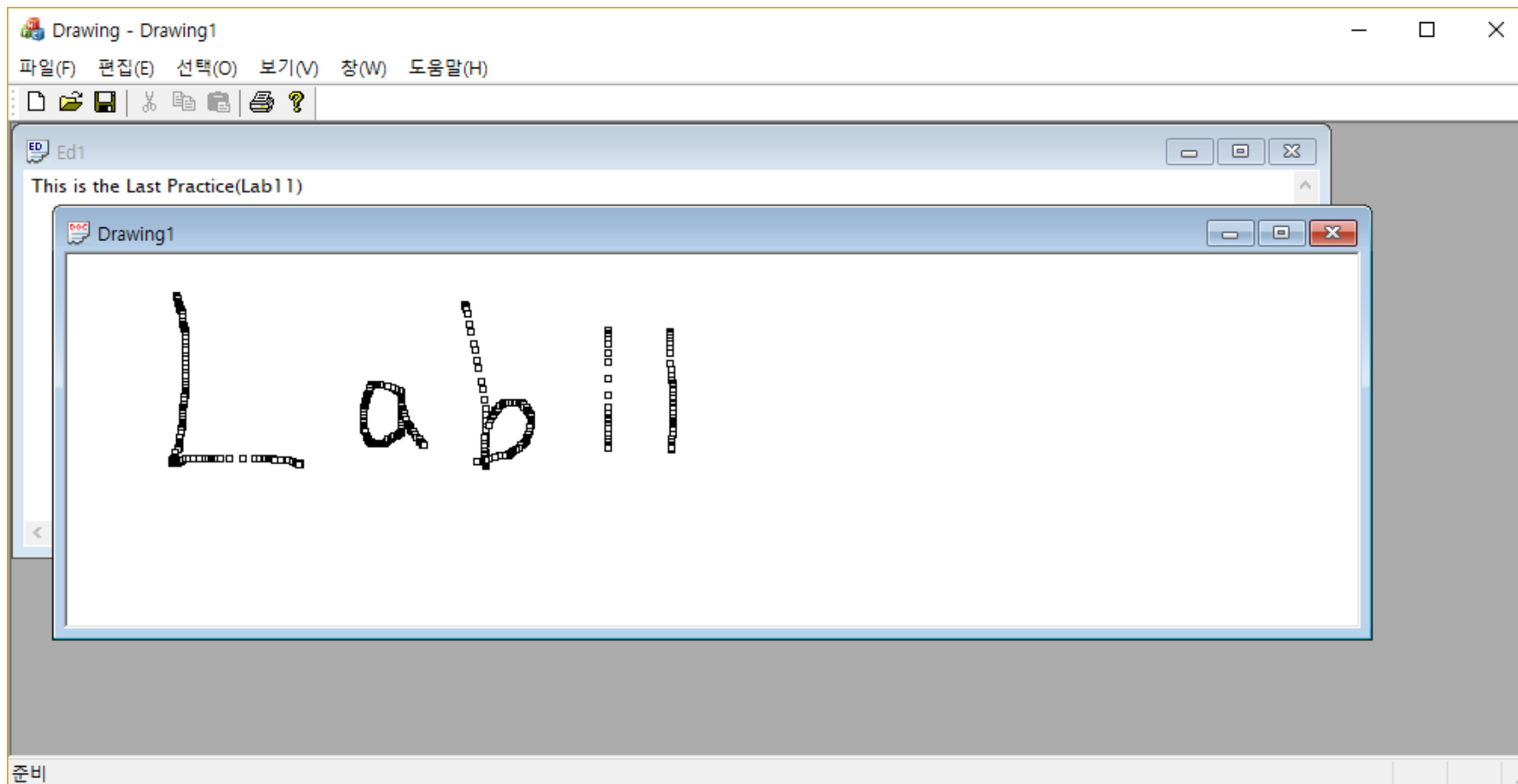


윈도우프로그래밍 Lab 10

- 6장 MDI 응용 프로그램 예제 -

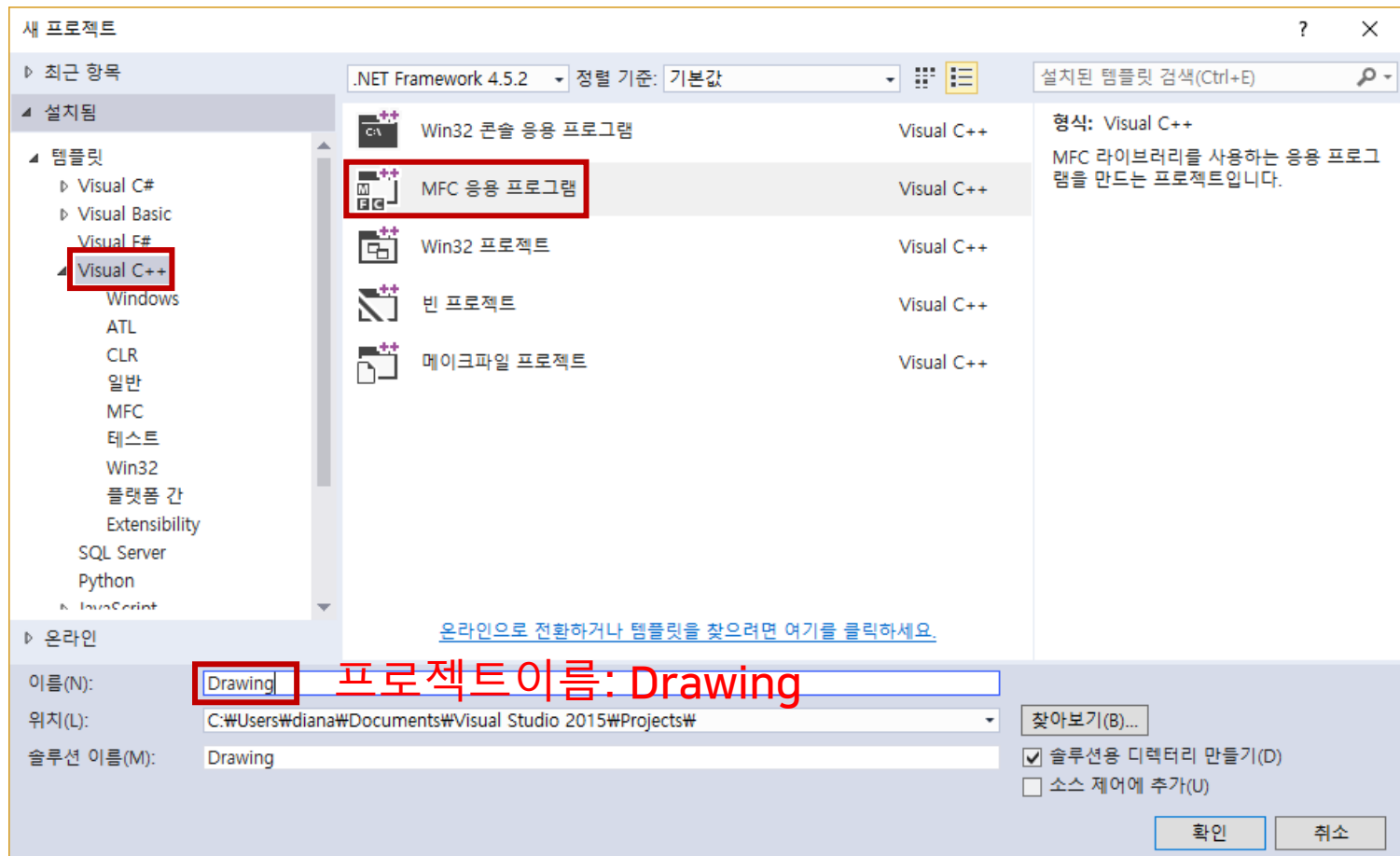
김한슬
uo3359@sookmyung.ac.kr

6장 MDI 응용 프로그램 예제



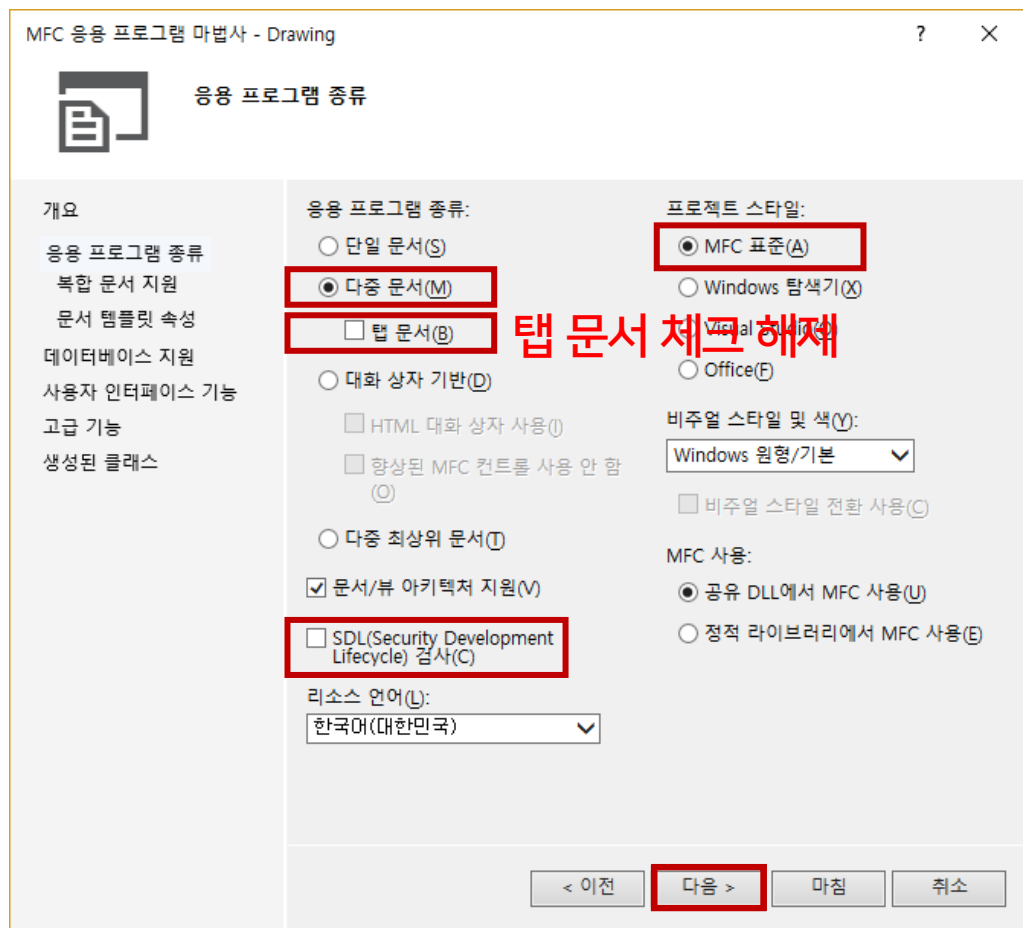
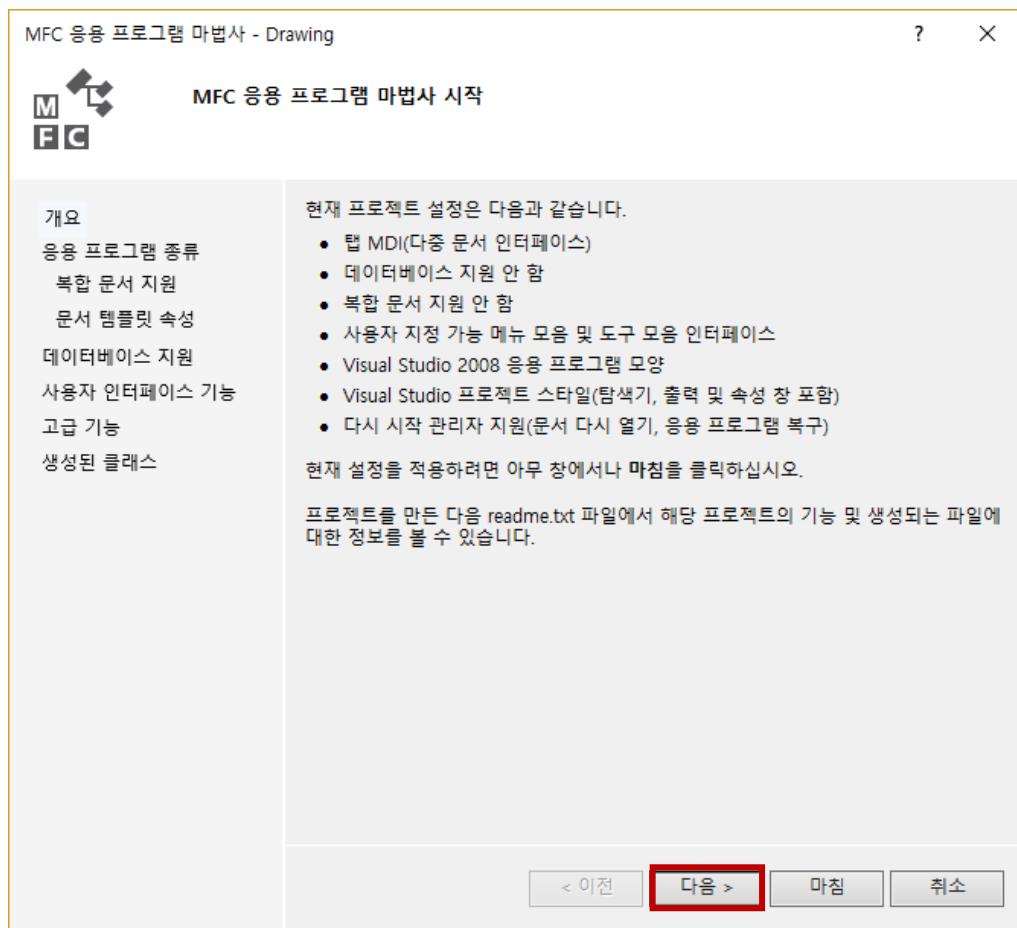
프로젝트 생성

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



프로젝트 생성

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



프로젝트 생성

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

MFC 응용 프로그램 마법사 - Drawing

복합 문서 지원

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

복합 문서 지원:

☒ 없음(N)
☐ 컨테이너(O)
☐ 미니 서버(M)
☐ 풀 서버(V)
☐ 컨테이너/풀 서버(S)

추가 옵션:

☐ 액티브 문서 서버(A)
☐ 액티브 문서 컨테이너(D)
☒ 복합 파일 지원(U)

< 이전 다음 > 마침 취소

MFC 응용 프로그램 마법사 - Drawing

문서 템플릿 속성

파일 확장명: mdi

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

파일 확장명(X):
mdi

필터 이름(N):
Drawing Files (*.mdi)

☐ 파일 형식에 대한 미리 보기 처리기 지원(R)
☐ 파일 형식에 대한 축소판 그림 처리기 지원(U)
☐ 파일 형식에 대한 검색 처리기 지원(E)

주 프레임 캡션(P):
Drawing

파일 형식 ID(I):
Drawing.Document

문서 형식 이름(T):
Drawing

파일 형식의 긴 이름(L):
Drawing.Document

파일의 새 약식 이름(S):
Drawing

< 이전 다음 > 마침 취소

프로젝트 생성

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

MFC 응용 프로그램 마법사 - Drawing

데이터베이스 지원

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

데이터베이스 지원:

☒ 없음(N)
☐ 헤더 파일만(O)
☐ 파일을 지원하지 않는 데이터베이스 뷰(V)
☐ 파일을 지원하는 데이터베이스 뷰(T)

특성을 사용하는 데이터베이스 클래스 생성(A)
☒ 모든 열 바인딩(B)

형식:
☒ 다이너셋(V)
☐ 스냅샷(P)

클라이언트 형식:
☒ OLE DB(L)
☐ ODBC(D)

데이터 소스:
데이터 소스(S)...

< 이전 다음 > 마침 취소

MFC 응용 프로그램 마법사 - Drawing

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)

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):
Drawing

자식 프레임 스타일:
☒ 자식 최소화 상자(M)
☒ 자식 최대화 상자(Z)
☐ 자식 창 최대화(D)

< 이전 다음 > 마침 취소

프로젝트 생성

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

MFC 응용 프로그램 마법사 - Drawing

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(V)
- ☒ 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

MFC 응용 프로그램 마법사 - Drawing

생성된 클래스

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

생성된 클래스(G):

- CDrawingView
- CDrawingApp
- CDrawingDoc
- CMainFrame
- CChildFrame

클래스 이름(L): CDrawingView

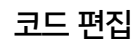
.h 파일(E): DrawingView.h

기본 클래스(A): CView

.cpp 파일(P): DrawingView.cpp

< 이전 다음 > 마침 취소

- [프로젝트] 메뉴 - [클래스 마법사]



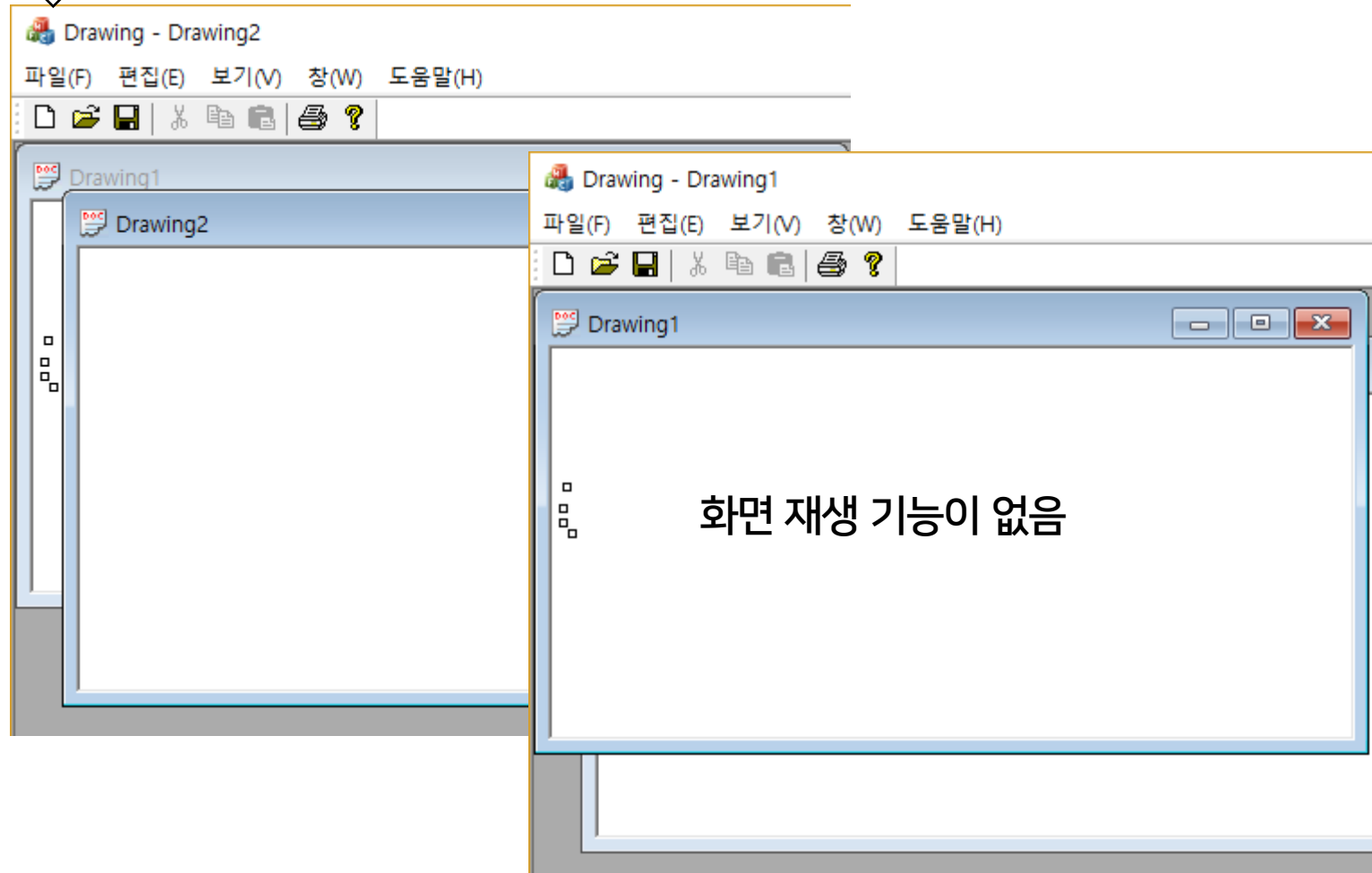
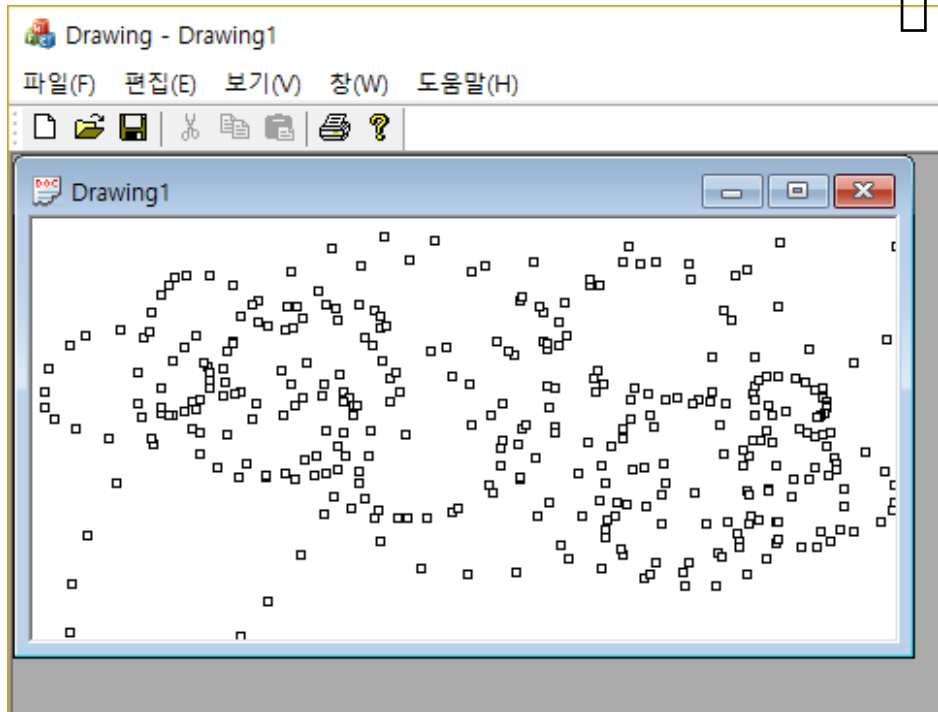
: nFlags는 마우스버튼, Ctrl, Shift 키 상태를 나타냄

- MK_CONTROL, MK_SHIFT, MK_LBUTTON,
- MK_MBUTTON, MK_RBUTTON

마우스 드래그 이벤트(메시지)처리기 함수 추가

- 실행 결과

[파일] - [새로 만들기]



도큐먼트/뷰 구조 사용

- 앞의 프로그램

- 문서 파일을 저장하거나 읽어 들이는 기능 / 사라졌다 나타나는 화면에 대한 재생 기능이 없음

- 도큐먼트/뷰 구조를 사용하여 해결하기

- ① 사용자가 그리는 점들에 관한 정보를 저장하기 위한 자료구조 추가

(도큐먼트 클래스)

- ② 자료구조를 디스크에 저장하고 읽어 들이기 위한 코드 추가

(도큐먼트 클래스)

- ③ 점 하나가 추가될 때마다 이 점을 자료구조에 추가하기 위한 코드 추가

(뷰 클래스)

- ④ 뷰의 노출 시 내용을 적절히 재생하기 위한 코드 추가 (뷰 클래스)

도큐먼트 클래스에 자료구조 추가

- 두 개의 CDWordArray 사용
 - 각기 x 좌표 열과 y 좌표 열을 나타냄
 - 32bit 정수인 DWORD 타입을 원소로 가짐
 - CDrawingDoc 클래스의 멤버 변수로 선언함
 - DrawingDoc.h 를 편집 or 클래스 뷰에서 도큐먼트 클래스의 팝업 메뉴
[추가 - 변수 추가] 사용
 - private 가시성으로 선언함

```
DrawingDoc.h 편집(변수 추가)

class CDrawingDoc : public CDocument
{
    ..... (중간생략)

    private:
        CDWordArray x, y;
};
```

도큐먼트 자료구조에 대한 저장 및 열기

- 도큐먼트 클래스의 Serialize() 함수 작성

- CDWordArray 클래스는 자체적으로 Serialize() 함수가 구현되어 있음 따라서 도큐먼트 클래스의 Serialize() 함수에서는 CDWordArray의 Serialize() 함수를 호출하여 해당 작업을 수행

DrawingDoc.cpp 편집

```
void CDrawingDoc::Serialize(CArchive& ar)
{
    x.Serialize(ar);
    y.Serialize(ar);
}
```

추가되는 점들을 도큐먼트 자료구조에 반영

- 뷰에 그릴 사각형의 크기 변경을 위한 멤버변수 w(private) 추가(뷰 클래스)

DrawingView.h 편집(변수 추가)

```
class CDrawingView : public CView
{
    ..... (중간생략)

    private:
        int w;
};
```

DrawingView.cpp 편집(변수 초기화)

```
CDrawingView::CDrawingView()
{
    // TODO: 여기에 생성 코드를 추가합니다.
    w = 5;
}
```

추가되는 점들을 도큐먼트 자료구조에 반영

DrawingView.cpp (OnMouseMove 함수 수정) 편집

```
void CDrawingView::OnMouseMove(UINT nFlags, CPoint
point)
{
    CDrawingDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    if (nFlags == MK_LBUTTON) { // 마우스 왼쪽 버튼이 눌러져
있으면
        CClientDC dc(this);
        dc.Rectangle(point.x, point.y, point.x + w,
point.y + w);
        pDoc->Add(point);
        pDoc->SetModifiedFlag();
    }
    CView::OnMouseMove(nFlags, point);
}
```

DrawingDoc.h 편집 (함수 추가)

```
class CDrawingDoc : public CDocument
{
    ..... (중간생략)

public:
    void Add(CPoint point);
};
```

DrawingDoc.cpp 편집 (함수 추가)

```
void CDrawingDoc::Add(CPoint point)
{
    x.Add(point.x);
    y.Add(point.y);
}
```

뷰 클래스에서 노출 이벤트의 처리

DrawingView.cpp (OnDraw 함수 수정)편집

```
void CDrawingView::OnDraw(CDC* pDC)
{
    CDrawingDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    for (int i = 0; i < pDoc->Size(); i++) {
        CPoint p = pDoc->GetPoint(i);
        pDC->Rectangle(p.x, p.y, p.x + w, p.y + w);
    }
}
```

DrawingDoc.h 편집 (함수 추가)

```
class CDrawingDoc : public CDocument
{
    ..... (중간생략)

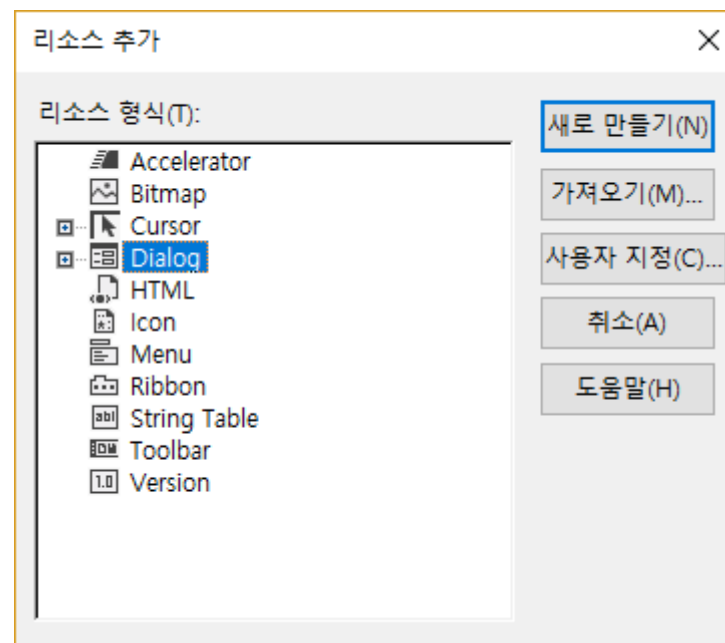
public:
    void Add(CPoint point);
    int Size();
    CPoint GetPoint(int i);
};
```

DrawingDoc.cpp 편집 (함수 추가)

```
int CDrawingDoc::Size()
{
    return x.GetSize();
}

CPoint CDrawingDoc::GetPoint(int i)
{
    return CPoint(x[i], y[i]);
}
```

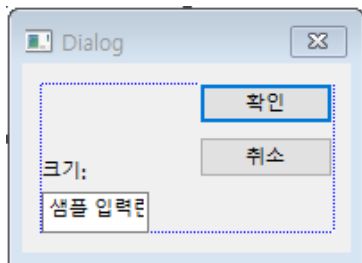
추가 기능 - 다이얼로그 리소스의 추가



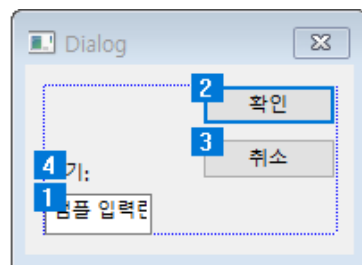
행 추가 메뉴 항목

- 다이얼로그 설계

- 사각형 크기를 입력 받을 다이얼로그 시각적 설계



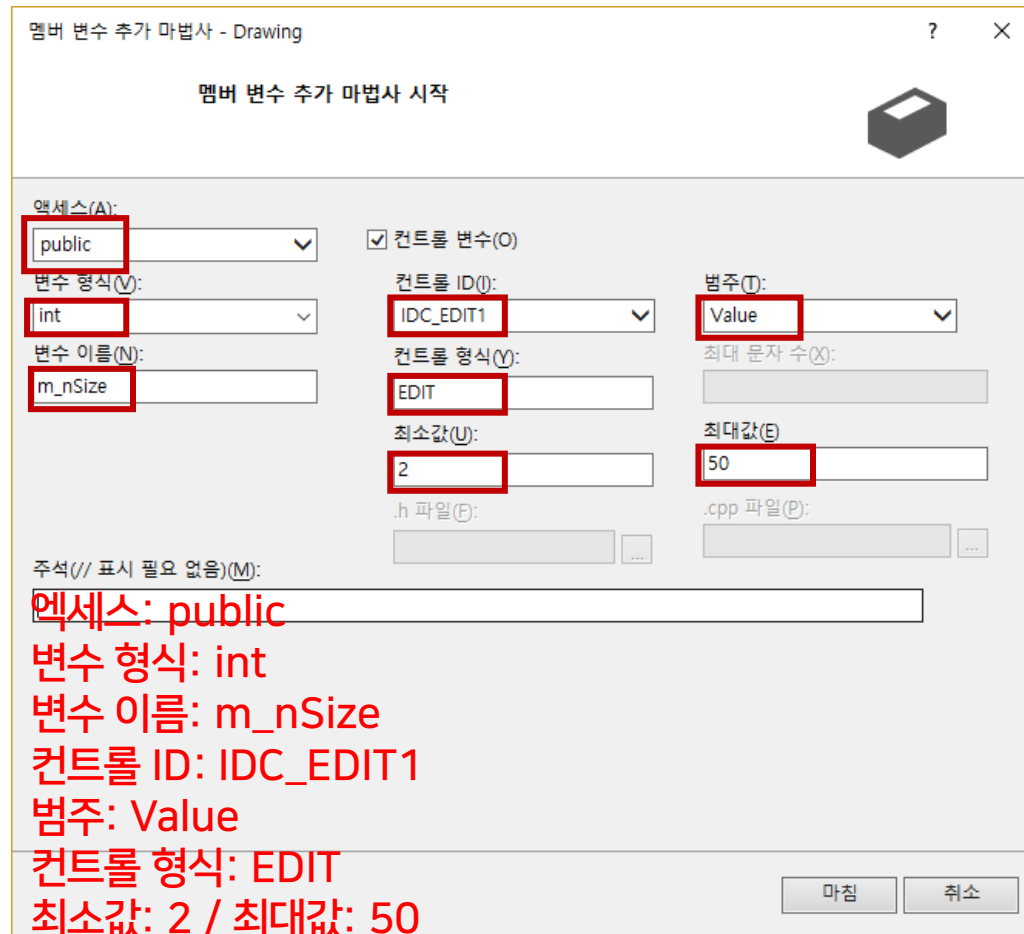
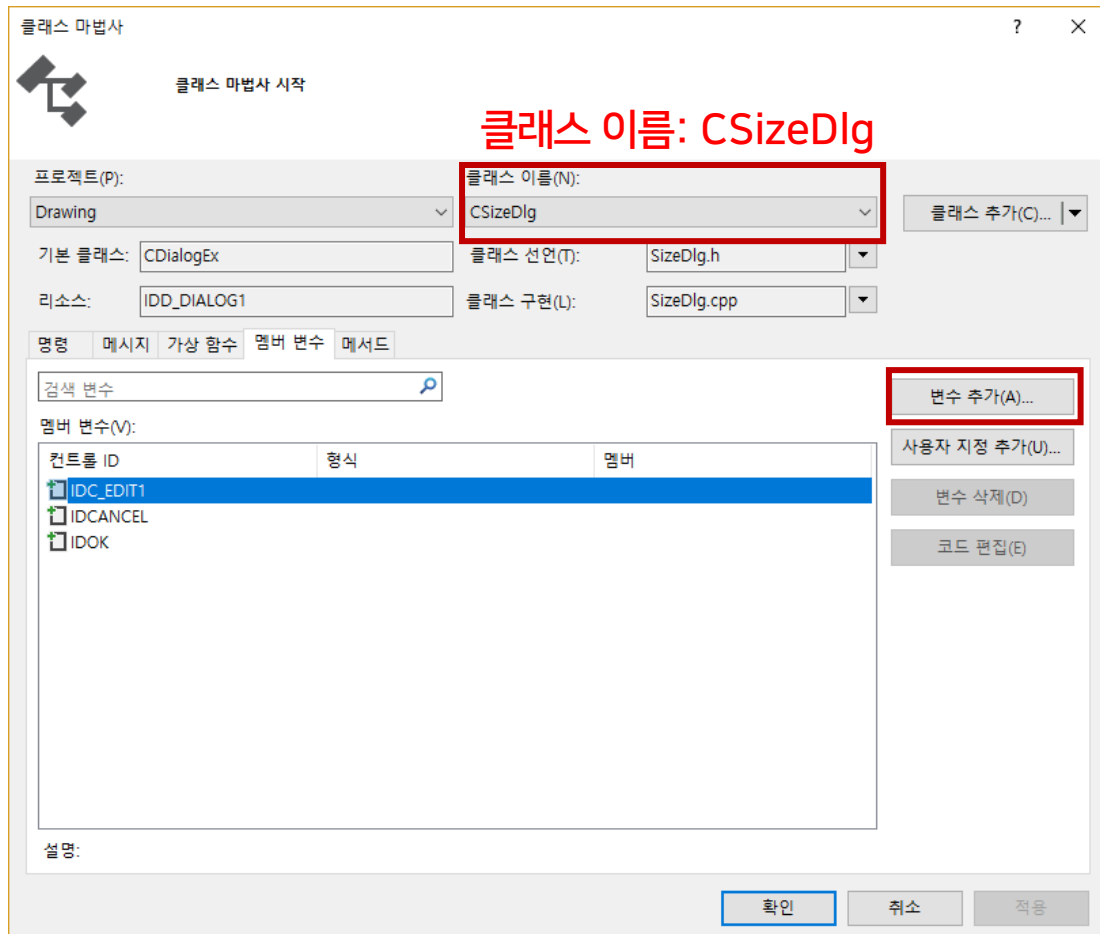
- [서식 - 탭 순서] 기능을 사용 / Edit 컨트롤이 1번이 되도록 변경



- 다이얼로그를 위한 클래스 생성
 - 다이얼로그 우클릭 - 클래스 추가
 - 클래스 이름: CSizeDlg

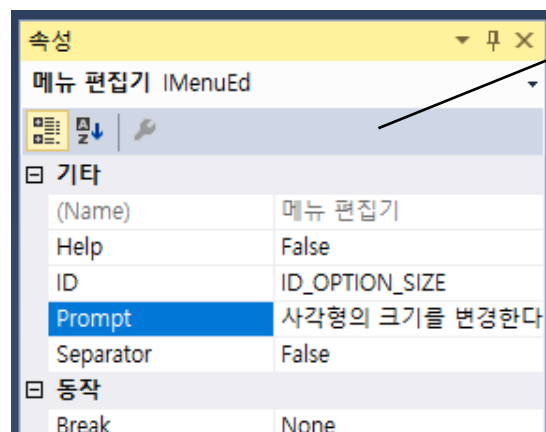
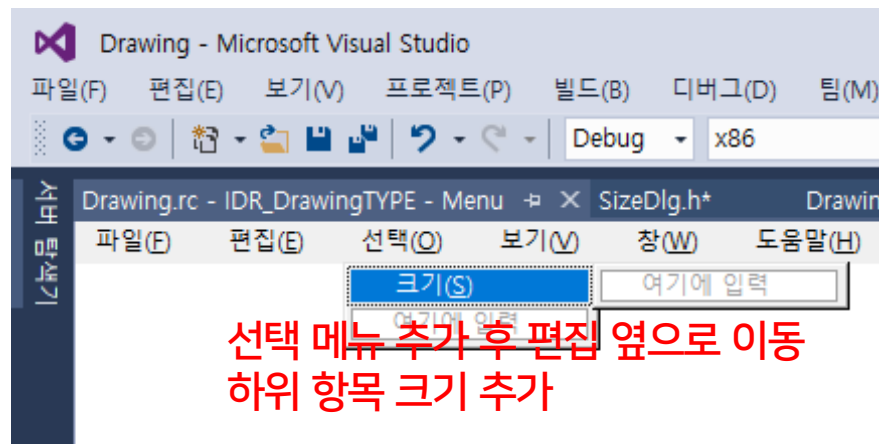
추가 기능 - 다이얼로그 리소스의 추가

- Edit 컨트롤을 위한 변수 추가
 - 다이얼로그 우클릭 - 클래스 마법사 - 멤버변수



추가 기능 - 메뉴항목 및 처리기 함수 추가

• 리소스 뷰 - Menu - IDR_DrawingTYPE



ID: ON_OPTION_SIZE
Prompt: 사각형의 크기를 변경한다

코드 편집

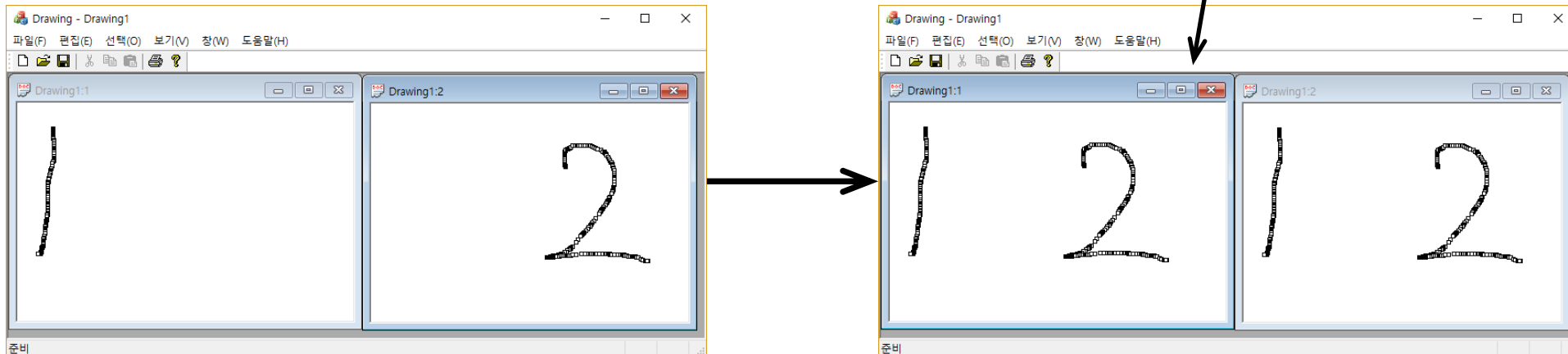
```
#include "SizeDlg.h"

void
CDrawingView::OnOptionSize()
{
    CSizeDlg dlg;
    dlg.m_nSize = w;
    If (dlg.DoModal() == IDOK)
        w = dlg.m_nSize;
}
```

- 크기(S) 우클릭 - 이벤트 처리기 추가
 - 메시지 형식: COMMAND
 - 함수 처리기 이름: OnOptionSize
 - 클래스 목록: CDrawingView

다중 뷰의 동기화

- 프로그램을 실행하여 두 개의 창에 각각 그림을 그려 결과를 확인
 - 두 창 : 하나의 문서에 대한 뷰(창->새창)
 - 두 창 모두에 각각 그림 그리기(OnMouseMove 호출)
 - 주 실행 창을 최소화했다가 다시 불러오기(OnDraw 호출): 두 창의 내용이 같은 것을 볼 수 있음
- 각 뷰에서 추가되는 점들은 모두 자료구조에 추가됨
- OnMouseMove : 해당 뷰 안에서만 이루어짐
- OnDraw : 자료구조 안에 저장되어 있는 좌표들을 사용하여 새로 그려짐. 따라서, 두 뷰가 같은 그림을 보여줌
- View Class의 OnUpdate 함수 통하여 실시간 동기화 가능



다중 뷰의 동기화

- CDocument 클래스 객체는 그 문서에 연계되어 있는 모든 뷰들의 목록을 유지
- UpdateAllViews() : 특정 문서에 대해 그 문서에 연계되어 있는 각 뷰에 대해 OnUpdate() 함수를 호출
- OnUpdate() 함수를 수정하여 뷰 동기화 구현
 - OnMouseMove()에서 동기화가 필요한 이벤트가 발생했음을 알림

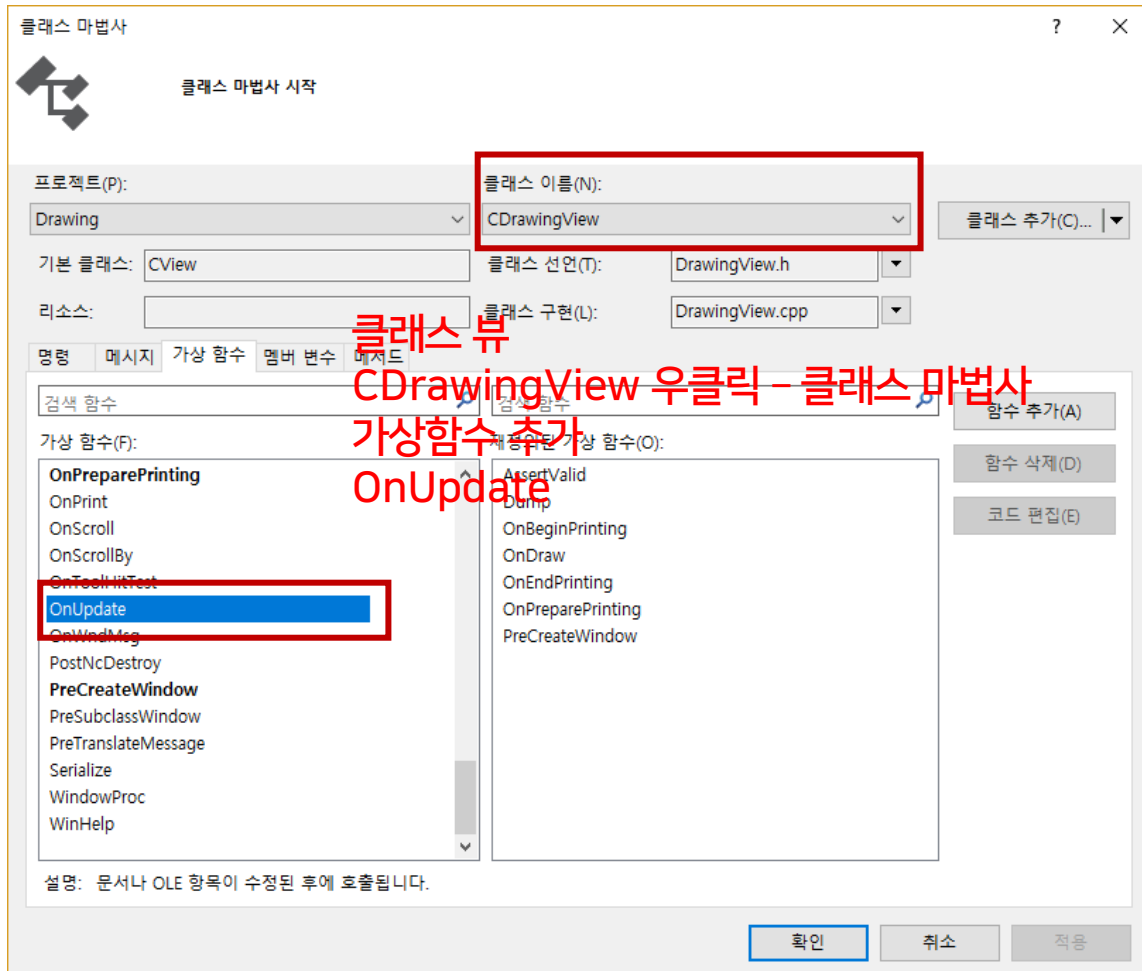
DrawingView.cpp (OnMouseMove 함수 수정)편집

```
void CDrawingView::OnMouseMove(UINT nFlags, CPoint point)
{
    CDrawingDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    if (nFlags == MK_LBUTTON) { // 마우스 왼쪽 버튼이 눌러져 있으면
        CClientDC dc(this);
        dc.Rectangle(point.x, point.y, point.x + w, point.y + w);
        pDoc->Add(point);
        pDoc->SetModifiedFlag();
        pDoc->UpdateAllViews(this, 0, 0);
    }
    CView::OnMouseMove(nFlags, point);
}
```

다중 뷰의 동기화

- OnUpdate() 함수를 수정하여 뷰 동기화 구현
 - 가상함수 OnUpdate() 추가 : 자료구조의 마지막 좌표를 받아와서 사각형을 그림



DrawingView.cpp 편집

```
void CDrawingView::OnUpdate(CView* pSender, LPARAM lHint, CObject* pHint)
{
    CDrawingDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

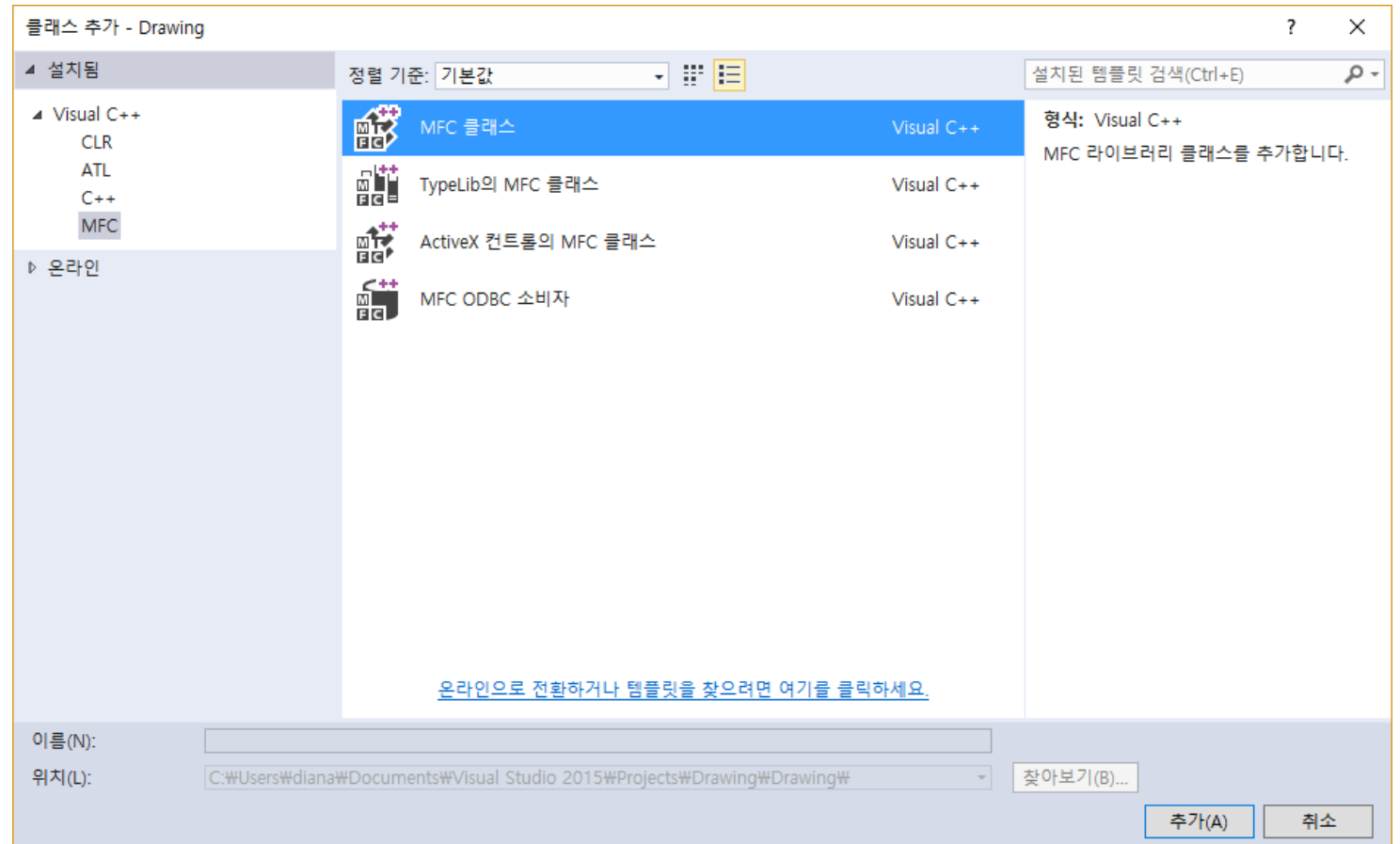
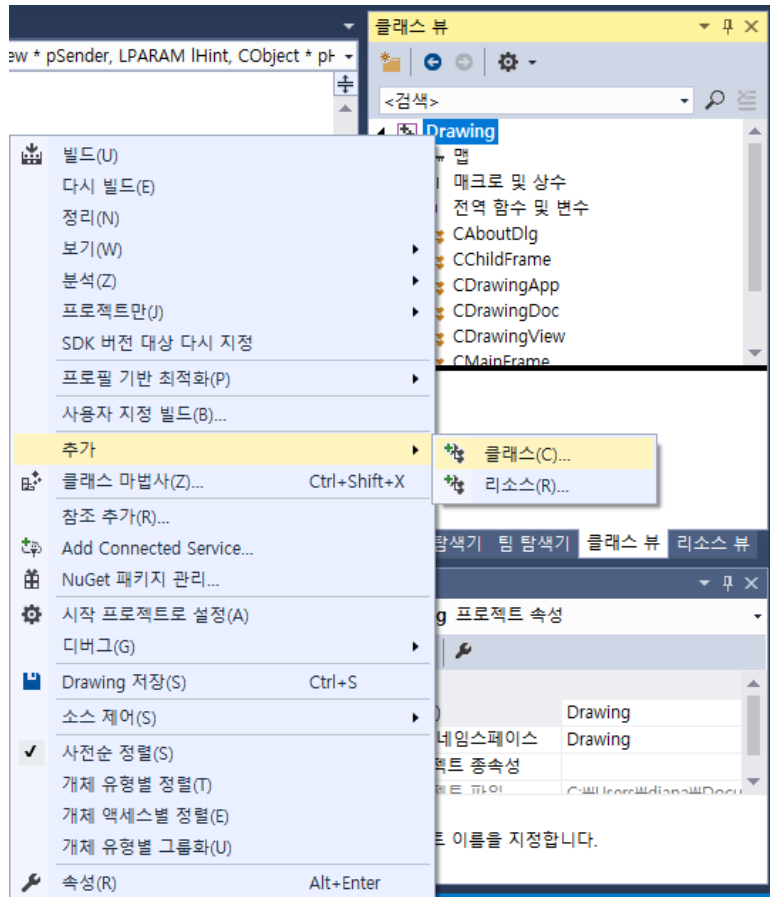
    int n = pDoc->Size();
    if (n > 0) {
        CPoint p = pDoc->GetPoint(n - 1);
        CClientDC dc(this);
        dc.Rectangle(p.x, p.y, p.x + w, p.y + w);
    }
}
```

다중 문서 처리와 도큐먼트 템플릿

- 그림 문서와 텍스트 문서 등, 두 가지의 서로 다른 유형의 문서를 취급할 수 있는 MDI 애플리케이션 작성
 - 각 문서 유형 마다 적절한 **도큐먼트 템플릿**을 만들어 애플리케이션 안에 등록해야 함
 1. 그림 편집기 완성
 2. 텍스트 편집 기능 추가
 - 1) 그림 편집기에 텍스트 편집기를 위한 도큐먼트 & 뷰 클래스 추가
 - 2) 텍스트 편집기를 위한 도큐먼트 템플릿을 만들어 등록
 - 3) 텍스트 편집기를 위한 3가지 리소스 추가

다중 문서 처리와 문서 템플릿

• 새 문서 클래스 및 뷰 클래스 추가



다중 문서 처리와 문서 템플릿

- 새 문서 클래스 및 뷰 클래스 추가

MFC 클래스 추가 마법사 - Drawing

MFC 클래스 추가 마법사 시작

이름
문서 템플릿 속성

클래스 이름(U):
CEdDoc

기본 클래스(B):
CDocument

대화 상자 ID(D):
IDD_EDDOC

.h 파일(I):
EdDoc.h

.cpp 파일(P):
EdDoc.cpp

☐ Active accessibility(Y)

클래스 이름
CEdDoc

기본 클래스
CDocument

자동화:
☒ 없음(N)
☐ 자동화(A)
☐ 형식 ID로 생성 가능(E)

형식 ID(I):
Drawing,EdDoc

☐ DocTemplate 리소스 생성(G)

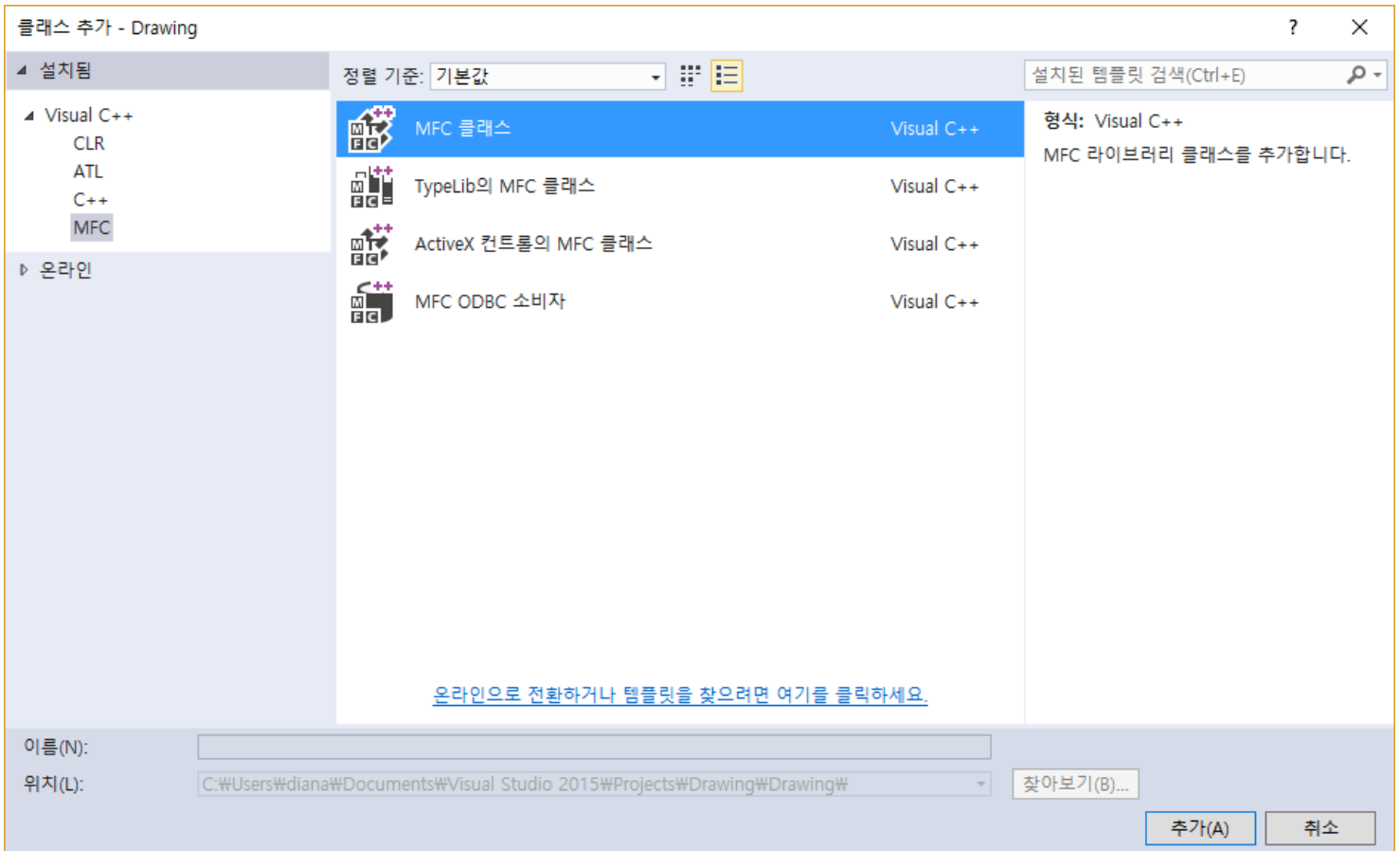
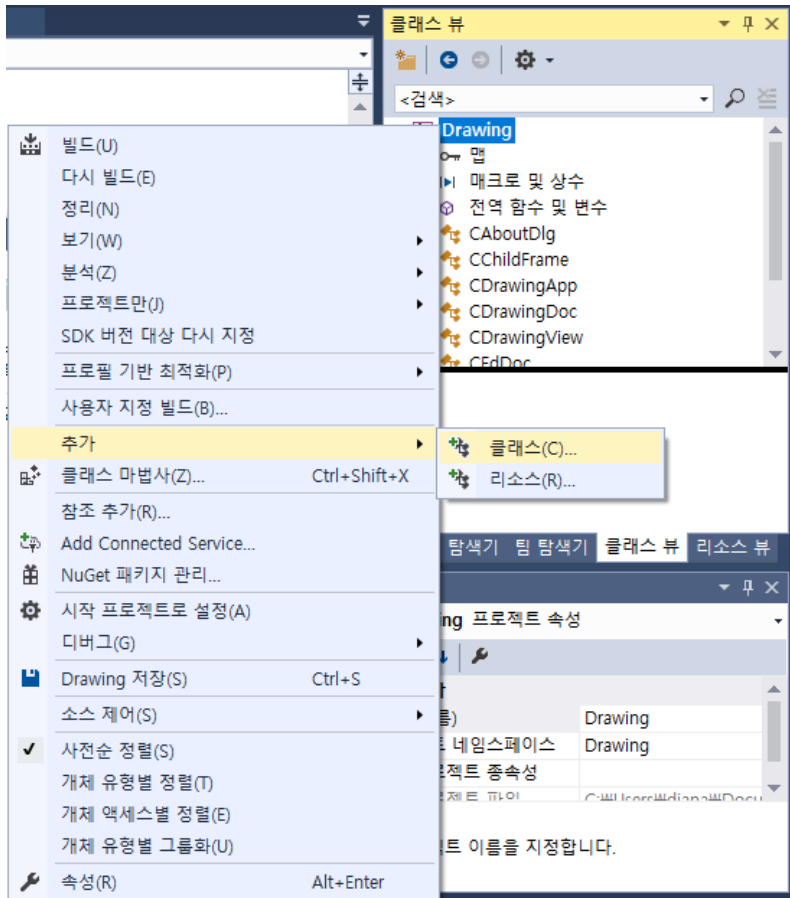
< 이전 다음 > 마침 취소

EdDoc.cpp 편집

```
void CEdDoc::Serialize(CArchive& ar)
{
    ((CEditView*)m_viewList.GetHead())->SerializeRaw(ar);
}
```

다중 문서 처리와 문서 템플릿

• 새 문서 클래스 및 뷰 클래스 추가



다중 문서 처리와 문서 템플릿

- 새 문서 클래스 및 뷰 클래스 추가

MFC 클래스 추가 마법사 - Drawing

MFC 클래스 추가 마법사 시작

이름
문서 템플릿 속성

클래스 이름(U):
CEdView

기본 클래스(B):
CEditView

대화 상자 ID(D):
IDD_EDVIEW

.h 파일(I):
EdView.h

.cpp 파일(P):
EdView.cpp

☐ Active accessibility(Y)

클래스 이름
CEdView
기본 클래스
CEditView

DRMML 리소스 ID(S):
IDR_HTML_EDVIEW

리소스 파일(N):
EdView.htm

기록
☒ 없음(N)
☐ 자동화(A)

형식 ID(I):
Drawing,EdView

☐ DocTemplate 리소스 생성(G)

< 이전 다음 > 마침 취소

다중 문서 처리와 도큐먼트 템플릿

- 새 도큐먼트 클래스 및 뷰 클래스 추가

Drawing.cpp 편집

```
// CDrawingApp 초기화

#include "EdDoc.h"
#include "EdView.h"

BOOL CDrawingApp::InitInstance()
{
    ..... (중간생략)

    CMultiDocTemplate* pDocTemplate;

    pDocTemplate = new CMultiDocTemplate(IDR_EdTYPE,
        RUNTIME_CLASS(CEdDoc),
        RUNTIME_CLASS(CChildFrame),
        RUNTIME_CLASS(CEdView));
    if (!pDocTemplate)
        return FALSE;
    AddDocTemplate(pDocTemplate);

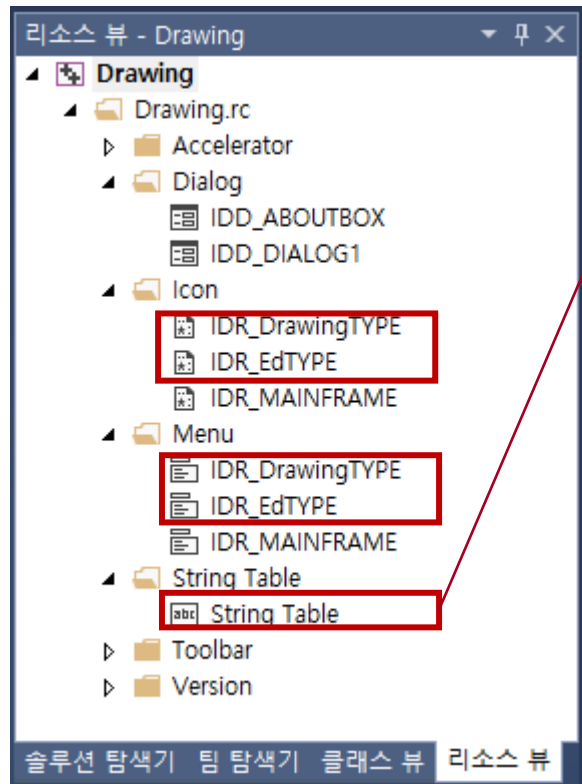
    pDocTemplate = new CMultiDocTemplate(IDR_DrawingTYPE,
        RUNTIME_CLASS(CDrawingDoc),
        RUNTIME_CLASS(CChildFrame), // 사용자 지정 MDI 자식 프레임입니다.

    ..... (중간생략)
}
```

다중 문서 처리와 도큐먼트 템플릿

• 리소스 만들기

- 리소스 추가
- 리소스(Icon, Menu, String Table)에 리소스 추가(IDR_DrawingTYPE을 복사 붙여넣기)
- 추가한 각 리소스 ID를 IDR_EdTYPE으로 변경



Drawing.rc - String Table		
ID	값	캡션
IDP_OLE_INIT_FAILED	100	OLE를 초기화할 수 없습니다. OLE 라이브러리 버전이 올바른지 확인하십시오.
IDR_MAINFRAME	128	Drawing
IDR_DrawingTYPE	130	\\nDrawing\\nDrawing\\nDrawing Files (*.drw)\\n.DRW\\nDrawing.Document\\n...
IDR_EdTYPE	131	\\nEd\\nEd\\nEd Files (*.tex)\\n.TEXT\\nEd.Document\\nEd Document
ID_OPTION_SIZE	32772	사각형의 크기를 변경한다

IDR_EdTYPE 캡션 변경

\\nEd\\nEd\\nEd Files (*.tex)\\n.TEXT\\nEd.Document\\nEd Document

IDR_DrawingTYPE 캡션 변경

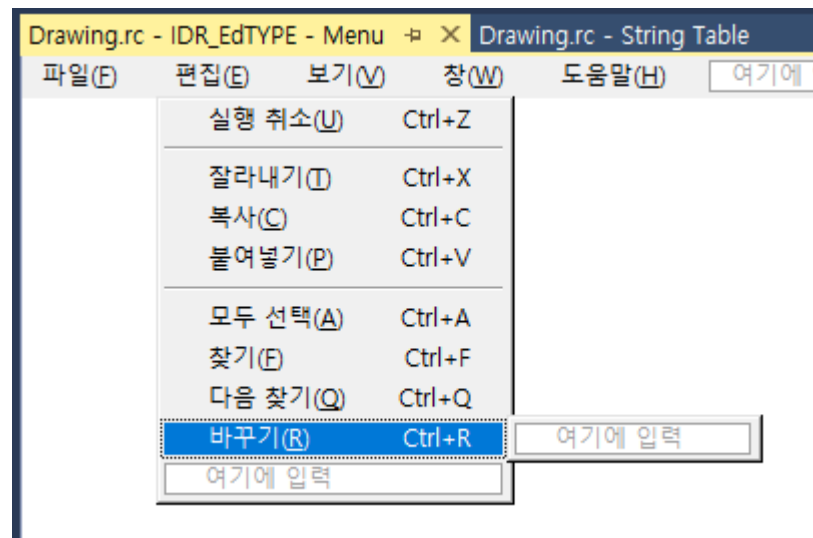
\\nDrawing\\nDrawing\\nDrawing Files (*.drw)\\n.DRW\\nDrawing.Document\\nDrawing Document

다중 문서 처리와 도큐먼트 템플릿

- 리소스 만들기

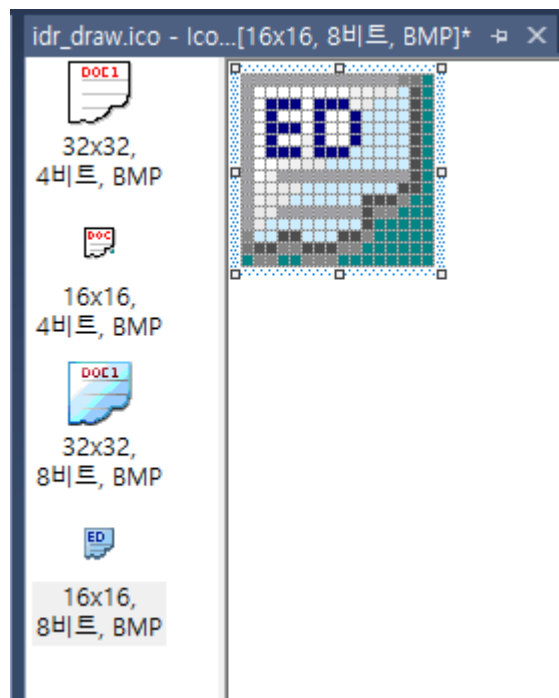
- Menu(IDR_EdTYPE)

- [선택] 메뉴 삭제 / [창 - 새 창] 삭제
- [모두 선택] / [찾기] / [다음 찾기] / [바꾸기] 추가
- 각각 ID_EDIT_SELECT_ALL, ID_EDIT_FIND, ID_EDIT_REPEAT, ID_EDIT_REPLACE
- 단축키 등록(5장의 내용 참고하여 단축키 등록하기 - 리소스 뷰의 Accelerator 이용)



다중 문서 처리와 도큐먼트 템플릿

- 리소스 만들기
 - Icon(IDR_EdTYPE)



과제 제출: 보고서 + 솔루션 폴더

- 내용
 - 각 과제의 프로그램 소스 및 설명
 - 각 과제의 실행 화면 캡처
- 제출 형식
 - 스노우보드에서 Lab10_이름_학번.doc 다운로드
 - 스노우보드에 제출: Lab10_이름_학번.zip (아래 두 파일 반드시 포함하도록)
 1. 소스 코드 : Lab10_이름_학번 (압축하지 않은 솔루션 폴더)
 2. 보고서: Lab10_이름_학번.doc (설명 없을 경우 감점)
- 제출 기한
 - 6월 1일(화) PM 11:59 까지