ps);
```

```
    TCHAR str1[200] = TEXT("마우스를 클릭하세요");
```

```
    TCHAR str2[200] =
```

```
    TEXT("오른쪽/왼쪽 마우스를 클릭하시면 메시지 상자가 나타납니다.");
```

```
    TextOut(hdc, 100, 100, str1, lstrlen(str1));
```

```
    TextOut(hdc, 100, 200, str2, lstrlen(str2));
```

```
    EndPaint(hWnd, &ps);
```

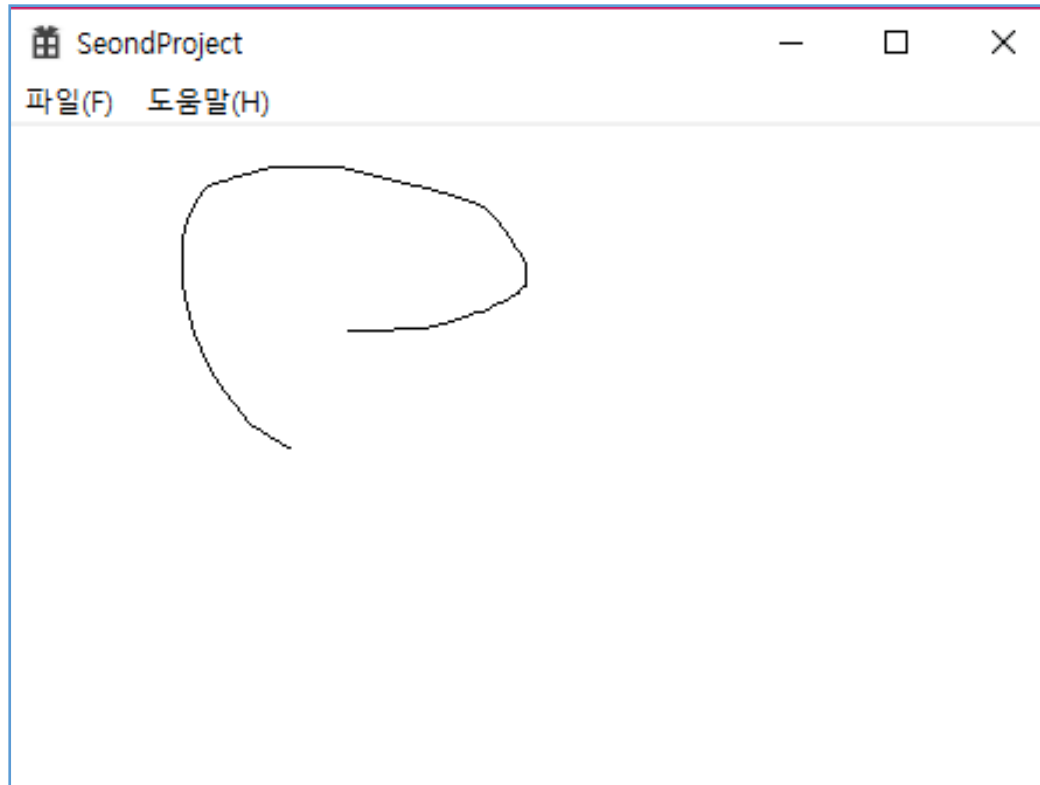
그림 그리기 프로그램 (SecondProject)

(1) 새 프로젝트 생성

(“파일”-“새로만들기”-“프로젝트”

-> “Visual C++”-“윈도우 데스크톱 응용프로그램”)

-> 이름: **SecondProject**



(2) SecondProject.cpp 수정

```
LRESULT CALLBACK WndProc(HWND hWnd,UINT message,WPARAM wParam,LPARAM lParam){
    HDC hdc;
    PAINTSTRUCT ps;
    static POINT MousePos; //마우스 x,y 좌표를 저장할 변수
    static BOOL bDrawing = FALSE;
    switch (message){
    case WM_LBUTTONDOWN:
        MousePos.x = LOWORD(lParam);
        MousePos.y = HIWORD(lParam);
        bDrawing = TRUE;
        SetCapture(hWnd);
        return 0;
    case WM_MOUSEMOVE:
        if (bDrawing == TRUE) {
            hdc = GetDC(hWnd);
            MoveToEx(hdc, MousePos.x, MousePos.y, NULL);
            MousePos.x = LOWORD(lParam);
            MousePos.y = HIWORD(lParam);
            LineTo(hdc, MousePos.x, MousePos.y);
            ReleaseDC(hWnd, hdc);
        }
        return 0;
    case WM_LBUTTONUP:
        bDrawing = FALSE;
        ReleaseCapture();
        return 0;
    }
```

LPARAM으로 마우스 위치가 들어옴.

**마우스 클릭 프로그램
(SecondProject)**

MFC 응용프로그램 (MouseMFC)

(1) 새 프로젝트 생성

(“파일”-“새로만들기”-“프로젝트”

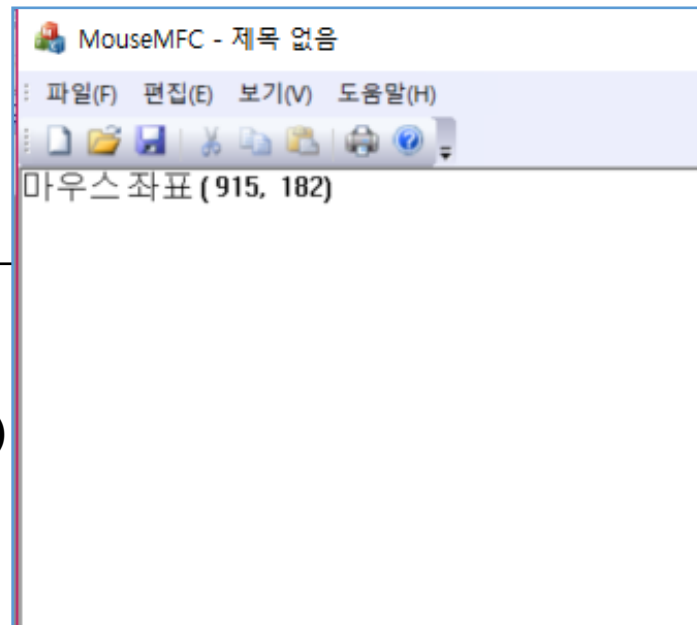
-> “Visual C++” - “MFC” - “MFC 응용프로그램”)

-> 이름: **MouseMFC**

-> 응용 프로그램 종류: “단일문서”

- 프로젝트 스타일: “MFC Standard” - “마침”

-> 고급 프레임 창 : 탐색기 도킹 창, 출력 도킹 창, 속성 도킹 창 **uncheck**



MFC 응용 프로그램

고급 기능 옵션

응용 프로그램 종류

문서 템플릿 속성

사용자 인터페이스 기능

고급 기능

생성된 클래스

고급 기능:

- 인쇄 및 인쇄 미리 보기(P)
- 자동화(U)
- ActiveX 컨트롤(R)
- MAPI(메시징 API)(I)
- Windows 소켓(W)
- Active Accessibility(A)
- 공용 컨트롤 매니페스트(M)
- 다시 시작 관리자 지원(G)
- 이전에 열려 있던 문서 다시 열기(Y)

고급 프레임 창:

- 탐색기 도킹 창(D)
- 출력 도킹 창(O)
- 속성 도킹 창(S)
- 탐색 창(T)
- 캡션 표시줄(B)
- 창을 표시하거나 활성화하는 고급 프레임 메뉴 항목(F)

최근 파일 목록의 파일 수(N):

4

_tWinMain

- **_tWinMain 함수 찾아가기.**
 - **InitInstance() 에서 break 잡아 callstack에서 _tWinMain 함수 찾기.**
- **Mfc 소스 찾아 가기**
 - C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\atl\mfcc\src 에서 AfxWinMain() 찾아보기.**

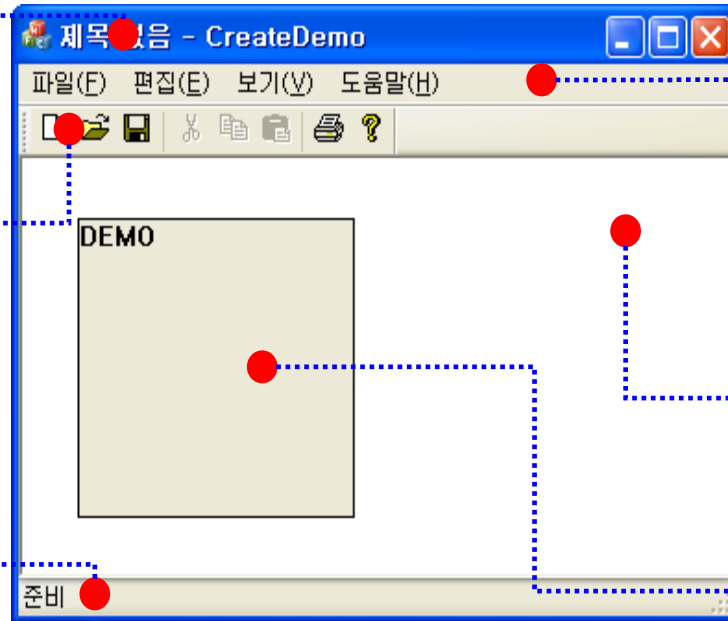
SDI 윈도우 구성

기본 도구 모음이 맘에 들지 않아 바꾸려면 ?

CMainFrame
Top frame
window

CToolBar
m_wndToolBar
(CMainFrame
class member)

CStatusBar
m_wndStatusBar
(CMainFrame
class member)



Menu
CMainFrame
window

CCreateDemoView
w
Default view
window
CWnd
m_wndChild
(CCreateDemoView
class member)

화면상에서 원하는 부분을 제어하려면 어떤 클래스의 어느 멤버 함수를 코딩 해야 할지 알아야 함.