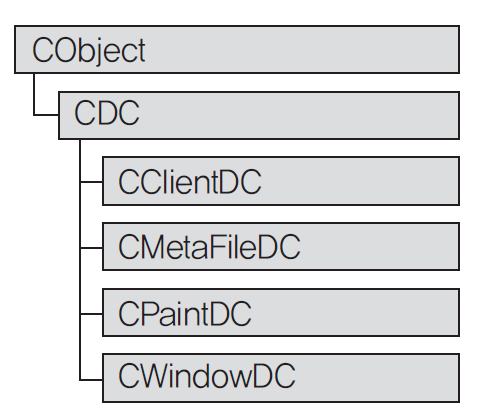
|  |  |
| --- | --- |
| CPaintDC | 클라이언트 영역에 출력할 때  (WM\_PAINT 메시지 핸들러에서만 사용) |
| CClientDC | 클라이언트 영역에 출력할 때  (WM\_PAINT 메시지 핸들러를 제외한  다른 모든 곳에서 사용) |
| CWindowDC | 윈도우 전체 영역(클라이언트 영역 +  비클라이언트 영역)에 출력할 때 |
| CMetaFileDC | 메타파일(Metafile)에 출력할 때 |

* MFC 클래스 계층도



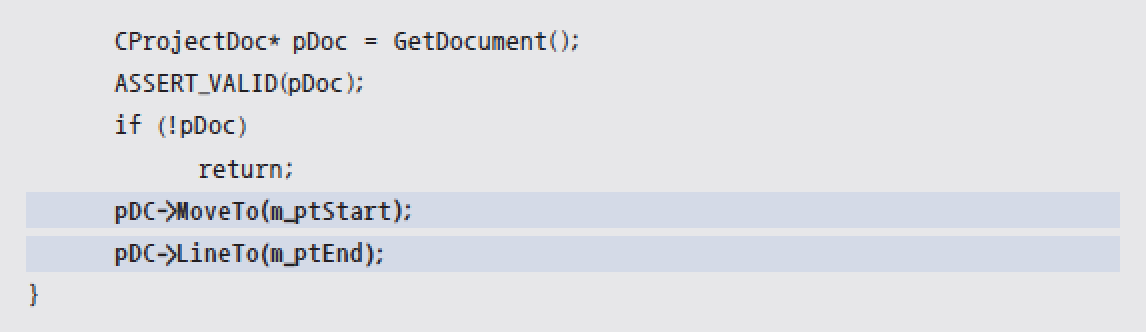
다양한 디바이스 컨텍스트 클래스

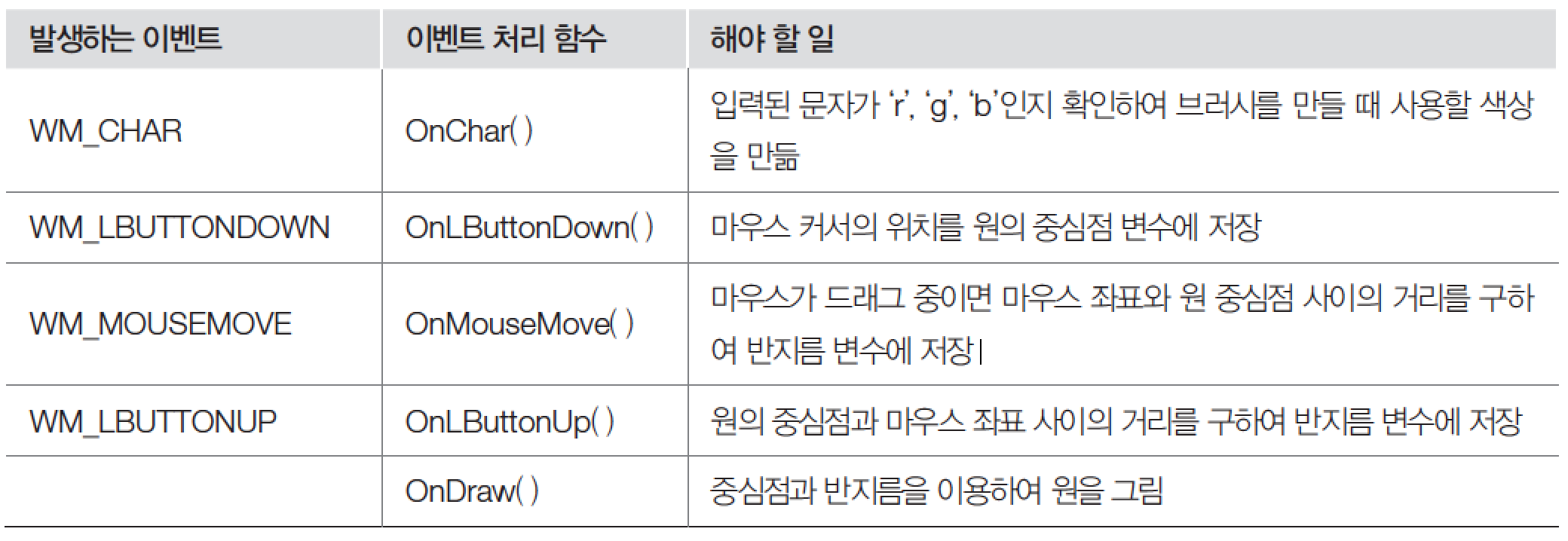
* MFC 프로그램 출력 과정
  + 디바이스 컨텍스트 객체를 생성한다.
  + 객체의 멤버 함수를 호출하여 출력한다.

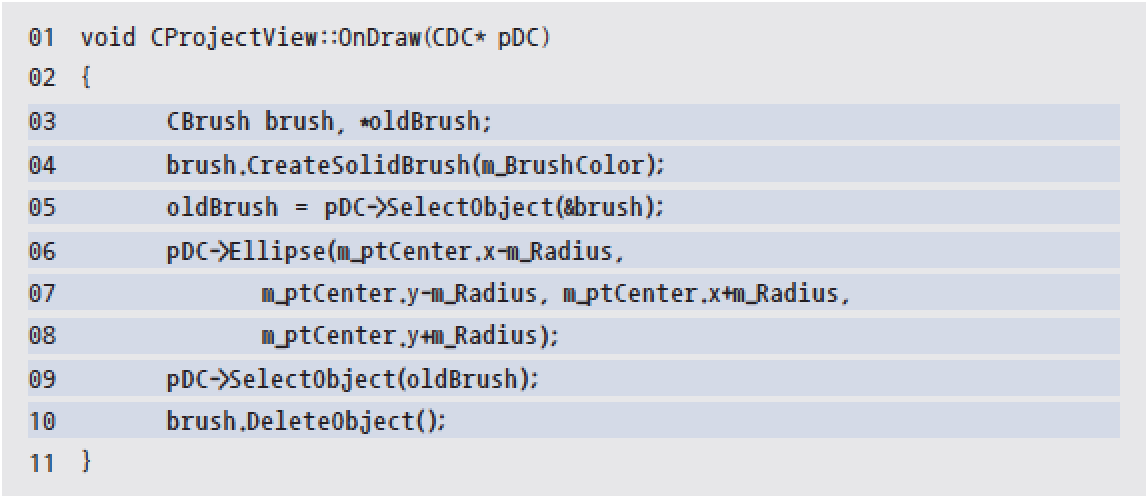
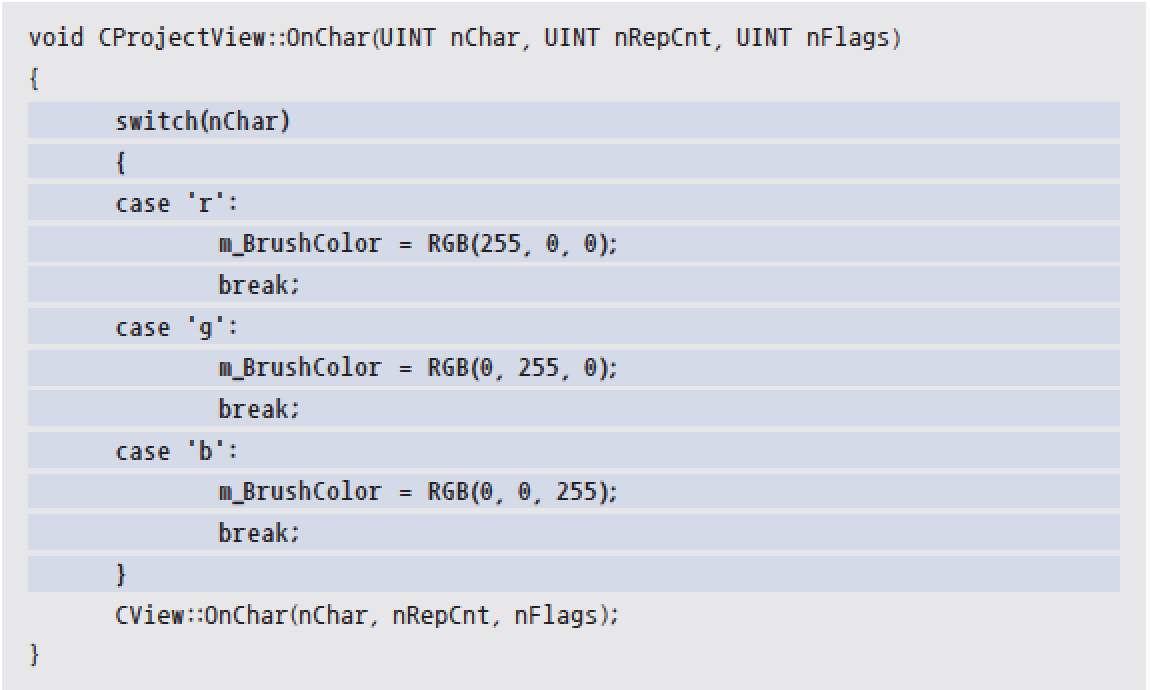
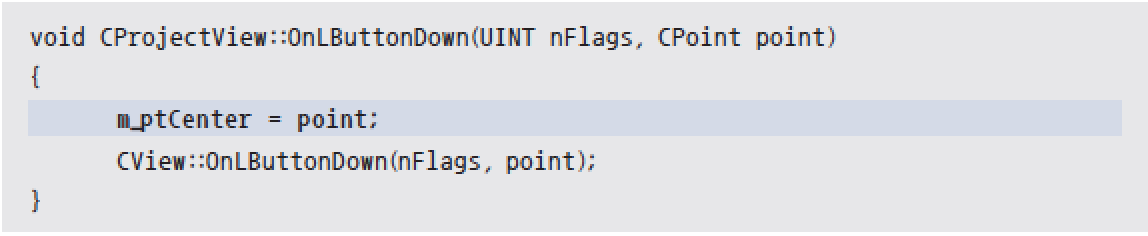
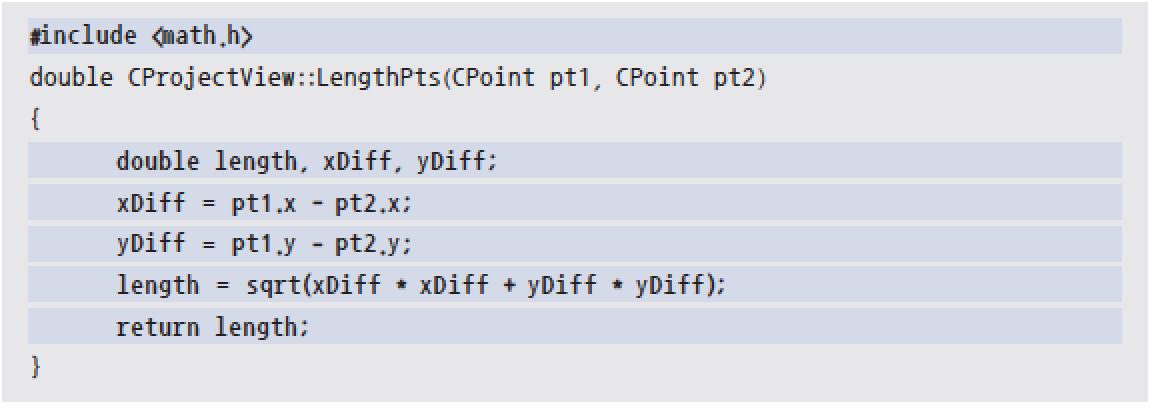
|  |  |
| --- | --- |
| 점찍기 | **기능** |
| GetPixel() | 화면 (x, y) 지점의 색을 얻는다. |
| SetPixel() | 화면 (x, y) 지점에 특정 색상의 점을 찍고, 원래 점의 색을 리턴한다. |
| SetPixelV() | SetPixel( ) 함수와 출력은 같지만, 원래 점의 색을 리턴하지 않으므로 속도가 좀더 빠르다. |

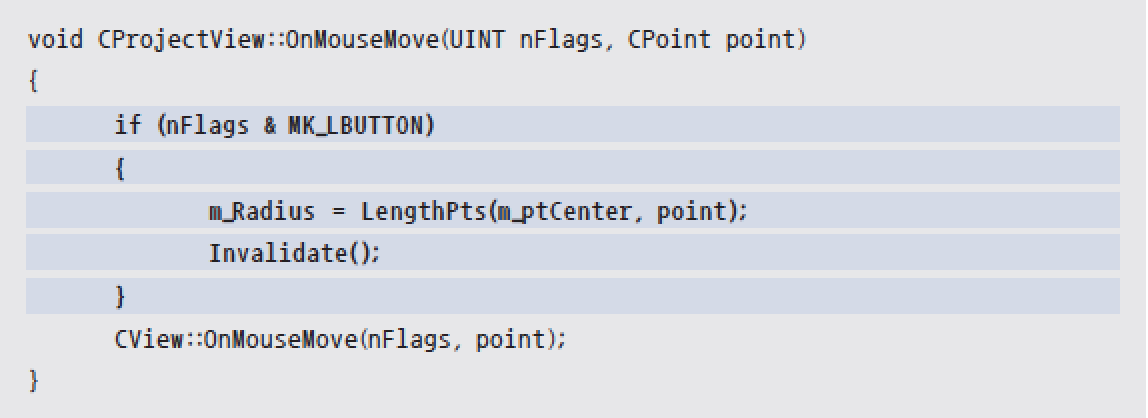
|  |  |
| --- | --- |
| 선 그리기 | **기능** |
| MoveTo() | 현재 위치를 (x, y) 위치로 옮긴다. |
| LineTo() | 현재 위치에서 (x, y) 위치까지 선을 그리고, 현재 위치를 (x, y)로 변경한다. |
| Polyline() | POINT 구조체 배열로 전달된 점들을 차례로 이어서 선을 그린다. |

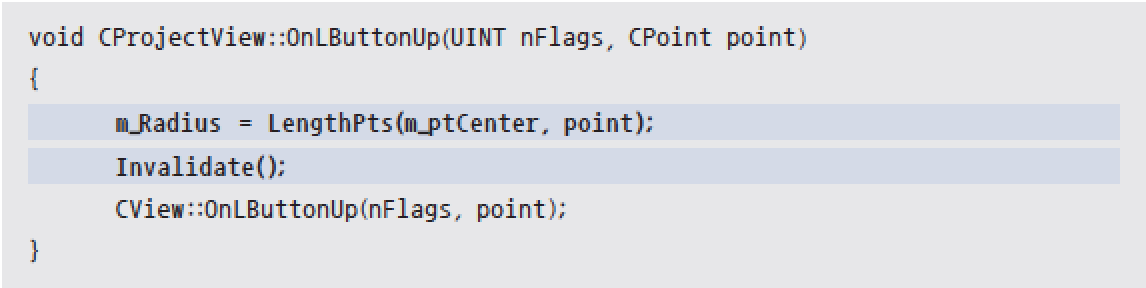
|  |  |
| --- | --- |
| 도형 그리기 | **기능** |
| Rectangle() | 직사각형을 그린다. |
| Ellipse() | 직사각형에 내접하는 타원을 그린다. |
| RoundRect() | 테두리가 둥근 직사각형을 그린다. |
| Polygon() | POINT 구조체 배열로 전달된 점들을 차례로 이어서 다각형을 그린다. |

OnDraw(CDC\* pDC)







펜, 브러쉬

cdc

CPaintDC dc(this); // 디바이스 컨텍스트 객체를 만든다.

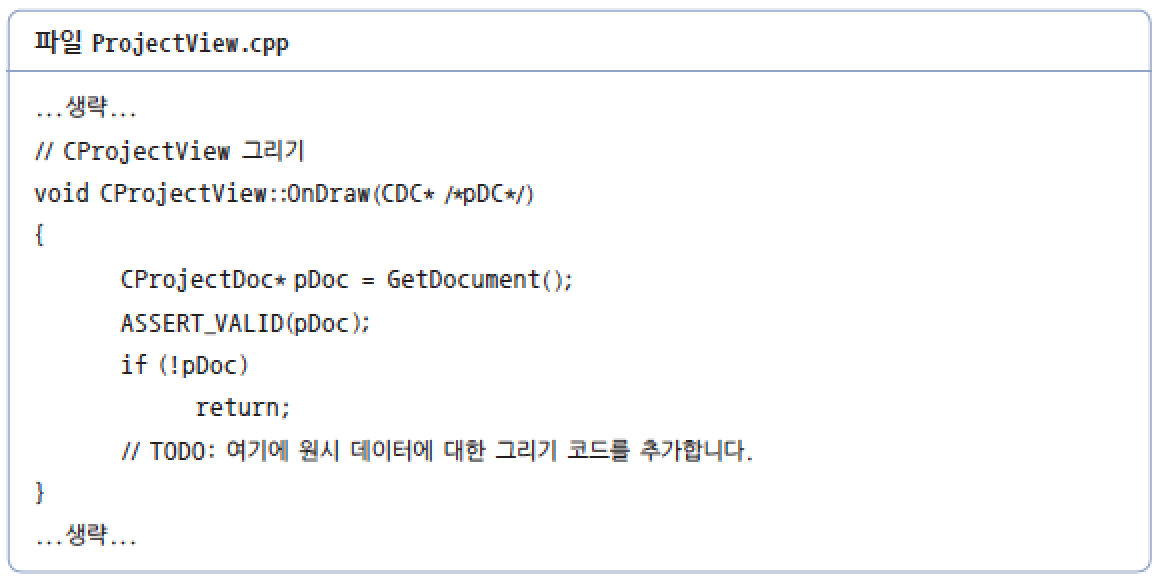
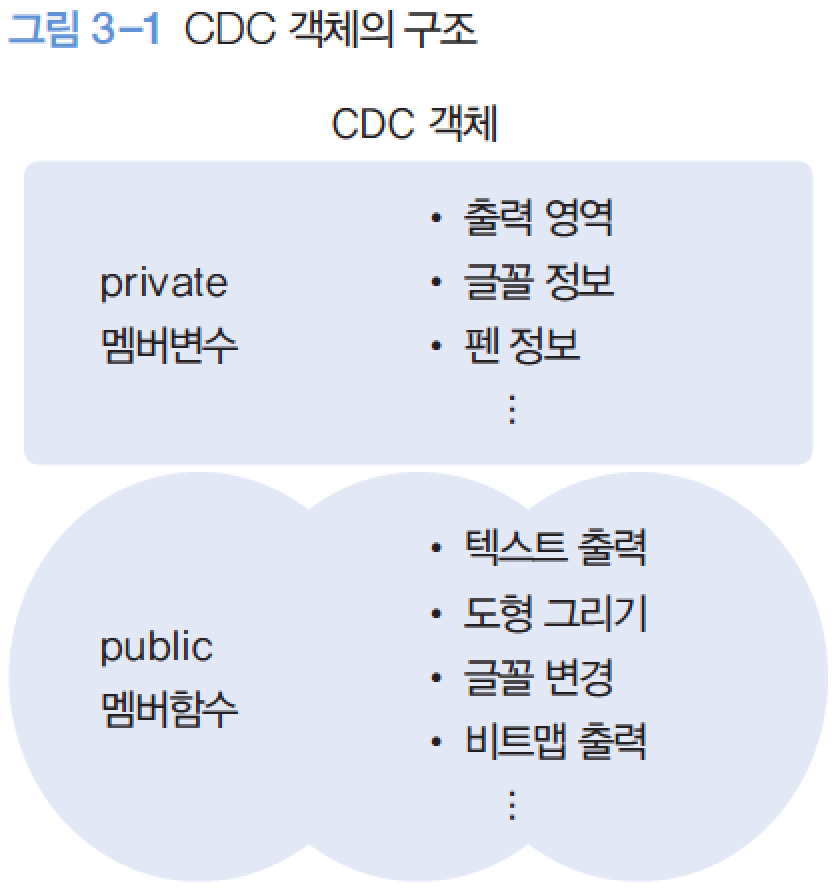
CPen pen(PS\_SOLID, 1, RGB(0, 0, 255)); // CBrush brush(RGB(255, 0, 0));

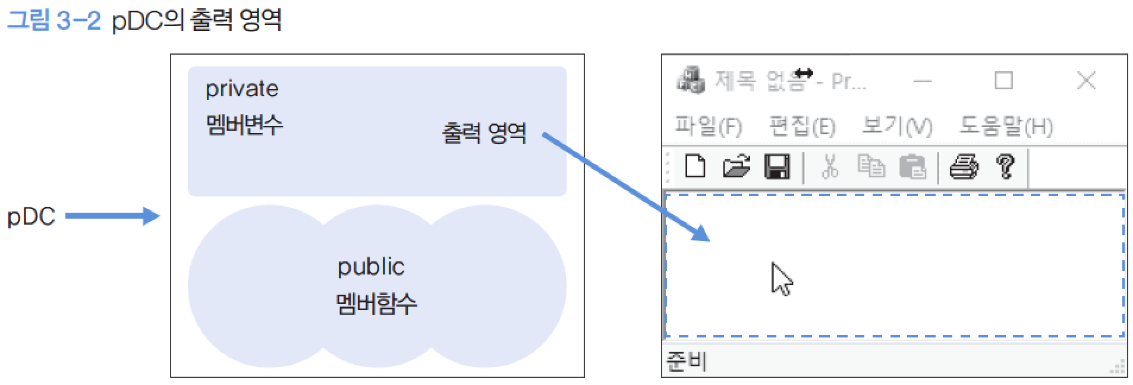
CPen \*pOldPen = dc.SelectObject(&pen); // CBrush \*pOldBrush = dc.SelectObject(&brush)

dc.Rectangle(100, 100, 200, 200); // 직사각형을 그린다.

dc.SelectObject(pOldPen); // SelectObject(pOldBrush)

DC



무효화 영역

void CMainFrame::OnPaint()

{

/\* 생성자에서 ::**BeginPaint**() 호출! \*/

**CPaintDC** dc(this);

TCHAR \*msg = \_T("Hello, MFC");

dc.TextOut(100, 100, msg, lstrlen(msg));

} /\* 소멸자에서 ::**EndPaint**() 호출! \*/

* 무효 영역 생성 함수

void CWnd::Invalidate(BOOL bErase = TRUE);

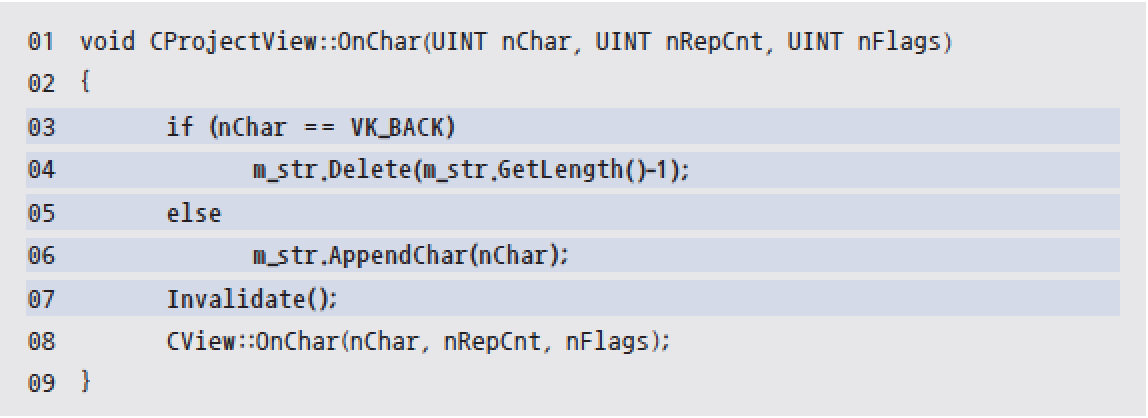
void CWnd::InvalidateRect(LPCRECT lpRect, BOOL bErase = TRUE);

void CWnd::InvalidateRgn(CRgn\* pRgn, BOOL bErase = TRUE);

Invalidate() <-> UpdateAllViews(NULL);

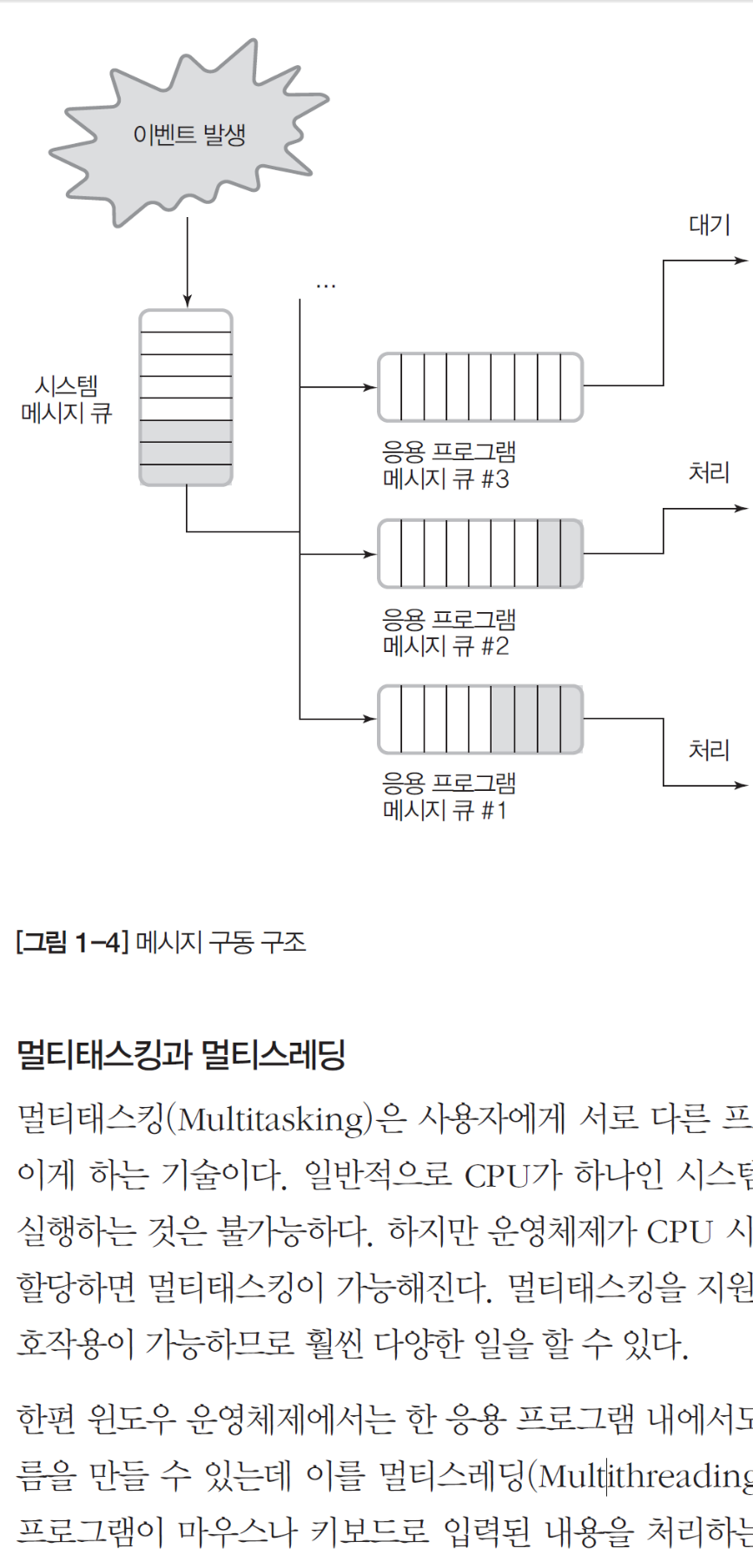
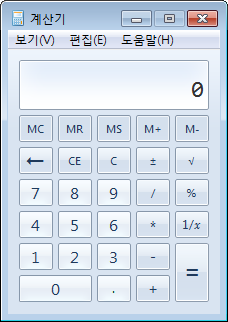
키보드

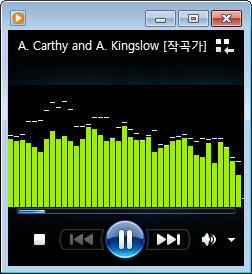
CProjectView::onDraw() : pDC->TextOut(0 ,0 ,m\_str, m\_str.GetLength());



|  |  |
| --- | --- |
| 텍스트 함수 | **기능** |
| TextOut() | (x, y) 위치에 문자열을 출력한다. |
| DrawText() | 직사각형 영역 내부에 문자열을 출력한다. |
| SetTextColor() | 글자의 색을 바꾼다. |
| SetBkColor() | 글자의 배경색을 바꾼다. |
| SetTextAlign() | 기준 위치에 대한 문자열 정렬 방식을 설정한다. |

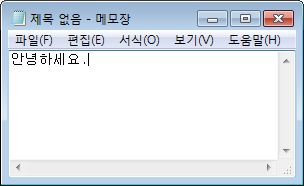
메시지 구동구조





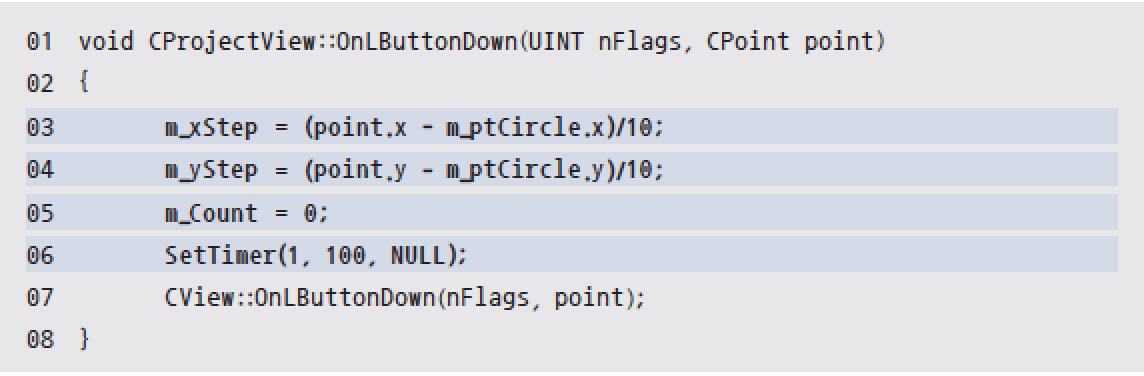
이벤트 (예: LButtonDown)

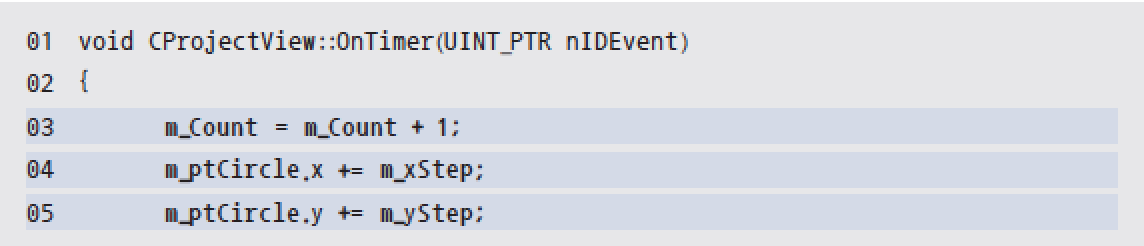
타이머

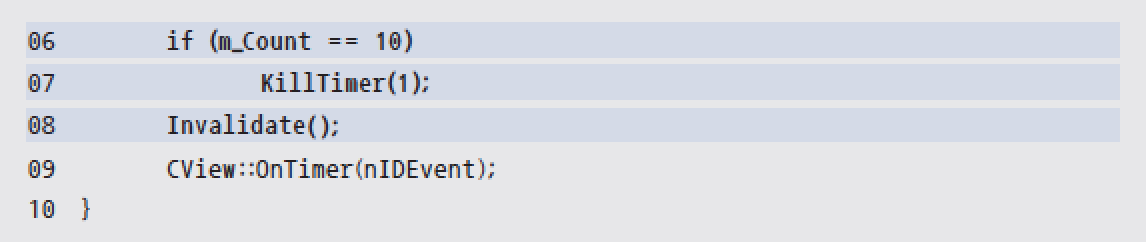
비트맵 (BitBlt, StretchBlt)

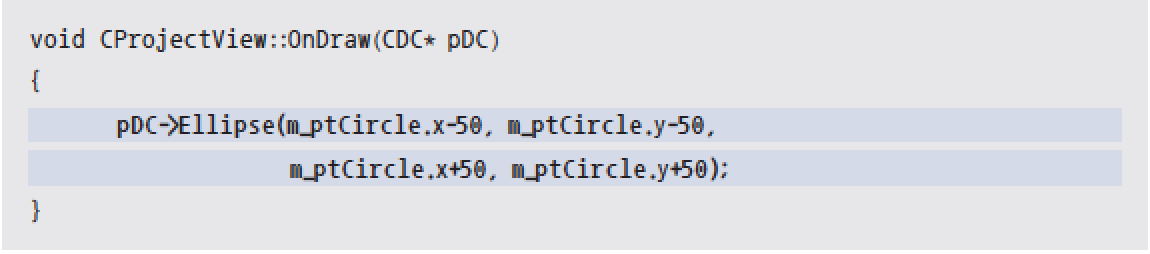
멀티테스킹 멀티 스레딩

* + 멀티태스킹(Multitasking)
    - 운영체제가 여러 개의 응용 프로그램을 동시에 실행
  + 멀티스레딩(Multithreading)
    - 응용 프로그램 내부에서 여러 개의 실행 흐름(=스레드)을 동시에 진행

타이머







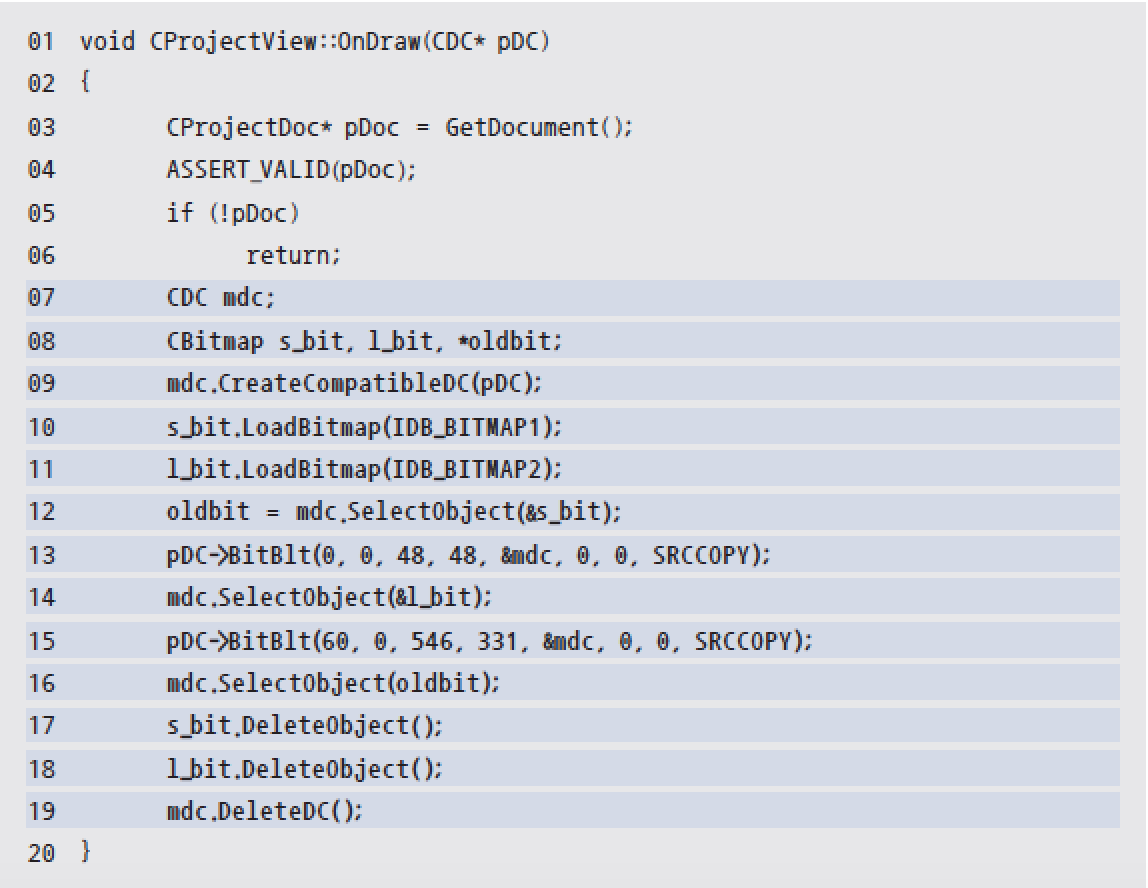
타이머 정지 : KillTimer(nIDEvent);

비트맵

출력과정 : 비트맵 파일 가져오기 -> 파일편집(id부여) -> 비트맵로드(LoadBitmap(id);) ->

사용할 비트맵선택(SelectObject(&s\_bit);) -> 비트맵 전송(BitBit()) ->

Dc(CDC) <-> memdc(CDC)



* pDC->StretchBlt(100, 100, 200, 300, &mdc, 100, 100, 500, 500, SRCCOPY);
* SRCAND : AND 연산자를 사용하여 대상 비트 맵과 소스 비트 맵의 픽셀을 결합  
  SRCCOPY : 소스 비트 맵을 대상 비트 맵에 복사.  
  SRCPAINT : OR 연산자를 사용하여 대상 비트 맵과 소스 비트 맵의 픽셀을 결합