

사용자 인터페이스

(11주차)

학습개요

- 학습 목표

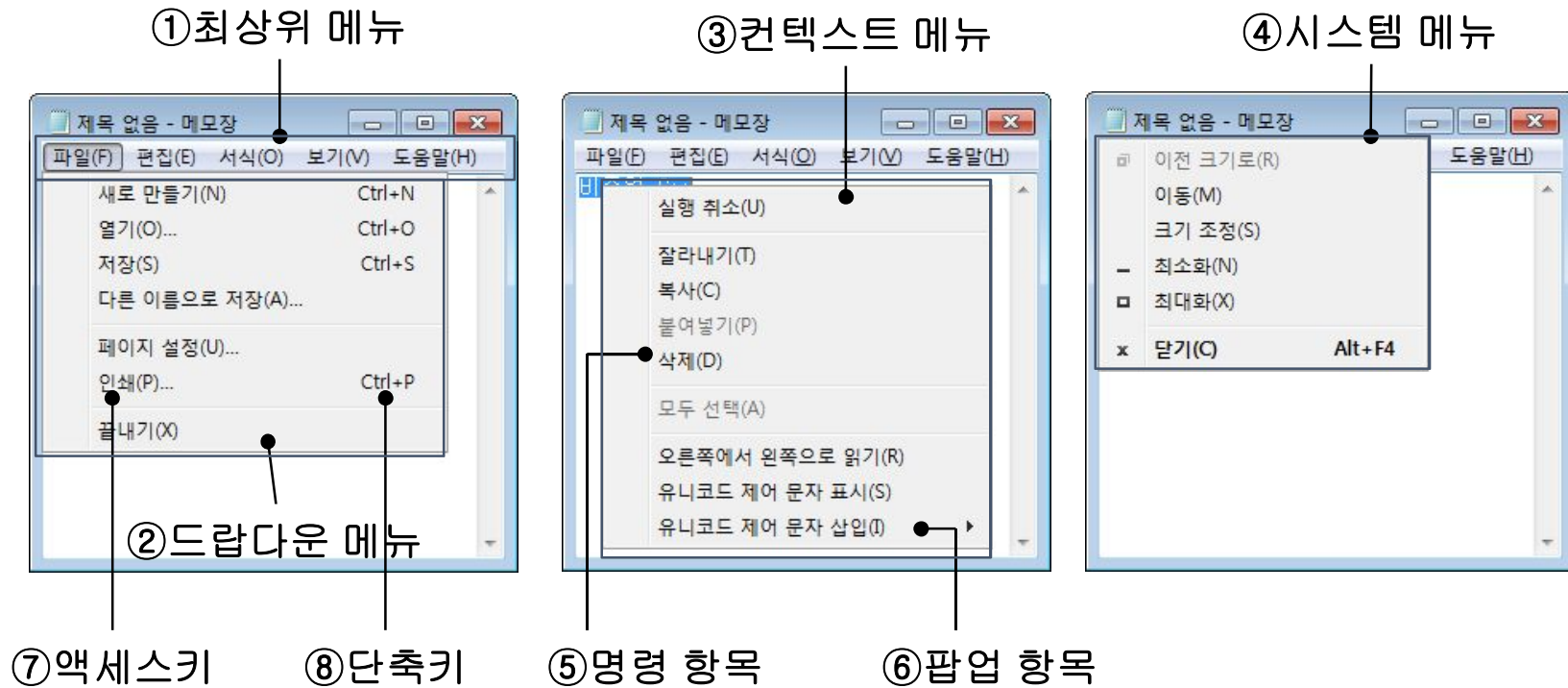
- 메뉴 명령을 처리하고 메뉴 항목을 적절하게 갱신하는 기법을 익힌다.
- 컨텍스트 메뉴와 시스템 메뉴를 다루는 방법을 익힌다.
- 툴바를 생성하고 사용하는 방법을 익힌다.
- 상태바를 생성하고 사용하는 방법을 익힌다.

- 학습 내용

- 메뉴
- 툴바
- 상태바
- 실습

메뉴 기초

• 메뉴 용어



메뉴 기초

■ 메뉴 항목의 종류

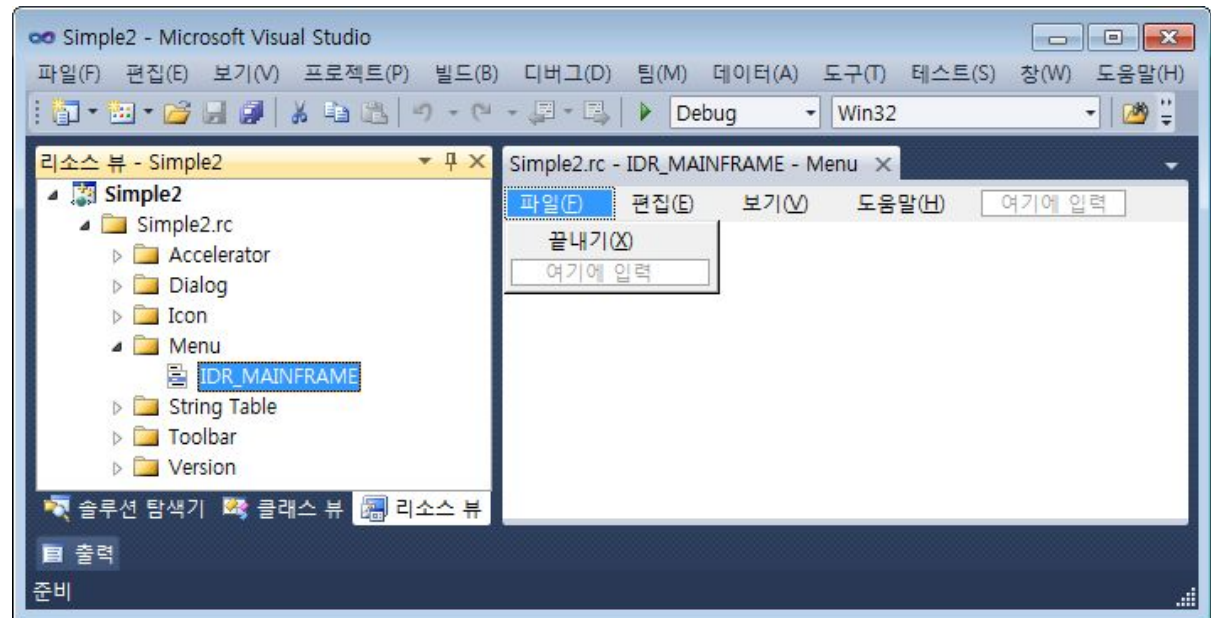
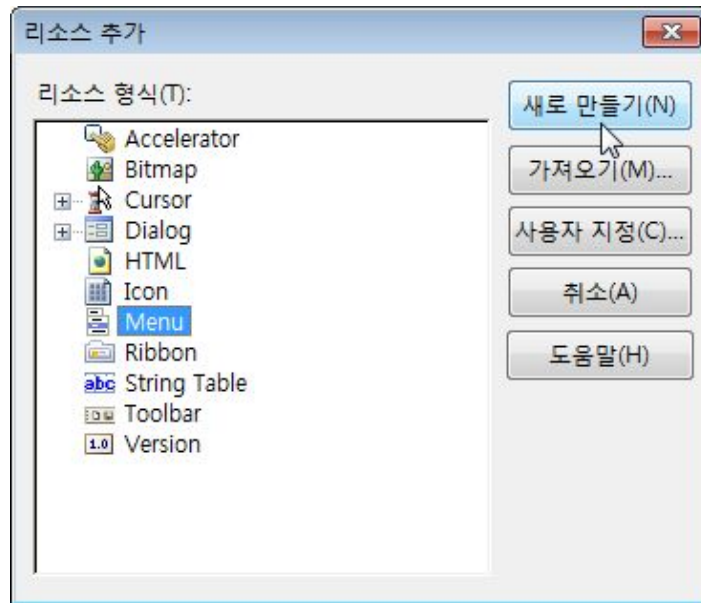
용어	의미	
명령 항목(⑤)	명령을 수행하는 메뉴 항목. 선택시 WM_COMMAND 메시지가 발생	
팝업 항목(⑥)	하위 메뉴를 표시하는 메뉴 항목. 선택시 WM_COMMAND 메시지가 발생하지 않음	

메뉴 생성하기

- 윈도우 응용 프로그램에서 메뉴 생성 방법 두 가지
 - 리소스를 이용한 메뉴 생성
 - 메뉴 리소스 정의 → 실행 파일에 포함 → 프로그램 실행 중에 불러옴
 - 코드를 이용한 메뉴 생성
 - 코드를 실행하여 메뉴 생성 → 윈도우에 붙여서 사용

리소스를 이용한 메뉴 생성

- 메뉴 리소스 추가



리소스를 이용한 메뉴 생성

- 응용 프로그램 마법사가 생성한 코드

```
BOOL CSimple2App::InitInstance()
{
    ...
    CMainFrame* pFrame = new CMainFrame;
    if(!pFrame)
        return FALSE;
    m_pMainWnd = pFrame;

    pFrame->LoadFrame(IDR_MAINFRAME,
        WS_OVERLAPPEDWINDOW | FWS_ADDTOTITLE, NULL,
        NULL);

    pFrame->ShowWindow(SW_SHOW);
    pFrame->UpdateWindow();

    return TRUE;
}
```

리소스를 이용한 메뉴 생성

■ 메뉴 항목 속성

속성	의미
HELP	최상위 메뉴 항목에만 설정할 수 있다. 이 속성을 설정하면 윈도우의 오른쪽 끝에 메뉴 항목이 표시된다. 요즘은 잘 쓰이지 않으나 과거에는 Help 메뉴 항목에 주로 설정했다.
ID	메뉴 항목을 구분하는 번호다. 일반적으로 ID_메뉴이름_항목이름 형태로 만든다. 예) ID_EDIT_CUT
Prompt	MFC로 작성한 프로그램에서만 사용할 수 있다. 현재 선택된 메뉴 항목에 관한 설명을 상태바나 툴바에 표시하기 위한 속성이다. 'Wn'을 기준으로 앞쪽 문자열은 상태바에 표시되고, 뒤쪽 문자열은 동일한 기능을 가진 툴바 항목 위에 툴팁(Tooltip)으로 표시된다. 예) 선택 영역을 잘라내어 클립보드에 넣습니다. Wn잘라내기

리소스를 이용한 메뉴 생성

■ 메뉴 항목 속성

속성	의미	
Separator	메뉴 항목을 구분하는 가로줄이 표시된다.	
Break	일반적으로 메뉴 항목은 하나의 열(Column)에 표시되지만 항목 수가 많으면 두 개 이상의 열에 표시되게 할 수 있다. Break 속성으로 Column 또는 Bar를 선택하면 다음 열에 메뉴 항목이 표시된다. Column과 Bar 속성은 기본적으로 기능이 같지만 Bar 속성을 선택하면 열 구분선(세로줄)이 보인다는 차이가 있다	
Right Justify	Help 속성과 기능이 동일하다. 최상위 메뉴 항목에만 설정할 수 있으며, 이 속성을 설정하면 윈도우의 오른쪽 끝에 메뉴 항목이 표시된다.	

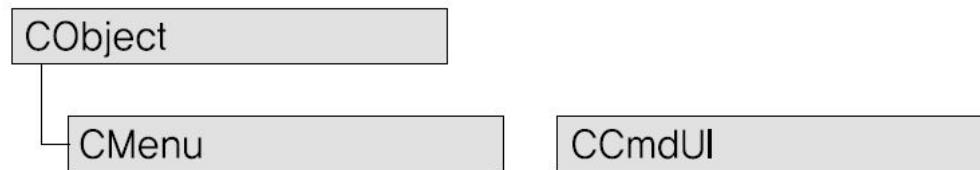
리소스를 이용한 메뉴 생성

■ 메뉴 항목 속성

속성	의미	
Right Order	Caption 문자열이 오른쪽에서 왼쪽 방향으로 표시된다. 아랍어나 히브리어를 위한 기능이다.	
Caption	화면에 표시되는 문자열이다. 액세스키를 지정하려면 해당 문자 앞에 & 기호를 사용한다. 단축키는 'Wt'를 삽입하여 단축키를 나타내는 문자열이 탭 위치에 정렬되게 하는 것이 좋다. 예) 잘라내기(&T)WtCtrl+X	
Checked	메뉴 항목의 왼쪽에 체크 표시를 한다.	
Enabled	이 값이 False면 메뉴 항목은 표시되지만 사용하지는 못한다.	
Grayed	메뉴 항목이 흐리게 표시되어 현재 사용할 수 없음을 나타낸다.	
Pop-up	이 속성을 설정하면 명령 항목이 아닌 팝업 항목이 된다(즉, 항목 선택 시 하위 메뉴가 펼쳐진다). 최상위 메뉴는 대개 Pop-up 속성을 가진다.	

코드를 이용한 메뉴 생성

- MFC 클래스 계층도



- CMenu
 - 메뉴를 코드 수준에서 제어할 수 있는 기능 제공
- CCmdUI
 - CObject의 파생 클래스가 아닌 독립된 클래스
 - 메뉴와 툴바 및 상태바에 필요한 기능 지원

코드를 이용한 메뉴 생성

- 프로그램 실행 중 메뉴 전체 생성하기 (1/2)

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    ...
    CMenu menuMain; // 메뉴 객체 생성
    menuMain.CreateMenu(); // 최상위 메뉴 생성

    CMenu menuPopup; // 메뉴 객체 생성
    menuPopup.CreatePopupMenu(); // 팝업 메뉴 생성

    // 팝업 메뉴에 메뉴 항목 세 개 추가
    menuPopup.AppendMenu(MF_STRING, 201, _T("빨간색(&R)"));
    menuPopup.AppendMenu(MF_STRING, 202, _T("초록색(&G)"));
    menuPopup.AppendMenu(MF_STRING, 203, _T("파란색(&B)"));

    // 최상위 메뉴에 팝업 메뉴 추가
    menuMain.AppendMenu(MF_POPUP, (UINT)menuPopup.Detach(),
        _T("색상(&C)"));
```

코드를 이용한 메뉴 생성

- 프로그램 실행 중 메뉴 전체 생성하기 (2/2)

```
SetMenu(&menuMain); // 메뉴를 윈도우에 붙임  
menuMain.Detach(); // 메뉴 객체와 메뉴를 분리  
return 0;  
}
```

- 코드에서 사용한 CMenu 클래스 멤버 함수
 - CMenu::CreateMenu()
 - CMenu::CreatePopupMenu()
 - CMenu::AppendMenu()

코드를 이용한 메뉴 생성

- 기존 메뉴에 추가하는 방법 (1/2)

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
```

```
...
```

```
// (1) '항목2'의 하위 메뉴를 생성한다.
```

```
CMenu Popup1; // 메뉴 객체 생성
```

```
Popup1.CreatePopupMenu(); // 팝업 메뉴 생성
```

```
Popup1.AppendMenu(MF_STRING, 301, _T("1"));
```

```
Popup1.AppendMenu(MF_STRING, 302, _T("2"));
```

```
Popup1.AppendMenu(MF_STRING, 303, _T("3"));
```

```
Popup1.AppendMenu(MF_STRING, 304, _T("4"));
```

코드를 이용한 메뉴 생성

- 기존 메뉴에 추가하는 방법 (2/2)

```
// (2) '테스트' 메뉴를 생성한다.  
CMenu Popup2; // 메뉴 객체 생성  
Popup2.CreatePopupMenu(); // 팝업 메뉴 생성  
Popup2.AppendMenu(MF_STRING | MF_CHECKED, 201, _T("항목&1"));  
Popup2.AppendMenu(MF_POPUP, (UINT)Popup1.Detach(), _T("항목&2"));  
Popup2.AppendMenu(MF_STRING, 203, _T("항목&3"));  
  
// (3) '테스트' 메뉴를 최상위 메뉴에 붙인다.  
CMenu *pTopLevel = GetMenu(); // 최상위 메뉴의 포인터를 얻는다.  
pTopLevel->AppendMenu(MF_POPUP, (UINT)Popup2.Detach(), _T("테스트(&T)"));  
  
return 0;  
}
```

메뉴 명령 처리

- 메뉴 명령 처리 과정
 - ① 명령 항목을 마우스나 키보드로 선택
 - ② WM_COMMAND 메시지 발생
 - ③ WM_COMMAND 메시지 핸들러에서 메뉴 명령 처리
- MFC에서는 각각의 명령 항목별로 처리 함수(명령 핸들러)를 따로 작성할 수 있게 해준다.
- 명령 라우팅
 - 명령 핸들러를 작성하는 위치에 관계없이 처리됨

메뉴 명령 처리

- 메뉴 명령 처리 코드 (1/3)

```
class CChildView : public CWnd
{
    // 생성입니다.
public:
    CChildView();

    // 특성입니다.
public:
    COLORREF m_color;
```

```
CChildView::CChildView()
{
    m_color = RGB(255, 0, 0);
}
```

메뉴 명령 처리

- 메뉴 명령 처리 코드 (2/3)

```
void CChildView::OnColorRed()
{
    m_color = RGB(255, 0, 0);
    Invalidate();
}

void CChildView::OnColorGreen()
{
    m_color = RGB(0, 255, 0);
    Invalidate();
}

void CChildView::OnColorBlue()
{
    m_color = RGB(0, 0, 255);
    Invalidate();
}
```

메뉴 명령 처리

- 메뉴 명령 처리 코드 (3/3)

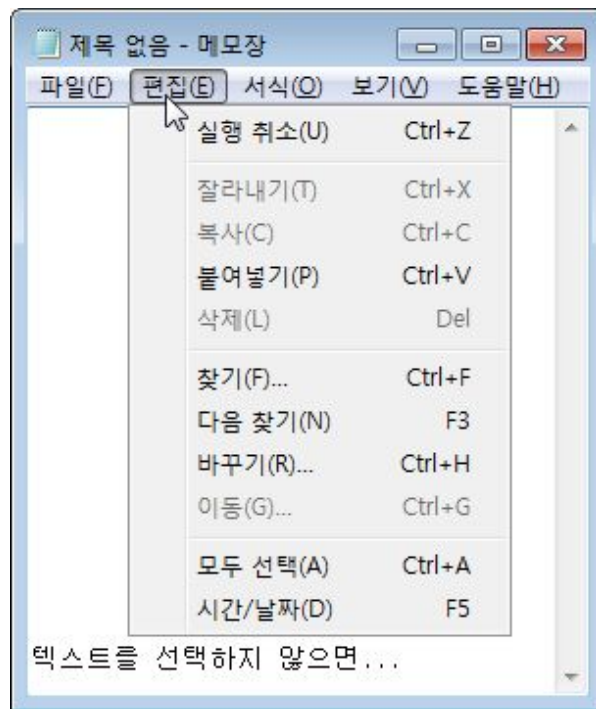
```
void CChildView::OnPaint()
{
    CPaintDC dc(this);

    CFont font;
    font.CreatePointFont(300, _T("궁서"));
    dc.SelectObject(&font);
    dc.SetTextColor(m_color);

    CRect rect;
    GetClientRect(&rect);
    CString str = _T("메뉴 테스트");
    dc.DrawText(str, &rect, DT_CENTER | DT_VCENTER | DT_SINGLELINE);
}
```

메뉴 항목 갱신

- 메뉴 항목 갱신의 예



메뉴 항목 갱신

- 명령 갱신 핸들러

- 명령 핸들러와 마찬가지로 각 메뉴 항목의 상태를 갱신하는 함수(명령 갱신 핸들러)를 정의한다.
- 명령 갱신 핸들러 작성시, 메뉴가 열리기 전에 MFC에서 자동으로 명령 갱신 핸들러를 호출하여 메뉴 항목의 상태를 바꾸어 준다.

메뉴 항목 갱신

■ CCmdUI 클래스 멤버 함수

멤버 함수	기능	사용 예
Enable()	활성화 상태 변경	pCmdUI->Enable(bDrawMode);
SetCheck()	체크 상태 변경	pCmdUI->SetCheck(m_color == RGB(255, 0, 0));
SetRadio()	라디오 표시 상태 변경	pCmdUI->SetRadio(m_color == RGB(255, 0, 0));
SetText()	문자열 변경	pCmdUI->SetText(_T("Light On"));

메뉴 항목 갱신

- 메뉴 항목 갱신 코드

```
void CChildView::OnUpdateColorRed(CCmdUI *pCmdUI)
{
    pCmdUI->SetCheck(m_color == RGB(255, 0, 0));
}
```

```
void CChildView::OnUpdateColorGreen(CCmdUI *pCmdUI)
{
    pCmdUI->SetCheck(m_color == RGB(0, 255, 0));
}
```

```
void CChildView::OnUpdateColorBlue(CCmdUI *pCmdUI)
{
    pCmdUI->SetCheck(m_color == RGB(0, 0, 255));
}
```

컨텍스트 메뉴

- WM_CONTEXTMENU 메시지 발생 상황
 - 클라이언트 영역이나 비클라이언트 영역에서 마우스 오른쪽 버튼을 누른 경우
 - [Shift] + [F10] 키를 누른 경우
 - 가상 키코드 VK_APPS에 해당하는 키를 누른 경우

- WM_CONTEXTMENU 메시지 핸들러

```
afx_msg void CWnd::OnContextMenu(CWnd* pWnd, CPoint pos);
```

- pWnd - 마우스 커서 아래쪽에 있는 윈도우
- pos - 마우스 커서의 위치(스크린 좌표)

컨텍스트 메뉴

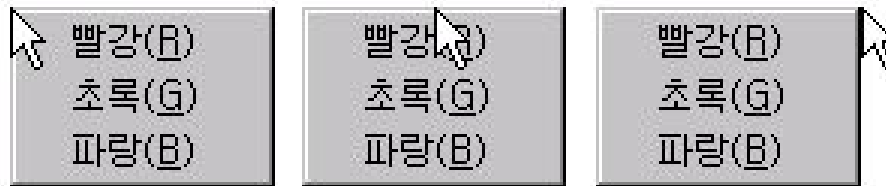
- CMenu::TrackPopupMenu() 함수

```
BOOL TrackPopupMenu(UINT nFlags, int x, int y, CWnd* pWnd, LPCRECT lpRect = 0);
```

① ② ③ ④ ⑤

- nFlags

- TPM_LEFTALIGN, TPM_CENTERALIGN, TPM_RIGHTALIGN



- TPM_LEFTBUTTON, TPM_RIGHTBUTTON

컨텍스트 메뉴

- x, y
 - 컨텍스트 메뉴가 표시될 위치(스크린 좌표)
- pWnd
 - 컨텍스트 메뉴에서 발생한 WM_COMMAND 메시지를 처리할 윈도우
- lpRect
 - 마우스 버튼을 클릭해도 컨텍스트 메뉴가 닫히지 않는 직사각형 영역(스크린 좌표)

컨텍스트 메뉴

- 컨텍스트 메뉴 구현 코드

```
void CChildView::OnContextMenu(CWnd* /*pWnd*/, CPoint point)
{
    CMenu menu;
    menu.LoadMenu(IDR_MAINFRAME);
    CMenu *pMenu = menu.GetSubMenu(4);
    pMenu->TrackPopupMenu(
        TPM_LEFTALIGN|TPM_RIGHTBUTTON,
        point.x, point.y, AfxGetMainWnd());
}
```

시스템 메뉴

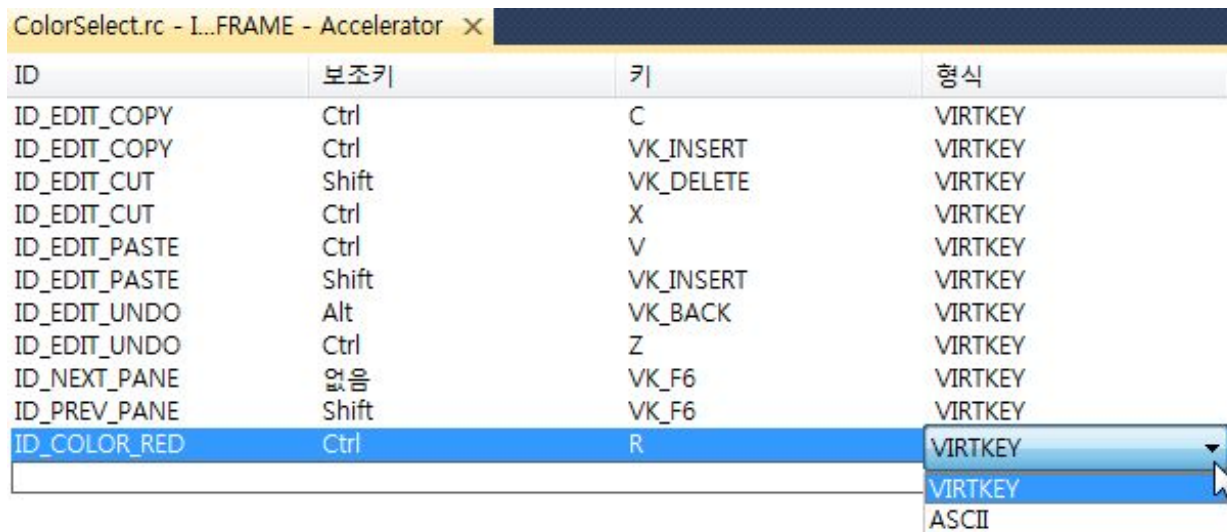
- 시스템 메뉴 조작하기
 - CWnd::GetSystemMenu() 함수를 이용하여 시스템 메뉴를 가리키는 CMenu 포인터를 얻는다.
 - CMenu 클래스가 제공하는 다양한 멤버 함수(AppendMenu(), InsertMenu(), DeleteMenu(), ...)를 적용한다.

시스템 메뉴

- 시스템 메뉴 조작 시 주의 사항
 - 시스템 메뉴를 변경하려면 `GetSystemMenu(FALSE)` 함수를 호출하고, 반대로 시스템 메뉴를 초기 상태로 되돌리려면 `GetSystemMenu(TRUE)` 함수를 호출한다.
 - 시스템 메뉴에 새로운 메뉴 항목을 추가할 때 메뉴 ID는 반드시 16의 정수배여야 하고 `0xF000`보다 작은 값이어야 한다.
 - 시스템 메뉴 항목을 선택하면 `WM_COMMAND`가 아닌 `WM_SYSCOMMAND` 메시지가 발생한다.

가속기

- 가속기 = 단축키
 - 메뉴 항목을 곧바로 실행할 수 있는 키 조합
- 가속기 리소스 추가



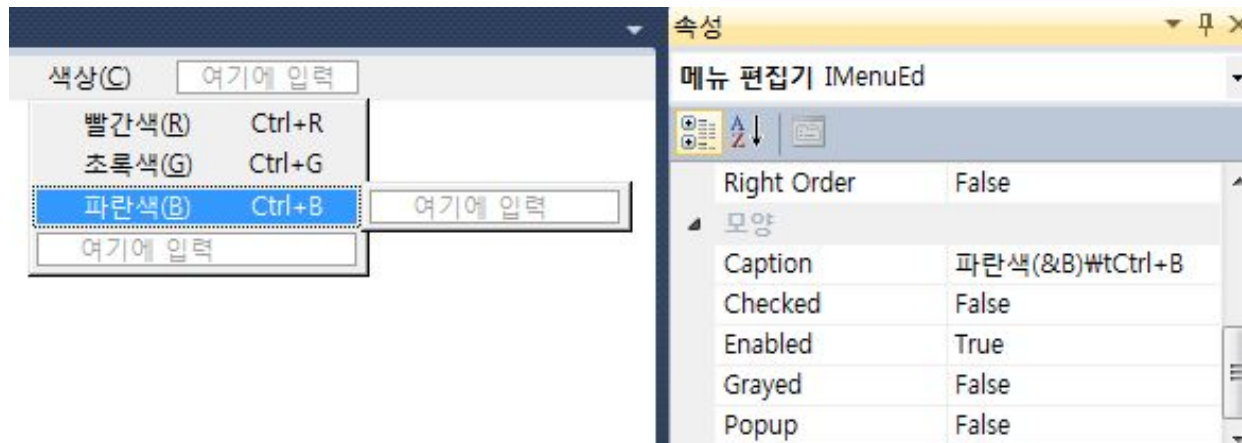
ID	보조키	키	형식
ID_EDIT_COPY	Ctrl	C	VIRTKEY
ID_EDIT_COPY	Ctrl	VK_INSERT	VIRTKEY
ID_EDIT_CUT	Shift	VK_DELETE	VIRTKEY
ID_EDIT_CUT	Ctrl	X	VIRTKEY
ID_EDIT_PASTE	Ctrl	V	VIRTKEY
ID_EDIT_PASTE	Shift	VK_INSERT	VIRTKEY
ID_EDIT_UNDO	Alt	VK_BACK	VIRTKEY
ID_EDIT_UNDO	Ctrl	Z	VIRTKEY
ID_NEXT_PANE	없음	VK_F6	VIRTKEY
ID_PREV_PANE	Shift	VK_F6	VIRTKEY
ID_COLOR_RED	Ctrl	R	VIRTKEY

Context menu for ID_COLOR_RED:

- VIRTKEY
- VIRTKEY
- ASCII

가속기

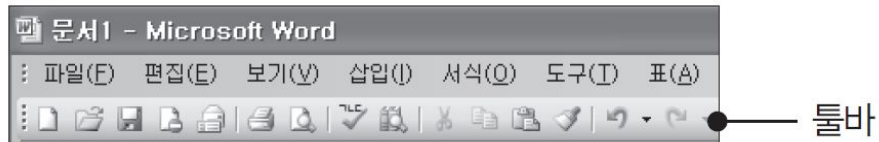
- 메뉴 항목에 가속기 표시



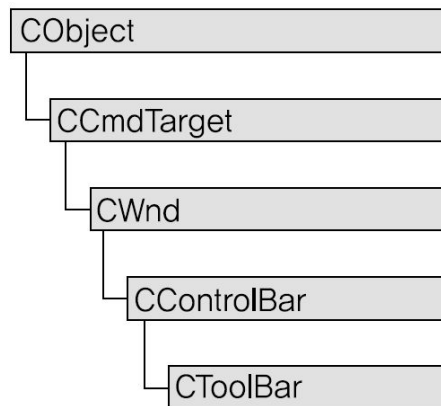
실습 - 메뉴

툴바

- 툴바

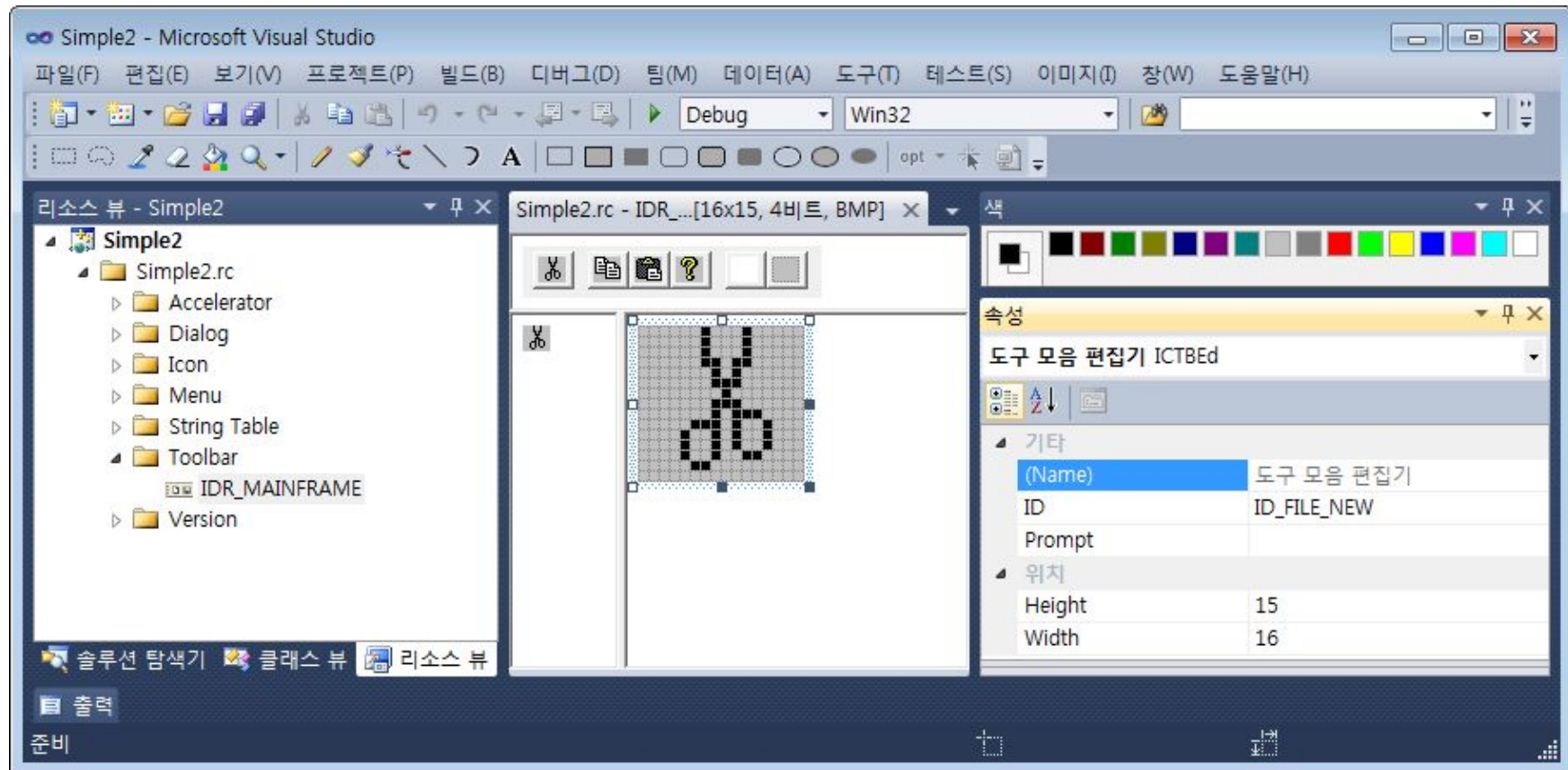


- MFC 클래스 계층도



툴바 생성

- 툴바 리소스 편집



툴바 생성

- 툴바 코드 (1/2)

```
class CMainFrame : public CFrameWnd
{
    ...
protected:
    CToolBar m_wndToolBar;
    CStatusBar m_wndStatusBar;
    CChildView m_wndView;
    ...
};
```

툴바 생성

- 툴바 코드 (2/2)

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    ...
    if(!m_wndToolBar.CreateEx(this, TBSTYLE_FLAT,
        WS_CHILD | WS_VISIBLE | CBRS_TOP | CBRS_GRIPPER |
        CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_SIZE_DYNAMIC) ||
        !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
    {
        TRACE0("도구 모음을 만들지 못했습니다.\n");
        return -1;
    }

    ...
    m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndToolBar);

    return 0;
}
```

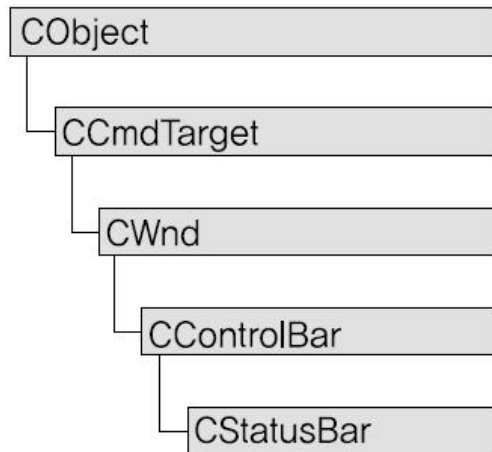
실습 - 툼바

상태바

- 상태바



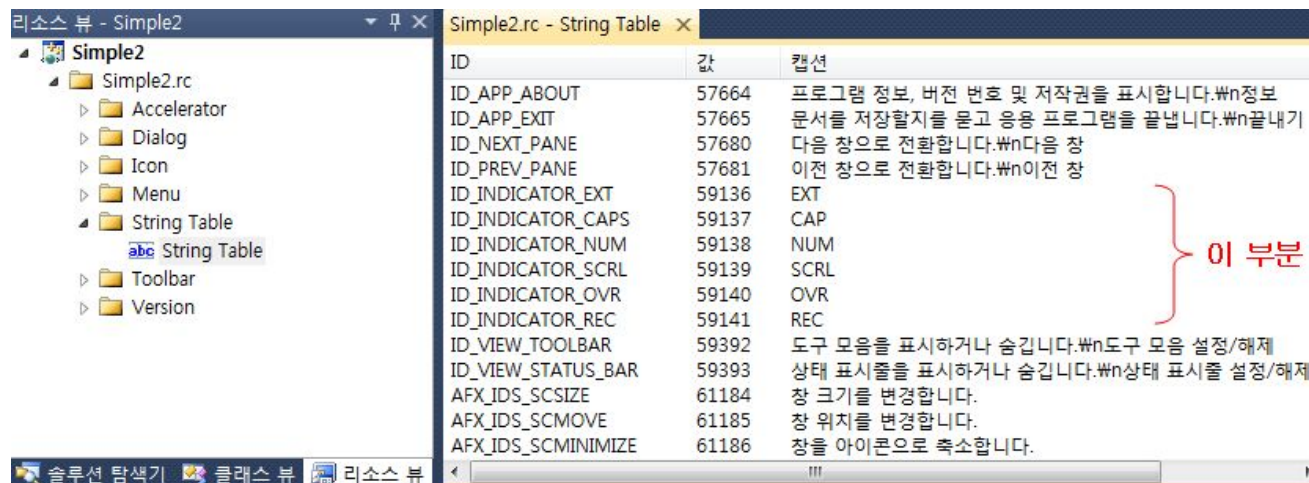
- MFC 클래스 계층도



상태바 생성

- 상태바 리소스

```
static UINT indicators[] =  
{  
    ID_SEPARATOR,  
    ID_INDICATOR_CAPS,  
    ID_INDICATOR_NUM,  
    ID_INDICATOR_SCRL,  
};
```



상태바 생성

- 상태바 코드 (1/2)

```
class CMainFrame : public CFrameWnd
{
    ...
protected:
    CToolBar m_wndToolBar;
    CStatusBar m_wndStatusBar;
    CChildView m_wndView;
    ...
};
```


상태바

- 상태바 코드 (2/2)

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    ...
    if(!m_wndStatusBar.Create(this))
    {
        TRACE0("상태 표시줄을 만들지 못했습니다.\n");
        return -1;
    }
    m_wndStatusBar.SetIndicators(indicators, sizeof(indicators)/sizeof(UINT));
    ...
}
```

실습 - 상태바

학습정리

- 윈도우 응용 프로그램에서 메뉴는 리소스를 이용해 생성하거나 코드를 이용해 생성할 수 있습니다.
- 메뉴 명령 처리는 명령 항목을 마우스나 키보드로 선택하면 WM_COMMAND 메시지 발생하고, WM_COMMAND 메시지 핸들러에서 메뉴 명령을 처리하는 순으로 이루어집니다.
- 메뉴 항목을 구분하는 ID 속성은 일반적으로 ID_메뉴이름_항목이름 형태로 만듭니다.
- Prompt 속성은 'Wn'을 기준으로 앞쪽 문자열은 상태바에 표시되고, 뒤쪽 문자열은 동일한 기능을 가진 툴바 항목 위에 툴팁(Tooltip)으로 표시됩니다.
- Caption 속성에서 액세스키를 지정하려면 해당 문자 앞에 & 기호를 사용하고, 단축키는 'Wt'를 삽입하여 단축키를 나타내는 문자열이 탭 위치에 정렬되게 하는 것이 좋습니다.
- Pop-up 속성을 설정하면 명령 항목이 아닌 팝업 항목이 됩니다.
- CMenu 클래스는 메뉴를 코드 수준에서 제어할 수 있는 기능 제공하고, CCmdUI 클래스는 메뉴 상태 �신 등의 기능을 지원합니다.
- 메뉴가 열리기 전에 MFC에서는 자동으로 명령 �신 핸들러를 호출하여 메뉴 항목의 상태를 바꾸어 줍니다.
- WM_CONTEXTMENU 메시지는 마우스 오른쪽 버튼을 누르거나, [Shift] + [F10] 키를 누른 경우, 가상 키코드 VK_APPS에 해당하는 키를 누른 경우에 발생합니다.
- 시스템 메뉴 항목을 선택하면 WM_COMMAND가 아닌 WM_SYSCOMMAND 메시지가 발생합니다.
- 메뉴 항목을 곧바로 실행할 수 있는 키 조합을 가속기, 단축키라고 합니다.