

2018학년도 2학기
JAVA 프로그래밍 II

514770
2018년 가을학기
9/11/2018
박경신

과제 Lab2_1 (method)

1. 본 과제에서 작성하는 코드는 Lab2_1.java 파일에 저장한다.
2. 재귀호출(recursive call)을 사용하여, 다음 메소드(method) 프로그램을 작성하라. 이 프로그램은 사용자 입력을 받아서 계산하여 출력한다.
 1. 양의 정수 n 의 $n!$ (팩토리얼) 메소드를 작성하라.
 2. 양의 정수 n 의 $f(n)$ (피보나치 수열) 메소드를 작성하라.
 3. 양의 정수 b , n 의 b^n (파워) 메소드를 작성하라.
 4. 양의 정수 a , b 의 최대공약수 메소드를 작성하라.
 5. 본인이 원하는 재귀호출 메소드를 추가로 작성하라.

과제 Lab2_2 (class)

1. 본 과제에서 작성하는 코드는 UserInput.java, Utility.java, Lab2_2.java 파일로 나뉜다.
2. 사용자로부터 정수를 입력받는 메소드를 포함하는, UserInput 클래스를 작성하라.
3. Lab2_1에서 작성했던 method를 포함하는, Utility 클래스를 작성하라.
4. 그리고, UserInput 클래스의 메소드와 Utility 클래스의 메소드를 사용하여, Lab2_1과 동일하게 동작하는 Lab2_2 프로그램을 작성하라.

과제 Lab2_3 (array)

1. 본 과제에서 작성하는 코드는 UserInput.java, Utility.java, Lab2_3.java 파일로 나뉜다.
2. UserInput 클래스를 작성하라.
 1. 사용자로부터 하나의 정수를 입력받는 메소드 `int getInteger()`
 2. 사용자로부터 두 개의 정수를 입력받는 메소드 `int[] getTwoIntegers()`
 3. 사용자로부터 여러 개의 정수를 입력받는 메소드 `int[] getMultipleIntegers()`
3. 그리고, UserInput 클래스의 `getInteger()`와 `getTwoIntegers()` 메소드를 사용하고 Utility 클래스의 메소드를 사용하여, Lab2_1, Lab2_2와 동일하게 동작하는 Lab2_3 프로그램을 작성하라.
4. 그리고, UserInput 클래스의 `getMultipleIntegers()` 메소드를 사용하여 사용자 입력된 숫자들의 Factorial과 Fibonacci 계산결과를 출력하라.

과제 Lab2_4 (enum)

1. 본 과제에서 작성하는 코드는 Calculator.java, UserInput.java, Utility.java, Lab2_4.java 파일로 나뉜다.
2. CalculatorMode 열거형을 작성하라. FACTORIAL, FIBONACCI, POWER, GCD
3. UserInput 클래스에 다음 메소드를 추가로 작성하라.
 1. 사용자로부터 하나의 Calculator 열거형을 입력받는 메소드 Calculator getCalculator()
 2. 사용자로부터 q-key를 입력받았으면 true, 아니면 false를 주는 메소드 Boolean getExitKey()
4. 그리고, do-while을 사용하여 Calculator를 선택하여 계산을 하는, Lab2_4 프로그램을 작성하라.

<https://introcs.cs.princeton.edu/java/23recursion/>

재귀호출 (Recursive call)

- 함수에서 자기 자신을 다시 부르는 것을 재귀호출 (recursive call)이라고 함
- 함수 내부에서 사용되는 지역 변수의 값들은 자기 자신을 호출하기 전의 값을 호출 후에도 그대로 보존함 (자기 자신을 다시 부르더라도 새로운 지역 변수들이 생성되는 것을 생각하면 됨)
- 재귀호출을 빠져나갈 수 있도록 검사하고 종료하는 부분이 반드시 존재해야 함 (재귀호출 탈출 조건)
- 재귀호출의 사용 예: 팩토리얼 (factorial), 최대공약수, 피보나치 수열, 등

재귀호출

- 팩토리얼을 구현하는 함수 작성
- $n! = n \times (n - 1) \times (n - 2) \times \dots \times 1 = n \times (n - 1)!$ (where $n \geq 1$)
- 알고리즘
 - 만약 n 이 0보다 작거나 같으면 값 1을 반환
 - 만약 n 이 0보다 크다면 $n * (n - 1)!$ 을 반환

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ n(n-1)! & \text{if } n > 0 \end{cases}$$

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

```
factorial(1); // 1
factorial(5); // 5! = 5 * 4 * 3 * 2 * 1 = 120
```

재귀호출

- 피보나치 수열 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233
 - 알고리즘
 - 1항과 2항은 1
 - 3항 이후부터의 n 항은 $(n - 1)$ 항 + $(n - 2)$ 항
- $$F(n) = F(n - 1) + F(n - 2) \text{ for } n \geq 2, \text{ with } F(0) = 0 \text{ and } F(1) = 1$$

```
static long fibonacci(int n) {
    if (n == 0) return 0;
    if (n == 1) return 1;           // 재귀호출 탈출 조건
    return fibonacci(n - 1) + fibonacci(n - 2);
}
fibonacci(1)           // 1
fibonacci(2)           // 1
fibonacci(3)           // 2
fibonacci(6)           // 8
```

재귀호출

□ power 함수 작성

□ 알고리즘

- 만약 n이 0이면, 값 1을 반환
- 만약 n이 0이 아니라면, $b * \text{power}(b, n - 1)$ 을 반환

$$b^n = \underbrace{b \times \dots \times b}_n$$

```
static int power(int b, int n){
    if (n == 0)
        return 1;
    else
        return(b * power(b, n-1));
}
```

```
power(3, 2); // 3^2 = 3 * 3 = 9
power(5, 5); // 5^5 = 5 * 5 * 5 * 5 * 5 = 3125
```

$$b^n = \begin{cases} 1 & \text{if } n = 0 \\ b * b^{(n-1)} & \text{if } n > 0 \end{cases}$$

재귀호출

□ 최대공약수(Greatest Common Divisor)

□ 알고리즘

- 만약 $a > b$, $\text{gcd}(a,b) = \text{gcd}(b, a\%b)$

$$\text{gcd}(a, b) = \begin{cases} a & \text{if } b = 0 \\ \text{gcd}(b, a\%b) & \text{if } b > 0 \end{cases}$$

```
static int gcd(int a, int b) {
    if (b == 0)
        return a;           // 재귀호출 탈출 조건
    else
        return gcd(b, a%b);
}
gcd(12, 4);                // 4
gcd(12, 18);               // 6
```

과제 제출

- Lab2_1 ~ Lab2_4와 보고서를 전체적으로 묶어서 e-learning에 과제 제출