

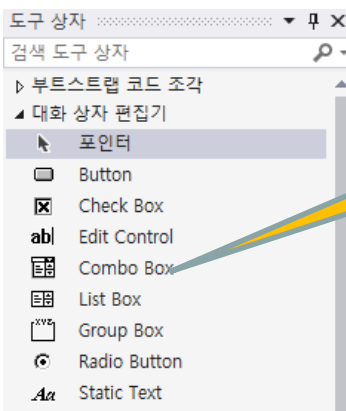
# 6장 대화상자와 컨트롤

2020년도 1학기 윈도우 프로그래밍

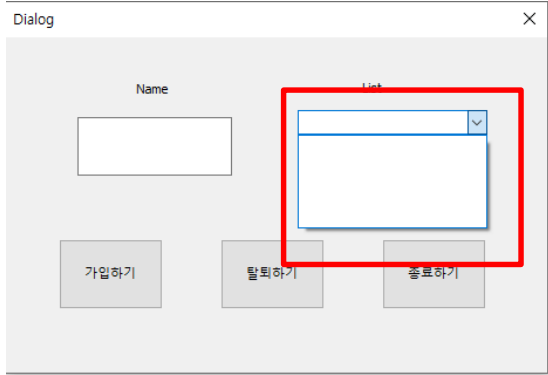
# 4) 콤보 박스 컨트롤

## • 콤보 박스 컨트롤

- 사용자의 키보드 입력 또는 출력을 위한 편집창
- 여러 항목들의 리스트를 나열하여 보여주는 컨트롤



콤보 박스: 여러 항목  
리스트 나열



## • 콤보 박스 통지 정보

인자값		내용	
wParam	HIWORD(wParam)	컨트롤에 따른 통지 정보	<ul style="list-style-type: none"><li>• CBN_DROPDOWN: 콤보 박스에 등록된 항목들이 아래로 펼쳐짐</li><li>• CBN_DBLCLK: 아래로 펼쳐진 항목 리스트에서 하나를 더 클릭으로 선택했음</li><li>• CBN_EDITCHANGE: 콤보 박스의 텍스트 편집 공간에 텍스트를 추가하거나 수정하였음</li><li>• CBN_SELCHANGE: 사용자가 항목 리스트에서 하나를 선택하였음</li></ul>
	LOWORD(wParam)	컨트롤 ID	
lParam		컨트롤 핸들값	

# 콤보박스에 보내는 메시지

- 콤보 박스 컨트롤에 보내는 메시지 (SendMessage 함수로 보내는 메시지)

메시지	의미	전달 값
<b>CB_ADDSTRING</b>	콤보 박스에 텍스트를 아이템으로 추가하는 메시지로써 리스트의 마지막에 추가된다.	<b>wParam:</b> 사용하지 않음 <b>lParam:</b> 텍스트 스트링의 시작 주소
<b>CB_DELETESTRING</b>	콤보 박스에 있는 아이템들 중 하나를 삭제하는 메시지	<b>wParam:</b> 삭제하기 원하는 아이템의 인덱스로 0부터 시작한다. <b>lParam:</b> 0
<b>CB_GETCOUNT</b>	콤보 박스의 아이템 리스트에 들어 있는 아이템의 개수를 얻기 위한 메시지로 개수 값은 SendMessage()함수가 리턴한다.	<b>wParam:</b> 0 <b>lParam:</b> 0
<b>CB_GETCURSEL</b>	현재 선택된 아이템의 인덱스 번호를 얻기 위한 메시지로 인덱스 번호는 SendMessage()함수가 리턴한다.	<b>wParam:</b> 0 <b>lParam:</b> 0
<b>CB_SETCURSEL</b>	콤보 박스 컨트롤의 텍스트 편집 공간에 지정한 항목의 텍스트를 보여준다.	<b>wParam:</b> 나타내고자 하는 항목의 인덱스 번호 <b>lParam:</b> 사용안함

# SendMessage()

## • SendMessage() 함수는

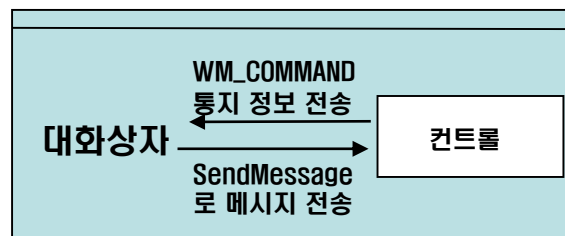
- 메시지를 메시지 큐에 넣지 않고 바로 윈도우 프로시저로 보냄
- 윈도우 프로시저로 메시지를 보내 바로 처리
- 메시지가 처리되기 전까지 반환되지 않음, 즉 윈도우 프로시저가 값을 반환해야만 SendMessage 도 반환하여 끝마칠 수 있음

예) SendMessage (hCombo, CB\_ADDSTRING, 0, (LPARAM)name);

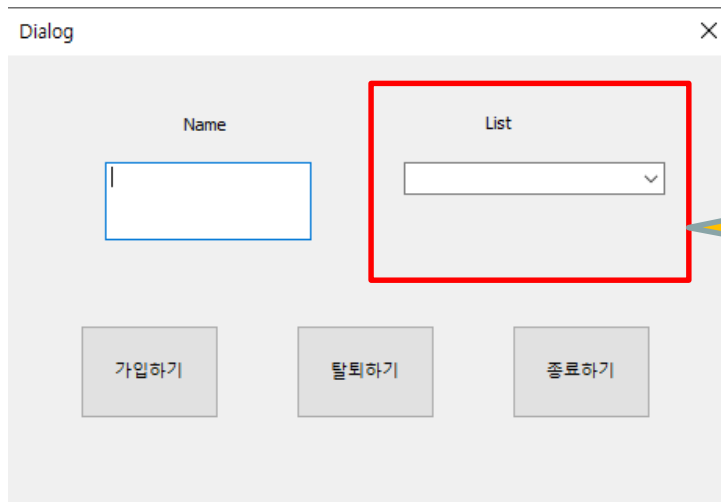
- hCombo 컨트롤에 CB\_ADDSTRING 메시지를 보내는데, 즉 문자열 name을 hCombo 에 추가하라는 메시지
- 윈도우에서 컨트롤로 메시지 전송 : ADD\_STRING, DELETE\_STRING
- 컨트롤에서 윈도우로 메시지 전송 : LBN\_DBLCLK, LBN\_SELCHG

**LRESULT SendMessage (HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam);**

- hWnd: 메시지를 전달받을 윈도우 핸들
- Msg : 전달할 메시지
- wParam, lParam 메시지의 추가적 정보, 메시지에 따라 다른 정보 반환



- 사용 예) 대화상자에 콤보박스 그리기
  - 회원이름을 넣고 가입하면 회원명단에 추가됨
  - 콤보박스 사용



리소스에서 콤보박스를 만들 때, 높이를 충분히 길게 만들어준다. 이 길이는 우측의 pull-down 버튼을 눌렀을 때 보여지는 콤보박스의 길이

컨트롤 종류	ID	
Static	IDC_STATIC	제목 보여주기
Static	IDC_STATIC	제목 보여주기
Edit	IDC_EDIT_NAME	이름 작성하기
<b>Combo</b>	<b>IDC_COMBO_LIST</b>	<b>작성한 이름 출력하기</b>
Button	IDC_BUTTON_INSERT	회원 이름을 명단에 넣기
Button	IDC_BUTTON_DELETE	이름 삭제하기
Button	IDC_CLOSE	대화상자 닫기

# 사용 예

```
BOOL CALLBACK Dlg6_6Proc (HWND hDlg, UINT iMsg, WPARAM wParam, LPARAM lParam)
{
    static int selection;
    static HWND hCombo;
    TCHAR name[20];

    switch(iMsg)
    {
    case WM_INITDIALOG:
        hCombo = GetDlgItem(hDlg, IDC_COMBO_LIST);          // 회원명단
        break;

    case WM_COMMAND:
        switch (LOWORD(wParam))
        {
        case IDC_BUTTON_INSERT:                               // 가입 버튼이 눌러짐
            GetDlgItemText (hDlg, IDC_EDIT_NAME, name, 20); // 이름 문자열 획득
            if (lstrcmp(name, L""))                          // 이름이 들어 왔으면, 그 값으로 채워라
                SendMessage (hCombo, CB_ADDSTRING, 0, (LPARAM)name);
            break;

        case IDC_BUTTON_DELETE:                               // 탈퇴하라 버튼이 눌러짐
            SendMessage (hCombo, CB_DELETESTRING, selection, 0);
            break;

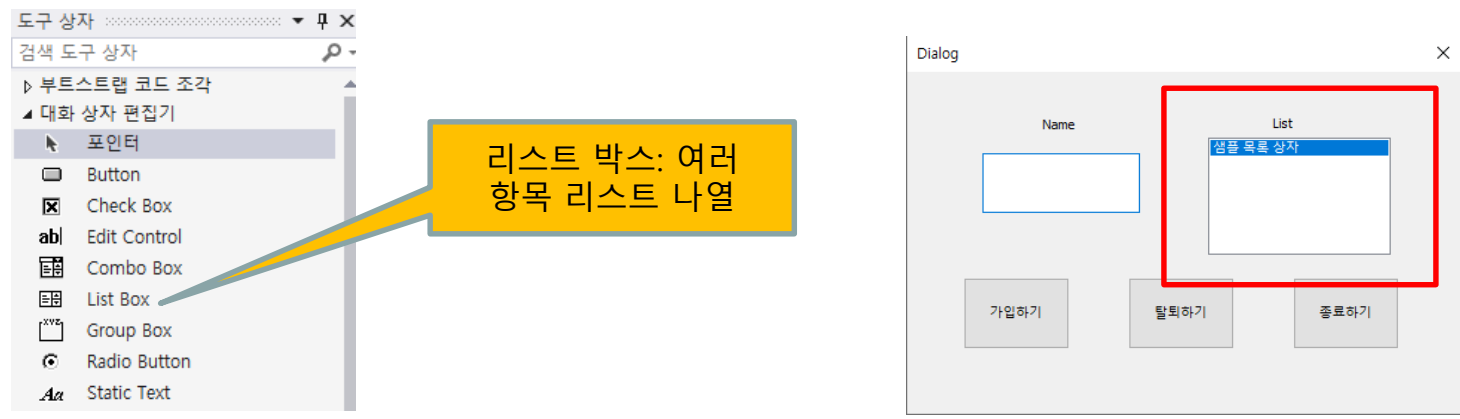
        case IDC_COMBO_LIST:                                  // 콤보박스가 눌러짐
            if (HIWORD(wParam) == CBN_SELCHANGE)              // 하나가 선택됨(상태 변경)
                selection = SendMessage (hCombo, CB_GETCURSEL, 0, 0);
            break;

        case IDC_CLOSE:
            EndDialog(hDlg,0);
            break;
        }
    }
    return 0;
}
```

# 5) 리스트 박스 컨트롤

## • 리스트 박스

- 사용자의 키보드 입력 또는 출력을 위한 편집창
- 여러 항목들의 리스트를 나열하여 보여주는 컨트롤
- 콤보 박스 컨트롤은 버튼을 누르기 전에는 항목 리스트 컨트롤을 보여주지 않지만, 리스트 컨트롤은 외부 입력이 없어도 항목을 보여준다.



## • 리스트 박스에서 오는 통지 정보

인자값		내용	
wParam	HIWORD(wParam)	컨트롤에 따른 통지 정보	<ul style="list-style-type: none"><li>• LBN_DBLCLK: 리스트 박스의 여러 아이템들 중 하나를 더블 클릭 했음</li><li>• LBN_SELCHANGE: 아이템들중 하나가 선택되었음</li><li>• LBN_SETFOCUS: 리스트 박스가 포커스를 받았음</li><li>• LBN_KILLFOCUS: 리스트 박스가 포커스를 잃었음</li></ul>
	LOWORD(wParam)	컨트롤 ID	
lParam		컨트롤 핸들값	

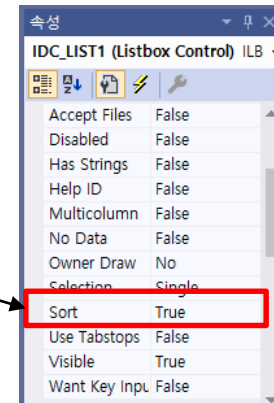
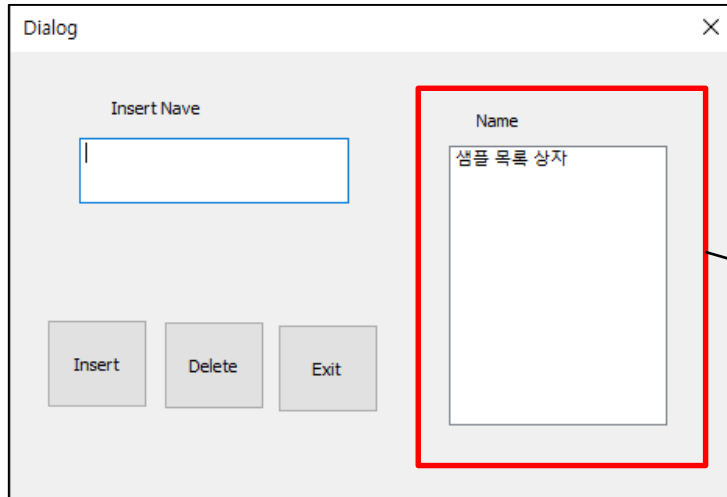
# 리스트 박스에 보내는 메시지

- 리스트 박스에 보내는 메시지 (SendMessage 함수로 보내는 메시지)

메시지	의미	전달 값
LB_ADDSTRING	리스트 박스에 텍스트를 아이템으로 추가하는 메시지로써 리스트의 마지막에 추가된다.	wParam: 사용하지 않음 lParam: 텍스트 스트링의 시작 주소
LB_DELETESTRING	리스트 박스에 있는 아이템들 중 하나를 삭제하는 메시지	wParam: 삭제하기 원하는 아이템의 인덱스로 0부터 시작한다. lParam: 0
LB_GETCOUNT	리스트 박스의 아이템 리스트에 들어 있는 아이템의 개수를 얻기 위한 메시지로 개수 값은 SendMessage()함수가 리턴한다.	wParam: 0 lParam: 0
LB_GETCURSEL	현재 선택된 아이템의 인덱스 번호를 얻기 위한 메시지로 인덱스 번호는 SendMessage()함수가 리턴한다.	wParam: 0 lParam: 0
LB_GETTEXT	아이템 리스트중 wParam에서 지정한 인덱스 아이템의 텍스트를 얻어오는 메시지	wParam: 얻어올 아이템의 인덱스 번호 lParam: 얻어온 텍스트를 저장할 버퍼의 시작 주소
LB_INSERTSTRING	리스트 박스에 텍스트를 아이템으로 리스트 중간에 추가하는 메시지	wParam: 아이템 리스트중 추가될 위치의 인덱스 번호 lParam: 텍스트 스트링의 시작 주소



- 사용 예) 대화상자에 리스트박스 그리기
  - 리스트박스 사용



컨트롤 종류	ID	기능
Static	IDC_STATIC	Insert Name 제목 보여주기
Static	IDC_STATIC	Name 보여주기
Edit	IDC_EDIT_NAME	이름 입력하는 에디트 박스
<b>List Box</b>	<b>IDC_LIST_NAME</b>	<b>추가된 이름 리스트 보여주는 리스트 박스</b>
Button	IDC_BUTTON_INSERT	이름을 리스트박스에 추가하는 버튼
Button	IDC_BUTTON_DELETE	리스트박스에서 현재 선택된 아이템을 삭제
Button	IDC_CLOSE	대화상자 닫기

```
BOOL CALLBACK Dlg6_7Proc (HWND hDlg, UINT iMsg, WPARAM wParam, LPARAM lParam)
{
```

```
    static int selection;
    static HWND hList;
```

```
switch(iMsg)
{
case WM_INITDIALOG:
    hList = GetDlgItem (hDlg, IDC_LIST_NAME);
    break;

case WM_COMMAND:
    switch (LOWORD(wParam))
    {
case IDC_BUTTON_INSERT:
    GetDlgItemText (hDlg, IDC_EDIT_NAME, name, 20);
    if (strcmp(name, ""))
        SendMessage (hList, LB_ADDSTRING, 0, (LPARAM)name);
    break;

case IDC_BUTTON_DELETE:
    SendMessage (hList, LB_DELETESTRING, selection, 0);
    break;

case IDC_LIST_NAME:
    if (HIWORD(wParam) == LBN_SELCHANGE)
        selection = SendMessage (hList, LB_GETCURSEL, 0, 0);
    break;
    }
    break;
}
return 0;
}
```

### 3. 모달리스 대화상자

- **모달(Modal)형 대화상자**

- 이 대화상자를 닫지 않으면 다른 윈도우로 전환할 수 없는 특징을 갖는 대화상자
  - 대화상자가 떠있는 상태에서 해당 프로그램의 대화상자 이외의 부분을 클릭하면 "뽕"하는 소리가 나는 경우
- 해당 프로그램의 다른 윈도우로는 전환할 수 없으나, 다른 프로그램은 실행 가능
- 대표적인 예: MessageBox() 함수에 의해서 만들어진 대화상자

- **모달리스(Modaless)형 대화상자**

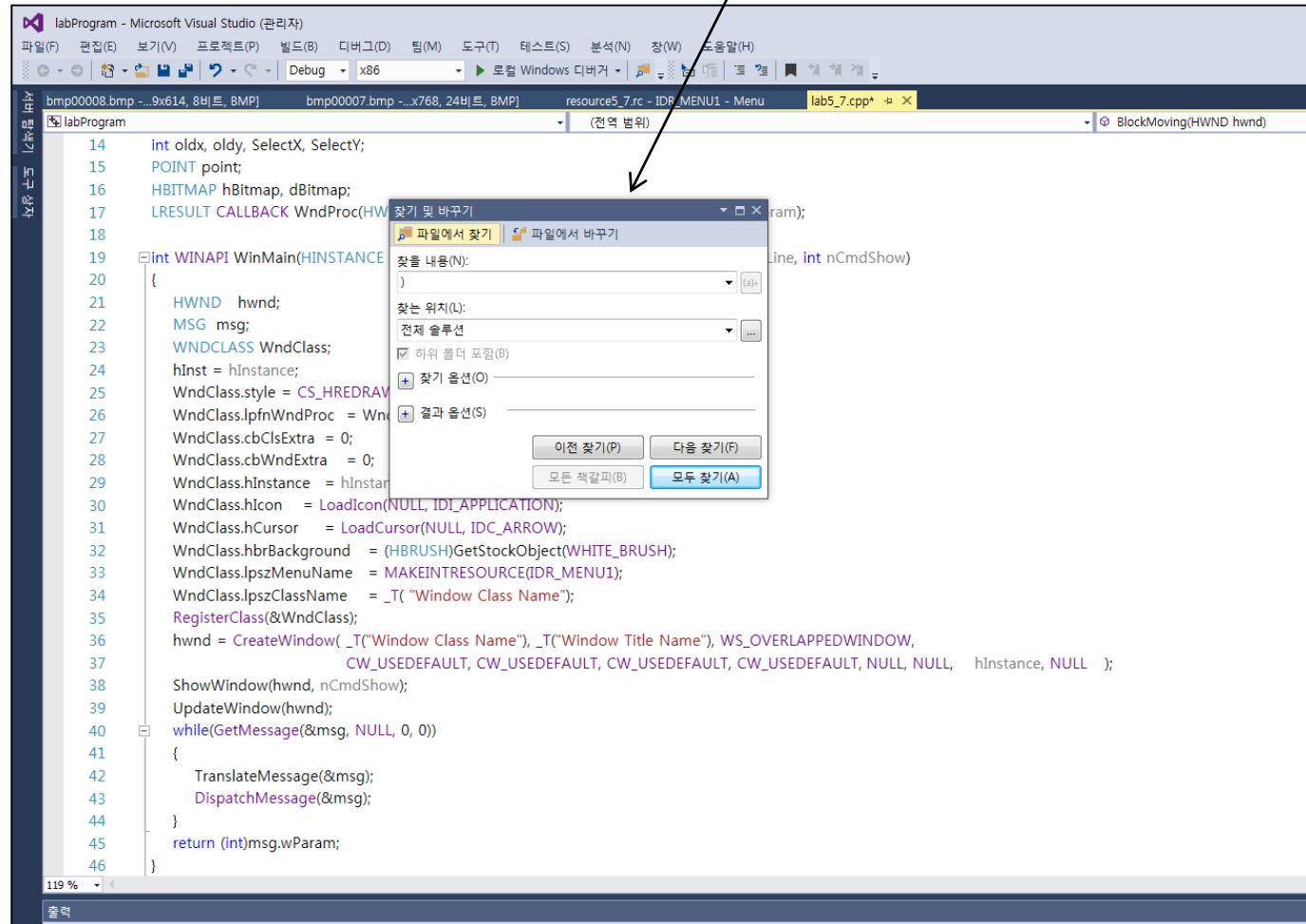
- 해당 대화상자를 닫지 않아도 다른 윈도우로 전환할 수 있는 특징을 갖는 대화상자
- 대표적인 예: 많은 프로그램에서 제공하는 "찾기" 대화상자
  - "찾기" 메뉴항목은 보통 해당 내용을 찾은 후 편집작업 등을 수행하고 다음 찾기를 하기 때문에 모달리스형 대화상자가 더 바람직

- **모달리스 대화상자 만들기**

- 모달형과 마찬가지로
  - 1) 리소스: 대화상자 만들기
  - 2) 대화상자 띄우기: 함수 호출
  - 3) 대화상자에서 발생하는 메시지 처리하기: 대화상자 프로시저 만들기

## 8. 모델리스 대화상자

- 모델리스 대화상자: 대화상자가 나타나도 부모 윈도우를 선택할 수 있는 대화상자



# 모델리스 대화상자 관련 함수

## • 모델리스 대화상자를 생성하는 함수

**HWND CreateDialog** ( **HINSTANCE** hInstance, **LPCTSTR** lpTemplate,  
**HWND** hWndParent, **DLGPROC** lpDialogFunc);

- 대화상자를 만들고 바로 대화상자의 핸들값을 리턴한다.
- **HINSTANCE** hInstance: 인스턴스 핸들
- **LPCTSTR** lpTemplate: 대화상자 ID
- **HWND** hWndParent: 윈도우 핸들
- **DLGPROC** lpDialogFunc: 메시지 처리 함수

## • 모델리스 대화상자를 보이거나 숨기는 함수

**BOOL ShowWindow** ( **HWND** hwnd, **int** nCmdShow );

- **HWND** hwnd: 윈도우 핸들
- **int** nCmdShow: 윈도우 보이기 (SW\_SHOW: 나타냄, SW\_HIDE: 감춤)

## • 모델리스 대화상자 종료하기 함수

**BOOL DestroyWindow** ( **HWND** hwnd);

- **HWND** hwnd: 윈도우 핸들

# 모달리스 대화상자

//--- 부모 윈도우의 윈도우 프로시저

```
LRESULT CALLBACK WndProc (HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam)
{
    HWND hDlg = NULL;
    switch (iMsg)
    {
        case WM_LBUTTONDOWN :    //--- 왼쪽 마우스 버튼을 누르면 모달리스 형태의 대화상자 열기
            hDlg = CreateDialog (hInst, MAKEINTRESOURCE(IDD_DIALOG6_8), hwnd, Dlg6_8Proc );
            ShowWindow (hDlg,SW_SHOW);
            break;

        return DefWndProc (hwnd, iMsg, wParam, lParam);
    }
}
```

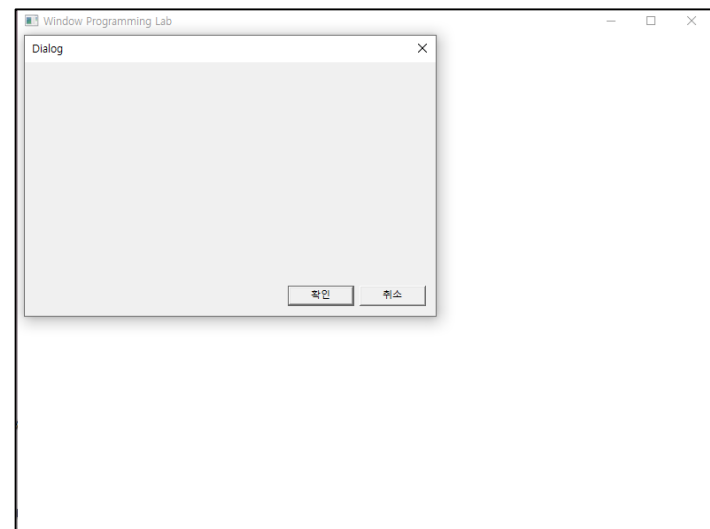
//--- 모달리스 대화상자의 대화상자 프로시저

//--- 모달형과 마찬가지로, 대화상자에 설정한 컨트롤들을 처리

```
BOOL CALLBACK Dlg6_8Proc (HWND hDlg, UINT iMsg, WPARAM wParam,LPARAM lParam)
{
    switch (iMsg) {
        case WM_COMMAND:
            switch (LOWORD (wParam)) {
                case IDOK:
                    DestroyWindow(hDlg);
                    hDlg=NULL;

                    break;
                case IDCANCEL:
                    DestroyWindow(hDlg);
                    hDlg=NULL;

                    break;
            }
        }
    return 0;
}
```

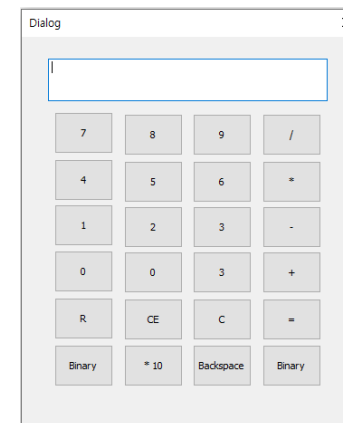


## • 모달리스 대화상자를 이용하여 계산기 구현하기

- 에디트 박스 컨트롤에 숫자를 직접 입력하거나 숫자 버튼을 눌러 입력한다.
- 버튼으로 숫자를 입력하는 계산기
  - 기본 사칙연산 버튼: + / - / \* / / (나눗셈)
  - 대입 버튼: = (연산을 하고 대입 버튼을 누르면 결과가 나온다 → 입력 스타일에 따라 진행)
- 기존 계산기에 버튼 추가한다.
- **버튼 1 (R 버튼):** 입력된 숫자들의 순서를 바꾸는 버튼
  - 예) 12+34 -> (R 버튼) -> 21 + 43
- **버튼 2 (CE 버튼):** 마지막으로 입력한 값을 지운다.
  - 예) 123 + 2 +3 → (CE 버튼) → 123 + 2 + 또는 123 + 2 (→ 모두 가능)
- **버튼 3 (C 버튼):** 모든 입력 값을 삭제한다.
- **버튼 4 (2진수 버튼):** 2진수로 바꿔서 출력한다.
- **버튼 5 (\*10 버튼):** 입력된 숫자에 10을 곱한다.
- **버튼 6 (← 버튼):** 입력된 숫자에서 마지막 한자리씩 삭제한다. (100자리 → 10자리 → 1자리)  
(실제 계산기에서 테스트해보기, 1자리에서 다시 버튼을 누르면 0이 된다.)
- **버튼 7 (종료하기):** 프로그램 (또는 대화상자)을 종료한다.

### \*\* 주의

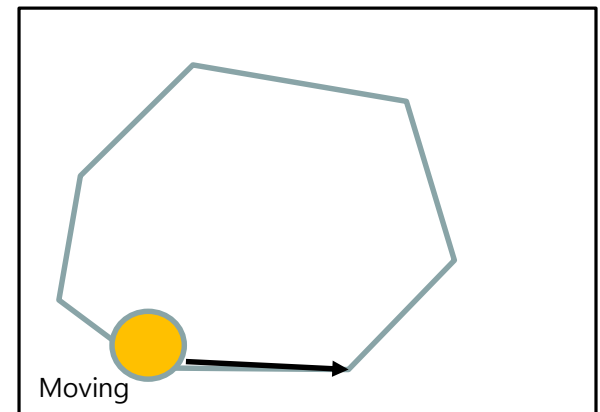
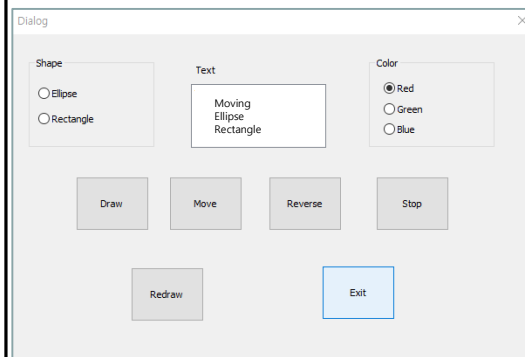
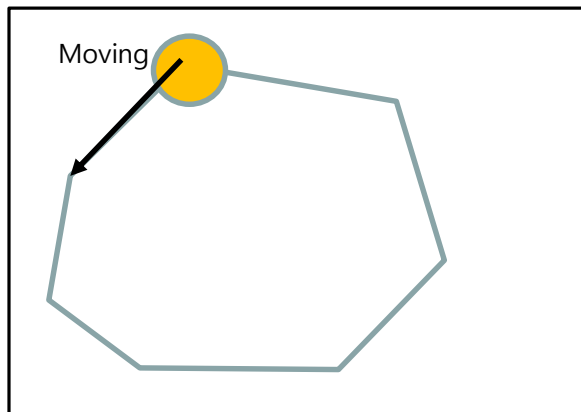
- 숫자: 정수를 기본으로 입력
- 나눗셈: 정수값으로 출력
- 숫자와 연산자 입력은 본인이 결정
  - 예) 연산자 입력하면 바로 결과 출력
- 사칙연산: 0.4점, 그 외의 버튼은 7번 버튼외에 각 0.1점으로 채점



## 모달리스 대화상자를 이용하여 경로따라 이동하는 도형 그리기

### 모달리스 대화상자 컨트롤

- **라디오 버튼: 도형** – 원, 사각형
- **라디오 버튼: 색상** – 도형 색 (임의의 색 3종류)
- **리스트박스: 문자열 선택** – 선택된 문자열이 이동하는 도형 옆에 따라다닌다. (최소 3개 문자열)
- **버튼1: 그리기** – 마우스 클릭, 드래그 이용하여 직선 경로 그리기 (최대 7개의 꼭짓점, 7개의 꼭짓점은 첫번째 꼭짓점과 연결하여 연결된 경로를 만든다.)
- **버튼2: 이동** – 라디오 버튼에서 선택된 도형을 직선 경로에 따라 이동 (옆에 문자열 같이 이동)
- **버튼3: 역방향 이동** – 경로를 반대로 이동한다.
- **버튼4: 멈추기** – 도형 이동 멈춤
- **버튼5: 다시 시작** – 그리기 다시 하기
- **버튼6: 종료** – 프로그램 (또는 대화상자) 종료





# 이번주에는

- 이번주에는
  - 대화상자 만들기
    - 컨트롤 만들기: 버튼, 에디트박스, 체크박스, 라디오버튼, 콤보박스, 리스트박스
- 다음주에는
  - 다중 윈도우 만들어 보기
    - 자식 윈도우 만들기
    - 윈도우로 컨트롤 만들기
- 이번주에도 수고하셨습니다. 다음주에 만나요~