

객체지향의 사실과 오해

20181676 이동엽

개요

안녕하세요 자바 개발자입니다



“OOP아세요?”
그거 뭐.. 상속.. 뭐시기 아닌가요?



개발 공부를 하면서 “객체지향” 이라는 단어는 셀 수 없이 들어봤을 것이다.
객체지향의 개념에 대해 읽어봤을 뿐이라면, 제대로 이해하지 못 했을 수도 있다.
이 글은 객체지향에 대해 오해할 수 있는 내용들을 주제로 한다.

객체지향이란 뭘까?

객체지향 프로그래밍(OOP)

부품에 해당하는 객체들을 먼저 만들고, 이들을 하나씩 조립해 전체 프로그램을 완성하는 기법이다.

- 이것이 자바다 1장



객체지향이란 뭘까?

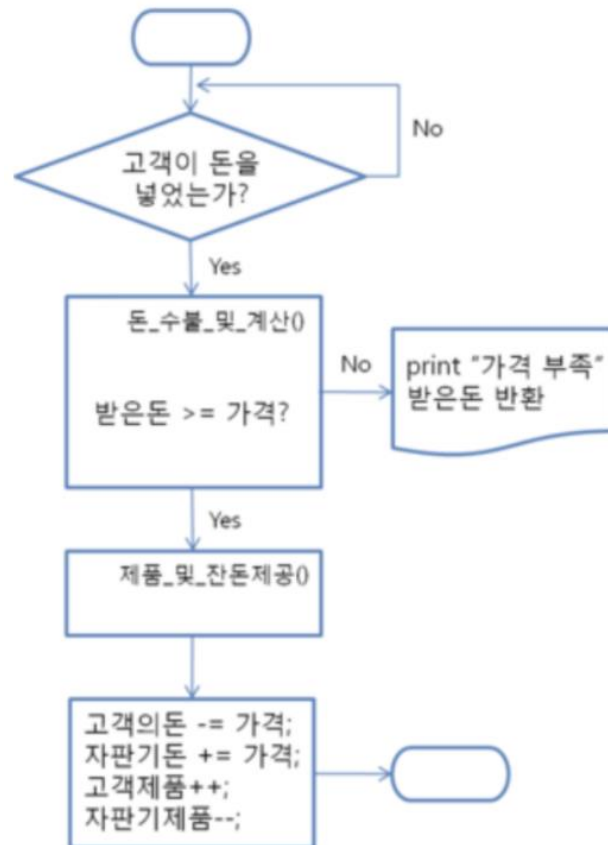
구현에 필요한 객체를 파악하고, 객체들의 역할을 정의하여 상호작용을 통해 프로그램을 만드는 것이다.

기능 중심인 절차지향의 "무엇을 어떤 절차로 할 것인가?" 와 다르게,

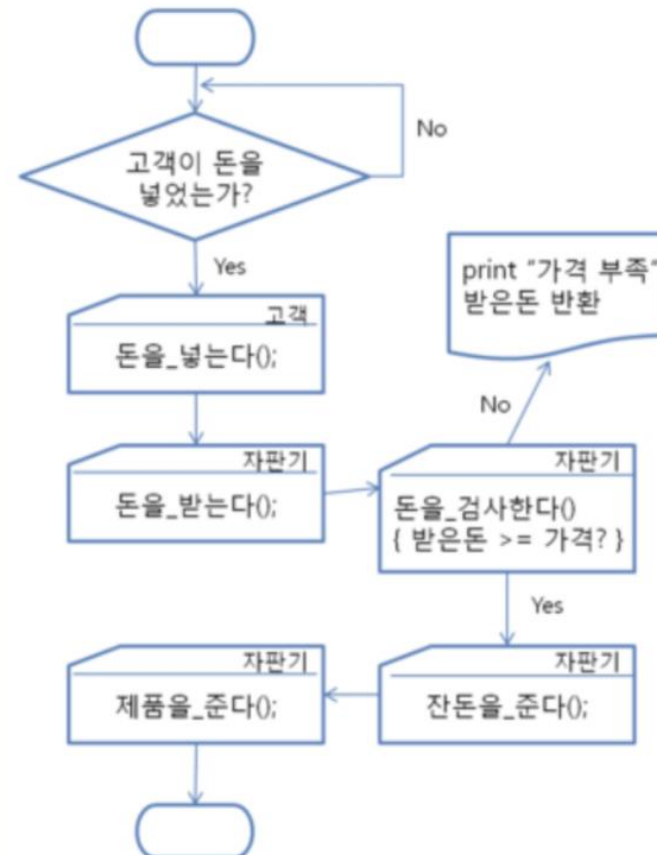
객체 중심인 객체지향은 "누가 어떤 일을 할 것인가?"가 핵심이다.

- 인코딩

절차지향 방식



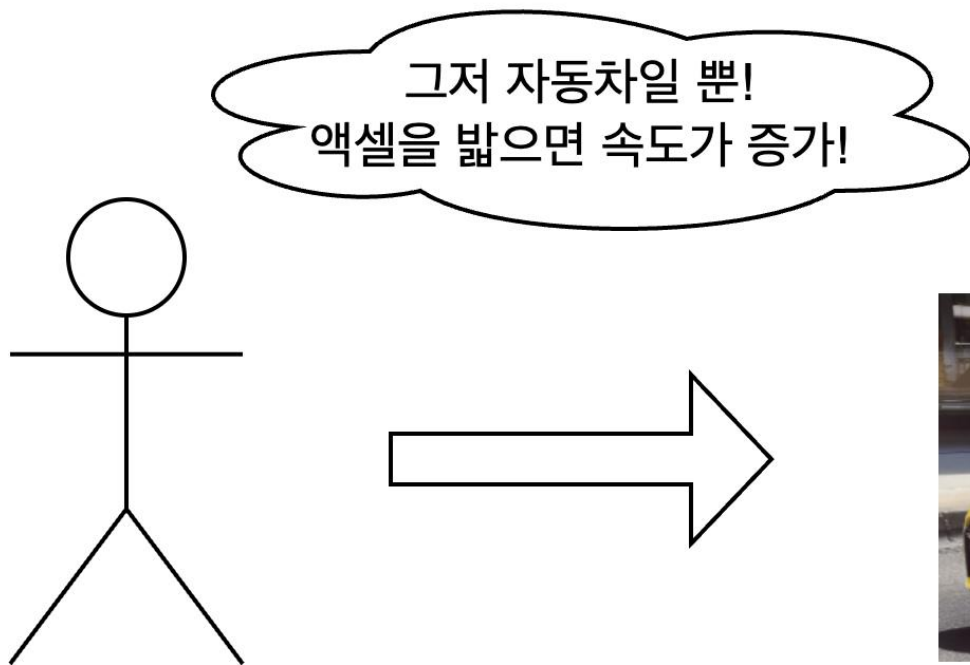
객체지향 방식



💡 객체지향은 현실세계를 일반화한 것이 아니다.

이론적인 접근 방식으로 설계를 한다면, 편협된 사고를 가질 수 있다.

그 중 대표적인 예시가 현실세계의 물체를 그대로 일반화시켜 프로그래밍하는 것이다.

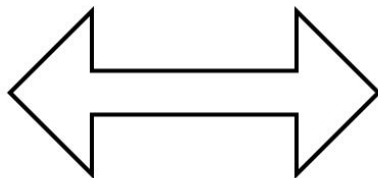
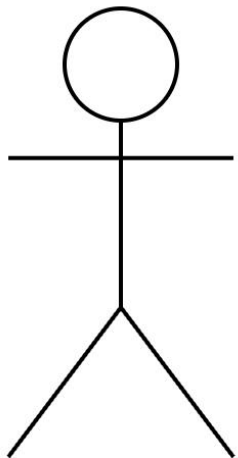


현실에서는 단방향적인 접근만 일어난다.

다만, 객체지향으로 설계할 때는 다른 시각을 가져야 한다.

💡 객체지향은 현실세계를 일반화한 것이 아니다.

엑셀 밟으면 속도 좀 올려줘



오케이 그럼 기름 좀 먹을게



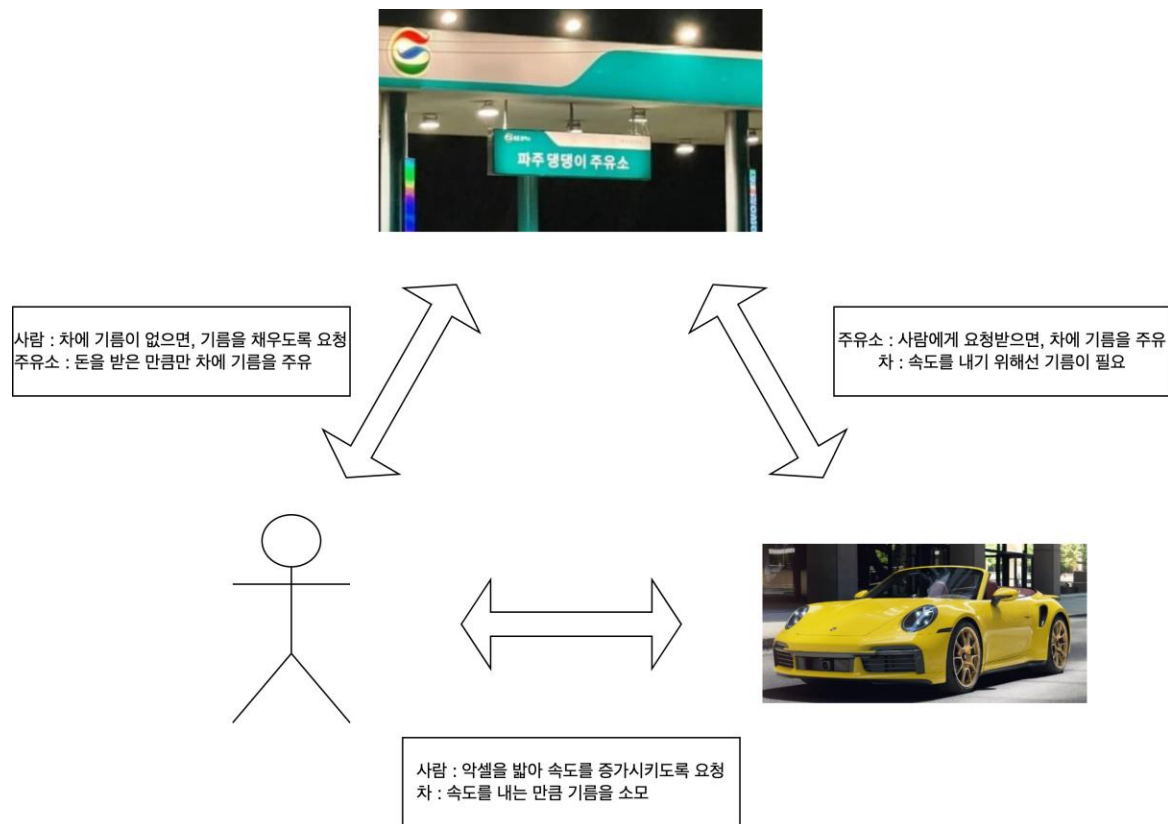
객체지향의 세계에선 사람도, 자동차도 **동일한 객체**로 다루어야 한다.

속도를 증가시킬지 말지는 자동차가 승인할 수도 있고, 반대로 요청을 거절할 수도 있다.

💡 책임 주도 설계 : 객체는 책임을 나눠갖도록 설계해야 한다.

책임 주도 설계란?

'어떤 책임이 어떤 객체에게 필요한가?'를 결정하는 과정이 전체 설계를 주도해야 한다는 것이다.



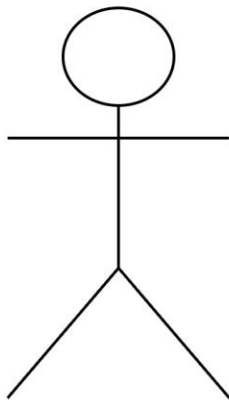
각 객체들에게 **적절한 책임을 부여**하여, 문제없이 **상호작용**을 하여 로직이 실행되도록 해야 한다.

💡 행동이 상태를 결정한다: 협력 관계를 잘 살펴보자

협력 관계가 정해지지 않았을 때 아래와 같은 선부른 설계는 옳바르지 않다.

- '자동차니까 speed 속성이 필요하겠지'
- '핸들은 당연히 왼쪽에 있지'

너 왜 핸들이 왼쪽에 있어?



우핸들에 익숙한 일본인

협력관계를 고려하지 않아 상호작용이 이루어지지 않는 경우

차도는 당연히 우측통행 아니야?



객체가 어떤 객체와 어떤 상호작용을 할 지 정해지고, 역할이 정해진 후에 그 내용을 토대로 상태, 필드, 속성 등을 결정하는 것이 올바른 객체지향 접근 방식이다.

감사합니다.