

ใบงานการทดลองที่ 3

เรื่อง อาร์เรย์ สตริง และฟังก์ชัน ในภาษาจาวา

1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการโปรแกรมเชิงวัตถุร่วมกับอาร์เรย์และสตริง
- 1.2. รู้และเข้าใจการโปรแกรมเชิงวัตถุร่วมกับฟังก์ชัน

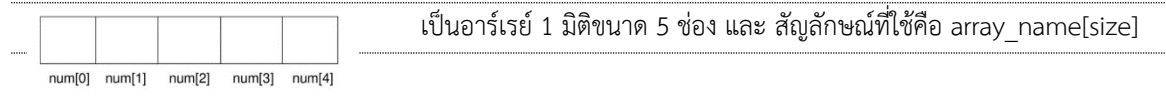
2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

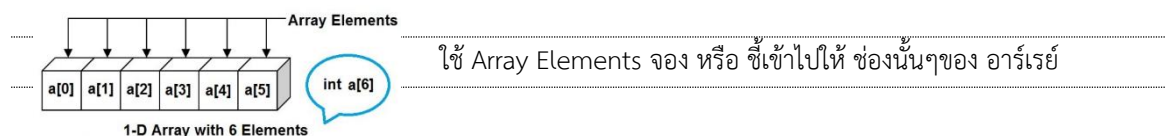
3. ทฤษฎีการทดลอง

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

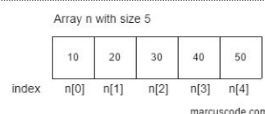
ประเภทของข้อมูลรูปแบบหนึ่งที่สามารถเก็บข้อมูลประเภทเดียวกันแบบเป็นลำดับได้ โดยข้อมูลนั้นจะอยู่ในตัวแปรตัวเดียวกันที่เรียกว่า ตัวแปรอาร์เรย์



- 3.2. การเข้าถึงแต่ละ Element ของอาร์เรย์สามารถทำได้อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ



- 3.3. คำสั่ง length เกี่ยวข้องกับอาร์เรย์อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ
เป็นคำสั่งที่สามารถหาขนาดของอาร์เรย์ได้



- 3.4. จงยกตัวอย่างประกอบในการวนรอบเพื่อแสดงค่าภายในตัวแปรอาร์เรย์ตั้งแต่ค่าแรกจนถึงค่าสุดท้าย

```
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 ตัวอย่างเช่น 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() );
แปลงสตริงตัวพิมพ์เล็กให้เป็นตัวอักษรพิมพ์ใหญ่ทั้งหมด
```

3.10. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toLowerCase() ในภาษาจาวา

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

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
ค้นหาข้อความ หรือตัวอักษร

```

3.12. จงอธิบายความแตกต่างระหว่างการเชื่อม String แบบปกติและแบบใช้คำสั่ง concat ()
Concat() เป็นการนำ String มารวมกัน

3.13. หากต้องการแสดงสัญลักษณ์พิเศษภายในตัวแปร String ควรทำอะไร ?
ต้องใช้เครื่องหมาย backslash ก่อนค้อยตามด้วยสัญลักษณ์พิเศษ เช่น \', \", \\

3.14. จงอธิบายและยกตัวอย่างประกอบการสร้างฟังก์ชันในภาษาจาวา
ฟังก์ชันหรือเรียกอีกชื่อหนึ่งว่าเมทอด(method) มีความสำคัญต่อการเขียนโปรแกรมมากเพราะช่วยในการนำส่วนของโปรแกรมที่มีการทำงานซ้ำๆ แยกออกมาเป็นส่วนย่อยๆ ช่วยลดความซ้ำซ้อนของโค้ดภายในโปรแกรม และทำให้อ่านง่ายด้วย

```

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 มีในภาษาซีเท่านั้น คือการส่งผ่านข้อมูล ผ่านตัวแปรแบบ pointer และแบบ string

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

Void จะมีการ return

Int, double, float, string จะไม่มี 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. ลำดับขั้นตอนการปฏิบัติการ

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. พังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามพังงาน)

พังงาน	โค้ดโปรแกรม
<pre> graph TD Start([start]) --> N[/int n/] N --> Ar[int[] ar = new int[n];] Ar --> I[/i = 0/] I --> Cond{i <= n} Cond -- True --> Rand[rand.nextInt(100);] Rand --> Inc[/i + 1/] Inc --> Cond Cond -- False --> Exit([exit]) </pre>	<pre> package lab3; import java.util.Scanner; import java.util.Random; public class lab3 { public static void main(String[] args) { System.out.print("Please enter your random value : "); int n; Scanner scanIn = new Scanner(System.in); n = scanIn.nextInt(); scanIn.close(); System.out.println(" "); System.out.println("-----"); int[] ar = new int[n]; Random rand = new Random(); for(int i = 0; i <= n ; i++) { ar[i]= rand.nextInt(100); System.out.println("Array[" + i + "] = "+ ar[i]); } } } </pre>

4.2. จงแก้ไข้ปัญหาดังต่อไปนี้

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> graph TD Start([start]) --> ReceiveStack[รับค่า stack] ReceiveStack --> IntN{int n} IntN --> IntI[int i = 0] IntI --> ReceiveValue[รับค่าที่ต้องการใส่] ReceiveValue --> Case1{case1} Case1 -- T --> N[n] N --> Overflow{ถ้า stack = n} Overflow -- T --> ReceiveOverflow[รับค่าจำนวน overflow] Overflow -- F --> DisplayOverflow[แสดงค่า overflow] Case1 -- F --> Case2{case2} Case2 -- T --> StackEmpty{ถ้า stack = 0} StackEmpty -- T --> DisplayEmpty[แสดงค่าที่ stack ว่าง] StackEmpty -- F --> PopValue[pop ค่าออก] Case2 -- F --> Case3{case3} Case3 -- T --> StackTrue{ถ้า stack = true} StackTrue -- T --> YES[YES] StackTrue -- F --> NO[NO] Case3 -- F --> Case4{case4} Case4 -- T --> StackZero{ถ้า stack = 0} StackZero -- T --> InputEmpty[ป้อนว่าง] StackZero -- F --> DisplayMax[แสดงค่าสูงสุดของ stack] Case4 -- F --> Case5{case5} Case5 -- T --> DisplayAll[แสดงค่า stack ทั้งหมด] Case5 -- F --> Case10{case10} Case10 -- T --> I10[i = 10] I10 --> ReceiveValue Case10 -- F --> ReceiveValue </pre>	<pre> package lab3; import java.util.*; import java.util.Scanner; public class lab3_1 { public static void main(String[] args) { int n; int i = 0; String text; System.out.println("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("-----"); System.out.println("StackFn"); System.out.println("-----"); System.out.println("1 :Push "); System.out.println("2 :Pop "); System.out.println("3 :isEmpty "); System.out.println("4 :Top "); System.out.println("5 :CheckStack "); System.out.println("10 : END"); System.out.println("-----"); System.out.print("Input"); i = sc.nextInt(); System.out.println(""); System.out.println("-----"); 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; case 2: if(stack.size() == 0) { System.out.println("---- STACK IS EMPTY"); }else { System.out.println("Pop"); stack.pop(); } break; case 3: if(stack.isEmpty() == true) { System.out.println("---- Yes"); }else { System.out.println("---- No"); } break; case 4: if(stack.size() == 0) { System.out.println("---- NULL"); }else { System.out.println("---- Top : "+stack.peek()); } break; case 5: System.out.println("---- STACK : "+stack); break; case 10: i = 10; break; } }while(i != 10); //endloop } //end main } </pre>

5. สรุปผลการปฏิบัติการ

สามารถเขียนโปรแกรม การสุ่ม โดยใช้ คำสั่ง Array และ Stack

6. คำถามท้ายการทดลอง

6.1. ข้อควรระวังในการใช้งาน Array ในภาษาจาวาคืออะไร ?

เมื่อประกาศตัวแปรอาร์เรย์และกำหนดขนาดของอาร์เรย์แล้ว อาจจะเขียนหรืออ่านเกินกว่าสมาชิกของอาร์เรย์ได้ โดยคอมไพเลอร์จะไม่แจ้งเตือน แต่การเขียนข้อมูลเกินกว่าขนาดของอาร์เรย์ที่จองไว้จะทำให้ข้อมูลที่เขียนลงไปในนั้นจะไปทับข้อมูลอื่นหรือไปทับส่วนของโปรแกรม อาจจะทำให้การทำงานของโปรแกรมเสียหายได้

6.2. ข้อควรระวังในการใช้งาน String ในภาษาจาวาคืออะไร ?

ระวังเรื่องสัญลักษณ์พิเศษต่างๆ

6.3. ฟังก์ชันในภาษาจาวาไม่สามารถใช้งานแบบ Pass by reference ในภาษาซีได้ คุณมีแนวทางการแก้ไขปัญหานี้ได้อย่างไร ?

ในภาษา java จะใช้ Pass by value การส่งค่า (value) เป็น argument ของฟังก์ชัน ดังนั้นค่าที่ทำในฟังก์ชันจึงไม่ส่งผลต่อตัวแปรนอกฟังก์ชัน

6.4. โครงสร้างข้อมูลแบบ Stack แตกต่างกับโครงสร้างข้อมูลแบบ Array อย่างไร ?

Stack - องค์ประกอบที่ใส่เข้าไปหลังสุดคือองค์ประกอบแรกที่จะออกมา การแทรกองค์ประกอบลงในสแต็กเรียกว่า การดำเนินการ PUSH และการลบองค์ประกอบออกจากสแต็กเรียกว่าการดำเนินการป๊อป

Array - ชุดของรายการที่จัดเก็บไว้ในตำแหน่งหน่วยความจำที่อยู่ติดกัน แนวคิดคือการจัดเก็บรายการประเภทเดียวกันหลายรายการไว้ด้วยกัน วิธีนี้ช่วยให้คำนวณตำแหน่งของแต่ละองค์ประกอบได้ง่ายขึ้นโดยการเพิ่มออฟเซตให้กับค่าพื้นฐาน
