

Inheritance(extends) I - 상속은 재활용 + α

상속은 재활용(10%) + α(90%)

- Java는 객체지향언어(Object-Oriented Language)이고, 대표적인 두가지는 Class와 Inheritance(상속)이다.
- 상속이란 무엇일까? **재활용을 위한 문법이 아니다.**
뜻은 상속 Part가 끝날 때 까지 공부하며 차근차근 이해를 해야한다.

상속의 관한 이야기

- i. 객체지향 패러다임의 처음은 재활용의 관점에 출발했으나, 실패한 패러다임
- ii. Java의 상속을 Class의 재활용적 측면에서 바라보는 것은 문제가 있다.
- iii. 재활용 할 수 있다는 이유만으로 상속을 하는 것은 드물다.

상속을 하는 이유?

- 상속을 통해 연관된 일련의 Class들에 대한 공통적인 **규약**을 정의하고, 적용하는데 상속의 실질적인 목적이 있다.
이 뜻도 상속 Part가 끝날 때 까지 공부하며 차근차근 이해를 해야한다.

상속은 재활용이다! 라는 공식을 깨는 것부터 시작해야 상속을 정확히 이해할 수 있다.

상속의 중요한 키워드들은 extends, overriding, 다형성, abstract and interface이다!

상속의 기본문법 이해

- **문법적 측면의 상속은 기존에 정의된 Class에 Method와 Variable을 추가하여 새로운 Class를 정의하는 것!**

// 상위, 부모 Class

```
class Man
{
    public static void main(String[] args){
        public String name;

        public void tellYourName(){ System.out.println("My Name is "+name); }
    }
}
```

// 하위, 자식 Class

```
class BusinessMan extends Man // Man이라는 class를 상속하겠다. Man의 모든 Member들을 상속받겠다.
{
    public String name;
    public String company;
    public String position;

    public void tellYourName(){
        System.out.println("My Name is "+name);
    }

    public void tellYourInfo() {
        System.out.println("My company is "+company);
        System.out.println("My position is "+position);
        tellYourName(); // Man을 상속 받았기 때문에 Man의 Member가 존재하게 되고 호출이 가능하다.
    }
}
```