

ใบงานการทดลองที่ 4

เรื่อง การกำหนดวัตถุ การใช้วัตถุ การสืบทอด และการห่อหุ้ม

1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจหลักการเขียนโปรแกรมเชิงวัตถุ คลาส การกำหนด และการใช้วัตถุ
- 1.2. รู้และเข้าใจหลักการสืบทอด และการห่อหุ้มวัตถุ

2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

3. ทฤษฎีการทดลอง

- 3.1. คลาสคืออะไร? มีลักษณะเด่นเป็นอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

คลาสคือชนิดหรือหมวดหมู่ เช่นผลไม้,รถ

- 3.2. วัตถุคืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือวัตถุแต่ละชนิดในหมวดหมู่นั้นๆ เช่นแอปเปิ้ล,กล้วย,มะม่วง หรือ Toyota,mazda,audi

- 3.3. คุณสมบัติ(Properties/Attributes) ควรมีลักษณะการประกาศอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

```
public class Main {  
    int x = 5;
```

- 3.4. การกระทำ/ฟังก์ชัน/เมธอด(Method) ควรมีลักษณะการประกาศอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

```
public static void main(String[] args) {  
    Main myObj = new Main();  
    System.out.println(myObj.x);  
}
```

- 3.5. เพราะเหตุใดจึงควรสร้าง 1 คลาสต่อ 1 ไฟล์?

เพราะใน.javaอนุญาตให้สร้าง1คลาสต่อ1ไฟล์

3.6. เมื่อสร้างวัตถุขึ้นมาแล้ว วัตถุจะสามารถอ้างอิง Properties หรือ Method ได้ด้วยวิธีการใด ?

อ้างอิงโดยใช้ **operator**

3.7. คำสั่ง **this** มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือการอ้างอิง **object** ปัจจุบัน

```
public class Main {
    int x;

    // Constructor with a parameter
    public Main(int x) {
        this.x = x;
    }

    // Call the constructor
    public static void main(String[] args) {
        Main myObj = new Main(5);
        System.out.println("Value of x = " + myObj.x);
    }
}
```

3.8. Constructor Method มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

Method ที่มีชื่อเดียวกับชื่อ Class โดยจะไม่มี Return หรือส่งค่าใด ๆ กลับออกไป โดยมีหน้าที่กำหนดค่าเริ่มต้นตอนสร้าง Class

```
public class Main {
    int modelYear;
    String modelName;

    public Main(int year, String name) {
        modelYear = year;
        modelName = name;
    }

    public static void main(String[] args) {
        Main myCar = new Main(1969, "Mustang");
        System.out.println(myCar.modelYear + " " + myCar.modelName);
    }
}

// Outputs 1969 Mustang
```

3.9. Destructor Method มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

method ที่สั่งให้ class ทำตามคำสั่งที่ method สั่งงานก่อนที่จะสิ้นสุดการทำงานของ class (สลาย class)

```
public class DestructorExample
{
    public static void main(String[] args)
    {
        DestructorExample de = new DestructorExample ();
        de.finalize();
        de = null;
        System.gc();
        System.out.println("Inside the main() method");
    }
    protected void finalize()
    {
        System.out.println("Object is destroyed by the Garbage Collector");
    }
}
```

```
Object is destroyed by the Garbage Collector
Inside the main() method
Object is destroyed by the Garbage Collector
```

3.10. การสืบทอด(Inheritance) คืออะไร? มีข้อดีและข้อเสียอย่างไร?

คือการสืบทอดจะได้ทั้ง attributes กับ method ข้อดีถ้าหากมี method หนึ่ง ใน super class sub class ก็จะสามารถเรียกนำมาใช้ได้เลย

3.11. จงยกตัวอย่างการสร้างคลาสรองเพื่อทำการสืบทอดจากคลาสหลัก

คลาสรองคือ sub class ที่สามารถสืบทอด method มาจากคลาสหลัก

3.12. จงยกตัวอย่างการสร้างวัตถุของคลาสหลักและคลาสรอง พร้อมกับยกตัวอย่างการเรียกใช้งานวัตถุในแต่ละคลาส เพื่อให้เห็นภาพการสืบทอดการทำงานซึ่งกันและกัน

```
class Vehicle {
    protected String brand = "Ford";           // Vehicle attribute
    public void honk() {                         // Vehicle method
        System.out.println("Tuut, tuut!");
    }
}

class Car extends Vehicle {
    private String modelName = "Mustang";       // Car attribute
    public static void main(String[] args) {

        // Create a myCar object
        Car myCar = new Car();

        // Call the honk() method (from the Vehicle class) on the myCar object
        myCar.honk();

        // Display the value of the brand attribute (from the Vehicle class) and the value of the modelName
        System.out.println(myCar.brand + " " + myCar.modelName);
    }
}
```

3.13. การควบคุมระดับการเข้าถึง(Access Modifier) ของตัวแปรแบบ Public, Protected และ Private คืออะไร ?

private ระดับการเข้าถึงของตัวดัดแปลงส่วนตัวนั้นอยู่ในคลาสเท่านั้น ไม่สามารถเข้าถึงได้จากภายนอกclass

public ระดับการเข้าถึงของตัวดัดแปลงสาธารณะมีอยู่ทั่วไป สามารถเข้าถึงได้จากภายในคลาส ภายนอกคลาส ภายในแพ็คเกจ และภายนอกแพ็คเกจ

protected ระดับการเข้าถึงของตัวดัดแปลงที่ได้รับการป้องกันนั้นอยู่ในแพ็คเกจและภายนอกแพ็คเกจผ่านคลาสย่อย หากคุณไม่ได้สร้างคลาสย่อย คลาสนั้นจะไม่สามารถเข้าถึงได้จากภายนอกแพ็คเกจ

3.14. การห่อหุ้ม(Encapsulation) คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

การกำหนดการเข้าถึงสมาชิกภายในClass ไม่ว่าภายนอกหรือภายในก็ตามจะถูกนำไปใช้เพื่อป้องกันข้อมูลภายในให้มีความปลอดภัยและเป็นความลับและง่ายต่อการเข้าใจในการเขียนโปรแกรม

```
// Java program to demonstrate encapsulation
class Encapsulate {
    // private variables declared
    // these can only be accessed by
    // public methods of class
    private String geekName;
    private int geekRoll;
    private int geekAge;

    // get method for age to access
    // private variable geekAge
    public int getAge() { return geekAge; }

    // get method for name to access
    // private variable geekName
    public String getName() { return geekName; }

    // get method for roll to access
    // private variable geekRoll
    public int getRoll() { return geekRoll; }

    // set method for age to access
    // private variable geekAge
    public void setAge(int newAge) { geekAge = newAge; }

    // set method for name to access
    // private variable geekName
    public void setName(String newName)
    {
        geekName = newName;
    }

    // set method for roll to access
    // private variable geekRoll
    public void setRoll(int newRoll) { geekRoll = newRoll; }
}

public class TestEncapsulation {
    public static void main(String[] args)
    {
        Encapsulate obj = new Encapsulate();

        // setting values of the variables
        obj.setName("Harsh");
        obj.setAge(19);
        obj.setRoll(51);

        // Displaying values of the variables
        System.out.println("Geek's name: " + obj.getName());
        System.out.println("Geek's age: " + obj.getAge());
        System.out.println("Geek's roll: " + obj.getRoll());

        // Direct access of geekRoll is not possible
        // due to encapsulation
        // System.out.println("Geek's roll: " +
        // obj.geekName);
    }
}
```

4. ลำดับขั้นการปฏิบัติการ

4.1. จงเขียนโปรแกรมสร้างคลาสในการจัดการอาเรย์ดังต่อไปนี้

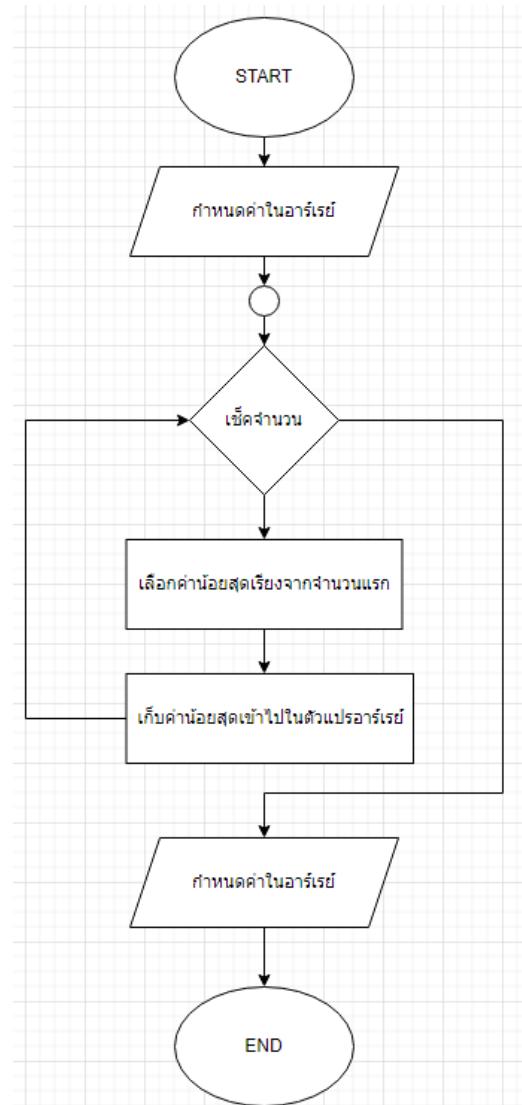
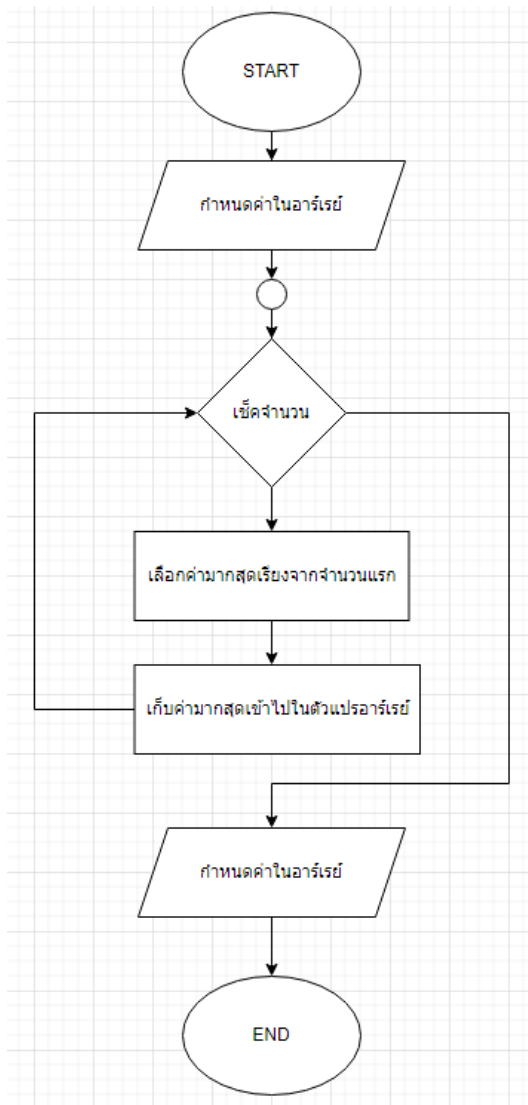
4.1.1. สร้างคลาสชื่อว่า MyClassicalArray

มี Properties ชื่อว่า MyArray[] พร้อมกับสุ่มค่าไว้ภายในตัวแปรทั้งหมด 10 ค่า

มี Method ชื่อว่า FindMax() ; เพื่อหาค่าที่มากที่สุดที่อยู่ในตัวแปร MyArray

มี Method ชื่อว่า FindMin() ; เพื่อหาค่าที่น้อยที่สุดในตัวแปร MyArray

```
class MyClassicalArray{  
  
    int [] number = {2,5,7,11,4,9,8,12,6,15};  
  
    public void FindMax() {  
        int max = number[0] ;  
        for (int i = 0 ; i < number.length ; i ++ ) {  
            if (number[i] < max)  
                max = number[i];  
        } // end for loop  
        System.out.println("Max Number is : " + max);  
    } // end Method FindMax  
  
    public void FindMin() {  
        int min = number[0] ;  
        for (int i = 0 ; i < number.length ; i ++ ) {  
            if (number[i] > min)  
                min = number[i];  
        } // end for loop  
        System.out.println("Min Number is : " + min);  
    } //end Method FindMin  
  
} // end class find
```

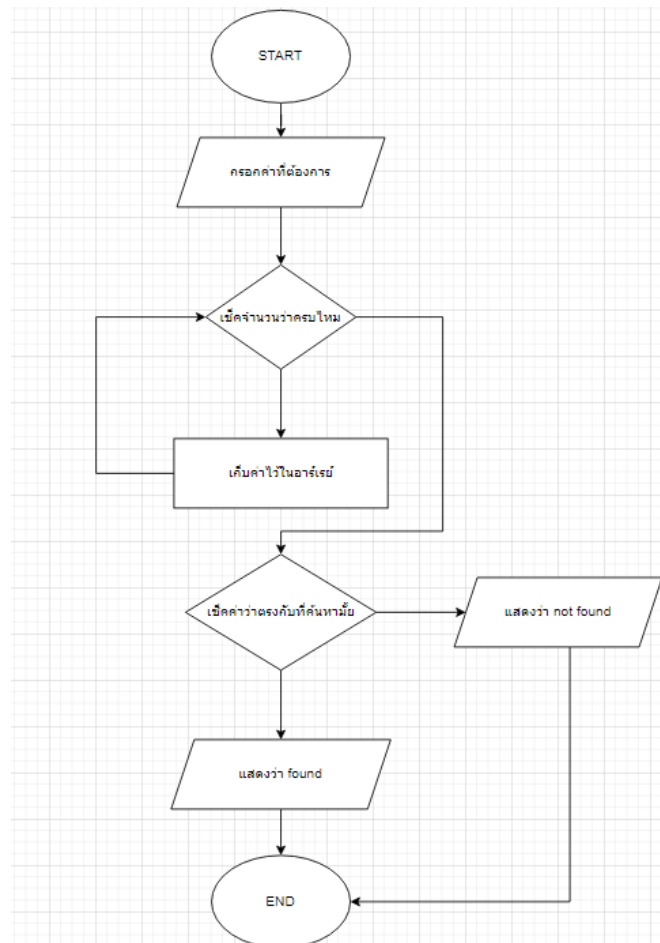
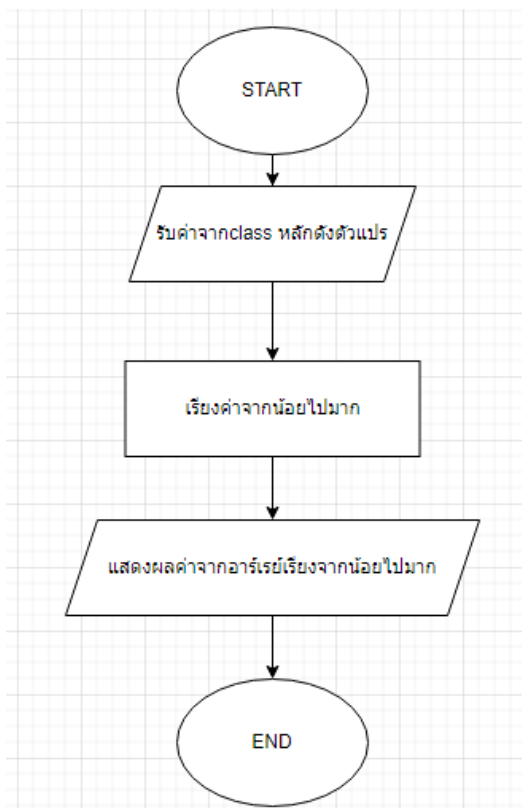


4.1.2. สร้างคลาสชื่อว่า MyCurrentArray ที่สืบทอดคลาส MyClassicalArray

มี Method ชื่อว่า Sort() ; เพื่อเรียงค่าภายในตัวแปร MyArray จากน้อยไปมาก

มี Method ชื่อว่า Search(Find) ; เพื่อค้นหาค่าที่อยู่ในตัวแปร MyArray

```
class MyCurrentArray extends MyClassicalArray{  
  
    public void Sort () {  
        Arrays.sort(number);  
        System.out.println("Number Max to Min is : " + Arrays.toString(number));  
    }// end Sort Method  
  
    public void Search () {  
        int scan = 5;  
        for (int i = 0 ; i < number.length ; i++) {  
            if(number[i] == scan){  
                System.out.println("Found");  
                break ;  
            }//end if  
        }//end loop  
    }// end Search Method  
}// end class MyCurrentArray
```



4.1.3. ในฟังก์ชันหลัก สร้างวัตถุจากคลาส MyClassicalArray ขึ้นมา และทดสอบการใช้งานคำสั่ง FindMax() ; และคำสั่ง FindMin() ;

4.1.4. ในฟังก์ชันหลัก สร้างวัตถุจากคลาส MyCurrentArray ขึ้นมา และทำการทดสอบการใช้งานคำสั่ง FindMax() ; คำสั่ง FindMin() ; คำสั่ง Sort() ; และคำสั่ง Search(Find) ;

```
switch(scan) {  
    case 1:  
        System.out.println("-----");  
        srcnum.FindMax();  
        System.out.println("-----");  
        break;  
    case 2:  
        System.out.println("-----");  
        srcnum.FindMin();  
        System.out.println("-----");  
        break;  
    case 3:  
        System.out.println("-----");  
        sort.Sort();  
        System.out.println("-----");  
        break;  
    case 4:  
        System.out.println("-----");  
        Search.Search();  
        System.out.println("-----");  
        break;  
}
```

```
1 : MAX , 2 : MIN , 3 : Sort , 4 : Search , 5 : End =  
1  
-----  
Max Number is : 2  
-----  
1 : MAX , 2 : MIN , 3 : Sort , 4 : Search , 5 : End =  
2  
-----  
Min Number is : 15  
-----  
1 : MAX , 2 : MIN , 3 : Sort , 4 : Search , 5 : End =  
3  
-----  
Number Max to Min is : [2, 4, 5, 6, 7, 8, 9, 11, 12, 15]  
-----  
1 : MAX , 2 : MIN , 3 : Sort , 4 : Search , 5 : End =
```


5. สรุปผลการปฏิบัติการ

รู้และเข้าใจหลักการเขียนโปรแกรมเชิงวัตถุ คลาส การกำหนด และการใช้วัตถุ และเข้าใจหลักการสืบทอด และการห่อหุ้มวัตถุ

6. คำถามท้ายการทดลอง

6.1. การสืบทอดในภาษาจาวาสามารถทำได้โดยใช้คำสั่งใด ?

Extends

6.2. จงอธิบายข้อควรระวังในการใช้งาน public, private และ protected

เลือกใช้งานตามเหมาะสมเพราะถ้าไม่เหมาะกับการทำงานจะ error

6.3. วัตถุ และ คลาส มีความเหมือนหรือแตกต่างกันอย่างไร ?

คลาสจะเป็นชนิดส่วนวัตถุจะวัตถุในแต่ละชนิดนั้นๆ

6.4. ในฐานะที่เป็นผู้พัฒนาระบบ คุณจะเลือกใช้การสืบทอดคลาสเมื่อใด? เพราะเหตุใด ?

เมื่อต้องการเพิ่มข้อมูลจากตัวของคลาสหลัก