# ใบงานการทดลองที่ 3 เรื่อง อาเรย์ สตริง และฟังก์ชัน ในภาษาจาวา

## จุดประสงค์ทั่วไป

### รู้และเข้าใจการโปรแกรมเชิงวัตถุร่วมกับอาเรย์และสตริง

### รู้และเข้าใจการโปรแกรมเชิงวัตถุร่วมกับฟังก์ชัน

## เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

## ทฤษฎีการทดลอง

### โครงสร้างข้อมูลแบบ “อาเรย์” มีลักษณะเป็นอย่างไร ? มีองค์ประกอบอะไรบ้าง ? อธิบายพร้อมยกตัวอย่างประกอบ

|  |
| --- |
|  |
| เป็นอาร์เรย์ 1 มิติขนาด 5 ช่อง และ สัญลักษณ์ที่ใช้คือ array\_name[size] |
|  |
| เป็นอาร์เรย์ 2 มิติขนาด 4 แถว 4 คอลัมน์ ข้อมูลการจัดรียงกันตามแนวแถว |
| (Row) และ แนวหลัก(Columm) |
|  |
|  |
| เป็นอาร์เรย์ 3 มิติ คือการนำอาร์เรย์สองมิติมาเรียงซ้อนกันชั้น(page)ดังนั้นจึง |
| ทำให้อาร์เรย์ 3 มิติ จะมีลักษณะเป็นแถวและคอลัมน์ก็จะมีความลึกเพิ่ม |
| ขึ้นมา |
|  |

### การเข้าถึงแต่ละ Element ของอาเรย์สามารถทำได้อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ

|  |
| --- |
|  |
| ใช้ Array Elements จอง หรือ ชี้เข้าไปให้ ช่องนั้นๆของ อาร์เรย์ |
|  |
|  |

### คำสั่ง length เกี่ยวข้องกับอาเรย์อย่างไร ? อธิบายพร้อมยกตัวอย่างประกอบ

|  |
| --- |
| เป็นคำสั่งที่สามารถหาขนาดของอาเรย์ได้ |
|  |
|  |
|  |

### จงยกตัวอย่างประกอบในการวนรอบเพื่อแสดงค่าภายในตัวแปรอาเรย์ตั้งแต่ค่าแรกจนถึงค่าสุดท้าย

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

### 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 ); |

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

|  |
| --- |
| ช่วยในการใช้ฟังก์ชั่นของอาเรย์ได้ง่ายขึ้นโดยที่ไม่ต้องพิมพ์เอง |

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

|  |
| --- |
| เปลี่ยนขนาดของอาเรย์ อย่างเช่น อาเรย์เดิมคือ 6 สามารถเปลี่ยนเป็น 20 ได้ |

### จงยกตัวอย่างการประกาศ String และกำหนดค่าคำว่า “Hello World” ในภาษาจาวา

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

### จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toUpperCase() ในภาษาจาวา

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

### จงอธิบายและยกตัวอย่างประกอบการใช้งานคำสั่ง toLowerCase() ในภาษาจาวา

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

### 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 |

### จงอธิบายความแตกต่างระหว่างการเชื่อม String แบบปกติและแบบใช้คำสั่ง concat( )

|  |
| --- |
| Concat() เป็นการนำ String มารวมกัน |

### หากต้องการแสดงสัญลักษณ์พิเศษภายในตัวแปร String ควรทำอย่างไร ?

|  |
| --- |
| ต้องใช้เครื่องหมาย backslash ก่อนค่อยตามด้วยสัญลักษณ์พิเศษ เช่น \’, \”,\\ |

### จงอธิบายและยกตัวอย่างประกอบการสร้างฟังก์ชันในภาษาจาวา

|  |
| --- |
| ฟังก์ชันหรือเรียกอีกชื่อหนึ่งว่าเมท็อด(method) มีความสำคัญต่อการเขียนโปรแกรมมากเพราะช่วยในการนำส่วนของโปรแกรมที่มีการทำงานซ้ำๆ แยกออกมาเป็นส่วนย่อยๆ ช่วยลดความซ้ำซ้อนของโค้ดภายในโปรแกรมและทำให้อ่านง่ายด้วย  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 |

### อธิบายข้อแตกต่างระหว่าง Pass by value และ Pass by reference

|  |
| --- |
| Pass by value เป็นการผ่านข้อมูล โดยผ่านการ return  Pass by reference มีในภาซีเท่านั้น คือการส่งผ่านข้อมูล ผ่านตัวแปรแบบ pointer และแบบ string |

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

|  |
| --- |
| Void มี return  Int, double, float, string ไม่มี return |

### โครงสร้างข้อมูลแบบ Stack แตกต่างกับ Array อย่างไร ?

|  |
| --- |
| Array จะเหมือนกับตาราง ส่วน Stack จะเหมือนกับกล่อง |

### อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง Push ในโครงสร้างข้อมูลแบบ Stack

|  |
| --- |
| คือการเพิ่มข้อมูลเข้าไปใน stack เมื่อใน stack เต็ม แล้วเพิ่มข้อมูลเข้าไปจะทำให้เกิด stack overflow |

### อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง Pop ในโครงสร้างข้อมูลแบบ Stack

|  |
| --- |
| จะเป็นการนำเอาข้อมูลบนสุดออกก่อน ทำงานแบบ FILO |

### อธิบายพร้อมยกตัวอย่างประกอบกระบวนการทำงานของคำสั่ง isEmpty ในโครงสร้างข้อมูลแบบ Stack

|  |
| --- |
| isEmpry จะ retrun ค่าออกมาแค่ true, false เพื่อเช็กว่ามีข้อมูลอยู่ใน stack หรือไม่ ถ้ามีจะขึ้นว่า true ถ้าไม่มี จะ false |

### อธิบายพร้อมยกตัวอย่างประกอบความหมายของคำว่า Stack overflow

|  |
| --- |
| เมื่อข้อมูลใน stack เต็มแล้วจะไม่สามารถ push ข้อมูลต่อได้ |

## ลำดับขั้นการปฏิบัติการ

### จงแก้โจทย์ปัญหาดังต่อไปนี้

#### จงเขียนโปรแกรมเพื่อสุ่มค่าเข้าไปในอาเรย์ 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 |

#### ผังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผังงาน)

|  |  |
| --- | --- |
| ผังงาน | โค้ดโปรแกรม |
|  |  |

### จงแก้โจทย์ปัญหาดังต่อไปนี้

#### 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 |

#### ผังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผังงาน)

|  |  |
| --- | --- |
| ผังงาน | โค้ดโปรแกรม |
|  |  |

## สรุปผลการปฏิบัติการ

|  |
| --- |
| สามารถเขียนโปรแกรม การสุ่ม โดยใช้ คำสั่ง Array และ stack |

## คำถามท้ายการทดลอง

### ข้อควรระวังในการใช้งาน Array ในภาษาจาวาคืออะไร ?

|  |
| --- |
| เมื่อประกาศตัวแปรอาร์เรย์และกำหนดขนาดของอาร์เรย์แล้ว อาจจะเขียนหรืออ่านเกินกว่าสมาชิกของอาร์เรย์ได้ โดยคอมไพเลอร์จะไม่แจ้งเตือน แต่การเขียนข้อมูลเกินกว่าขนาดของอาร์เรย์ที่จองไว้จะทำให้ข้อมูลที่เขียนลงไปนั้นจะไปทับข้อมูลอื่นหรือไปทับส่วนของโปรแกรม อาจจะทำให้การทำงานของโปรแกรมเสียหายได้ |

### ข้อควรระวังในการใช้งาน String ในภาษาจาวาคืออะไร ?

|  |
| --- |
| ระวังเรื่องสัญลักษณ์พิเศษ |

### ฟังก์ชันในภาษาจาวาไม่สามารถใช้งานแบบ Pass by reference ในภาษาซีได้ คุณมีแนวทางการแก้ไขปัญหานี้ได้อย่างไร ?

|  |
| --- |
| ในภาษา java จะใช้ Pass by value |

### โครงสร้างข้อมูลแบบ Stack แตกต่างกับโครงสร้างข้อมูลแบบ Array อย่างไร ?

|  |
| --- |
| Array จะเหมือนกับตาราง ส่วน Stack จะเหมือนกับกล่อง |