

제7장 그래픽 프로그램

윈도우 프로그래밍 설계

청주대학교 전자공학과
한철수

- 그래픽 관련 객체의 이해
- 문자열 프로그램 작성
- 그림 그리기 프로그램 작성
- 간단한 그래픽 프로그램 작성

GDI와 DC

- 그래픽 출력은 그래픽 관련 객체들에 의해 이루어짐.
 - GDI (Graphic Device Interface)
 - DC (Device Context)
- GDI는 그래픽 출력의 세세한 설정 함수 또는 그 함수들을 포함하는 객체임.
 - 펜, 브러시, 글자 폰트 등을 설정할 수 있는 GDI 객체가 있음.
- DC는 그래픽 출력이 이루어지는 객체임.
 - 한 개의 DC는 내부적으로 한 개의 펜과 한 개의 브러시만을 사용하므로, 선 또는 면의 정보를 바꾸고 싶다면 펜 또는 브러시 GDI 객체의 설정을 새롭게 바꾸어 주고 나서 그림을 그려야 함.
- 그림을 그릴 때 사용 중인 종이를 DC라고 할 수 있고, 붓이나 연필 등의 도구를 GDI라고 할 수 있음.

GDI 객체

- GDI 객체에는 그래픽 관련 설정 값들이 저장됨.
- GDI 객체의 값을 변경한 후 DC에 적용시켜야만 GDI 설정대로 그래픽이 출력됨.
- GDI 객체의 Create() 계열의 함수는 GDI 객체를 설정하는 함수이고, DC 객체의 SelectObject() 함수는 GDI 객체를 DC에 적용시키는 함수임.

- 예

```
CPen pen; // 펜 GDI 객체를 생성
pen.CreatePen(PS_DOT, 3, RGB(0,255,0)); // 펜 속성을 설정
CClientDC dc(this); // DC 가져오기
CPen* old_pen=dc.SelectObject(&pen); // DC에 새 펜 설정 적용
dc.Rectangle(100,100,200,200); // DC로 그림 그리기
dc.SelectObject(old_pen); // DC의 기존 펜 설정 복구
```

많이 사용되는 GDI 객체들

- CPen
 - 선과 관련된 객체.
 - 선의 두께, 선 모양(점선, 실선 등), 색깔 등을 설정할 수 있음.
- CBrush
 - 면과 관련된 객체.
 - 면의 색깔, 면의 패턴 등을 설정할 수 있음.
- CFont
 - 글자 폰트와 관련된 객체.
- CBitmap
 - 비트맵 파일(그림 파일)과 관련된 객체.

예제 1

```
CClientDC dc(this);           // DC 가져오기.  
dc.MoveTo(10,10);           // 선의 시작 위치 설정.  
dc.LineTo(30,30);          // 시작 위치부터 (30,30)까지 선을 긋기.
```

예제 2

```
CPen pen; // 펜 GDI 객체 생성.
pen.CreatePen(PS_DOT, 3, RGB(0,255,0)); // 펜 속성 설정.
CClientDC dc(this); // DC 가져오기.
CPen* old_pen=dc.SelectObject(&pen); // DC에 GDI 적용.
dc.MoveTo(30,30); // 선 시작 위치 설정.
dc.LineTo(10,30); // 시작 위치부터 (10,30)까지 선을 긋기.
dc.SelectObject(old_pen); // 기존 펜 설정 복구.
```

예제 3

```
CPen pen; // 펜 GDI 객체 생성.
pen.CreatePen(PS_DOT, 3, RGB(0,255,0)); // 펜 속성 설정.
CClientDC dc(this); // DC 가져오기.
CPen* old_pen=dc.SelectObject(&pen); // DC에 GDI 적용.
dc.MoveTo(30,30); // 선 시작 위치 설정.
dc.LineTo(10,30); // 시작 위치부터 (10,30)까지 선을 긋기.
dc.SelectObject(old_pen); // 기존 펜 설정 복구.

pen.DeleteObject(); // 기존의 펜 정보 삭제.
pen.CreatePen(PS_SOLID, 6, RGB(255,0,0)); // 펜 속성 설정.
old_pen=dc.SelectObject(&pen); // DC에 GDI 적용.
dc.MoveTo(10,30); // 선 시작 위치 설정.
dc.LineTo(10,10); // 시작 위치부터 (10,10)까지 선을 긋기.
dc.SelectObject(old_pen); // 기존 펜 설정 복구.
```

DC의 종류

- DC는 출력 장치에 문자나 그림을 출력하기 위한 객체임.
- 프로그램의 모든 출력은 DC에 의해 이루어짐.
- DC를 관리하기 위한 클래스로 CDC가 있음.
- CDC의 파생 클래스
 - CClientDC
 - CWindowDC
 - CPaintDC
 - CMetaFileDC
- CDC 객체를 가져오는 방법
 - 방법 1 (CWnd 클래스의 파생클래스 내에서, 예: CView, CDialog 등)
 - `CDC* pDC=GetDC();`
 - 방법 2 (Cview 클래스 내에서)
 - OnDraw() 함수에서는 주석 처리된 매개 변수의 주석을 해제하여 CDC 객체를 사용할 수 있음.

DC의 종류

- CDC
 - CClientDC
 - 클라이언트 영역에서의 처리만 가능함.
 - CWindowDC
 - 캡션 바, 메뉴바, 상태 바 등을 포함한 윈도우의 모든 영역에서의 처리가 가능함.
 - CPaintDC
 - WM_PAINT 메시지 처리기 내에서만 사용이 가능함.
 - CMetaFileDC
 - 실행 순서를 고려한 GDI 객체의 시퀀스를 갖는 DC.

CDC의 멤버 함수

- GDI 적용 함수
 - SelectObject // GDI 객체를 DC에 적용시킴.
 - SelectStockObject // 미리 정의된 GDI 객체를 DC에 적용시킴.
- 글자 관련
 - SetBkColor // 글자의 배경 색 지정.
 - SetTextColor // 글자의 색 지정.
 - SetTextAlign // 글자의 정렬 방법 지정.
 - TextOut // 글자 출력.
 - DrawText // 글자 출력.
- 선 관련
 - MoveTo // 선 시작 위치 설정.
 - LineTo // 시작 위치부터 지정 위치까지 선을 긋기.
- 도형 관련
 - Rectangle // 사각형 그리기.
 - Ellipse // 타원 그리기.
- 비트맵 관련
 - BitBlt // 비트 블릿. 비트맵 이미지를 그대로 그리기.
 - StretchBlt // 스트레치 블릿. 확대 축소하여 그리기.

화면을 다시 그리고 싶을 때

- Invalidate() 함수
 - 화면 전체를 다시 그리고 싶을 때 사용함.
- InvalidateRect()
 - 화면의 특정 영역만 다시 그리고 싶을 때 사용함.

문자열 프로그램

- 만들고자 하는 프로그램
 - GDI와 DC를 이용해 문자열을 다양한 형태로 표시함.



문자열 프로그램

```
void CChildView::OnPaint() ←
```

OnPaint() 함수는 WM_PAINT 메시지를 처리하는 함수임.

```
{
```

```
    CPaintDC dc(this); // 그리기를 위한 디바이스 컨텍스트입니다.
```

```
    // TODO: 여기에 메시지 처리기 코드를 추가합니다.
```

```
    CRect rect;
```

```
    GetClientRect(&rect); ←
```

GetClientRect() 함수는 클라이언트 영역의 좌표 값을 반환하는 함수임.

```
    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 );
```

```
    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 있는 곳에 길이 있다." ) );
```

```
    // 그리기 메시지에 대해서는 CWnd::OnPaint()를 호출하지 마십시오.
```

```
}
```



그림 그리기 프로그램

- 만들고자 하는 프로그램
 - GDI와 DC를 이용해 사각형과 원을 그림.

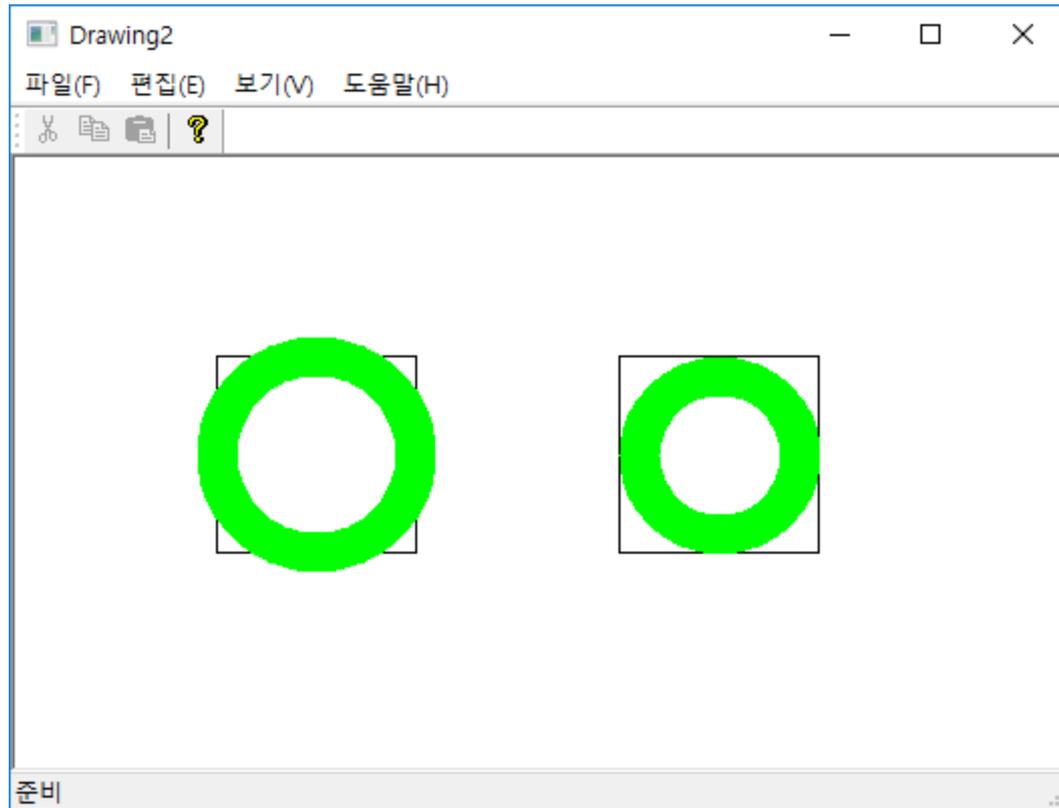


그림 그리기 프로그램

```

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

    // TODO: 여기에 메시지 처리기 코드를 추가합니다.
    CPen blackpen(PS_SOLID, 1, RGB(0, 0, 0));
    dc.SelectObject(&blackpen);
    dc.Rectangle(100, 100, 200, 200);

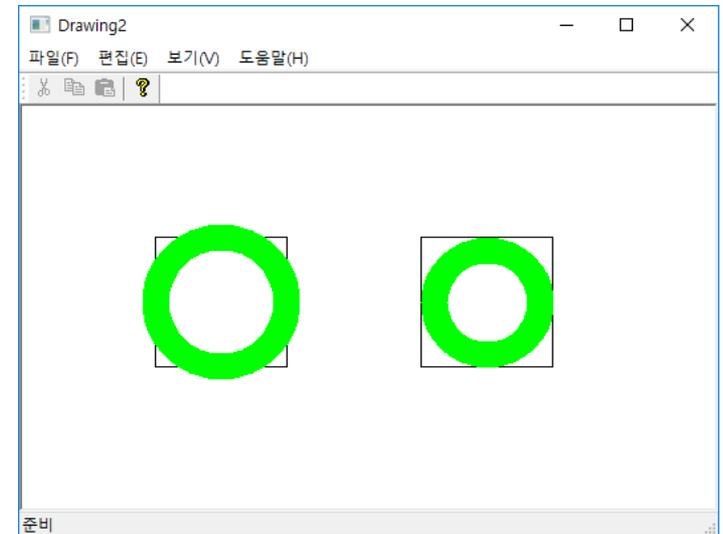
    CPen pen1(PS_SOLID, 20, RGB(0, 255, 0));
    dc.SelectObject(&pen1);
    dc.Ellipse(100, 100, 200, 200);

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

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

    // 그리기 메시지에 대해서는 CWnd::OnPaint()를 호출하지 마십시오.
}

```



그래픽 프로그램

- 만들고자 하는 프로그램
 - 메뉴가 있는 그래픽 에디터 프로그램을 작성함.



질문

Q & A