

# Class Member - Class Method

## Class Method - Static Method

- 기본적인 특성은 Static Variable과 동일
- Instance와 아무 상관이 없는 Method
- 다른 Instance에 의해서 접근이 가능하다.
- Public으로 선언되면 같은 class가 아니더라도 외부의 class에서도 호출 가능

```
class NumberPrinter {
    public static void showInt(int n) {
        System.out.println(n);
    }

    public static void showDouble(double n) {
        System.out.println(n);
    }
}

class ClassMethod {
    NumberPrinter.showInt(20); // Class에 직접 접근

    NumberPrint np = new NumberPrint();
    np.showDouble(3.15); // Instance를 통한 접근
}
```

---

## Use Static Method - Case By Case

```
class SimpleMath {
    public static final double PI = 3.1415;
    public static double add(double n1, double n2) { return n1+n2 }
    public static double min(double n1, double n2) { return n1-n2 }
    public static double mul(double n1, double n2) { return n1*n2 }
}

class AreaMath {
    public static double calCircleArea(double rad) {
        double result = SimpleMath.mul(rad * rad);
        result = SimpleMath(result, SympleMath.PI);
        return result;
    }

    public static double calRectangleArea(double width, double height) {
        return SympleMath.mul(width, height);
    }
}
```

- 굳이 Instance의 생성이 필요하지 않고, 누구나 메소드 중심으로 쉽게 가져다 쓸 수 있도록 한다.
  - Static Method들로 구성된 Class는 Static Method들을 묶기 위해 별도로 정의
  - Instance화 할 필요없는 Method는 Instance Variable이 포함되어 있지 않다.
- 

## Static Method의 Instance Member Access?

- Static Method는 Instance에 속하지 않기 때문에 Instance Member에 접근불가!

```
class AAA {  
  
    int num1;  
    static int num2;  
  
    static void changeNum() {  
        num1++; // instance Member 접근 불가!
```

num1은 static 영역에 있는 것이 아니다.  
그리고 num1은 어떤 Instance의 Member인가?

```
        num2++;  
    }  
}
```

---