

## ใบงานการทดลองที่ 3

### เรื่อง อาร์เรย์ สตริง และฟังก์ชัน ในภาษาจาวา

#### 1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการโปรแกรมเชิงวัตถุร่วมกับอาร์เรย์และสตริง
- 1.2. รู้และเข้าใจการโปรแกรมเชิงวัตถุร่วมกับฟังก์ชัน

#### 2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

#### 3. ทฤษฎีการทดลอง

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

อาร์เรย์เป็นโครงสร้างข้อมูลที่จัดเก็บคอลเลกชันของรายการในทีเดียว เพื่อให้สามารถเข้าถึงแต่ละรายการโดยใช้ดัชนี รายการในอาร์เรย์สามารถเป็นประเภทข้อมูลใดก็ได้ รวมถึงประเภทดั้งเดิม เช่น จำนวนเต็มและค่าศัณยัม และประเภทการอ้างอิง เช่น ออบเจกต์และสตริง

- 3.2. การเข้าถึงแต่ละ Element ของอาร์เรย์สามารถทำได้อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ

ในการเข้าถึงแต่ละองค์ประกอบของอาร์เรย์ คุณสามารถใช้ index ขององค์ประกอบที่ต้องการเข้าถึงได้ index คือค่าจำนวนเต็มที่ระบุตำแหน่งขององค์ประกอบในอาร์เรย์ องค์ประกอบแรกของอาร์เรย์มี index เป็น 0 องค์ประกอบที่สองมี index เป็น 1 เป็นต้น ตัวอย่างเช่น พิจารณาอาร์เรย์ของจำนวนเต็มต่อไปนี้ใน Java

```
int[] numbers = {10, 20, 30, 40, 50};
```

- 3.3. คำสั่ง length เกี่ยวข้องกับอาร์เรย์อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ

length เป็นคุณสมบัติของอาร์เรย์ที่ระบุจำนวนขององค์ประกอบในอาร์เรย์ มักใช้เพื่อวนซ้ำองค์ประกอบของอาร์เรย์หรือเพื่อดำเนินการกับองค์ประกอบทั้งหมดของอาร์เรย์

```
for (int i = 0; i < numbers.length; i++) {  
    int element = numbers[i];  
    System.out.println("The element at index " + i + " is " + element);  
}
```

- 3.4. จงยกตัวอย่างประกอบในการวนรอบเพื่อแสดงค่าภายในตัวแปรอาร์เรย์ตั้งแต่ค่าแรกจนถึงค่าสุดท้าย

```
for (int i = 0; i < numbers.length; i++) {  
    int element = numbers[i];  
    System.out.println("The element at index " + i + " is " + element);  
}
```

- 3.5. จงยกตัวอย่างการใช้งานคำสั่ง for each เพื่อแสดงค่าภายในตัวแปรอาร์เรย์

```

for (int element : numbers) {
    System.out.println("The current element is " + element);
}

```

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

ใน Java คลาส java.util.Arrays เป็นคลาสยูทิลิตี้ที่มีเมธอดต่างๆ สำหรับการทำงานกับอาร์เรย์ นำเข้าคลาสนี้ที่จุดเริ่มต้นของไฟล์ จะสามารถใช้เมธอดเพื่อดำเนินการกับอาร์เรย์ เช่น การเรียงลำดับ การค้นหา และการเติมค่าเฉพาะ

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

Arrays.copyOf() เป็นเมธอดในภาษา Java ที่ใช้สร้างอาร์เรย์ใหม่ที่เป็นสำเนาของอาร์เรย์ที่มีอยู่แล้ว

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

```
String greeting = "Hello World";
```

### 3.9. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toUpperCase() ในภาษาจาวา

```
String s = "hello";
String uppercase = s.toUpperCase(); // ตัวพิมพ์ใหญ่ตอนนี้เป็น "HELLO"
```

toUpperCase() เป็นเมธอดในภาษา Java ที่ใช้ในการแปลงสตริงเป็นตัวพิมพ์ใหญ่ เป็นเมธอดของคลาส String

### 3.10. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toLowerCase() ในภาษาจาวา

```
String s = "HELLO";
String lowercase = s.toLowerCase(); // ตัวพิมพ์เล็กตอนนี้เป็น "hello"
```

toLowerCase() เป็นเมธอดในภาษา Java ที่ใช้ในการแปลงสตริงเป็นตัวพิมพ์เล็ก เป็นเมธอดของคลาส String

### 3.11. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง indexOf() ในภาษาจาวา

```
String s = "Hello World";
int index = s.indexOf('o'); // index is now 4
```

indexOf() เป็นเมธอดในภาษา Java ที่ใช้ค้นหาอักขระหรือสตริงย่อยที่ระบุภายในสตริง เป็นเมธอดของคลาส String

### 3.12. จงอธิบายความแตกต่างระหว่างการเชื่อม String แบบปกติและแบบใช้คำสั่ง concat( )

ใน Java สามารถเชื่อมสตริงเข้าด้วยกันโดยใช้ตัวดำเนินการ + หรือเมธอด concat()

ทั้งตัวดำเนินการ + และเมธอด concat() สามารถใช้เชื่อมสตริงใน Java ได้ อย่างไรก็ตาม โดยทั่วไปแล้วตัวดำเนินการ + จะมีประสิทธิภาพมากกว่า เนื่องจากมีการใช้เมธอดที่มากกว่าเมธอด Java

### 3.13. หากต้องการแสดงสัญลักษณ์พิเศษภายในตัวแปร String ควรทำอย่างไร ?

ใช้ escape characters: สามารถใช้เพื่อแสดงอักขระพิเศษภายในสตริง ตัวอย่างเช่น การใช้ \n เพื่อแสดงการขึ้นบรรทัดใหม่ หรือ \t เพื่อแสดงแท็บ

### 3.14. จงอธิบายและยกตัวอย่างประกอบการสร้างฟังก์ชันในภาษาจาวา

ใน Java ฟังก์ชันคือบล็อกของโค้ดที่ทำงานเฉพาะและสามารถเรียกใช้จากส่วนอื่นๆ ของโปรแกรมได้ ฟังก์ชันเรียกอีกอย่างว่าเมธอดในภาษาจาวา

```
..... public int addNumbers(int num1, int num2) { ..... public static void main(String[] args) {  
.....     int result = num1 + num2; .....     int sum = addNumbers(5, 7); .....     การใช้งาน  
.....     return result; .....     System.out.println("The sum is: " + sum);  
..... } ..... }
```

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

ความแตกต่างระหว่างการ Pass by value และการ Pass by reference คือการที่ฟังก์ชันได้รับอาร์กิวเมนต์ ใน Pass by value ฟังก์ชันจะได้รับสำเนาของค่าของอาร์กิวเมนต์ ในการ Pass by reference ฟังก์ชันจะได้รับการอ้างอิงไปยังตำแหน่งหน่วยความจำของอาร์กิวเมนต์ Java รวบรวมค่าผ่านเท่านั้น

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

ความแตกต่างระหว่างฟังก์ชัน void และฟังก์ชันที่มีประเภทการส่งคืนที่ไม่ใช่โมฆะคือฟังก์ชันส่งคืนค่าหรือไม่ ฟังก์ชัน void จะไม่ส่งคืนค่า ในขณะที่ฟังก์ชันที่มีประเภทการส่งคืนที่ไม่ใช่โมฆะจะต้องส่งคืนค่าของประเภทข้อมูลที่ระบุ

### 3.17. โครงสร้างข้อมูลแบบ Stack แตกต่างกับ Array อย่างไร ?

สแต็กและอาร์เรย์มีความคล้ายกันตรงที่เป็นโครงสร้างข้อมูลเชิงเส้นที่เก็บข้อมูลเป็นลำดับจะแตกต่างกันในวิธีการจัดการกับการแทรกและการลบ ขนาด และการจัดทำ Array

### 3.18. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง Push ในโครงสร้างข้อมูลแบบ Stack

1. สร้าง stack object ใหม่โดยใช้ Stack class: Stack<Type> stack = new Stack<>(); โดยที่ Type คือประเภทของอีลิเมนต์ที่ stack จะเก็บไว้ (เช่น Integer, String เป็นต้น)
2. ใช้วิธีการ push เพื่อแทรกองค์ประกอบลงในสแต็ก: stack.push(องค์ประกอบ); โดยที่องค์ประกอบคือองค์ประกอบที่จะแทรก
3. วิธีการ push จะเพิ่มองค์ประกอบที่ด้านบนสุดของสแต็กและทำให้เป็นองค์ประกอบบนสุดใหม่

### 3.19. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง Pop ในโครงสร้างข้อมูลแบบ Stack

1. ตรวจสอบว่าสแต็กว่างหรือไม่ ถ้าใช่ ให้ส่งคืนข้อความแสดงข้อผิดพลาด (เช่น "Stack underflow")
  2. เก็บองค์ประกอบบนสุดของสแต็กไว้ในตัวแปรชั่วคราว
  3. ตั้งค่าองค์ประกอบบนสุดของสแต็กเป็นองค์ประกอบถัดไป
  4. ลดขนาดของสแต็กลง 1
  5. ส่งคืนองค์ประกอบที่ถูกลบออกจากสแต็ก
- ในโครงสร้างข้อมูลสแต็ก การดำเนินการป๊อปใช้เพื่อลบและส่งคืนองค์ประกอบบนสุดจากสแต็ก องค์ประกอบที่ถูกลบจะกลายเป็นองค์ประกอบบนสุดใหม่

### 3.20. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง isEmpty ในโครงสร้างข้อมูลแบบ Stack

สามารถใช้เมธอด isEmpty ของคลาส java.util.Stack เพื่อตรวจสอบว่าสแต็กว่างหรือไม่ เมธอด isEmpty คืนค่าจริงหากสแต็กว่าง และเป็นเท็จหากสแต็กไม่ว่าง

```
..... if (stack.isEmpty()) {  
.....     System.out.println("The stack is empty");  
..... } else {  
.....     System.out.println("The stack is not empty");  
..... }
```

### 3.21. อธิบายพร้อมยกตัวอย่างประกอบความหมายของคำว่า Stack overflow

.....  
สแต็กโอเวอร์โฟลว์เป็นเงื่อนไขที่เกิดขึ้นเมื่อโปรแกรม Java พยายามเพิ่มองค์ประกอบให้กับโครงสร้างข้อมูลสแต็กมากกว่าความจุสูงสุดของสแต็ก

.....  
อาจเกิดขึ้นได้หากขนาดสแต็กคงที่และจำนวนองค์ประกอบที่เพิ่มในสแต็กเกินความจุสูงสุดของสแต็ก

```
.....  
void push(int value) {  
  
    // ตรวจสอบว่าสแต็กเต็ม  
  
    if (top == stackSize - 1) {  
  
        System.out.println("Error: Stack overflow");  
  
        return;  
  
    }  
}
```

#### 4. ลำดับขั้นการปฏิบัติการ

4.1. จงแก้โจทย์ปัญหาดังต่อไปนี้

4.1.1. จงเขียนโปรแกรมเพื่อสุ่มค่าเข้าไปในอาร์เรย์ 1 มิติ ตามจำนวนค่าที่รับจากผู้ใช้ โดยค่าที่ถูกสุ่มจะต้องเป็นตัวเลขจำนวนเต็มที่อยู่ระหว่าง 0 ถึง 99 เท่านั้น

Test case 1	Test case 2
Please enter your random value : <b>8</b> ----- 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 : <b>12</b> ----- 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	โค้ดโปรแกรม
	<pre> 1 package lab3; 2 import java.util.Random; 3 import java.util.Scanner; 4 5 public class lab3_1 { 6     public static void main(String[] args) { 7         int numValues = getNumValuesFromUser(); 8         int[] values = generateRandomValues(numValues); 9         printValues(values); 10    } 11 12    public static int getNumValuesFromUser() { 13        Scanner scanner = new Scanner(System.in); 14        System.out.print("Please enter your random value: "); 15        return scanner.nextInt(); 16    } 17 18    public static int[] generateRandomValues(int numValues) { 19        Random random = new Random(); 20        int[] values = new int[numValues]; 21        for (int i = 0; i &lt; numValues; i++) { 22            values[i] = random.nextInt(100); 23        } 24        return values; 25    } 26 27    public static void printValues(int[] values) { 28        for (int i = 0; i &lt; values.length; i++) { 29            System.out.println("Array[" + i + "] = " + values[i]); 30        } 31    } </pre>

--	--

--	--

#### 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( ) ; เพื่อตรวจสอบข้อมูลที่อยู่บนสุด

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)

ผลงาน	โค้ดโปรแกรม
	<pre> 1 package lab3; 2 import java.util.*; 3 4 public class Stack { 5     private String[] stack; 6     private int size; 7     private int top; 8 9     public Stack(int size) { 10         this.size = size; 11         stack = new String[size]; 12         top = -1; 13     } 14 15     // Method to add a value to the stack 16     public void push(String value) { 17         if (top == size - 1) { 18             System.out.println("\n----  Stack: Overflow"); 19         } else { 20             top++; 21             stack[top] = value; 22             System.out.println("\nPush: " + stack[top]); 23         } 24     } 25 26     // Method to remove the top element from the stack 27     public String pop() { 28         if (top == -1) { 29             System.out.println("----  NULL"); 30             return null; 31         } else { 32             String value = stack[top]; 33             top--; 34             System.out.println("Pop"); 35             return value; 36         } 37     } 38 39     // Method to check if the stack is empty 40     public boolean isEmpty() { 41         System.out.println("\nisEmpty"); 42         if (top == -1) { 43             System.out.println("----  Yes"); 44             return true; 45         } else { 46             System.out.println("----  No"); 47             return false; 48         } 49     } 50 51     // Method to retrieve the top element of the stack 52     public String top() { 53         System.out.println("\nTop"); 54         if (top == -1) { 55             System.out.println("----  NULL"); 56             return null; 57         } else { 58             System.out.println("----  Top = " + stack[top]); 59             return stack[top]; 60         } 61     } 62 63     // Method to check the current stack size 64     public void checkStack() { 65         System.out.println("\nCheckStack"); 66         if (top == -1) { 67             System.out.println("----  Stack : NULL"); 68         } else { 69             for (int i = top; i &gt;= 0; i--) { 70                 System.out.printf(stack[i]); 71                 if (i &gt; 0) { 72                     System.out.print(", "); 73                 } 74             } 75         } 76     } 77 78     // Method to set the maximum size of the stack 79     public void setStackSize(int size) { 80         System.out.println("SetStackSize("+size+")"); 81         this.size = size; 82         stack = new String[size]; 83         top = -1; 84     } </pre>

```

86 public static void main(String[] args) {
87     Scanner sc = new Scanner(System.in);
88     Stack stack = new Stack(10); // Initialize stack with maximum size of 10
89
90     while (true) {
91         System.out.println("\nEnter a command (push, pop, isEmpty, top,"
92             * "checkStack, setStackSize, or exit): ");
93         String input = sc.nextLine();
94
95         switch (input) {
96             case "push":
97                 System.out.println("Enter a value to push onto the stack: ");
98                 String value = sc.nextLine();
99                 stack.push(value);
100                 break;
101             case "pop":
102                 stack.pop();
103                 break;
104             case "isEmpty":
105                 stack.isEmpty();
106                 break;
107             case "top":
108                 stack.top();
109                 break;
110             case "checkStack":
111                 stack.checkStack();
112                 break;
113             case "setStackSize":
114                 System.out.println("Enter a new stack size:");
115                 String sizeInput = sc.nextLine();
116                 int size = Integer.parseInt(sizeInput);
117                 stack.setStackSize(size);
118                 break;
119             case "exit":
120                 sc.close();
121                 return;
122             default:
123                 System.out.println("Invalid command. Please try again.");
124                 break;
125         }
126     }
127 }
128 }

```

--	--

## 5. สรุปผลการปฏิบัติการ

ใน Java อาร์เรย์และสตริงถูกใช้เพื่อจัดเก็บและจัดการกับข้อมูล และเมธอดถูกใช้เพื่อดำเนินการกับข้อมูลนั้น ในโปรแกรม Stack จะมีองค์ประกอบหลักๆ คือ push, pop, peek โดยใน lab นี้จะมีการเรียกใช้งาน package Stack เพื่อดำเนินการต่างๆในโปรแกรม ในโปรแกรมสุ่มค่าเข้าไปใน array มีการเรียกใช้งาน package Random เพื่อนำมาสุ่มค่าเมื่อทำการสุ่มค่าเสร็จจะสร้าง array เพื่อเก็บค่าที่สุ่ม หลังจากนั้นจะทำการ print ข้อมูลที่สุ่มออกมา

## 6. คำถามท้ายการทดลอง

### 6.1. ข้อควรระวังในการใช้งาน Array ในภาษาจาวาคืออะไร ?

ตรวจสอบความยาวของอาร์เรย์ก่อนเข้าถึงองค์ประกอบเสมอ การเข้าถึงองค์ประกอบที่ index ที่มากกว่าหรือเท่ากับความยาวของอาร์เรย์จะส่งผลให้เกิด `ArrayIndexOutOfBoundsException`, ตรวจสอบให้แน่ใจว่าได้เริ่มต้นอาร์เรย์ก่อนที่จะพยายามเข้าถึงองค์ประกอบต่างๆ หากพยายามเข้าถึงองค์ประกอบของอาร์เรย์ที่ไม่ได้กำหนดค่าเริ่มต้น จะได้รับ `NullPointerException`, หากต้องการจัดเก็บองค์ประกอบจำนวนมากในอาร์เรย์ ให้พิจารณาใช้ `ArrayList` แทน `ArrayList` เป็นอาร์เรย์ที่ปรับขนาดได้ หมายความว่าสามารถขยายหรือลดขนาดได้ตามต้องการเพื่อรองรับองค์ประกอบใหม่

### 6.2. ข้อควรระวังในการใช้งาน String ในภาษาจาวาคืออะไร ?

ตรวจสอบให้แน่ใจว่าได้เริ่มต้นตัวแปรสตริงก่อนที่จะพยายามดำเนินการกับตัวแปรนั้น หากพยายามดำเนินการกับตัวแปรสตริงที่ไม่ได้กำหนดค่าเริ่มต้น จะได้รับ `NullPointerException`, ระวังความแตกต่างระหว่างสตริงที่ไม่เปลี่ยนรูปและสตริงที่ไม่เปลี่ยนรูป อ็อบเจกต์สตริงไม่เปลี่ยนรูปแบบ หมายความว่าเมื่อสร้างขึ้นแล้ว ค่าของอ็อบเจกต์จะไม่สามารถเปลี่ยนแปลงได้ หากต้องการแก้ไขสตริง ควรใช้ `StringBuilder` หรือ `StringBuffer` แทน, ใช้ `equals()` เพื่อเปรียบเทียบเนื้อหาของวัตถุสตริงสองรายการ แทนที่จะใช้ตัวดำเนินการ `==` ตัวดำเนินการ `==` จะตรวจสอบความเท่าเทียมกันของการอ้างอิง หมายความว่ามันจะคืนค่าจริงก็ต่อเมื่อวัตถุทั้งสองเป็นอินสแตนซ์เดียวกัน ในทางกลับกัน `equals()` เปรียบเทียบเนื้อหาของวัตถุทั้งสอง

### 6.3. ฟังก์ชันในภาษาจาวาไม่สามารถใช้งานแบบ Pass by reference ในภาษาซีได้ คุณมีแนวทางการแก้ไขปัญหานี้ได้อย่างไร ?

การ Pass by reference จะต้องใช้ตัวแปร ref ในการส่งตัวแปรไปยัง function โดยตัวแปรที่ถูกส่งผ่านจะส่งเป็นอาร์กิวเมนต์โดยจะต้องกำหนด ref ก่อนที่จะส่งผ่านไปยังเมธอด

### 6.4. โครงสร้างข้อมูลแบบ Stack แตกต่างกับโครงสร้างข้อมูลแบบ Array อย่างไร ?

- สแต็คมีขนาดที่คงที่โดยต้องระบุจำนวนสูงสุดที่จะเก็บไว้ กลับกัน อาร์เรย์สามารถปรับขนาดได้แบบไดนามิก
- ในการแก้ไของค์ประกอบในสแต็คสามารถใช้การ push, pop ได้แต่ในอาร์เรย์จะต้องใช้ indices ในการแก้ไข

