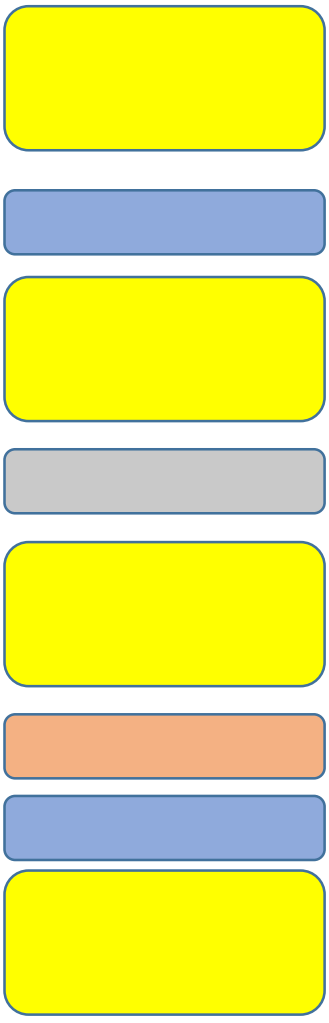
프로그래밍 언어의 종류

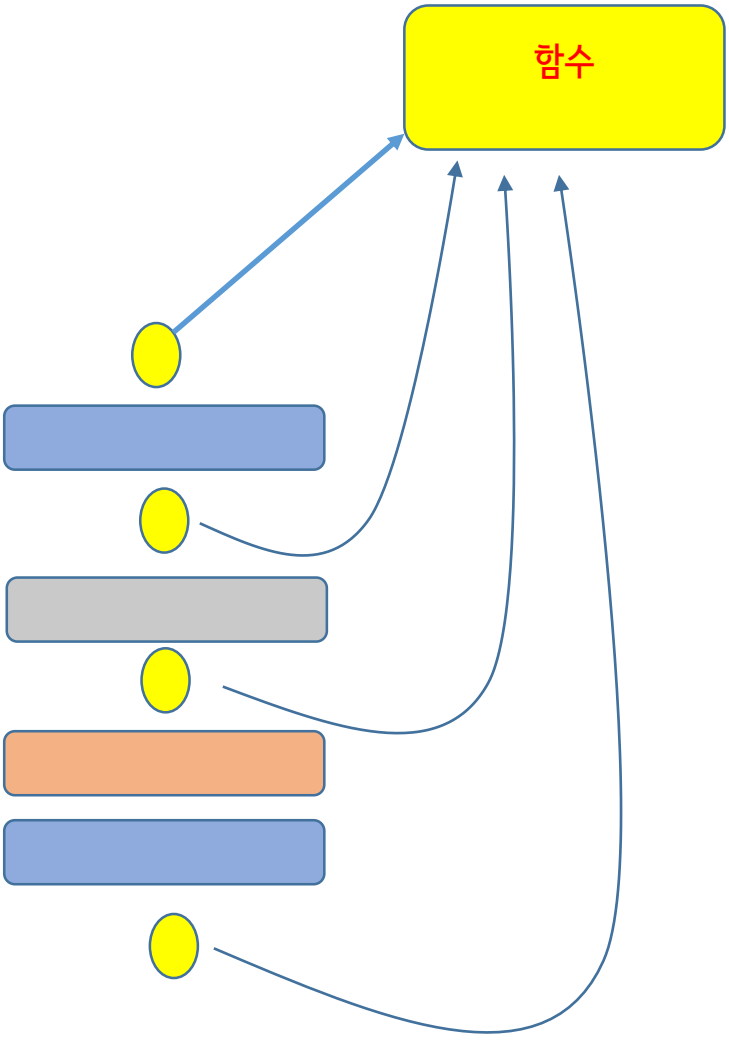
- 순차지향 프로그래밍 언어(Sequential Oriented Programming Language)
- 절차지향 프로그래밍 언어(Procedural Oriented Programming Language)
- 객체지향 프로그래밍 언어(Object Oriented Programming Language)

순차지향 언어



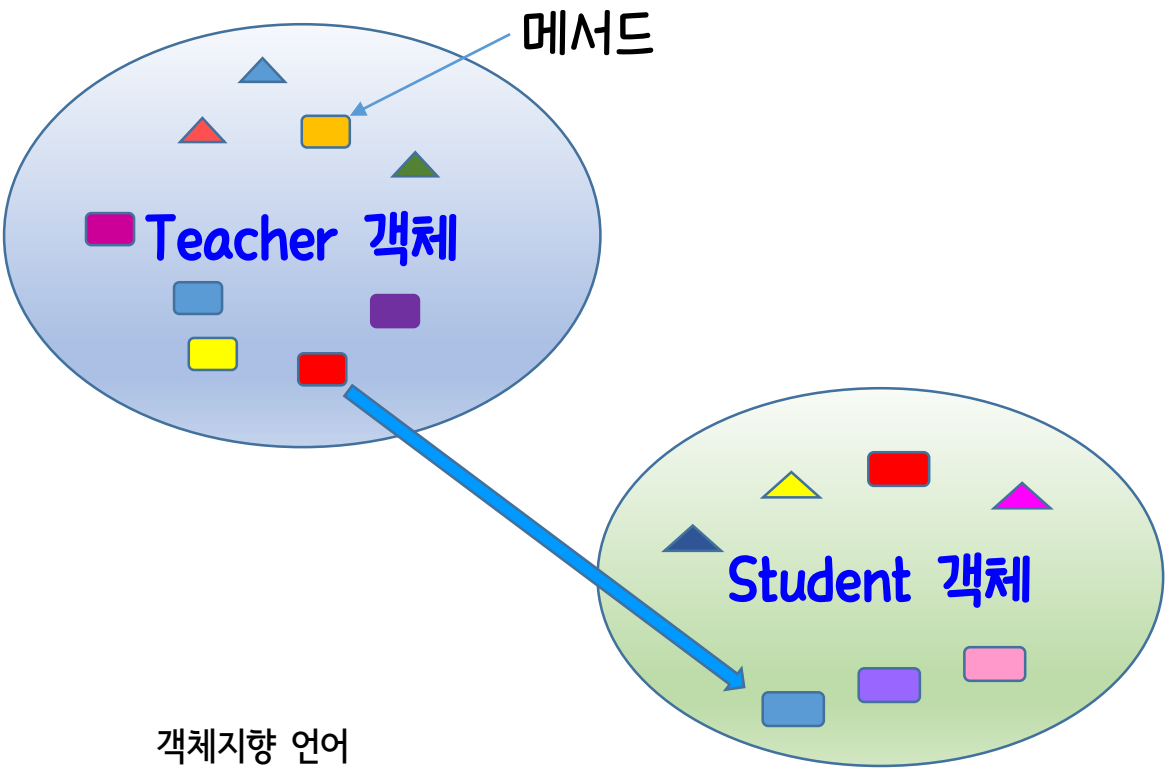
어셈블리, 베이직

절차지향 언어



C

절차 지향 언어



객체지향 언어

[객체지향 언어의 역사]

- 과학, 군사적 모의실험(simulation)을 위해 컴퓨터를 이용한 가상세계를 구현하려는 노력으로부터 객체지향이론이 시작됨

- 1960년대 최초의 객체지향언어 Simula탄생

- 1980년대 절차방식의 프로그래밍의 한계를 객체지향방식으로 극복하려고 노력함.
  (C++, Smalltalk과 같은 보다 발전된 객체지향언어가 탄생)
- 1995년 말 Java탄생. 객체지향언어가 프로그래밍 언어의 주류가 됨.

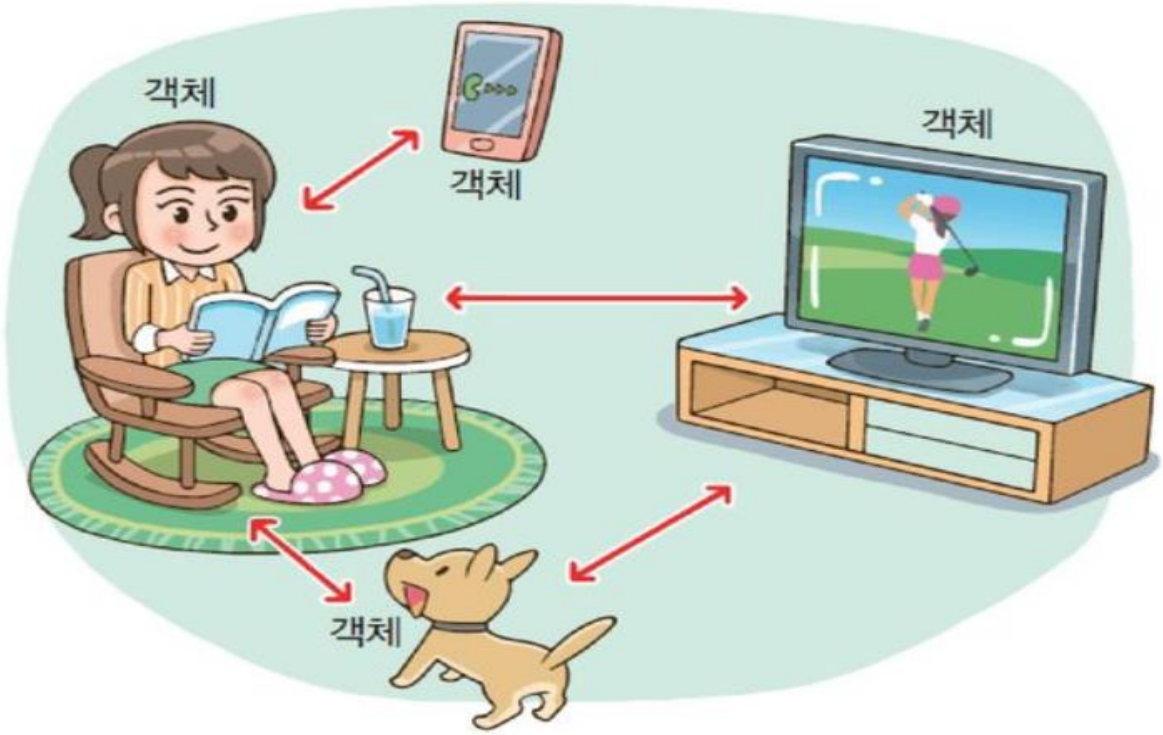
[객체지향 언어의 장점과 특징]

- 프로그램을 보다 유연하고 변경이 용이하게 만들 수 있다.

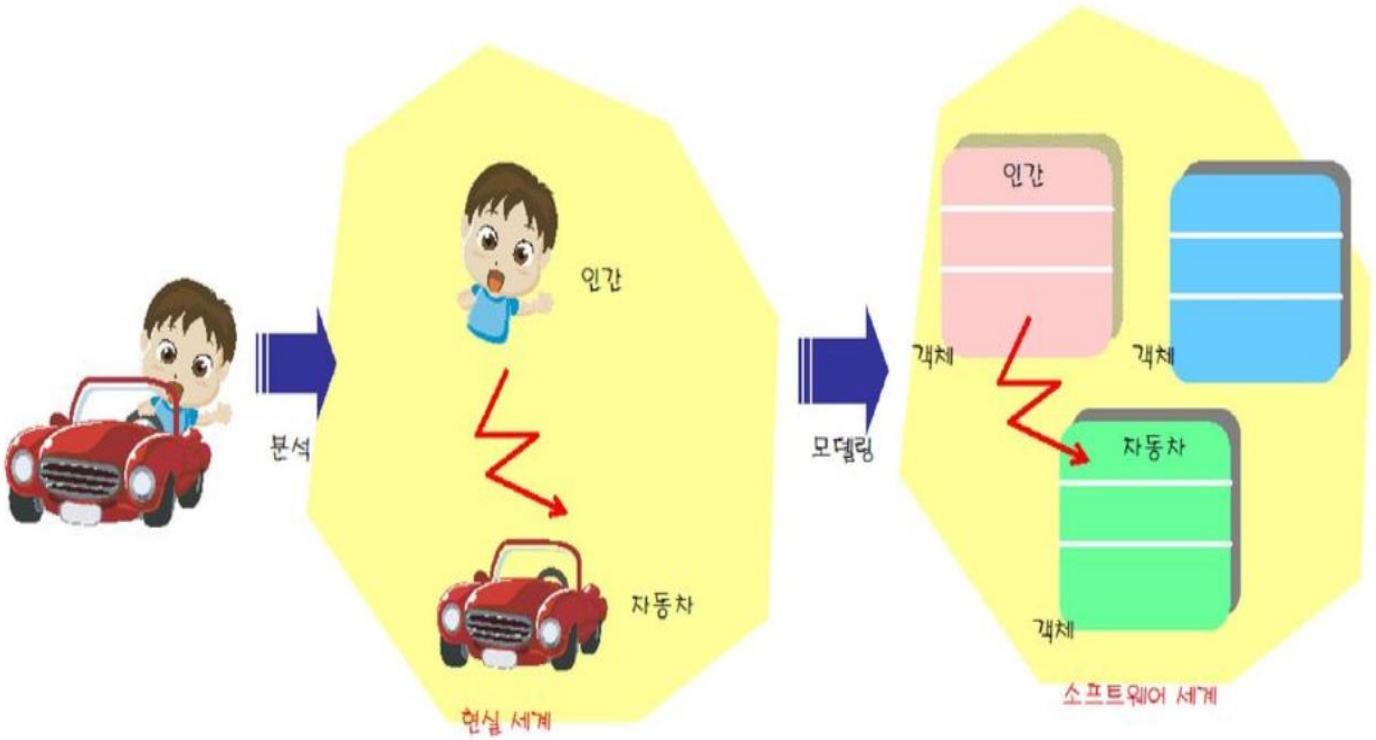
- 마치 컴퓨터 부품을 갈아 끼울 때, 해당하는 부품만 쉽게 교체하고 나머지 부품들을 건드리지 않아도 되는 것처럼 소프트웨어를 설계할 때 객체 지향적 원리를 잘 적용해 둔 프로그램은 각각의 부품들이 각자의 독립적인 역할을 가지기 때문에 코드의 변경을 최소화하고 유지보수를 하는 데 유리하다.

- 코드의 재사용을 통해 반복적인 코드를 최소화하고, 코드를 최대한 간결하게 표현할 수 있다.

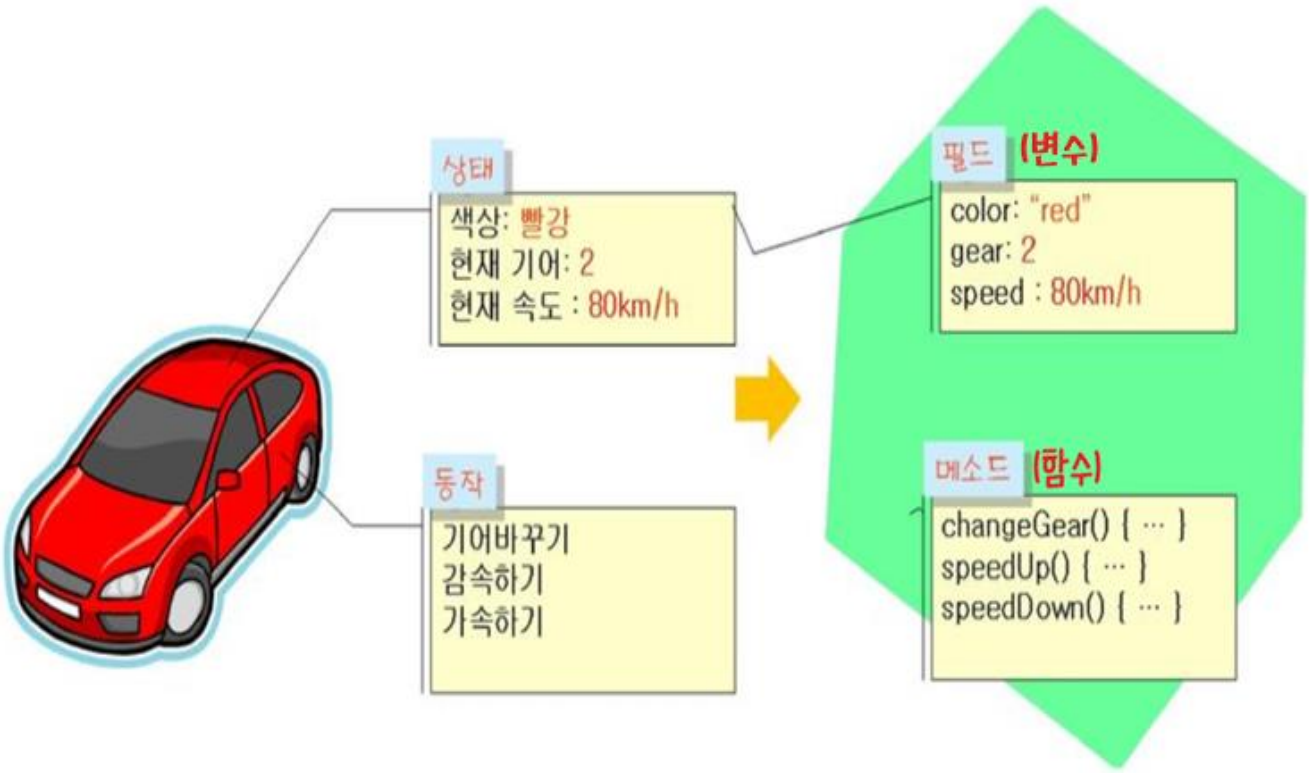
- 객체 지향 프로그래밍의 4가지 특징은 각각 추상화, 상속, 다형성, 캡슐화인데, 모두 이러한 객체지향적 설계의 이점들을 가장 잘 살릴 수 있는 방향으로 발전되어 왔다.



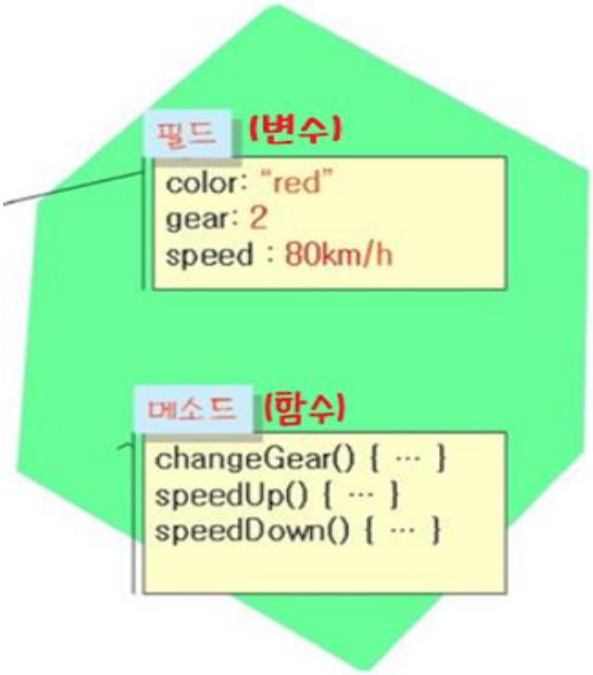
[객체의 관계를 프로그램으로 설계]



[객체의 구성 요소 - 상태(속성)와 동작]



[객체를 만들기 위한 프로그램 구조 - 클래스]



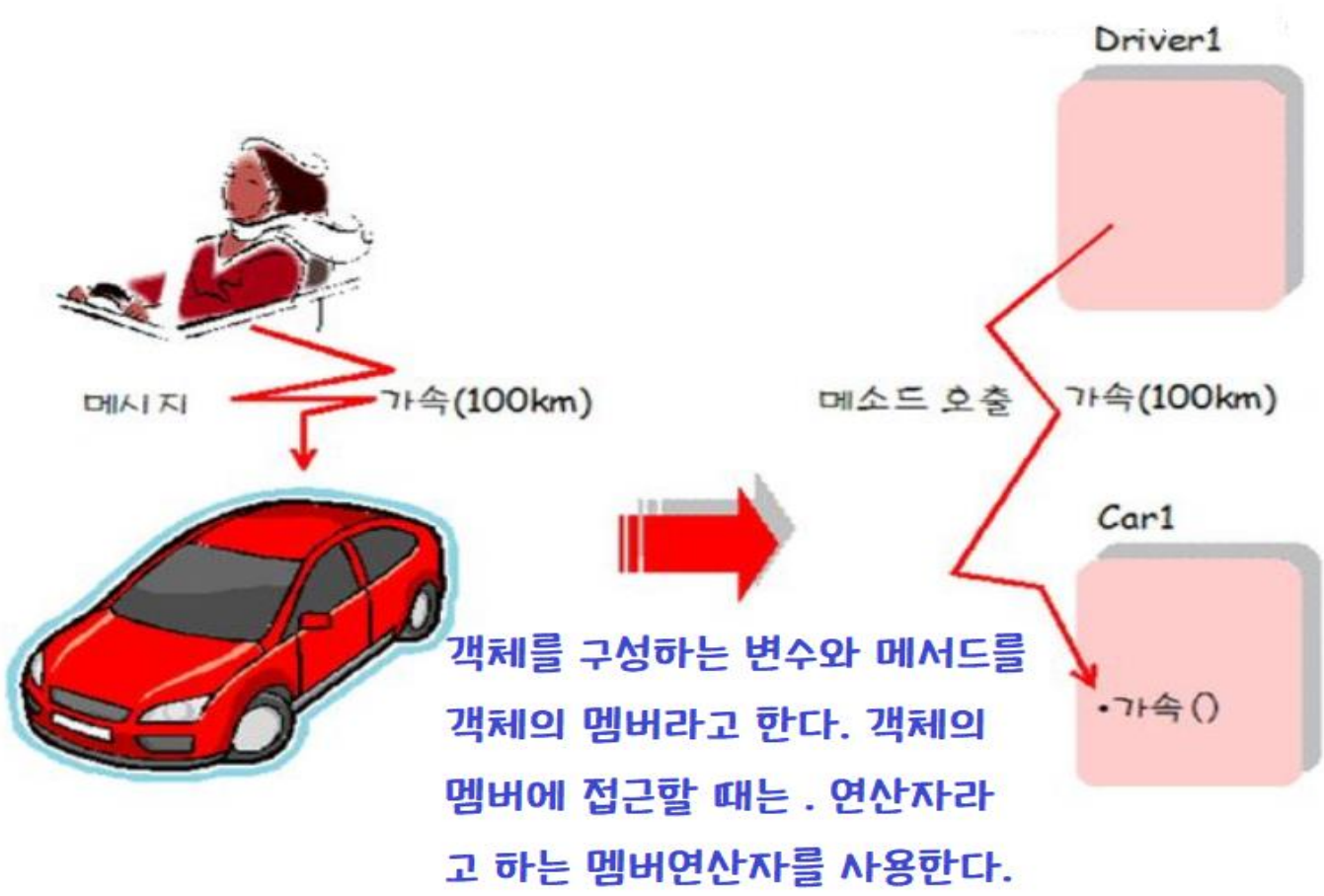

```
class Car {
    String color;
    int gear;
    double speed;

    void changeGear() {
        :
    }
    void speedUp() {
        :
    }
    void speedDown() {
        :
    }
}
```

```
Car obj = new Dar();
```

객체(Object) = 속성(Attribute) + 기능(Method)

= 변수(Field) + 함수(Function)

객체에 속한 함수를 메서드라고 한다.



객체 또는 객체를 담고 있는 변수 . 멤버

[OOP 구문에서 공부할 내용]

- 클래스 생성 방법(멤버변수, 메서드, 생성자 메서드)
- 객체 생성하는 방법과 사용 방법
- static 멤버변수와 non-static 멤버변수
- static 메서드와 non-static 메서드
- 객체 초기화
- 메서드 오버로딩
- 상속
- 메서드 오버라이딩
- 추상메서드와 추상 클래스, 인터페이스
- 추상화
- 다형성
- 캡슐화