

클래스와 객체

객체지향프로그램 특징 이해하기



객체 지향의 개요

■ 절차 지향 프로그래밍

- 일련의 동작을 순서에 맞추어 단계적으로 실행하도록 명령어를 나열
- 데이터를 정의하는 방법보다는 **명령어의 순서와 흐름**에 중점
- 수행할 작업을 예상할 수 있어 **직관적인데, 규모가 작을 때는** 프로그래밍과 이해하기가 용이
- 소프트웨어는 계산 위주이므로 절차 지향 프로그래밍이 적합

■ 객체 지향 프로그래밍

- 소프트웨어의 규모가 커지면서 동작과 분리되어 전 과정에서 서로 복잡하게 얽혀 있는 데이터를 사용했기 때문에 절차 지향 프로그래밍 방식의 한계
- 절차 지향 프로그램은 추후 **변경하거나 확장하기도 어려움**
- 현실 세계를 **객체 단위로 프로그래밍하며**, 객체는 필드(데이터)와 메서드(코드)를 하나로 묶어 표현



객체 지향의 개요

■ 객체와 클래스

설계도와 설계도로 구현한 대상



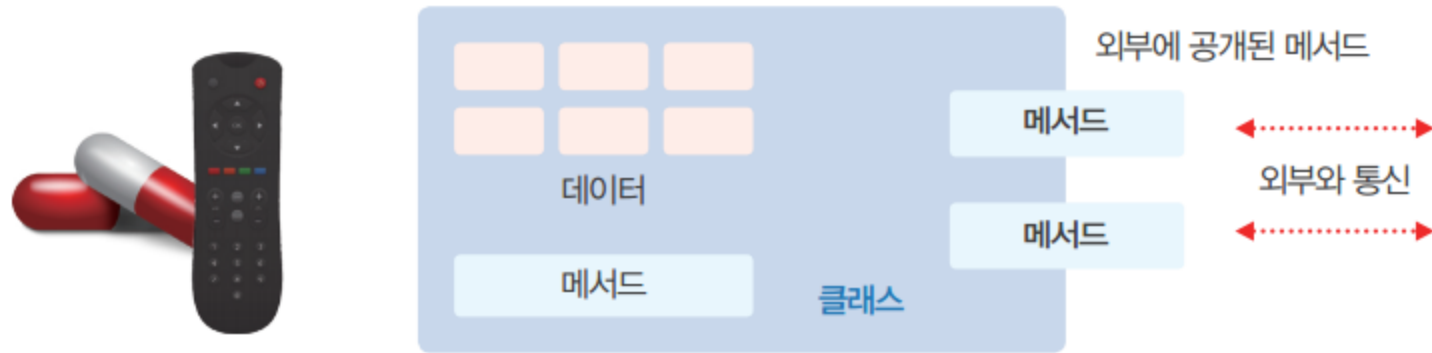
우리는 붕어빵(인스턴스)를 사서 먹음, 형틀은 인스턴스를 만들기위해 사용



객체 지향 프로그래밍

■ 캡슐화(정보 은닉)

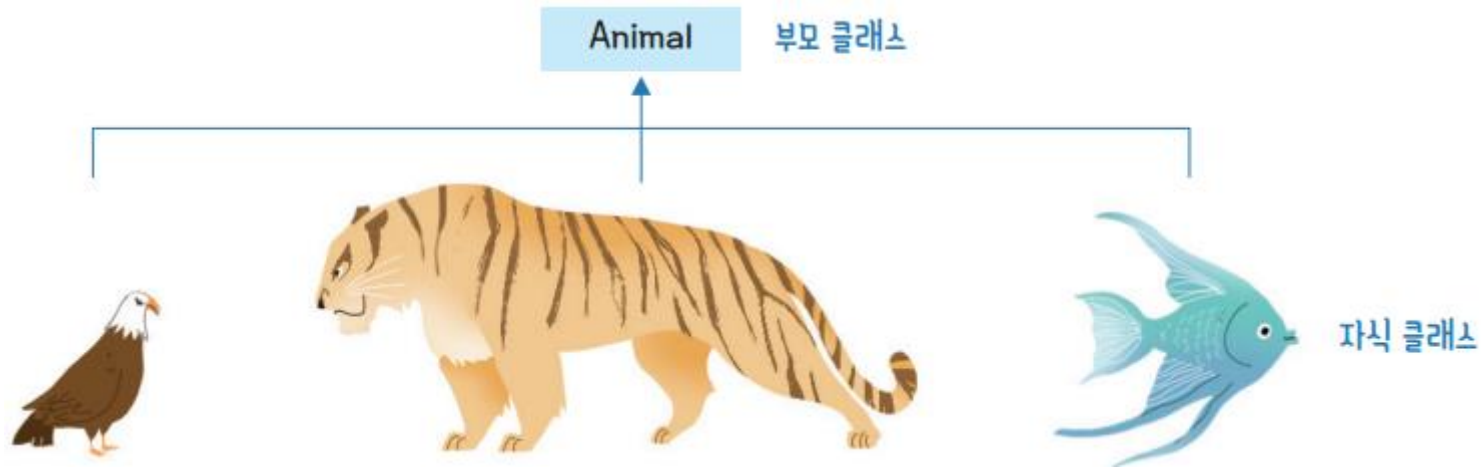
- 관련된 필드와 메서드를 하나의 캡슐처럼 포장해 세부 내용을 외부에서 알 수 없도록 감추는 것



객체 지향 프로그래밍

■ 상속

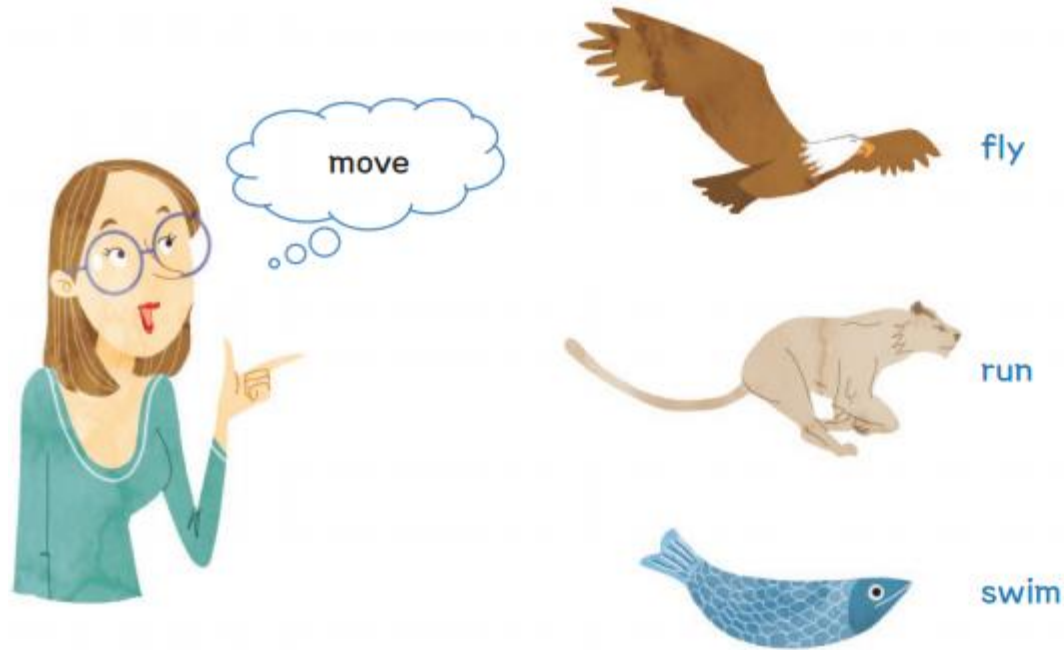
- 자녀가 부모 재산을 상속받아 사용하듯이 상위 객체를 상속받은 하위 객체가 상위 객체의 메서드와 필드를 사용하는 것
- 상속은 기존에 만들어진 클래스를 재사용하는 방법 중 하나



객체 지향 프로그래밍

■ 다형성

- 대입되는 객체에 따라서 메서드를 다르게 동작하도록 구현하는 기술
- 실행 도중 동일한 이름의 다양한 구현체 중에서 메서드를 선택 가능



다형성은 동일한 명령을 내리더라도
객체의 종류에 따라 다르게 실행되는
프로그래밍 기법이다.

