

## ใบงานการทดลองที่ 4

### เรื่อง การกำหนดวัตถุ การใช้วัตถุ การสืบทอด และการห่อหุ้ม

#### 1. จุดประสงค์ทั่วไป

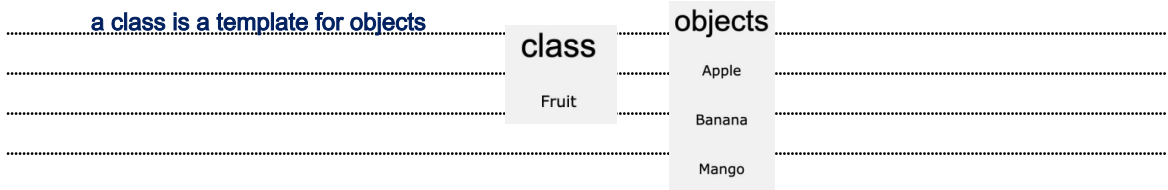
- 1.1. รู้และเข้าใจหลักการเขียนโปรแกรมเชิงวัตถุ คลาส การกำหนด และการใช้วัตถุ
- 1.2. รู้และเข้าใจหลักการสืบทอด และการห่อหุ้มวัตถุ

#### 2. เครื่องมือและอุปกรณ์

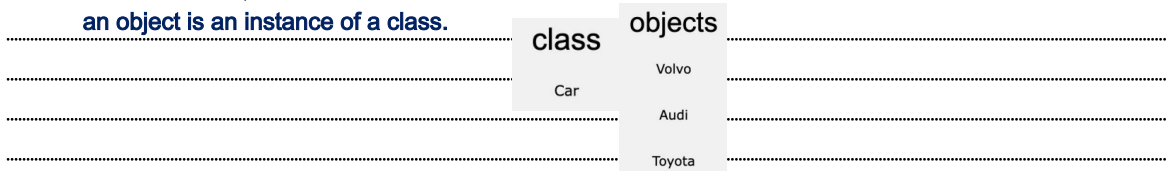
เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

#### 3. ทฤษฎีการทดลอง

- 3.1. คลาสคืออะไร? มีลักษณะเด่นเป็นอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ



- 3.2. วัตถุคืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ



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

มี 2 รูปแบบ คือ

- **Local Attribute** คือ คุณสมบัติของคลาสที่ประกาศใน Method ซึ่งสามารถใช้งานได้เพียงใน Method ดังกล่าวเท่านั้น
- **Global Attribute** คือ คุณสมบัติของคลาสที่ประกาศนอก Method ซึ่งสามารถให้คลาสอื่น ๆ เข้าถึงได้ และสามารถนำไปใช้ในส่วนใดของคลาสก็ได้

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

โดยการประกาศจะต้องมี **public** ขึ้นต้นนำหน้าก่อน จากนั้นก็ประกาศแบบปกติได้เลย

- 3.5. เพราะเหตุใดจึงควรสร้าง 1 คลาสต่อ 1 ไฟล์ ?

1. มันคือ **syntax** ของภาษา **java**

2. แยกวัตถุเป็นก้อนๆ ไม่ยากให้ปนกัน

- 3.6. เมื่อสร้างวัตถุขึ้นมาแล้ว วัตถุจะสามารถอ้างอิง Properties หรือ Method ได้ด้วยวิธีการใด ?

ใช้ **Dot(.)**

- 3.7. คำสั่ง **this** มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือตัวแทนของคลาส คลาสนี้ ใช้เพื่อเรียก ตัวแปร หรือ **method** ของ **Class** เราย่อย

3.8. Constructor Method มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

Method โดยที่ไม่มีชื่อเดียวกับชื่อ Class ซึ่งเมื่อ Object โดยถูกสร้างขึ้นมาจากได้ 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 มาสู่ลูก เป็นต้น

- ข้อดี เวลาสืบทอดไปคลาสลูกสามารถใช้ Properties และ Method ของคลาสแม่ได้หมด
- ข้อเสีย ไม่สามารถเปลี่ยนแปลงฟังก์ชันได้

3.11. จงยกตัวอย่างการสร้างคลาสรองเพื่อทำการสืบทอดจากคลาสหลัก

```
Public class Monster extends Hero {  
    Public Monster(String Name) {  
        Super(Name);  
    }  
}
```

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();  
  
    }  
}
```

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

Access Modifiers เป็นคำสั่งในการควบคุมระดับการเข้าถึงของตัวแปรหรือเมธอดที่อยู่ในคลาส เหมือนที่คุณได้เห็นในการสร้างคลาสหลัก ชื่อของคลาสและไฟล์ต้องตรงกัน และต้องกำหนดการเข้าถึงให้เป็นแบบ public ซึ่งทำให้คลาสนี้สามารถเข้าถึงได้จากที่ทุกของโปรแกรม นอกจากนี้ Access Modifiers ยังเป็นคำสั่งที่ใช้ในการกำหนดการเข้าถึงของออบเจกต์ต่างๆ 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 นี้ เท่านั้น public เข้าถึงได้ทั้งจากภายใน Class, Subclass และจากภายนอก Class

3.14. การห่อหุ้ม(Encapsulation) คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือการปิดหรือควบคุมการเข้าถึงข้อมูลของออบเจกต์จากภายนอก ซึ่งวิธีการที่ใช้ในการควบคุมคือการ ใช้งานผ่านเมธอดเพื่อดำเนินการกับข้อมูลนั้น ในภาษา Java และภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุภาษาอื่นๆ นั้นสามารถใช้คุณสมบัตินี้ในการปกปิดส่วนประกอบภายใน คลาสได้

#### 4. ลำดับขั้นการปฏิบัติการ

4.1. จงเขียนโปรแกรมสร้างคลาสในการจัดการอาร์เรย์ดังต่อไปนี้

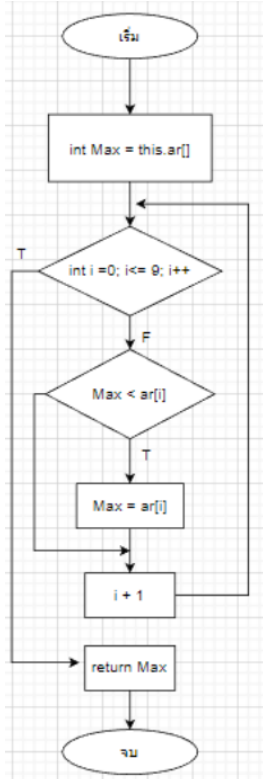
4.1.1. สร้างคลาสชื่อว่า MyClassicalArray

มี Properties ชื่อว่า MyArray[ ] พร้อมกับค่าไว้ภายในตัวแปรทั้งหมด 10 ค่า

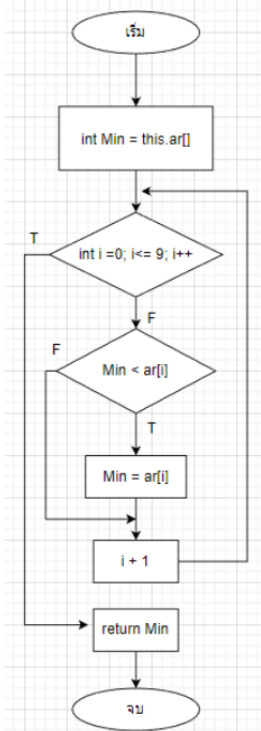
มี Method ชื่อว่า FindMax( ) ; เพื่อหาค่าที่มากที่สุดที่อยู่ในตัวแปร MyArray

มี Method ชื่อว่า FindMin( ) ; เพื่อหาค่าที่น้อยที่สุดที่อยู่ในตัวแปร MyArray

Method : FindMax();

ผังงาน	โค้ดโปรแกรม
	<pre>public int FindMax() {     int Max = this.ar[0];     for(int i = 0; i &lt;= 9 ; i++ ) {          if(Max &lt; this.ar[i]) {             Max = this.ar[i];         }     }     return Max; }</pre>

Method : FindMin() ;

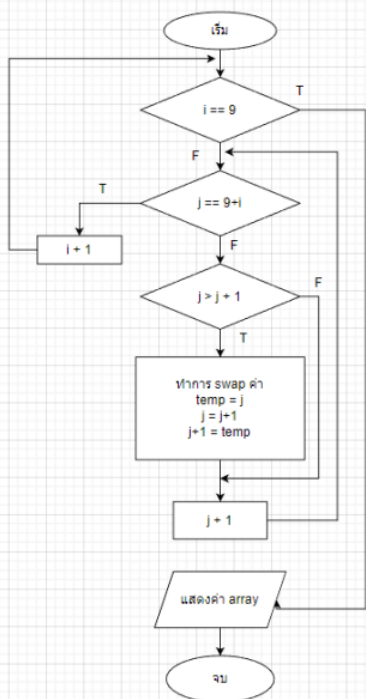
ผังงาน	โค้ดโปรแกรม
 <pre> graph TD     Start([เริ่ม]) --&gt; Init[int Min = this.ar[0]]     Init --&gt; LoopStart[int i = 0; i &lt;= 9; i++]     LoopStart -- T --&gt; Decision{Min &lt; ar[i]}     Decision -- F --&gt; LoopStart     Decision -- T --&gt; Assign[Min = ar[i]]     Assign --&gt; Increment[i + 1]     Increment --&gt; LoopStart     LoopStart -- F --&gt; Return[return Min]     Return --&gt; End([จบ])         </pre>	<pre> public int FindMin() {     int Min = this.ar[0];     for(int i = 0; i &lt;= 9; i++) {         if(Min &gt; 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

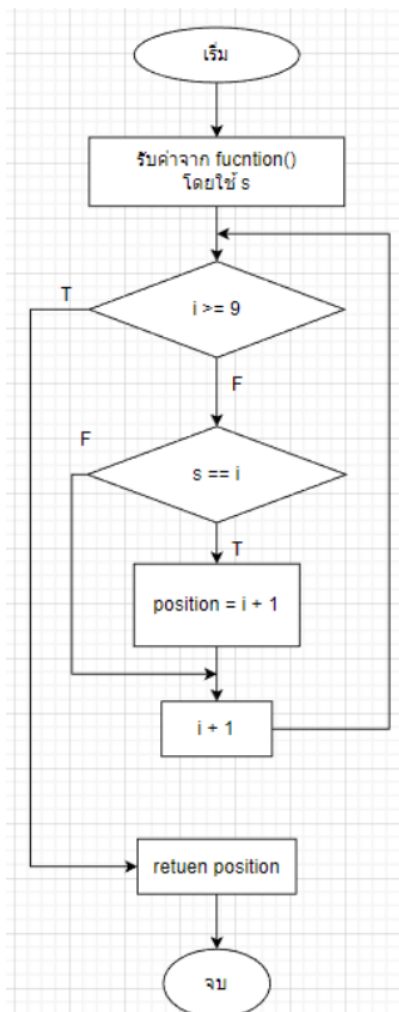
Method : Sort() ;

ผังงาน	โค้ดโปรแกรม
 <pre> graph TD     Start([เริ่ม]) --&gt; Decision1{i == 9}     Decision1 -- T --&gt; End([จบ])     Decision1 -- F --&gt; Decision2{j == 9-i}     Decision2 -- T --&gt; Decision3{j &gt; j+1}     Decision3 -- F --&gt; Decision2     Decision3 -- T --&gt; Swap[ทำการ swap ค่า temp = j j = j+1 j+1 = temp]     Swap --&gt; Increment[j + 1]     Increment --&gt; Display[/แสดงค่า array/]     Display --&gt; End         </pre>	<pre> public void Sort() {     for (int i = 0; i &lt; 9; i++){         for (int j = 0; j &lt; 9-i; j++) {             if (this.ar[j] &gt; this.ar[j+1]) {                  int temp = this.ar[j];                 this.ar[j] = this.ar[j+1];                 this.ar[j+1] = temp;             } //end if         } //end for j     } //end for i     System.out.print("Sort min -&gt; max : ");     for(int i = 0; i &lt;= 9; i++) {         System.out.print(" "+this.ar[i]+" ");     } //end for     System.out.println(""); } //End Method Sor         </pre>

--	--

Method : Search( Find ) ;

พนักงาน	โค้ดโปรแกรม



```

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
  
```

- 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 ) ;

โค้ดโปรแกรมภายในฟังก์ชันหลัก

```

package MyClassicalArray;
import java.util.Scanner;

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



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 จะทำงานคล้ายกับ Function ของ ภาษา C การที่จะเรียกใช้ Class ใน Class main จะต้องนำเข้า Class โดยการ Newclass `g = new Newclass();` แล้วเราสามารถเรียก Method ใน Class นั้นได้ ส่วนการสืบทอดคือการที่จะใช้ Property กับ Method ในคลาสแม่มาใช้ในคลาสนั้นได้ ด้วยคำสั่ง `extends`

## 6. คำถามท้ายการทดลอง

6.1. การสืบทอดในภาษาจาวาสามารถทำได้โดยใช้คำสั่งใด ?

**Extends**

6.2. จงอธิบายข้อควรระวังในการใช้งาน public, private และ protected

**Public class** หรือ **class** สามารถใช้ได้ หาก **class** นั้นมี **property** ชื่อเดียวกันอาจจะทำให้คำผิดพลาดได้ **Private class** ตัวเองเท่านั้น ที่ใช้ได้ จะมีปัญหาเวลาต้องการใช้ **property** นี้ใน **class** อื่น **Protected class** ตัวเองกับ **class** ที่สืบทอดไปที่มีปัญหาตอนคลาสที่ไม่ได้สืบทอดจะใช้ไม่ได้

6.3. วัตถุ และ คลาส มีความเหมือนหรือแตกต่างกันอย่างไร ?

**วัตถุ**คือ **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**

---

---

---

---