

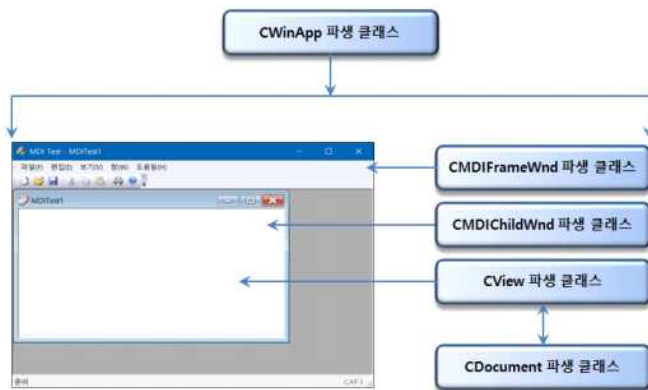
MFC란 윈도우 애플리케이션을 생성하기 위해 만들어진 C++ 클래스 라이브러리이다.

MFC는 헝가리안 표기법을 사용하며 클래스의 앞에는 모두 C가 붙는다.  
멤버변수는 m\_, 멤버함수는 M\_, 전역함수는 Afx라는 접두어가 붙는다.

SDI(Single Document Interface)와 MDI(Multiple Document Interface)가 있는데.

## MFC 프로그램의 구조

### MDI 애플리케이션의 구조

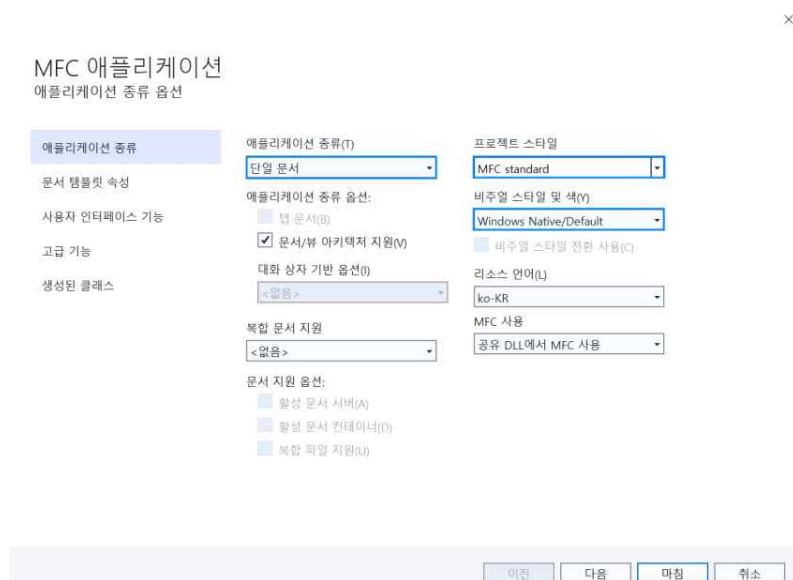


MDI애플리케이션(위 그림)

**CWinApp** 프로그램 전체의 진입점이며 초기화와 파일간의 연결을 설정.  
**CMDIFrameWnd** 메뉴, 툴바, 상태바 같은 프로그램의 틀(Frame)을 관리.  
**CMDIChildWnd** MDI 구조에서 문서 하나당의 자식 프레임을 할당.  
**CView** 실제 화면에 사용자의 입력을 처리. 시각적인 것을 보여주는 역할.  
**CDocument** 문서 데이터를 관리. 저장/불러오기 등 데이터 로직 처리 담당.

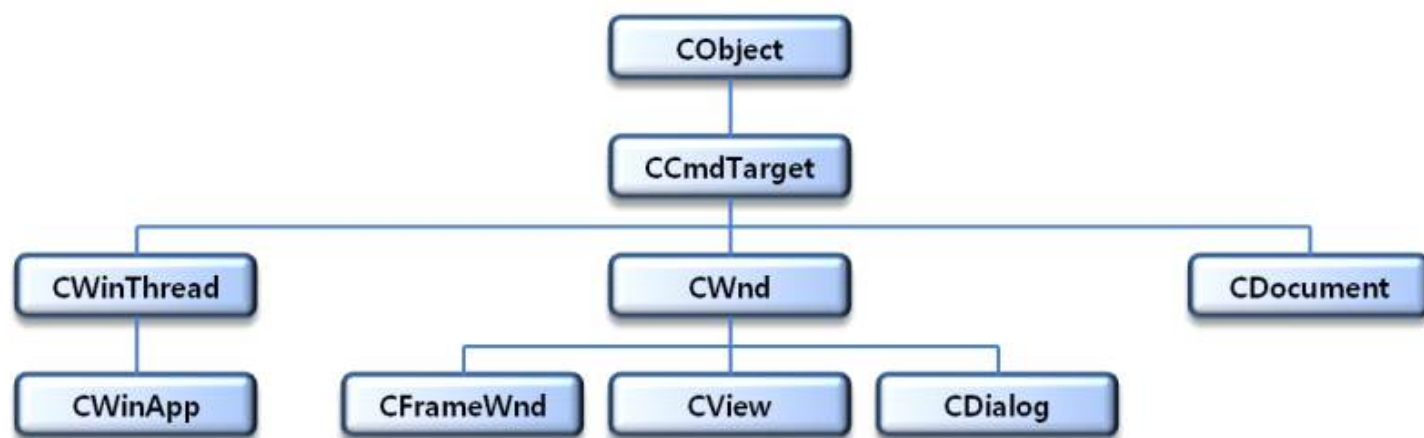
SDI는 여기서ChileWND에 해당하는 클래스가 없다. 하나의 창으로 관리되는 것.

사실 이번주 수업에는 Visual Studio 2022의 클래스 마법사라는 기능을 사용해본게 핵심같다.  
MFC 애플리케이션으로 프로젝트를 생성하려면



이러한 창이 뜨는데 단일 문서로 설정하면 SDI로 생성되며 프로젝트 스타일은 MFC 스탠다드 기본으로 세팅한다.  
만약 MDI로 설정하고 싶다면 애플리케이션 종류를 여러 문서라고 하여 MDI로 설정하면 된다.

아래 사진은 MFC 클래스의 기본 구조로 계층적 형태를 시각화한 것이다.



MFC는 이러한 상속 클래스들로 이루어져 있다.

지금부터는 실습 내용을 설명 하겠다.

MFC는 기본적으로 1주차에 실습 한 구조와 유사하다. 입력장치를 건드리면 메시지큐에 입력되고 메시지 루프에서 받아와서 해당하는 멤버함수를 순차적으로 실행 시킨다.

2주차에 한 실습은 이 순차적으로 실행되는 멤버함수를 수정했다.

1주차 코드에는 switch case 문으로 해당 것들을 구현 했다면 2주차에는 이미 만들어져있는 틀인 MFC를 불러와서 이미 만들어져있는 멤버함수를 수정한 것이다.

MFC standard로 프로젝트를 생성한후 실행해보면 그냥 하얀 창만 나온다.

우리는 입출력 신호를 받아 이 MFC 툴 실행창에 메시지를 띄워 볼것이다.

우선 우리는 비주얼 스튜디오 2022의 간단한 툴로 멤버 함수 형식을 가져와서 추가할 수 있다.

Ctrl + Shift + X 키로 클래스 마법사라는 창을 켤 수 있는데 이 클래스 마법사는

메시지 핸들러 함수를 자동으로 생성해준다. 이 함수를 생성하기 위해서는

함수 원형 추가, 메시지 맵에 매크로 추가, cpp에 함수 뼈대 생성 등을 해야하지만 이 마법사 기능을 사용해서 우리는 코드만 작성하면 되는 꼴이 된다.

사실상 2주차는 1주차에 한 내용을 MFC로 구현해본것이고.

여기서 중요한 것은 \_T 혹은 \_TEXT 라는 매크로인데 이 매크로가

```
m_strWindowSize.Format(_T("윈도우 크기는 너비 %d, 높이 %d입니다."), cx, cy);
```

```
m_strOutput = _T("왼쪽 마우스 버튼을 눌렀습니다");
```

이 문자열 앞에 괄호와 함께 붙는 이유는 문자 인코딩 방식(유니코드, 멀티바이트)에 상관없이 문자열을 다듬어주는 역할을 하기 때문. 이것 말고는 1주차의 switch case 문으로 구별하여 했던 것을 MFC 에서는 간단하게 구현된걸 불러와서 함수를 조금 수정해준다 정도이다.

예시 코드

원형

```
void CMy20221057PTestView::OnDraw(CDC* /*pDC*/)
```

```
{  
  
    CMy20221057PTestDoc* pDoc = GetDocument();  
    ASSERT_VALID(pDoc);  
    if (!pDoc)  
        return;  
  
    // TODO: 여기에 원시 데이터에 대한 그리기 코드를 추가합니다.  
}
```

수정

```
void CMy20221057P21View::OnDraw(CDC* pDC)
{
    CMy20221057P21Doc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    // TODO: 여기에 원시 데이터에 대한 그리기 코드를 추가합니다.
    //윈도우 크기를 나타내는 문자열을 윈도우 좌측 상단(10, 10)에 출력
    pDC->TextOut(10, 10, m_strWindowSize);
    CRect rect;
    GetClientRect(&rect);
    pDC->DrawText(m_strOutput, rect, DT_SINGLELINE | DT_CENTER | DT_VCENTER);
}
```

2주차 실습 리뷰 끝