

## ใบงานการทดลองที่ 3

เรื่อง อาร์เรย์สตริง และฟังก์ชัน ในภาษาจาวา

### 1. จุดประสงค์ทั่วไป

1.1. รู้และเข้าใจการ โปรแกรมเชิงวัตถุร่วมกับอาร์เรย์และสตริง

1.2. รู้และเข้าใจการ โปรแกรมเชิงวัตถุร่วมกับฟังก์ชัน

### 2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

### 3. ทฤษฎีการทดลอง

3.1. โครงสร้างข้อมูลแบบ “อาร์เรย์” มีลักษณะเป็นอย่างไร ? มีองค์ประกอบอะไรบ้าง ? อธิบายพร้อมยกตัวอย่างประกอบ

อาร์เรย์ (Array) คือ ประเภทของข้อมูลรูปแบบหนึ่งที่สามารถเก็บข้อมูลประเภทเดียวกันแบบเป็นลำดับได้ โดยข้อมูลนั้นจะอยู่ในตัวแปรตัวเดียวกันที่เรียกว่า ตัวแปรอาร์เรย์ เปรียบเสมือนว่าเราสร้างห้องขึ้นมาห้องหนึ่งเพื่อเก็บข้อมูลต่างๆในปริมาณมาก โดยข้อมูลแต่ละตัวของอาร์เรย์ จะเรียกว่า อีลีเมนต์ (Element) และข้อมูลแต่ละอีลีเมนต์จะมีหมายเลขเพื่อใช้ในการกำกับตำแหน่งคือ เลขดัชนี (Index) ทำให้ถึงจะเป็นตัวแปรที่ชื่อเหมือนกัน แต่ก็แตกต่างกันตรงหมายเลขดัชนีที่กำกับไว้ จึงทำให้ตัวแปรอาร์เรย์นั้นมีประโยชน์อย่างมากในกรณีที่ต้องเก็บข้อมูลปริมาณมากเพราะสามารถเก็บทีละได้หลายจำนวน บวกกับง่ายต่อการนำออกไปใช้งานเพราะแต่ละข้อมูลถึงจะเป็นข้อมูลที่เหมือนกัน แต่มีตัวเลข index กำกับไว้อยู่เสมอ ดังนั้นจึงไม่ทำให้เกิดการสับสนของข้อมูล

เช่น เราสร้างข้อมูล 3 ตัว โดยปกติจะต้องสร้างตัวแปร 3 ตัว เพื่อที่จะเก็บข้อมูลนั้นๆ ดังรูปต่อไปนี้

```
$flower1 = "Rose";  
$flower2 = "Violet";  
$flower3 = "tulip";
```

### 3.2. การเข้าถึงแต่ละ Element ของอาร์เรย์สามารถทำได้อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ

array คือกลุ่มของข้อมูลที่เรียงลำดับกัน มีจำนวนแน่นอนซึ่งข้อมูลจะเป็นประเภทเดียวกัน ข้อมูลแต่ละตัวของอาร์เรย์ จะเรียกว่า อีลีเมนต์(Element) และข้อมูลแต่ละอีลีเมนต์จะมีหมายเลขเพื่อใช้ในการอ้างอิงถึงเรียกตัวเลขนี้ ว่า เลขดัชนี (Index) จะเป็นตัวแปรที่ชื่อ เหมือนกัน แต่จะแตกต่างกันตรงหมายเลข



เช่น

```
int Score[4];
```

ในที่นี้มีความหมายว่า เป็นการประกาศตัวแปร array ชื่อ Score มีจำนวน 4 รายการ โดยมีรายการที่

```
Score[0]
```

```
Score[1]
```

```
Score[2]
```

```
Score[3]
```

```
Score[0] Score[1] Score[2] Score[3]
```

```
int int int int
```

รายการของ array จะเริ่มที่ 0 ไม่ได้เริ่มที่ 1 ถ้าเราประกาศตัวแปร array เช่น `int i[3]` ก็จะมีรายการที่ 0 ถึง 2 จะไม่มีหมายเลข

### 3.3. คำสั่ง length เกี่ยวข้องกับอาเรย์อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ

อาเรย์เก็บข้อมูลด้วย Index และ Index ของอาเรย์นั้นเริ่มต้นจาก 0 นั่นหมายความว่าเมื่ออาเรย์ของเรามีสมาชิกทั้งหมด N ตัว เราสามารถเข้าถึงค่าของอาเรย์ได้จาก Index 0 ถึง N - 1 โดยที่ N เป็นขนาดของอาเรย์ที่สามารถอ่านค่าได้จาก Property length นี่เป็นตัวอย่าง

accessing\_array1.js

```
let numbers = [10, 20, 30, 40, 50];
console.log("numbers[0]:", numbers[0]);
console.log("numbers[1]:", numbers[1]);
console.log("numbers[2] + numbers[3]:", numbers[2] + numbers[3]);
console.log("numbers[4]:", numbers[4]);
console.log("numbers[4]:", numbers[numbers.length - 1]);
```

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

```
numbers[0]: 10
numbers[1]: 20
numbers[2] + numbers[3]: 70
numbers[4]: 50
numbers[4]: 50
```

### 3.4. จงยกตัวอย่างประกอบในการวนรอบเพื่อแสดงค่าภายในตัวแปรอาเรย์ตั้งแต่ค่าแรกจนถึงค่าสุดท้าย

อาเรย์ขนาด 2 มิติขึ้นไปจะไม่ระบุขนาดได้เฉพาะมิติที่ 1 เท่านั้น ส่วนมิติอื่นๆ ต้องมีการระบุขนาดด้วยทุกครั้ง

```
#include <stdio.h>
void main()
{
    int table[12][11], row, col;
    printf("*** multiplication table ***");
    for( row = 1 ; row <= 12 ; row++ )
    {
        printf("\n");
        for( col = 2 ; col <= 12 ; col++ )
        {
            table[row - 1][col - 2] = row * col;
            printf(" %3d ", table[row - 1][col - 2]);
        }
    }
}
```

### 3.5. จงยกตัวอย่างการใช้งานคำสั่ง for each เพื่อแสดงค่าภายในตัวแปรอาเรย์

ถ้าเราต้องการจะเข้าถึงข้อมูลแต่ละคู่ที่ถูกเก็บอยู่ใน associative array เราอาจจะใช้วิธีเรียกผ่านฟังก์ชัน each() และ list() ตามตัวอย่างต่อไปนี้

```
<?php
unset($a);
$a = array( "a" => 10, "b" => 20, "c" => 30 );
while (list($key,$value) = each($a)) {
    echo "$key=$value <BR> ";
}
?>
```

### 3.6. เหตุใดจึงต้องมีคำสั่ง import java.util.Arrays ; ในส่วนต้นของไฟล์?

สำหรับการใช้ Arrays จะต้องทำการ import คลาสก่อนถึงจะใช้ได้ ด้วย import java.util. Arrays ที่นี้เราก็มีสแกนเนอร์สำหรับอ่านค่าแล้ว ต่อไปก็คือสั่งให้ Arrays เก็บข้อมูลประเภทเดียวกันแบบเป็นลำดับ โดยจะต้องบอกให้ Arrays รู้ว่าเราอยากได้ค่า (Value) เป็นตัวแปรชนิดไหน แบบนี้

### 3.7. คำสั่ง Arrays.copyOf( **Original** , **15** ) ; มีหน้าที่ทำอะไร ?

คุณสามารถใช้เมธอด Arrays.copyOf() เพื่อคัดลอกอาร์เรย์ใน Java วิธีนี้ทำให้คุณสามารถคัดลอกองค์ประกอบทั้งหมดหรือบางส่วนจากอาร์เรย์ใน Java ได้ แต่องค์ประกอบจะต้องต่อเนื่องกัน เช่น องค์ประกอบ 5 ตัวแรกหรือ 10 ตัวแรกของอาร์เรย์

### 3.8. จงยกตัวอย่างการประกาศ String และกำหนดค่าคำว่า “Hello World” ในภาษาจาวา

```
public class lab1 {

    public static void main (String[] args) {

        System.out.println( "Hello World" ); //

        // YOUR CODE HERE

    } //end method

} //end class
```

### 3.9. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toUpperCase() ในภาษาจาวา

Java toUpperCase() - String เป็นรูปแบบ property และ method เกี่ยวกับข้อความ (String) โดย toUpperCase() จะเป็นการแปลงข้อความ String ให้อยู่ในรูปแบบของ ตัวอักษรพิมพ์ใหญ่

```
01. package com.java.myapp;
02.
03. public class MyClass {
04.
05.     public static void main(String[] args) {
06.
07.         String a = "Welcome to ThaiCreate.Com";
08.
09.         System.out.println(a.toUpperCase());
10.
11.     }
12.
13. }
```

### 3.10. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toLowerCase() ในภาษาจาวา

javascript tolowercase คือ หนึ่งในเมธอดที่มีไว้ใช้เพื่อแปลงตัวอักษรใน javascript สำหรับผู้ใช้งาน ที่ทำงานผ่านทาง javascript เป็นหลักอาจจะกำลังประสบกับปัญหาการจัดระเบียบตัวอักษรในคอมไพเลอร์ ชนิดนี้

```
for fruit in fruits
    fruit.toUpperCase()

# => "APPLES"
# => "ORANGES"
# => "BANANAS"
```

### 3.11. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง indexOf() ในภาษาจาวา

Java indexOf() - String เป็นรูปแบบ property และ method เกี่ยวกับข้อความ (String) โดย indexOf() จะเป็นการหาตำแหน่งของข้อความที่ต้องการค้นหา

```
01. package com.java.myapp;
02.
03. public class MyClass {
04.
05.     public static void main(String[] args) {
06.
07.         String a = new String("Welcome to ThaiCreate.Com Version 2013");
08.         String b = new String("ThaiCreate.Com");
09.
10.         System.out.println(a.indexOf("Version"));
11.         System.out.println(a.indexOf(b));
12.
13.     }
14.
15. }
```

### 3.12. จงอธิบายความแตกต่างระหว่างการเชื่อม String แบบปกติและแบบใช้คำสั่ง concat()

ฟังก์ชัน CONCAT จะรวมข้อความจากช่วงและ/หรือสตริงหลายๆ ส่วน แต่จะไม่มีตัวกันหรืออาร์กิวเมนต์ IgnoreEmpty CONCAT จะแทนที่ฟังก์ชัน CONCATENATE อย่างไรก็ตาม ฟังก์ชัน CONCATENATE จะยังใช้งานได้กับ Excel เวอร์ชันก่อนหน้า

### 3.13. หากต้องการแสดงสัญลักษณ์พิเศษภายในตัวแปร String ควรทำอะไร ?

บางตัวอักษรเช่น " หรือ \ เป็นสัญลักษณ์ที่ Java ใช้สำหรับการโปรแกรม แต่ถ้าเราจะให้มันเป็นตัวอักษรเฉย ๆ ไม่เกี่ยวกับการเขียนโปรแกรม เราต้องใช้ \ นำหน้า เช่น

```
System.out.println("Hello"); // print Hello
```

```
System.out.println("\\"Hello\\"); // print "Hello"
```

### 3.14. จงอธิบายและยกตัวอย่างประกอบการสร้างฟังก์ชันในภาษาจาวา

ฟังก์ชันหรือเรียกอีกชื่อหนึ่งว่าเมทอด(method) มีความสำคัญต่อการเขียนโปรแกรมมากเพราะช่วยในการนำส่วนของโปรแกรมที่มีการทำงานซ้ำๆ แยกออกมาเป็นส่วนย่อยๆ ช่วยลดความซ้ำซ้อนของโค้ดภายในโปรแกรมและทำให้อ่านง่ายด้วย

เมื่อสร้างเมทอดในภาษา Java มันสามารถที่จะเรียกใช้งานได้จากส่วนใดๆ ของโปรแกรม ขึ้นกับขอบเขตและระดับการเข้าถึงที่ได้กำหนดขึ้น โดยรูปแบบในการสร้างเมทอดในภาษา Java เป็นดังนี้

```
type name ( parameter1, parameter2, ... ) {  
    statements  
}  
  
access_modifier type name ( parameter1, parameter2, ... ) {  
    statements  
}
```

### 3.15. อธิบายข้อแตกต่างระหว่าง Pass by value และ Pass by reference

Pass by Value คือ การส่งค่า (value) เป็น argument ของฟังก์ชัน ดังนั้นค่าที่ทำในฟังก์ชันจึงไม่ส่งผลต่อตัวแปรนอกฟังก์ชัน Pass by Reference คือ การส่งตัวแปร (variable) เป็น argument ของฟังก์ชัน ดังนั้นตัวแปรที่มีการดำเนินการใด ๆ ในฟังก์ชันจะส่งผลให้ตัวแปรนอกฟังก์ชันมีการเปลี่ยนแปลงด้วย

### 3.16. ความแตกต่างระหว่างการประกาศฟังก์ชันแบบ void กับแบบ int, double, float, string คืออะไร?

public void Java คือการกำหนดระดับการเข้าถึง method หรือ function ด้วย public แปลว่า  
สาธารณะสามารถเรียกใช้งานได้ทั้งภายใน และภายนอก class ส่วน void คือ method ดังกล่าวไม่มีการ  
return ค่า สามารถเขียนโปรแกรม

int เป็นชนิดข้อมูลจำนวนเต็ม ในทางคณิตศาสตร์ (ซึ่งไม่ควรนำมาเก็บตัวแปรประเภท จำนวนเงิน  
หรือตัวเลขที่มีทศนิยมควรใช้ float,double,decimal)

Java double (data types) สำหรับ double ตัวแปรในภาษา Java จะใช้จัดเก็บตัวเลขในรูปแบบของ  
ทศนิยม ที่มีขนาดอยู่ในช่วง  $-4.9 \times 10^{-324}$  ถึง  $1.79769 \times 10^{308}$  ใช้ขนาด (8 byte) Syntax

Java float (data types) สำหรับ float ตัวแปรในภาษา Java จะใช้จัดเก็บตัวเลขในรูปแบบของ  
ทศนิยม ที่มีขนาดอยู่ในช่วง  $-1.4 \times 10^{-45}$  ถึง  $3.40282 \times 10^{38}$  ใช้ขนาด (4 byte) Syntax

String เป็น Class หนึ่งใน Package ของภาษาจาวาชื่อ java.lang ทำหน้าที่ใน การเก็บข้อมูลที่เป็น  
“ชุดของตัวอักษร”

### 3.17. โครงสร้างข้อมูลแบบ Stack แตกต่างกับ Array อย่างไร ?

สแตก(stack) เป็นโครงสร้างข้อมูลแบบเชิงเส้น ที่มีการใส่ข้อมูลเข้า และนำข้อมูลออกเพียงด้าน  
เดียว ดังนั้น ข้อมูลที่เข้าไปอยู่ใน stack ก่อนจะออกจาก stack หลังข้อมูลที่เข้าไปใน stack ที่หลัง นั่นคือ การ  
"เข้าทีหลังแต่ออกก่อน" (Last In First Out : LIFO)

อาร์เรย์ (Array)เป็นโครงสร้างข้อมูลที่มีการจองพื้นที่หน่วยความจำ (Memory) เป็นชุด ๆ แต่ละชุด  
ประกอบด้วยจำนวนช่องข้อมูลหลายช่อง พื้นที่แต่ละช่องข้อมูลจะเก็บข้อมูลชนิดเดียวกัน และอยู่ใน  
ตำแหน่งที่ต่อเนื่องกันไปตามลำดับ



3.18. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง Push ในโครงสร้างข้อมูลแบบ Stack

- การนำข้อมูลเข้าไปในกองซ้อน (Push) กระทำที่ส่วนบนของสแตก (Top) ซึ่งต้องมีการตรวจสอบก่อนว่าสแตกเต็มหรือไม่ เป็นการดำเนินการที่นำข้อมูลเข้าไปเก็บไว้ด้านบนสุดของกองซ้อน (Top of the Stack) เรื่อย ๆ จนกว่ากองซ้อนไม่สามารถนำข้อมูลเข้าไปเก็บได้จะเรียกว่า กองซ้อนเต็ม (Stack Full)

3.19. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง Pop ในโครงสร้างข้อมูลแบบ Stack

- การนำข้อมูลออกจากกองซ้อน (Pop)การทำงานจะตรงข้ามกับ Push จะดึงเอาข้อมูลที่อยู่บนสุดออกมาก่อน แต่ก่อนที่จะดึงจะมีการตรวจสอบว่ากองซ้อนว่างหรือไม่ ถ้าว่างจะไม่สามารถนำข้อมูลออกได้ แสดงว่ากองซ้อนว่าง (Stack Empty)ถ้าไม่ว่างจะนำเอาข้อมูลออกแล้วเลื่อนตัวชี้ไปยังตำแหน่งถัดลงไป

3.20. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง isEmpty ในโครงสร้างข้อมูลแบบ Stack

ส่วน isEmpty() ใช้เพื่อตรวจสอบว่า "ตัวแปรนั้นๆ ยังไม่ได้ถูกกำหนดขึ้น หรือมีค่าที่ว่างเปล่า หรือ "ไม่" พูดย่างๆ คือตรวจว่า มีค่าอยู่หรือไม่ ซึ่งรูปแบบของ "ค่าที่ว่างเปล่า" หรือ "ไม่มีค่า" นี้ได้แก่

empty() นั่นก็เหมือนกับ isset() ใช้ตรวจสอบเฉพาะตัวแปรเท่านั้น

```
$a = 'Hello World';  
$c = '';  
$d = array();  
empty($a); // false เพราะมีค่าที่ไม่ใช่ค่าที่ว่างเปล่า  
empty($b); // true เพราะไม่มีตัวแปร $b อยู่  
empty($c); // true เพราะมีค่าเป็นสตริงว่าง  
empty($d); // true เพราะมีค่าเป็น array ที่ว่างเปล่า
```

### 3.21. อธิบายพร้อมยกตัวอย่างประกอบความหมายของคำว่า Stack overflow

สำหรับโปรแกรมเมอร์มือใหม่ที่เริ่มหัดเขียนโปรแกรม หลาย ๆ คนอาจจะยังไม่รู้จักเว็บ Stackoverflow แต่สำหรับโปรแกรมเมอร์มืออาชีพ จะรู้จักเว็บ Stackoverflow เป็นอย่างดีแน่นอน ซึ่งเว็บ Stackoverflow จะคล้าย ๆ กับเว็บ pantip.com ของทางบ้านเรา แต่เว็บ Stackoverflow จะเป็นศูนย์รวมของการตั้งกระทู้ถามตอบกันที่ใหญ่ที่สุดในโลกของเหล่าโปรแกรมเมอร์ และกระทู้ที่ถามตอบกันจะใช้ภาษาอังกฤษ

ซึ่งภายในเว็บ Stackoverflow จะมีการตั้งกระทู้ถามถึงปัญหาเกี่ยวกับโปรแกรม ถ้าโปรแกรมเมอร์คนไหนไม่เคยเข้าไปอ่านกระทู้ถามตอบของ Stackoverflow เลยแสดงว่าโปรแกรมเมอร์คนนั้นไม่เคยเขียนโปรแกรมด้วยตนเอง !!! สำหรับเจ้าของบทความแล้ว จะเข้าไปหาอ่านคำตอบของกระทู้มากกว่าจะไปตั้งกระทู้ถาม เนื่องจากทุกปัญหาของโปรแกรมส่วนใหญ่ไม่ได้เกิดกับเจ้าของบทความเพียงคนเดียว แต่มีคนพบเจอปัญหานี้มาก่อนที่เจ้าของบทความจะเจอแล้ว ยกเว้นแต่เจ้าของบทความจะเขียนโปรแกรมเฉพาะทางจริง ๆ ที่คนส่วนใหญ่เขาไม่ทำกันก็จะไม่มีใครมาตั้งกระทู้ให้อ่าน เจ้าของบทความจะยกตัวอย่างการค้นหาวิธีการแปลงชนิดตัวแปรจาก string ไปเป็น int ในภาษา Java

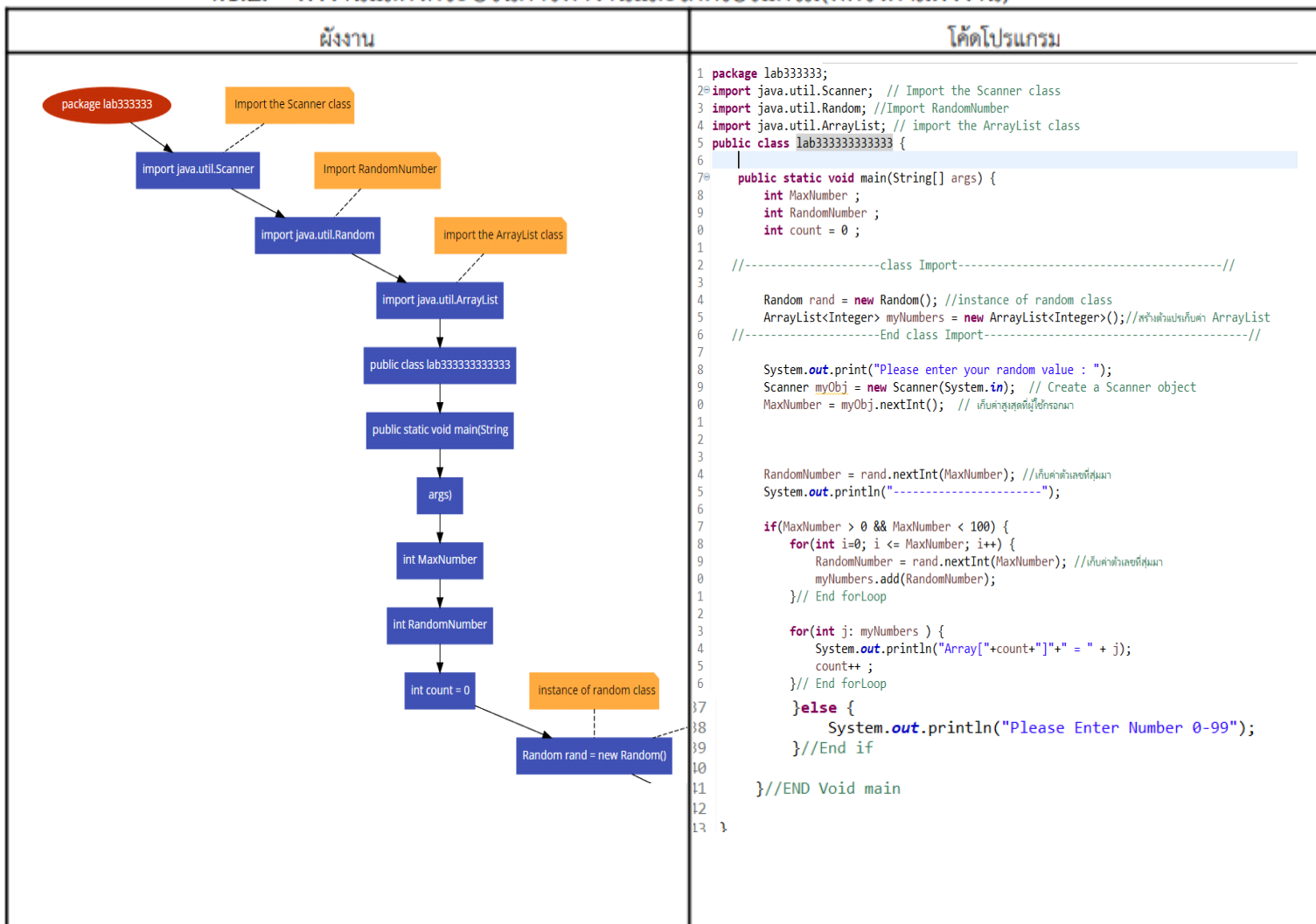
#### 4. ลำดับขั้นการปฏิบัติการ

##### 4.1. จงแก้โจทย์ปัญหาดังต่อไปนี้

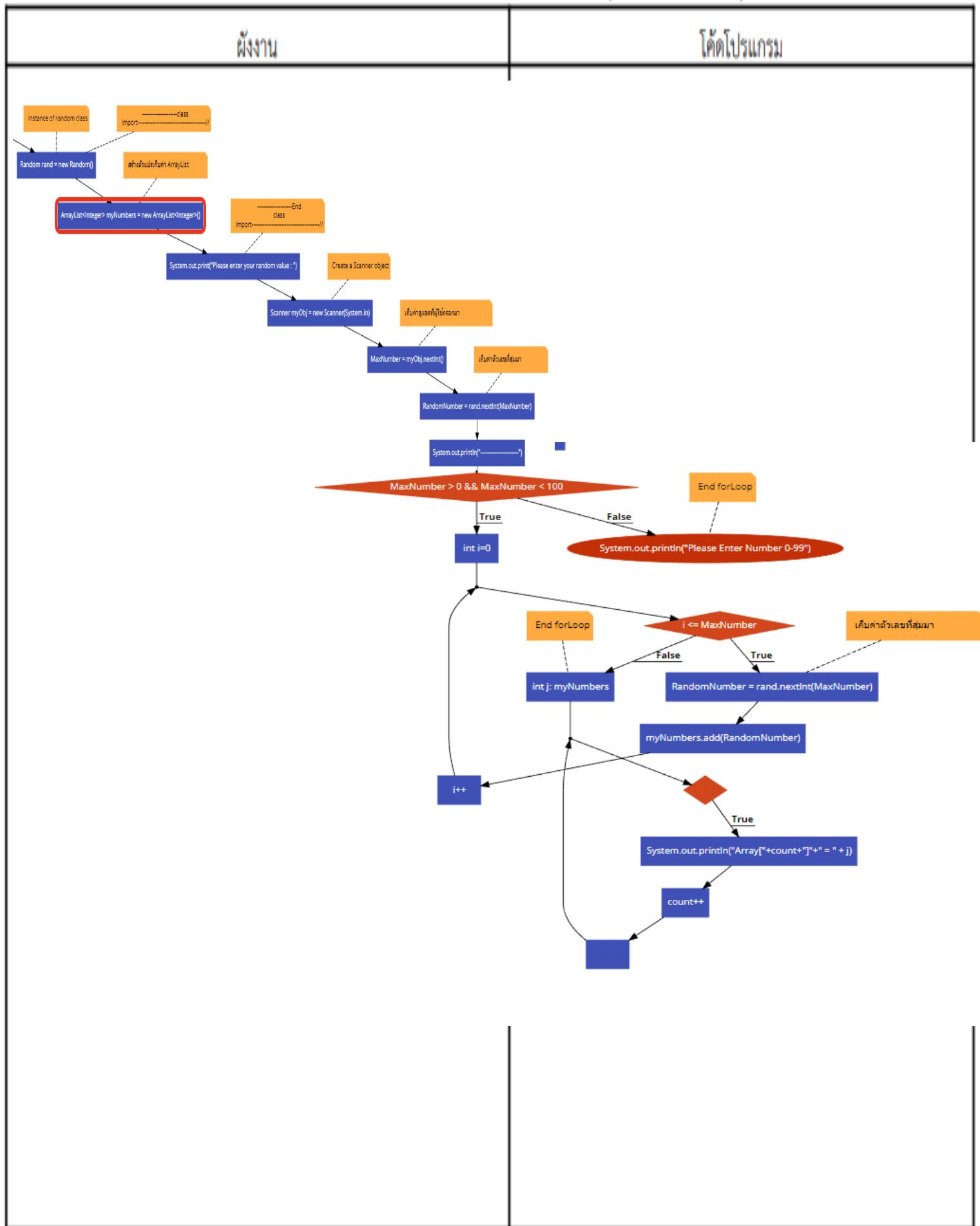
4.1.1. จงเขียนโปรแกรมเพื่อสุ่มค่าเข้าไปในอาร์เรย์ 1 มิติตามจำนวนค่าที่รับจากผู้ใช้ โดยค่าที่ถูกสุ่มจะต้องเป็นตัวเลขจำนวนเต็มที่อยู่ระหว่าง 0 ถึง 99 เท่านั้น

Test case 1	Test case 2
Please enter your random value : 8	Please enter your random value : 12
-----	-----
Array[0] = 94	Array[0] = 56
Array[1] = 32	Array[1] = 27
Array[2] = 46	Array[2] = 13
Array[3] = 18	Array[3] = 15
Array[4] = 27	Array[4] = 65
Array[5] = 5	Array[5] = 29
Array[6] = 31	Array[6] = 11
Array[7] = 17	Array[7] = 92
	Array[8] = 95
	Array[9] = 47
	Array[10] = 58
	Array[11] = 62

##### 4.1.2. ผังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผังงาน)



## 4.1.2. ผังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผังงาน)



## 4.2. จงแก้ไขข้อปัญหาดังต่อไปนี้

### 4.2.1. จงเขียนฟังก์ชันการจัดการโครงสร้างข้อมูลแบบ Stack พร้อมจำลองการทำงานโดยการเรียกใช้

คำสั่งพื้นฐานดังต่อไปนี้

คำสั่ง Push( String Value ) ; เพื่อนำข้อมูลเข้าไปเก็บไว้ใน Stack

คำสั่ง Pop() ; เพื่อนำข้อมูลบนสุดออกจาก Stack

คำสั่ง isEmpty() ; เพื่อตรวจสอบข้อมูลใน Stack ว่ามีอยู่หรือไม่

คำสั่ง Top() ; เพื่อตรวจสอบข้อมูลที่อยู่ชั้นบนสุด

คำสั่ง CheckStack() ; เพื่อตรวจสอบค่าภายใน Stack ทั้งหมด

คำสั่ง SetStackSize( int size ) ; เพื่อกำหนดขนาดเริ่มต้นของ Stack

Test case
SetStackSize( 3 )
isEmpty
---  Yes
Top
---  NULL
Push : Hello
CheckStack
---  STACK : Hello
Push : Hi
CheckStack
---  STACK : Hi, Hello
Push : Test
CheckStack
---  STACK : Test, Hi, Hello
Top
---  Top = Test
Pop
CheckStack

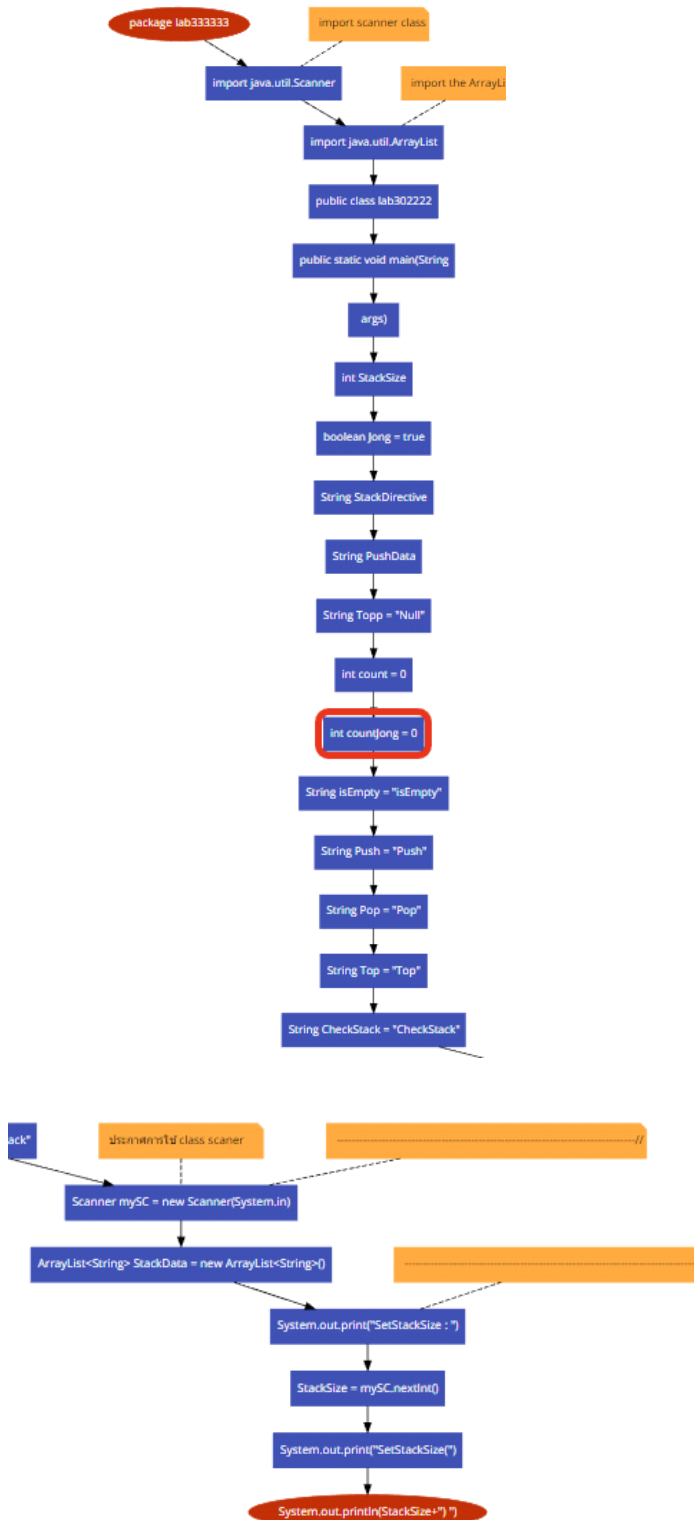
```

---| STACK : Hi, Hello
isEmpty
---| No
Push : OK
CheckStack
---| STACK : OK, Hi, Hello
Push : RMUTL
---| Stack Overflow
CheckStack
---| STACK : OK, Hi, Hello
    
```

## 4.2.2. ผลงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผลงาน)

ผังงาน

โค้ดโปรแกรม



```

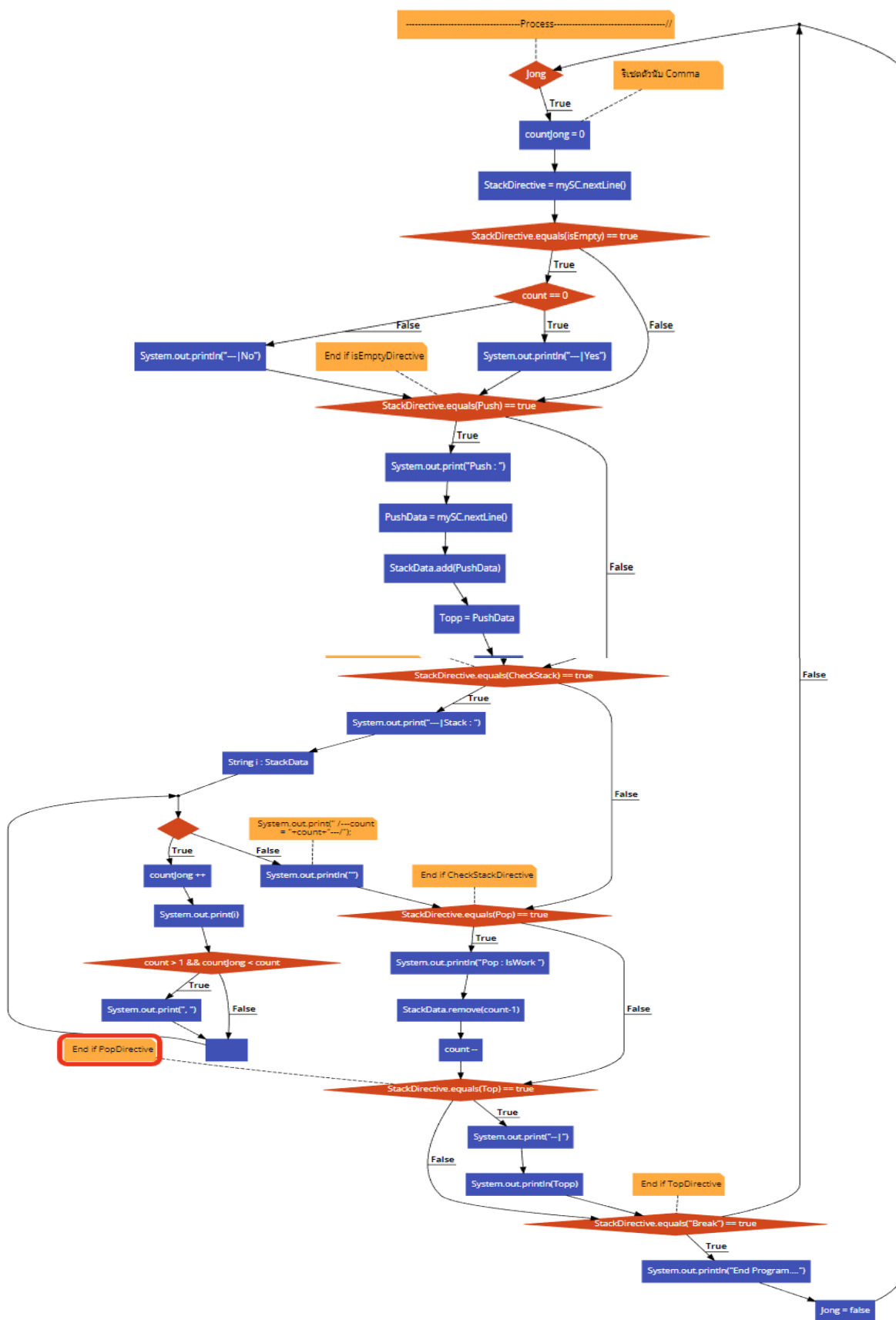
1 package lab333333;
2
3 import java.util.Scanner; //import scanner class
4 import java.util.ArrayList; // import the ArrayList class
5 public class lab302222 {
6
7     public static void main(String[] args){
8         int StackSize ;
9         boolean Jong = true ;
10        String StackDirective ;
11        String PushData ;
12        String Topp = "Null" ;
13        int count = 0 ;
14        int countJong = 0;
15        String isEmpty = "isEmpty" ;
16        String Push = "Push" ;
17        String Pop = "Pop" ;
18        String Top = "Top" ;
19        String CheckStack = "CheckStack" ;
20
21        //Scanner mySC = new Scanner(System.in); //ประกาศการใช้ class scanner
22        ArrayList<String> StackData = new ArrayList<String>();
23        //System.out.print("SetStackSize : ");
24        StackSize = mySC.nextInt();
25        System.out.print("SetStackSize(");
26        System.out.println(StackSize+" ") ;
27
28        //-----Process-----
29        while(Jong) {
30            countJong = 0 ; //รีเซ็ตตัวนับ Comma
31            StackDirective = mySC.nextLine();
32
33            if(StackDirective.equals(isEmpty) == true ) {
34                if(count == 0) {
35                    System.out.println("---|Yes");
36                } else {
37                    System.out.println("---|No");
38                }
39            }
40            //End if isEmptyDirective
41
42            if(StackDirective.equals(Push) == true ) {
43                System.out.print("Push : ");
44                PushData = mySC.nextLine();
45                StackData.add(PushData);
46                Topp = PushData ;
47                count ++ ;
48            }
49            //End if PushDirective
50
51            if(StackDirective.equals(CheckStack) == true ) {
52                System.out.print("---|Stack : ");
53                for (String i : StackData) {
54                    countJong ++ ;
55                    System.out.print(i);
56                    if(count > 1 && countJong < count) {
57                        System.out.print(", ");
58                    }
59                }
60                //System.out.print(" /---count = "+count+"---/");
61                System.out.println("");
62            }
63            //End if CheckStackDirective
64
65            if(StackDirective.equals(Pop) == true ) {
66                System.out.println("Pop : IsWork ");
67                StackData.remove(count-1);
68                count -- ;
69            }
70            //End if PopDirective
71
72            if(StackDirective.equals(Top) == true ) {
73                System.out.print("--|");
74                System.out.println(Topp);
75            }
76            //End if TopDirective
77
78            if(StackDirective.equals("Break") == true ) {
79                System.out.println("End Program...");
80                Jong = false ;
81            }
82            //End Condition Exit to Loop
83        }
84    }
85 }
86

```

## 4.2.2. แผนผังแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผังงาน)

ผังงาน

โค้ดโปรแกรม



## 5. สรุปผลการปฏิบัติการ

จากการทดลองที่ 4.1.1. ได้ผลดังนี้สุ่มค่าเข้าไปในอาร์เรย์ 1 มิติตามจำนวนค่าที่รับจากผู้ใช้งาน โดยค่าที่ถูกสุ่มจะต้องเป็นตัวเลขจำนวนเต็มที่อยู่ระหว่าง 0 ถึง 99 เท่านั้น

```
<terminated> lab33333333333 [Java Application] C:\Users\Augsu
Please enter your random value : 8
-----
Array[0] = 6
Array[1] = 3
Array[2] = 6
Array[3] = 0
Array[4] = 6
Array[5] = 0
Array[6] = 3
Array[7] = 5
Array[8] = 7
```

```
<terminated> lab33333333333 [Java Application] C:\Users\A
Please enter your random value : 12
-----
Array[0] = 2
Array[1] = 10
Array[2] = 7
Array[3] = 10
Array[4] = 10
Array[5] = 9
Array[6] = 10
Array[7] = 0
Array[8] = 3
Array[9] = 6
Array[10] = 11
Array[11] = 11
Array[12] = 3
```

จากการทดลองที่ 4.2.1. ฟังก์ชันการจัดการโครงสร้างข้อมูลแบบ Stack พร้อมจำลองการทำงานโดยการใช้คำสั่งที่เห็นในผลลัพธ์ดังในรูป

```
lab302222 [Java Application] C:\Users\Augsupan tharawa
SetStackSize : 3
SetStackSize(3)

isEmpty
---|Yes

top
Top
--|Null

Push
Push : Hello
CheckStack
---|Stack : Hello

push
Push
Push : hi
CheckStack
---|Stack : Hello, hi

Push
Push : Test
CheckStack
---|Stack : Hello, hi, Test

Top
--|Test

Pop
Pop : IsWork

CheckStack
---|Stack : Hello, hi

isEmpty
---|No
```

```
Push
Push : Ok
CheckStack
---|Stack : Hello, hi, Ok

Push
Push : RMUTL
CheckStack
---|Stack : Hello, hi, Ok, RMUTL
```



## 6. คำถามท้ายการทดลอง

### 6.1. ข้อควรระวังในการใช้งาน Array ในภาษาจาวาคืออะไร ?

คอมไพเตอร์มักจะถูกใช้ในการคำนวณข้อมูลที่มีปริมาณมาก ๆ เช่น การหาค่าเฉลี่ยของตัวเลข ถ้าเราเก็บข้อมูลของแต่ละค่าไว้ในตัวแปรแต่ละตัว เราจะต้องใช้ตัวแปรปริมาณมาก ๆ ดังนั้น จึงมีการนำ Array มาใช้ในการเก็บข้อมูลประเภทนี้ บทความนี้ เรามาทำความรู้จักกับ Array การสร้าง การใช้งานต่าง ๆ กันครับ

### 6.2. ข้อควรระวังในการใช้งาน String ในภาษาจาวาคืออะไร ?

String ถือว่าสิ่งที่เราจะพบเจอบ่อยที่สุดในการเขียนโปรแกรม เพราะ String นั้นสามารถจัดเก็บรูปแบบข้อความได้หลายหลายมาก ทั้งข้อความทั่ว ๆ ไป ที่รับค่าจาก Input หรือตัวแปรแล้วนำไปแสดงผลทางหน้าจอ หรือข้อความที่เป็นแบบ Object ที่ถูกสร้างขึ้นมาแล้วจัดเก็บไว้ในตัวแปร ที่สามารถนำไปใช้งานอย่างอื่นได้

### 6.3. ฟังก์ชันในภาษาจาวาไม่สามารถใช้งานแบบ Pass by reference ในภาษาซีได้คุณมีแนวทางการ

แก้ไขปัญหานี้ได้อย่างไร ?

Pass by Reference คือ การส่งตัวแปร (variable) เป็น argument ของฟังก์ชัน ดังนั้นตัวแปรที่มีการดำเนินการใด ๆ ในฟังก์ชันจะส่งผลให้ตัวแปรนอกฟังก์ชันมีการเปลี่ยนแปลงด้วย หากไม่กำหนดอะไรเพิ่มเติม C#/Java จะ pass by value เสมอ (มองในมุมมองของตัวแปร) เรียกว่า pass the reference by value แต่ที่เราเรียนเรามองในมุมมองของ Object ที่ตัวแปรนั้น ๆ ซึ่อยู่จึงเป็นการ pass by reference ของ Object นั้น ดังนั้นตัวแปร reference type หากมีการส่งค่าให้กับ method จะมีการ pass by reference โดยอัตโนมัติมองในมุมมองของ Object เพราะที่เราสอนเรามองในมุมมองของ Object นั้นเอง

6.4. โครงสร้างข้อมูลแบบ Stack แตกต่างกับโครงสร้างข้อมูลแบบ Array อย่างไร ?

โครงสร้างข้อมูลแบบสแตก (Stack) เป็นโครงสร้างข้อมูลแบบเชิงเส้น ที่มีการใส่ข้อมูลเข้า และนำข้อมูลออกเพียงด้านเดียว ดังนั้น ข้อมูลที่เข้าไปอยู่ใน stack ก่อนจะออกจาก stack หลังข้อมูลที่เข้าไปใน stack ที่หลัง นั่นคือ การ "เข้าทีหลังแต่ออกก่อน" (Last In First Out : LIFO)

โครงสร้างข้อมูลแบบ Array เป็นโครงสร้างข้อมูลที่มีการจองพื้นที่หน่วยความจำ (Memory) เป็นชุด ๆ แต่ละชุดประกอบด้วยจำนวนช่องข้อมูลหลายช่อง พื้นที่แต่ละช่องข้อมูลจะเก็บข้อมูลชนิดเดียวกัน และอยู่ในตำแหน่งที่ต่อเนื่องกัน ไปตามลำดับ