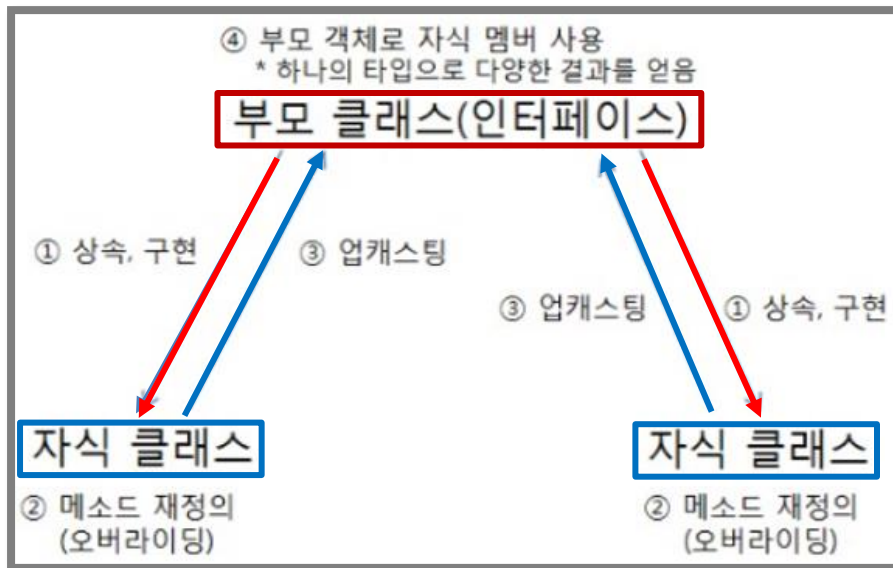
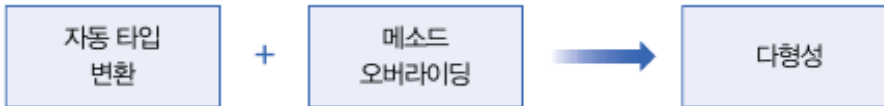


다형성

## ■ 다형성

- 다형성이란 사용 방법은 동일하지만, 실행 결과가 다양하게 나오는 성질을 말한다.



## ● 필드 다형성

- 필드 타입은 동일하지만(사용 방법은 동일하지만),  
대입되는 객체가 달라져서 실행 결과가 다양하게 나올 수 있는 것을 말한다.

```
public class Car {  
    //필드 선언  
    public Tire tire;  
  
    //메소드 선언  
    public void run() {  
        tire.roll();  
    }  
}
```

tire 필드에 대입된 객체의 roll() 메소드 호출

```
//Car 객체 생성  
Car myCar = new Car();  
//HankookTire 장착  
myCar.tire = new HankookTire();  
//KumhoTire 장착  
myCar.tire = new KumhoTire();
```

```
myCar.run();
```

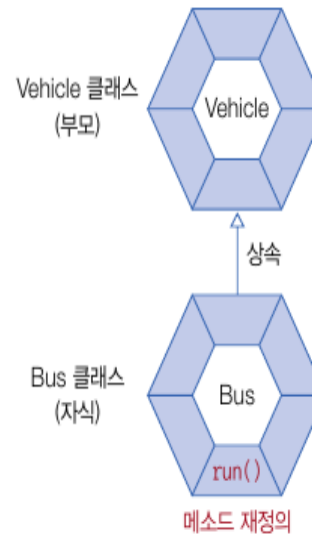
대입된(장착된) 타이어의 roll() 메소드 호출

## ● 매개변수 다형성

- 메소드가 클래스 타입의 매개변수를 가지고 있을 경우, 호출할 때 동일한 타입의 객체를 제공하는 것이 정석이지만 자식 객체를 제공할 수도 있다. 여기서 다형성이 발생한다.

```
public class Driver {  
    public void drive(Vehicle vehicle) {  
        vehicle.run();  
    }  
}
```

```
Driver driver = new Driver();  
Vehicle vehicle = new Vehicle();  
driver.drive(vehicle);
```



```
Driver driver = new Driver();  
Bus bus = new Bus();  
driver.drive( bus ); •.....Bus 객체의 run() 호출
```

자동 타입 변환 발생  
Vehicle vehicle = bus;

