

## ใบงานการทดลองที่ 3

### เรื่อง อาร์เรย์สตริง และฟังก์ชัน ในภาษาจาวา

#### 1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการโปรแกรมเชิงวัตถุร่วมกับอาร์เรย์และสตริง
- 1.2. รู้และเข้าใจการโปรแกรมเชิงวัตถุร่วมกับฟังก์ชัน

#### 2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

#### 3. ทฤษฎีการทดลอง

- 3.1. โครงสร้างข้อมูลแบบ “อาร์เรย์” มีลักษณะเป็นอย่างไร ? มีองค์ประกอบอะไรบ้าง ? อธิบายพร้อมยก

ตัวอย่างประกอบ

- Array คือการจัดเก็บข้อมูลประเภทเดียวกันเรียงเป็นลำดับ โดยข้อมูลเหล่านั้นจะถูกเก็บไว้ในที่ ที่เรียกว่า ตัวแปรอาร์เรย์ โดยข้อมูลแต่ละตัว ของอาร์เรย์ จะเรียกว่า อีลีเมนต์(Element) และข้อมูลแต่ละอีลีเมนต์จะมีหมายเลขกำกับเพื่อง่ายต่อการเรียกใช้ข้อมูลส่วนนั้น เรียกว่า เลขดัชนี (Index)
- เช่น เราสร้างข้อมูล 3 ตัว โดยปกติจะต้องสร้างตัวแปร 3 ตัว เพื่อที่จะเก็บข้อมูลนั้นๆ ในกรณีนี้เราต้องสร้างตัวแปร 3 ตัว ซึ่งการสร้างตัวแปรจำนวนมากเป็นวิธีที่ลำบากเกินไปและสิ้นเปลืองทรัพยากรได้ ดังนั้นจึงมีการใช้งาน array เกิดขึ้นโดยสร้างตัวแปรอาร์เรย์ขึ้นมาเพียงหนึ่งตัว เพื่อเก็บค่า ของทั้ง 3 ตัวแปร

- 3.2. การเข้าถึงแต่ละ Element ของอาร์เรย์สามารถทำได้อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ

- โดยใช้เลขอ้างอิงตำแหน่งคือ เลขดัชนี (Index) เพื่อกำหนดตัวเลขดัชนี (index) เพื่อเป็นการกำกับว่าต้องการดึงข้อมูลชนิดไหนออกมา
- เช่น 

```
System.out.println(myList[0]);  
System.out.println(myList[1]);  
System.out.println(myList[2]);
```

- 3.3. คำสั่ง length เกี่ยวข้องกับอาร์เรย์อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ

- เป็นการนับขนาดของตัวแปรอาร์เรย์นั้นว่ามีขนาดที่เก็บข้อมูลอยู่มากน้อยแค่ไหน
- เช่น นำข้อมูลไปเก็บที่ ตัวแปรอาร์เรย์ แต่จำไม่ได้ว่านำข้อมูลเข้าไปเก็บเท่าไร สามารถใช้คำสั่ง length เพื่อตรวจสอบขนาดของอาร์เรย์ได้

3.4. จงยกตัวอย่างประกอบในการวนรอบเพื่อแสดงค่าภายในตัวแปรอาเรย์ตั้งแต่ค่าแรกจนถึงค่าสุดท้าย

```
int[] number = {1, 2, 3};  
for (int i = 0; i < number.length; i++) {  
    System.out.println(number[i]);  
}
```

3.5. จงยกตัวอย่างการใช้งานคำสั่ง for each เพื่อแสดงค่าภายในตัวแปรอาเรย์

```
int[] number = {1, 2, 3};  
for (int i : number) {  
    System.out.println(i);  
}
```

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

- การจะใช้คลาส Arrays จำเป็นที่จะต้องทำการ import แพคเกจ java.util.Arrays

3.7. คำสั่ง Arrays.copyOf( \_\_\_\_, \_\_\_\_) ; มีหน้าที่ทำอะไร ?

- เป็นคำสั่งที่ใช้ในการคัดลอก ข้อมูลในตัวแปรอาเรย์หนึ่ง ไปยังอีกตัวแปรอาเรย์หนึ่ง
- ตัวอย่าง `int[] dest = Arrays.copyOf(source, source.length);`

3.8. จงยกตัวอย่างการประกาศ String และกำหนดค่าคำว่า "Hello World" ในภาษาจาวา

- `String str = "Hello World";`

3.9. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toUpperCase() ในภาษาจาวา

- คำสั่งทำให้อักษรที่เก็บไว้ในตัวแปร String นั้น ๆ เป็นตัวพิมพ์ใหญ่ทั้งหมด

3.10. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toLowerCase() ในภาษาจาวา

- คำสั่งทำให้อักษรที่เก็บไว้ในตัวแปร String นั้น ๆ เป็นตัวพิมพ์เล็กทั้งหมด

3.11. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง indexOf() ในภาษาจาวา

- เป็นการหาตำแหน่งอักษรตัวแรกที่เราระบุ เริ่มหาจากซ้ายไปทางขวา

3.12. จงอธิบายความแตกต่างระหว่างการเชื่อม String แบบปกติและแบบใช้คำสั่ง concat ( )

- การเชื่อม String แบบปกติ เป็นการใส่ตัวดำเนินการ + ระหว่างสตริงเพื่อรวมเข้าด้วยกัน
- แบบใช้คำสั่ง concat ( ) เป็นการเชื่อม String โดยไม่ใช้ตัวดำเนินการ +

3.13. หากต้องการแสดงสัญลักษณ์พิเศษภายในตัวแปร String ควรทำอย่างไร ?

- ใส่เครื่องหมาย backslash ( \ ) หน้าสัญลักษณ์พิเศษ

### 3.14. จงอธิบายและยกตัวอย่างประกอบการสร้างฟังก์ชันในภาษาจาวา

- ต้องประกาศ method ภายใน class กำหนดด้วยชื่อ method ตามด้วยวงเล็บ () Java มี method ที่กำหนดไว้ล่วงหน้า เช่น System.out.println() แต่คุณสามารถสร้างเมธอดของคุณเองเพื่อดำเนินการบางอย่าง
- ตัวอย่างเช่น

```
public class Main {  
    static void myMethod() {  
        // code to be executed  
    }  
}
```

### 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 ?

- ฟังก์ชันแบบ void คือฟังก์ชันที่ไม่ส่งค่ากลับมาเมื่อจบการทำงานของฟังก์ชันนั้นแล้ว
- ฟังก์ชันแบบ int, double, float, string คือฟังก์ชันที่ส่งค่ากลับ

### 3.17. โครงสร้างข้อมูลแบบ Stack แตกต่างกับ Array อย่างไร ?

- ข้อมูลแบบ Stack เป็นการเก็บข้อมูลแบบทางเดียว การจะดึงข้อมูลจำเป็นต้องดึงข้อมูลล่าสุดที่เก็บไว้ก่อน ถ้าต้องการข้อมูลตัวแรกจำเป็นต้องดึงข้อมูลทุกตัวออกมาก่อนถึงจะได้
- ข้อมูลแบบ Array เป็นการเก็บข้อมูลแบบเรียงที่เลือกตำแหน่งเก็บและดึงข้อมูลได้อิสระได้

### 3.18. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง Push ในโครงสร้างข้อมูลแบบ Stack

- คำสั่ง Push คือ การนำข้อมูลเข้าจาก Stack โดยจะตรวจสอบก่อนว่า Stack เต็มหรือไม่
- เช่นการพับผ้าแล้วเก็บไว้ในตะกร้าทับกันไปเรื่อย ๆ

### 3.19. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง Pop ในโครงสร้างข้อมูลแบบ Stack

- คำสั่ง Pop คือ การนำข้อมูลออกจาก Stack โดยจะตรวจสอบก่อนว่า Stack ว่างหรือไม่
- เช่นการนำผ้าที่พับแล้วในตะกร้าออกมา

### 3.20. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง isEmpty ในโครงสร้างข้อมูลแบบ Stack

- คำสั่ง isEmpty คือคำสั่งที่จะตรวจสอบ Stack ว่า ว่างเปล่าหรือไม่ ค่าที่ได้จะเป็น boolean

### 3.21. อธิบายพร้อมยกตัวอย่างประกอบความหมายของคำว่า Stack overflow

- ในกรณีที่ PUSH ข้อมูลลงสู่ Stack จนตัวชี้สแต็กเท่ากับจำนวนช่องของสแต็กแล้ว จะไม่สามารถทำการ PUSH ข้อมูล Stack ได้อีก เนื่องจากตัวชี้ Stack ไม่สามารถที่จะขยับไปยังช่องต่อไปได้ จะเกิด Error ที่เรียกว่า Stack Overflow

#### 4. ลำดับชั้นการปฏิบัติการ

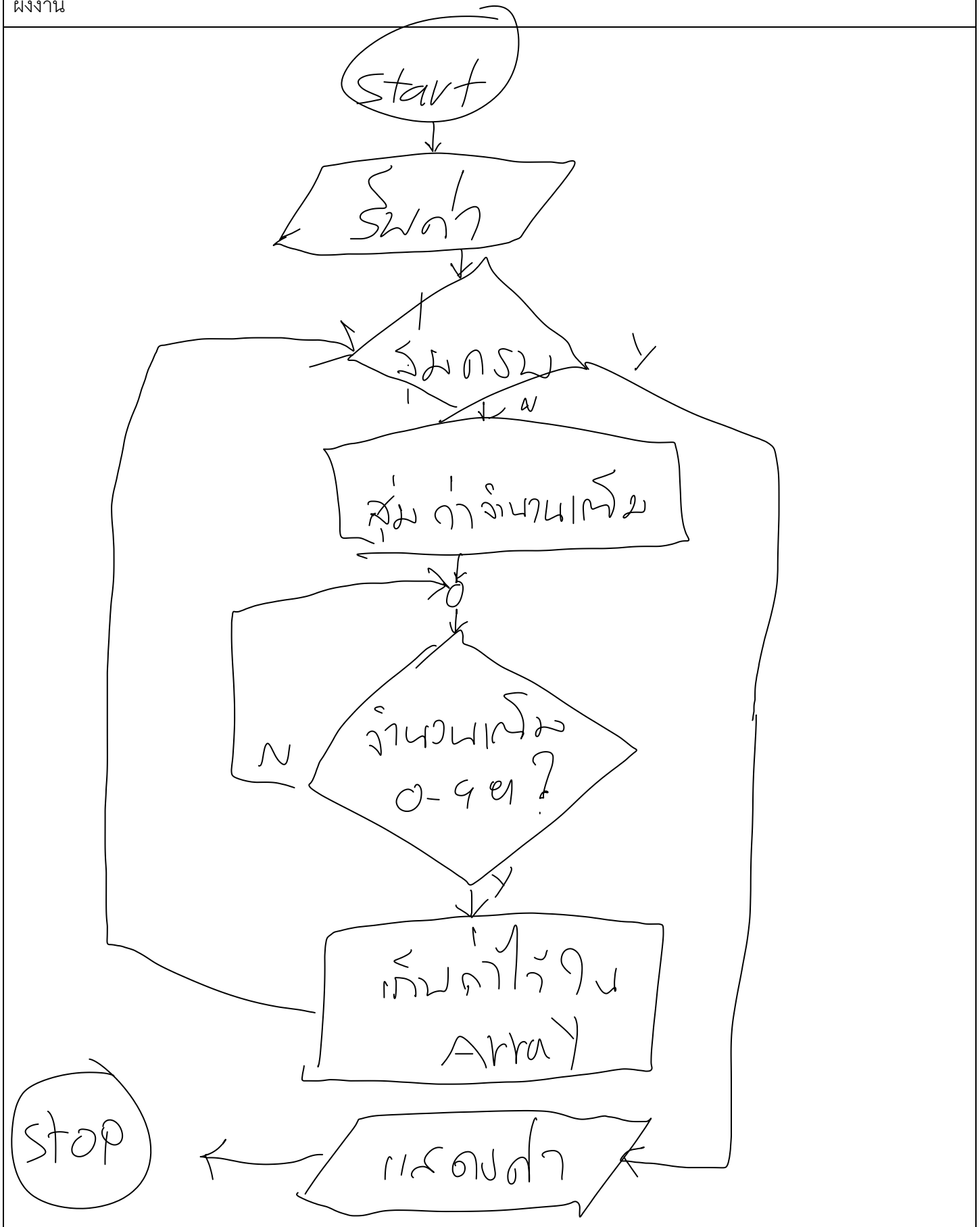
##### 4.1. จงแก้โจทย์ปัญหาดังต่อไปนี้

4.1.1. จงเขียนโปรแกรมเพื่อสุ่มค่าเข้าไปในอาร์เรย์ 1 มิติตามจำนวนค่าที่รับจากผู้ใช้ โดยค่าที่ถูกสุ่มจะต้องเป็นตัวเลขจำนวนเต็มที่อยู่ระหว่าง 0 ถึง 99 เท่านั้น

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

4.1.2. ผังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผังงาน)

ผังงาน



```

1 package OOPPackge;
2
3 import java.util.Scanner; // Import the Scanner class
4
5 public class week3lab1 {
6
7     static int RandNum(int i) {
8         int min = 0; //lowest value of random number
9         int max = 99; //highest value of random number
10        int randnum = (int)(Math.random()*(max-min+1)+min);
11        //Narrowing Casting int randnum <- double random()
12        return randnum ;
13    } //end method
14
15    public static void main(String[] args) {
16        // TODO Auto-generated method stub
17
18        Scanner myObj = new Scanner(System.in); // Create a Scanner object
19        System.out.print("Please enter your value : ");
20        int num = myObj.nextInt(); // Read user input
21        System.out.println("-----");
22
23        int[] TableNum ;
24        TableNum = new int[num];
25
26        for(int i = 0 ; i < num ; i++ ) {
27            TableNum[i] = RandNum(i);
28            //input number to array[i]
29        } //end for
30
31        for(int i = 0 ; i < num ; i++ ) {
32            System.out.println("Array["+i+"] = "+ TableNum[i] );
33            //show number in array[i]
34
35        } //end for
36    } //end void main
37 } //end class
38

```

#### 4.2. จงแก้ไขข้อปัญหาดังต่อไปนี้

4.2.1. จงเขียนฟังก์ชันการจัดการโครงสร้างข้อมูลแบบ Stack พร้อมจำลองการทำงานโดยการเรียกใช้คำสั่งพื้นฐานดังต่อไปนี้

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. ฝั่งงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามฝั่งงาน)

โค้ดโปรแกรม

```
import java.util.Scanner;

public class week3lab2 {

    public static void main(String[] args) {

        int n;
        int i = 0;
        String text;
        System.out.print("Input Stack Size :");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        Stack<String> stack = new Stack<String>();
        stack.setSize(n);
        stack.clear();
        do {
            System.out.println("-----Stack Fn-----");
            System.out.println("[1]Push [2]Pop [3]isEmpty [4]Top [5]CheckStack [0]END ");
            System.out.println("-----");
            System.out.print("Input ");
            i = sc.nextInt();
            switch(i) {
                case 1:
                    System.out.print("Push :");
                    Scanner sct = new Scanner(System.in);
                    text = sct.nextLine();

                    if(stack.size() == n) {
                        System.out.println("----| STACK OVERFLOW!!!!!!");
                    } else {
                        stack.push(text);
                    }
                    break;
            }
        } while(i != 0);
    }
}
```



```
        if(stack.size()==0){
            System.out.println("----| STACK IS EMPTY");
        }else{
            System.out.println("Pop");
            stack.pop();
        }//end if
        break;
    case 3:
        if(stack.isEmpty()==true){
            System.out.println("----| Yes");
        }else{
            System.out.println("----| No");
        }//end if
        break;
    case 4:
        if(stack.size()==0){
            System.out.println("----| NULL");
        }else{
            System.out.println("----| Top :"+stack.peek());
        }//end if
        break;
    case 5:
        System.out.println("----| STACK :"+stack);
        break;
    case 0:
        System.out.println("END STACK");
        i=0;
        break;
```



## 5. สรุปผลการปฏิบัติการ

- ได้เรียนรู้และหลักการใช้งานในการโปรแกรมเชิงวัตถุร่วมของอาเรย์ สตริง และ ฟังก์ชัน ทำให้ง่ายในการเก็บข้อมูลไว้ในตัวแปรตัวเดียว การเก็บตัวอักษรหลายๆตัวในตัวแปรเดียว การเขียนฟังก์ชันทำให้ง่ายต่อการออกแบบและทำความเข้าใจโปรแกรมเชิงวัตถุ

## 6. คำถามท้ายการทดลอง

### 6.1. ข้อควรระวังในการใช้งาน Array ในภาษาจาวาคืออะไร ?

- ดัชนีอาร์เรย์เริ่มต้นด้วย 0 และ ข้อมูลที่จะเก็บไว้ใน Array นั้นต้องเป็นชนิดเดียวกันเท่านั้น

### 6.2. ข้อควรระวังในการใช้งาน String ในภาษาจาวาคืออะไร ?

- String เริ่มนับตำแหน่งจากศูนย์ ตัวเลขใน String เป็นอักษร ไม่สามารถทำการคำนวณทางคณิตศาสตร์ได้ แต่จะเป็นการต่อข้อความเท่านั้น

### 6.3. ฟังก์ชันในภาษาจาวาไม่สามารถใช้งานแบบ Pass by reference ในภาษาซีได้คุณมีแนวทางการแก้ไขปัญหานี้ได้อย่างไร ?

- ต้องกำหนดอาร์กิวเมนต์ของฟังก์ชันเป็น pointer(\*)
- การเรียกใช้งานผ่าน Address เป็น argument (&)
- ภายในฟังก์ชัน การเรียกใช้ formal argument เวลาใช้งานจะใช้ dereferencing operator (\*)

### 6.4. โครงสร้างข้อมูลแบบ Stack แตกต่างกับโครงสร้างข้อมูลแบบ Array อย่างไร ?

- ข้อมูลแบบ Stack เป็นการเก็บข้อมูลแบบทางเดียว การจะดึงข้อมูลจำเป็นต้องดึงข้อมูลล่าสุดที่เก็บไว้ก่อน ถ้าต้องการข้อมูลตัวแรกจำเป็นต้องดึงข้อมูลทุกตัวออกมาก่อนถึงจะได้
- ข้อมูลแบบ Array เป็นการเก็บข้อมูลแบบเรียงที่เลือกตำแหน่งเก็บและดึงข้อมูลได้อิสระได้