

객체지향 프로그래밍 응용

Chap 3. GDI를 이용한 그래픽

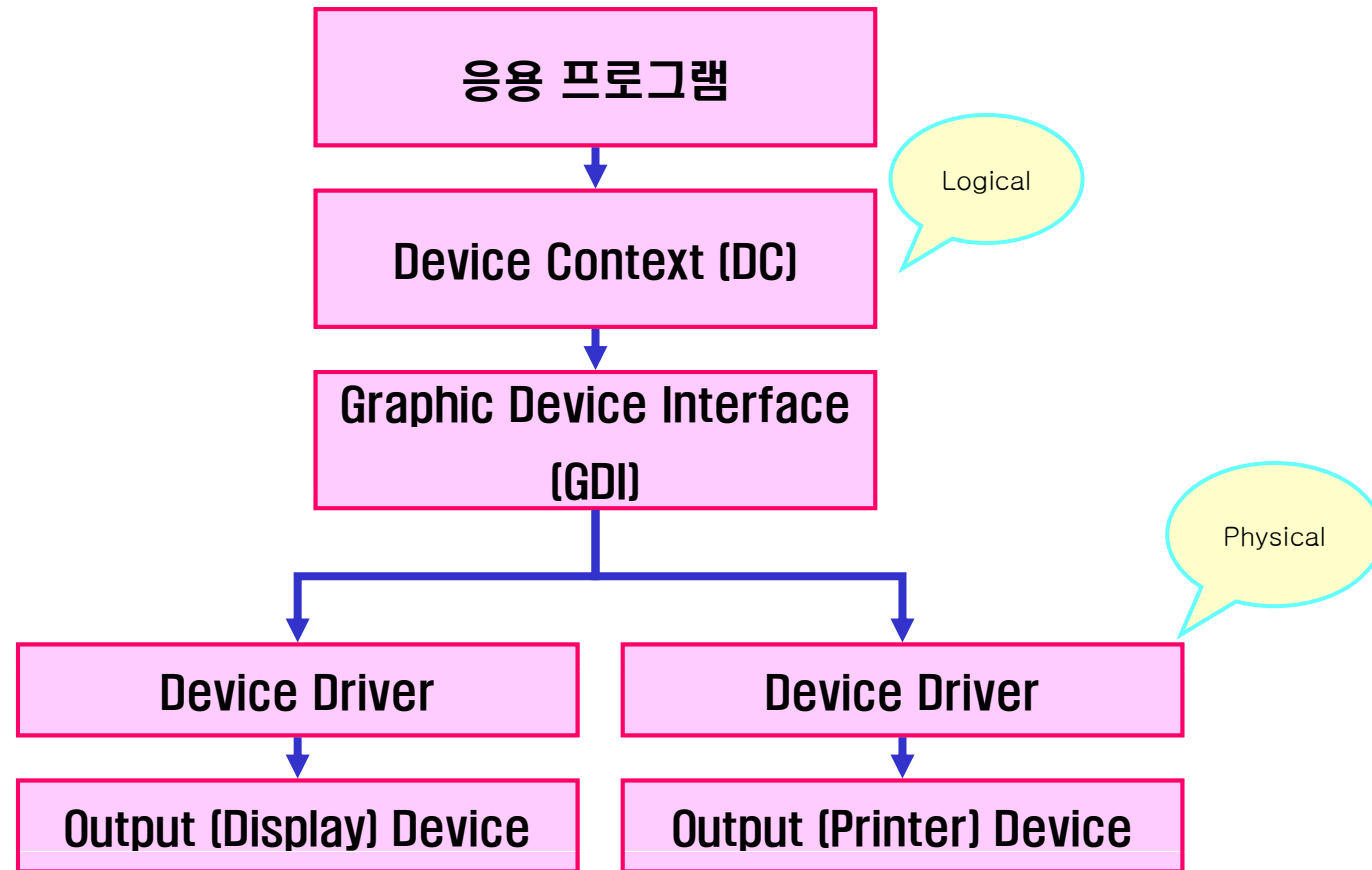
2012.09.24.

오 병 우

컴퓨터공학과
금오공과대학교

윈도우 그래픽의 이해

Device-Independent Graphics



GDI

■ GDI: Graphic Device Interface

- ◆ 그래픽 기능과 관련해서 윈도우 운영체제가 응용 프로그램에 제공하는 모든 기능

■ DC: Device Context

- ◆ GDI에게 알려줄 각종 출력 정보(색, 폰트 등)를 한 곳에 모아둔 구조체

■ GDI Object

- ◆ 독립적으로 저장되는 각 범주의 그래픽 옵션

범주	GDI 오브젝트	기본값
선 그리기 옵션	펜 (Pen)	1 픽셀 굵기의 검은색 실선
영역의 내부를 채우는 옵션	브러시 (Brush)	단일 흰색 브러시
글꼴 옵션	글꼴 (Font)	시스템 글꼴
비트맵 옵션	비트맵 (Bitmap)	없음
팔레트 옵션	팔레트 (Palette)	없음
영역 옵션	영역 (Region)	없음

그래픽 관련 MFC 클래스

■ CDC 클래스

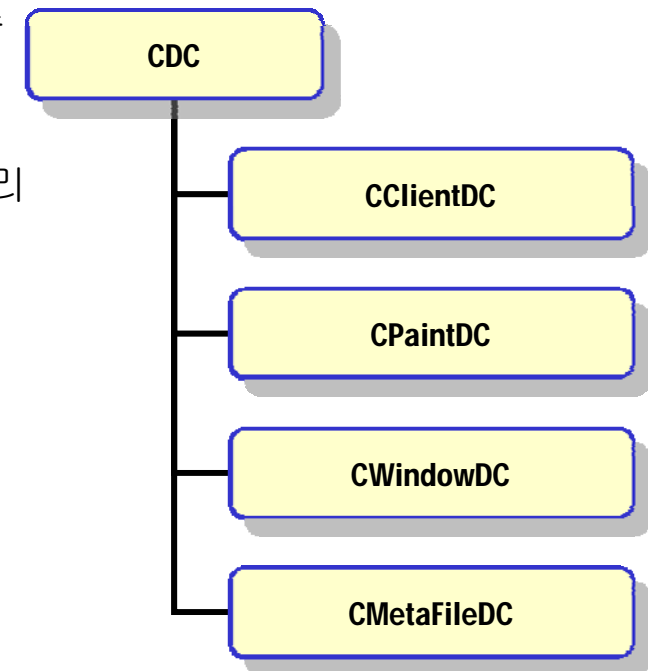
◆ Device Context 캡슐화 클래스

- 오브젝트를 통해 그래픽에 관련된 옵션을 저장하고 메소드 포함

◆ 모든 그래픽 함수들은 CDC 클래스의 멤버 함수

- CClientDC: 클라이언트 영역
 - Constructor에서 GetDC() and Destructor에서 ReleaseDC() 호출
- CPaintDC: WM_PAINT 발생시 Invalid Region 처리
 - BeginPaint() and EndPaint()
- CWindowDC: 창 전체
 - GetWindowDC() and ReleaseDC()
- CMetaFileDC: 메타 파일 () 처리
 - 그래픽 명령 저장 파일, .wmf, .emf

```
CDC *pDC = GetDC();
pDC->Rectangle(10, 10, 100, 100);
ReleaseDC(pDC); // 반드시 필요
```



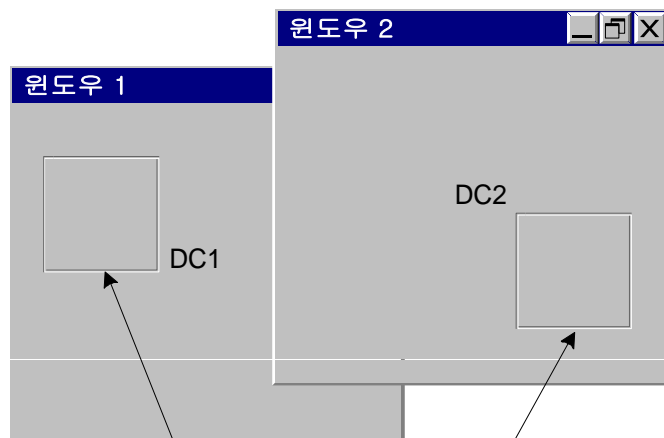
CClientDC

□ CDC 클래스에서 상속

- ◆ 생성자 함수에서 GetDC를 호출
- ◆ 소멸자 함수에서 ReleaseDC를 호출

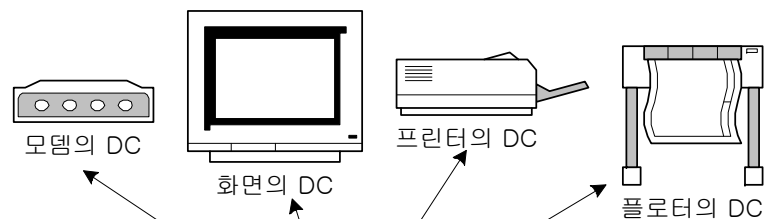
OnDraw()에서 사용하는 방식

```
CClientDC dc(this);
dc.Rectangle(10, 10, 100, 100);
```



DC1.Rectangle(10, 10, 100, 100)

DC2.Rectangle(200, 200, 300, 300)



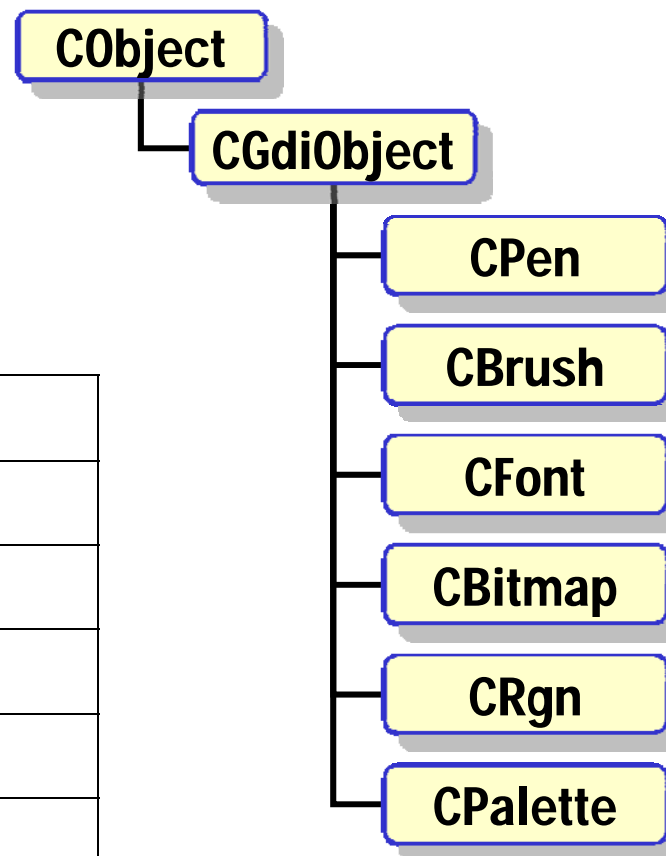
DC.Rectangle(10, 10, 100,100);

GDI 오브젝트 클래스

▣ 그래픽 옵션 변경

- ◆ 새로운 설정을 갖는 GDI Object 생성
- ◆ DC에서 SelectObject()
- ◆ DC를 이용하여 그래픽 작업
- ◆ 원래의 GDI Object로 복원 위해 SelectObject()
 - 또는 SaveDC() 및 RestoreDC() 사용

GDI 오브젝트	MFC 클래스
펜	CPen
브러시(붓)	CBrush
글꼴	CFont
비트맵	CBitmap
팔레트	CPalette
영역	CRgn



펜과 브러시

□ 펜과 브러시를 이용한 그래픽

- ◆ 현재 DC에 설정되어 있는 펜을 이용하여 경계선을 그림
- ◆ 폐곡선일 경우 안쪽을 현재 DC에 설정되어 있는 브러시로 채움

함수	그리기 기능
MoveTo, LineTo	선
Rectangle, FillRect, FrameRect, Draw3dRect	사각형
Ellipse	타원
Pie	파이
Arc, ArcTo, AngleArc, Chord	호
Polygon	다각형
PolyDraw, PolyBezier, PolyBezierTo	베지어 곡선
FrameRgn	영역의 경계선

그래픽 옵션 바꾸기

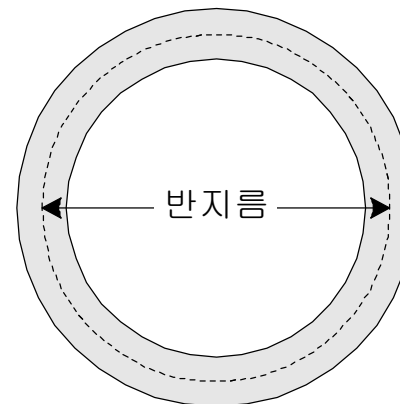
□ 펜 생성 예제

```

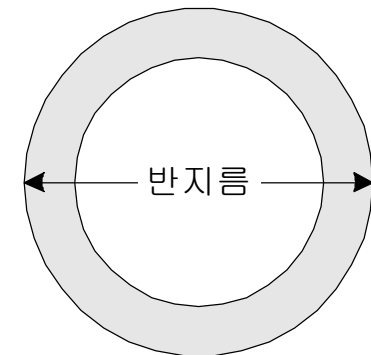
CPen pen;
pen.CreatePen(PS_SOLID, 3, RGB(255, 0, 0));
CClientDC dc(this);
dc.SelectObject(&pen);
dc.Rectangle(10, 10, 100, 100);
    
```

어떤 문제가 존재?

- PS_SOLID _____
- PS_DASH _____
- PS_DOT (dashed line)
- PS_DASHDOT _____
- PS_DASHDOTDOT _____
- PS_NULL _____
- PS_INSIDEFRAME _____



PS_SOLID



PS_INSIDEFRAME

그래픽 옵션 바꾸기

- 새로운 펜으로 선택하고 되돌리지 않으면 함수가 종료되는 순간에 DC에 저장되어 있던 펜 오브젝트는 파괴
- `SelectObject` 함수의 리턴값으로 넘어오는 GDI 오브젝트 저장

```
CPen pen;  
pen.CreatePen(PS_SOLID, 3, RGB(255, 0, 0));  
CClientDC dc(this);  
  
CPen *pOldPen = (CPen *)dc.SelectObject(&pen);  
dc.Rectangle(10, 10, 100, 100);  
  
dc.SelectObject(pOldPen);
```

내장 (Stock) GDI 오브젝트

- 자주 쓰이는 스타일의 GDI 오브젝트는 윈도우 운영체제가 내장
- GetStockObject 함수 사용

```
CClientDC dc;  
dc.SelectObject( GetStockObject(WHITE_PEN) );
```

- 내장 오브젝트는 항상 메모리에 상주

Raster Operation

- 새로 그릴 그림과 기존에 그려져 있는 그림을 합성
- `int SetROP2(int nDrawMode)`
 - ◆ `R2_COPYPEN`
 - 새로이 그려지는 그림으로 화면에 덮어 그림
 - ◆ `R2_XORPEN`
 - 배경을 깨뜨리지 않고, 배경 위에서 움직이는 그림을 그릴 때 사용
 - 똑같은 그림을 두 번 그려주면 원래 있던 바탕색이 복원

바탕색	출력되는 색
흰색	새로 그려진 색이 반전되어 출력됩니다.
검정색	새로 그려진 색이 그대로 출력됩니다.
임의의 색	새로 그려진 색과 바탕색이 XOR 연산되어 출력되는데, 이 값은 바탕색과 새로 그려진 색 양쪽과 모두 무관한 엉뚱한 색이 됩니다.

Text 함수

□ 텍스트 출력 함수

◆ TextOut()

- 지정한 좌표에 텍스트 출력

◆ TabbedTextOut()

- 탭 문자 처리 가능

◆ ExTextOut()

- 출력 영역 지정 가능

◆ DrawText()

- 사각형 영역에 텍스트 출력

□ 텍스트 속성 함수

◆ GetTextColor() and SetTextColor()

◆ GetTextAlign() and SetTextAlign()

◆ GetBkColor() and SetBkColor()

◆ GetBkMode() and SetBkMode()

Font

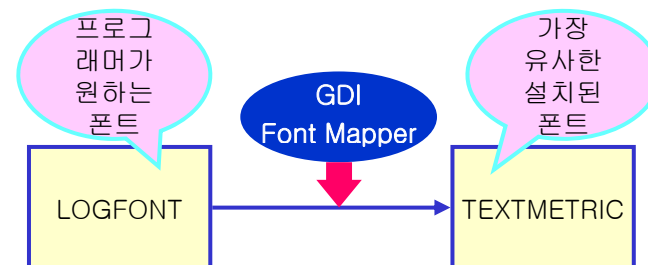
□ 관련 구조체

◆ LOGFONT

- Logical Font
- Defines the attributes of a font
 - Height, Underline, Italic, charset (Character Set: 어떤 문자 사용 - HANGUL_CHARSET) etc.

◆ TEXTMETRIC

- Physical Font
- Contains basic information about a physical font



Font 사용

□ 사용 예제

```
CClientDC dc(this);  
CFont font, *oldfont;  
  
font.CreatePointFont(400, "MD아롱체");  
  
oldfont = dc.SelectObject(&font);  
  
dc.TextOut(point.x + 2, point.y + 2, GetDocument()->GetString());  
  
dc.SelectObject(oldfont);
```

Font Dialog 사용

CFontDialog
에서 설명

색깔은?

□ 사용 예제

```

CClientDC dc(this);
CFont font, *oldfont;
/*
LOGFONT logfont;
CFontDialog dlg; // Default Font 없음
*/
static LOGFONT logfont; // 멤버변수 또는 static이면
CFontDialog dlg(&logfont); // 이전의 폰트를 Default Font로 설정 가능

if (dlg.DoModal() == IDOK) {
    dlg.GetCurrentFont(&logfont);

    font.CreateFontIndirect(&logfont);

    oldfont = dc.SelectObject(&font);
    dc.TextOut(point.x + 2, point.y + 2, GetDocument()->string);
    dc.SelectObject(oldfont);
}
    
```