

Report : Assignment 1

Written by Sorathorn Kaewchotchuankul 63070501067 CPE REGULAR

สิ่งที่ทำใน Assignment นี้

ใน Assignment นี้ได้ทำฟังก์ชันในการคำนวณแบบ Recursive และแสดงผลข้อความทั้งก่อน return คำตอบและหลัง return คำตอบของการคำนวณค่าแบบ Recursion ฟังก์ชันรับข้อมูล ฟังก์ชันในการแสดงผลข้อความและเรียนรู้การเขียนภาษา java ไปบางส่วนที่จำเป็นต่อ Assignment นี้

Sourcecode

```
1 import java.util.Scanner;
2
3 public class Ass1 {
4     static int getvalue(String text,int min,int max) {
5         int a = 0;
6         boolean check = false;
7         char ch=0;
8         Scanner input = new Scanner(System.in);
9         while(!check) {
10             System.out.printf(text);
11             try {
12                 a = input.nextInt();
13                 if(a>=min && a<=max)
14                     check = true;
15                 else
16                     System.out.printf("Invalid Input please Enter value between %d - %d\n",min,max);
17             }
18             catch(Exception err){
19                 System.out.printf("Invalid Input please Enter value between %d - %d\n",min,max);
20                 input.nextLine();
21             }
22         }
23         return a;
24     }
25 }
26
27
28 }
```

รูปที่ 1 method getvalue

```
29 static void show_factorial(int n,int n2) {
30     System.out.printf("%d! is recursive case. Answer = %d * recursion of %d! \n",n,n,n2);
31     System.out.printf("\tRecursion to calculate %d!\n",n2);
32 }
33
34 }
35 static void show_factorial2(int n,int n2,long back,long x){
36     System.out.printf("\tReturn answer from %d! = %d to calculate %d! = [ %d * %d! ] = %d * %d = %d\n",n2,back,n,n,n2,n,back,x);
37 }
38 static long factorial(int n) {
39     long x,back;
40     int n2;
41     if(n == 0) { //base
42         System.out.printf("0! is base case return answer of 0! = 1\n");
43         System.out.printf("Calculate 0! complete. \n");
44         return 1;
45     }
46     else { // recursive
47         show_factorial(n,n-1);
48         x = n*factorial(n-1);
49         back = x / n;
50         if(n-1>0) {
51             n2 = n-1;
52             System.out.printf("Calculate %d! complete. \n",n2);
53         }
54         show_factorial2(n,n-1,back,x);
55         return x;
56     }
57 }
```

รูปที่ 2 method show_factorial,show_factorial 2, factorial

```

58* public static void main(String[] args) {
59     int n;
60     long ans;
61     char ch = 0;
62     Scanner in = new Scanner(System.in);
63     System.out.printf("My Recursion Program. \n");
64     System.out.printf("Calculate Factorial n! by recursion \n");
65     do {
66         n = getValue("Enter the value of n = ",0,15);
67         ans = factorial(n);
68         System.out.printf("Complete calculation of %d! , answer = %d\n",n,ans);
69         System.out.printf("press[y] to continue, others to exit. ");
70         if (in.hasNext())
71             ch = in.next().charAt(0);
72     }while(ch == 'y' || ch == 'Y');
73     System.out.println("End Program.");
74     System.out.println("Program written by 63070501067 Sorathorn Kaewchotchuangkul CPE Regular.");
75 }
76
77
78 }

```

รูปที่ 3 method main

Test case

```
My Recursion Program.
Calculate Factorial n! by recursion
Enter the value of n = 4
4! is recursive case. Answer = 4 * recursion of 3!
    Recursion to calculate 3!
3! is recursive case. Answer = 3 * recursion of 2!
    Recursion to calculate 2!
2! is recursive case. Answer = 2 * recursion of 1!
    Recursion to calculate 1!
1! is recursive case. Answer = 1 * recursion of 0!
    Recursion to calculate 0!
0! is base case return answer of 0! = 1
Calculate 0! complete.
    Return answer from 0! = 1 to calculate 1! = [ 1 * 0! ] = 1 * 1 = 1
Calculate 1! complete.
    Return answer from 1! = 1 to calculate 2! = [ 2 * 1! ] = 2 * 1 = 2
Calculate 2! complete.
    Return answer from 2! = 2 to calculate 3! = [ 3 * 2! ] = 3 * 2 = 6
Calculate 3! complete.
    Return answer from 3! = 6 to calculate 4! = [ 4 * 3! ] = 4 * 6 = 24
Complete calculation of 4! , answer = 24
press[y] to continue, others to exit. y
```

- Case 1

สรุปผล : ใส่ค่า n เท่ากับ 4 โปรแกรมแสดงผลออกมาโดยจะคำนวณไปเป็นลำดับของ recursion ไปเรื่อยๆจนเจอ base case และรีเทิร์นค่า base case กลับมาและกลับไปคิด recursion ที่โปรแกรมเก็บค้างไว้และรีเทิร์นค่าคำตอบของแต่ละตัวออกมาเป็นลำดับจนถึงตัวที่ n-1 ก็จะได้คำตอบออกมาและสรุปผลว่าได้คำตอบเท่าไรให้ผู้ใช้ทราบ

อธิบายโค้ด : เมื่อโปรแกรมทำงานก็จะแสดงผลข้อความให้ผู้ใช้ทราบและถามค่า n ซึ่งเป็นการทำงานใน method getvalue โดยสร้างตัวสำหรับรับข้อมูลชื่อ input และวน loop จนกว่าจะรับค่าถูกต้อง เมื่อทำเสร็จจะส่งค่ากลับมาใน method main และใน case นี้ผู้ใช้ใส่ค่า n = 4 โปรแกรมก็จะรีเทิร์นค่ากลับมา n ก็จะเท่ากับ 4 และส่งค่า n ไปคิดใน method factorial โดย method factorial จะส่งค่าที่ได้มาไปเปรียบเทียบว่าเป็น base case หรือ recursive case ในกรณีนี้เป็น recursive ก็จะทำเงื่อนไข recursive คือแสดงข้อความก่อนคำนวณค่าทำใน method show_factorial คำนวณค่า recursion และแสดงข้อความหลังจากคำนวณได้ base case ใน method show_factorial2 มาแล้วทำจนครบและจะรีเทิร์นคำตอบกลับไปให้ main

หลังจากนั้นจะแสดงคำตอบให้ผู้ใช้ทราบและตั้งคำถามถามผู้ใช้ว่าต้องการคำนวณต่อหรือไม่โดยจะมีตัวแปรชื่อ in ที่เป็น Scanner ที่สร้างขึ้นมาไว้เพื่อรับค่าโดยมีเงื่อนไขว่าถ้าตัวต่ออ่านได้นั้นจะอ่านมาเก็บไว้ในตัวแปร ch และนำไปตรวจสอบกับเงื่อนไขถ้าตรงกับตัว y หรือ Y ก็จะรันโปรแกรมต่อถ้าไม่ตรงก็จบโปรแกรม

```
press[y] to continue, others to exit. y
Enter the value of n = 2.0
Invalid Input please Enter value between 0 - 15
Enter the value of n = 2.
Invalid Input please Enter value between 0 - 15
Enter the value of n = x
Invalid Input please Enter value between 0 - 15
Enter the value of n = 2x
Invalid Input please Enter value between 0 - 15
Enter the value of n = -1
Invalid Input please Enter value between 0 - 15
Enter the value of n = 20|
Invalid Input please Enter value between 0 - 15
```

- Case 2

สรุปผล : ใน case นี้เป็นการทดลองป้อนข้อมูลที่ผิดพลาดให้โปรแกรมปรากฏว่าโปรแกรมแสดงผลว่าข้อมูลผิดพลาดให้กรอกข้อมูลตั้งแต่ 0-15 เท่านั้น

อธิบายโค้ด : เป็นการทำงานตอนอยู่ใน method getvalue เมื่อเข้า loop ในการรับค่าแล้ว Error ที่ 1 นั้นตอนรับค่าโปรแกรมต้องการข้อมูลประเภท Integer ทำให้เกิด Error เมื่อป้อนค่าไม่ใช่จำนวนเต็มใน try จึงไปทำใน catch แล้วแสดงผลออกมาว่าข้อมูลไม่ถูกต้องแล้วอ่านข้อมูลทิ้งเพื่อป้องกันข้อมูลที่ค้างอยู่ในคีย์บอร์ด Error ที่ 2 คือกรอกข้อมูลที่เป็น Integer แต่ไม่อยู่ในช่วงที่กำหนดไว้ทำให้กรณีนี้ทำงานใน try {} แต่ไม่เข้าเงื่อนไขที่ค่าอยู่ในช่วงที่กำหนดไว้ทำให้ยังไม่สามารถรับค่าได้สำเร็จจึงแสดงผลออกมาว่าข้อมูลผิดพลาดไม่ตรงกับเงื่อนไขที่กำหนดไว้

```

press[y] to continue, others to exit. y
Enter the value of n = 1
1! is recursive case. Answer = 1 * recursion of 0!
    Recursion to calculate 0!
0! is base case return answer of 0! = 1
Calculate 0! complete.
    Return answer from 0! = 1 to calculate 1! = [ 1 * 0! ] = 1 * 1 = 1
Complete calculation of 1! , answer = 1
press[y] to continue, others to exit. n
End Program.
Program written by 63070501067 Sorathorn Kaewhotchuangkul CPE Regular.

```

- case 3

สรุปผล : ใน case นี้จะเหมือนกันกับ case 1 แตกต่างกันแค่ค่าของ n เท่านั้นโดยใส่ n = 0 หรือ 1 ไปจะได้คำตอบออกมาเท่ากันคือ 1

อธิบายโค้ด : case นี้เป็นการทำงานต่อเนื่องจากโปรแกรมโดยใส่ค่า y เมื่อโปรแกรมถามว่าต้องการคำนวณต่อหรือไม่โปรแกรมจึงตั้งถามถาม n ใหม่เพื่อจะคำนวณต่อและ case นี้ใส่ค่าเท่ากับ 0 และ 1 โปรแกรมจะทำงานเหมือนกันกับ case 1 ต่างกันตรงที่เมื่อ n = 0 การทำงานใน method factorial ไม่ต้องการในกรณี recursive case แต่ทำเพียง base case ก็ได้คำตอบแล้ว หลังจากนั้นจะแสดงคำตอบให้ผู้ใช้ทราบและตั้งคำถามถามผู้ใช้ว่าต้องการคำนวณต่อหรือไม่โดยจะมีตัวแปรชื่อ in ที่เป็น Scanner มารับข้อมูลที่ผู้ใช้ป้อนใน case นี้ป้อนค่าไม่ใช่ y หรือ Y ทำให้โปรแกรมหลุด loop และแสดงผลข้อความจบโปรแกรมพร้อมชื่อผู้เขียน

หมายเหตุ : รูป sourcecode ของฟังก์ชันต่างๆอยู่ด้านบนหัวข้อ sourcecode

สรุปความเข้าใจของตนเอง

ใน Assignment นี้เราจะคำนวณค่า factorial แบบ Recursion และดูลำดับการทำงานแบบ recursive และแสดงผลข้อความออกมาช่วงต่างกันจะมีทั้งก่อนจะได้ค่าของ base case หลังจากได้ค่าของ base case แล้วเป็นต้นและได้สร้างฟังก์ชันเพื่อรับค่าของ n โดยมีค่าที่กำหนดไว้ให้อยู่ในช่วงที่ต้องการโดยการใช้อการวนลูปเมื่อตรวจสอบค่าที่รับมาโดยมีทั้งค่าที่รับแบบไม่ใช่ประเภทที่ต้องการเลยและเป็นประเภทที่ต้องการแต่ค่าไม่อยู่ในช่วงที่กำหนด

ประเมินตนเอง

ให้ตัวเองอยู่ในระดับ 80 เพราะสามารถทำโจทย์ได้แต่ยังมีบางจุดที่คิดผิดทำให้เขียนโปรแกรมออกมาแล้วแสดงผลไม่ได้ตามที่ต้องการบ้างเล็กน้อย



63070501067
SORATHORN
KAEWCHOTCHUANGKUL

Grading Rubric

[View Full Rubric](#)

Criterion 1

20	40	60	80	100
----	----	----	----	-----

80

การเขียนโปรแกรมด้วย Algorithms ที่กำหนดให้

20	40	60	80	100
----	----	----	----	-----

80