

윈도우프로그래밍

Lab 08

- 4.2절 예제 덧셈 / 곱셈 문제 프로그램 만들기 -

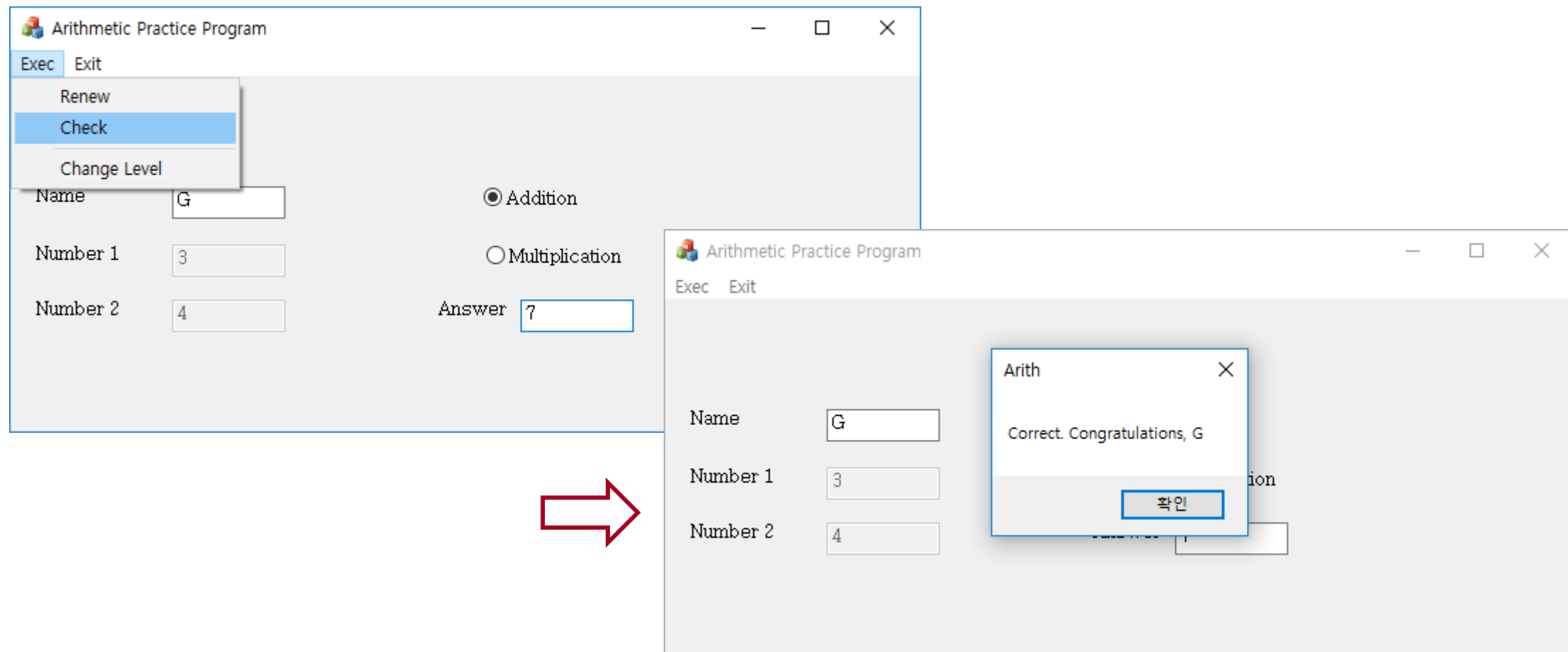
김한슬

uo3359@sookmyung.ac.kr

AppWizard를 사용한 MFC 프로그래밍

- AppWizard를 사용하는 **프로젝트 생성**
- 리소스 편집기를 사용하는 **시각적 설계**(visual design)
- 클래스 마법사(Class Wizard)를 활용하는 **코딩**

4.2절 예제 프로그램

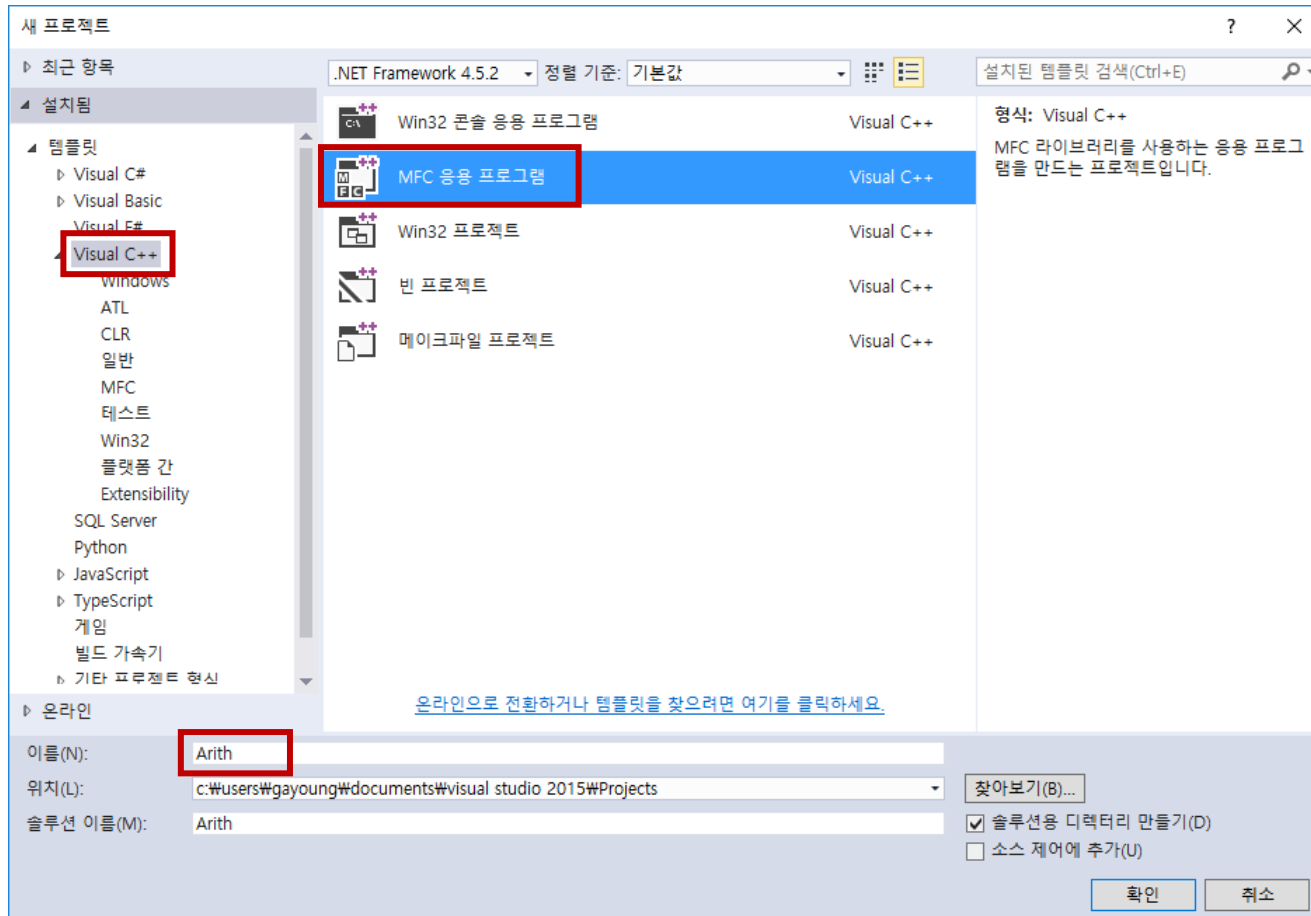


4.2절 예제 프로그램

- 덧셈 / 곱셈 문제 제시하고 답을 검사해주는 프로그램 작성
- 초기 실행 시 Number1 / Number2 옆 에디트 컨트롤에 한 자리 숫자가 무작위로 생성되어 나타남
- Exec 메뉴의 Renew 항목 선택 시 숫자들이 변경됨
- Exec 메뉴의 Change Level 항목 선택 후 Renew 사용 시 두 자리 숫자 나타남, Change Level 항목을 다시 선택 시 한 자리 숫자로 돌아감
- Name 옆 에디트 컨트롤에는 사용자 이름을 입력함
- 라디오 버튼을 이용하여 문제 유형을 선택함(덧셈 or 곱셈)
- Exec 메뉴의 Check 항목 선택 시 답을 확인하여 알맞은 메시지 다이얼로그 박스를 띄움

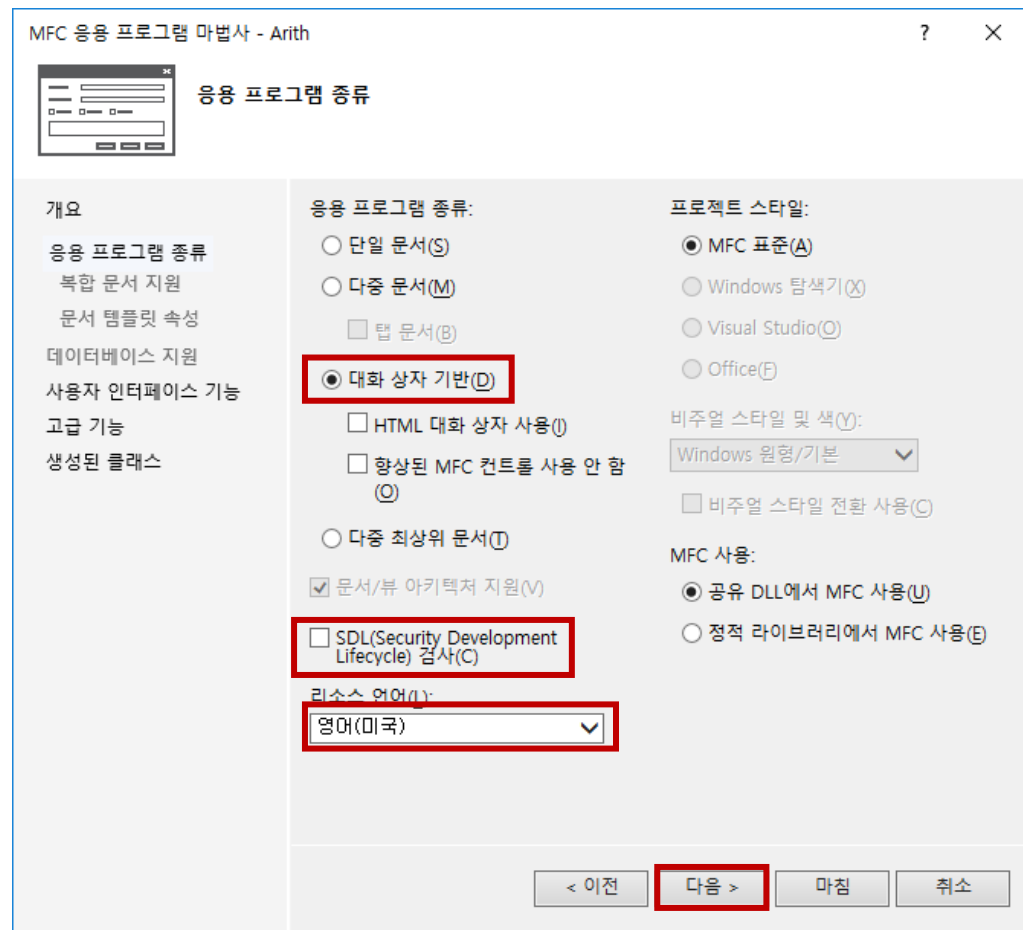
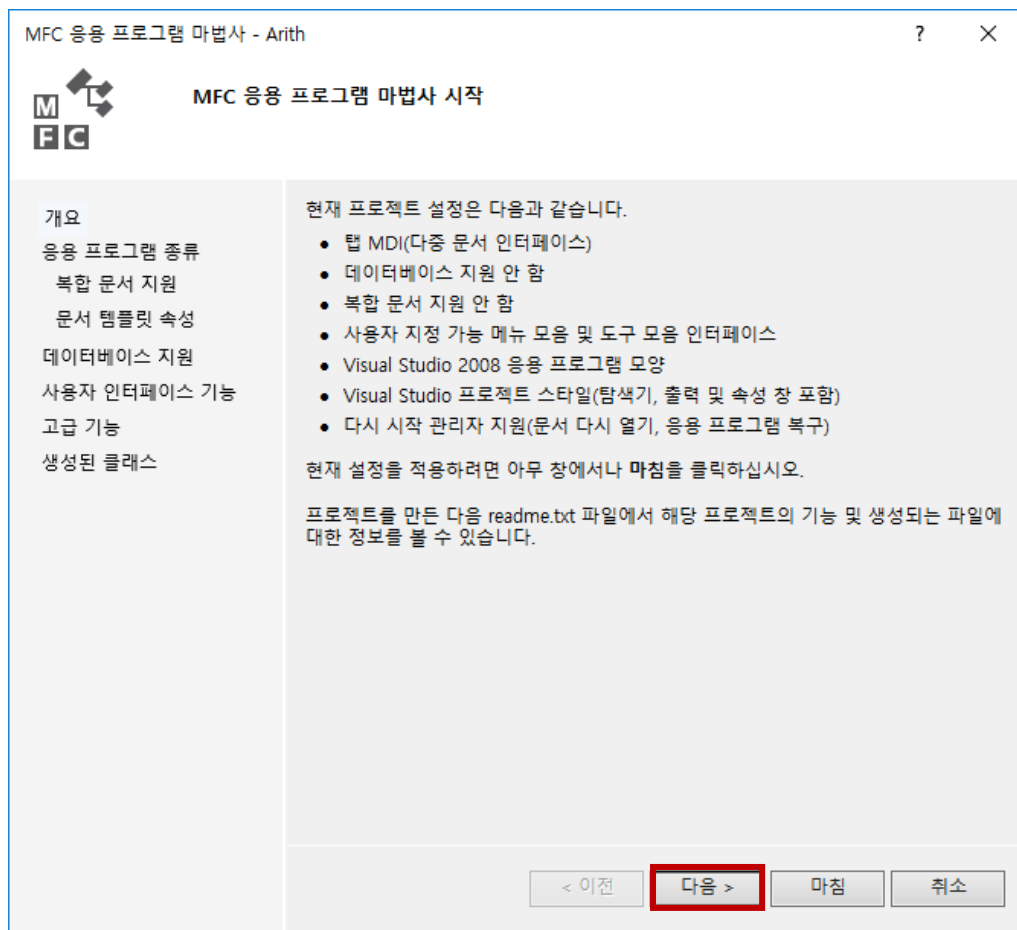
프로젝트 생성

- [파일] - [새로 만들기] - [프로젝트] - <MFC 응용 프로그램>



프로젝트 생성

- [파일] - [새로 만들기] - [프로젝트] - <MFC 응용 프로그램>



AppWizard를 사용하는 프로젝트 생성

- [파일] - [새로 만들기] - [프로젝트] - <MFC 응용 프로그램>

MFC 응용 프로그램 마법사 - Arith

사용자 인터페이스 기능

개요
응용 프로그램 종류
복합 문서 지원
문서 템플릿 속성
데이터베이스 지원
사용자 인터페이스 기능
고급 기능
생성된 클래스

주 프레임 스타일:
☒ 두꺼운 프레임(T)
☐ 최소화 상자(I)
☐ 최대화 상자(A)
☐ 최소화(N)
☐ 최대화(X)
☒ 시스템 메뉴(Y)
☒ 정보 상자(B)
☒ 초기 상태 표시줄(U)
☐ 분할 창(P)
자식 프레임 스타일:
☒ 자식 최소화 상자(M)
☒ 자식 최대화 상자(Z)
☐ 자식 창 최대화(D)

명령 모음(메뉴/도구 모음/리본):
☒ 클래식 메뉴 사용(S)
☒ 도킹 가능한 클래식 도구 모음 사용(K)
☐ 브라우저 스타일 도구 모음 사용(W)
☐ 메뉴 모음 및 도구 모음 사용(O)
☒ 사용자 정의 도구 모음 및 이미지 사용(E)
☒ 개인 설정 메뉴 동작(H)
☐ 리본 사용(R)

대화 상자 제목(G):
Arithmetic Practice Program

< 이전 다음 > 마침 취소

MFC 응용 프로그램 마법사 - Arith

고급 기능

개요
응용 프로그램 종류
복합 문서 지원
문서 템플릿 속성
데이터베이스 지원
사용자 인터페이스 기능
고급 기능
생성된 클래스

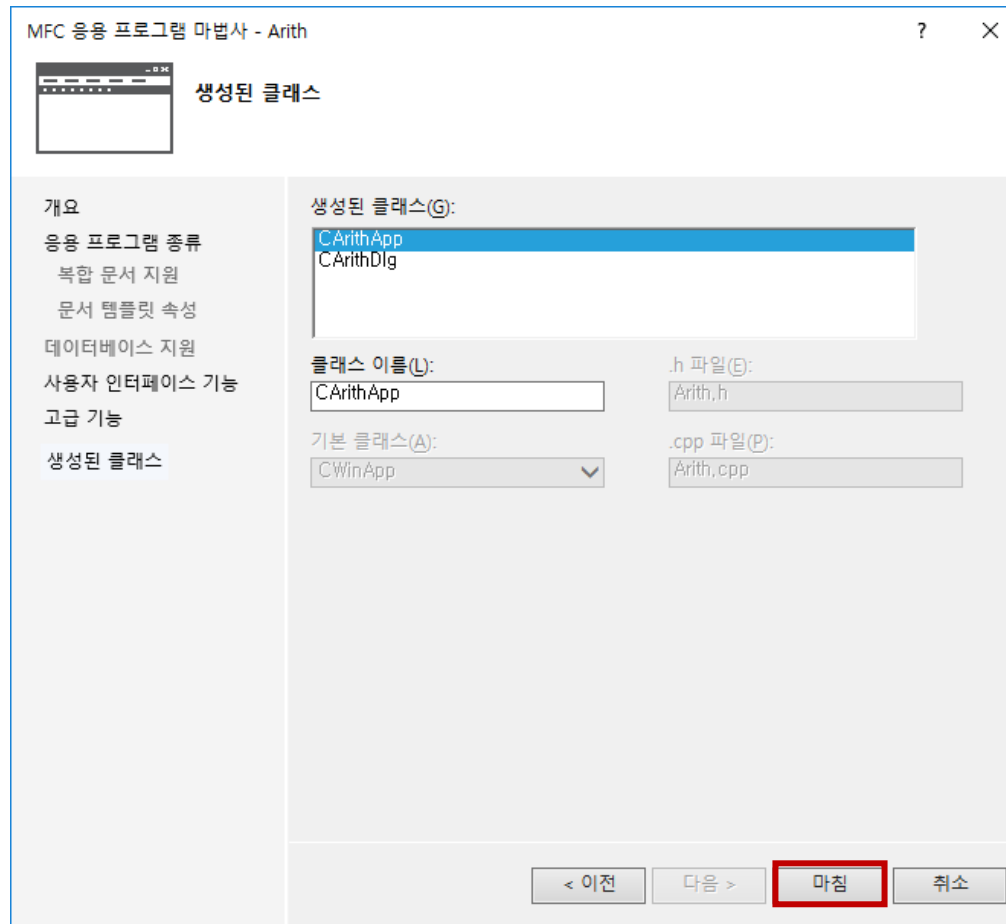
고급 기능:
☐ 인쇄 및 인쇄 미리 보기(P)
☐ 자동화(U)
☒ ActiveX 컨트롤(R)
☐ MAPI(메시징 API)(I)
☐ Windows 소켓(W)
☐ Active Accessibility(A)
☒ 공용 컨트롤 매니페스트(M)
☒ 다시 시작 관리자 지원(G)
☐ 이전에 열려 있던 문서 다시 열기(Y)
☐ 응용 프로그램 복구 지원(V)

고급 프레임 창:
☐ 탐색기 도킹 창(D)
☐ 출력 도킹 창(O)
☐ 속성 도킹 창(S)
☐ 탐색 창(T)
☐ 캡션 표시줄(B)
☐ 창을 표시하거나 활성화하는 고급 프레임 메뉴 항목(F)
최근 파일 목록의 파일 수(N):
4

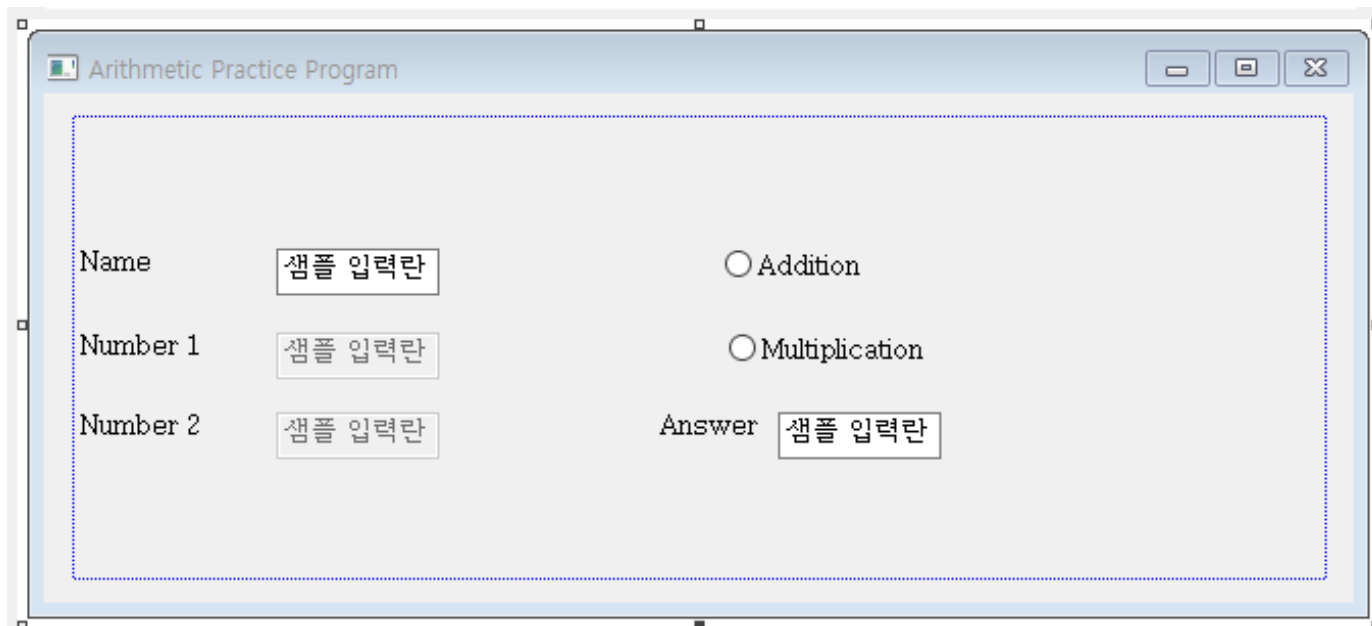
< 이전 다음 > 마침 취소

AppWizard를 사용하는 프로젝트 생성

- [파일] - [새로 만들기] - [프로젝트] - <MFC 응용 프로그램>



시각적 설계 - 다이얼로그 설계



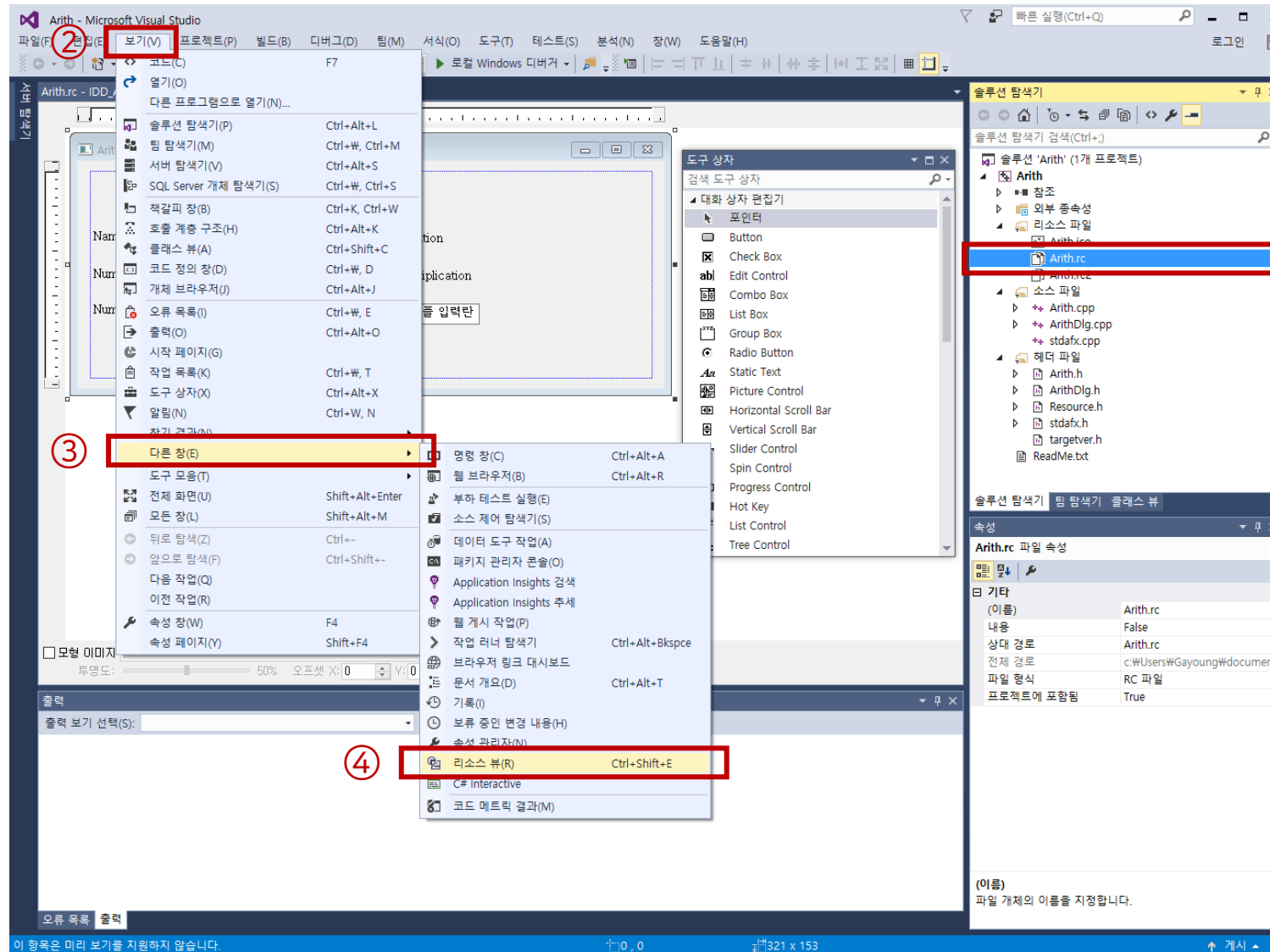
- 4개의 에디트 컨트롤과 이를 설명하는 4개의 텍스트 컨트롤
 - Name, Number1, Number2, Answer
- 2개의 라디오 버튼 그룹
 - Addition, Multiplication

시각적 설계 - 다이얼로그 설계

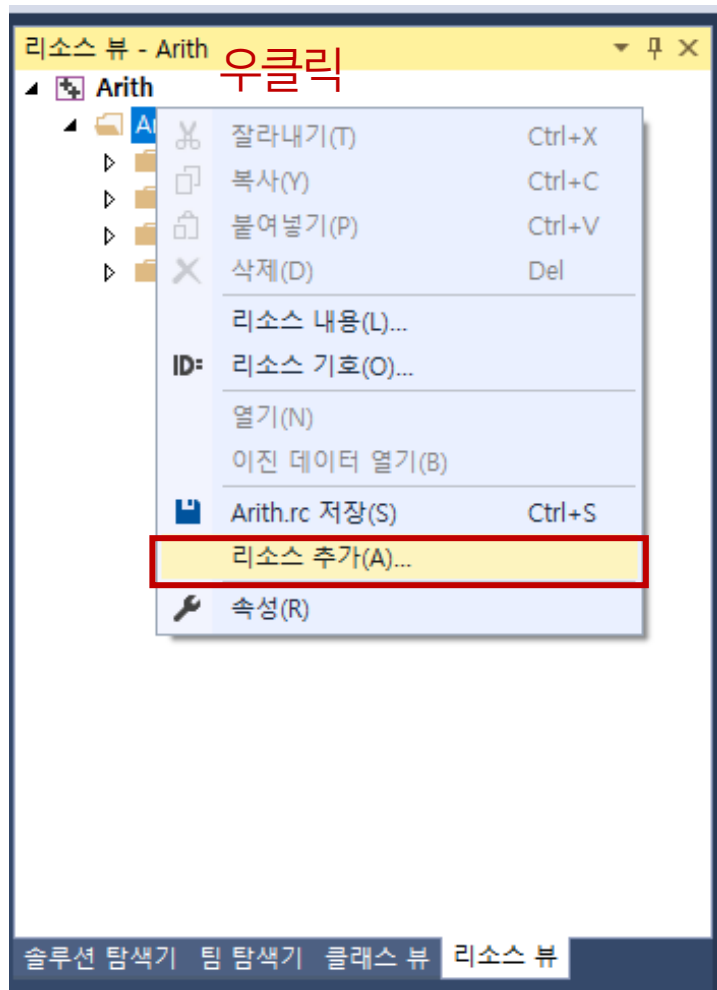
Object	Property	Setting
Dialog Box	ID Caption Font (Size) Minimize box Maximize box	IDD_ARITH_DIALOG Arithmetic Practice Program 바탕(10) True True
Static Text	Caption	Name
Static Text	Caption	Number 1
Static Text	Caption	Number 2
Static Text	Caption	Answer
Edit Box	ID	IDC_NAME_EDIT
Edit Box	ID Disabled	IDC_NUM1_EDIT True
Edit Box	ID Disabled	IDC_NUM2_EDIT True
Edit Box	ID	IDC_ANSWER_EDIT
Radio Button	ID Caption Group	IDC_ADD_RADIO Addition True
Radio Button	ID Caption Group	IDC_MULT_RADIO Multiplication False

- 다이얼로그 설계 시
각 컨트롤들의 속성 설정하기

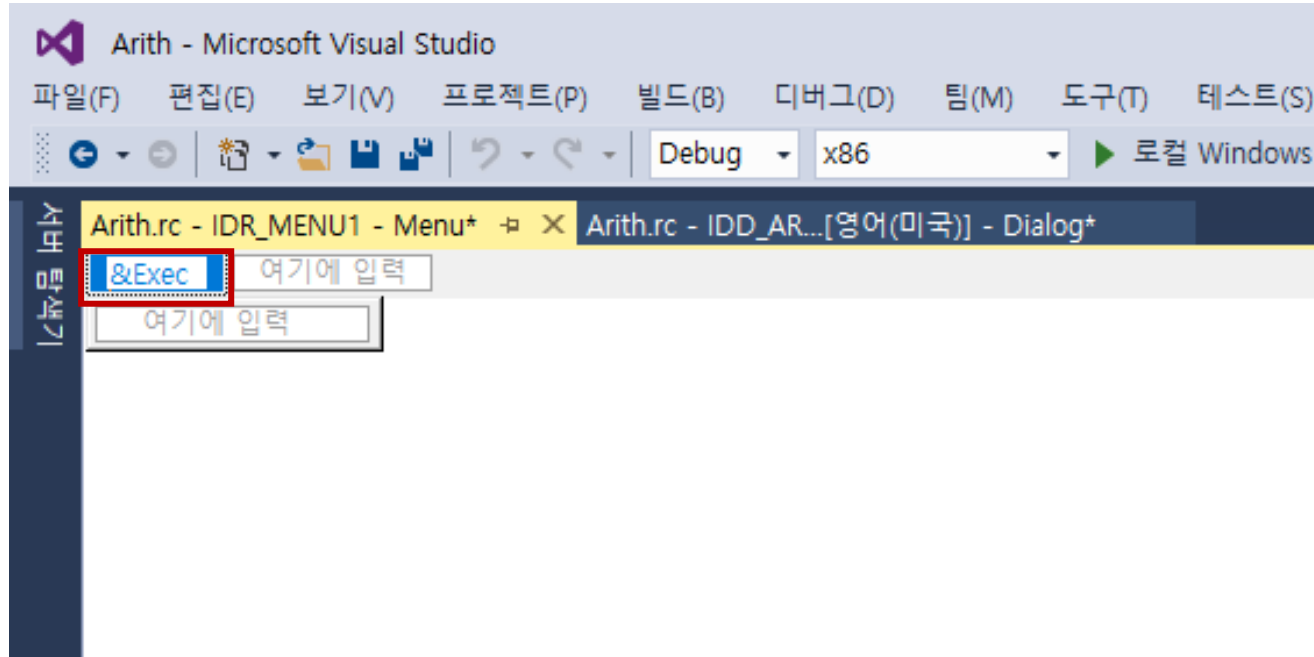
시각적 설계 - 메뉴 설계



시각적 설계 - 메뉴 설계

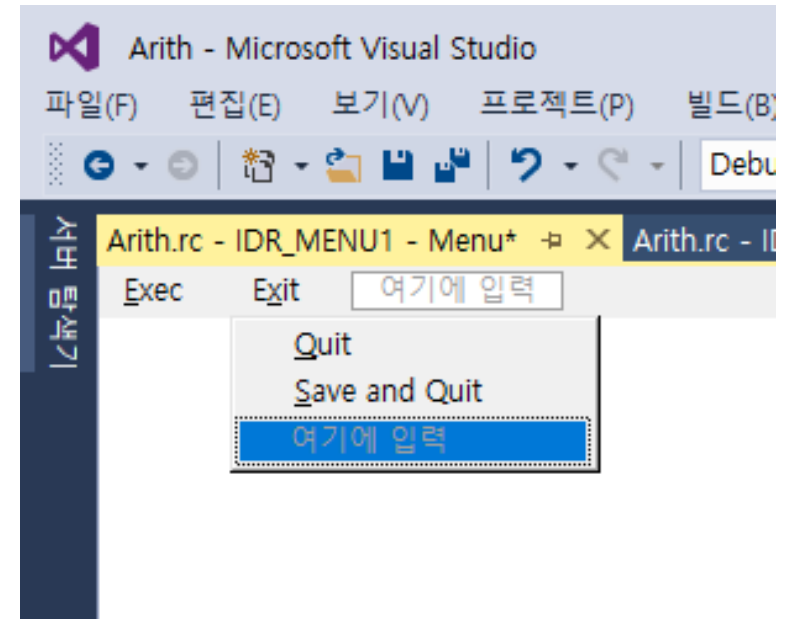
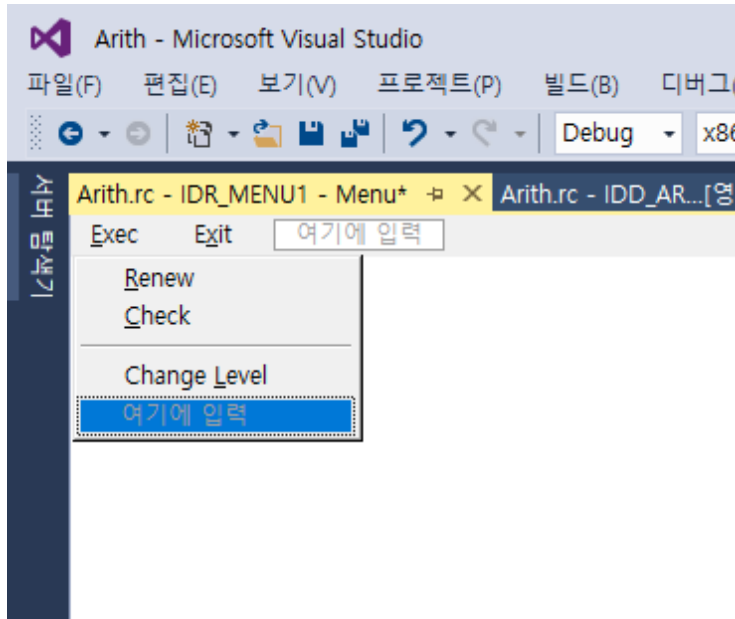
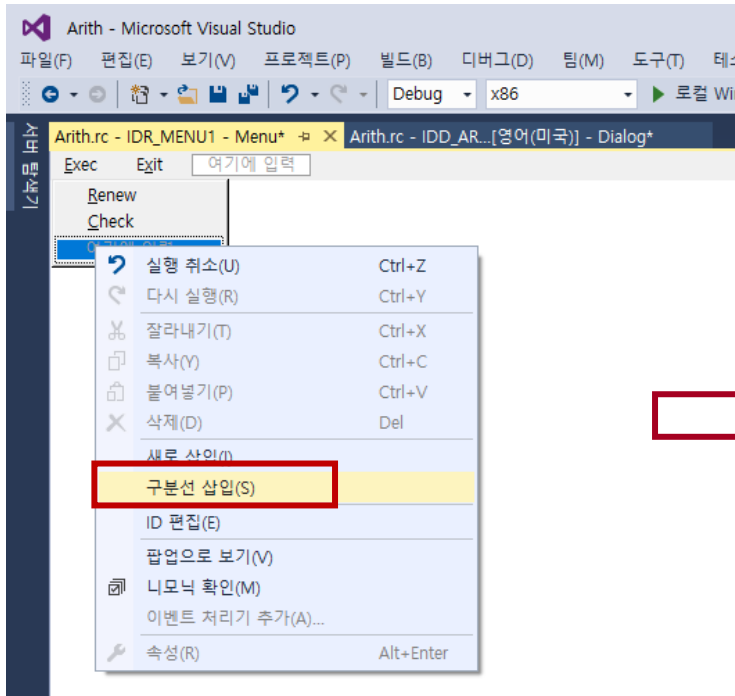


시각적 설계 - 메뉴 설계



- 편의키(mnemonic key)
 - 메뉴 / 메뉴 항목 / 컨트롤 등의 캡션에서 밑줄 표시된 문자
 - > 마우스 대신 키보드를 사용하여 메뉴나 컨트롤에 접근 가능
 - 해당 문자 앞에 & 기호를 덧붙여 설정
- 예) &Exec -> Eexec

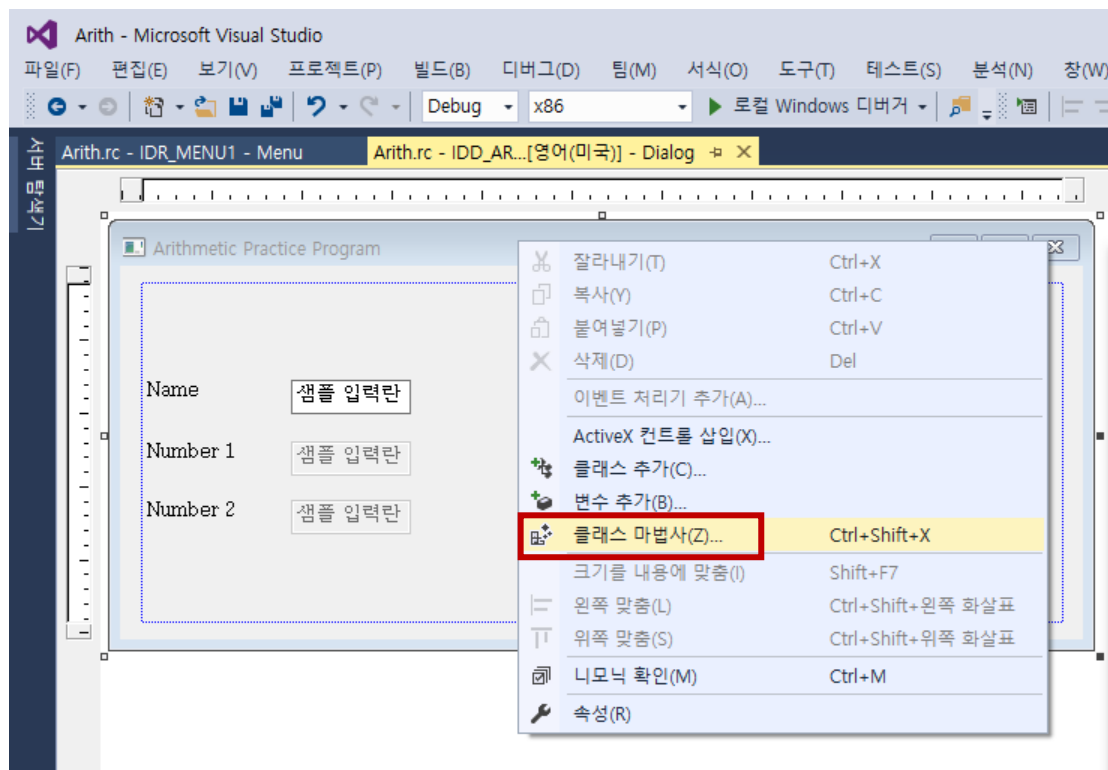
시각적 설계 - 메뉴 설계



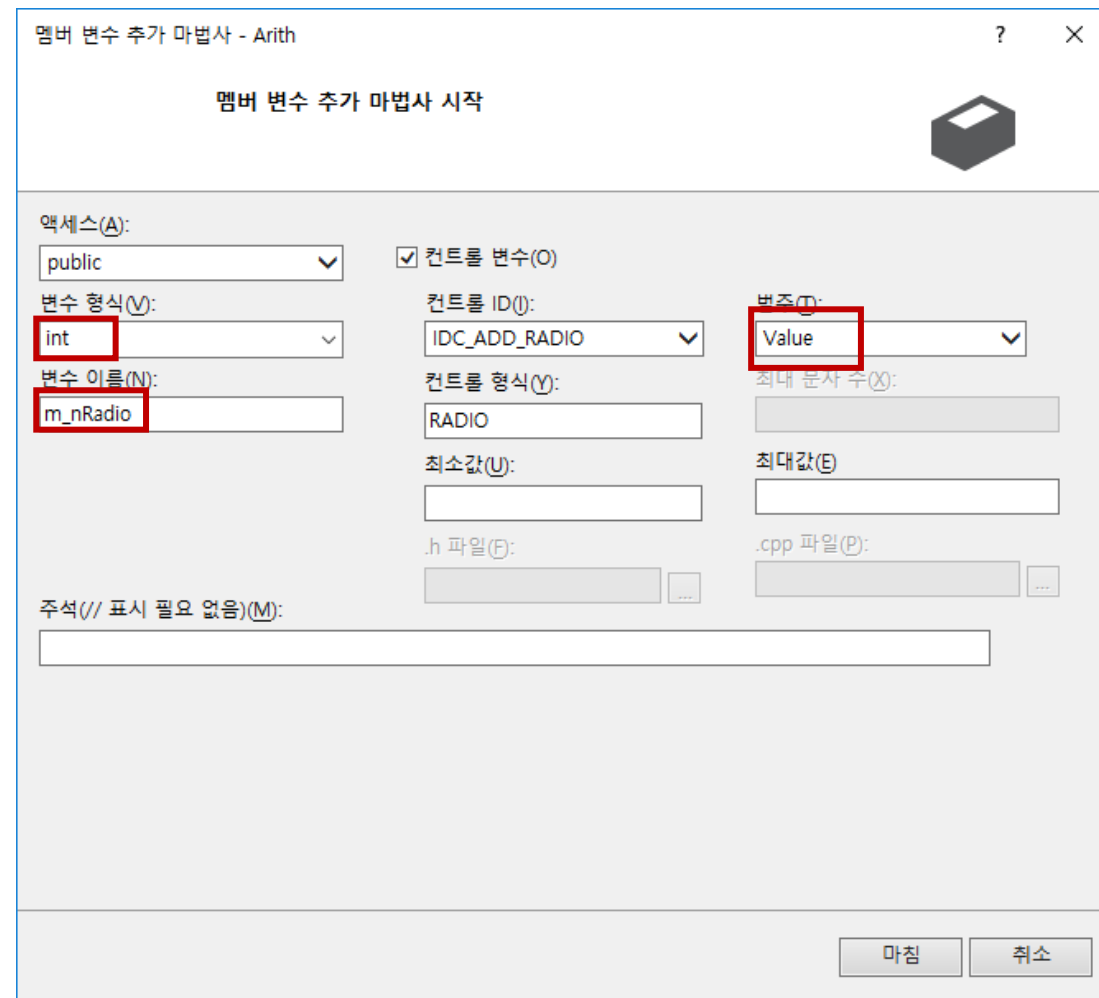
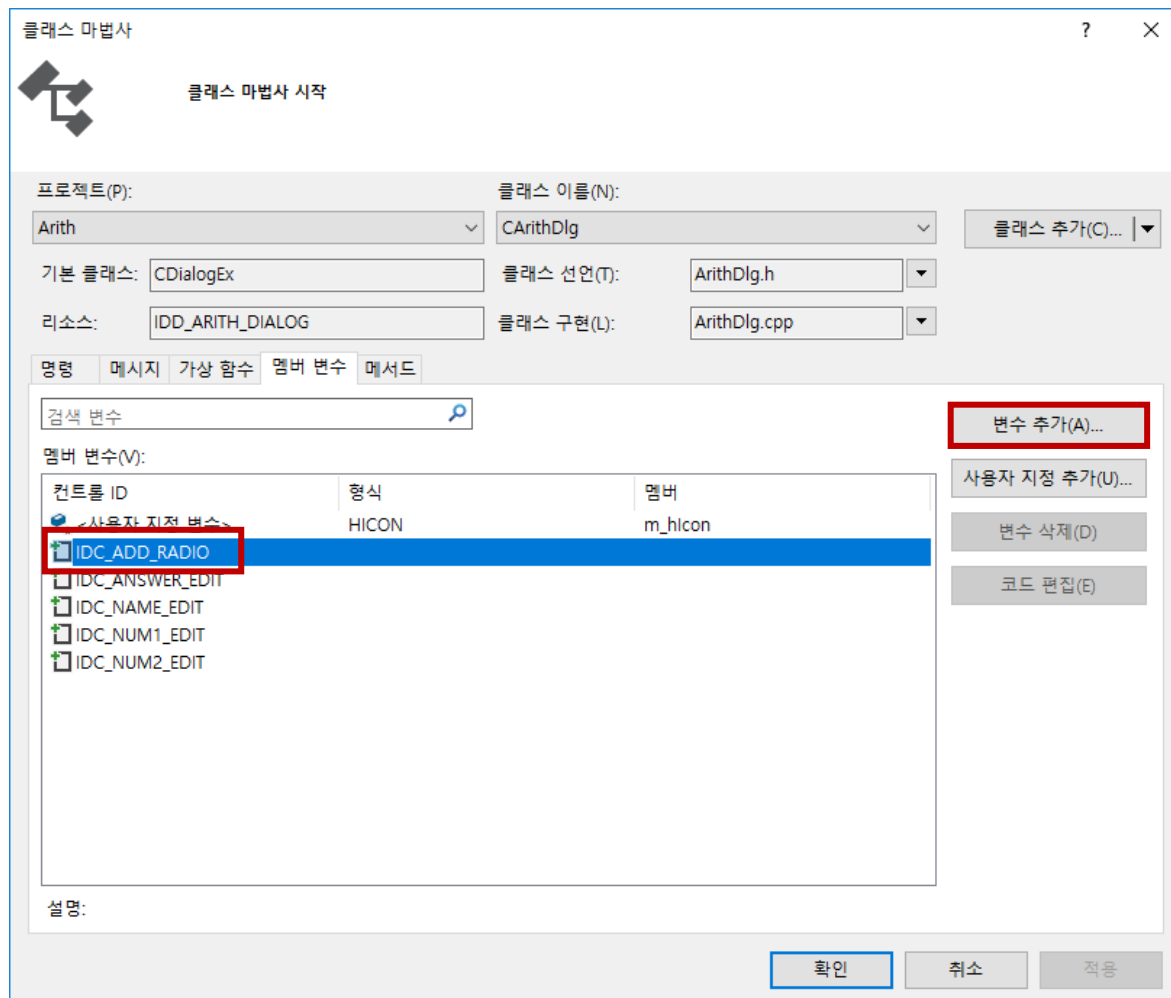
- 구분선 삽입

- 메뉴 설계 완성
 - Exec 메뉴
 - 하위 항목: Renew, Check, Change Level
 - Exit 메뉴
 - 하위 항목: Quit, Save and Quit

클래스 마법사를 사용한 코딩

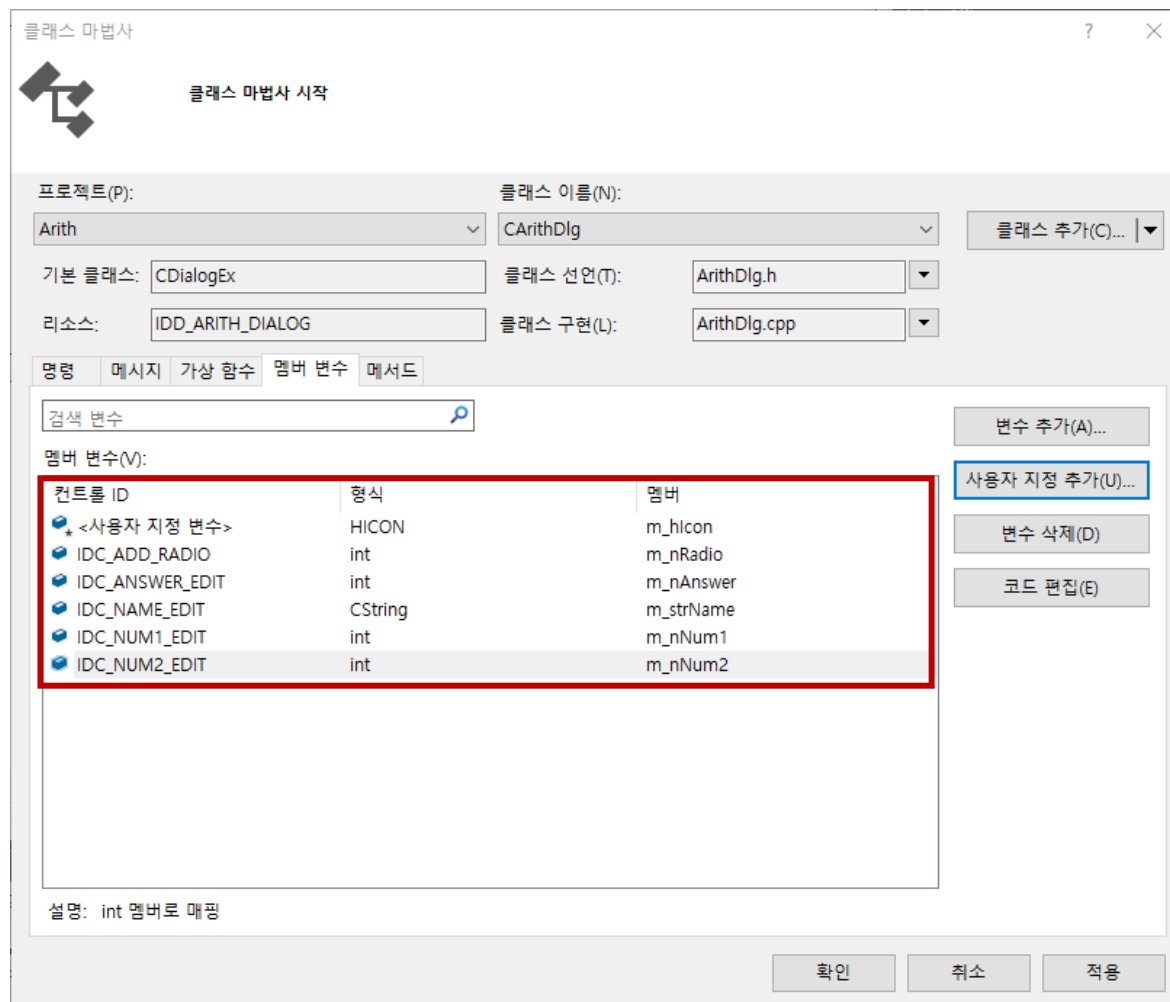


클래스 마법사를 사용한 코딩 - 변수 추가

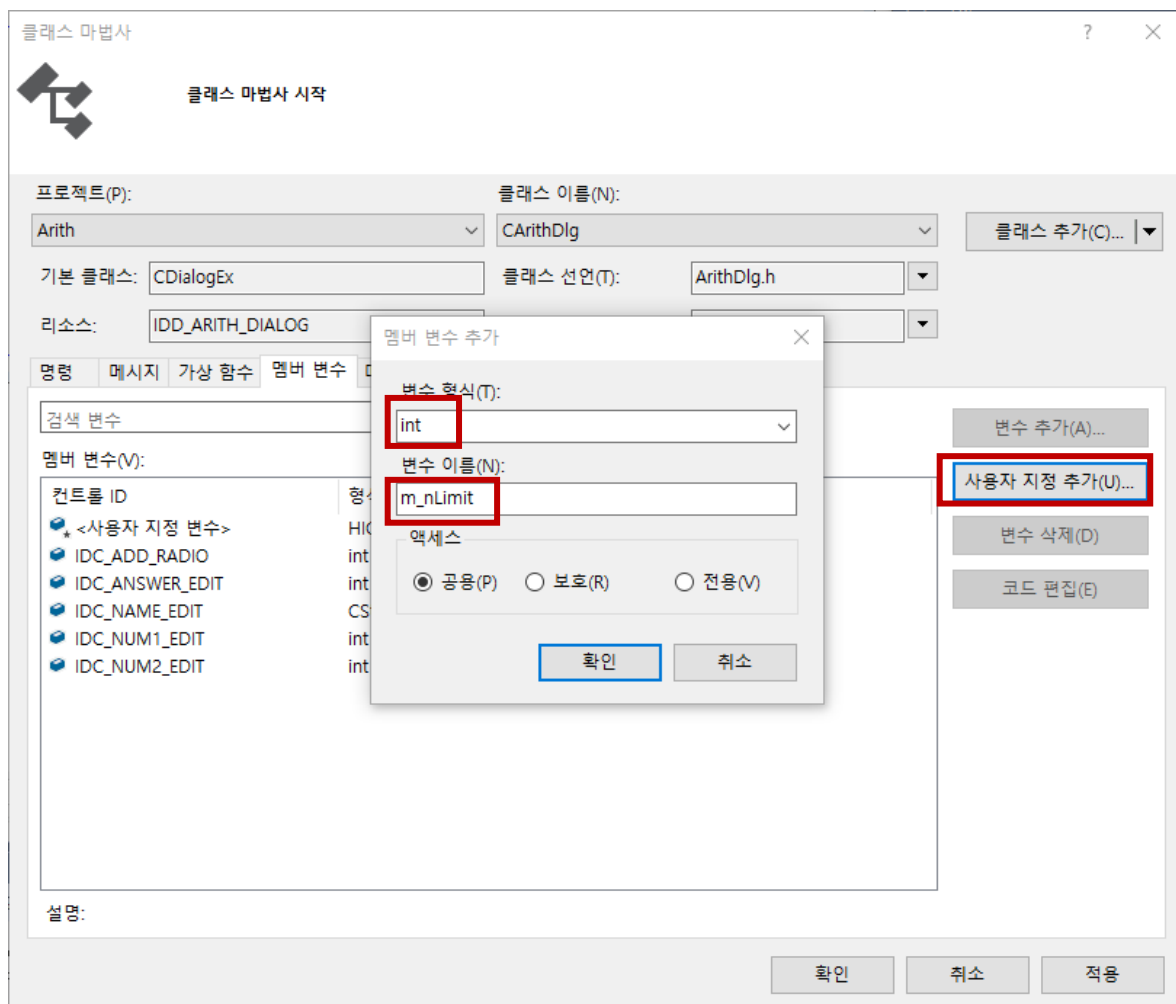


클래스 마법사를 사용한 코딩 - 변수 추가

- 라디오 버튼 그룹
 - m_nRadio (int 타입)
- IDC_NAME_EDIT
 - m_strName(CString 타입)
 - 최대 길이 10
- IDC_NUM1_EDIT
 - m_nNum1 (int 타입)
- IDC_NUM2_EDIT
 - m_nNum2 (int 타입)
- IDC_ANSWER_EDIT
 - m_nAnswer (int 타입)
 - 최솟값 1 / 최댓값 9999
- 공통: 범주 - value

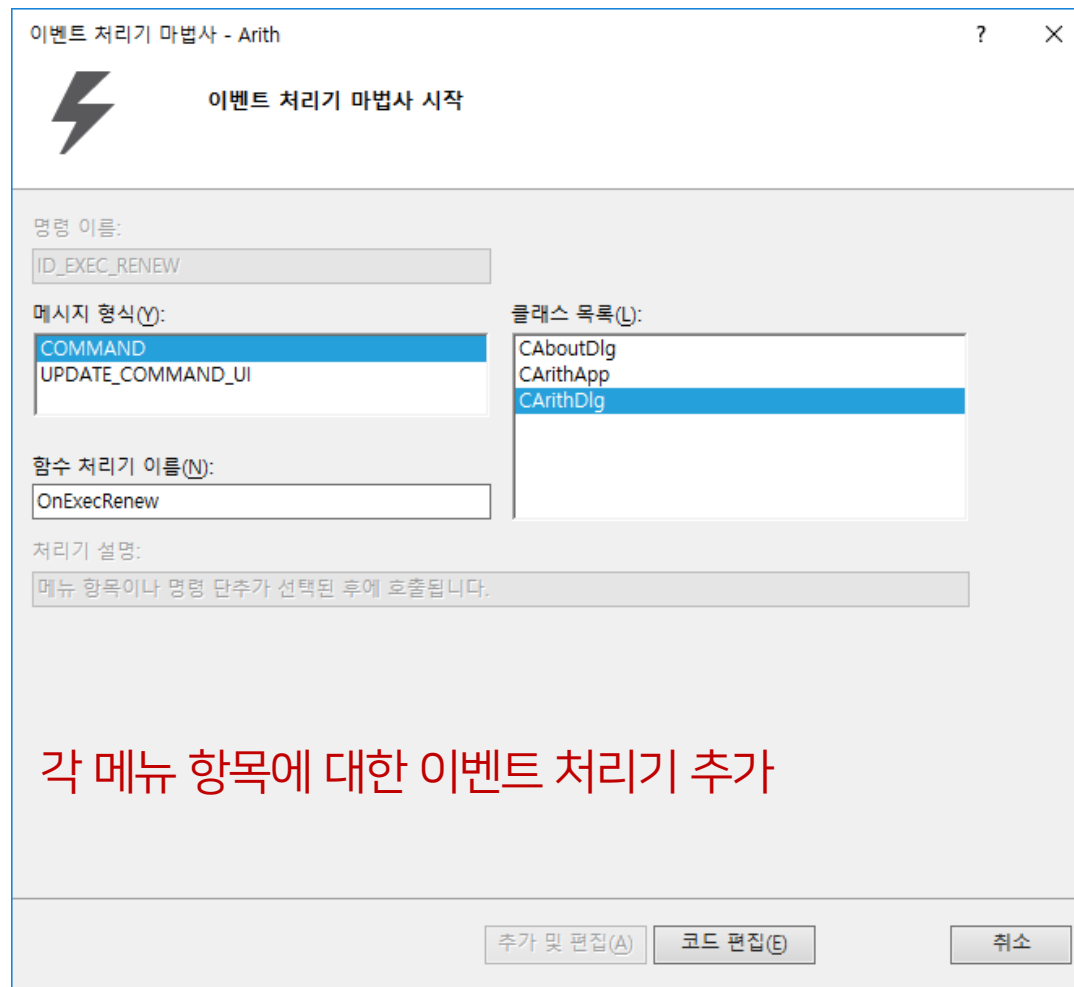
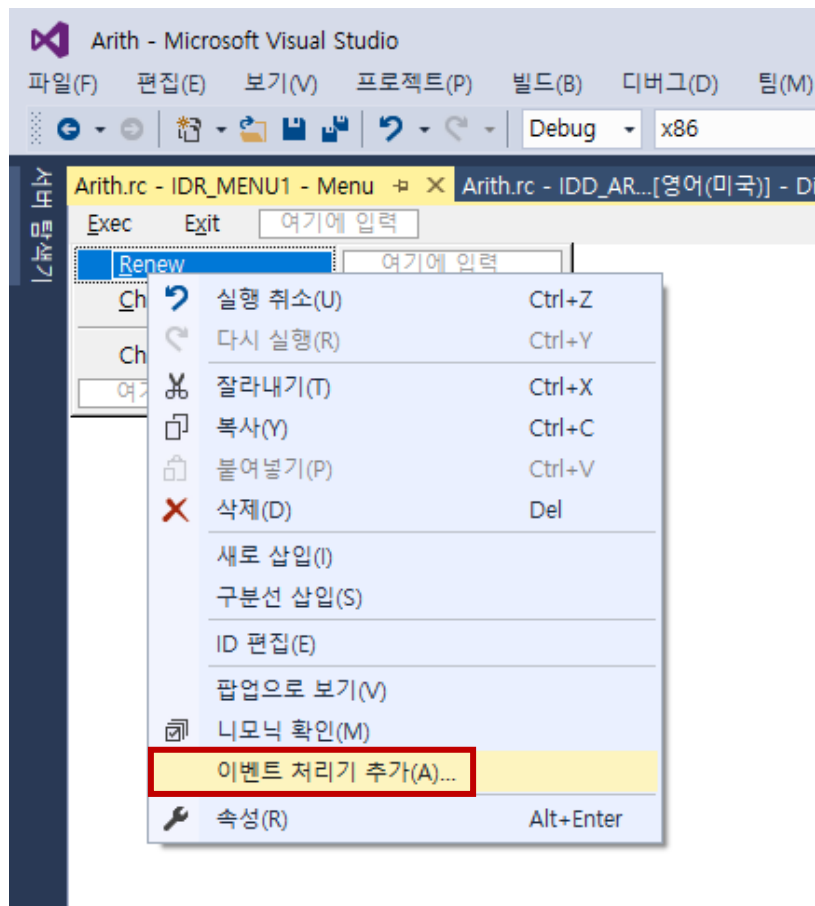


클래스 마법사를 사용한 코딩 - 변수 추가



- 사용자 지정 추가
m_nLimit (int 타입)

클래스 마법사를 사용한 코딩 - 이벤트 처리기 추가



각 메뉴 항목에 대한 이벤트 처리기 추가

클래스 마법사를 사용한 코딩 - 메시지 처리기 함수 정의

- 교재 124 ~ 129 쪽 코드를 보며 함수를 정의함
 - ArithDlg.h / ArithDlg.cpp
- 특히, 음영으로 표시된 부분에 유의하며 코드를 작성함

클래스 마법사를 사용한 코딩 - 메시지 처리기 함수 정의

// ArithDlg.h : header file

```
//
#pragma once
// CArithDlg dialog
class CArithDlg : public CDialogEx
{
// Construction
public:
    CArithDlg(CWnd* pParent = NULL);    // standard
    constructor
// Dialog Data
    enum { IDD = IDD_ARITH_DIALOG };
    protected:
        virtual void DoDataExchange(CDataExchange* pDX); //
// DDX/DDV support
// Implementation
    protected:
        HICON m_hIcon;
        // Generated message map functions
        virtual BOOL OnInitDialog();
        afx_msg void OnSysCommand(UINT nID, LPARAM
IParam);
        afx_msg void OnPaint();
        afx_msg HCURSOR OnQueryDragIcon();
        DECLARE_MESSAGE_MAP()
```

public:

```
    int m_nRadio;
    int m_nAnswer;
    CString m_strName;
    int m_nNum1;
    int m_nNum2;
    int m_nLimit;
    afx_msg void OnExecRenew();
    afx_msg void OnExecChangelevel();
    afx_msg void OnExecCheck();
    afx_msg void OnExitQuit();
    afx_msg void OnExitSaveandquit();
```

};

클래스 마법사를 사용한 코딩 - 메시지 처리기 함수 정의

```
// ArithDlg.cpp : implementation file
//
// CArithDlg dialog
CArithDlg::CArithDlg(CWnd* pParent /*=NULL*/)
    : CDialogEx(IDD_ARITH_DIALOG, pParent)
    , m_nRadio(0)
    , m_nAnswer(0)
    , m_strName(_T(""))
    , m_nNum1(0)
    , m_nNum2(0)
    , m_nLimit(0)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}
void CArithDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
    DDX_Radio(pDX, IDC_ADD_RADIO, m_nRadio);
    DDX_Text(pDX, IDC_ANSWER_EDIT, m_nAnswer);
    DDV_MinMaxInt(pDX, m_nAnswer, 1, 9999);
    DDX_Text(pDX, IDC_NAME_EDIT, m_strName);
    DDV_MaxChars(pDX, m_strName, 10);
    DDX_Text(pDX, IDC_NUM1_EDIT, m_nNum1);
    DDX_Text(pDX, IDC_NUM2_EDIT, m_nNum2);
}
```

```
void CArithDlg::OnExecRenew()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    m_nNum1 = 1 + rand()%m_nLimit;
    // 편집 - 추가
    m_nNum2 = 1 + rand()%m_nLimit;
    // 편집 - 추가
    UpdateData (FALSE);
    // 편집 - 추가
}
void CArithDlg::OnExecChangelevel()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    m_nLimit = (9 + 99) - m_nLimit;
    // 편집 - 추가
}
void CArithDlg::OnExecCheck()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.

    int        correct_answer;
    // 편집 -
    추가
}
```

클래스 마법사를 사용한 코딩 - 메시지 처리기 함수 정의

```
// 편집 - 추가
UpdateData();

// 편집 - 추가
if (m_nRadio == 0)

// 편집 - 추가
correct_answer = m_nNum1 + m_nNum2;
// 편집 - 추가
else

//
편집 - 추가
correct_answer = m_nNum1 * m_nNum2;
// 편집 - 추가

//
편집 - 추가
if (m_nAnswer == correct_answer)
// 편집 - 추가
    MessageBox (_T("Correct. Congratulations, ") +
m_strName);
// 편집 - 추가
else

//
편집 - 추가
    MessageBox (_T("Wrong. Try again, ") + m_strName);
// 편집 - 추가
}
```

```
void CArithDlg::OnExitQuit()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    OnOK();

    // 편집 -
    추가
}
void CArithDlg::OnExitSaveandquit()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    OnOK();

    // 편집 -
    추가
}
```

클래스 마법사를 사용한 코딩 - 메시지 처리기 함수 정의

```
BOOL CArithDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();
    // Add "About..." menu item to system menu.
    // IDM_ABOUTBOX must be in the system command
range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) ==
IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);
    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        BOOL bNameValid;
        CString strAboutMenu;
        bNameValid =
strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }
}
```

```
// Set the icon for this dialog. The framework does this automatically
// when the application's main window is not a dialog
SetIcon(m_hIcon, TRUE);           // Set big icon
SetIcon(m_hIcon, FALSE);          // Set small icon
// TODO: Add extra initialization here
m_nLimit = 9;                                     // 편집 -
추가
m_nRadio = 0;                                     // 편집 -
추가
srand ((int) time (NULL));                         // 편집 - 추가
m_nNum1 = 1 + rand()%m_nLimit;                     // 편집 - 추가
m_nNum2 = 1 + rand()%m_nLimit;                     // 편집 - 추가
UpdateData (FALSE);                                // 편집 - 추가
return TRUE; // return TRUE unless you set the focus to a
control
}
```


클래스 마법사를 사용한 코딩 - 메시지 처리기 함수 정의

```
BEGIN_MESSAGE_MAP(CArithDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_COMMAND(ID_EXEC_RENEW,
&CArithDlg::OnExecRenew)
    ON_COMMAND(ID_EXEC_CHANGELEVEL,
&CArithDlg::OnExecChangelevel)
    ON_COMMAND(ID_EXEC_CHECK,
&CArithDlg::OnExecCheck)
    ON_COMMAND(ID_EXIT_QUIT,
&CArithDlg::OnExitQuit)
    ON_COMMAND(ID_EXIT_SAVEANDQUIT,
&CArithDlg::OnExitSaveandquit)
END_MESSAGE_MAP()
```

과제 제출: 보고서 + 솔루션 폴더

- 내용
 - 각 과제의 프로그램 소스 및 설명
 - 각 과제의 실행 화면 캡처
- 제출 형식
 - 스노우보드에서 Lab08_이름_학번.doc 다운로드
 - 스노우보드에 제출: Lab08_이름_학번.zip (아래 두 파일 반드시 포함하도록)
 1. 소스 코드 : Lab08_이름_학번 (압축하지 않은 솔루션 폴더)
 2. 보고서: Lab08_이름_학번.doc (설명 없을 경우 감점)
- 제출 기한
 - 5월 18일(화) PM 11:59 까지