

# ใบงานการทดลองที่ 3

## เรื่อง อาร์เรย์ สตริง และฟังก์ชัน ในภาษาจาวา

### 1. จุดประสงค์ทั่วไป

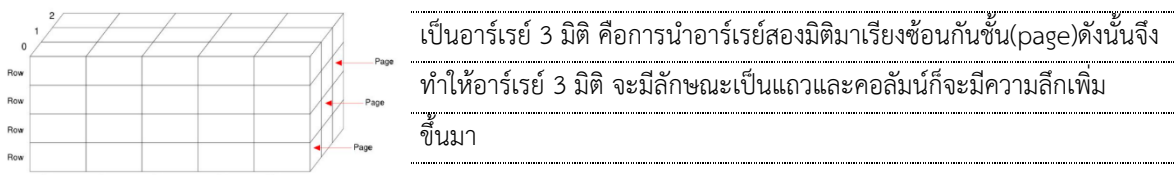
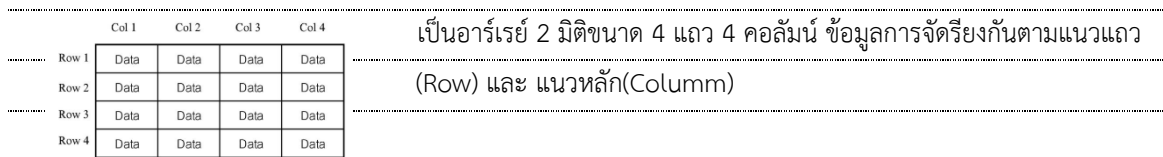
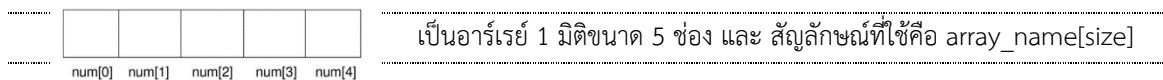
- 1.1. รู้และเข้าใจการโปรแกรมเชิงวัตถุร่วมกับอาร์เรย์และสตริง
- 1.2. รู้และเข้าใจการโปรแกรมเชิงวัตถุร่วมกับฟังก์ชัน

### 2. เครื่องมือและอุปกรณ์

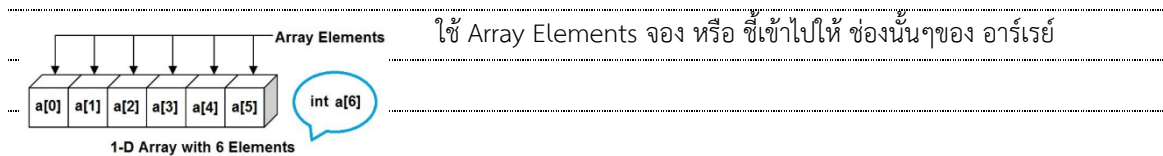
เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

### 3. ทฤษฎีการทดลอง

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



- 3.2. การเข้าถึงแต่ละ Element ของอาร์เรย์สามารถทำได้อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ



- 3.3. คำสั่ง length เกี่ยวข้องกับอาร์เรย์อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ
- เป็นคำสั่งที่สามารถหาขนาดของอาร์เรย์ได้

3.4. จงยกตัวอย่างประกอบในการวนรอบเพื่อแสดงค่าภายในตัวแปรอาเรย์ตั้งแต่ค่าแรกจนถึงค่าสุดท้าย

```
int[] cars = { 0, 1, 2, 3 };  
for( int i = 0 ; i < cars.length ; i++ ) {  
    System.out.print( cars[i] + " , " );  
}  
//end for
```

---

3.5. จงยกตัวอย่างการใช้งานคำสั่ง for each เพื่อแสดงค่าภายในตัวแปรอาเรย์

```
int Sum = 0 ;  
int[] a = { 91, 75, 43, 22, 18, 64 } ;  
for( int i : a ) {  
    Sum += i ;  
    System.out.print( i + " " );  
}  
//end foreach  
System.out.println( "" );  
System.out.println( "Sum = " + Sum );
```

---

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

ช่วยในการใช้ฟังก์ชันของอาเรย์ได้ง่ายขึ้นโดยไม่ต้องพิมพ์เอง

---

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

เปลี่ยนขนาดของอาเรย์ อย่างเช่น อาเรย์เดิมคือ 6 สามารถเปลี่ยนเป็น 20 ได้

---

3.8. จงยกตัวอย่างการประกาศ String และกำหนดค่าคำว่า “Hello World” ในภาษาจาวา

```
String e = "Hello World" ;  
System.out.println( e );
```

---

3.9. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toUpperCase() ในภาษาจาวา

```
String e = "Hello World" ;  
System.out.println( e.toUpperCase() );
```

---

3.10. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toLowerCase() ในภาษาจาวา

```
String e = "Hello World" ;  
System.out.println( e.toLowerCase() );
```

---

3.11. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง indexOf() ในภาษาจาวา

```
String e = "Hello World" ;  
String e = "Hello World" ;  
System.out.println( e.toLowerCase() );
```

---

```

System.out.println( e.toUpperCase() );
System.out.println( e.indexOf( "Sakkarin" ) );
if( e.indexOf( "ld" ) != -1 ) {
    System.out.println( e.indexOf( "ld" ) );
    System.out.println( "FOUND" );
} else {
    System.out.println( "NOT FOUND" );
}
} //end if

```

---

### 3.12. จงอธิบายความแตกต่างระหว่างการเชื่อม String แบบปกติและแบบใช้คำสั่ง concat( )

Concat() เป็นการนำ String มารวมกัน

---

### 3.13. หากต้องการแสดงสัญลักษณ์พิเศษภายในตัวแปร String ควรทำอย่างไร ?

ต้องใช้เครื่องหมาย backslash ก่อนค้อยตามด้วยสัญลักษณ์พิเศษ เช่น \', \", \\

---

### 3.14. จงอธิบายและยกตัวอย่างประกอบการสร้างฟังก์ชันในภาษาจาวา

ฟังก์ชันหรือเรียกอีกชื่อหนึ่งว่าเมทอด(method) มีความสำคัญต่อการเขียนโปรแกรมมากเพราะช่วยในการนำส่วนของโปรแกรมที่มีการทำงานซ้ำๆ แยกออกมาเป็นส่วนย่อยๆ ช่วยลดความซ้ำซ้อนของโค้ดภายในโปรแกรมและทำให้อ่านง่ายขึ้น

```

package HelloWorldProject;

public class lab2 {

    public static void main( String[] args ) {

        int a ;
        float b ;
        String c ;
        Test d = new Test();

        d.DodgerBlue();
        System.out.println( "Answer = " + d.Answer );

        d.Answer = 99 ;
        System.out.println( "Answer = " + d.Answer );

        f() ;
        abc() ;
    } //end method

    public static int f() {

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

```

---

```
        return 1 ;
    }//end function

    public static void abc() {
        System.out.println( "Hello World" );
    }//end function
}//end class
```

---

3.15. อธิบายข้อแตกต่างระหว่าง Pass by value และ Pass by reference

Pass by value เป็นการผ่านข้อมูล โดยผ่านการ return

Pass by reference มีในภาษานั้น คือการส่งผ่านข้อมูล ผ่านตัวแปรแบบ pointer และแบบ string

---

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

Void มี return

Int, double, float, string ไม่มี return

---

3.17. โครงสร้างข้อมูลแบบ Stack แตกต่างกับ Array อย่างไร ?

Array จะเหมือนกับตาราง ส่วน Stack จะเหมือนกับกล่อง

---

3.18. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง Push ในโครงสร้างข้อมูลแบบ Stack คือการเพิ่มข้อมูลเข้าไปใน stack เมื่อใน stack เต็ม แล้วเพิ่มข้อมูลเข้าไปจะทำให้เกิด stack overflow

---

3.19. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง Pop ในโครงสร้างข้อมูลแบบ Stack จะเป็นการนำเอาข้อมูลบนสุดออกก่อน ทำงานแบบ FILO

---

3.20. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง isEmpty ในโครงสร้างข้อมูลแบบ Stack isEmpty จะ return ค่าออกมาแค่ true, false เพื่อเช็คว่ามีข้อมูลอยู่ใน stack หรือไม่ ถ้ามีจะขึ้นว่า true ถ้าไม่มี จะ false

---

3.21. อธิบายพร้อมยกตัวอย่างประกอบความหมายของคำว่า Stack overflow เมื่อข้อมูลใน stack เต็มแล้วจะไม่สามารถ push ข้อมูลต่อได้

---

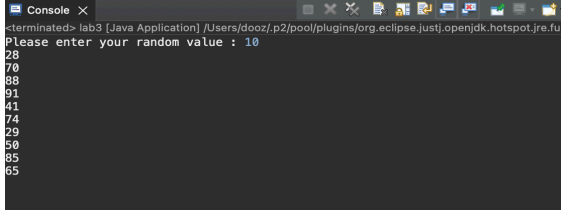
#### 4. ลำดับขั้นการปฏิบัติการ

##### 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 : 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. ฟังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามฟังงาน)

ฟังงาน	โค้ดโปรแกรม
	<pre>1 package lab3; 2 3 4 5 import java.util.Random; 6 import java.util.Scanner; 7 8 public class lab3 { 9     public static void main(String[] args) { 10         Scanner scanner = new Scanner(System.in); 11         System.out.print("Please enter your random value : "); 12         int input = scanner.nextInt(); // อ่านค่า 13         Random random = new Random(input); // create a Random object with 14         int numNumbers = input; // specify the number of random numbers to 15         for (int i = 0; i &lt; numNumbers; i++) { 16             int randomInt = random.nextInt(99) + 1; // generate a random num 17             System.out.println(randomInt); // print the random number 18         } 19     } 20 } 21 22 23 24</pre> 

--	--

--	--

#### 4.2. จงแก้ไขข้อผิดพลาดต่อไปนี้

##### 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. ผังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผังงาน)

ผังงาน	โค้ดโปรแกรม
	<pre> 1 package lab3; 2 import java.util.*; 3 import java.util.Scanner; 4 public class lab3_2 { 5     public static void main(String[] args) { 6         int n; 7         int i = 0; 8         String text; 9         System.out.print("Input Stack Size :"); 10        Scanner sg = new 11            Scanner(System.in); 12        n = sg.nextInt(); 13        Stack&lt;String&gt; stack = new 14            Stack&lt;String&gt;(); 15        stack.setSize(n); 16        stack.clear(); 17 18        do{ 19            System.out.println("-----"); 20            System.out.println("StackFn "); 21            System.out.println("1 :Push "); 22            System.out.println("2 :Pop "); 23            System.out.println("3 :isEmpty "); 24            System.out.println("4 :Top "); 25            System.out.println("5 :CheckStack "); 26            System.out.println("10 : END"); 27            System.out.print("Input"); 28 29            i = sg.nextInt(); 30            System.out.println(""); 31            System.out.println("-----"); 32            switch(i) { 33 34 35                case 1: 36                    System.out.print("Push : "); 37                    Scanner sct = new 38                        Scanner(System.in); 39                    text = sct.nextLine(); 40 41                    if(stack.size() == n) { 42                        System.out.println("---- STACK OVERFLOW!!!!!!"); 43                    }else { 44                        stack.push(text); 45                    } 46                    break; 47 48                case 2: 49                    if(stack.size() == 0) { 50                        System.out.println("----  STACK IS EMPTY"); 51                    }else { 52                        System.out.println("Pop"); 53                        stack.pop(); 54                    } 55                    break; 56 57                case 3: 58                    if(stack.isEmpty() == true) 59                    { 60                        System.out.println("----  Yes"); 61                    }else { 62                        System.out.println("----  No"); 63                    } 64                    break; 65 66                case 4: 67                    if(stack.size() == 0) { 68                        System.out.println("---- NULL"); 69                    }else { 70                        System.out.println("---- Top : "+stack.peek()); 71                    } 72                    break; 73 74                case 5: 75                    System.out.print("---- STACK : "+stack); 76 </pre>

--	--

--	--

## 5. สรุปผลการปฏิบัติการ

สามารถเขียนโปรแกรม การสุม โดยใช้ คำสั่ง Array และ stack

---

## 6. คำถามท้ายการทดลอง

6.1. ข้อควรระวังในการใช้งาน Array ในภาษาจาวาคืออะไร ?

เมื่อประกาศตัวแปรอาร์เรย์และกำหนดขนาดของอาร์เรย์แล้ว อาจจะเขียนหรืออ่านเกินกว่าสมาชิกของอาร์เรย์ได้ โดยคอมไพเลอร์จะไม่แจ้งเตือน แต่การเขียนข้อมูลเกินกว่าขนาดของอาร์เรย์ที่จองไว้จะทำให้ข้อมูลที่เขียนลงไปในนั้นจะไปทับข้อมูลอื่นหรือไปทับส่วนของโปรแกรม อาจจะทำให้การทำงานของโปรแกรมเสียหายได้

---

6.2. ข้อควรระวังในการใช้งาน String ในภาษาจาวาคืออะไร ?

ระวังเรื่องสัญลักษณ์พิเศษ

---

6.3. ฟังก์ชันในภาษาจาวาไม่สามารถใช้งานแบบ Pass by reference ในภาษาซีได้ คุณมีแนวทางการแก้ไขปัญหานี้ได้อย่างไร ?

ในภาษา java จะใช้ Pass by value

---

6.4. โครงสร้างข้อมูลแบบ Stack แตกต่างกับโครงสร้างข้อมูลแบบ Array อย่างไร ?

Array จะเหมือนกับตาราง ส่วน Stack จะเหมือนกับกล่อง

---