표준 컨트롤

(14주차)

unit 1. 들어가기

학습개요

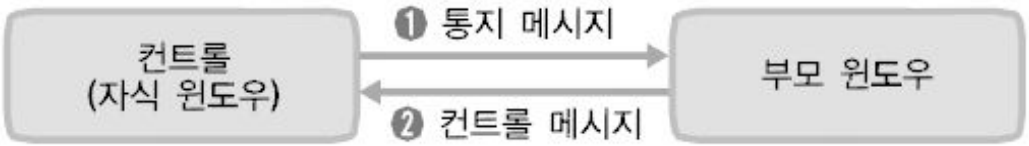
- 학습 목표
  - 컨트롤의 동작 원리를 이해한다.
  - 표준 컨트롤의 다양한 속성과 통지 메시지를 이해한다.
  - MFC의 컨트롤 클래스를 이용해 표준 컨트롤을 다루는 방법을 익힌다.
  - 서브클래싱과 메시지 반사 기법을 이해한다.

- 학습 내용
  - 컨트롤 기초
  - 고급 컨트롤 기법
  - 실습

컨트롤이란?

- 표준화된 형태와 특성을 가진 윈도우
- 사용자의 입력을 받거나 정보를 보여줌

컨트롤과 부모 윈도우의 통신



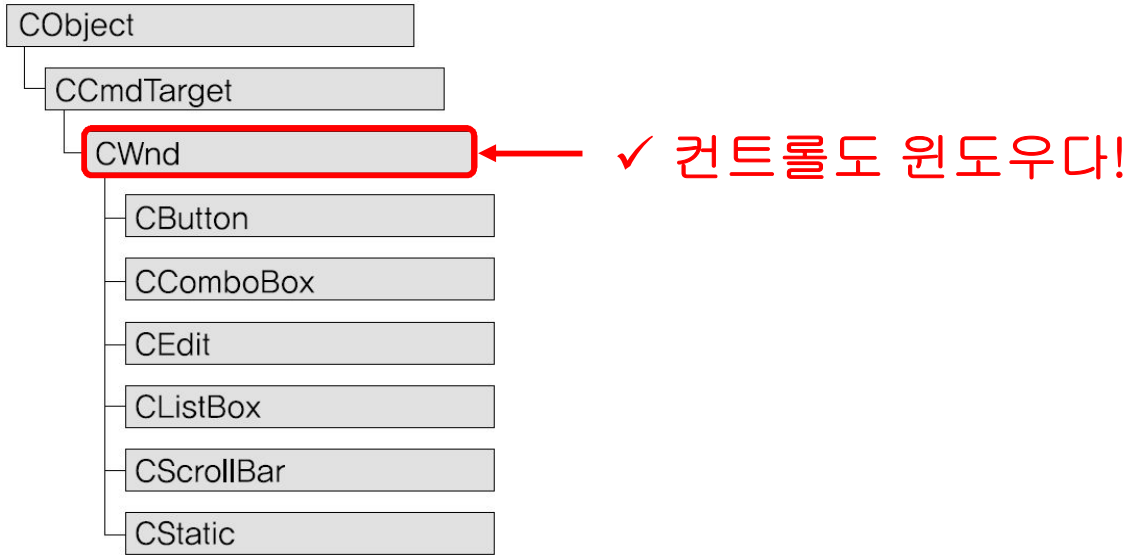
- 통지 메시지(Notification Message)
  - 컨트롤이 부모 윈도우에 보내는 메시지
  - 컨트롤의 상태가 변화되었음을 알림
  - 메모리 부족 등으로 인한 오류를 알림

- 컨트롤 메시지
  - 부모 윈도우가 컨트롤에 보내는 메시지
  - 컨트롤의 상태를 알아내거나 변경함
  - 컨트롤에 따라 보낼 수 있는 컨트롤 메시지의 종류가 다름

MFC 컨트롤 클래스

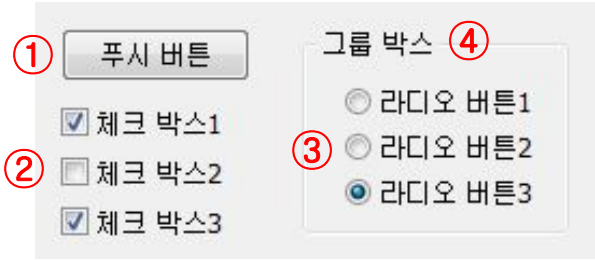
컨트롤	MFC 클래스	도구 상자 아이콘
버튼	CButton	 (푸시 버튼),  (체크 박스),  (라디오 버튼),  (그룹 박스)
정적	CStatic	 (텍스트),  (프레임, 직사각형, 아이콘, 비트맵, 메타파일)
편집	CEdit	 (편집)
리스트 박스	CListBox	 (리스트 박스)
콤보 박스	CComboBox	 (콤보 박스)
스크롤 바	CScrollBar	 (수평 스크롤바),  (수직 스크롤바)

MFC 클래스 계층도

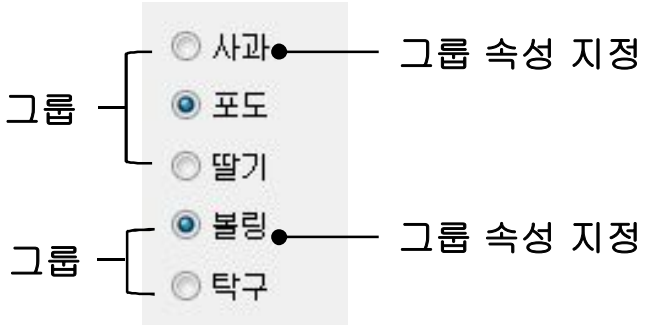

CObject
CCmdTarget
CWnd
✓ 컨트롤도 윈도우다!
CButton
CComboBox
CEdit
CListBox
CScrollBar
CStatic

버튼 컨트롤

- 버튼 컨트롤 종류



- 라디오 버튼 그룹



버튼 컨트롤

- 컨트롤 생성 방법
  ① 코드를 실행하여 만들기
  ② 대화상자 리소스에 포함하여 만들기

- 컨트롤 생성 - 첫 번째 방법

```
CButton m_button; // C++ 버튼 객체 생성
m_button.Create(_T("누르세요"), WS_CHILD | WS_VISIBLE | BS_PUSHBUTTON,
    CRect(100, 100, 200, 130), this, 101); // 푸시 버튼 생성
```

버튼 컨트롤

- CButton::Create() 함수

```
BOOL CButton::Create(LPCTSTR lpszCaption, DWORD dwStyle, const RECT& rect,
                                  ①                ②                ③
                     CWnd*pParentWnd, UINT nID);
                            ④              ⑤
```

- lpszCaption - 컨트롤에 표시할 문자열
- dwStyle - 컨트롤 스타일(윈도우 스타일 + 버튼 스타일)
- rect - 컨트롤 크기와 위치
- pParentWnd - 부모 윈도우 지정
- UINT nID - 컨트롤 ID

버튼 컨트롤

- 버튼 스타일(일부)

스타일	의미
BS_PUSHBUTTON	푸시 버튼
BS_DEFPUSHBUTTON	디폴트 푸시 버튼 (일반 윈도우에서는 차이가 없으나, 대화상자에서 사용하면 [Enter] 키를 누를 때 눌려진다.)
BS_CHECKBOX	체크 박스
BS_AUTOCHECKBOX	자동 체크 박스 (클릭하면 자동으로 체크 표시가 On/Off 된다.)
BS_3STATE	3상태 체크 박스
BS_AUTO3STATE	자동 3상태 체크 박스 (클릭하면 자동으로 체크 표시가 On/Grayed/Off 된다.)
BS_RADIOBUTTON	라디오 버튼
BS_AUTORADIOBUTTON	자동 라디오 버튼 (클릭하면 자동으로 선택과 선택 해제가 이루어진다.)
BS_GROUPBOX	그룹 박스

버튼 컨트롤

- 통지 메시지 처리하기 ⇨ 메시지 핸들러 작성

```
ON_BN_CLICKED(101, OnButtonClicked)
...
void CButton1View::OnButtonClicked()
{
    MessageBox(_T("버튼을 눌렀습니다. “));
}
```

- 컨트롤 메시지 보내기 ⇨ 멤버 함수 호출

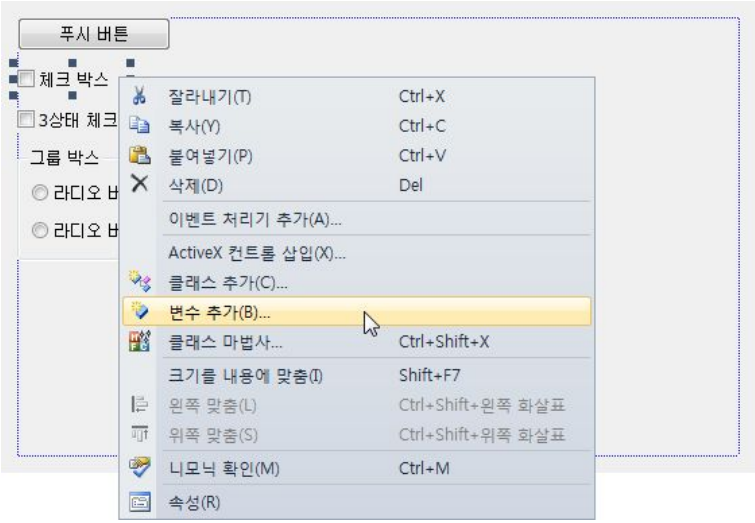
```
m_button.SetCheck(BST_CHECKED);
```

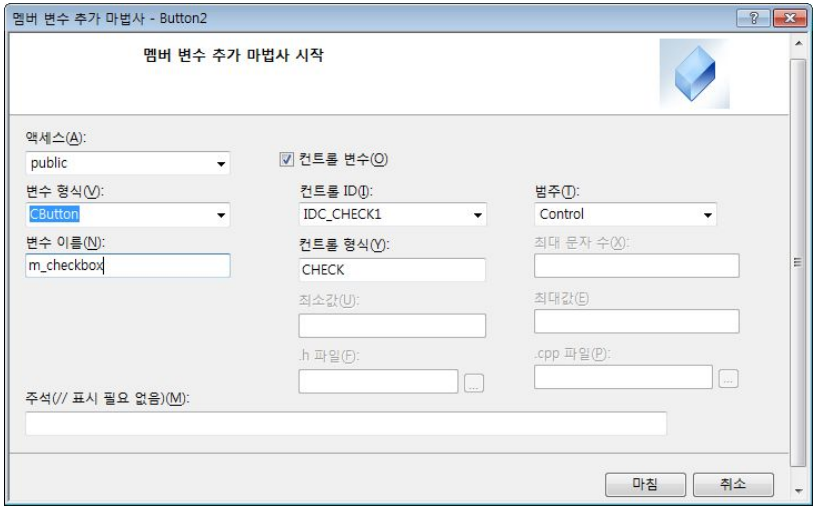
버튼 컨트롤

- 컨트롤 생성 - 두 번째 방법
  - 대화상자 리소스에 컨트롤 추가
    - 대화상자가 생성될 때 컨트롤도 자동으로 생성
  - 컨트롤 변수 생성
    - 컨트롤 자체를 나타내는 변수(=컨트롤 변수)를 생성하고 이를 이용하여 컨트롤을 조작

버튼 컨트롤

- 컨트롤 변수 생성





버튼 컨트롤

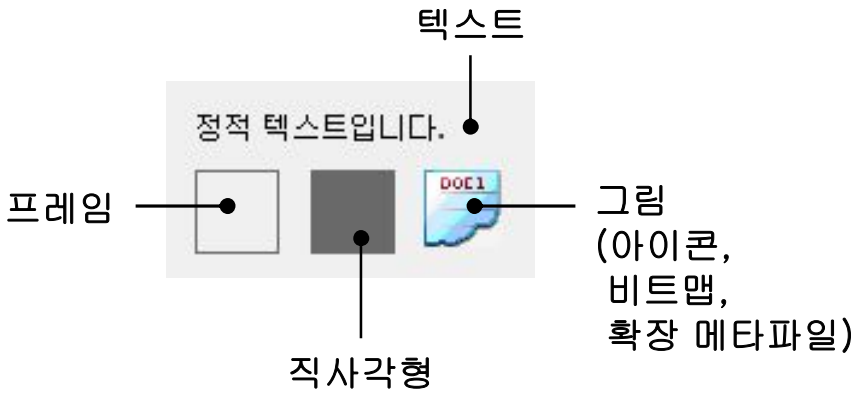
- 컨트롤 변수 생성

```
class CButton2View : public CFormView
{
    ...
public:
    CButton m_checkbox;
    CButton m_3state;
    CButton m_radio1;
    CButton m_radio2;
};
```

```
...
void CButton2View::DoDataExchange(CDataExchange* pDX)
{
    CFormView::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_CHECK1, m_checkbox);
    DDX_Control(pDX, IDC_CHECK2, m_3state);
    DDX_Control(pDX, IDC_RADIO1, m_radio1);
    DDX_Control(pDX, IDC_RADIO2, m_radio2);
}
...
```

정적 컨트롤

- 정적 컨트롤 종류
  - 텍스트
  - 프레임(색이 채워지지 않은 직사각형)
  - 직사각형(색이 채워진 직사각형)
  - 아이콘
  - 비트맵
  - 확장 메타파일(Enhanced Metafile)

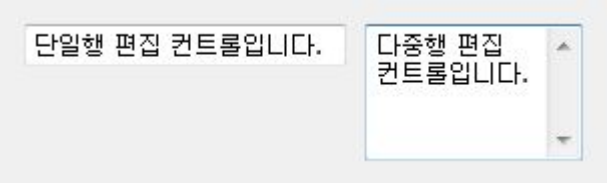


정적 컨트롤

- 정적 컨트롤 통지 메시지
  - 일반적으로 통지 메시지를 발생시지지 않음
  - 단, SS_NOTIFY 스타일을 설정하면 통지 메시지 발생
    - STN_CLICKED(클릭), STN_DBLCLK(더블 클릭), STN_DISABLE(비활성화), STN_ENABLE(활성화)

편집 컨트롤

- 편집 컨트롤 종류



- 편집 컨트롤 스타일

스타일/속성 항목	기능
ES_AUTOHSCROLL/Auto HScroll	줄 끝에 도달하면 자동으로 수평 스크롤한다.
ES_AUTOVSCROLL/Auto Vscroll	줄 끝에 도달하면 자동으로 수직 스크롤한다.
ES_CENTER/Align Text: Centered	텍스트를 가운데 정렬한다.
ES_LEFT/Align text: Left	텍스트를 왼쪽 정렬한다.
ES_LOWERCASE/Lowercase	입력된 모든 문자를 소문자로 변환한다.

편집 컨트롤

- 편집 컨트롤 스타일

스타일/속성 항목	기능
ES_MULTILINE/Multiline	다중행 편집 컨트롤로 동작한다.
ES_NOHIDESEL/No hide selection	컨트롤이 키보드 포커스를 잃어도 선택된 텍스트가 반전 상태를 계속 유지한다.
ES_NUMBER/Number	숫자만 입력할 수 있다.
ES_PASSWORD/Password	입력된 글자를 *로 표시한다. (단일행 편집 컨트롤에만 사용 가능)
ES_READONLY/Read only	텍스트를 읽기만 할 수 있다.
ES_RIGHT/Align Text: Right	텍스트를 오른쪽 정렬한다.
ES_UPPERCASE/Uppercase	입력된 모든 문자를 대문자로 변환한다.
ES_WANTRETURN/Want return	이 스타일을 지정하지 않으면 대화상자에서 [Enter] 키를 눌러도 줄바꿈이 되지 않는다. [Ctrl]+[Enter] 키를 이용하면 이 스타일과 무관하게 줄바꿈을 할 수 있다. (다중행 편집 컨트롤에만 사용 가능)

편집 컨트롤

- 편집 컨트롤 통지 메시지

통지 메시지	의미
EN_CHANGE	사용자가 내용을 변경하면 화면에 컨트롤을 다시 그리는데, 그 후에 이 메시지가 발생한다.
EN_ERRSPACE	메모리가 부족하다.
EN_HSCROLL	사용자가 편집 컨트롤의 수평 스크롤 바를 클릭했다.
EN_KILLFOCUS	키보드 포커스를 잃었다.
EN_MAXTEXT	더 이상 문자를 입력할 수 없다. CEdit::SetLimitText() 함수로 문자의 개수를 제한한 경우나 ES_AUTOHSCROLL, ES_AUTOVSCROLL 스타일을 지정하지 않은 상태에서 줄 끝까지 입력할 때 발생한다.
EN_SETFOCUS	키보드 포커스를 얻었다.
EN_UPDATE	사용자가 내용을 변경하면 화면에 컨트롤을 다시 그리는데, 그 전에 이 메시지가 발생한다.
EN_VSCROLL	사용자가 편집 컨트롤의 수직 스크롤 바를 클릭했다.

편집 컨트롤

- 텍스트를 변경하거나 입력된 텍스트를 알아내기

```
m_edit.SetWindowText(_T("초깃값")); // 편집 컨트롤의 텍스트를 변경한다.
CString str;
m_edit.GetWindowText(str); // 편집 컨트롤에 입력된 텍스트를 얻는다.
```

- 입력 가능한 텍스트의 길이를 제한하기

```
m_edit.SetLimitText(10); // 열 글자로 제한한다.
```

- 각종 편집 작업 - 삭제, 잘라내기, 복사, 붙여넣기, 실행 취소

```
m_edit.Clear(); // 현재 선택된 텍스트를 삭제한다.
m_edit.Cut(); // 텍스트를 잘라내서 클립보드에 저장한다.
m_edit.Copy(); // 텍스트를 복사해서 클립보드에 저장한다.
m_edit.Paste(); // 클립보드에 저장된 내용을 붙여 넣는다.
m_edit.Undo(); // 이전에 실행한 동작을 취소한다.
```

편집 컨트롤

- 텍스트 선택과 치환하기

```
❶ m_edit.SelSel(5, 7); // 여섯 번째 글자부터 시작해 글자 두 개가 선택된다.
❷ m_edit.ReplaceSel(_T("ABC")); // 현재 선택된 부분을 새로운 텍스트로 치환한다.
```

리스트 박스 컨트롤

- 리스트 박스 컨트롤 종류

- 리스트 박스 컨트롤 스타일

스타일/속성 대화상자 항목	기능
LBS_DISABLENOSCROLL / Disable no scroll	표시할 항목의 개수가 적어도 수직 스크롤 바가 사라지지 않는다.
LBS_EXTENDEDSEL / Selection: Extended	[Shift], [Ctrl] 키와 마우스 클릭을 이용한 다중 선택이 가능하다.
LBS_HASSTRINGS / Has strings	LBS_OWNERDRAW* 스타일을 지정하지 않았을 때의 기본 스타일로, 컨트롤이 문자열을 저장하고 관리한다.

리스트 박스 컨트롤

- 리스트 박스 컨트롤 스타일

스타일/속성 대화상자 항목	기능
LBS_MULTICOLUMN / Multicolumn	여러 열(Column)로 구성된 리스트 박스를 생성하며, 항목이 많으면 수평 스크롤이 가능하다.
LBS_MULTIPLESEL / Selection: Multiple	마우스 클릭을 이용한 다중 선택이 가능하다.
LBS_NODATA / No Data	항목 데이터를 컨트롤이 아닌 부모 윈도우가 유지하며, 필요할 때마다 부모 윈도우가 직접 그린다. 항목이 1,000개 이상일 경우에 사용을 권장한다.
LBS_NOINTEGRALHEIGHT / No integral height	리스트 박스의 높이가 응용 프로그램이 지정한 크기로 유지된다. 이 스타일을 지정하면 항목의 일부가 잘려서 보이지 않는 경우가 발생할 수 있다.
LBS_NOREDRAW / No Redraw	항목이 변해도 리스트 박스 컨트롤을 다시 그리지 않는다.
LBS_NOSEL / Selection: None	항목을 볼 수 있지만 선택할 수는 없다.

리스트 박스 컨트롤

- 리스트 박스 컨트롤 스타일

스타일/속성 대화상자 항목	기능
LBS_NOTIFY / Notify	사용자가 항목을 클릭하거나 더블 클릭하면 부모 윈도우에게 통지 메시지를 보낸다.
LBS_OWNERDRAWFIXED / Owner Draw: Fixed	부모 윈도우가 리스트 박스 항목을 직접 그리되 항목의 높이가 일정한 경우다.
LBS_OWNERDRAWVARIABLE / Owner Draw: Variable	부모 윈도우가 리스트 박스 항목을 직접 그리되 항목의 높이가 일정하지 않은 경우다.
LBS_SORT / Sort	항목이 문자열인 경우 정렬하여 표시한다.
LBS_STANDARD / Notify, Sort, Border, Vertical scrollbar	LBS_NOTIFY, LBS_SORT, WS_VSCROLL, WS_BORDER 스타일의 조합이다.
LBS_USETABSTOPS / Use Tabstops	이 스타일을 지정하면 항목 문자열에 포함된 탭 문자('\t')를 제대로 처리할 수 있다.
LBS_WANTKEYBOARDINPUT /Want Key Input	리스트 박스 컨트롤이 키보드 포커스를 가진 상태에서 사용자가 키를 누르면 부모 윈도우가 이를 감지하여 특별한 처리를 할 수 있다.

리스트 박스 컨트롤

- 리스트 박스 컨트롤 통지 메시지

통지 메시지	의미
LBN_DBLCLK	사용자가 항목을 더블 클릭했다.
LBN_SELCHANGE	사용자가 선택을 변경했다.
LBN_SELCANCEL	사용자가 선택을 취소했다.
LBN_SETFOCUS	키보드 포커스를 얻었다.
LBN_KILLFOCUS	키보드 포커스를 잃었다.
LBN_ERRSPACE	메모리가 부족하다.

  - LBN_DBLCLK, LBN_SELCHANGE, LBN_SELCANCEL은 LBS_NOTIFY 스타일을 설정해야 발생한다.

리스트 박스 컨트롤

- 항목을 추가하거나 삭제하기

```
m_list.AddString(_T("사과")); // 문자열 항목을 추가한다.
m_list.DeleteString(3); // 네 번째 항목을 삭제한다.
```

- 항목을 선택된 상태로 만들기

```
/* 단일 선택 리스트 박스 컨트롤인 경우* /
m_list.SetCurSel(2); // 세 번째 항목을 선택한다.

/* 다중 선택 리스트 박스 컨트롤인 경우* /
m_list.SetSel(2); // 세 번째 항목을 선택한다.
m_list.SetSel(3, FALSE); // 네 번째 항목을 선택 해제한다.
```

리스트 박스 컨트롤

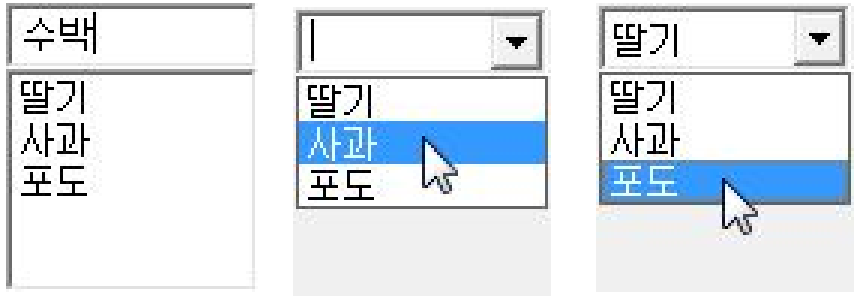
- 선택된 항목을 알아내기

```
/* 단일 선택 리스트 박스 컨트롤인 경우* /
int nIndex = m_list.GetCurSel();
if(nIndex != LB_ERR){
    CString str;
    m_list.GetText(nIndex, str);
}

/* 다중 선택 리스트 박스 컨트롤인 경우* /
int nIndex = m_list.GetCaretIndex();
if(nIndex != LB_ERR){
    CString str;
    m_list.GetText(nIndex, str);
}
```

콤보 박스 컨트롤

- 콤보 박스 컨트롤 종류



콤보 박스 컨트롤

- 콤보 박스 컨트롤 스타일

스타일/속성 대화상자 항목	기능
CBS_AUTOHSCROLL/Auto	= ES_AUTOHSCROLL
CBS_DISABLENOSCROLL/Disable No Scroll	= LBS_DISABLENOSCROLL
CBS_DROPDOWN/Type: Dropdown	드랍다운 스타일을 지정한다.
CBS_DROPDOWNLIST/Type: Drop List	드랍다운 리스트 스타일을 지정한다.
CBS_HASSTRINGS/Has Strings	= LBS_HASSTRINGS
CBS_LOWERCASE/Lowercase	= ES_LOWERCASE
CBS_NOINTEGRALHEIGHT/No Integral Height	= LBS_NOINTEGRALHEIGHT
CBS_OWNERDRAWFIXED/Owner Draw: Fixed	= LBS_OWNERDRAWFIXED
CBS_OWNERDRAWVARIABLE/Owner Draw: Variable	= LBS_OWNERDRAWVARIABLE
CBS_SIMPLE/Type: Simple	단순 스타일을 지정한다.
CBS_SORT/Sort	= LBS_SORT
CBS_UPPERCASE/Uppercase	= ES_UPPERCASE

콤보 박스 컨트롤

- 콤보 박스 컨트롤 통지 메시지

통지 메시지	의미	스타일		
		단순	드랍다운	드랍다운 리스트
CBN_CLOSEUP	리스트 박스가 닫혔다.		●	●
CBN_DBLCLK	사용자가 항목을 더블 클릭했다.		●	
CBN_DROPDOWN	리스트 박스가 열리기 직전이다.		●	●
CBN_EDITCHANGE	= EN_EDITCHANGE	●	●	
CBN_EDITUPDATE	= EN_EDITUPDATE	●	●	
CBN_ERRSPACE	메모리가 부족하다.	●	●	●
CBN_KILLFOCUS	키보드 포커스를 잃었다.	●	●	●
CBN_SELCHANGE	= LBN_SELCHANGE	●	●	●
CBN_SELENDCANCEL	= LBN_SELENDCANCEL		●	●
CBN_SELENDOK	사용자가 항목을 선택했다.	●	●	●
CBN_SETFOCUS	키보드 포커스를 얻었다.	●	●	●

콤보 박스 컨트롤

- 항목을 추가하거나 삭제하기

```
m_combo.AddString(_T("사과")); // 문자열 항목을 추가한다.
m_combo.DeleteString(3); // 네 번째 항목을 삭제한다
```

- 선택된 항목을 알아내기

```
int nIndex = m_combo.GetCursel();
if(nIndex != CB_ERR){
    CString str;
    m_combo.GetLBText(nIndex, str);
}
```

콤보 박스 컨트롤

- 입력 가능한 텍스트의 길이를 제한하기

```
m_combo.LimitText(10); // 열 글자로 제한한다.
```

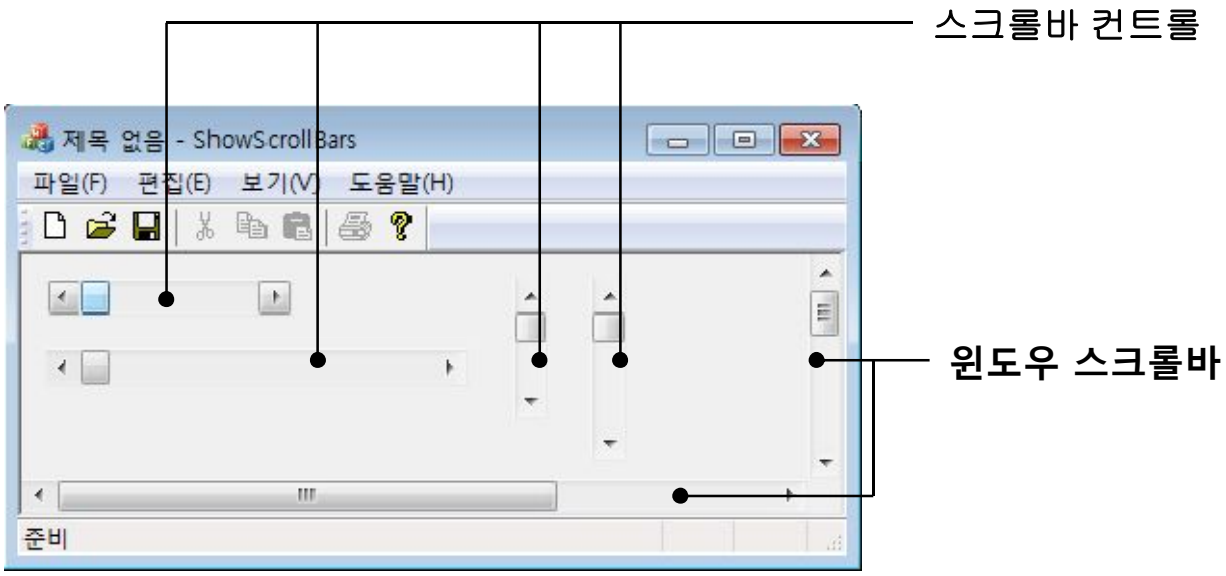
- 각종 편집 작업 - 삭제, 잘라내기, 복사, 붙여넣기

```
m_combo.Clear(); // 현재 선택된 텍스트를 삭제한다.
m_combo.Cut(); // 텍스트를 잘라내서 클립보드에 저장한다.
m_combo.Copy(); // 텍스트를 복사해서 클립보드에 저장한다.
m_combo.Paste(); // 클립보드에 저장된 내용을 붙여넣는다.
```

스크롤 바 컨트롤

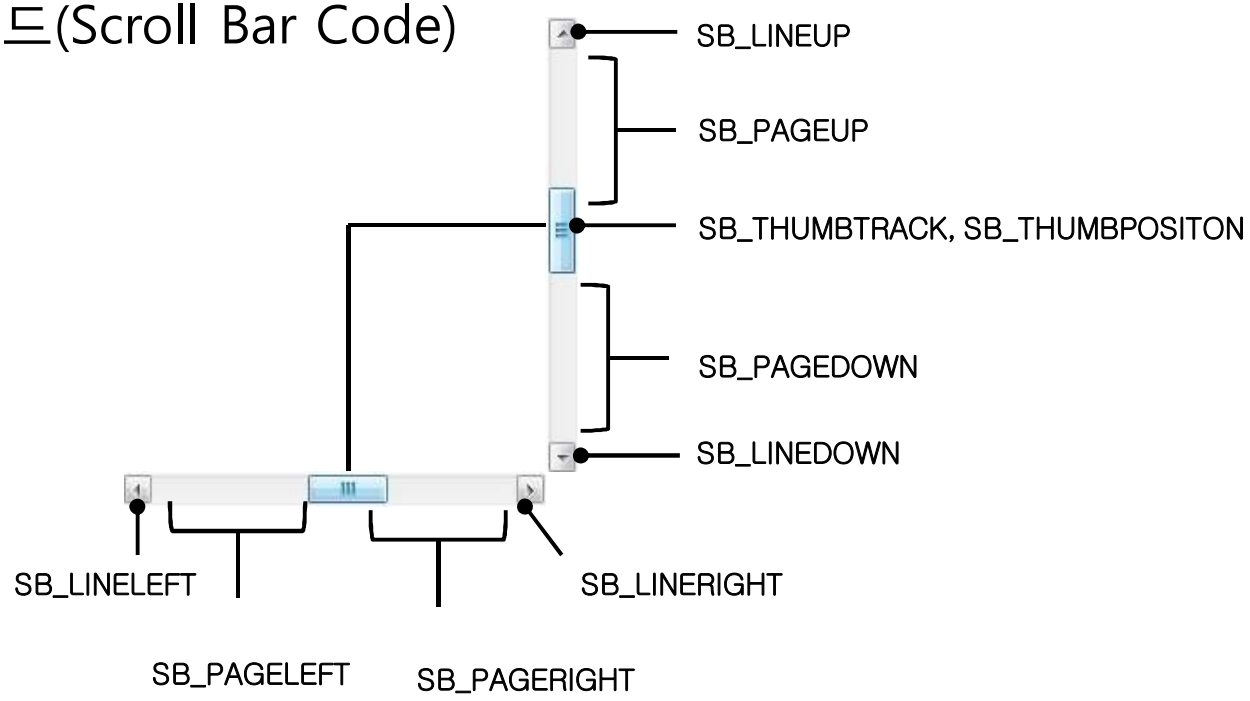
- 스크롤 바 컨트롤
  - 화면의 일정 영역을 스크롤하거나 정해진 범위의 값을 변경하는 용도로 사용

- 스크롤 바의 종류



스크롤 바 컨트롤

- 스크롤 바 코드
  - 사용자가 스크롤 바를 조작하면 WM_HSCROLL 또는 WM_VSCROLL 메시지 발생
    - 메시지와 더불어 사용자의 구체적인 행위를 알 수 있는 정보가 전달됨 ☞ 스크롤 바 코드(Scroll Bar Code)



스크롤 바 컨트롤

- 스크롤 바 코드

스크롤 바 코드	응용 프로그램이 해야 할 작업
SB_LINELEFT, SB_LINERIGHT	왼쪽 또는 오른쪽으로 한 줄만큼 데이터를 스크롤한다.
SB_PAGELEFT, SB_PAGERIGHT	왼쪽 또는 오른쪽으로 한 페이지만큼 데이터를 스크롤한다.
SB_LINEUP, SB_LINEDOWN	위쪽 또는 아래쪽으로 한 줄만큼 데이터를 스크롤한다.
SB_PAGEUP, SB_PAGEDOWN	위쪽 또는 아래쪽으로 한 페이지만큼 데이터를 스크롤한다.
SB_THUMBTRACK, SB_THUMBPOSITION	스크롤 박스의 현재 위치로 이동한다.

스크롤 바 컨트롤

- WM_HSCROLL/WM_VSCROLL 메시지 핸들러

```
void OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar);
                     ①          ②                  ③
void OnVScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar);
```

  - nSBCode
    - 스크롤 바 코드(앞의 표 참조)
  - nPos
    - 스크롤 박스의 위치(스크롤 바 코드가 SB_THUMBTRACK 또는 SB_THUMBPOSITION인 경우에만 의미가 있음)
  - pScrollBar
    - CScrollBar 객체(스크롤 바 컨트롤에서 발생한 메시지인 경우)
      또는 NULL(윈도우 스크롤 바에서 발생한 메시지인 경우)

스크롤 바 컨트롤

- CScrollBar 멤버 함수(일부)

멤버 함수	기능
SetScrollRange()	스크롤 박스 위치의 최소값과 최대값을 설정한다.
SetScrollPos()	스크롤 박스의 현재 위치를 설정한다.
GetScrollPos()	스크롤 박스의 현재 위치를 얻는다.

실습

고급 컨트롤 기법

- 서브 클래싱
  - 기존 윈도우나 컨트롤의 형태 또는 동작 변경

- 메시지 반사
  - 스스로 통지 메시지를 처리하는 독립적인 컨트롤 클래스를 제작

서브 클래싱

- 서브 클래싱
  - 윈도우(정확하게는 윈도우 프로시저)로 가는 메시지를 중간에서 붙잡아 처리하는 기법

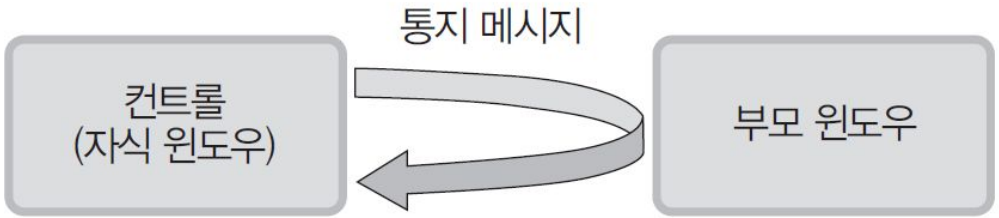
  - 메시지의 종류에 따라, 처리가 끝난 후 원래의 윈도우 프로시저에 전달할 수도 있고 전달하지 않을 수도 있음

서브 클래싱

- 서브 클래싱 구현 순서
  ① 기존의 클래스를 기반으로 새로운 윈도우나 컨트롤 클래스를 생성
  ② 메시지 핸들러를 추가해서 기능을 재정의
  ③ CWnd::SubclassWindow() 또는 CWnd::SubclassDlgItem() 함수를 호출

메시지 반사

- 메시지 반사
  - 부모 윈도우가 처리하지 않는 통지 메시지를 컨트롤 자신이 처리



  - 부모 윈도우가 통지 메시지를 처리하지 않는 경우에만 컨트롤이 자신의 통지 메시지 처리 가능

메시지 반사

- 메시지 반사 구현 순서
  ① 기존 컨트롤 클래스를 상속받아 새로운 클래스를 만들고 자신의 통지 메시지를 처리하는 함수를 작성
  ② 새로 만든 클래스를 이용하여 컨트롤 생성

실습

학습평가 1

- 다음 중 컨트롤에 대해 설명한 것 중 잘못된 것을 고르십시오.

1. 컨트롤은 표준화된 형태와 특성을 가진 윈도우로 사용자의 입력을 받거나 정보를 보여준다.
2. MFC 표준 컨트롤에는 버튼, 정적, 편집, 리스트박스, 콤보박스, 스크롤바가 있다.
3. 컨트롤은 상태 변화를 알리기 위해 통지 메시지를 부모 윈도우에 보내며, 부모 윈도우는 컨트롤의 상태를 알아내거나 변경하기 위해 컨트롤 메시지를 컨트롤에 보낸다.
4. 통지메시지를 처리하기 위해 컨트롤 객체의 멤버 함수를 호출하며, 컨트롤 메시지를 보내기 위해 부모 윈도우의 메시지 핸들러를 사용한다.

- 정답: 4
- 해설: 통지메시지를 처리하기 위해 부모 윈도우에 메시지 핸들러를 정의하며, 컨트롤 메시지를 보내기 위해 컨트롤 객체의 멤버 함수를 호출한다.

학습평가 2

- 다음 중 컨트롤의 서브 클래싱 기법에 대해 설명한 것 중 잘못된 것을 고르십시오.

1. 서브 클래싱은 기존 윈도우나 컨트롤의 형태 또는 동작 변경하기 위해 사용하는 기법입니다.
2. 기존의 클래스를 기반으로 새로운 윈도우나 컨트롤 클래스의 생성이 필요합니다.
3. 메시지 핸들러를 추가해서 기능을 재정의합니다.
4. CWnd::SubclassDlgItem() 함수를 호출을 통해 정적 서브클래싱을 할 수 있습니다.

- 정답: 4
- 해설: CWnd::SubclassDlgItem() 함수를 호출을 통해 동적 서브클래싱을 할 수 있습니다.

학습평가 3

- 다음 중 컨트롤의 메시지 반사 기법에 대해 설명한 것 중 잘못된 것을 고르십시오.

1. 메시지 반사는 부모 윈도우가 처리하지 않는 통지 메시지를 컨트롤 자신이 처리할 수 있도록 하는 기법입니다.
2. 부모 윈도우가 통지 메시지를 처리하지 않는 경우에도 컨트롤이 자신의 통지 메시지 처리를 할 수 있습니다.
3. 기존 컨트롤 클래스를 상속받아 새로운 클래스를 만들고 자신의 통지 메시지를 처리하는 함수를 정의하고, 새로 만든 클래스를 이용하여 컨트롤 생성합니다.

- 정답: 2
- 해설: 부모 윈도우가 통지 메시지를 처리하지 않는 경우에만 컨트롤이 자신의 통지 메시지 처리가 가능합니다.

학습정리

- 컨트롤은 표준화된 형태와 특성을 가진 윈도우로 사용자의 입력을 받거나 정보를 보여준다.
- 컨트롤은 상태 변화를 알리기 위해 통지 메시지를 부모 윈도우에 보내며, 부모 윈도우는 컨트롤의 상태를 알아내거나 변경하기 위해 컨트롤 메시지를 컨트롤에 보낸다.
- 통지메시지를 처리하기 위해 부모 윈도우에 메시지 핸들러를 정의하며, 컨트롤 메시지를 보내기 위해 컨트롤 객체의 멤버 함수를 호출한다.
- 컨트롤은 코드를 이용한 생성 방법과 대화상자 리소스와 컨트롤 변수 추가를 통한 생성 방법이 있다.
- 표준 컨트롤에는 버튼, 정적, 편집, 리스트박스, 콤보박스, 스크롤바가 있다.
- 기존 윈도우나 컨트롤의 형태 또는 동작 변경하기 위해 서브 클래싱 기법을 사용한다.
- 스스로 통지 메시지를 처리하는 독립적인 컨트롤 클래스를 만들기 위해 메시지 반사 기법을 사용한다.