

CHAPTER

16

품



1 윈도우 프로그래밍

1.1 Win32와의 관계

별도의 구분은 없지만 이 장부터 2부이며 다루는 주제와 프로젝트 형태가 달라진다. 1부에서는 C# 문법 학습을 위해 콘솔에서 예제를 만들었는데 2부에서부터는 칙칙하고 썰렁한 콘솔 환경을 벗어나 화려한 그래픽 환경에서 실습을 진행할 것이다. 그래픽 환경은 표현력이 풍부하고 멀티태스킹 과도 훨씬 잘 어울린다. 사실 콘솔은 디버깅용과 교육용 이상의 실용성이 없는 구닥다리 환경임이 분명하다. 최신 개발 툴인 닷넷이 아직까지도 콘솔을 지원하는 주된 이유는 초보자들이 조금이라도 쉽게 C#을 배울 수 있도록 하기 위한 배려이다. 문법을 배울 때는 콘솔같이 단순한 환경이 제일 이상적이다.

윈도우즈용 응용 프로그램을 작성하는 데는 여러 가지 방법들이 존재한다. 가장 원론적인 방법은 운영체제가 제공하는 Win32 API를 사용하는 것인데 섬세하고 강력하기는 하지만 일일이 손으로 코드를 작성해야 하므로 생산성이 떨어진다. Win32보다는 좀 더 객체 지향적이고 편리한 MFC 라이브러리를 활용하는 것이 더 일반적이며 그 외에 성능보다 빠른 개발을 위해 비주얼 베이직이나 델파이 같은 비주얼 툴도 많이 활용된다. 닷넷에서는 윈폼(WinForm) 라이브러리로 그래픽 응용 프로그램을 작성하며 최근에 WPF라는 좀 더 화려한 고급 라이브러리가 발표되었다.

윈폼은 C#이라는 강력하고도 깔끔한 언어와 BCL이라는 방대한 라이브러리, 그리고 비주얼 스튜디오라는 편리한 개발 환경을 활용하여 멋지고도 안정적인 윈도우즈용 응용 프로그램을 신속하게 작성하는 방법이다. 윈폼은 닷넷이 제공하는 완전히 새로운 개발 방법이지만 그 하부 구조는 역시 Win32이다. 물론 플랫폼 독립적인 닷넷이 Win32를 직접 호출하지는 않지만 많은 영향을 받았음은 사실이다. 윈폼은 Win32의 여러 가지 이론이나 구조를 거의 그대로 차용하고 있는데 몇 가지 예를 들어 보자.