

7장. 그래픽 프로그램

-GDI와 DC를 이용하여 그래픽 출력

GDI

(Graphic Device Interface)

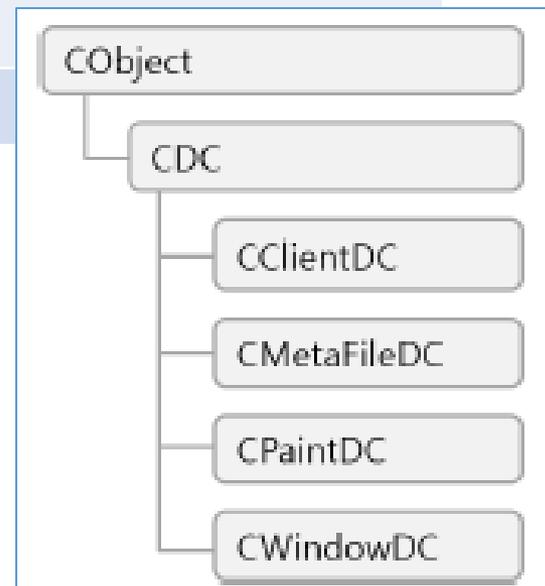
- **GDI**

- 모든 데이터 출력 시 사용되는 도구 (화면, 프린터 등)
- GDI 관련 클래스

CPen	점과 선 그리기
CBrush	내부 채우기
CFont	폰트 출력
CBitmap	미트맵 출력
CRgn	영역 만들기
CPallet	팔레트 조정

- **DC (Device Context)**

- 모든 출력이 이루어지는 대상
- DC 관련 클래스



CDC를 상속받은 클래스들

- **CClientDC**
 - 캡션바, 메뉴바, 상태바를 제외한 클라이언트 영역의 DC
- **CWindowDC**
 - 캡션바, 메뉴바, 상태바를 포함한 영역의 DC
- **CPaintDC**
 - WM_PAINT 메시지 발생시 OnPaint 함수에서만 사용하는 DC
- **CMetaFileDC**
 - Meta File 생성하고 지우기

CDC의 그리기 함수

- 선그리기 : pen이 적용됨
 - MoveTo(), LineTo()
- 도형그리기
 - Ellipse(), Rectangle()
- 윈도우 제공하는 자원 사용하기
 - SelectStockObject()

BLACK_BRUSH	검정색 브러시
GRAY_BRUSH	회색 브러시
LTGRAY_BRUSH	밝은 회색 브러시
WHITE_BRUSH	흰색 브러시
DKGRAY_BRUSH	어두운 회색 브러시
HOLLOW_BRUSH	속이 빈 브러시
BLACK_PEN	검정 펜
WHITE_PEN	흰색 펜
NULL_PEN / NULL_BRUSH	NULL PEN / NULL BRU
ANSI_FIXED_FONT	ANSI 고정 시스템 폰트
SYSTEM_FONT	시스템 폰트
DEFAULT_PALETTE	디폴트 컬러 팔레트 / 20

- Invalid region
 - 새로 그릴 필요가 있는 영역
 - InvalidateRect() : invalid region에 편입시켜서 WM_PAINT 메시지가 발생시킴

문자열 출력 프로그램 (Drawing1)

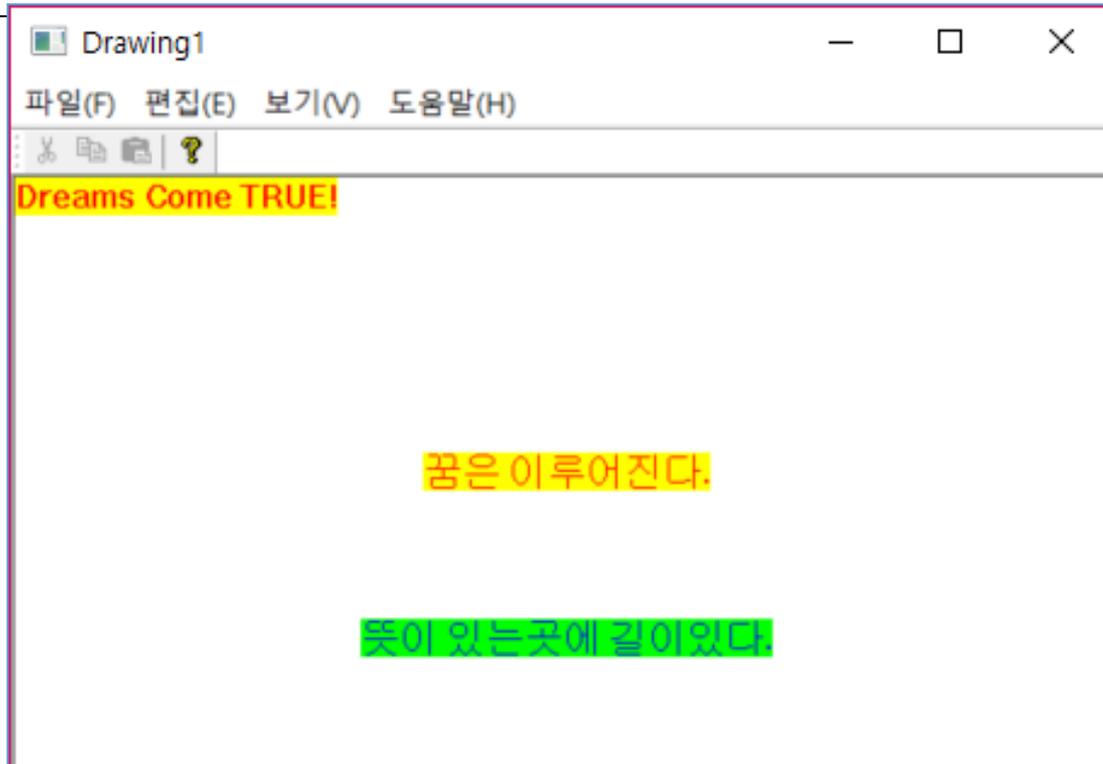
(1) 새 프로젝트 생성

(“파일”-“새로만들기”-“프로젝트”

-> “Visual C++” - “MFC” - “MFC 응용프로그램”)

-> 이름: **CDrawing1**

-> 응용 프로그램 종류: “**단일문서**” - “문서/뷰 아키텍처 지원”해제 -
“ActiveX 컨트롤”해제 - “마침”



문자열 출력 프로그램 (Drawing1)

(2) OnPaint() 함수에 작성

```
void CChildView::OnPaint() {  
    CPaintDC dc(this); // 그리기를 위한 디바이스 컨텍스트입니다.
```

```
    CRect rect;  
    GetClientRect(&rect);
```

클라이언트 영역 크기 가져오기

```
    dc.SetTextColor( RGB(255, 0, 0) );  
    dc.SetBkColor( RGB(255, 255, 0) );  
    dc.DrawText( CString( "Dreams Come TRUE!" ), &rect, 0 );  
    dc.DrawText( CString( "꿈은 이루어진다." ),  
    &rect, DT_CENTER | DT_VCENTER | DT_SINGLELINE );
```

dc의 특성을 지정하
고 문자열 출력

```
    dc.SetTextAlign( TA_CENTER );  
    dc.SetTextColor( RGB(0, 0, 255) );  
    dc.SetBkColor( RGB(0, 255, 0) );  
    dc.TextOutW( rect.right / 2, 3 * rect.bottom / 4,  
    CString( "뜻이 있는곳에 길이있다." ) );
```

dc의 특성을 지정하
고 문자열 출력

```
}
```

그림 그리기 프로그램 (Drawing2)

(1) 새 프로젝트 생성

(“파일”-“새로만들기”-“프로젝트”

-> “Visual C++” - “MFC” - “MFC 응용프로그램”)

-> 이름: **CDrawing2**

-> 응용 프로그램 종류: “**단일문서**” - “문서/뷰 아키텍처 지원”해제 -
“ActiveX 컨트롤”해제 - “마침”

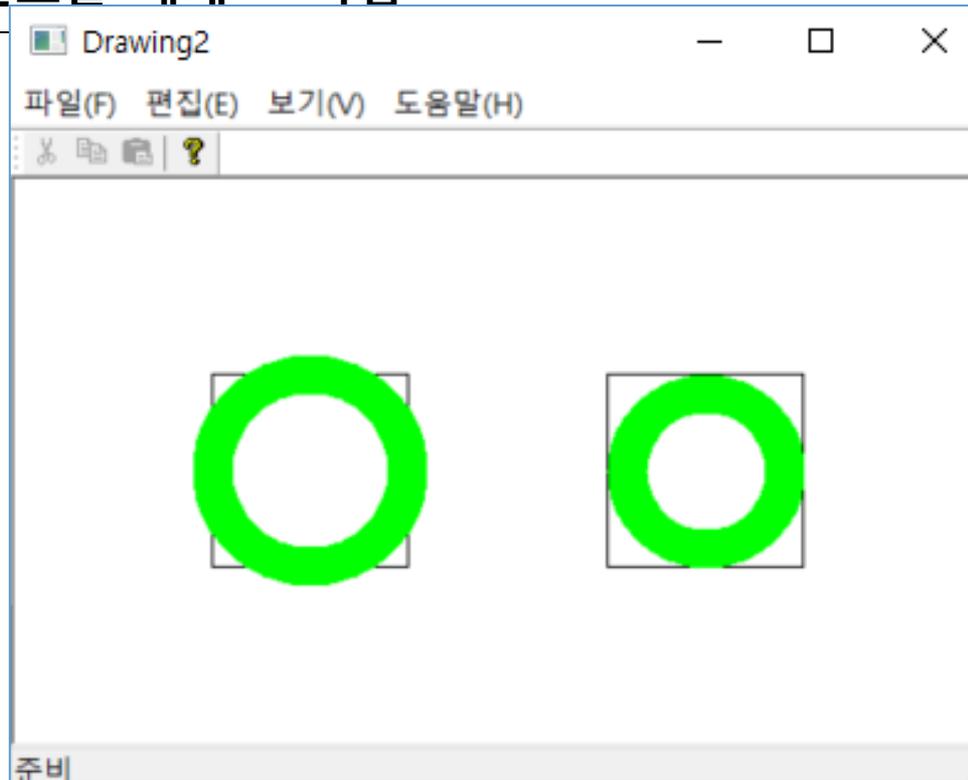


그림 그리기 프로그램 (Drawing2)

(2) OnPaint() 함수에 작성

```
void CChildView::OnPaint() {  
    CPaintDC dc(this); // 그리기를 위한 디바이스 컨텍스트입니다.
```

```
    CPen blackpen(PS_SOLID, 1, RGB(0, 0, 0));  
    CPen *pOldPen = dc.SelectObject(&blackpen);  
    dc.Rectangle(100, 100, 200, 200);
```

blackpen으로 그리기

```
    CPen pen1(PS_SOLID, 20, RGB(0, 255, 0));  
    dc.SelectObject(&pen1);  
    dc.Ellipse(100, 100, 200, 200);
```

pen1으로 그리기

```
    dc.SelectObject(&blackpen);  
    dc.Rectangle(300, 100, 400, 200);
```

blackpen으로 그리기

```
    CPen pen2(PS_INSIDEFRAME, 20, RGB(0, 255, 0));  
    dc.SelectObject(&pen2);  
    dc.Ellipse(300, 100, 400, 200);
```

pen2로 그리기

```
    dc.SelectObject(pOldPen);
```

원래 pen으로 되돌려주기

```
}
```

그래픽 프로그램 (SEditGraphic)

(1) 새 프로젝트 생성

(“파일”-“새로만들기”-“프로젝트”

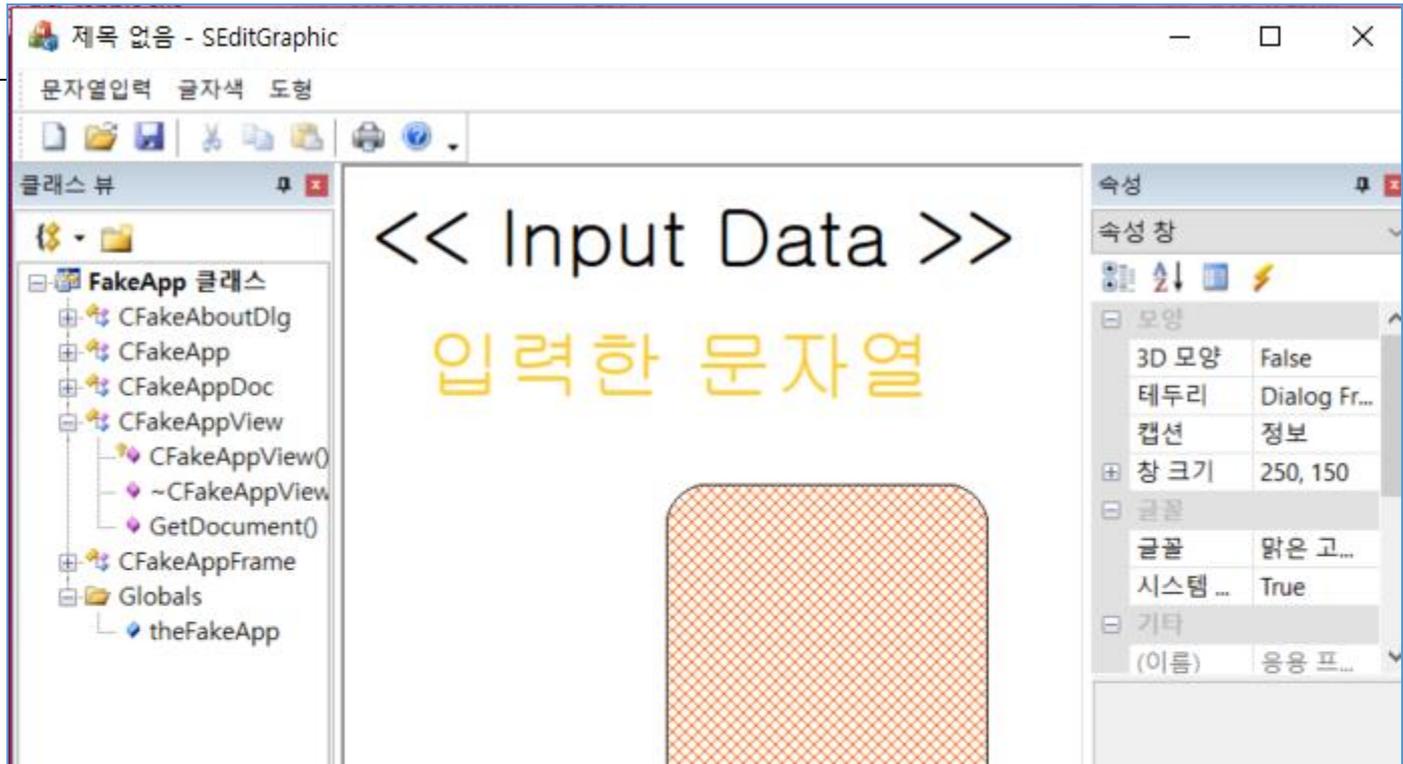
-> “Visual C++” - “MFC” - “MFC 응용프로그램”)

-> 이름: **SEditGraphic**

-> 응용 프로그램 종류: “단일문서” - 프로젝트 스타일: “Visual studio”

- 비주얼스타일 및 색: “Windows 원형/기본” - “ActiveX 컨트롤”해제 -

“마침”



그래픽 프로그램 (SEditGraphic)

(2) 메뉴 편집

리소스 뷰 - 메뉴 - IDR_MAINFRAME : 기존 항목들 모두 지우고 아래와 같이 새로 만들기.

문자열 입력 - 입력(ID_INPUT)

글자색 - 빨간색 글자 (ID_COLOR_RED)

글자색 - 파란색 글자 (ID_COLOR_BLUE)

글자색 - 노란색 글자 (ID_COLOR_YELLOW)

도형 - 원(ID_FIG_CIRCLE)

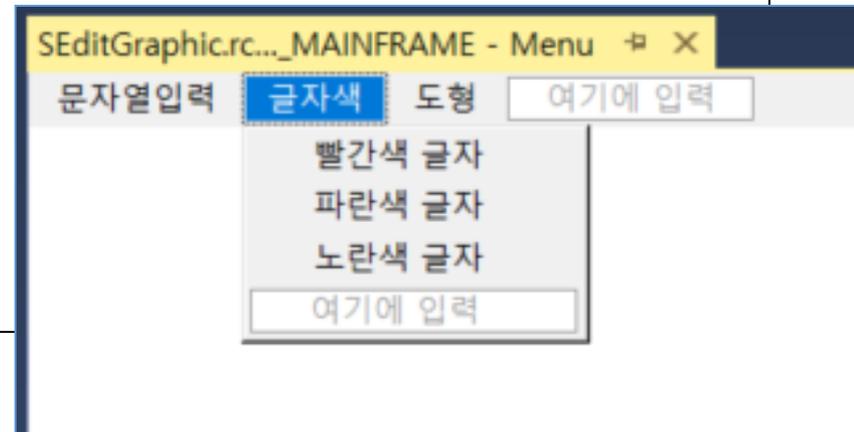
도형 - 사각형(ID_FIG_RECT)

도형 - 선(ID_FIG_LINE)

도형 - seperator

도형 - 십자무늬패턴(ID_FIG_CROSS)

도형 - 사과무늬패턴(ID_FIG_APPLE)



그래픽 프로그램 (SEditGraphic)

(3) 문자열 입력 대화상자 준비

- 리소스뷰 - "대화상자 추가"(IDD_TEXT_DIALOG)
- 아래와 같이 대화상자 만들고 m_strTextInput 멤버 변수 추가
- 대화상자에 클래스 추가. CTextDlg

The image shows a screenshot of the SEditGraphic application. At the top, a dialog box titled "문자열 입력" (String Input) is displayed. It contains a text input field labeled "생물 입력란" (Biology Input Field) and two buttons: "확인" (OK) and "취소" (Cancel). A red box highlights the text input field, and a red arrow points from it to the configuration panel below.

The configuration panel, titled "멤버 변수 추가 마법사 - SEditGraphic" (Member Variable Addition Wizard - SEditGraphic), shows the following settings:

- 액세스(A): public
- 변수 형식(V): CString
- 변수 이름(N):
- 컨트롤 변수(O):
- 컨트롤 ID(I): IDC_EDIT_TEXT_INPUT
- 컨트롤 형식(C):
- 변주(M): Value
- 최대 문자 수(M):

그래픽 프로그램 (SEditGraphic)

(4) Doc 클래스에 CString m_strData; 멤버 변수를 두어 "문자열 입력-" "입력" 메뉴에서 받은 문자열을 저장한다.

[SEditGraphicDoc.h – 멤버 변수 추가
("클래스 뷰"에서 클래스명에 R-click->"변수 추가")]

```
CString m_strData;
```

[CEditGraphic 클래스에 get/set 함수 추가
("클래스 뷰"에서 클래스명에 R-click->"함수 추가")]

```
CString CSEditGraphicDoc::GetText()  
{  
    return m_strData;  
}  
void CSEditGraphicDoc::SetText(CString strInput)  
{  
    m_strData = strInput;  
    UpdateAllViews(NULL);  
}
```

모든 view를 갱신

그래픽 프로그램 (SEditGraphic)

(5) CTextDlg에 editbox에 대한 value 멤버 변수를 만들어 활용.

[TextDlg.h – CTextDlg 클래스의 멤버]

```
CString m_strTextInput;
```

[“문자열 입력-“입력” 메뉴 처리 함수 구현(클래스마법사에서 “명령”탭 이용)]

```
void CSEditGraphicDoc::OnDataInput()  
{  
    CTextDlg m_TextDlg;  
    m_TextDlg.m_strTextInput = GetText();  
    if (m_TextDlg.DoModal() == IDOK)  
    {  
        SetText(m_TextDlg.m_strTextInput);  
    }  
}
```

그래픽 프로그램 (SEditGraphic)

(6) View 클래스의 OnDraw 함수 구현

```
void CSEditGraphicView::OnDraw(CDC* pDC) {  
    CSEditGraphicDoc* pDoc = GetDocument();  
    ASSERT_VALID(pDoc);  
    if (!pDoc)  
        return;  
  
    CFont font, *pOldFont;  
    font.CreatePointFont(300, _T("Broadway"));  
    pOldFont = (CFont*)pDC->SelectObject(&font);  
    CString strInput1 = _T("<< Input Data >>");  
    CString strInput2 = pDoc->GetText();  
    pDC->SetTextColor(RGB(0, 0, 0));  
    pDC->TextOut(0, 20, strInput1);  
    pDC->TextOut(50, 100, strInput2);  
    pDC->SelectObject( pOldFont );  
}
```

글자크기와 폰트 설정
30point font size (300/10=30)

글자색 변경

-> 실행 : view의 내용과 각 메뉴를 확인.

그래픽 프로그램 (SEditGraphic)

(7) Doc 클래스에 글자색 저장을 위한 변수 추가.

[SEditGraphicDoc.cpp 상단에 색깔 정의]

```
#define BLACK   RGB(0,0,0)
#define RED     RGB(255,0,0)
#define BLUE    RGB(51,51,255)
#define YELLOW  RGB(255,204,51)
```

[CSEditGraphicDoc 클래스에 색깔 저장용 멤버 변수 추가]

```
COLORREF m_curColor;  RGB(r, g, b) 매크로 사용 가능.
```

[CSEditGraphicDoc 클래스에 get/set 멤버 함수 추가]

```
void CSEditGraphicDoc::ColorSet(COLORREF color) {
    m_curColor = color;
    UpdateAllViews(NULL);
}
```

```
COLORREF CSEditGraphicDoc::ColorGet() {
    return m_curColor;
}
```

그래픽 프로그램 (SEditGraphic)

(8) 글자색 바꾸는 메뉴 구현

["글자색" - "빨간색 글자", "파란색 글자", "노란색 글자" 메뉴 구현
(클래스마법사에서 "명령"탭 이용)]

```
void CSEditGraphicDoc::OnColorBlue() {  
    ColorSet(BLUE);  
}  
void CSEditGraphicDoc::OnColorRed() {  
    ColorSet(RED);  
}  
void CSEditGraphicDoc::OnColorYellow() {  
    ColorSet(YELLOW);  
}
```

[OnDraw()에서 글자색 적용]

```
CString strInput2 = pDoc->GetText();  
pDC->SetTextColor(RED);  
pDC->TextOut(0, 20, strInput1);  
pDC->SetBkColor(RED);  
pDC->SetTextColor(pDoc->ColorGet());  
pDC->TextOut(50, 100, strInput2);
```

그래픽 프로그램 (SEditGraphic)

(9) Context menu에서도 글자색 바꾸기
-> view 영역에서 R-click하여 활용하기.

[리소스뷰의 메뉴 중 IDR_POPUP_EDIT 편집]

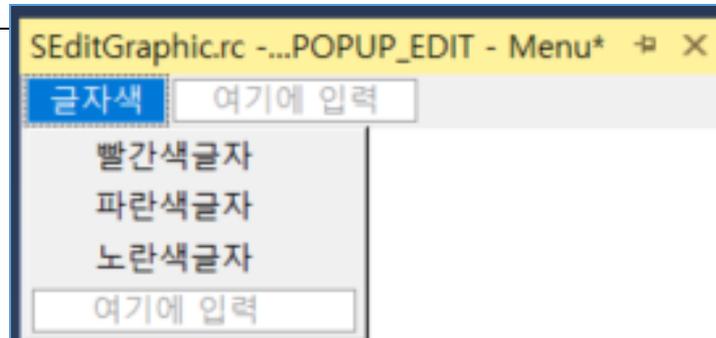
메뉴 ID를 각각

ID_COLOR_RED,

ID_COLOR_BLUE,

ID_COLOR_YELLOW

로 한다.



그래픽 프로그램 (SEditGraphic)

(10) View 클래스에 현재 도형의 종류를 기억할 멤버 변수 추가.

[CSEditGaphicView 클래스에 멤버 변수 추가]

```
int m_nOption;
```

[constructor에서 초기값 주기]

```
CSEditGraphicView::CSEditGraphicView() : m_nOption(0)
```

[도형모양 바꾸는 메뉴 처리 함수들 구현]

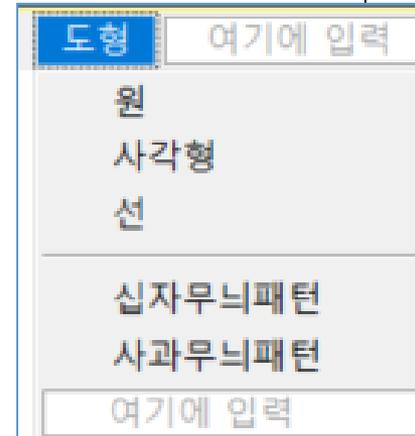
```
void CSEditGraphicView::OnFigCir() {  
    m_nOption = 1; Invalidate();  
}
```

```
void CSEditGraphicView::OnFigCross() {  
    m_nOption = 4; Invalidate();  
}
```

```
void CSEditGraphicView::OnFigLine() {  
    m_nOption = 3; Invalidate();  
}
```

```
void CSEditGraphicView::OnFigRect() {  
    m_nOption = 2; Invalidate();  
}
```

```
void CSEditGraphicView::OnFigApple() {  
    m_nOption = 5; Invalidate();  
}
```



그래픽 프로그램 (SEditGraphic)

(11) OnDraw()에서 원, 사각형 그리기.

```
switch (m_nOption) {
case 1: {
    CBrush brush(RGB(0, 255, 0));
    CBrush *pOldBrush = pDC->SelectObject(&brush);
    pDC->Ellipse(200, 200, 400, 400);
    pDC->SelectObject(pOldBrush);
    brush.DeleteObject();
    break;
}
case 2: {
    CBrush brush(RGB(150, 0, 150));
    CBrush *pOldBrush = pDC->SelectObject(&brush);
    pDC->Rectangle(200, 200, 400, 400);
    pDC->SelectObject(pOldBrush);
    brush.DeleteObject();
    break;
}
}
```

그래픽 프로그램 (SEditGraphic)

(12) OnDraw()에서 선 그리기.

```
case 3: {
    CDC* pDC = GetDC();
    int nPenType[] = { PS_SOLID, PS_DOT, PS_DASHDOT, PS_DASH, PS_DASHDOTDOT };
    for (int i = 0; i < sizeof(nPenType) / sizeof(nPenType[0]); i++) {
        CPen pen(nPenType[i], 1, RGB(0, 0, 255));
        pDC->SelectObject(&pen);
        /*----- 직선 -----*/
        pDC->MoveTo(100, 200 + i * 50);
        pDC->LineTo(500, 200 + i * 50);
        /*----- 사선 -----*/
        /*
        pDC->MoveTo(100, 200 + i * 50);
        pDC->LineTo(500, 300 + i * 50);
        */
    }
    break;
}
```

그래픽 프로그램 (SEditGraphic)

(13) OnDraw()에서 십자무늬패턴 그리기.

case 4:

```
{
    CBrush brush(HS_DIAGCROSS, RGB(255, 90, 0)); //Orange Color
    CBrush *pOldBrush = pDC->SelectObject(&brush); //Brush Object 선택
    택

    pDC->RoundRect(200, 200, 400, 400, 50, 50);

    pDC->SelectObject(pOldBrush);
    brush.DeleteObject(); //Brush Object 제거
    break;
}
```

그래픽 프로그램 (SEditGraphic)

(14) OnDraw()에서 사과무늬패턴 그리기.

[“리소스뷰” - “리소스 추가” - “Bitmap” - IDB_BITMAP1 에 사과 그리기]

[OnDraw()에 그리는 코드 추가]

```
case 5: {
```

```
    CBitmap bitmap;  
    bitmap.LoadBitmap(IDB_BITMAP1);  
    CBrush brush(&bitmap);
```

Bitmap 브러시 생성

```
    CBrush *pOldBrush = pDC->SelectObject(&brush);  
    pDC->SetBkColor(TRANSPARENT);
```

보통 브러시처럼
select하여 사용

```
    pDC->RoundRect(200, 200, 400, 400, 50, 50);
```

```
    pDC->SelectObject(&brush);  
    brush.DeleteObject();
```

```
    break;
```

```
}
```

