

ใบงานการทดลองที่ 3

เรื่อง อาร์เรย์สตริง และฟังก์ชัน ในภาษาจาวา

1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการ โปรแกรมเชิงวัตถุร่วมกับอาร์เรย์และสตริง
- 1.2. รู้และเข้าใจการ โปรแกรมเชิงวัตถุร่วมกับฟังก์ชัน

2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์1 เครื่อง ที่ติดตั้ง โปรแกรม Eclipse

3. ทฤษฎีการทดลอง

3.1. โครงสร้างข้อมูลแบบ “อาร์เรย์” มีลักษณะเป็นอย่างไร ? มีองค์ประกอบอะไรบ้าง ? อธิบายพร้อมยกตัวอย่างประกอบ

= เป็นโครงสร้างข้อมูลที่มีการจองพื้นที่หน่วยความจำ เป็นชุด คล้ายรูปแบบของตารางแต่ละชุดจะประกอบไปด้วยจำนวนช่องข้อมูลหลายช่องซึ่งแต่ละช่องจะเก็บข้อมูลได้หน่วยอาจเป็นตัวเลขหรือตัวอักษรก็ได้ แต่พื้นที่แต่ละช่องข้อมูลจะเก็บข้อมูลชนิดเดียวกัน(เลขอยู่กับเลข อักษรอยู่กับอักษร) และอยู่ในตำแหน่งที่ต่อเนื่องกันไปตามลำดับ

โดยการสร้างจะใช้เป็น Array[row : column]

เช่น Number[1 : 5]

Number = { 1, 2, 3, 4, 5 }

Number	1	2	3	4	5
--------	---	---	---	---	---

3.2. การเข้าถึงแต่ละ Element ของอาร์เรย์สามารถทำได้อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ

= ข้อมูลแต่ละตัวของอาร์เรย์ จะเรียกว่า Element และข้อมูลแต่ละ Element จะมีหมายเลขเพื่อใช้ในการอ้างอิงถึงข้อมูลนั้น เรียกว่าเลข Index ซึ่งจะเป็นตัวแปรที่ชื่อ เหมือนกัน แต่จะแตกต่างกันตรงหมายเลข

เช่น Number	index	0	1	2	3	4
		1	2	3	4	5

จาก Array 1 มิติที่มี 5 element นั้น จะเห็นได้ว่าแต่ละ element จะมีเลข index กำกับเป็นตำแหน่งไว้ เหมือนกับเป็นเลขบัตรประจำตัวการเรียกใช้และเข้าถึง จะใช้เป็นการระบุตำแหน่งของ index

Number[1] ; //result จะเป็น เลข “2”

3.3. คำสั่ง length เกี่ยวข้องกับอาร์เรย์อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ

= คำสั่ง length เป็นการนับหรือหาจำนวน element ใน array ทั้งหมด ว่ามีกี่ตัว

0	1	2	3	4
1	2	3	4	5

จากภาพ เมื่อใช้คำสั่ง length จะได้ค่าเป็น 5 เพราะมีข้อมูลอยู่ 5 ตัวใน array

3.4. จงยกตัวอย่างประกอบในการวนรอบเพื่อแสดงค่าภายในตัวแปรอาร์เรย์ตั้งแต่ค่าแรกจนถึงค่าสุดท้าย

```
String[] animals = { "Dog", "Cat", "Tiger", "Lion", "Bear" } ;  
For ( int i = 0 ; i < animals.length ; i++ ) {  
    System.out.println( animals[i] ) ;  
}
```

Result จะได้ Dog Cat Tiger Lion Bear

3.5. จงยกตัวอย่างการใช้งานคำสั่ง for each เพื่อแสดงค่าภายในตัวแปรอาร์เรย์

= ใช้ loop for-each เช่น

```
String[] animals = { "Dog", "Cat", "Tiger", "Lion", "Bear" } ;  
For ( String i : animals ) {  
    System.out.println( i ) ;  
}
```

Result จะได้ Dog Cat Tiger Lion Bear

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

= เป็นการนำเข้าเพื่อใช้คำสั่งหรือการทำงานต่างๆที่เกี่ยวข้องกับ array

3.7. คำสั่ง Arrays.copyOf(____ , ____) ; มีหน้าที่ทำอะไร ?

= มีหน้าที่ในการ copy Array จากชุดหนึ่งไปยังอีกชุดหนึ่ง เป็นการ copy เพียงข้อมูลภายใน array ออกไป

3.8. จงยกตัวอย่างการประกาศ String และกำหนดค่าคำว่า "Hello World" ในภาษาจาวา

= String call = "Hello World" ;

3.9. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toUpperCase() ในภาษาจาวา

= เป็นการแปลงข้อความ String ให้อยู่ในรูปแบบของ ตัวอักษรพิมพ์ใหญ่ เช่น

```
String word = "Hello World" ;
```

```
System.out.print( word.toUpperCase() ) ;
```

Result : HELLO WORLD

3.10. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toLowerCase() ในภาษาจาวา

= เป็นการแปลงข้อความ String ให้อยู่ในรูปแบบของ ตัวอักษรพิมพ์เล็ก เช่น

```
String word = "Hello World" ;
```

```
System.out.print( word.toLowerCase() ) ;
```

Result : hello world

3.11. จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง indexOf() ในภาษาจาวา

= เป็นการหาตำแหน่งของข้อความที่ต้องการค้นหา

```
String a = new String("Welcome to MyLab") ;
```

```
System.out.println(a.indexOf("MyLab"));
```

Result : 11 (การนับจะรวมการเว้นวรรคด้วย)

3.12. จงอธิบายความแตกต่างระหว่างการเชื่อม String แบบปกติและแบบใช้คำสั่ง concat()

= การเชื่อม String แบบปกติจะใช้เครื่องหมาย + เพื่อให้ข้อความต่อกันคล้ายการเอามาปะ

concat()เป็นการรวมข้อความเข้าด้วยกัน หรือก็คือการนำ String มาเชื่อมต่อกันข้อความ

ประโยค

3.13. หากต้องการแสดงสัญลักษณ์พิเศษภายในตัวแปร String ควรทำอย่างไร ?

= ในการแสดงสัญลักษณ์พิเศษภายในตัวแปร String จะใช้เครื่องหมาย \ เพื่อให้ใช้ได้ โดยการนำไปใส่หน้าสัญลักษณ์พิเศษที่จะใช้

3.14. จงอธิบายและยกตัวอย่างประกอบการสร้างฟังก์ชันในภาษาจาวา

= Function ในภาษา Java นั้นคือ Method คือกลุ่ม process ที่สามารถเรียกใช้ได้

```
Public class Main {  
  
    Static void nyMethod() {  
  
        // code to process  
  
    }  
  
}
```

3.15. อธิบายข้อแตกต่างระหว่าง Pass by value และ Pass by reference

= Pass by Value คือ การส่งค่า (value) เป็น argument ของฟังก์ชัน ดังนั้นค่าที่ทำในฟังก์ชันจึงไม่ส่งผลต่อตัวแปรนอกฟังก์ชัน Pass by value คือ การส่งค่าแบบปกติ a = 5, a = "string" เป็นต้น

Pass by Reference คือ การส่งตัวแปร (variable) เป็น argument ของฟังก์ชัน ดังนั้นตัวแปรที่มีการดำเนินการใด ๆ ในฟังก์ชันจะส่งผลให้ตัวแปรนอกฟังก์ชันมีการเปลี่ยนแปลงด้วย Pass by reference คือ การส่งค่าโดยใช้ที่อยู่ (Address) หรือก็คือ ไม่ว่าเราจะเปลี่ยนแปลงอะไรกับตัวแปรที่มี reference เดียวกัน ก็จะทำให้ค่าเปลี่ยนแปลงตามกันไปหมด

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

= void เป็นการเรียกใช้โดยไม่ return ค่า แต่ int, double, float, string จะมีการ return ในบางกรณี

3.17. โครงสร้างข้อมูลแบบ Stack แตกต่างกับ Array อย่างไร ?

= Stack จะเป็นการเก็บข้อมูลแบบเรียงคิวเข้าไป หากจะเรียกใช้จะนำข้อมูลที่เข้ามาหลังสุด (Last-In First Out) ออกมาใช้ก่อน แต่ในส่วนของ Array จะเป็นการเก็บข้อมูลตามตำแหน่งที่วางอยู่ ซึ่งสามารถเลือกข้อมูลนำออกมาใช้ได้

3.18. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง Push ในโครงสร้างข้อมูลแบบ

Stack

= คือการกระทำที่ส่วนบนของสแตค (Top) ซึ่งต้องมีการตรวจสอบก่อนว่าสแตคเต็มหรือไม่ เป็นการดำเนินการที่นำข้อมูลเข้าไปเก็บไว้ด้านบนสุดของสแตค (Top of the Stack) เรื่อย ๆ จนกว่าสแตคไม่สามารถนำข้อมูลเข้าไปเก็บได้จะเรียกว่า สแตคเต็ม (Stack Full)

3.19. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง Pop ในโครงสร้างข้อมูลแบบ

Stack

= คือการนำข้อมูลที่อยู่บนสุดออกมาก่อน แต่ก่อนที่จะดึงจะมีการตรวจสอบว่ากองซ้อนว่างหรือไม่ ถ้าว่างจะไม่สามารถนำข้อมูลออกได้ แสดงว่ากองซ้อนว่าง (Stack Empty) ถ้าไม่ว่างจะนำเอาข้อมูลออกแล้วเลื่อนตัวชี้ไปยังตำแหน่งถัดลงไป

3.20. อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง isEmpty ในโครงสร้างข้อมูลแบบ

Stack

= เป็นคำสั่งในการตรวจสอบว่าสแตค ว่างหรือไม่ ซึ่งผลลัพธ์จะออกมาในรูปแบบของ bool : 1(True), 0(False)

3.21. อธิบายพร้อมยกตัวอย่างประกอบความหมายของคำว่า Stack overflow

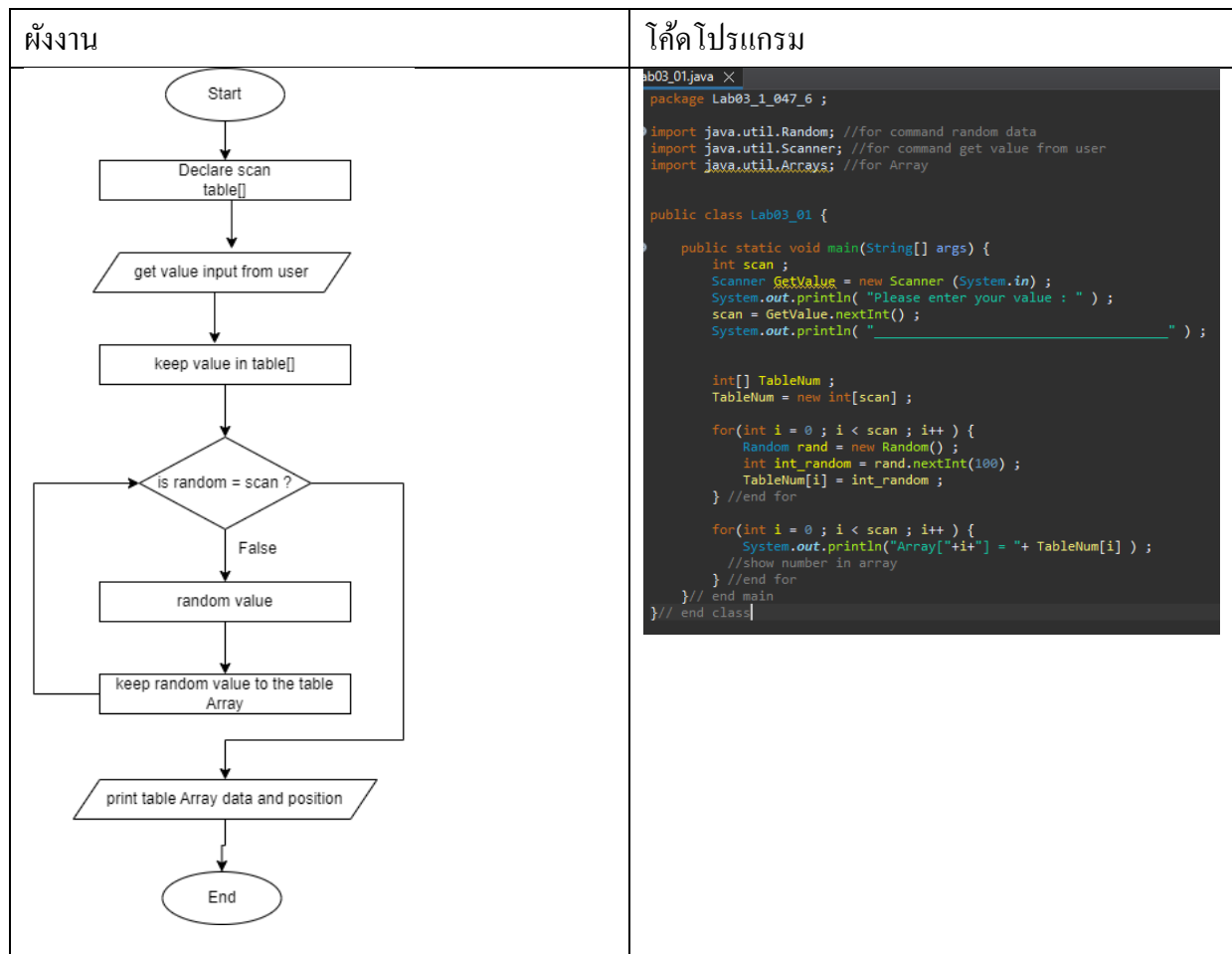
= Stack overflow คือการที่สแตคเต็มแล้ว แต่ยัง push ข้อมูลเข้าไปอีก ซึ่งไม่มีที่ว่างให้ใส่ข้อมูลเข้าไป จึงทำให้เกิดการล้นของสแตค นั่นคือ Stack overflow

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



4.2. จงแก้โจทย์ปัญหาดังต่อไปนี้

4.2.1. จงเขียนฟังก์ชันการจัดการโครงสร้างข้อมูลแบบ Stack พร้อมจำลองการทำงานโดยการเรียกใช้

คำสั่งพื้นฐานดังต่อไปนี้

คำสั่ง Push(String Value) ; เพื่อนำข้อมูลเข้าไปเก็บไว้ใน Stack

คำสั่ง Pop() ; เพื่อนำข้อมูลบนสุดออกจาก Stack

คำสั่ง isEmpty() ; เพื่อตรวจสอบข้อมูลใน Stack ว่ามีอยู่หรือไม่

คำสั่ง Top() ; เพื่อตรวจสอบข้อมูลที่อยู่มุมบนสุด

คำสั่ง CheckStack() ; เพื่อตรวจสอบค่าภายใน Stack ทั้งหมด

คำสั่ง SetStackSize(int size) ; เพื่อกำหนดขนาดเริ่มต้นของ Stack

ฟังก์ชัน	โค้ดโปรแกรม

5. สรุปผลการปฏิบัติการ

= Array เป็นโครงสร้างการเก็บข้อมูลที่มาช่วยสร้างความสะดวกในการจัดเก็บข้อมูลและการนำข้อมูลออกไปใช้ นอกจากนี้แล้วการจัดเก็บข้อมูลยังมีในรูปแบบอื่นอีก เช่น Stack ซึ่งในการเก็บข้อมูลจะมีคำสั่งในการจัดการภายในอีก เช่น การ push, pop, remove รวมไปถึงการเรียงข้อมูล

6. คำถามท้ายการทดลอง

6.1. ข้อควรระวังในการใช้งาน Array ในภาษาจาวาคืออะไร ?

= ในการจัดเก็บข้อมูลใน Array จะต้องใช้ข้อมูลชนิดเดียวกันเท่านั้น อักษรกับอักษร เลขกับเลข

และในการใช้index เพื่อหาตำแหน่งของข้อมูล indexจะเริ่มนับข้อมูลตัวแรกเป็นตำแหน่งที่ 0, 1, 2, 3, ...

6.2. ข้อควรระวังในการใช้งาน String ในภาษาจาวาคืออะไร ?

= เช่นเดียวกับArray คือการนับตำแหน่ง จะเริ่มที่0เช่นกัน แต่ String จะนับทุกอย่างเป็นตัวอักษร เลข
เว้นวรรค

6.3. ฟังก์ชันในภาษาจาวาไม่สามารถใช้งานแบบ Pass by reference ในภาษาซีได้คุณมีแนวทางการแก้ไขปัญหานี้ได้อย่างไร ?

= ต้องกำหนด argument ของฟังก์ชันเป็น pointer(*)

การเรียกใช้งานผ่าน Address เป็น argument ภายในฟังก์ชัน

การเรียกใช้ formal argument เวลาใช้งานจะใช้ dereferencing operator (*)dress)

6.4. โครงสร้างข้อมูลแบบ Stack แตกต่างกับโครงสร้างข้อมูลแบบ Array อย่างไร ?

= Stack จะเป็นการเก็บข้อมูลแบบเรียงคิวเข้าไป หากจะเรียกใช้จะนำข้อมูลที่เข้ามาหลังสุด (Last-In First Out) ออกมาใช้ก่อน แต่ในส่วนของ Array จะเป็นการเก็บข้อมูลตามตำแหน่งที่ว่างอยู่ ซึ่งสามารถเลือกข้อมูลนำออกมาใช้ได้