

Method - Recursive

Recursive

- 수학적 관점에서의 재귀적(순환적) 사고

5! = 5x4x3x2x1	5! = 5x4!
4! = 4x3x2x1	4! = 4x3!
3! = 3x2x1	3! = 3x2!
2! = 2x1	2! = 2x1!
1! = 1	1! = 1

- $f(n) = n \times f(n-1)$ ($n \geq 2$)
1 ($n=1$)
 - 함수 $f(n)$ 을 사용하는데 있어서 다시 그 안에 $f(n)$ 자신을 사용하는 것이 재귀
-

Java - Recursive

- Java는 재귀적 메소드 호출을 허용한다.

```
f(n)
{
    f(n);
}
```

Factorial

- $f(n) = n \times f(n-1)$ ($n \geq 2$)
1 ($n=1$)

```
class ReculFactorial {
    public static void main(String[] args) {
        System.out.println("3 factorial : "+factorial(3));
        System.out.println("12 factorial : "+factorial(12));
    }

    public static void factorial(int n) {
        if(n==1)
            return 1;

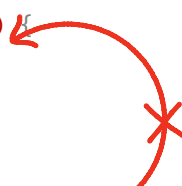
        else
            return n * factorial(n-1);
    }
}
```

처리과정

잘못된 이해!!!!

```
public static void factorial(3) {
    if(n==1)
        return 1;

    else
```



```

        return 1;

    else
        return 3 * factorial(2);
}

```

메소드가 호출될 때에 복사본을 하나 마련한다라고 생각해야 한다. 실제로도 바이트코드를 복사한다.

```

public static void main(String[] args) {
    System.out.println("3 factorial : "+factorial(3));
}

```

1. Copy.1 (인자 3)

```

public static void factorial(3) {
    if(n==1)
        return 1;

    else
        return 3 * factorial(2);
}

```

2. Copy.2 (인자 2)

```

public static void factorial(2) {
    if(n==1)
        return 1;

    else
        return 2 * factorial(1);
}

```

3. Copy.3 (인자 1)

```

public static void factorial(1) {
    if(n==1)
        return 1; // 재귀의 종료, 자신을 호출했던 복사본으로 return하며 거슬러 올라간다.

    else
        return n * factorial(n-1);
}

```

4. Copy.2로 반환

```

public static void factorial(2) {
    if(n==1)
        return 1;

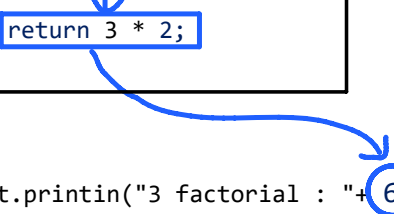
    else
        return 2 * 1;
}

```

5. Copy.1으로 반환

5. Copy.1으로 반환

```
public static void factorial(3) {  
    if(n==1)  
        return 1;  
  
    else  
        return 3 * 2;  
}
```



6. 최종반환

```
System.out.println("3 factorial : "+6);
```

잘못된 Recursive Method

```
class InfRecul {  
    public static void main(String[] args) {  
        showHi(3);  
    }  
  
    public static void showHi(int cnt) {  
        System.out.println("Hi~");  
        showHi(cnt--); // --가 Postfix라서 계속 3이 전달되므로 무한루프..  
  
        // 종료조건 위치도 잘못되었다.  
        if(cnt==1)  
            return;  
    }  
}
```

정답

```
class Recul {  
    public static void main(String[] args) {  
        showHi(3);  
    }  
  
    public static void showHi(int cnt) {  
        if(cnt==1)  
            return;  
  
        System.out.println("Hi~");  
        showHi(--cnt);  
    }  
}
```
