

## MFC 비주얼프로그래밍

### A. 계산기 만들기

A-1. 프로젝트 만들기

A-2. 화면에 배치하기

A-3. 버튼을 눌렀을 때 덧셈 수행하기

### B. 계산기에 UpdateData() 사용하기

B-1. 컨트롤에 변수 추가하기

B-2. 버튼을 눌렀을 때 덧셈 수행하기

### C. 다이얼로그 창에 펜으로 그림 그리기

C-1. MOUSEMOVE 메시지에 대한 함수 생성

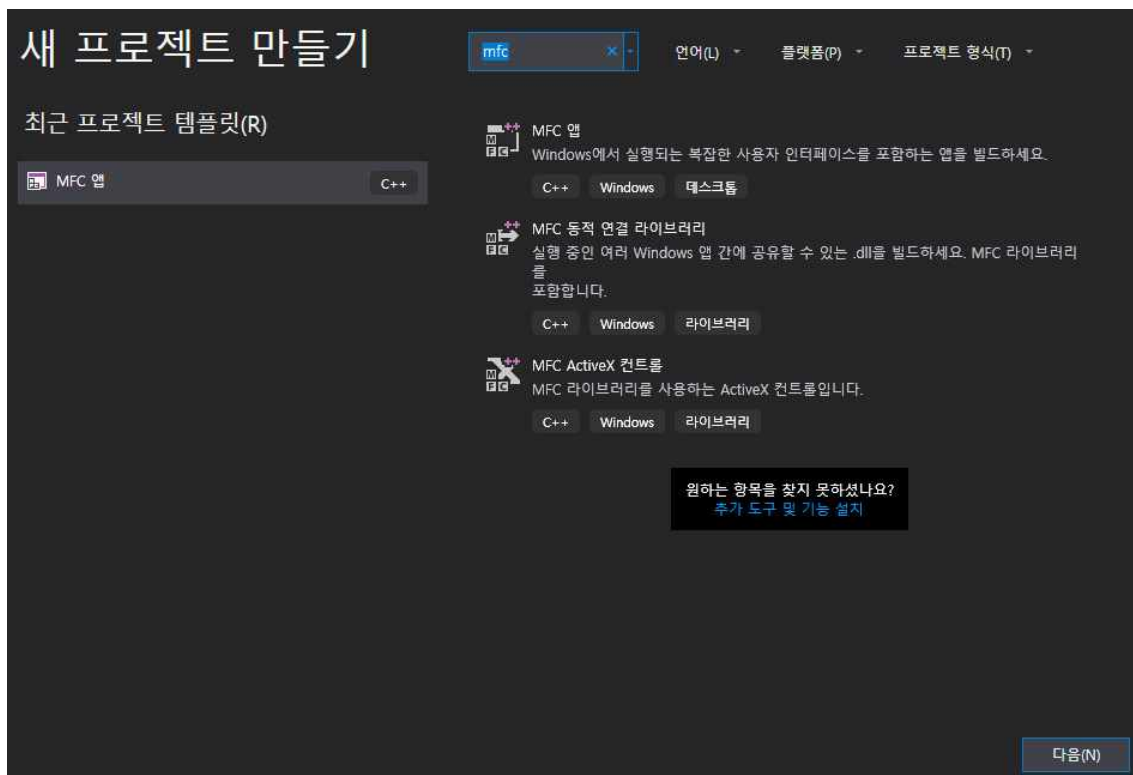
C-2. 코드 작성 및 설명

## A. 계산기 만들기

### A-1. 프로젝트 만들기



1. 비주얼 스튜디오를 켜고 상단 메뉴에서 **파일(F) -> 새로 만들기(N) -> 프로젝트(P)**를 클릭한다.



2. 검색창에 mfc를 입력하면 나오는 첫 번째 항목 **MFC 앱**을 클릭한다.

## 새 프로젝트 구성

MFC 앱 C++ Windows 데스크톱

프로젝트 이름(N)  
by\_190916\_02

위치(L)  
C:\Users\User\source\repos

솔루션 이름(M) ⓘ  
by\_190916\_02

☐ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

뒤로(B) 만들기(C)

3. 새 프로젝트 구성에서 **프로젝트 이름**을 입력하고 만들기를 누른다. 프로젝트 이름을 입력하면 솔루션 이름은 자동으로 완성이 된다.

## MFC 애플리케이션

애플리케이션 종류 옵션

애플리케이션 종류  
여러 문서

단일 문서  
여러 문서  
대화 상자 기반  
여러 최상위 문서

대화 상자 기반 옵션(O)  
<없음>

복합 문서 지원  
<없음>

문서 지원 옵션:  
☐ 활성 문서 서버(A)  
☐ 활성 문서 컨테이너(D)  
☐ 복합 파일 지원(U)

프로젝트 스타일  
Visual Studio

비주얼 스타일 및 색(Y)  
Visual Studio 2008

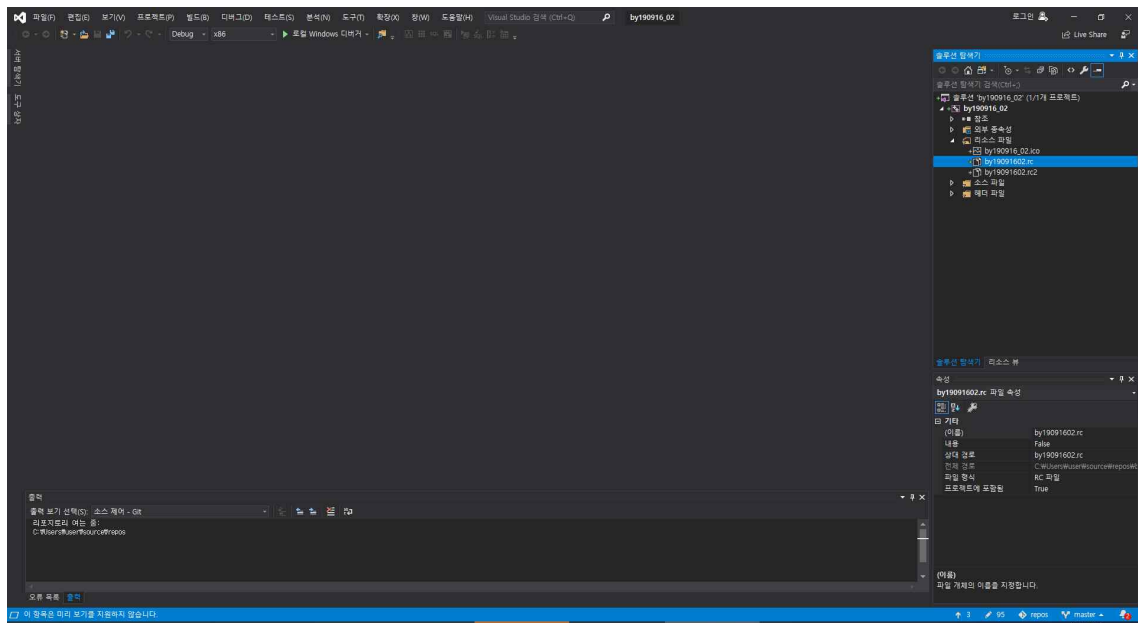
☒ 비주얼 스타일 전환 사용(C)

리소스 언어(L)  
English (United States)

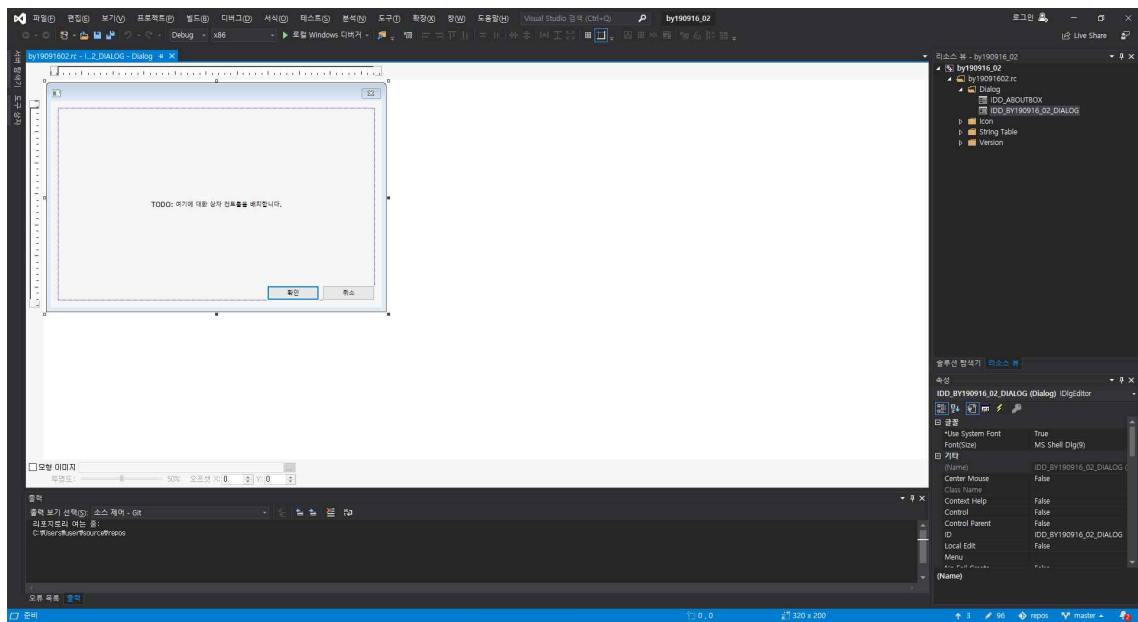
MFC 사용  
공유 DLL에서 MFC 사용

이전 다음 마침 취소

4. 애플리케이션 종류에서 대화 상자 기반을 선택 한 뒤 마침 버튼을 누른다.

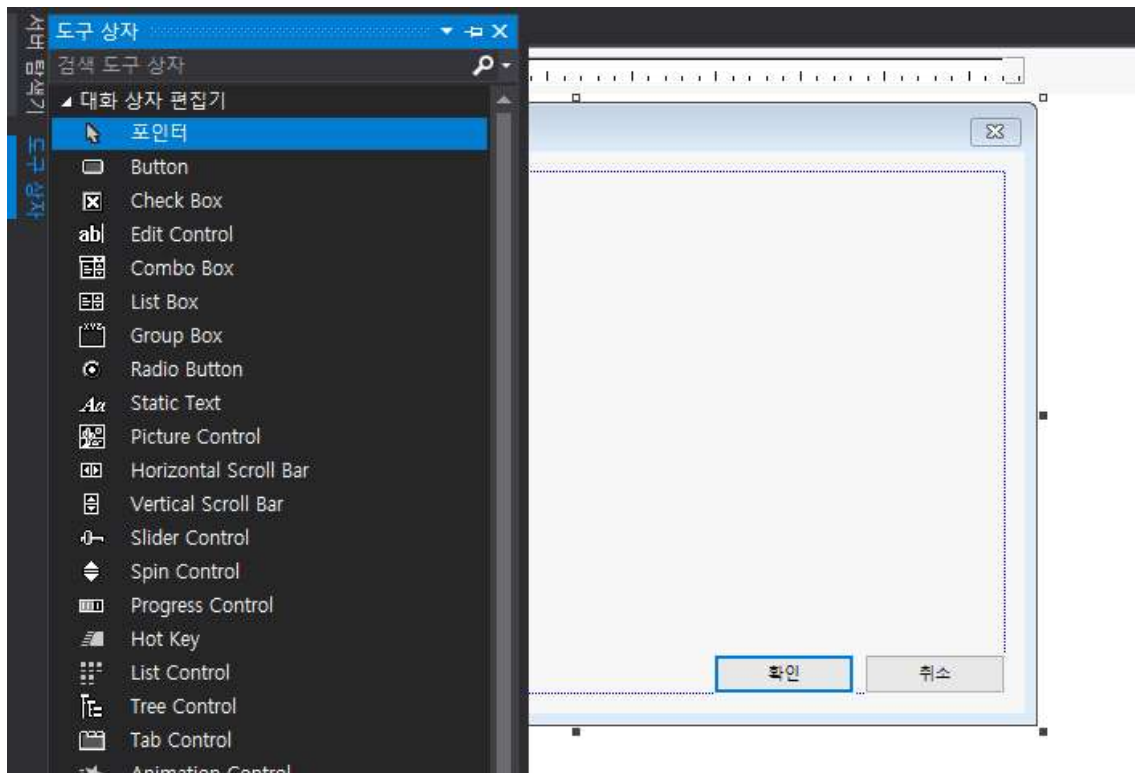


5. 오른쪽 상단 솔루션 탐색기에서 리소스 파일 폴더 안에 확장자가 rc로 끝나는 파일을 더블 클릭한다.



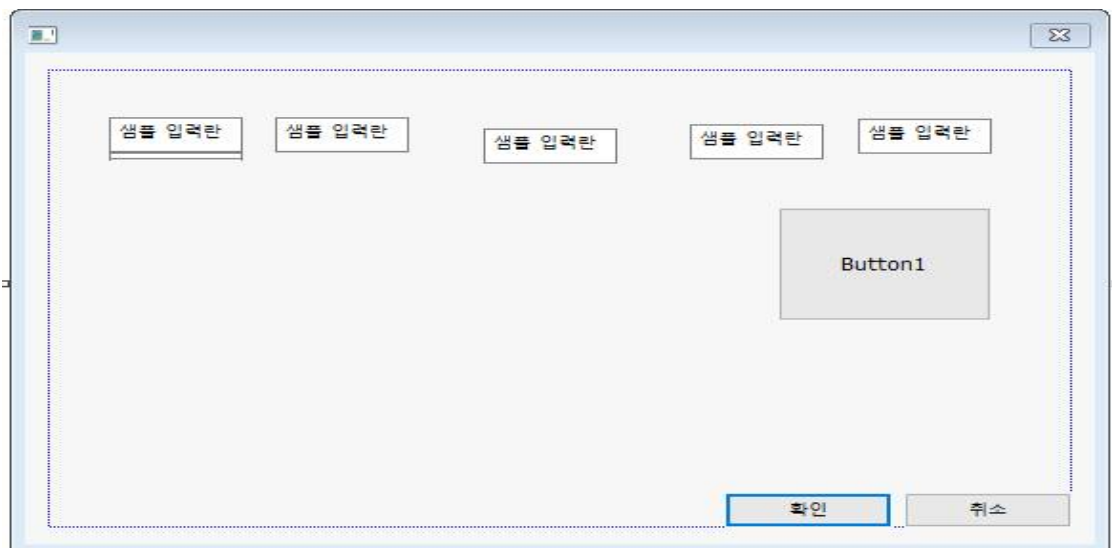
6. 리소스 뷰에서 Dialog 폴더에 파일명이 DIALOG로 끝나는 파일을 열면 다음과 같은 창이 열리며 프로젝트 생성 및 개발 준비가 완료 된다.

## A-2. 화면에 배치하기

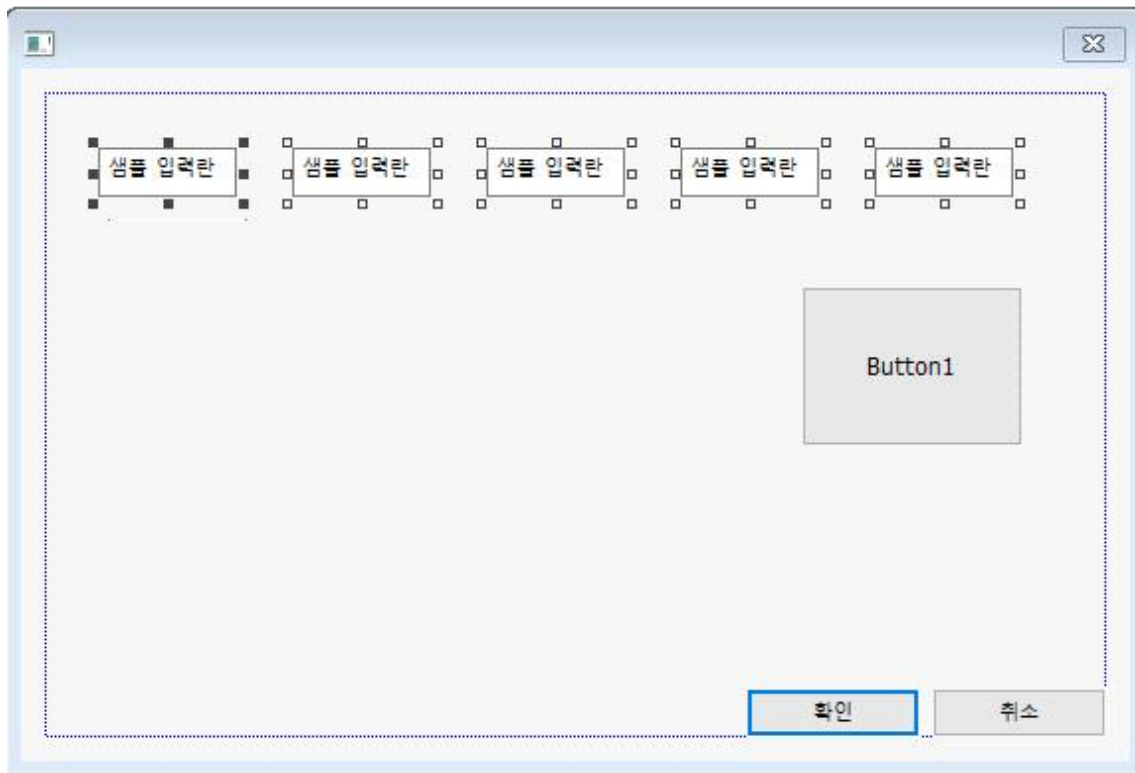


1. 회색창에서 TODO: 여기에 대화 상자 컨트롤들을 배치합니다.를 클릭하고 Delete 키를 눌러 제거해 준다. 그리고 왼쪽에 도구상자를 누르면 사용할 수 있는 컨트롤들이 나오는데 드래그를 통해 사용할 수 있다.

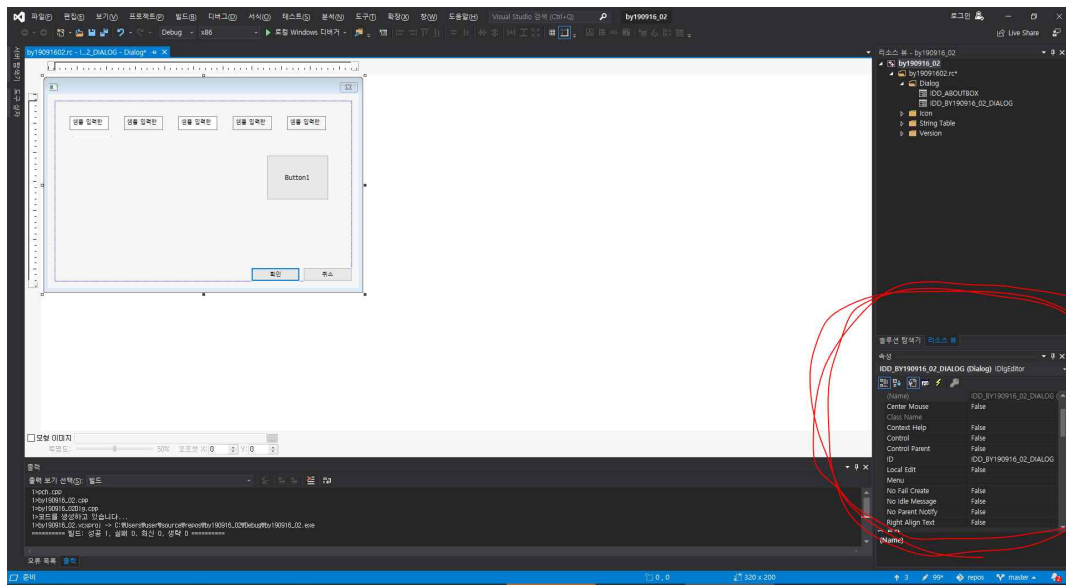
컨트롤이란? MFC에서 다이얼로그창 위에 배치할 수 있는 버튼, 글상자 등을 말함



2. 위와 같이 edit\_control과 button을 배치한다.

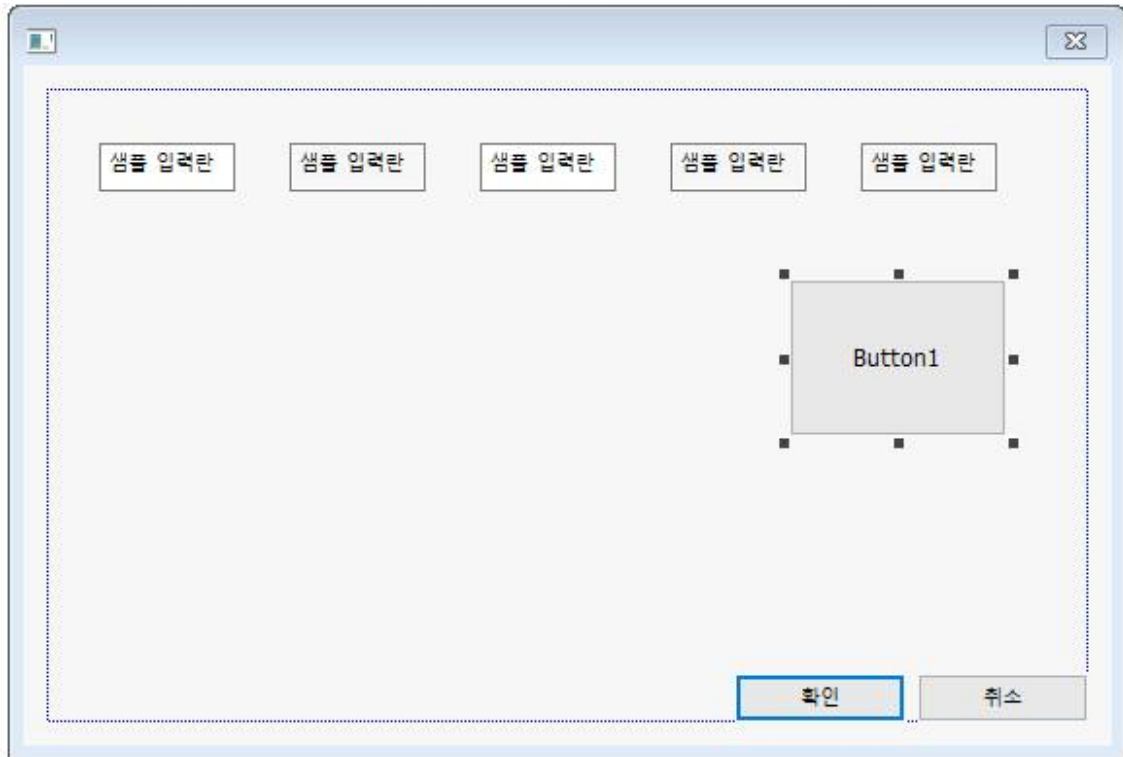


3. 드래그나 ctrl + 마우스 왼쪽으로 선택을 통해 5개의 edit\_control을 선택 한 후 차례대로 F9, alt + 화살표 위쪽, alt + 화살표 오른쪽을 눌러주면 세로로 중간으로 정렬되면 너비, 높이가 정렬이 된다.



4. 화면 오른쪽 하단에 속성들을 제어 할 수 있는 창이 나오는데 2번째, 4번째, 5번째 edit\_control을 3단계 과정과 같이 클릭하고 read\_only속성을 true로 바꿔준다.

### A-3. 버튼을 눌렀을 때 덧셈 수행하기



1. 버튼을 더블클릭한다.

```
by190916_02Dlg.cpp* -# X by19091602.rc - I...2_DIALOG - Dialog+
by190916_02
138         int x = (rect.Width() - cxIcon + 1) / 2;
139         int y = (rect.Height() - cyIcon + 1) / 2;
140
141         // 아이콘을 그립니다.
142         dc.DrawIcon(x, y, m_hIcon);
143
144     else
145     {
146         CDialogEx::OnPaint();
147     }
148 }
149
150 // 사용자가 최소화된 창을 끄는 동안에 커서가 표시되도록 시스템에서
151 // 이 함수를 호출합니다.
152 HCURSOR Cby19091602Dlg::OnQueryDragIcon()
153 {
154     return static_cast<HCURSOR>(m_hIcon);
155 }
156
157
158
159 void Cby19091602Dlg::OnBnClickedButton1()
160 {
161     // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
162 }
163
164
```

2. 위와 같이 창이 뜨면 빨간색 밑줄이 그어진 곳에 코드를 입력한다.

SetDlgItemInt(IDC\_EDIT5, GetDlgItemInt(IDC\_EDIT1)+ GetDlgItemInt(IDC\_EDIT3));

```

74  BOOL Cby19091602Dlg::OnInitDialog()
75  {
76      CDialogEx::OnInitDialog();
77
78      // 시스템 메뉴에 "정보..." 메뉴 항목을 추가합니다.
79
80      // IDM_ABOUTBOX는 시스템 명령 범위에 있어야 합니다.
81      ASSERT((IDM_ABOUTBOX & 0xFFFF) == IDM_ABOUTBOX);
82      ASSERT(IDM_ABOUTBOX < 0xF000);
83
84      CMenu* pSysMenu = GetSystemMenu(FALSE);
85      if (pSysMenu != nullptr)
86      {
87          BOOL bNameValid;
88          CString strAboutMenu;
89          bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
90          ASSERT(bNameValid);
91          if (!strAboutMenu.IsEmpty())
92          {
93              pSysMenu->AppendMenu(MF_SEPARATOR);
94              pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
95          }
96      }
97
98      // 이 대화 상자의 아이콘을 설정합니다. 응용 프로그램의 주 창이 대화 상자가 아닐 경우에는
99      // 프레임워크가 이 작업을 자동으로 수행합니다.
100      SetIcon(m_hIcon, TRUE); // 큰 아이콘을 설정합니다.
101      SetIcon(m_hIcon, FALSE); // 작은 아이콘을 설정합니다.
102
103      // TODO: 여기에 추가 초기화 작업을 추가합니다.
104
105      return TRUE; // 포커스를 컨트롤에 설정하지 않으면 TRUE를 반환합니다.
106
107

```

3.스크롤을 위로 올려서 104번째 줄에 OnInitDialog 함수의 해당 줄을 찾아가서 코드를 입력한다.

```
SetDlgItemTextW(IDC_EDIT2, _T("+"));
```

```
SetDlgItemTextW(IDC_EDIT4, _T("="));
```

```
SetDlgItemTextW(IDC_EDIT5, _T("Result"));
```

4. ctrl + F5를 눌러 실행을 하면 성공!

#### A-4. 함수 설명

SetDlgItemInt(컨트롤 ID, 정수);

컨트롤ID와 일치하는 해당 컨트롤에 정수값을 넣는다.

SetDlgItemInt(컨트롤 ID);

컨트롤ID와 일치하는 해당 컨트롤에 정수값을 가져온다.

SetDlgItemTextW(컨트롤 ID, 텍스트);

컨트롤ID와 일치하는 해당 컨트롤에 텍스트를 넣는다.

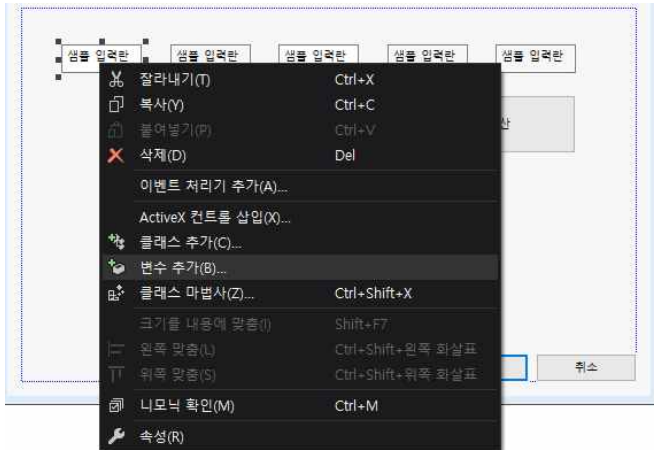
\_T("텍스트");

컨트롤에 넣을 수 있는 텍스트를 생성한다.

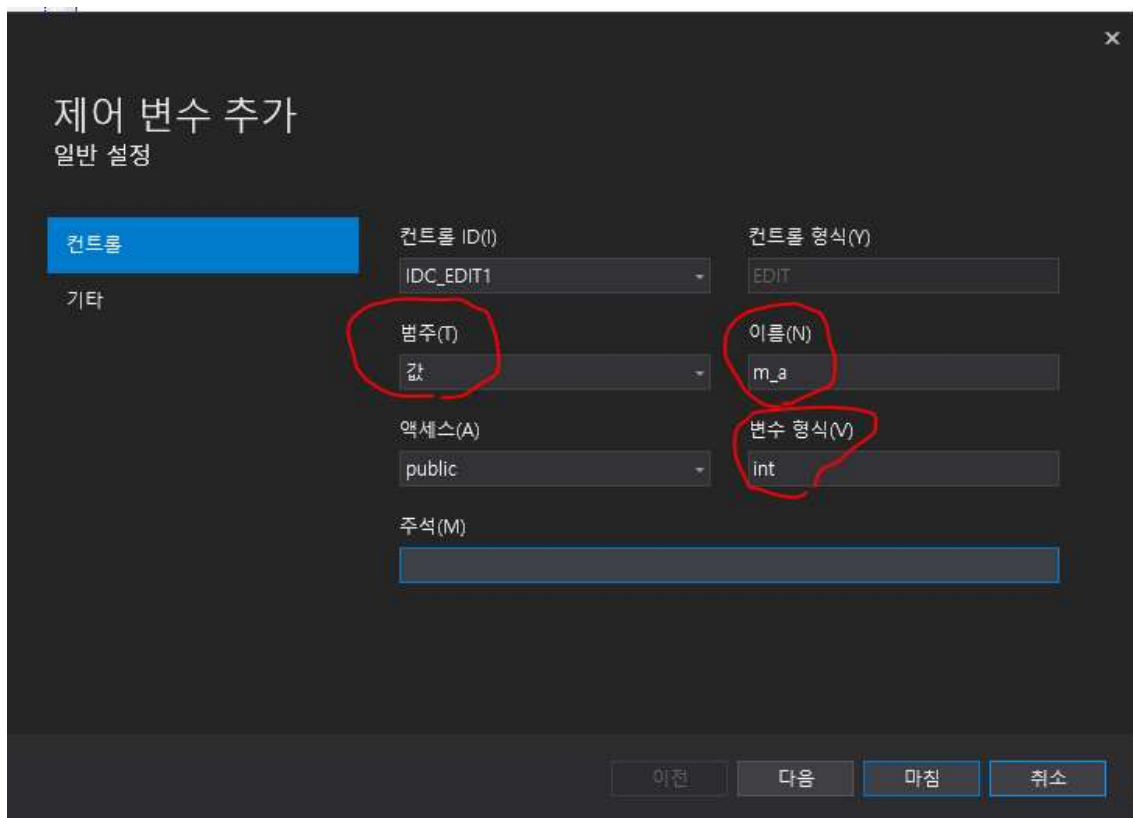


## B. 계산기에 UpdateData() 사용하기

### B-1. 컨트롤에 변수 추가하기

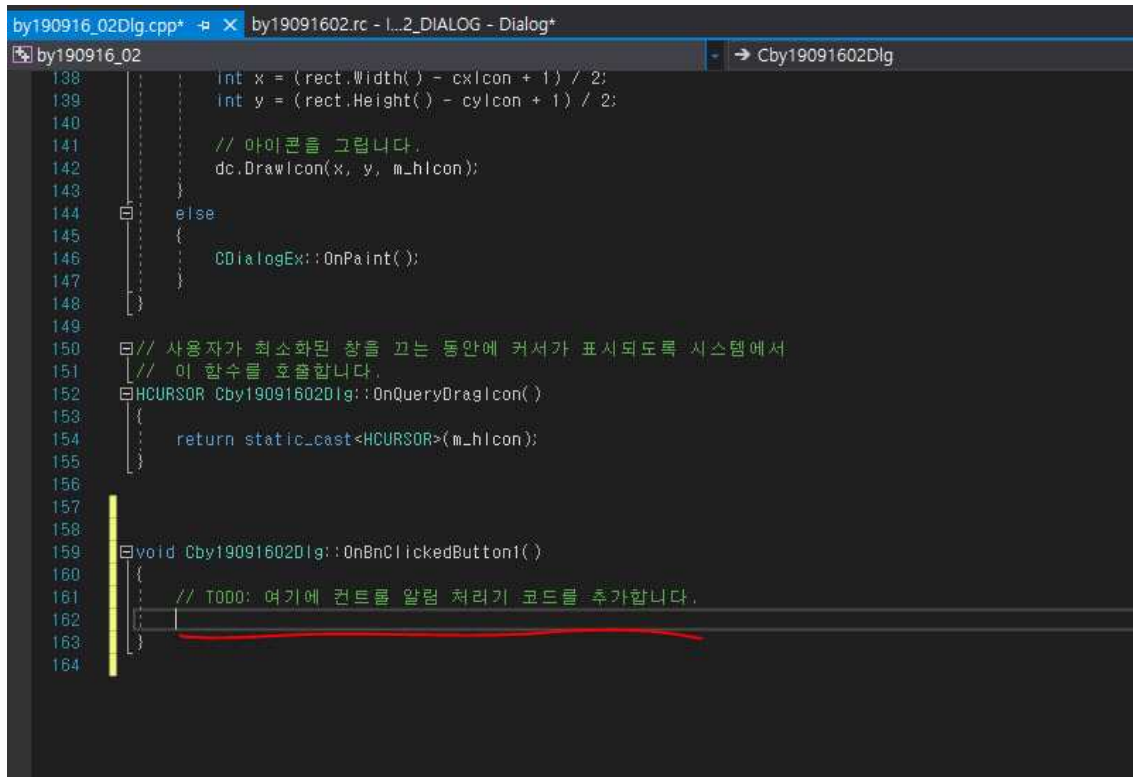


1. edit\_control 컨트롤을 마우스 오른쪽으로 클릭후 변수 추가(B)...를 클릭



2. 범주를 값으로 바꾼 후 이름은 m\_a로 변수 형식은 int로 바꾸어 준다. 마찬가지로 3번째 edit\_control도 바꾸며 이름만 m\_b로 5번째 edit\_control은 이름만 m\_c로 바꾸어 준다.

1. 버튼을 더블클릭한다.



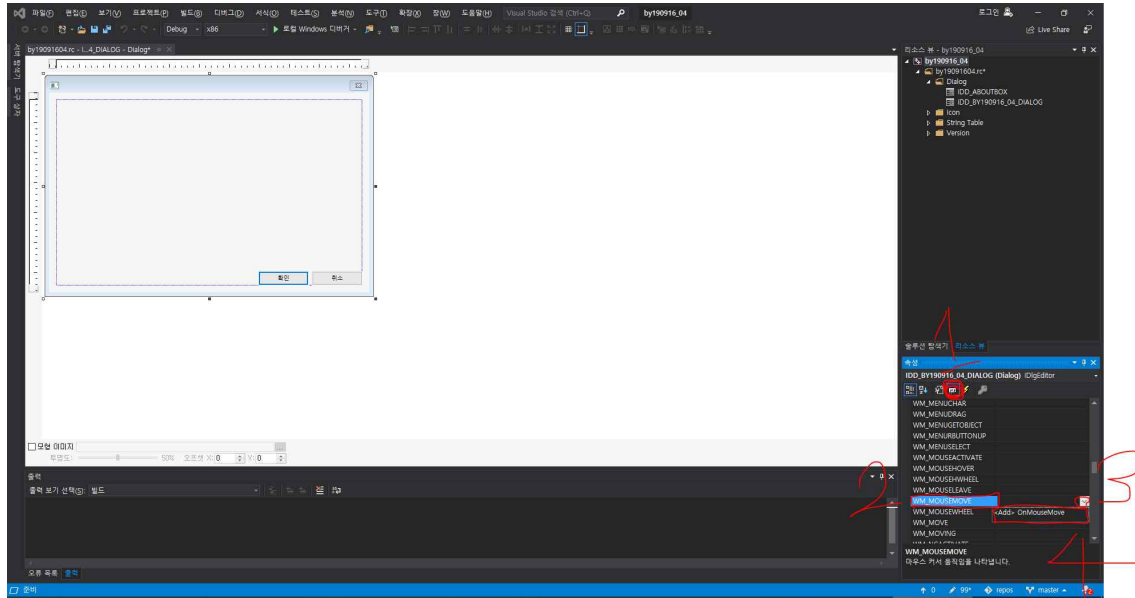
```
138         int x = (rect.Width() - cxIcon + 1) / 2;
139         int y = (rect.Height() - cyIcon + 1) / 2;
140
141         // 아이콘을 그립니다.
142         dc.DrawIcon(x, y, m_hIcon);
143     }
144     else
145     {
146         CDialogEx::OnPaint();
147     }
148 }
149
150 // 사용자가 최소화된 창을 끄는 동안에 커서가 표시되도록 시스템에서
151 // 이 함수를 호출합니다.
152 HCURSOR Cby19091602Dlg::OnQueryDragIcon()
153 {
154     return static_cast<HCURSOR>(m_hIcon);
155 }
156
157
158
159 void Cby19091602Dlg::OnBnClickedButton1()
160 {
161     // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
162 }
163
164
```

2. 위와 같이 창이 뜨면 빨간색 밑줄이 그어진 곳에 코드를 입력한다.

```
pdateData(true);
m_c = m_a + m_b;
UpdateData(false);
```

## C. 다이얼로그 창에 펜으로 그림 그리기

### C-1. MOUSEMOVE 메시지에 대한 함수 생성



1. 다이얼로그를 클릭 후 오른쪽 상단의 메시지 탭에서 WM\_MOUSEMOVE를 클릭 후 오른쪽의 아랫방향 화살표 클릭 후 <Add> OnMouseMove를 클릭

## C-2. 코드 작성 및 설명

```
int size = 16;
CPoint pnt;
void Cby19091604Dlg::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: 여기에 메시지 처리기 코드를 추가 및/또는 기본값을 호출합니다.
    if (nFlags == MK_LBUTTON) {
        CClientDC dc(this);
        CPen pen(PS_SOLID, size, RGB(0, 255, 0));
        dc.SelectObject(&pen);
        dc.MoveTo(pnt);
        dc.LineTo(point);
    }
    pnt = point;
    CDialogEx::OnMouseMove(nFlags, point);
}
```

### 1. 다음과 같이 코드 입력

1. size와 pnt를 함수의 바깥에 선언한 이유는 함수가 선언 된 뒤 사라지지 않게 하기 위함이다.
2. 마우스가 움직였을 때 nFlags를 통해 마우스 왼쪽 버튼을 누르고 있는 지 알 수 있다. 마우스가 눌렀다면 if문 안에서 nFlags == MK\_LBUTTON가 true가 되어 if문 안의 블록이 실행된다.
3. CClientDC 클래스를 통해 선을 긋는다.
4. CPen의 속성을 수정하면 여러 가지 효과를 줄 수 있다.