

회사에 출근해서 업무를 시작하기 전에 제일 먼저 하는 일이 아마 책상 정리일 것이다. 커피를 내리는 경우도 있을 것이고, 컴퓨터 부팅을 시작하기도 할 것이다. 즉 어떤 일을 시작하기 전에 준비를 하게 되는데 이것을 다른 말로 초기화라고 한다.

#### \* 개요

```
Calculator c1 = new Calculator();
c1.set0prands(10, 20);
c1.sum();
c1.avg();
```

위의 예에서 메소드 set0prands의 값으로 10과 20을 지정했다. 이 값들은 객체 내부에서 인스턴스 변수 left와 right의 값으로 설정되어서 유지된다. 그런데 이 객체를 이용하기 위해서는 기억해야 할 것이 있다. 아래와 같이 메소드 set0prands를 호출하기 전에 sum과 avg를 호출한다면 원하는 결과를 얻을 수 없을 것이다.

```
Calculator c1 = new Calculator();
c1.sum();
c1.avg();
```

이것은 객체 Calculator를 사용하기 위해서 사용자는 메소드 sum을 호출하기 전에 set0prands를 호출해야 한다는 것을 기억하고 있어야 한다는 것을 의미한다. 이러한 절차를 기억해야 한다는 것은 사용자 입장에서는 불편할 뿐 아니라 잘못된 사용으로 오류가 발생할 확률을 높이는 결과를 초래 할 수 있다.

#### \* 생성자

그래서 사용하는 것이 생성자(Constructor)이다. 아래와 같이 인스턴스가 생성될 때 left, right의 값을 입력하도록 강제한다면 어떨까?

```
class Calculator {
    int left, right;

    public Calculator(int left, int right) {
        this.left = left;
        this.right = right;
    }

    public void sum() {
        System.out.println(this.left + this.right);
    }

    public void avg() {
        System.out.println((this.left + this.right) / 2);
    }
}

public class CalculatorDemo1 {
```

```
public static void main(String[] args) {  
  
    Calculator c1 = new Calculator(10, 20);  
    c1.sum();  
    c1.avg();  
  
    Calculator c2 = new Calculator(20, 40);  
    c2.sum();  
    c2.avg();  
}  
  
}
```

#### \* 생성자의 특징

##### 1) 값을 반환하지 않는다.

생성자는 인스턴스를 생성해주는 역할을 하는 특수한 메소드라고 할 수 있다. 그런데 반환 값이 있다면 엉뚱한 객체가 생성될 것이다. 따라서 반환 값을 필요로 하는 작업에서는 생성자를 사용하지 않는다. 반환 값이 없기 때문에 return도 사용하지 않고, 반환 값을 메소드 정의에 포함시키지도 않는다.

##### 2) 생성자의 이름은 클래스의 이름과 동일하다.

자바에서 클래스의 이름과 동일한 메소드는 생성자로 사용하기로 약속되어 있다.