

ใบงานการทดลองที่ 4

เรื่อง การกำหนดวัตถุ การใช้วัตถุ การสืบทอด และการห่อหุ้ม

1. จุดประสงค์ทั่วไป

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

2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

3. ทฤษฎีการทดลอง

- 3.1. คลาสคืออะไร? มีลักษณะเด่นเป็นอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

คลาสเป็นการกำหนดส่วนประกอบต่างๆ ที่จะนำไปสร้างออบเจ็ค คลาสจะประกอบไปด้วยสมาชิกสองอย่างคือ ตัวแปร และเมธอด ตัวแปรใช้สำหรับเก็บข้อมูลต่างๆ เกี่ยวกับออบเจ็ค และเมธอดเป็นการกำหนดฟังก์ชันการทำงานของออบเจ็ค นี่เป็นรูปแบบการประกาศคลาสในภาษา Java

```
//class
```

```
Public class Lab4 {
```

```
//Method
```

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

```
//attribute
```

```
Int i = 1;
```

```
}
```

```
}
```

- 3.2. วัตถุคืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

Object คือ วัตถุที่สร้างขึ้นมาจาก Class หรืออาจจะเรียก Object ได้อีกอย่างว่า instance (หรืออีกหลาย ๆ ชื่อที่คนใช้เรียกกัน) โดย Object จะสามารถนำไปใช้ดำเนินงานต่าง ๆ ได้ตามความสามารถที่ระบุไว้ใน Class

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

มี 2 รูปแบบ คือ

Global Attribute คือ คุณสมบัติของคลาสที่ประกาศนอก Method ซึ่งสามารถให้คลาสอื่น ๆ เข้าถึงได้ และสามารถนำไปใช้ในส่วนใดของคลาสก็ได้

Local Attribute คือ คุณสมบัติของคลาสที่ประกาศใน Method ซึ่งสามารถใช้งานได้เพียงใน Method ดังกล่าวเท่านั้น

- 3.4. การกระทำ/ฟังก์ชัน/เมธอด(Method) ควรมีลักษณะการประกาศอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ
- โดยการประกาศจะต้องมี public ขึ้นต้นนำหน้าก่อน จากนั้นก็ประกาศแบบปกติได้เลย
- 3.5. เพราะเหตุใดจึงควรสร้าง 1 คลาสต่อ 1 ไฟล์ ?
1. มันผิด syntax ของภาษา java
 2. แยกวัตถุเป็นก้อนๆ ไม่อยากให้ปนกัน
- 3.6. เมื่อสร้างวัตถุขึ้นมาแล้ว วัตถุจะสามารถอ้างถึง Properties หรือ Method ได้ด้วยวิธีการใด ?
- ใช้ Dot(.)
- 3.7. คำสั่ง this มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ
- คือตัวแทนของคลาส คลาสนี้ ใช้เพื่อเรียก ตัวแปร หรือ method ของ Class เราเอง
- 3.8. Constructor Method มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ
- Method ใดๆ ที่มีชื่อเดียวกับชื่อ Class ซึ่งเมื่อ Object ใดๆ ถูกสร้างขึ้นภายใต้ Class หนึ่ง โปรแกรมจะต้องไปเรียก ใช้ Method นี้ทันที เป็นฟังก์ชันการทำงานแรกเริ่มต้นเมื่อเริ่มสร้างวัตถุ
- 3.9. Destructor Method มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ
- จะทำงานเมื่อวัตถุ ถูกเลิกใช้งานแล้วหรือถูกทำลายไปแล้ว
- 3.10. การสืบทอด(Inheritance) คืออะไร? มีข้อดีและข้อเสียอย่างไร?
- การสืบทอดคุณสมบัติ และคุณลักษณะเช่น พ่อแม่ ถ่ายทอด DNA มาสู่ลูกเป็นต้น
- ข้อดี** เวลาสืบทอดไปคลาสลูกสามารถใช้ Properties และ Method ของคลาสแม่ได้หมด
- ข้อเสีย** ไม่สามารถเปลี่ยนแปลงฟังก์ชันได้
- 3.11. จงยกตัวอย่างการสร้างคลาสรองเพื่อทำการสืบทอดจากคลาสหลัก
- ```
Public class Monster extends Hero {
 Public Monster(String Name) {
 Super(Name);
 } //end method
} //class
```
- 3.12. จงยกตัวอย่างการสร้างวัตถุของคลาสหลักและคลาสรอง พร้อมกับยกตัวอย่างการเรียกใช้งานวัตถุในแต่ละคลาส เพื่อให้เห็นภาพการสืบทอดการทำงานซึ่งกันและกัน
- ```
Public class lab4_main {
    Public static void main(String[] args) {
        Hero a = new Hero();
        a.SetName("Thane");
        a.GetName();

        Monster b = new Monster();
        b.SetName("Dragon");
        b.GetName();
    } //end Method
} //end class
```

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

Access Modifiers เป็นคำสั่งในการควบคุมระดับการเข้าถึงของตัวแปรหรือเมธอดที่อยู่ภายในคลาส เหมือนที่คุณได้เห็นในการสร้างคลาสหลัก ชื่อของคลาสและไฟล์ต้องตรงกัน และต้องกำหนดการเข้าถึงให้เป็นแบบ public ซึ่งทำให้คลาสนี้สามารถเข้าถึงได้จากที่ทุกของโปรแกรม นอกจากนี้ Access Modifiers ยังเป็นคำสั่งที่ใช้ในการกำหนดการเข้าถึงของออบเจ็กต์ต่างๆ ใน Package เช่น คลาส และ Interfaces เป็นต้น

แบ่งเป็น 3 ประเภท ได้แก่

private เข้าถึงได้เฉพาะภายใน Class นี้เท่านั้น

protected เข้าถึงได้เฉพาะภายใน Class นี้ และภายใน Subclass ที่ extends หรือสืบทอดสมาชิกจาก Class นี้เท่านั้น public เข้าถึงได้ทั้งจากภายใน Class, Subclass และจากภายนอก Class

3.14. การห่อหุ้ม(Encapsulation) คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือการปกปิดหรือควบคุมการเข้าถึงข้อมูลของออบเจ็กต์จากภายนอก ซึ่งวิธีการที่ใช้ในการควบคุมคือการทำงานผ่านเมธอดเพื่อดำเนินการกับข้อมูลนั้น ในภาษา Java และภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุภาษาอื่นๆ นั้นสามารถใช้คุณสมบัตินี้ในการปกปิดส่วนประกอบภายในคลาสได้

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() ; เพื่อหาค่าที่น้อยที่สุดที่อยู่ในตัวแปร MyArray

Method : FindMax() ;

ผังงาน	โค้ดโปรแกรม
<pre> graph TD Start([Start]) --> Init[Max = this.ar[0]] Init --> Cond1{i < this.ar.length?} Cond1 -- Yes --> Update[Max = this.ar[i]] Cond1 -- No --> Return[return Max] Update --> Cond2{i < this.ar.length?} Cond2 -- Yes --> Inc[i++] Cond2 -- No --> Return Inc --> Cond1 Return --> End([End]) </pre>	<pre> public int FindMax() { int Max = this.ar[0]; for(int i = 0; i <= 9 ; i++) { if(Max < this.ar[i]) { Max = this.ar[i]; } //end if } //end for return Max; } //End Method FindMax </pre>

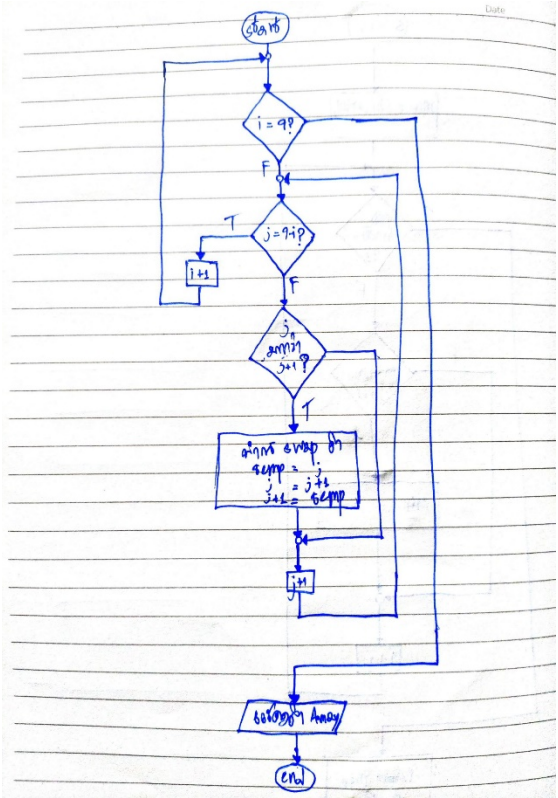
--	--

Method : FindMin() ;

ผังงาน	โค้ดโปรแกรม
<pre> graph TD Start([Start]) --> Init[Min = this.ar[0]] Init --> Cond1{i <= 9?} Cond1 -- T --> Cond2{this.ar[i] < Min?} Cond1 -- F --> Inc[i++] Cond2 -- T --> Assign[Min = i] Cond2 -- F --> Inc Assign --> Inc Inc --> Cond1 Cond1 -- F --> Ret[return Min] Ret --> End([End]) </pre>	<pre> public int FindMin() { int Min = this.ar[0]; for(int i = 0; i <= 9 ; i++) { if(Min > this.ar[i]) { Min = this.ar[i]; } //end if } //end for return Min; } //End Method FindMin </pre>

- 4.1.2. สร้างคลาสชื่อว่า MyCurrentArray ที่สืบทอดคลาส MyClassicalArray
 มี Method ชื่อว่า Sort() ; เพื่อเรียงค่าภายในตัวแปร MyArray จากน้อยไปมาก
 มี Method ชื่อว่า Search(Find) ; เพื่อค้นหาค่าที่อยู่ในตัวแปร MyArray

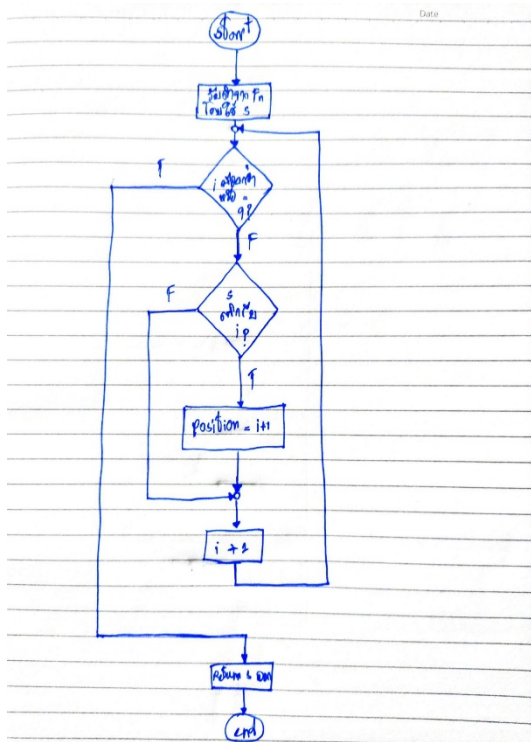
Method : Sort();

ผังงาน	โค้ดโปรแกรม
 <pre> graph TD Start([Start]) --> I9{i = 9?} I9 -- F --> J9i{j = 9-i?} J9i -- T --> Iplus[i++] Iplus --> I9 J9i -- F --> Compare{this.ar[j] > this.ar[j+1]?} Compare -- T --> Swap[Swap: temp = this.ar[j], this.ar[j] = this.ar[j+1], this.ar[j+1] = temp] Swap --> Jplus[j++] Jplus --> J9i Compare -- F --> Iplus Iplus --> I9 J9i -- F --> PrintHeader[Print "Sort min -> max : "] PrintHeader --> PrintLoopStart PrintLoopStart --> PrintLoopEnd PrintLoopEnd --> PrintHeader PrintHeader --> PrintNewline[Print "\n"] PrintNewline --> End([End]) </pre>	<pre> public void Sort() { for (int i = 0; i < 9; i++){ for (int j = 0; j < 9-i; j++) { if (this.ar[j] > this.ar[j+1]) { int temp = this.ar[j]; this.ar[j] = this.ar[j+1]; this.ar[j+1] = temp; } } } System.out.print("Sort min -> max : "); for(int i = 0; i <= 9 ; i++) { System.out.print(" "+this.ar[i]+" "); } System.out.println(""); } //End Method Sor </pre>

--	--

Method : Search(Find) ;

ผังงาน	โค้ดโปรแกรม
	<pre> public int Search(int s) { int position = 0 ; for(int i = 0; i <= 9 ; i++) { if(s == this.ar[i]) position = i+1 }//end for return position; } //End Method Search </pre>



- 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) ;

```

package MyClassicalArray;
import java.util.Scanner;

public class MyClassicalArray {
    public static void main(String[] args) {
        MyClassicalArray mclassi = new MyClassicalArray();
        MyCurrentArray m = new MyCurrentArray() ;

        System.out.println("");
        System.out.println("MyClassicalArray");
        mclassi.GetArray();
        //Fine Maximum
        int Mac = mclassi.FindMax();
        System.out.println("Max is " + Mac);
        //Find Minimum
        int Mic = mclassi.FindMax();
        System.out.println("Min is " + Mic);

        System.out.println("");
        System.out.println("MyCurrentArray");
        m.GetArray();
        //Fine Maximum
        int Max = m.FindMax();
        System.out.println("Max is " + Max);
        //Find Minimum
        int Min = m.FindMax();
        System.out.println("Min is " + Min);
        //Sort Minimum -> Maximum
        m.Sort();
        // Search Position number
        System.out.print("What search position number : ");
        Scanner sc = new Scanner(System.in);
        int s = sc.nextInt();
        System.out.print("Position " + s + " is : " +
m.Search(s));

    } // End main

    private int FindMax() {
        // TODO Auto-generated method stub
        return 0;
    }

    private void GetArray() {
        // TODO Auto-generated method stub
    }
}

```


ผลลัพธ์การทำงานของโปรแกรม

```
MyClassicalArray
Array : 5 89 34 18 7 91 47 74 11 55
Max is 91
MIn is 5

MyCurrentArray
Array : 19 24 26 37 48 51 52 75 86 88
Array : 48 88 51 37 19 52 86 75 26 24
Max is 88
MIn is 19
Sort min -> max : 19 24 26 37 48 51 52 75 86 88
What search position number : 51
position 51 is : 6
```

5. สรุปผลการปฏิบัติการ

ภายในคลาสจะประกอบด้วย Property โดย property สามารถกำหนดชนิดได้คือ Public Protected และ Private ต่อมาคือ Method จะทำงานคล้ายๆกับ Function ของ ภาษา C การที่จะเรียนใช้ Class ในClass main จะต้องนำเข้า Class โดยการ Newclass n = new.Newclass(); แล้วเราสามารถใช้ Method ใน Class นั้นได้ ส่วน การสืบทอดคือการที่จะใช้ Property กับ Method ในคลาสแม่มาใช้ในคลาสลูกได้ ด้วยคำสั่ง extends

6. คำถามท้ายการทดลอง

6.1. การสืบทอดในภาษาจาวาสามารถทำได้โดยใช้คำสั่งใด ?

Extends

6.2. จงอธิบายข้อควรระวังในการใช้งาน public, private และ protected

Public class ทุกclass สามารถใช้ได้ หากclass นั้นมี property ชื่อเดียวกันอาจจะทำให้คำผิดพลาดได้ Private class ตัวเองเท่านั้นที่ใช้ได้ จะมีปัญหาเวลาต้องการใช้ property นี้ใน class อื่น Protected class ตัวเองกับclass ที่สืบทอดไปเท่านั้นที่ได้ จะมีปัญหาตอนคลาสที่ไม่ได้สืบทอดจะใช้

6.3. วัตถุ และ คลาส มีความเหมือนหรือแตกต่างกันอย่างไร ?

วัตถุคือ class ที่นำเข้ามาใน class main หรืออื่นๆ คลาส คือ Code ที่เหมือนกับ Structure ที่จะประกอบไปด้วย property กับ method โดยใช้เรียกใช้ได้ผ่านการเป็น วัตถุ

6.4. ในฐานะที่เป็นผู้พัฒนาระบบ คุณจะเลือกใช้การสืบทอดคลาสเมื่อใด? เพราะเหตุใด ?

เมื่อคลาสต้องการใช้ property กับ method ของคลาสนั้น เพราะจะได้ไม่ต้องทำ property กับ method ใหม่