

도큐먼트/뷰 구조 I

(2주차)

학습개요

- 학습 목표

- SDI 응용 프로그램의 기본 구조를 이해한다.
- 도큐먼트/뷰 구조의 핵심 함수를 익힌다.
- 명령 라우팅의 개념을 이해한다.
- 분할 윈도우를 생성하고 활용하는 방법을 익힌다.

- 학습 내용

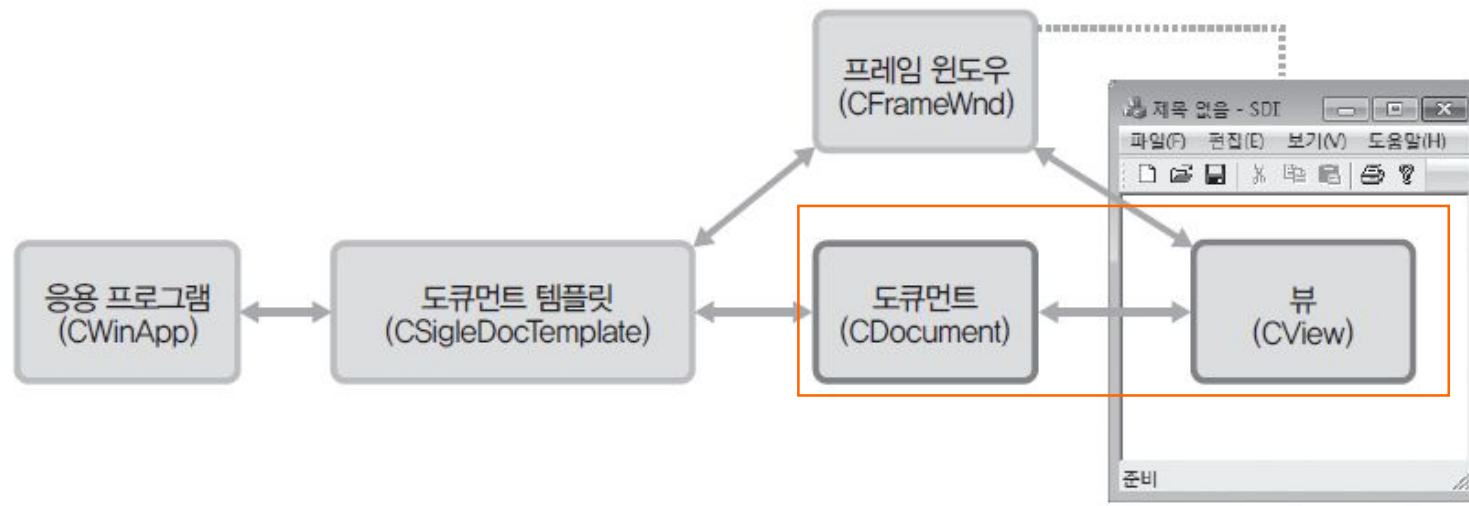
- SDI 응용 프로그램 구조
- 도큐먼트/뷰 클래스 주요 함수
- 명령 라우팅
- 분할 윈도우
- 실습

개요

- SDI(Single Document Interface)
 - 응용 프로그램이 한 번에 하나의 문서만 열어서 작업할 수 있는 사용자 인터페이스
- MDI(Multiple Document Interface)
 - 응용 프로그램이 동시에 여러 개의 문서를 열어서 작업할 수 있는 사용자 인터페이스

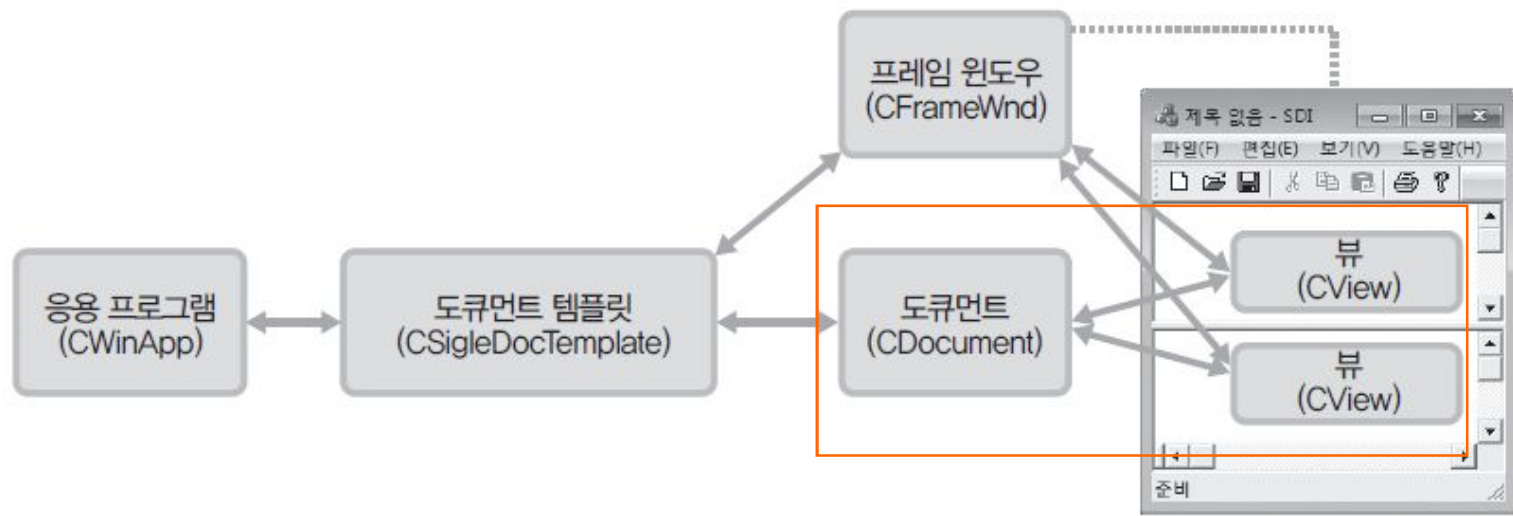
SDI 응용 프로그램 구조

- SDI 응용 프로그램 구조 1



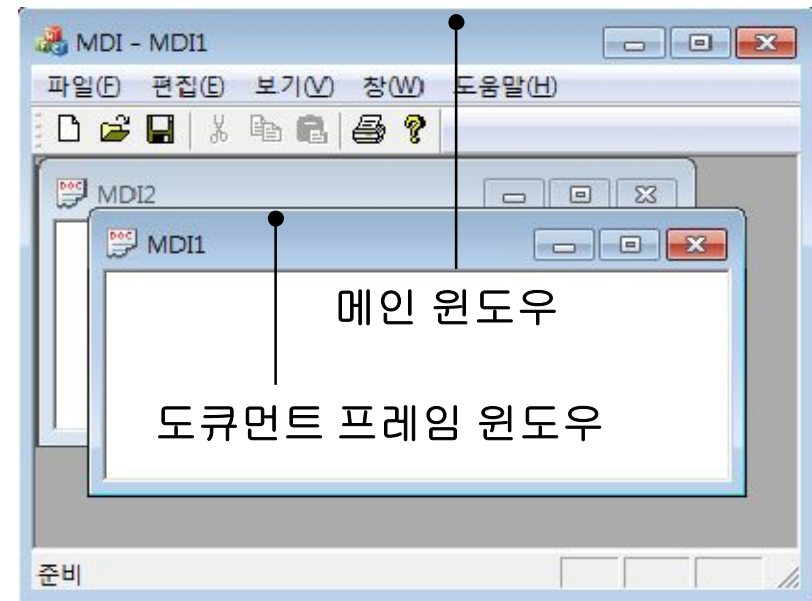
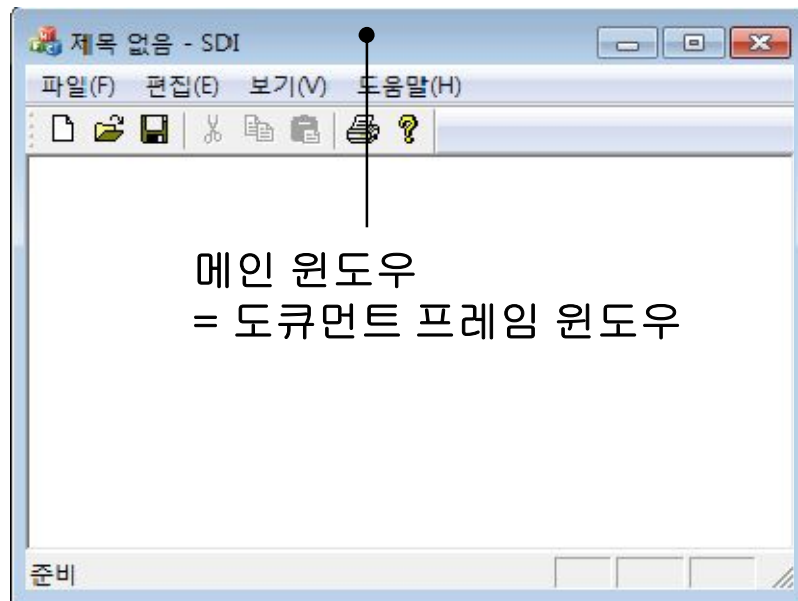
SDI 응용 프로그램 구조

- SDI 응용 프로그램 구조 2



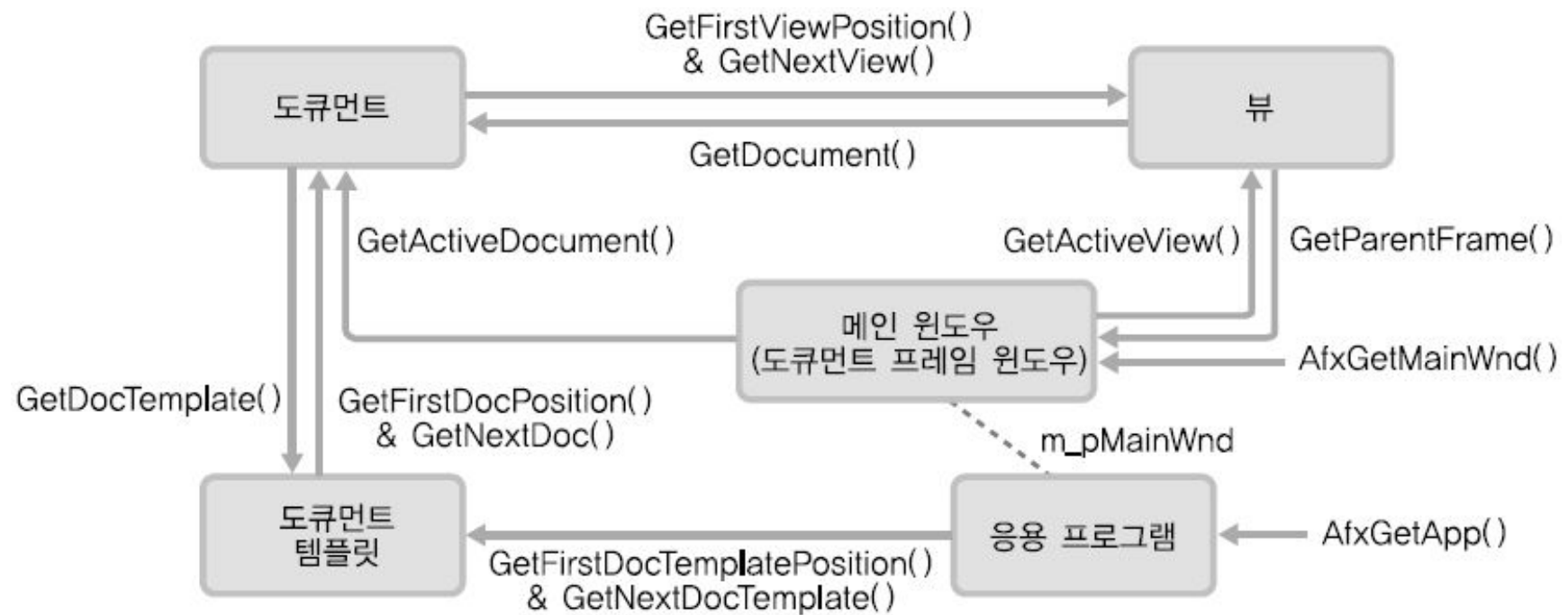
SDI 응용 프로그램 구조

- 도큐먼트 프레임 윈도우
 - 도큐먼트의 내용을 화면에 표시하는 역할을 하는 뷰를 자식으로 갖는 윈도우 = 뷰의 부모 윈도우 = 뷰를 감싸고 있는 윈도우



SDI 응용 프로그램 구조

- 주요 객체 참조 함수 - SDI



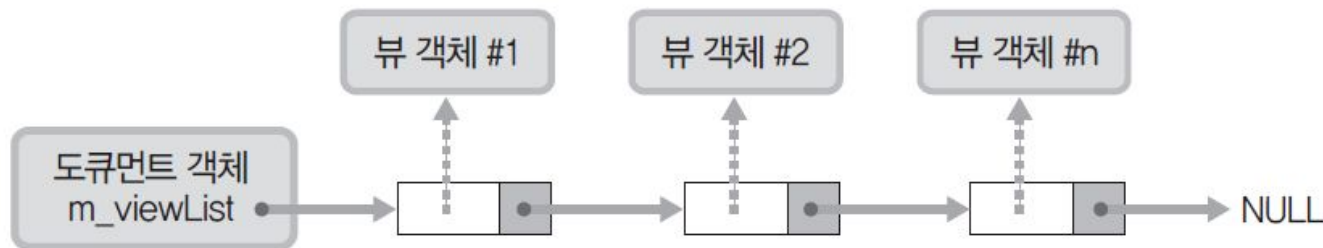
SDI 응용 프로그램 구조

- 함수의 형태와 의미 (1/4)
 - CWinApp* AfxGetApp();
 - 응용 프로그램 객체의 주소를 리턴
 - CWnd* AfxGetMainWnd();
 - 메인 윈도우 객체의 주소를 리턴
 - CFrameWnd* CWnd::GetParentFrame();
 - 도큐먼트 프레임 윈도우 객체의 주소를 리턴
 - CView* CFrameWnd::GetActiveView();
 - 활성 뷰 객체의 주소를 리턴

SDI 응용 프로그램 구조

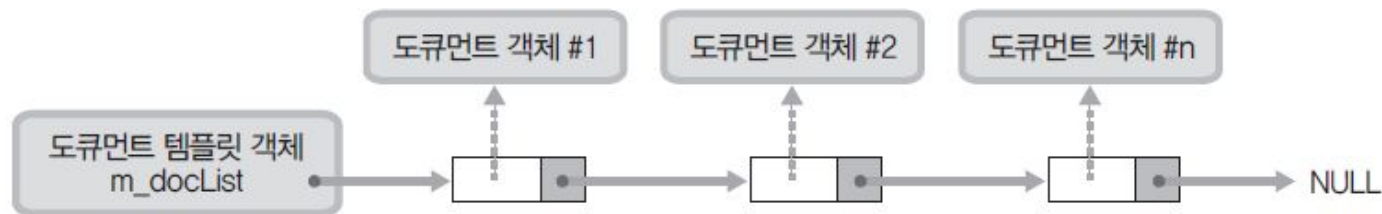
- 함수의 형태와 의미 (2/4)

- `CDocument* CFrameWnd::GetActiveDocument();`
 - 활성 도큐먼트 객체의 주소를 리턴
- `CDocument* CView::GetDocument();`
 - 뷰 객체와 연결된 도큐먼트 객체의 주소를 리턴
- `POSITION CDocument::GetFirstViewPosition();`
`CView* CDocument::GetNextView(POSITION& rPosition);`
 - 도큐먼트 객체와 연결된 모든 뷰 객체의 주소를 리턴



SDI 응용 프로그램 구조

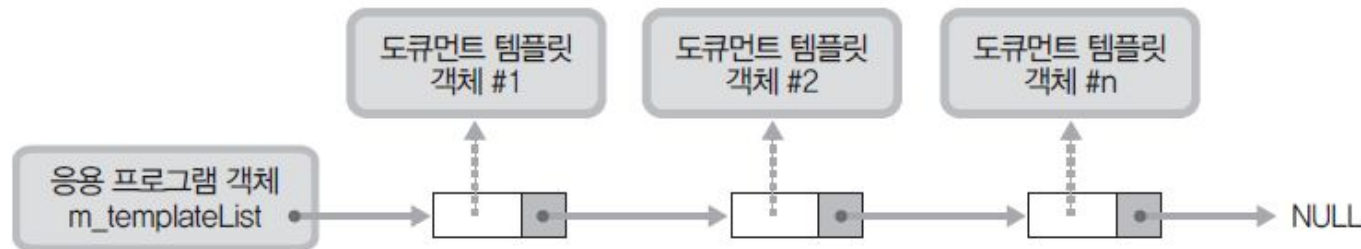
- 함수의 형태와 의미 (3/4)
 - POSITION CDocTemplate::GetFirstDocPosition();
CDocument* CDocTemplate::GetNextDoc(POSITION& rPos);
 - 도큐먼트 템플릿 객체가 관리하는 모든 도큐먼트 객체의 주소를 리턴



SDI 응용 프로그램 구조

- 함수의 형태와 의미 (4/4)

- POSITION CWinApp::GetFirstDocTemplatePosition();
CDocTemplate* CWinApp::GetNextDocTemplate(POSITION&pos);
 - 응용 프로그램 객체가 관리하는 모든 도큐먼트 템플릿 객체의 주소를 리턴



- CDocTemplate* CDocument::GetDocTemplate();
 - 도큐먼트 템플릿 객체의 주소를 리턴

SDI 응용 프로그램 구조

- InitInstance() 함수 (1/2)

```
BOOL CSDITestApp::InitInstance()  
{
```

```
...
```

```
①SetRegistryKey(_T("로컬 응용 프로그램 마법사에서 생성된 응용 프로그램"));  
②LoadStdProfileSettings(4);
```

```
CSingleDocTemplate* pDocTemplate;  
pDocTemplate = new CSingleDocTemplate(  
    IDR_MAINFRAME,  
    RUNTIME_CLASS(CSDITestDoc),  
    RUNTIME_CLASS(CMainFrame),  
    RUNTIME_CLASS(CSDITestView));  
if(!pDocTemplate)  
    return FALSE;  
AddDocTemplate(pDocTemplate);
```

SDI 응용 프로그램 구조

- InitInstance() 함수 (2/2)

```
CCommandLineInfo cmdInfo;  
ParseCommandLine(cmdInfo);  
  
③EnableShellOpen();  
④RegisterShellFileTypes(TRUE);  
  
if(!ProcessShellCommand(cmdInfo))  
    return FALSE;  
  
m_pMainWnd->ShowWindow(SW_SHOW);  
m_pMainWnd->UpdateWindow();  
  
⑤m_pMainWnd->DragAcceptFiles();  
    return TRUE;  
}
```

- **도큐먼트 문자열**

SDITest\n\nSDITest\nSDITest Files (*.sdi)\n.sdi\nSDITest.Document\nSDITest.Document

① ② ③ ④ ⑤ ⑥ ⑦

번호	기능
①	프레임 윈도우의 타이틀 바에 표시되는 제목이다.
②	새로 생성한 문서의 제목이다. 위와 같이 생략하면 기본값인 '제목없음'으로 설정된다.
③	두 개 이상의 도큐먼트 타입을 지원하는 MDI 응용 프로그램에서만 사용된다. 새로운 문서를 생성할 때 나타나는 도큐먼트 타입을 묻는 대화 상자에 표시되는 문자열이다.
④	[열기]와 [다른 이름으로 저장] 대화상자에 표시된다.
⑤	파일의 기본 확장자로 사용된다.
⑥	레지스트리에 등록되는 도큐먼트 타입 ID로, 공백을 허용하지 않는다.
⑦	레지스트리에 등록되는 도큐먼트 타입 문자열로, 공백을 허용한다.

도큐먼트 클래스

- 도큐먼트 클래스의 주요 함수 (1/2)
 - void SetModifiedFlag(BOOL bModified = TRUE);
 - 데이터가 수정되었음을 도큐먼트 객체에 알림
 - void UpdateAllViews(CView* pSender, LPARAM lHint = 0L, CObject* pHint = NULL);
 - 모든 뷰 객체에게 화면을 갱신하도록 지시함

```
CDocument::UpdateAllViews();  
→CView::OnUpdate(); // Invalidate() 함수를 호출하여 화면 전체를 무효화한다.  
→CWnd::Invalidate(TRUE); // WM_PAINT 메시지를 발생시킨다.  
→CWnd::OnPaint(); // CPaintDC 객체를 만든 후 OnDraw() 함수에 전달한다.  
→CView::OnDraw(); // 화면에 출력한다.
```

도큐먼트 클래스

- 도큐먼트 클래스의 주요 함수 (2/2)
 - virtual BOOL OnNewDocument();
 - 새 문서를 생성할 때 자동으로 호출됨
 - virtual BOOL OnOpenDocument(LPCTSTR lpszPathName);
 - [파일]-[열기...] 메뉴를 선택할 때 자동으로 호출됨
 - virtual void DeleteContents();
 - OnNewDocument()와 OnOpenDocument() 함수 이전에 자동으로 호출됨
 - virtual void Serialize(CArchive& ar);
 - 파일을 열거나 저장할 때 자동으로 호출됨

도큐먼트 클래스

- 도큐먼트 클래스의 가상 함수 호출 순서
 - [파일]-[새 파일] 메뉴를 선택할 때

`DeleteContents()` ⇒ `OnNewDocument()`

- [파일]-[열기...] 메뉴를 선택할 때

`DeleteContents()` ⇒ `Serialize()` ⇒ `OnOpenDocument()`

- [파일]-[저장] 또는 [파일]-[다른 이름으로 저장...] 메뉴를 선택할 때

`Serialize()`

뷰 클래스

- 뷰 클래스의 주요 함수
 - virtual void OnDraw(CDC* pDC);
 - 화면 출력, 인쇄, 인쇄 미리보기를 할 때 자동으로 호출됨
 - virtual void OnInitialUpdate();
 - 뷰 객체가 도큐먼트 객체와 연결된 후 화면에 보이기 전에 자동으로 호출됨
 - virtual void OnUpdate(CView* pSender, LPARAM lHint, CObject* pHint);
 - CDocument::UpdateAllViews() 함수와 CView::OnInitialUpdate() 함수에서 호출함

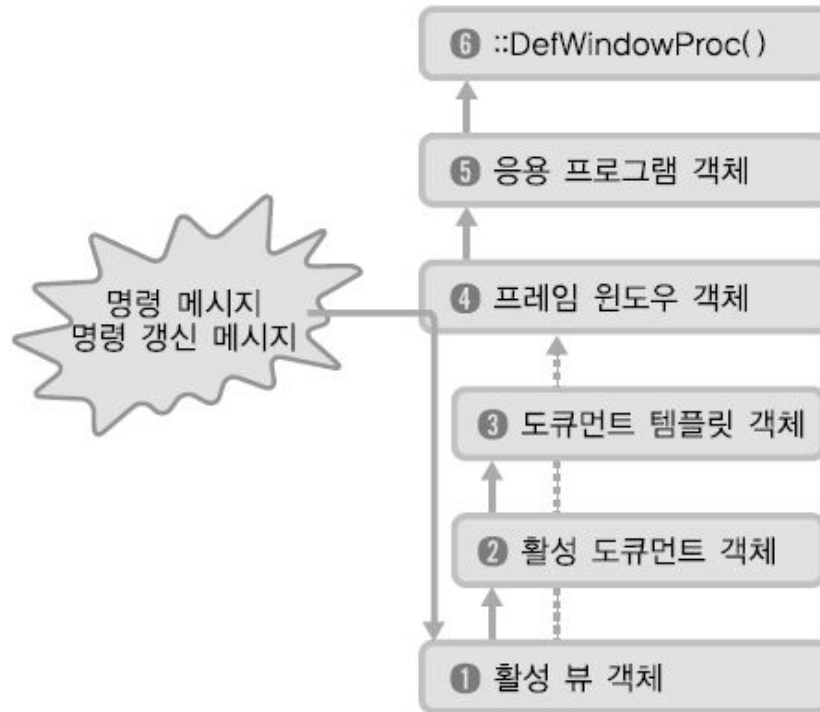
명령 라우팅

- MFC 응용 프로그램에서 사용하는 메시지의 종류

종류	설명	메시지 맵 매크로
윈도우 메시지	윈도우 생성, 종료, 마우스, 키보드 등 다양한 원인으로 발생한다.	ON_WM_□□□()
명령 메시지	메뉴, 툴바, 가속기에서 발생한다.	ON_COMMAND(ID, 함수)
명령 갱신 메시지	메뉴, 툴바, 상태바의 상태를 갱신할 필요가 있을 때 발생한다. MFC에서만 사용하는 고유 메시지다.	ON_UPDATE_COMMAND_UI(ID, 함수)
통지 메시지	컨트롤(자식 윈도우)이 부모 윈도우에 보내는 메시지이다.	ON_□□□(ID, 함수)

명령 라우팅

- 명령 라우팅 순서



unit 5. 분할 윈도우

분할 윈도우

- 동적 분할 윈도우
 - 같은 뷰 클래스를 기반으로 여러 개의 뷰를 생성
 - 총 네 개의 구획(Pane) 생성 가능

```
BEGIN_MESSAGE_MAP(CSDITestView, CView)
//{{AFX_MSG_MAP(CSDITestView)
// NOTE - the ClassWizard will add and remove mappings here
// DO NOT EDIT what you see in these blocks of code
//}}AFX_MSG_MAP
// Standard printing commands
ON_COMMAND(ID_FILE_PRINT, CView::OnFilePrint)
ON_COMMAND(ID_FILE_PRINT_DIRECT, CView::OnFilePrint)
ON_COMMAND(ID_FILE_PRINT_PREVIEW, CView::OnFilePrintPreview)
END_MESSAGE_MAP()
```

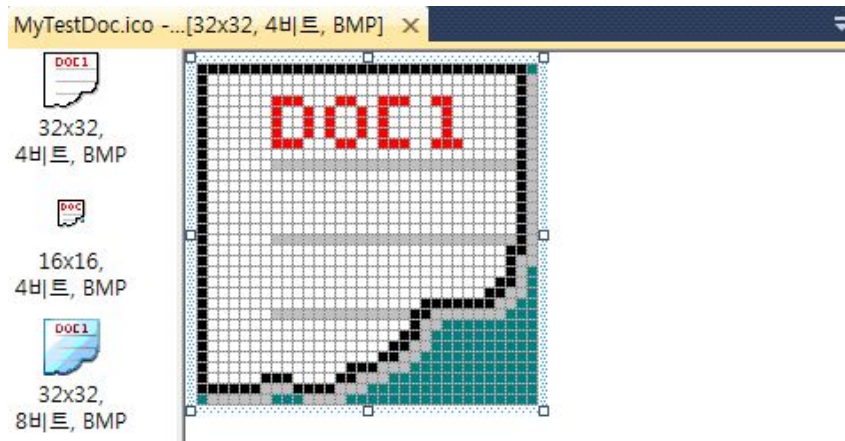


```
BEGIN_MESSAGE_MAP(CSDITestView, CView)
//{{AFX_MSG_MAP(CSDITestView)
// NOTE - the ClassWizard will add and remove mappings here
// DO NOT EDIT what you see in these blocks of code
//}}AFX_MSG_MAP
// Standard printing commands
ON_COMMAND(ID_FILE_PRINT, CView::OnFilePrint)
ON_COMMAND(ID_FILE_PRINT_DIRECT, CView::OnFilePrint)
ON_COMMAND(ID_FILE_PRINT_PREVIEW, CView::OnFilePrintPreview)
END_MESSAGE_MAP()
```

분할 윈도우

- 정적 분할 윈도우

- 서로 다른 뷰 클래스를 기반으로 뷰를 여러 개 생성
- 가로 16 x 세로 16 = 총 256개의 구획 생성 가능



분할 윈도우

- 정적 분할 윈도우 구현

```
#include "stdafx.h"
#include "Splitter.h"

#include "MainFrm.h"
#include "SplitterDoc.h"
#include "SplitterView.h"
#include "KeyInputView.h"

..
BOOL CMainFrame::OnCreateClient(LPCREATESTRUCT /*lpcs*/,
    CCreateContext* pContext)
{
    m_wndSplitter.CreateStatic(this, 1, 2);
    m_wndSplitter.CreateView(0, 0, RUNTIME_CLASS(CSplitterView),
        CSize(300, 300), pContext);
    m_wndSplitter.CreateView(0, 1, RUNTIME_CLASS(CKeyInputView),
        CSize(300, 300), pContext);
    SetActiveView((CView *)m_wndSplitter.GetPane(0, 0));
    return TRUE;
}
```


학습정리

- SDI 응용프로그램은 한 번에 하나의 문서만 열어서 작업할 수 있는 사용자 인터페이스를 제공한다.
- 도큐먼트 클래스는 주요 함수로 SetModifiedFlag(), UpdateAllViews(), OnNewDocument(), OnOpenDocument(), DeleteContents(), Serialize()를 제공한다.
- 뷰 클래스는 주요 함수로 OnDraw(), OnInitialUpdate(), OnUpdate()를 제공한다.
- 명령 라우팅은 명령 메시지나 명령 갱신 메시지를 여러 객체에서 처리할 수 있도록 전달하는 기법이다.
- 도큐먼트 한 개에 대해 뷰가 여러 개인 경우를 분할 윈도우라고 한다. 정적 분할 윈도우와 동적 분할 윈도우가 있으며 CSplitterWnd 클래스를 이용해 구현한다.