



코딩 교육 메뉴얼 (3)

코딩 교육 메뉴얼은 현재 객체지향 프로그래밍을 수강하고 있는 학생들에게 프로그래밍 공부를 해매지 않고 지속적으로 공부할 수 있을 지 고민하여, 주기적으로 코딩 공부에 대한 가이드라인이나 간략한 노하우를 알려주기 위해 작성되고 있습니다.

객체지향 패러다임은 지식을 **추상화**하고 추상화한 지식을 객체 안에 **캡슐화**함으로써 실세계 문제에 내재된 복잡성을 관리하려고 한다. 객체를 발견하고 창조하는 것은 **지식과 행동을 구조화**하는 문제다.

- 레베카 워프스브록(Rebecca J. Wirfs-Brock)

객체의 추상화

컴퓨터 공학을 공부하다 보면 '추상화'라는 단어를 굉장히 자주 접하게 됩니다.

흔히 추상적이라는 의미를 떠올리게 되어 추상화하였다는 표현이 어떤 것을 구체적이지 않고 막연하고 두루뭉실하게 만드는 것이라는 생각이 들기 마련입니다.

그러나 컴퓨터 공학에서 '추상화'라는 표현은 아래와 같은 의미로 정의할 수 있습니다.

추상화 : 어떤 양상, 세부 사항, 구조를 좀 더 명확하게 이해하기 위해 특정 절차나 물체를 의도적으로 생략하거나 감춤으로써 복잡도를 극복하는 방법.

다시 말해, 객체지향 패러다임에서 추상화는 객체 간의 복잡성을 다루기 위한 하나의 방법인 것입니다.

이러한 추상화는 두 가지 차원으로 이루어진다고 할 수 있습니다.

1. 구체적인 사물들(개별 객체) 간의 공통점은 취하고 차이점은 버리는 일반화를 통해 단순화
개별 객체 간의 차이점은 무시하고 공통점을 취한 결과
2. 중요한 부분을 강조하기 위해 불필요한 세부 사항을 제거함으로써 단순화
(객체들의 공통점 중에서 중요한 특징 외의 불필요한 사항들은 전적으로 무시하여 추상화)

이 두 경우를 구분하고 외우는 것은 큰 의미가 없습니다. 중요한 것은 객체 추상화의 목적이 **복잡성을 이해하기 쉬운 수준으로 단순화하는 것**에 있다는 것입니다.

이쯤에서 객체지향 프로그래밍이 가지는 오해 중 하나를 짚고 넘어가려고 합니다.

객체지향 프로그래밍이 가지는 굉장히 흔한 오해 중 하나는 **‘객체지향이 현실세계에 대한 모방’**이라고 생각한다는 것입니다.

소프트웨어 객체를 창조할 때 우리는 결코 현실 세계의 객체를 모방하지 않습니다.

오히려 소프트웨어 안에 창조하는 객체에게 현실 세계의 객체와는 전혀 다른 특징을 부여하는 것이 일반적이죠. 소프트웨어 객체는 현실 객체의 부분적인 특징을 모방하는 것이 아니라 현실 객체가 가지지 못한 추가적인 능력을 보유할 수 있게 됩니다.

그렇다면 객체지향의 세계와 현실 세계 사이에는 전혀 상관이 없는 것일까요?

그렇지는 않습니다. 다만 모방이나 추상화의 수준이 아닌 다른 관점에서 유사성을 가지고 있습니다. 현실 세계와 객체 지향 세계 사이의 관계를 좀 더 정확하게 설명할 수 있는 단어는 **은유(metaphor)** 이다.

은유에 대한 자세한 설명은 다음 코딩 교육 메뉴얼에 이어서 설명하도록 하겠습니다.