

다양한 뷰클래스 II

(7주차)

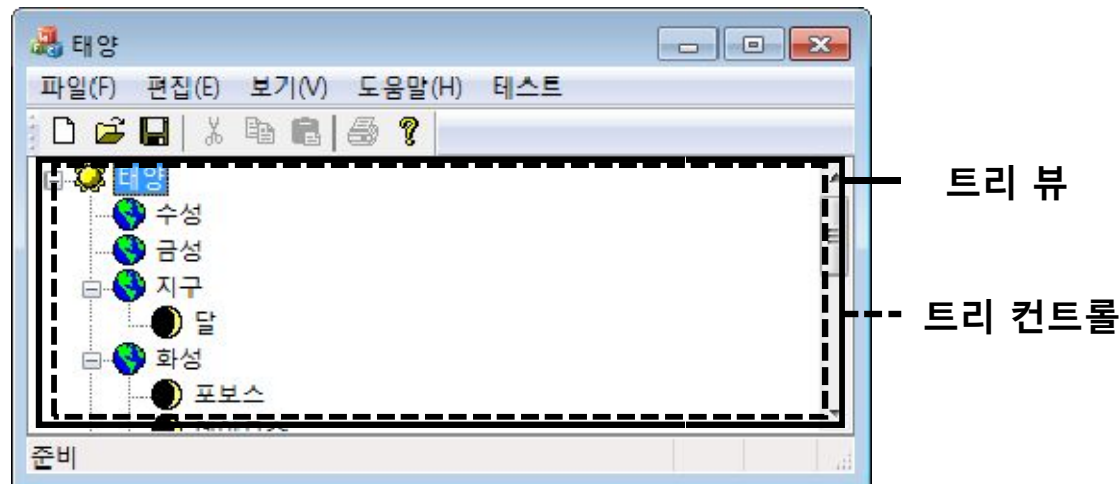
학습개요

- 학습 목표
 - MFC 뷰 클래스의 특징과 용도를 이해한다.
 - 다양한 뷰 클래스를 다루는 방법을 익힌다.
- 학습 내용
 - 트리 뷰
 - 스크롤 뷰
 - HTML 뷰
 - 실습

트리 뷰

- CTreeView 클래스

- 도큐먼트/뷰 구조 응용 프로그램에서 트리 컨트롤을 이용하여 뷰의 기능을 제공
 - 트리 컨트롤의 기능, 즉 여러 항목의 계층 관계를 이미지와 텍스트로 보여주는 기능을 뷰에서 손쉽게 활용 가능



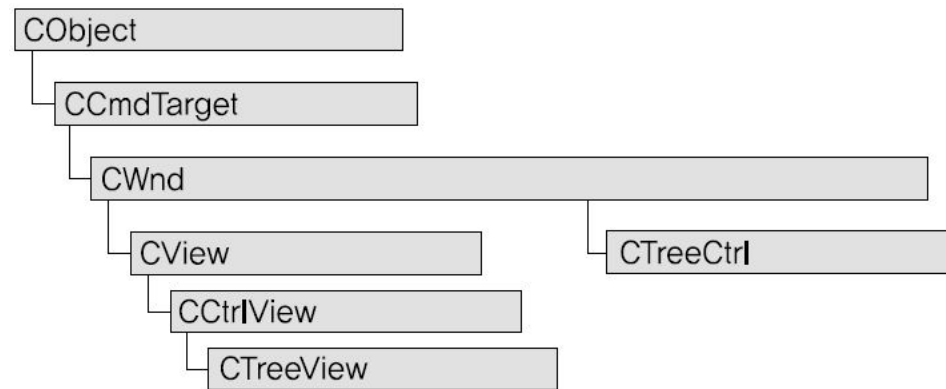
트리 뷰

CTreeView 클래스 사용 예

```
// 트리 컨트롤 객체에 대한 레퍼런스를 얻는다.  
CTreeCtrl& tree = GetTreeCtrl();  
  
// 레퍼런스를 통해 트리 컨트롤을 사용한다.  
tree.SetImageList(...);  
tree.InsertItem(...);  
...
```

트리 뷰

- MFC 클래스 계층도



트리 컨트롤

- 트리 컨트롤(=트리 뷰 컨트롤)
 - 이미지와 텍스트를 이용하여 정보를 계층 형태로 표시
- 용어
 - 항목(Item)
 - 트리 컨트롤에 표시되는 정보
 - 부모 항목(Parent Item)
 - 하위 항목을 하나 이상 가진 항목
 - 자식 항목(Child Item)
 - 부모 항목에 딸린 하위 항목
 - 루트 항목(Root Item)
 - 계층 구조에서 최상위 항목
 - 부모 항목을 갖지 않음

트리 컨트롤

- 트리 컨트롤 스타일

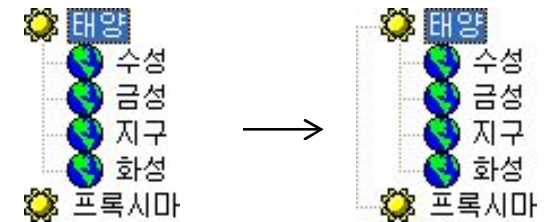
(a) Has Buttons



(b) Has Lines



(c) Lines At Root



(d) Edit Labels



(e) Check Boxes



(f) Full Row Select



(g) Tooltips



트리 컨트롤 클래스

- 트리 컨트롤 생성과 초기화

- ① CTreeCtrl 객체 선언 후 CTreeCtrl::Create() 함수로 트리 컨트롤 생성
- ② CImageList 객체 선언 후 CImageList::Create(), CImageList::Add() 등의 함수로 이미지 리스트 초기화
- ③ CTreeCtrl::SetImageList() 함수로 트리 컨트롤에서 사용할 이미지 리스트 설정
- ④ CTreeCtrl::InsertItem() 함수로 항목 추가

트리 컨트롤 클래스

• 예제 코드

```
int CTreeCtrlTest1View::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if(CView::OnCreate(lpCreateStruct) == -1)
        return -1;

    // ❶ 트리 컨트롤 생성
    m_tree.Create(WS_CHILD | WS_VISIBLE | WS_BORDER |
        TVS_HASBUTTONS | TVS_HASLINES | TVS_LINESATROOT,
        CRect(10, 10, 180, 180), this, IDC_TREE1);

    // ❷ 이미지 리스트 생성과 초기화
    CImageList il;
    il.Create(IDB_BITMAP1, 16, 1, RGB(255, 255, 255));

    // ❸ 이미지 리스트 설정
    m_tree.SetImageList(&il, TVSIL_NORMAL);
    il.Detach();
}
```



트리 컨트롤 클래스

• 예제 코드

```
// ❹ 항목 추가
/* 1-레벨 초기화 */
HTREEITEM hSun = m_tree.InsertItem(_T("태양"), 0, 0, TVI_ROOT, TVI_LAST);

/* 2-레벨 초기화 */
m_tree.InsertItem(_T("수성"), 1, 1, hSun, TVI_LAST);
m_tree.InsertItem(_T("금성"), 1, 1, hSun, TVI_LAST);
HTREEITEM hEarth = m_tree.InsertItem(_T("지구"), 1, 1, hSun, TVI_LAST);
HTREEITEM hMars = m_tree.InsertItem(_T("화성"), 1, 1, hSun, TVI_LAST);

/* 3-레벨 초기화 */
m_tree.InsertItem(_T("달"), 2, 2, hEarth, TVI_LAST);
m_tree.InsertItem(_T("포보스"), 2, 2, hMars, TVI_LAST);
m_tree.InsertItem(_T("데이모스"), 2, 2, hMars, TVI_LAST);

return 0;
}
```



트리 컨트롤 클래스

- 단계별 핵심 함수 - ① 트리 컨트롤 생성

```

BOOL CTreeCtrl::Create(DWORD dwStyle, const RECT& rect, CWnd* pParentWnd, UINT nID);
                        ①                ②                ③                ④

```

- dwStyle: 일반 윈도우 스타일(WS_*)과 트리 컨트롤 스타일(TVS_*)을 조합하여 컨트롤 스타일을 지정
- rect: 컨트롤의 크기와 위치
- pParentWnd: 부모 윈도우
- nID: 컨트롤 ID

- 단계별 핵심 함수- ② 이미지 리스트 생성과 초기화

트리 컨트롤 클래스

• 단계별 핵심 함수 - ③ 이미지 리스트 설정

```
CImageList* CTreeCtrl::SetImageList(CImageList * pImageList, int nImageListType);
```

- pImageList: 트리 컨트롤에서 사용할 이미지 리스트
- nImageListType: 이미지 리스트에 포함된 이미지의 용도를 나타내는 상수값
 - 항목을 나타내는 일반 이미지일 경우 → TVSIL_NORMAL
 - 사용자 정의 상태 이미지일 경우 → TVSIL_STATE

트리 컨트롤 클래스

- 단계별 핵심 함수 - ④ 항목 추가

```
HTREEITEM CTreeCtrl::InsertItem(LPCTSTR lpszItem, int nImage, int nSelectedImage,  
                                ①           ②           ③  
HTREEITEM hParent = TVI_ROOT, HTREEITEM hInsertAfter = TVI_LAST);  
          ④           ⑤
```

- lpszItem: 항목에 표시할 텍스트
- nImage: 항목에 표시할 이미지를 이미지 리스트에 포함된 이미지의 인덱스 값으로 지정
- nSelectedImage: 항목이 선택되면 표시할 이미지를 이미지 리스트에 포함된 이미지의 인덱스 값으로 지정

트리 컨트롤 클래스

- 단계별 핵심 함수 - ④ 항목 추가
 - hParent: 부모 항목을 나타내는 HTREEITEM 값. 루트 항목을 추가할 경우에는 TVI_ROOT 값을 사용
 - hInsertAfter: 항목을 추가할 위치를 나타냄
 - 보통은 다음 표의 값 중 하나를 사용.

값	의미
TVI_FIRST	맨 앞쪽에 추가한다.
TVI_LAST	맨 뒤쪽에 추가한다.
TVI_ROOT	루트 항목으로 추가한다.
TVI_SORT	철자순으로 정렬되게 추가한다.

- 특정 항목 다음 위치에 추가하고 싶을 경우에는 그 항목을 나타내는 HTREEITEM 값을 사용

트리 컨트롤 클래스

- 스타일 변경하기

```
BOOL CWnd::ModifyStyle(DWORD dwRemove, DWORD dwAdd, UINT nFlags = 0);
```

- dwRemove: 제거할 스타일
- dwAdd: 추가할 스타일
- nFlags: 컨트롤의 크기나 위치를 제어(대개 기본값 사용)

트리 컨트롤 클래스

- 선택 항목 알아내기

```
void CTreeCtrlTest2View::OnTestShowSelectedItem()  
{  
    HTREEITEM hItem = m_tree.GetSelectedItem();  
    if(hItem != NULL){  
        CString str = m_tree.GetItemText(hItem);  
        MessageBox(str);  
    }  
}
```


트리 컨트롤 클래스

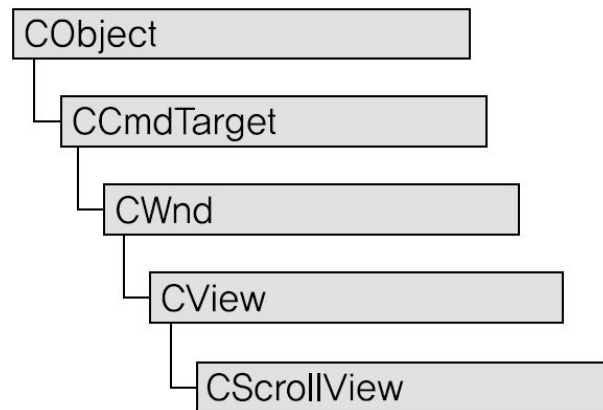
- 항목 추가와 삭제
 - 항목 추가
 - CTreeCtrl::InsertItem()
 - 항목 삭제
 - CTreeCtrl::DeleteItem()

```
void CTreeCtrlTest2View::OnTestDeleteSelectedItem()  
{  
    HTREEITEM hItem = m_tree.GetSelectedItem();  
    if(hItem != NULL){  
        m_tree.DeleteItem(hItem);  
    }  
}
```

실습

스크롤 뷰

- CScrollView 클래스
 - 스크롤 기능을 쉽게 구현 가능
- MFC 클래스 계층도



스크롤 뷰

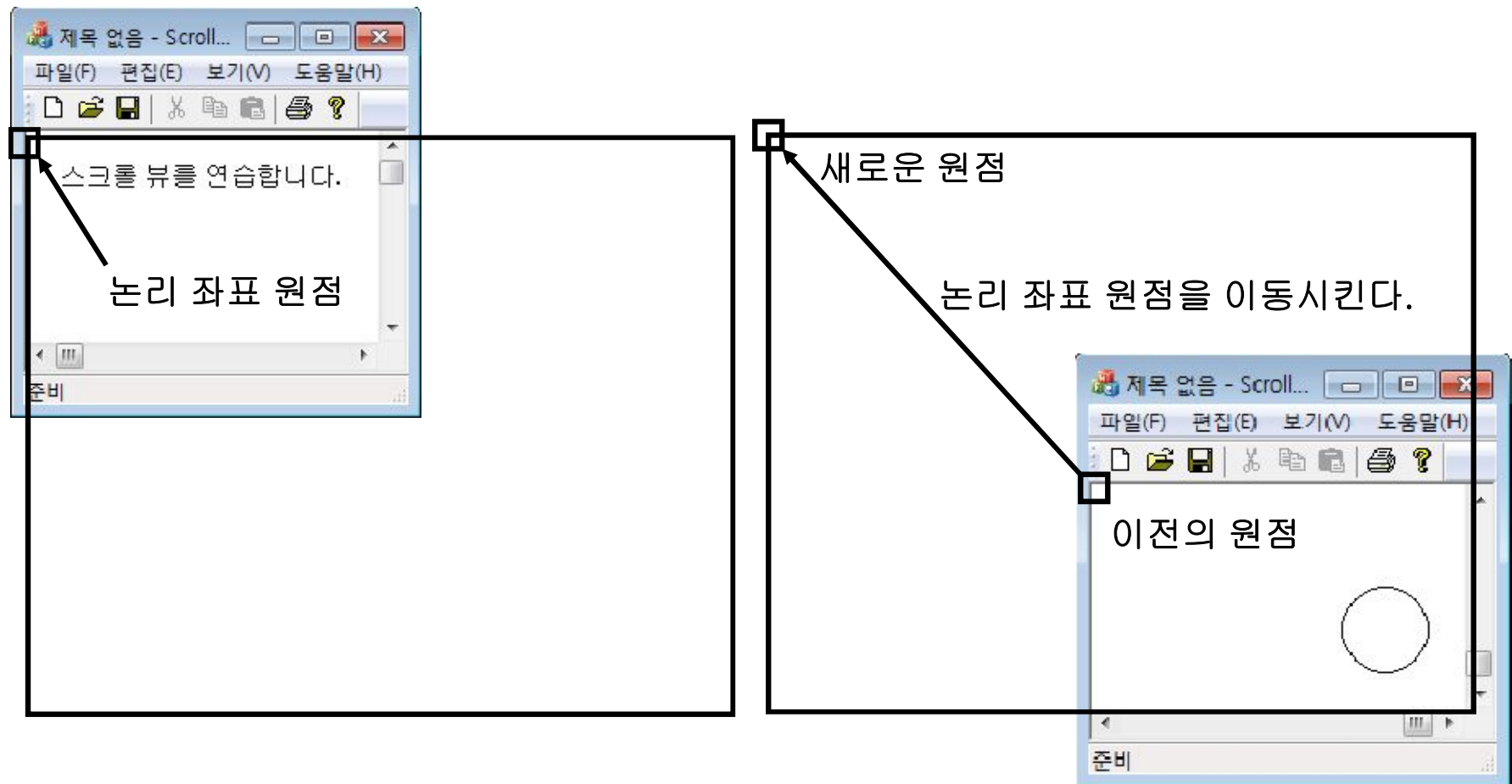
- 주요 함수

```
void CScrollView::SetScrollSizes(int nMapMode, SIZE sizeTotal,  
                                ①                ②  
const SIZE& sizePage = sizeDefault, const SIZE& sizeLine = sizeDefault);  
                                ③                ④
```

- nMapMode: 사용할 매핑 모드. MM_ISOTROPIC과 MM_ANISOTROPIC을 제외하고 모두 사용 가능
- sizeTotal: 작업 영역의 전체 크기(논리 단위)
- sizePage: 페이지 크기(논리 단위)
- sizeLine: 한 줄의 크기(논리 단위)

스크롤 뷰

- 스크롤 바의 위치에 따른 논리 좌표의 원점 이동



스크롤 뷰

- 스크롤 뷰 출력

```
void CScrollViewTestView::MyDraw()  
{  
    CClientDC dc(this);  
    OnPrepareDC(&dc); /* 매핑 모드 변경 & 논리 좌표 원점 보정 */  
    ...  
}
```

스크롤 뷰

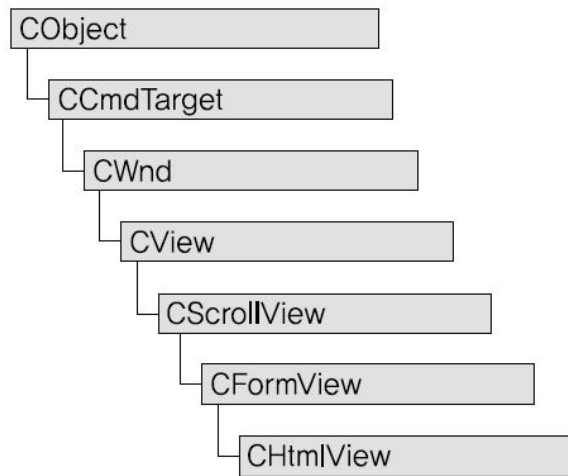
- 스크롤 최적화

```
void CScrollOptimizeView::OnDraw(CDC* pDC)
{
    CRect rect;
    pDC->GetClipBox(&rect);
    // rect 영역만 다시 그린다.
}
```

실습

HTML 뷰

- CHtmlView 클래스
 - Microsoft 인터넷 익스플로러의 핵심 요소인 웹브라우저 컨트롤 (WebBrowser Control)을 뷰에서 사용 가능
- MFC 클래스 계층도



HTML 뷰

- CHtmlView 클래스 멤버 함수

함수	기능
Navigate(), Navigate2()	URL로 지정한 웹 문서 또는 로컬 파일을 열어 서 표시한다.
GoHome()	인터넷 익스플로러에서 설정한 시작 페이지로 이동한다.
GoBack()	히스토리 목록에서 이전 항목으로 이동한다.
GoForward()	히스토리 목록에서 다음 항목으로 이동한다.
Refresh(), Refresh2()	현재 표시하고 있는 내용을 다시 로드한다.
Stop()	현재 작업을 중지한다.

HTML 뷰

- CHtmlView::Navigate2() 함수 사용 예

```
// 폴더와 파일을 표시한다.  
Navigate2(_T("C:WW"), NULL, NULL);  
// MS 워드 문서를 열어서 편집하거나 저장한다.  
Navigate2(_T("C:WWTempWWtest.doc"), NULL, NULL);  
// MS 엑셀 문서를 열어서 편집하거나 저장한다.  
Navigate2(_T("C:WWTempWWtest.xls"), NULL, NULL);  
// 텍스트 문서를 표시한다.  
Navigate2(_T("C:WWTempWWtest.txt"), NULL, NULL);  
// JPEG 포맷으로 저장된 그림을 표시한다.  
Navigate2(_T("C:WWTempWWtest.jpg"), NULL, NULL);
```

실습

학습정리

- CTreeView 클래스는 도큐먼트/뷰 구조 응용 프로그램에서 트리 컨트롤을 이용하여 뷰의 기능을 제공하며, 여러 항목의 계층 관계를 이미지와 텍스트로 보여주는 기능을 제공
- CScrollView 클래스는 자동화된 스크롤 기능을 제공하며, 스크롤 바의 위치에 따라 논리 좌표의 원점 이동 기능을 제공함
- CHtmlView 클래스를 이용하면 Microsoft 인터넷 익스플로러의 핵심 요소인 웹브라우저 컨트롤 (WebBrowser Control)을 기반으로 하는 뷰임