

클래스 변수에 이어서 클래스 메소드도 살펴봅시다. 과연 클래스 메소드는 언제 사용할까요?

이 질문을 해보세요: 생성된 인스턴스가 하나도 없더라도 이 메소드를 호출하는 게 말이 되나요? 정답이 'yes'라면 클래스 메소드, 즉 **static** 메소드를 사용하면 됩니다!

## static 변수를 다룰 때

만약 인스턴스와 직접적인 관계가 없는 **static** 변수를 바꾸려면 인스턴스와 직접적인 관계가 없는 **static** 메소드를 쓰는 게 맞겠죠?

```
public class Counter {
    static int count;

    public static void increment() {
        count++;
    }

    public static void main(String[] args) {
        System.out.println(Counter.count);

        Counter.increment();
        System.out.println(Counter.count);

        Counter.increment();
        System.out.println(Counter.count);

        Counter.increment();
        System.out.println(Counter.count);
    }
}
```

0  
1  
2  
3

## main 메소드

사실 우리가 처음 접한 **static** 메소드는 바로 **main** 메소드입니다.

```
public static void main(String[] args) {
    ...
}
```

**main** 은 프로그램의 시작점이기 때문에 아무 인스턴스도 생성되어 있지 않습니다. 따라서 **static** 인 게 말이 되죠?

## 관련 메소드 묶음

지금까지는 클래스를 정의하고, 그 클래스를 기반으로 한 인스턴스를 여러 개 만들어 사용했습니다. 그런데 인스턴스를 전혀 만들지 않고, 클래스에 서로 관련된 여러 가지 기능들을 **static** 메소드로 정리해서 사용하는 경우도 많습니다.

예를 들어서 수학 관련 메소드들을 모아둔 `Math` 클래스를 `import` 해서 봅시다.


```
import java.lang.Math;


public class Driver {
    public static void main(String[] args) {
        System.out.println(Math.abs(-10));    // 절댓값
        System.out.println(Math.max(3, 7));    // 두 값 중 최댓값
        System.out.println(Math.random());    // 0.0과 1.0 사이의 랜덤값
    }
}

10
7
0.40910432549890663
```

이 메소드들은 모두 인스턴스와 상관이 없는 **static** 메소드들이죠?

☒ 수업을 완료하셨으면 체크해주세요.

 수강생 Q&A 보기

 (/questions?) 질문하기

assignment\_id=353&sort\_by=popular)  
(/questions/new?

assignment\_id=353&op1=%EA%B0%9D%EC%B2%B4+%EC%A7

[이전 강의 클래스 변수 \(/assignments/455\)](/assignments/455)

다음 강의  
단위 변환기 > (/assignments/440)