

제4장 윈도우 메뉴

2019년 1학기 윈도우 프로그래밍

학습 목표

- 학습목표

- 리소스 파일을 만들고 윈도우에 메뉴를 등록하는 방법을 배운다.
- 메뉴에서 발생하는 메시지를 처리하는 방법을 배운다.
- 공용 대화상자의 종류를 알아보고 이를 적용해 프로그램의 완성도를 높인다.
- 윈도우에 등록된 메뉴를 프로그램 실행 중에 수정하는 방법을 배운다.

- 내용

- 메뉴 만들기
- 메뉴 사용하기
- 공용 대화상자 사용하기
- 메뉴 수정하기

1. 메뉴 만들기

- 리소스 파일 작성

- 방법: 소스 파일 작성과 유사
- C++ 소스 대신에 Resource Script 선택
- 리소스 파일 이름 명시: 리소스-파일.rc(리소스 정의), resource.h(리소스 ID 정의)
생성

- 메뉴화면 편집

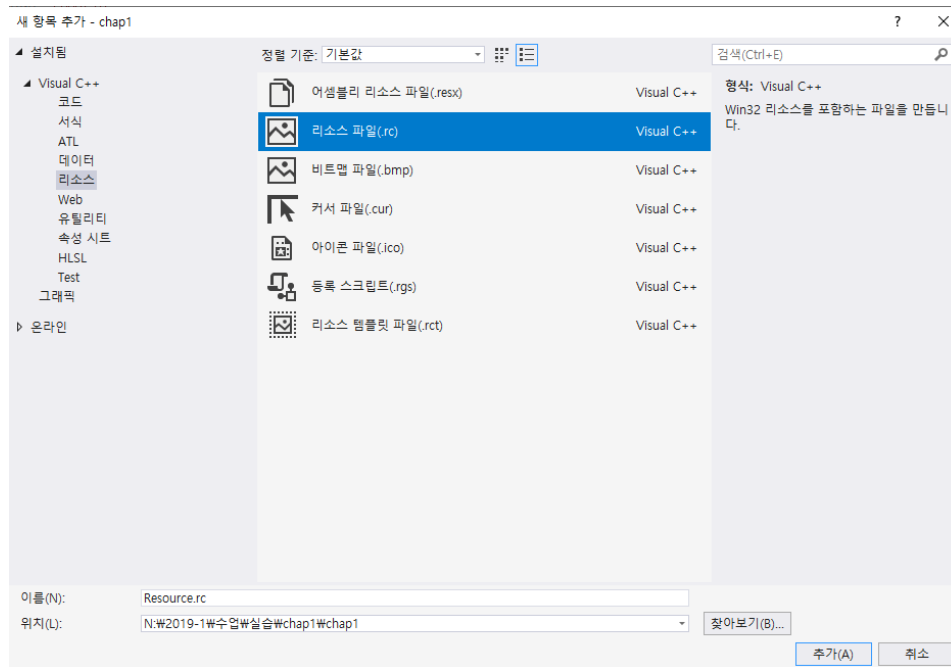
- 리소스 도구상자 이용
- 속성 정의 : 프로그램과 연계
- 메뉴 ID 정의 : 메뉴바에서

- 프로그램에서 메뉴 사용

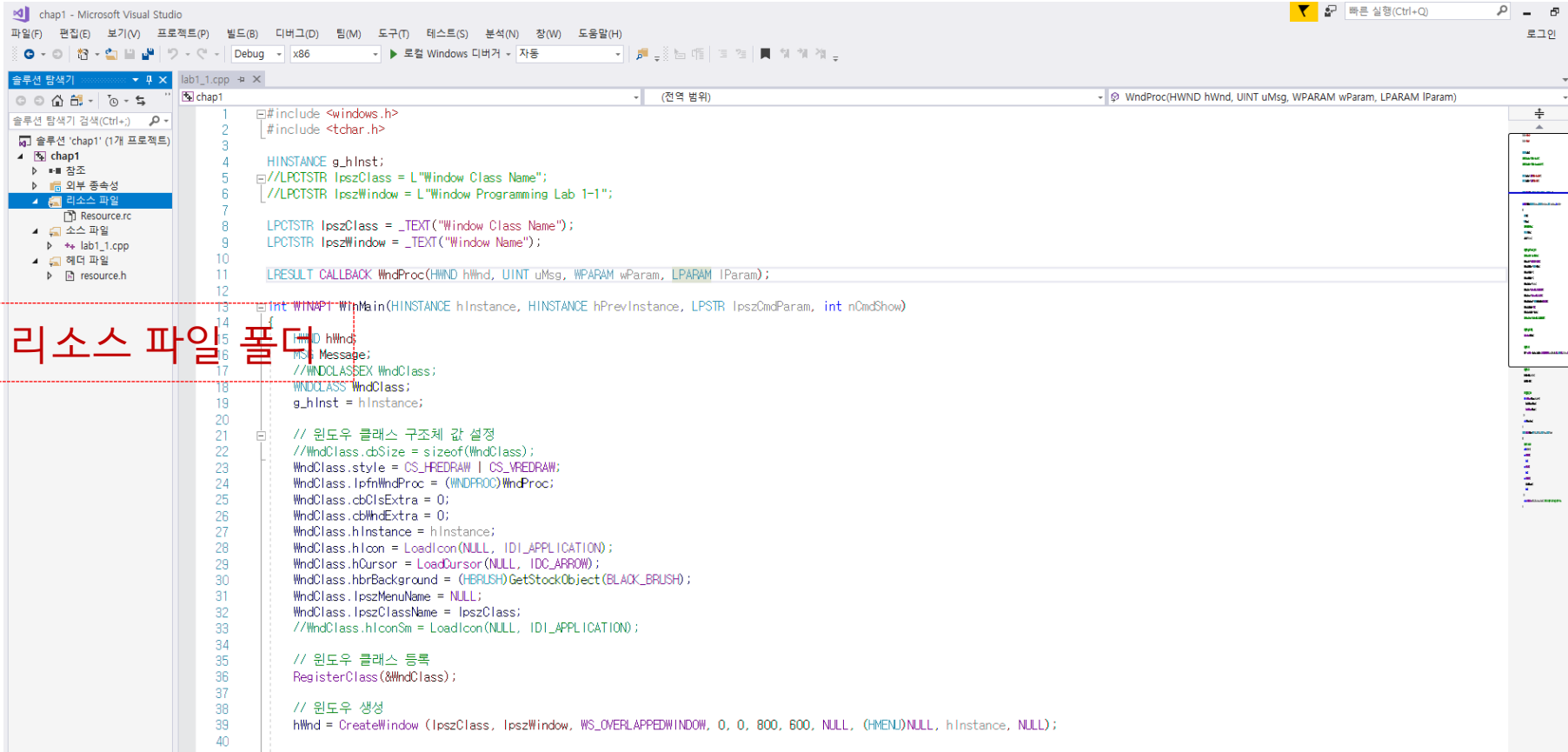
- 리소스 헤더파일 추가
 - Include "resource.h"
- 메뉴ID 등록
 - 윈도우 클래스 생성 시 또는 윈도우 생성 시 또는 메뉴 생성 함수를 통하여 메뉴 등록
- 메뉴 처리
 - 메뉴를 눌렀을 때 WM_COMMAND 메시지 발생, 처리

메뉴 만들기

- Visual Studio 2017 환경
 - [프로젝트] -> [리소스 파일] -> [새 항목 추가]

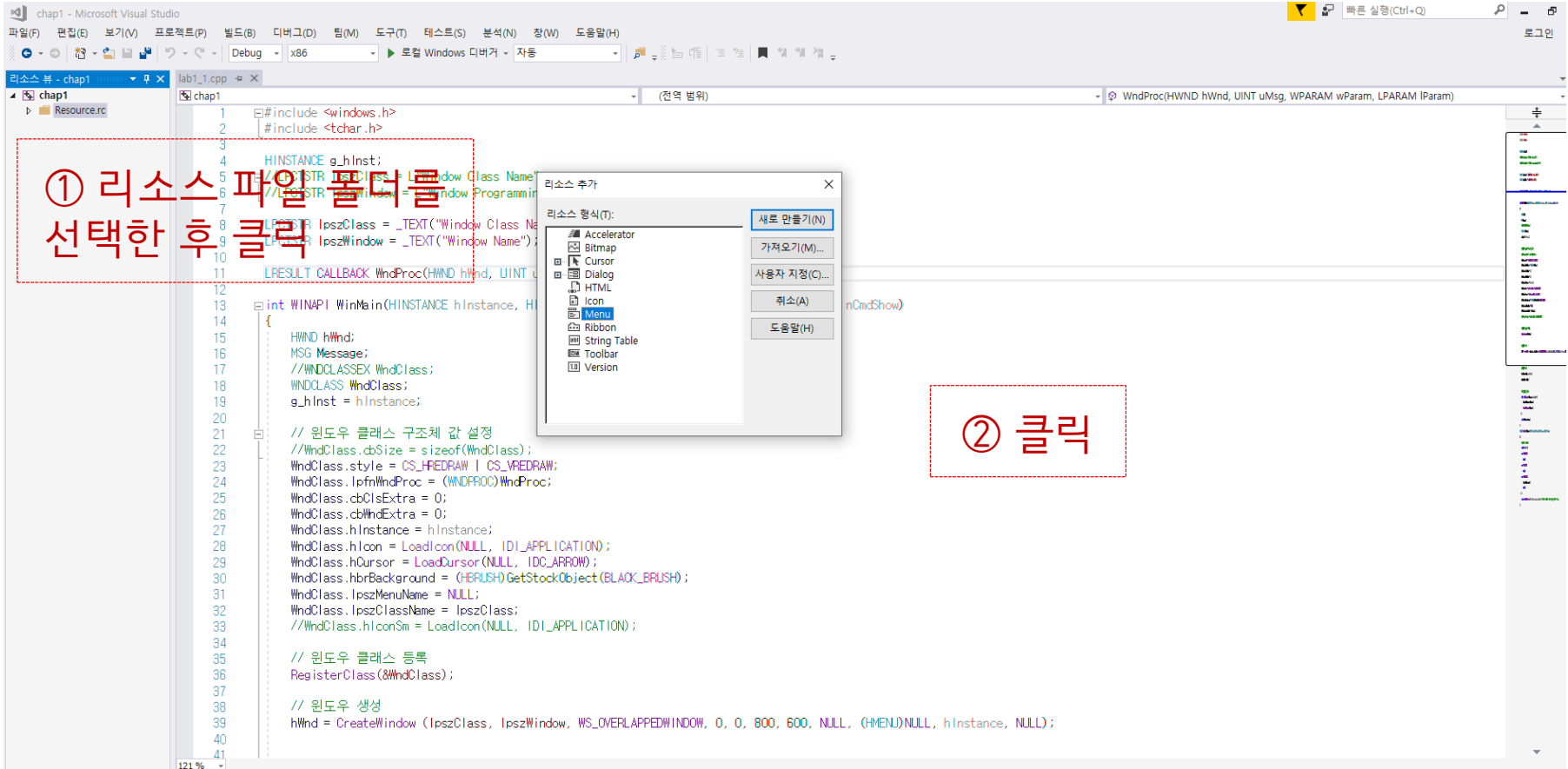


생성된 리소스 파일

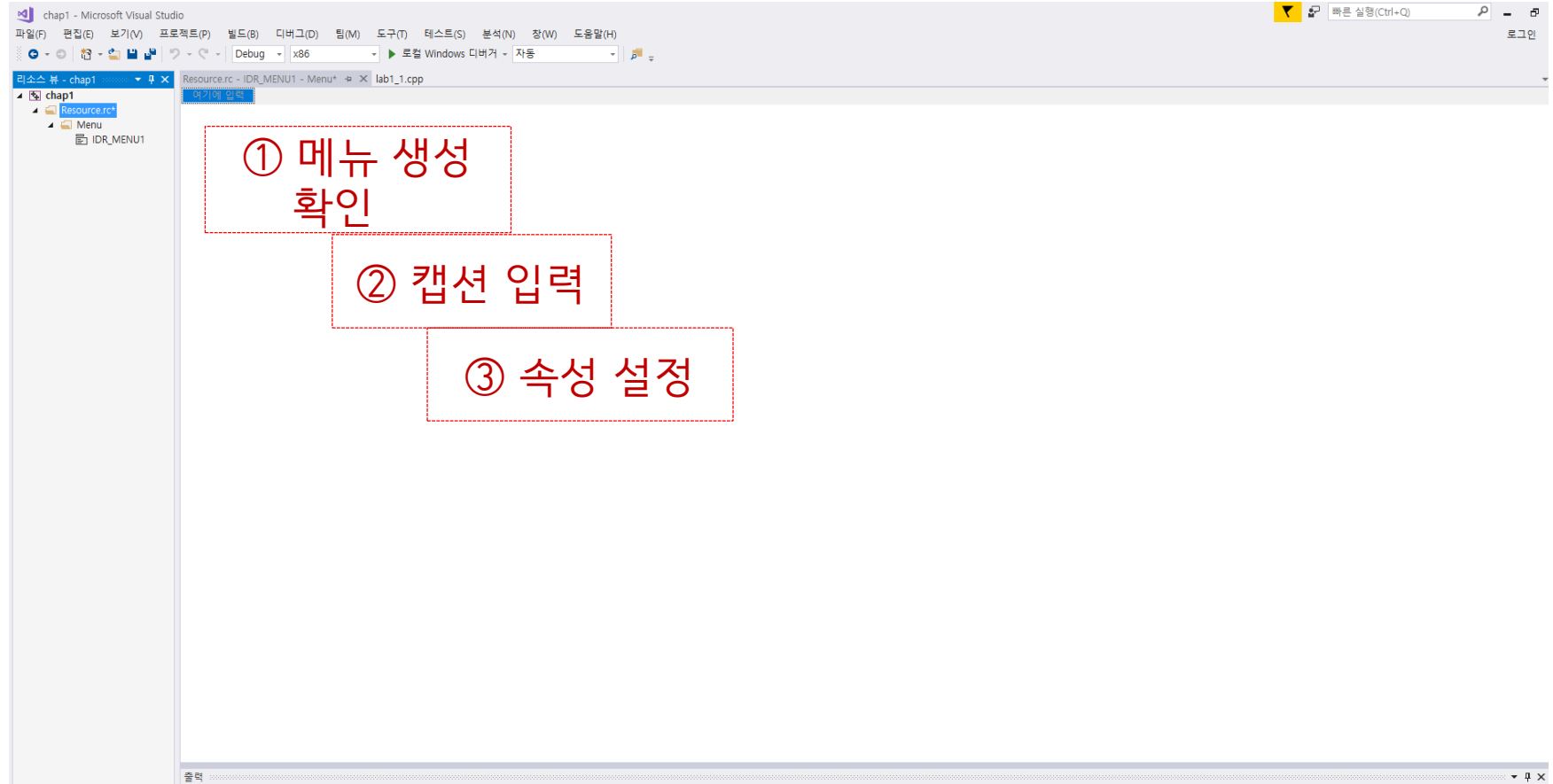


```
1 #include <windows.h>
2 #include <tchar.h>
3
4 HINSTANCE g_hInst;
5 //LPCTSTR IpszClass = L"Window Class Name";
6 //LPCTSTR IpszWindow = L"Window Programming Lab 1-1";
7
8 LPCTSTR IpszClass = _TEXT("Window Class Name");
9 LPCTSTR IpszWindow = _TEXT("Window Name");
10
11 LRESULT CALLBACK WndProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam);
12
13 int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpszCmdParam, int nCmdShow)
14 {
15     HWND hWnd;
16     MSG Message;
17     //WNDCLASSEX WndClass;
18     WNDCLASS WndClass;
19     g_hInst = hInstance;
20
21     // 윈도우 클래스 구조체 값 설정
22     //WndClass.cbSize = sizeof(WndClass);
23     WndClass.style = CS_HREDRAW | CS_VREDRAW;
24     WndClass.lpfnWndProc = (WNDPROC)WndProc;
25     WndClass.cbClsExtra = 0;
26     WndClass.cbWndExtra = 0;
27     WndClass.hInstance = hInstance;
28     WndClass.hIcon = LoadIcon(NULL, IDI_APPLICATION);
29     WndClass.hCursor = LoadCursor(NULL, IDC_ARROW);
30     WndClass.hbrBackground = (HBRUSH)GetStockObject(BLACK_BRUSH);
31     WndClass.lpszMenuName = NULL;
32     WndClass.lpszClassName = IpszClass;
33     //WndClass.hIconSm = LoadIcon(NULL, IDI_APPLICATION);
34
35     // 윈도우 클래스 등록
36     RegisterClass(&WndClass);
37
38     // 윈도우 생성
39     hWnd = CreateWindow(IpszClass, IpszWindow, WS_OVERLAPPEDWINDOW, 0, 0, 800, 600, NULL, (HMENU)NULL, hInstance, NULL);
40 }
```

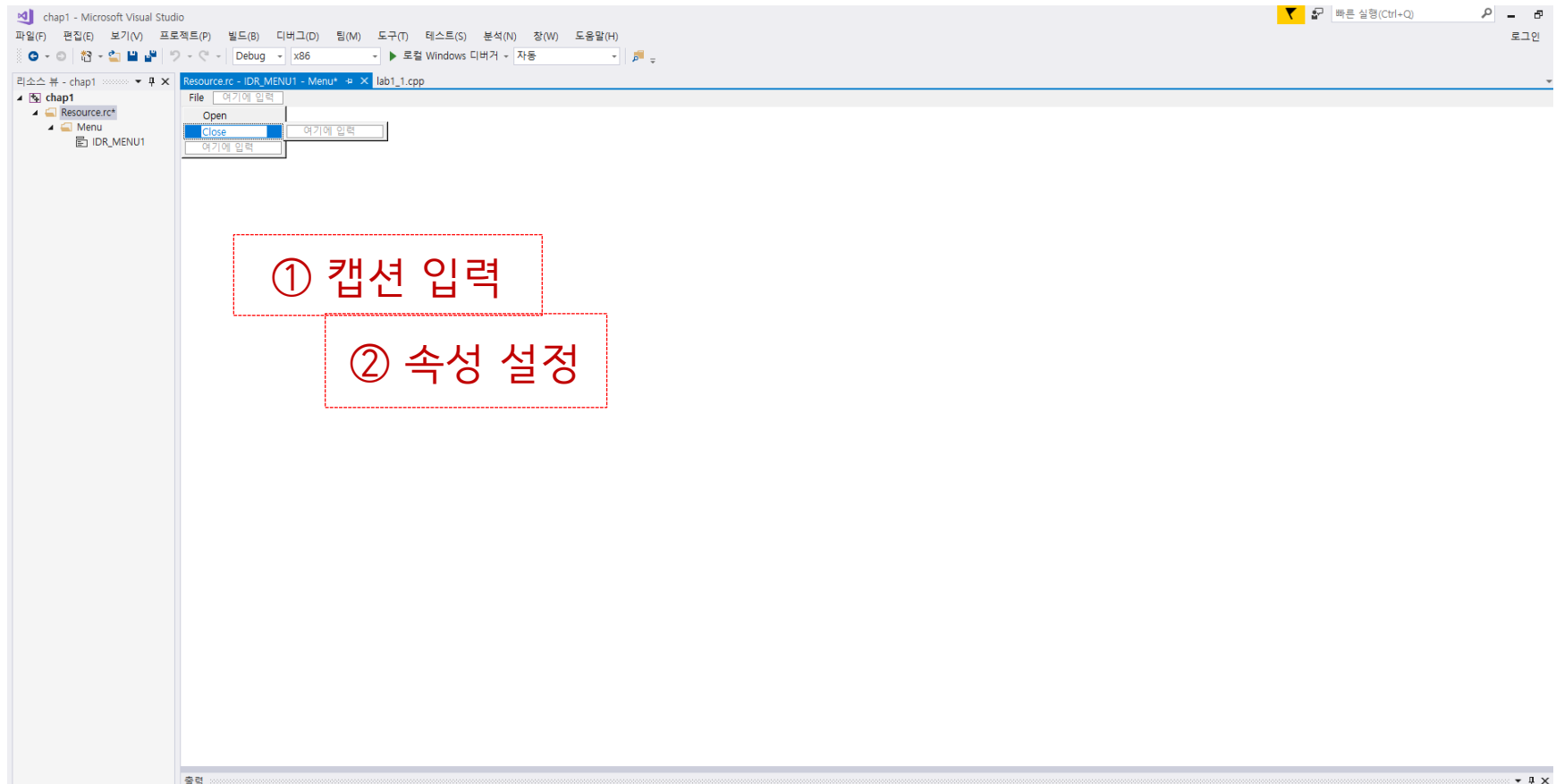
메뉴 만들기



메뉴 항목 편집



메뉴 항목 추가

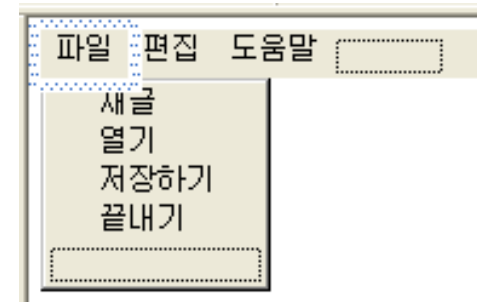


메뉴 만들기

- 예제 내용

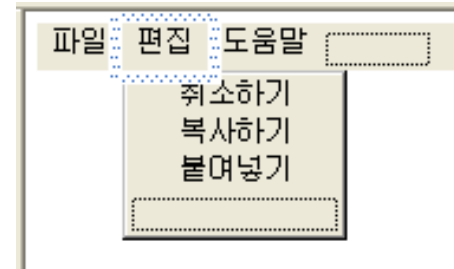
- 다음의 메뉴항목 설정표를 참고하여 기본 메뉴를 작성하기.
-> 다음 페이지 설명 계속

Caption	ID	속성
파일		Pop-up
새글	ID_FILENEW	디폴트
열기	ID_FILEOPEN	디폴트
저장하기	ID_FILESAVE	디폴트
끝내기	ID_EXIT	디폴트

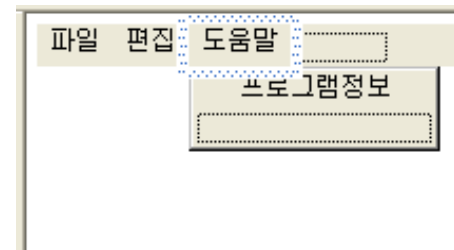


실습 4-1(계속)

Caption	ID	속성
편집		Pop-up
취소하기	ID_EDITUNDO	디폴트
복사하기	ID_EDITCOPY	디폴트
붙여넣기	ID_EDITPASTE	디폴트



Caption	ID	속성
도움말		Pop-up
프로그램정보	ID_INFORM	디폴트



2. 메뉴 사용하기

- 메뉴를 선택했을 때 WM_COMMAND 메시지가 전달
 - WM_COMMAND
 - 메뉴의 메뉴항목을 선택하면 발생하는 메시지
 - **Command 메시지**라 부름
 - 윈도우 프로시저로 보내지는 파라미터에는
 - LOWORD(wParam): 선택된 메뉴항목의 ID가 정수로 들어 있음
 - HIWORD(wParam): 0 (이벤트 소스)
 - lParam: 0
- 응용 프로그램에 메뉴 리소스를 불러오는 방법 3가지

방법	함수 사용
윈도우 클래스를 만들 때 메뉴를 정의	<code>winclass.lpszMenuName = MAKEINTRESOURCE (MYMENU);</code> // 윈도우 클래스 만들 때 추가
메인 윈도우를 생성할 때 메뉴를 첨부	<code>winclass.lpszMenuName = NULL;</code> <code>CreateWindow (szWindowClass, szTitle, WS_OVERLAPPEDWINDOW, 0, 0, 800, 600, NULL,</code> <code>LoadMenu (hinstance, MAKEINTRESOURCE (MYMENU)),</code> <code>hInstance, NULL);</code> // 윈도우 생성할 때 추가
초기 윈도우 생성이 끝난 후에 붙인다.	<code>HMENU hmymenu = LoadMenu (hinstance, MAKEINTRESOURCE (MYMENU));</code> <code>SetMenu (hwnd, hmymenu);</code> // 따로 만들어서 추가

윈도우에 메뉴 붙이기

```
#include "resource.h"                // 리소스 파일 첨부

LRESULT WINAPI WinMain ( ... )
{
    WNDCLASSEX WndClass;

    // wndclass 속성 설정에서
    WndClass.cbSize = sizeof(WndClass);
    WndClass.style=CS_HREDRAW | CS_VREDRAW;
    WndClass.lpfnWndProc=(WNDPROC)WndProc;
    WndClass.cbClsExtra=0;
    WndClass.cbWndExtra=0;
    WndClass.hInstance=hInstance;
    WndClass.hIcon=LoadIcon(NULL,IDI_APPLICATION);
    WndClass.hCursor=LoadCursor(NULL,IDC_ARROW);
    WndClass.hbrBackground=
        (HBRUSH)GetStockObject(BLACK_BRUSH);

    WndClass.lpszMenuName= MAKEINTRESOURCE (IDR_MENU);      // 메뉴 id 등록

    WndClass.lpszClassName=lpszClass;
    WndClass.hIconSm = LoadIcon(NULL,IDI_APPLICATION);

    RegisterClassEx (&WndClass);

    ...
}
```

윈도우에 메뉴 붙이기

- **MAKEINTRESOURCE**: 리소스에 대한 정수형 상수를 문자열로 변환하는 매크로 함수
 - LPTSTR **MAKEINTRESOURCE**(
WORD wInt; // 리소스에 대한 정수형 상수
);

커맨드 메시지 처리하기

```
int answer;
```

```
LRESULT CALLBACK WndProc (HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
```

```
{
    switch (iMsg)
    {
        case WM_COMMAND:
            switch (LOWORD(wParam)) {
                case ID_FILENEW:
                    MessageBox (hwnd, "새파일을 열겠습니까 ?", "새파일 선택" , MB_OKCANCEL );
                    break;

                case ID_EXIT:
                    answer = MessageBox (hwnd, "파일을 저장하고 끝내겠습니까 ? ",
                                           "끝내기 선택", MB_YESNOCANCEL );
                    if (answer == IDYES || answer == IDNO)
                        PostQuitMessage (0);;
                    break;
            }
            break;
    }
}
```

메시지박스

- 메시지박스: 사용자에게 경고나 알림 메시지를 주는 대화상자

int **MessageBox** (HWND hwnd, LPCTSTR lpText, LPCTSTR lpCaption, UINT uType);

- lpText: 메시지 박스에 표시될 글
- lpCaption: 메시지 박스의 타이틀바에 표시될 글
- uType과 함수 반환값

uType	반환 값
MB_OK	IDOK
MB_OKCANCEL	IDOK, IDCANCEL
MB_YESNO	IDYES, IDNO
MB_YESNOCANCEL	IDYES, IDNO, IDCANCEL

리소스: 아이콘

- 아이콘은 프로그램의 메인 윈도우가 최소화(아이콘화)되었을 때나 배경화면에 등록 될 때 응용 프로그램을 나타내는 작은 그래픽 이미지이다.
- 아이콘과 커서는 크게 두 가지 종류의 리소스가 있다.
 - 내장(built-in) 리소스와 사용자 정의 리소스이다.
 - 내장 리소스: 윈도우즈에서 기본적으로 제공하는 것이다.
 - LoadIcon() 함수의
 - 첫 번째 인자: "NULL"
 - 두 번째 인자: 아이콘 이름 문자열을 지정하며 리턴된 핸들을 wc.hIcon에 대입
- 손(IDI_HAND), 느낌표(IDI_WARNING) 등 9가지가 있으며 지금까지 사용한 표준 내장 아이콘(IDI_APPLICATION)은 다음과 같은 윈도우 모양을 갖는다.

```
wc.hIcon      = LoadIcon ( NULL, IDI_APPLICATION );  
wc.hIconSm = LoadIcon ( NULL, IDI_APPLICATION );
```

값	아이콘모양	같은 값
IDI_APPLICATION (디폴트)		IDI_WINLOGO
IDI_ASTERISK		IDI_INFORMATION
IDI_ERROR		IDI_HAND
IDI_EXCLAMATION		IDI_WARNING
IDI_QUESTION		

리소스: 커서

- 커서는 마우스의 위치를 나타내는 작은 그래픽 이미지이다.
 - 내장 커서는 아래의 표와 같은 모양이 제공된다.
 - LoadCursor()함수의
 - 첫 번째 인자: "NULL"
 - 두 번째 인자: 커서 이름 문자열을 지정하며 리턴된 핸들을 wc.hCursor에 대입
- 화살표, 모래시계 등 11가지가 있으며 지금까지 사용한 표준 내장 커서는 화살표 모양을 갖는다.

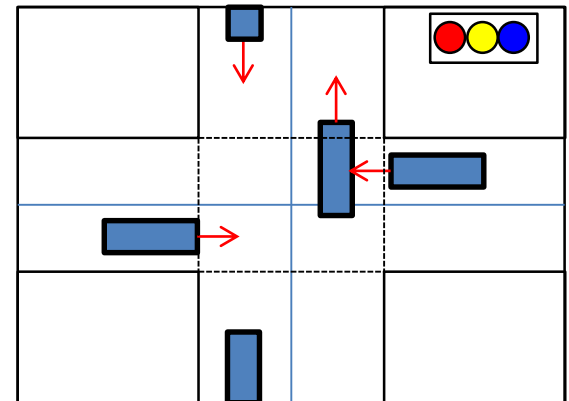
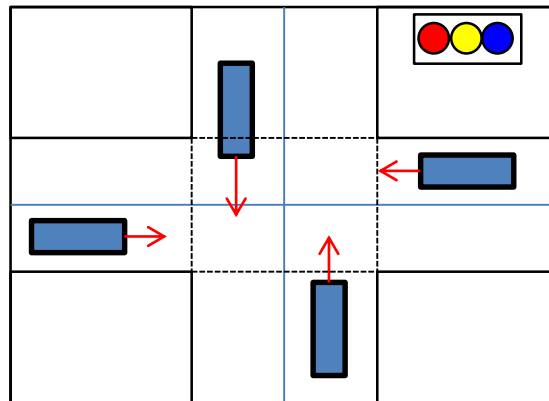
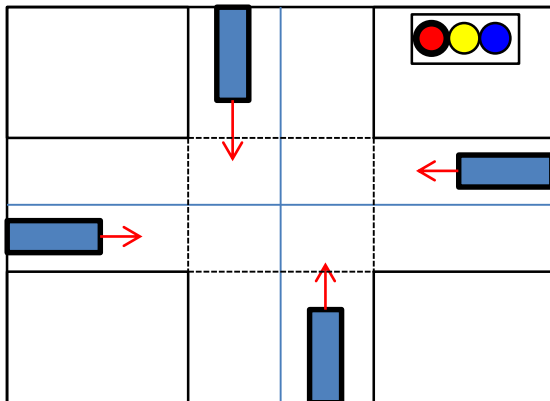
```
wc.hCursor = LoadCursor ( NULL ,IDC_ARROW);
```

값	커서	모양
IDC_APPSTARTING	프로그램이 시작될 때 사용된다.	
IDC_ARROW	표준 화살표 커서	
IDC_CROSS	십자 모양의 커서. 정확한 선택을 해야 할 때 사용된다.	
IDC_IBEAM	I자 모양의 커서. 주로 문자열 입력 영역에 사용된다.	
IDC_ICON	Win32에서는 사용되지 않음	
IDC_NO	원 안의 빗금이 쳐진 커서이며 드래그 금지 영역을 나타낸다.	
IDC_SIZE	Win32에서는 사용되지 않음	
IDC_SIZEALL	4방향 화살표	
IDC_SIZENESW	좌하우상 크기조절 커서	
IDC_SIZENS	수직 크기조절 커서	
IDC_SIZENWSE	좌상우하 크기조절 커서	
IDC_SIZEWE	수평 크기조절 커서	
IDC_UPARROW	수직 화살표	
IDC_WAIT	모래 시계 커서. 시간이 오래 걸리는 작업을 할 때 사용된다.	

실습 4-1

- 실습 3-3에 메뉴 붙이기

- 게임: Start/end (자동차들이 움직이기 시작/멈춤)
- 속도: 가속/감속 (자동차들의 속도를 가속/감속)
- 신호등: 빨강/파랑/노랑 (위의 신호등 내용을 메뉴로 실행) - 선택된 신호등 둘레를 두껍게 그린다.



실습 4-2

- 실습 3-2에 메뉴 붙이기
 - 게임: Start/end (공 튀기기 시작/프로그램 종료)
 - 속도: Fast/Medium/Slow (공의 이동 속도가 빠르게/중간/느리게)
 - 보드 크기: 작게/중간/크게 (공을 튀기는 보드의 크기를 작게/중간/크게)
 - 공 모양: 원/네모 (튀기는 공의 모양을 원/네모)
 - 벽돌 단: 2/3/4 (벽돌을 2단/3단/4단)

실습 4-3