

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

- ประเภท หมวดหมู หรือกลุ่ม ที่บ่งบอกถึง วัตถุ ได้หลากหลายชนิด เช่น ผลไม้ ที่บ่งบอกถึง ส้ม แอปเปิ้ล
- a class is like an object constructor, or a blueprint for creating object.

3.2. วัตถุคืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

- สิ่งที่เราเรียกย่อในหมวดหมู่นั้น เช่น อาชีพ ย่อยออกมาจะมีหลายหลาย เช่น ช่าง ครู หมอ

3.3. คุณสมบัติ(Properties/Attributes) ควรมีลักษณะการประกาศอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

- ตัวแปรที่อยู่ใน คลาส และมีลักษณะที่เกี่ยวข้องกับ object (วัตถุ) หรือ วัตถุ ทำอะไรได้บ้าง ? เช่น

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

- การสร้างเมธอดจะสามารถทำให้เรากระทำนั้นซ้ำ ๆ โดยที่ไม่ต้องเขียนโปรแกรมใหม่ ขึ้นกับขอบเขตและระดับการ

เข้าถึงที่ได้กำหนดขึ้น เช่น ใช้ในการสุ่มเลขจำนวนหลาย ๆ ชุด

3.5. เพราะเหตุใดจึงควรสร้าง 1 คลาสต่อ 1 ไฟล์?

- การเขียนโปรแกรมที่ดีของ Java เมื่อต้องการ หา class ที่ต้องการเราสามารถรู้ได้ทันทีด้วยชื่อไฟล์

3.6. เมื่อสร้างวัตถุขึ้นมาแล้ว วัตถุจะสามารถอ้างถึง Properties หรือ Method ได้ด้วยวิธีการใด ?

- สร้างวัตถุขึ้นมาก่อน แล้วอ้างอิงด้วยการประกาศ ชื่อวัตถุ ตามด้วยจุด (.) แล้วตามด้วย Properties หรือ Method
- Main myObj = new main();
- System.out.print(myObj.x);

3.7. คำสั่ง this มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

- เป็นคำสั่งสำหรับการอ้างถึงสมาชิกใน object ปัจจุบัน สำหรับตัวอย่างใน Constructor นั้น เราได้ใช้ this.name นั้น หมายถึง name ของ object ในขณะที่ name นั้นจะเป็นของเมธอด ถ้าหากในขอบเขตของโปรแกรมที่คุณใช้นั้นไม่มีชื่อของตัวแปรที่ซ้ำกันคุณไม่จำเป็นต้องใช้คำสั่ง this ก็ได้

3.8. Constructor Method มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

- เมธอดนี้จะเริ่มทำงานจะเริ่มต้นการทำงาน เมื่อทำการสร้าง object มาทำการใช้งาน เป็นเมธอดนี้ใช้กำหนดให้ค่าต่างๆ ให้ object และเป็นเมธอดที่ต้องมีชื่อเดียวกันกับคลาสเท่านั้น
- เช่น เมื่อสร้าง object Constructor Method จะเริ่มทำงาน และค่าที่ถูกกำหนดไว้ จะถูกส่งไปให้ object ที่สร้าง

3.9. Destructor Method มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

- เมธอดนี้จะเริ่มทำงานจะเริ่มต้นการทำงาน ก่อนที่จะลบ object มีวิธีการสร้างเหมือนกับ Constructor Method

3.10. การสืบทอด(Inheritance) คืออะไร? มีข้อดีและข้อเสียอย่างไร?

- คลาสลูก ที่อยู่ในคลาสแม่จะสามารถใช้งาน method และ attributes ของคลาสแม่ได้ทั้งหมด เรียกว่า การสืบทอด

- คำสั่งที่ใช้ extends

3.11. จงยกตัวอย่างการสร้างคลาสรองเพื่อทำการสืบทอดจากคลาสหลัก

- คลาสรองสร้างโดย ใช้คำสั่ง extends เช่น class Car extends Vehicle { } เพื่อให้สามารถเรียกใช้ method และ attributes ในคลาสแม่ได้

3.12. จงยกตัวอย่างการสร้างวัตถุของคลาสหลักและคลาสรอง พร้อมกับยกตัวอย่างการเรียกใช้งานวัตถุในแต่ละคลาส เพื่อให้เห็นภาพการสืบทอดการทำงานซึ่งกันและกัน

=====สร้างคลาสแม่=====

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

=====สร้างคลาสแม่=====

=====สร้างคลาสลูก=====

```
class Car extends Vehicle {  
    private String modelName = "Mustang"; // Car attribute  
    public static void main(String[] args) {  
        Car myCar = new Car //สร้างวัตถุ  
        myCar.honk(); //เรียกใช้ method และ attributes ในคลาสแม่  
        System.out.println(myCar.brand + " " + myCar.modelName);  
    }  
}
```

=====สร้างคลาสลูก=====

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

- Public คลาส มีขอบเขตการเข้าถึงได้ทุกระดับ สามารถเข้าถึงได้จากทุก Class
- Protected มีขอบเขตการเข้าถึงเฉพาะ Class ตัวเองกับ Class แม่ที่สืบทอดมาเท่านั้น
- Private คลาส มีขอบเขตการเข้าถึงเฉพาะภายใน Class เดียวกันเท่านั้น

3.14. การห่อหุ้ม(Encapsulation) คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

- คือ คุณสมบัติในการเขียนโปรแกรมเชิงวัตถุแล้วมีการกำหนดการเข้าถึงสมาชิกภายใน Class ไม่ว่าภายนอกหรือภายในก็ตามจะถูกนำไปใช้เพื่อป้องกันข้อมูลภายในให้มีความปลอดภัยและเป็นความลับและง่ายต่อการเข้าใจในการเขียนโปรแกรม

- ยกตัวอย่าง Encapsulation จากชีวิตความเป็นจริง เช่น เมื่อเราต้องการจะกดน้ำเปล่าจากเครื่องกรอกน้ำเราต้องใส่เหรียญ 1 2 5 และ 10 บาท นั้นเพื่อที่จะได้น้ำตามปริมาณตามที่เราต้องการและในการทำงานที่จะได้น้ำเปล่ามานั้นจะมีการทำงานที่อยู่ภายในเครื่องซ่อนอยู่โดยสิ่งที่ปกปิดเราจะเรียกว่า Encapsulation คือเราไม่จำเป็นต้องรู้วิธีการทำงานของเครื่องแต่สิ่งที่เราต้องรู้คือการใช้งานของเครื่องเพื่อที่จะได้น้ำเปล่าออกมา

- หรือ คุณสมบัติในการเขียนโปรแกรมวัตถุโดยการกำหนดการเข้าถึงสมาชิกภายในและการป้องกันข้อมูลภายในให้มีความปลอดภัยและยังเก็บเป็นความลับ โดยการทำงานนั้นออกแบบมาให้ง่ายต่อความเข้าใจในการเขียนโปรแกรม

4. ลำดับชั้นการปฏิบัติการ

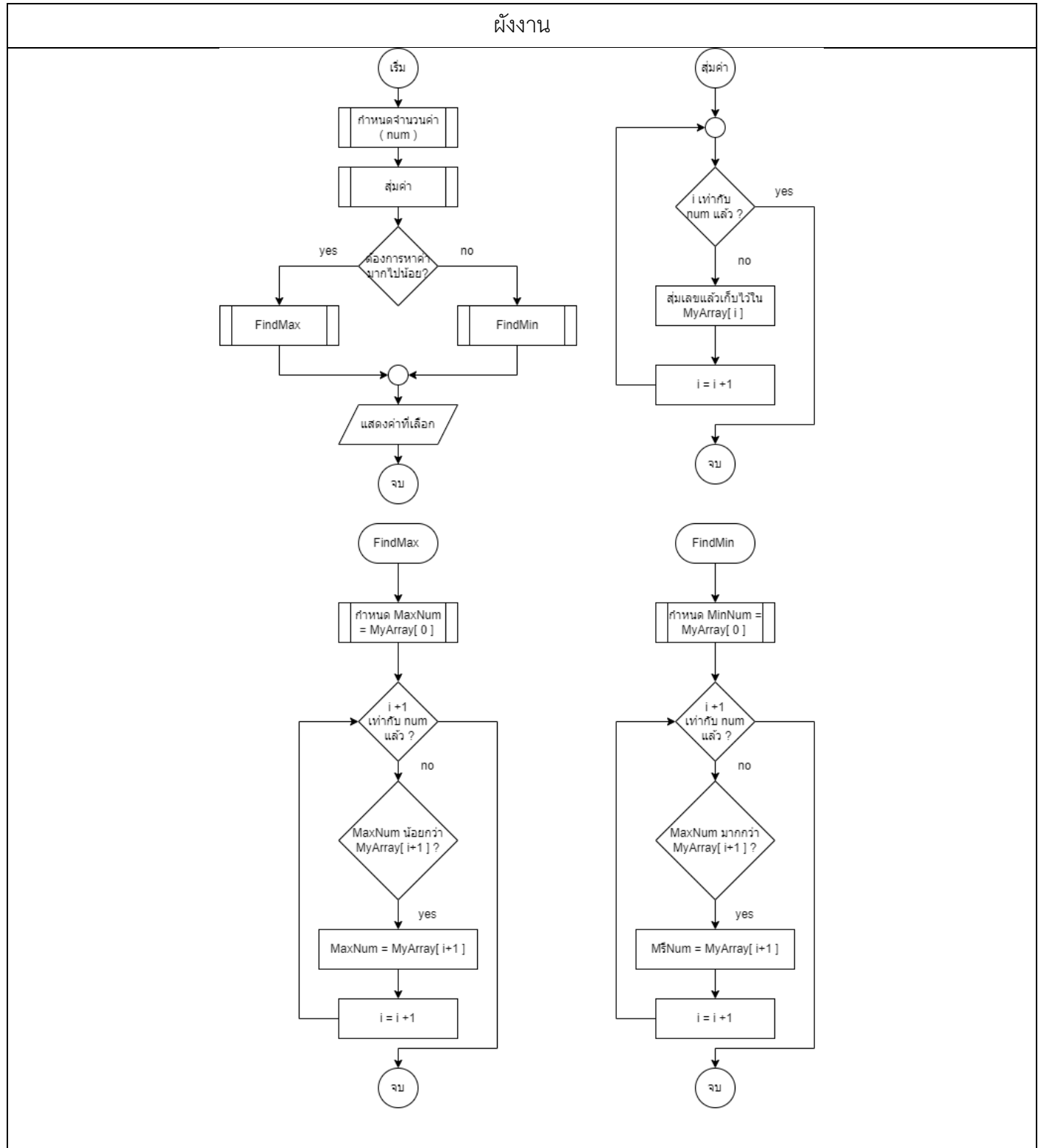
4.1. จงเขียนโปรแกรมสร้างคลาสในการจัดการอาเรย์ดังต่อไปนี้

4.1.1. สร้างคลาสชื่อว่า MyClassicalArray

มี Properties ชื่อว่า MyArray[] พร้อมกับสุ่มค่าไว้ภายในตัวแปรทั้งหมด 10 ค่า

มี Method ชื่อว่า FindMax() ; เพื่อหาค่าที่มากที่สุดที่อยู่ในตัวแปร MyArray

มี Method ชื่อว่า FindMin() ; เพื่อหาค่าที่น้อยที่สุดที่อยู่ในตัวแปร MyArra



โค้ดโปรแกรม

```
package OOPPackage;

public class MyClassicalArray {

    static void RandNum(int MyArray[]) {
        int num = MyArray.length;
        int min = 0; //lowest value of random number
        int max = 99; //highest value of random number
        for (int i = 0; i < num; i++) {
            int randnum = (int) Math.random() * max - min + 1 + min; //random number
            MyArray[i] = randnum; //input number to array
        } //end for

        for (int i = 0; i < num; i++) {
            System.out.print(" " + MyArray[i] + " "); //output all number in array
        } //end for
        System.out.println();
    } //end method

    static void FindMax(int MyArray[]) {
        int num = MyArray.length;
        int Max = MyArray[0];
        for (int i = 1; i < num; i++) {
            if (Max < MyArray[i]) {
                Max = MyArray[i]; //get greater number
            }
        } //end for
        System.out.println("FindMax = " + Max);
    } //end method

    static void FindMin(int MyArray[]) {
        int num = MyArray.length;
        int Min = MyArray[0];
        for (int i = 1; i < num; i++) {
            if (Min > MyArray[i]) {
                Min = MyArray[i]; //get lesser number
            }
        } //end for
        System.out.println("FindMin = " + Min);
    } //end method

    public static void main(String[] args) {

        int num = 10;
        int[] MyArray;
        MyArray = new int[num];

        RandNum(MyArray);
        FindMax(MyArray);
        FindMin(MyArray);

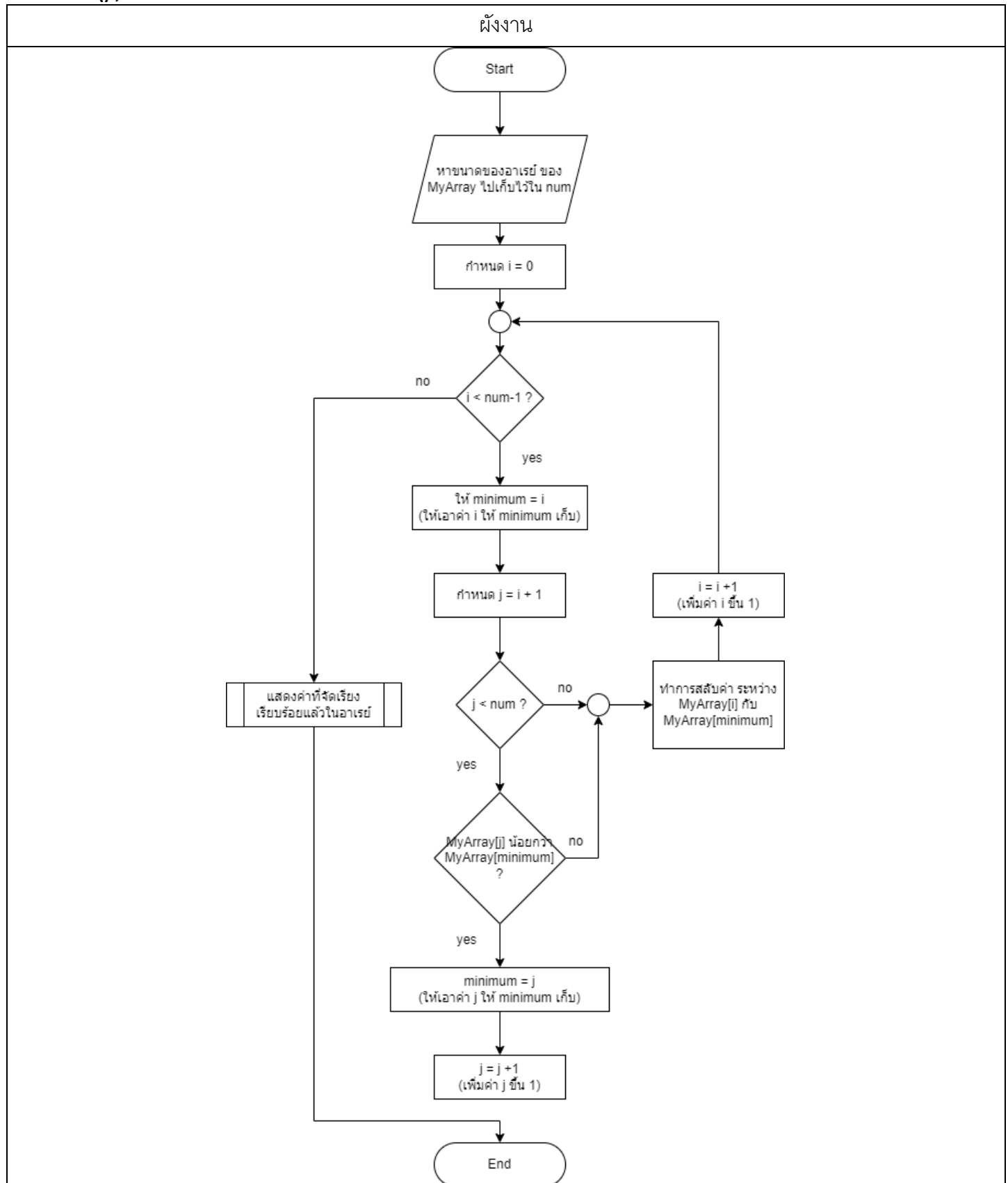
    } //end main method
} //class
```

4.1.2. สร้างคลาสชื่อว่า MyCurrentArray ที่สืบทอดคลาส MyClassicalArray

มี Method ชื่อว่า Sort() ; เพื่อเรียงค่าภายในตัวแปร MyArray จากน้อยไปมาก

มี Method ชื่อว่า Search(Find) ; เพื่อค้นหาค่าที่อยู่ในตัวแปร MyArray

Method : c() ;

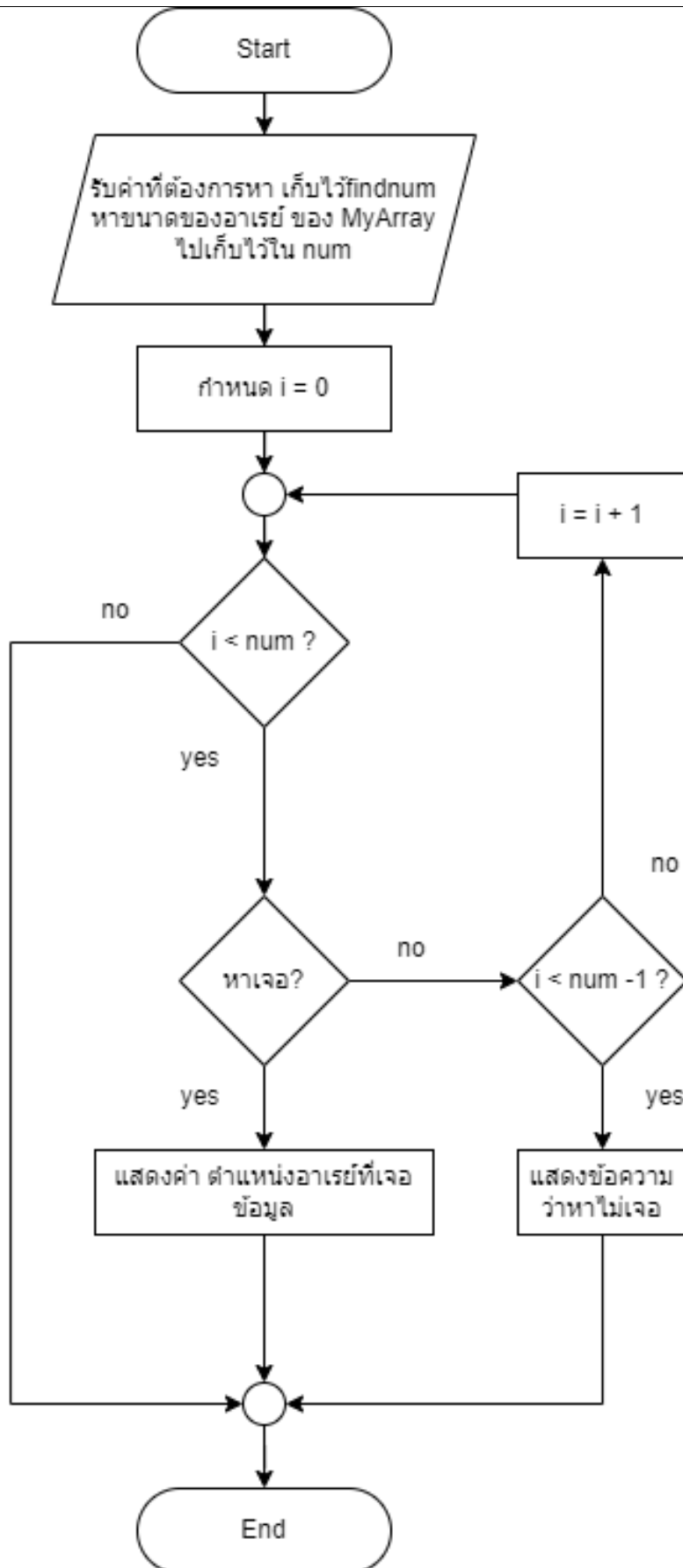


โค้ดโปรแกรม

```
static void Sort(int MyArray[]) { //Ascending
    int num = MyArray.length; //One by one move boundary of unsorted subarray
    for (int i = 0; i < num - 1; i++) { //Find the minimum element in unsorted array
        int minimum = i;
        for (int j = i + 1; j < num; j++)
            if (MyArray[j] < MyArray[minimum])
                minimum = j;
        //Swap the found minimum element with the first
        //element
        int temp = MyArray[minimum];
        MyArray[minimum] = MyArray[i];
        MyArray[i] = temp;
    } //end for
    PrintNum(MyArray);
} //end method
```

Method : Search(Find) ;

ผังงาน



โค้ดโปรแกรม

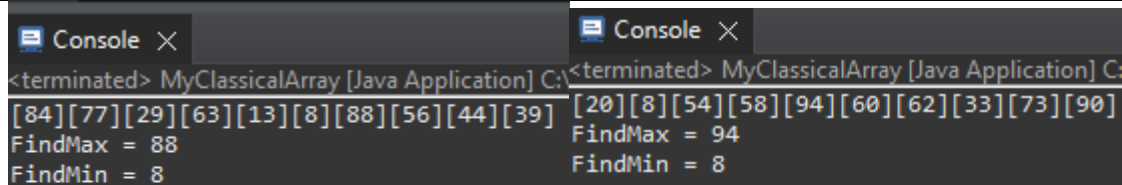
```
static void Search(int finenum, int MyArray[]) {
    int num = MyArray.length;
    for (int i = 0; i < num; i++) {
        if (MyArray[i] == finenum) {
            System.out.println("Search(" + finenum + " Found MyArray[" + i + "]);
            System.out.println("Number " + (num + 1));
            break;
        } else if (i == num - 1) {
            System.out.println("Search(" + finenum + ") Not Found");
        }
    }
}
} //end method
```

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

โค้ดโปรแกรมภายในฟังก์ชันหลัก

```
public static void main(String[] args) {
    MyClassicalArray myObj = new MyClassicalArray();
    RandNum(myObj.MyArray);
    FindMax(myObj.MyArray);
    FindMin(myObj.MyArray);
} //end main method
```

ผลลัพธ์การทำงานของโปรแกรม



The image shows two side-by-side console windows. The left window displays the output for an array [84][77][29][63][13][8][88][56][44][39], showing FindMax = 88 and FindMin = 8. The right window displays the output for an array [20][8][54][58][94][60][62][33][73][90], showing FindMax = 94 and FindMin = 8.

Array	FindMax	FindMin
[84][77][29][63][13][8][88][56][44][39]	88	8
[20][8][54][58][94][60][62][33][73][90]	94	8

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

โค้ดโปรแกรมภายในฟังก์ชันหลัก

```
public static void main(String[] args){

    MyCurrentArray myObj1 = new MyCurrentArray();
    MyCurrentArray myObj2 = new MyCurrentArray();

    RandNum(myObj1.MyArray);
    Sort(myObj1.MyArray);
    FindMax(myObj1.MyArray);
    FindMin(myObj1.MyArray);
    Search(69, myObj1.MyArray);

    System.out.println();
    RandNum(myObj2.MyArray);
    Sort(myObj2.MyArray);
    FindMax(myObj2.MyArray);
    FindMin(myObj2.MyArray);
    Search(69, myObj2.MyArray);

}/end main method
```

ผลลัพธ์การทำงานของโปรแกรม

Console	Console
<pre><terminated> MyCurrentArray [Java Application] C:\Users\parad\p [38][23][62][75][93][36][94][44][2][1] Sorting = [1][2][23][36][38][44][62][75][93][94] FindMax = 94 FindMin = 1 Search(69) Not Found [50][35][12][69][21][22][68][45][21][20] Sorting = [12][20][21][21][22][35][45][50][68][69] FindMax = 69 FindMin = 12 Search(69)Found MyArray[9] Number 11</pre>	<pre><terminated> MyCurrentArray [Java Application] C:\Users\parad\p [57][2][33][52][76][40][58][40][59][61] Sorting = [2][33][40][40][52][57][58][59][61][76] FindMax = 76 FindMin = 2 Search(69) Not Found [53][30][44][43][3][25][63][94][43][48] Sorting = [3][25][30][43][43][44][48][53][63][94] FindMax = 94 FindMin = 3 Search(69) Not Found</pre>

5. สรุปผลการปฏิบัติการ

- ได้ทดสอบความรู้และเข้าใจในหลักการของ คลาส การกำหนด การใช้วัตถุ หลักการสืบทอด และการห่อหุ้มวัตถุ ในการเขียนโปรแกรมเชิงวัตถุ ได้ดีขึ้น

6. คำถามท้ายการทดลอง

6.1. การสืบทอดในภาษาจาวาสามารถทำได้โดยใช้คำสั่งใด ?

- ใช้คำสั่ง extends

6.2. จงอธิบายข้อควรระวังในการใช้งาน public, private และ protected

- Public คลาส มีขอบเขตการเข้าถึงได้ทุกระดับ สามารถเข้าถึงได้จากทุก Class

- Protected มีขอบเขตการเข้าถึงเฉพาะ Class ตัวเองกับ Class แม่ที่สืบทอดมาเท่านั้น

- Private คลาส มีขอบเขตการเข้าถึงเฉพาะภายใน Class เดียวกันเท่านั้น

6.3. วัตถุ และ คลาส มีความเหมือนหรือแตกต่างกันอย่างไร ?

- วัตถุ คือตัวแปร ที่ระบุการกระทำ และคุณสมบัติ ของตัววัตถุเอง

- คลาส คือ ต้นแบบ หรือหมวดหมู่ ในการกำหนด การกระทำ และคุณสมบัติ ของวัตถุในคลาสนั้น

6.4. ในฐานะที่เป็นผู้พัฒนาระบบ คุณจะเลือกใช้การสืบทอดคลาสเมื่อใด? เพราะเหตุใด ?

- เมื่อคลาสที่เรา กำลังจะสร้าง จำเป็นต้องใช้ method ของคลาสที่เราเขียนไปแล้ว เพื่อไม่เราต้องเขียนซ้ำ เราจึงใช้การสืบทอดคลาส