8장. 단일문서/다중문서 프로그램

this 포인터의 이해

```
#include <iostream>
using namespace std;

class Mine {
    int x, y;
    public:
    void Set(int x, int y) {
        cout << "Set to : " << this << endl;
        this->x = x;
        this->y = y;
    }
    void Print() {
        cout << "this : " << this  << endl;
        cout << "x=" << x << ", y=" << y << endl;
    }
};
```

```
int main() {
    Mine m1, m2;

    m1.Set(1, 2);
    m2.Set(10, 20);

    m1.Print();
    m2.Print();

    return 0;
}
```

Document/View 구조

[doc.h]

```
public:
    CString m_str;
```

[doc.cpp]

```
CMFCApplication1Doc::CMFCApplication1Doc()
{
    m_str = _T("my text");
}
```

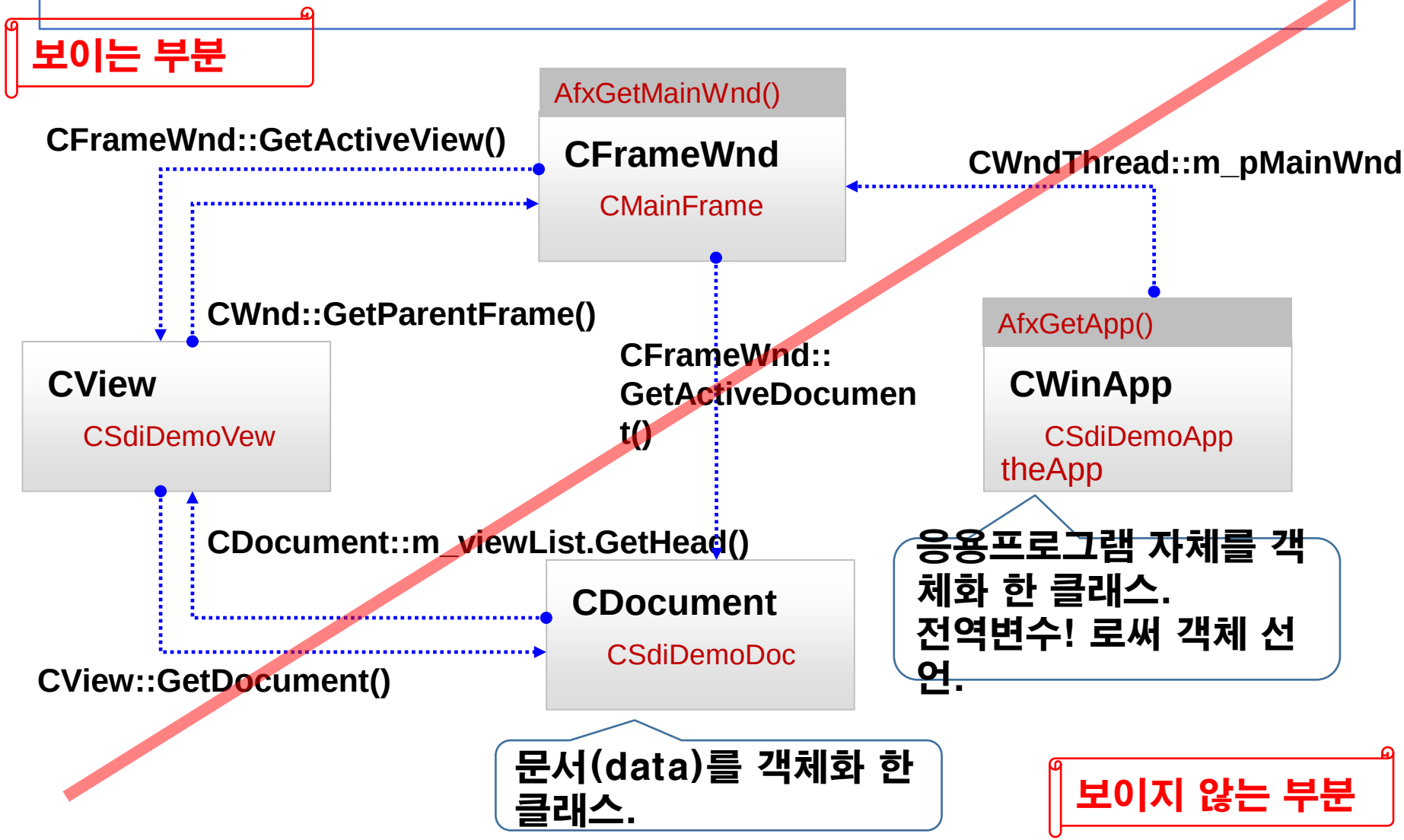
Break point!

```
void CMFCApplication1Doc::Serialize(CArchive& ar)
{
    if (ar.IsStoring()) {
        ar << m_str;
    }
    else {
        m_str = _T("");
        ar >> m_str;
    }
}
```

[view.cpp]

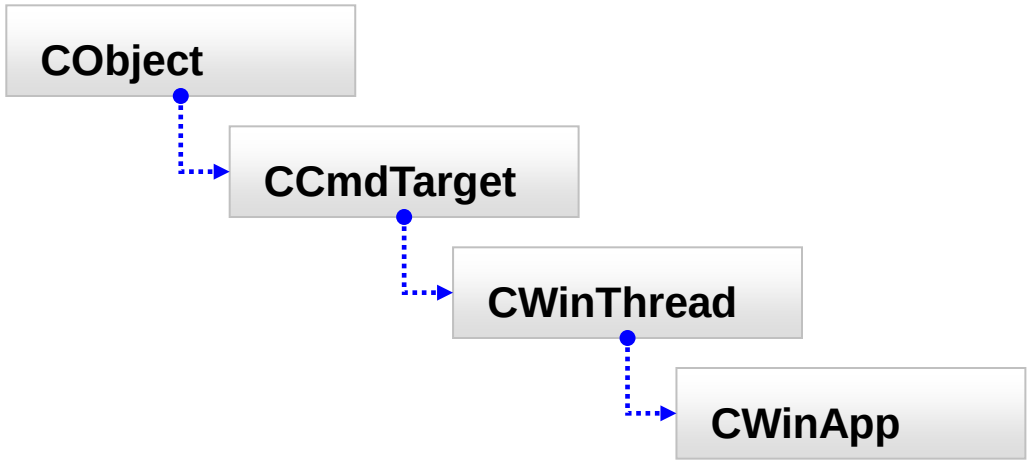
```
void CMFCApplication1View::OnDraw(CDC* pDC)
{
    CMFCApplication1Doc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    pDC->TextOutW(0, 0, pDoc->m_str);
}
```

SDI 클래스 관계도


보이는 부분
AfxGetMainWnd()
CFrameWnd
CMainFrame
CFrameWnd::GetActiveView()
CWndThread::m_pMainWnd
CWnd::GetParentFrame()
CFrameWnd::
GetActiveDocumen
t()
AfxGetApp()
CWinApp
CSdiDemoApp
theApp
CView
CSdiDemoVew
CDocument::m_viewList.GetHead()
CDocument
CSdiDemoDoc
CView::GetDocument()
응용프로그램 자체를 객체화 한 클래스.
전역변수! 로써 객체 선언.
문서(data)를 객체화 한 클래스.
보이지 않는 부분

CWinApp

응용 프로그램 자체를 구현한 클래스. 그러므로 이의 객체는 "응용 프로그램".



CWinApp 주요 멤버

멤버 변수	기능
m_hInstance	현재 응용 프로그램의 인스턴스 핸들입니다. WinMain( ) 함수의 첫 번째 파라미터인 hInstance와 같은 것입니다.
m_lpCmdLine	WinMain( ) 함수의 lpCmdLine 파라미터와 같은 것입니다. 프로그램을 실행하였을 때 명령줄(Command-line) 정보가 들어 있습니다.
m_nCmdShow	WinMain( ) 함수의 마지막 파라미터인 nCmdShow와 같은 것입니다.
m_pActiveWnd	응용 프로그램의 최상위 프레임 윈도우에 대한 포인터입니다. SDI 구조에서 이 값은 CMainFrame 클래스 객체의 포인터입니다.
m_pszAppName	응용 프로그램의 제목에 해당하는 문자열의 포인터입니다. CreateDemo 예제의 경우 이 값이 CreateDemo였습니다. 문자열의 좀더 정확한 정보는 문자열 테이블(String Table)에 들어 있는 AFX_IDS_APP_TITLE에 해당하는 값입니다.
m_pszExeName	빌드한 실행 파일에서 확장자(.exe)를 제외한 파일명입니다.

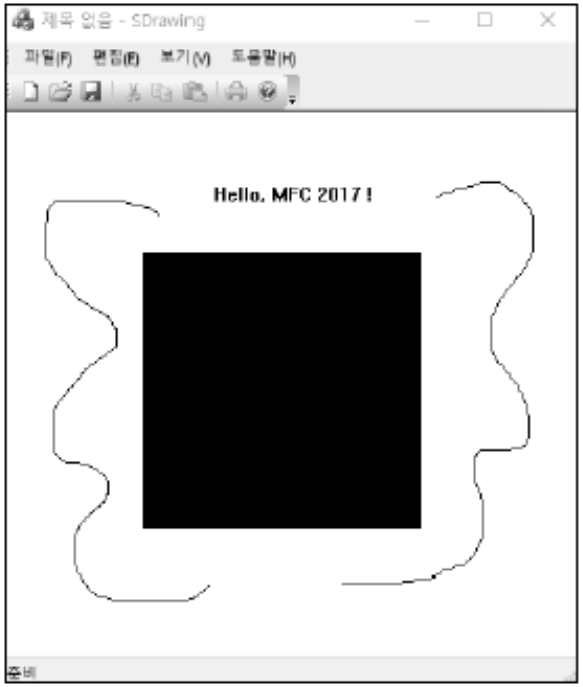
가상 함수	기능
InitInatance()	응용 프로그램 초기화시점에 호출됨. 처음에 재정의 되어 있음.
ExitInstance()	응용 프로그램 종료직전에 호출됨. (사용한 자원 정리 등)
Run()	Message loop를 포함하고 있음. 이 함수에서 return하면 프로그램 종료

SDI 프로그램

(1) 새 프로젝트 생성
("파일"–"새로만들기"–"프로젝트"
-> "Visual C++" – "MFC" - "MFC 응용프로그램")
-> 이름: SDrawing
-> 응용 프로그램 종류: "단일문서" - 프로젝트 스타일: "Visual Studio"
– 비주얼 스타일 및 색: "Visual Studio 2008"

(2) 화면에 기본 문자열 보이기

```
void CSDrawingView::OnDraw(CDC* pDC) {
    CSDrawingDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    pDC->TextOutW(150, 50, _T("Hello, MFC 2017 !"));
}
```



SDI 프로그램
(네모 영역에 대한 마우스 입력 처리)

(3) 네모 영역 그리기

View 클래스에 멤버 변수 추가 (SDrawingView.h)

```
    int m_crColor;
    CRect m_reRect;
```

View클래스 constructor에서 초기화.

```
CSDrawingView::CSDrawingView()
: m_crColor(BLACK_BRUSH), m_reRect(100,100,300,300)
{
}
```

OnDraw에서 현재 색으로 네모 그리기.

```
void CSDrawingView::OnDraw(CDC* pDC) {
    CSDrawingDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
    return;
        pDC->TextOutW(150, 50, _T("Hello, MFC 2017 !"));
    pDC->SelectStockObject(m_crColor);
    pDC->Rectangle(m_reRect);
}
```

SDI 프로그램
(네모 영역에 대한 마우스 입력 처리)

(4) 네모 영역에 대한 마우스 입력 처리

WM_LBUTTONDOWN 메시지 핸들러 작성

```
void CSDrawingView::OnLButtonDown(UINT nFlags, CPoint point){
    if (m_reRect.PtInRect(point)) {
        if (m_crColor == BLACK_BRUSH || m_crColor == GRAY_BRUSH)
            m_crColor = WHITE_BRUSH;
        else
            m_crColor = GRAY_BRUSH;
        InvalidateRect(m_reRect);
    }
    CView::OnLButtonDown(nFlags, point);
}
```

SDI 프로그램
(네모 영역에 대한 마우스 입력 처리)

(5) 마우스 움직임에 따라 선 그리기

마우스 위치 저장 멤버 변수 추가

```
    int m_ptX, m_ptY;
```

"새 문서" : 기존 문서를 닫고 새 문서가 열림.

View클래스 constructor에서 초기화.

```
CSDrawingView::CSDrawingView()
: m_ptX(0), m_ptY(0)
, m_crColor(0) ,m_reRect(100,100,300,300)
```

WM_LBUTTONDOWN 왔을 때 위치 저장

```
void CSDrawingView::OnLButtonDown(UINT nFlags, CPoint point) {
    m_ptX = point.x;
    m_ptY = point.y;
```

WM_MOUSEMOVE 메시지 핸들러 추가

```
void CSDrawingView::OnMouseMove(UINT nFlags, CPoint point) {
    if (nFlags & MK_LBUTTON) {
        CClientDC dc(this);
        dc.MoveTo(m_ptX, m_ptY);
        dc.LineTo(point.x, point.y);
        m_ptX = point.x;
        m_ptY = point.y;
    }
    CView::OnMouseMove(nFlags, point);
```

MDI 프로그램

(1) 새 프로젝트 생성

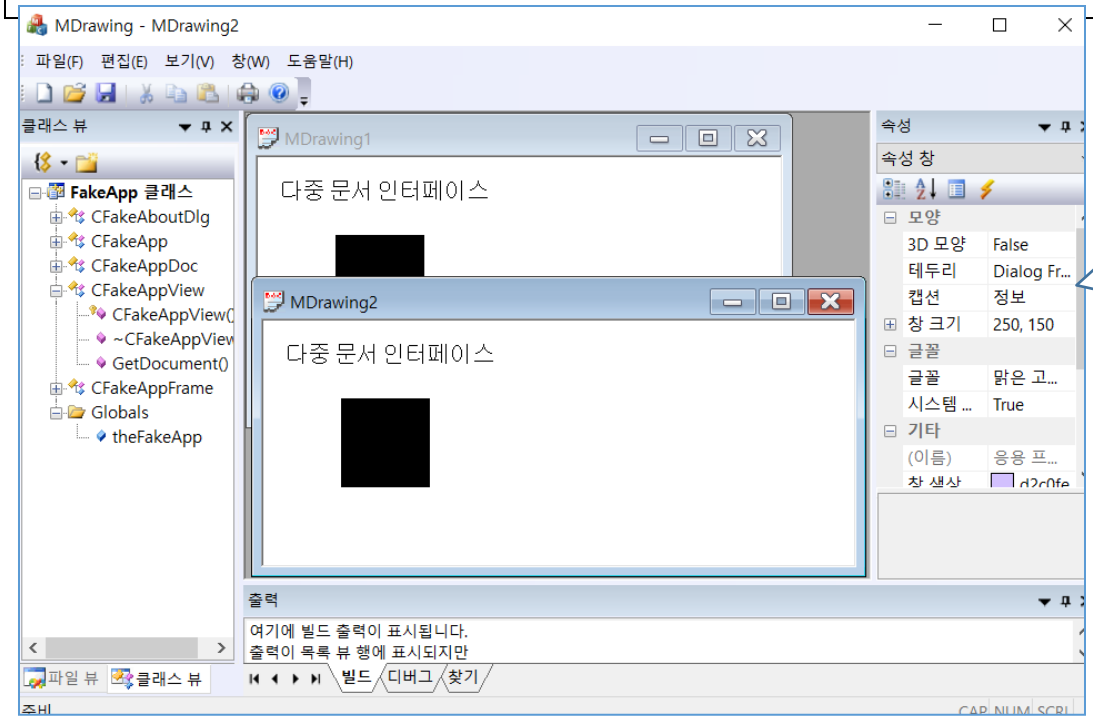
("파일"–"새로만들기"–"프로젝트"

-> "Visual C++" – "MFC" - "MFC 응용프로그램")

-> 이름: MDrawing

-> 응용 프로그램 종류: "다중문서" - 프로젝트 스타일: "Visual Studio"

– 비주얼 스타일 및 색: "Visual Studio 2008"



"새 문서" : 기존 문서를 유지한 채로 새 문서 창을 추가로 연다.

MDI 프로그램

(2) 화면에 보이기

```
void CMDrawingView::OnDraw(CDC* pDC) {
    CMDrawingDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    RECT rect;
    rect.left = 70;
    rect.top = 70;
    rect.bottom = 150;
    rect.right = 150;
    pDC->SelectStockObject(BLACK_BRUSH);
    pDC->Rectangle(&rect);

    pDC->TextOutW(20, 20, _T("다중 문서 인터페이스"));
}
```