

제8장

객체와 클래스 - 2



15. 재귀 호출(recursive call)

■ 재귀 호출

- 메서드 내에서 자기자신을 반복적으로 호출하는 것을 말한다.
- 재귀 호출은 얼마든지 반복문으로 바꿀 수 있다.
- 또한, 반복문보다 성능이 떨어진다.(스택에 계속 쌓인다.)
- 그래도, 코드가 간결하고 가독성이 좋아 많이 사용한다.

■ 재귀 호출의 예

- 팩토리얼, 제곱, 트리, 폴더목록 등

*팩토리얼(factorial)

$4! = 4 * 3 * 2 * 1$

$f(n) = n * f(n-1)$ 단, $f(1) = 1$

```
long factorial(int n) {  
    long result = 0;  
    if(n==1) {  
        result = 1;  
    } else {  
        result = n * factorial(n-1);  
    }  
    return result;  
}
```

16. 재귀 호출의 예(팩토리얼)

```
public class FactorialEx {  
  
    public static void main(String[] args) {  
  
        long result = factorial(4L);  
        System.out.println("4! (팩토리얼) 값 : " + result);  
    }  
    //자기 자신을 호출하는 재귀  
    //for문으로 해도 되나, 코드를 보는 것과 코드 중복을 제거함으로써 좋다.  
    public static long factorial(long n) {  
        long result = 0L;  
        //비로소 1일 때, 재귀 호출이 더 이상 안어두어진다. f(1) = 1 이니깐.  
        if (n == 1) {  
            result = 1;  
        }  
        else {  
            System.out.println("result값 : " + result + " n값 : " + n);  
            result = n * factorial(n-1); // 메서드 자신을 호출한다.  
        }  
        return result;  
    }  
}
```

factorial	n	<input type="text" value="1"/>	result	<input type="text" value="1"/>
factorial	n	<input type="text" value="2"/>	result	<input type="text" value="0"/>
factorial	n	<input type="text" value="3"/>	result	<input type="text" value="0"/>
factorial	n	<input type="text" value="4"/>	result	<input type="text" value="0"/>
main	result	<input type="text"/>		

```
Console X X  
<terminated> FactorialEx [Java Applicatio  
result값 : 0 n값 : 4  
result값 : 0 n값 : 3  
result값 : 0 n값 : 2  
4! (팩토리얼) 값 : 24
```

감사합니다.

