

객체란

컴퓨터공학과 2016003854 이효원

나의 자동차를 만든다고 가정해보자. 효원카를 만들기 위한 틀이 최우선적으로 필요할 것이다. 이 경우 하나의 전체적인 틀을 만드는 것과, 부품부품별로 틀을 만들어 결합하는 것, 둘 중 어느 경우가 더 바람직하다고 생각하는가? 아무래도 나는 후자라고 생각한다. 전자의 경우 해당 틀로 효원카만 생산할 수 있으나 후자의 경우에는 해당 부품의 틀이 필요할 때 마다 얼마든지 다른 모델의 차량에도 재사용할 수 있기 때문이다.

코딩에도 이와 비슷한 개념이 존재한다. 전자는 절차지향, 후자는 객체지향적이라고 부른다. 객체지향언어의 대표적인 예시로는 자바가 있다. 위에서 아래로 쪽 훑고 실행시키는 절차지향적 언어와는 달리 객체지향적 언어는 메인 함수를 가장 먼저 실행시킨 뒤에 각 함수에서 호출하는 클래스들을 차례대로 실행시킨다.

객체지향언어의 가장 큰 특징은 코드의 재사용성이 높다는 것이다. 기본 강의실 설계도를 하나의 클래스로 작성해두었다고 가정하자. 이후에 프로그래밍설계방법론 강의를 설계해야 할 때에는 기본 강의실 설계도를 상속받아 기본 틀을 작성한 뒤 필요한 몇몇 세부사항만 수정하면 된다. 이후에 시스템프로그래밍 기초 강의실 등, 비슷한 성격의 설계도가 필요할 때에는 기존의 설계도를 상속받음으로써 번거로움을 덜 수 있다.

객체지향언어에서 사용하는 클래스를 대강 설명해보았다. 이 모든 것은 객체를 설명하기 위함이다. IT 용어사전에 따르면, 객체란 '객체지향프로그래밍이나 설계에서 데이터(실체)와 그 데이터에 관련되는 동작(절차, 방법, 기능)을 모두 포함한 기능'이라고 한다. 쉽게 설명하자면 위에서 언급한 클래스를 실행하면 설계도에서 설정해 둔 옵션에 따라 코드가 돌아가고, 이에 따른 결과물이 생성된다. 이 결과물이 객체이다. '붕어빵 틀 : 붕어빵 = 클래스 : 객체' 라고 이해해도 무방하다.

우선 본인이 아는 객체는 이 정도이다. 그 속성에 대해 안다기보다는 막연히 이런 것을 객체라고 부르나보다 정도의 지식 수준이기에 정말 초보입문자를 위해 간결하게 작성해보았다. 앞으로 더 많이 배워서 객체를 진정으로 이해하게 된다면 그 때 다시 돌아오겠다!