

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 class และ method

Class ใช้สำหรับเก็บข้อมูลต่างๆ เกี่ยวกับออบเจ็ค

Method เป็นการกำหนดฟังก์ชันการทำงานของออบเจ็ค

นี่เป็นรูปแบบการประกาศคลาสในภาษา Java

```
//class
Public class Lab4 {
//Method
Public static void main(String[] args) {
    //attribute
    Int i = 1;
}
}
```

- 3.2. วัตถุคืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

วัตถุที่สร้างขึ้นมาจาก Class หรืออาจจะเรียกได้อีกอย่างว่า instance โดยวัตถุจะสามารถนำไปใช้ดำเนินงานต่างๆ ได้ตามความสามารถที่ระบุไว้ใน Class

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

มี 2 รูปแบบ คือ

Global Attribute คือ คุณสมบัติของคลาสที่ประกาศนอก Method ซึ่งสามารถให้คลาสอื่น ๆ เข้าถึงได้ และสามารถใช้ในส่วนใดของคลาสก็ได้

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. ผิด syntax ของภาษา java
2. ควรแยกวัตถุเป็นก้อนๆ ไม่ควรนำมาปนกัน

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. ลำดับขั้นตอนการปฏิบัติการ

4.1. จงเขียนโปรแกรมสร้างคลาสในการจัดการอาเรย์ดังต่อไปนี้

4.1.1. สร้างคลาสชื่อว่า MyClassicalArray

มี Properties ชื่อว่า MyArray[] พร้อมกับสุ่มค่าไว้ภายในตัวแปรทั้งหมด 10 ค่า

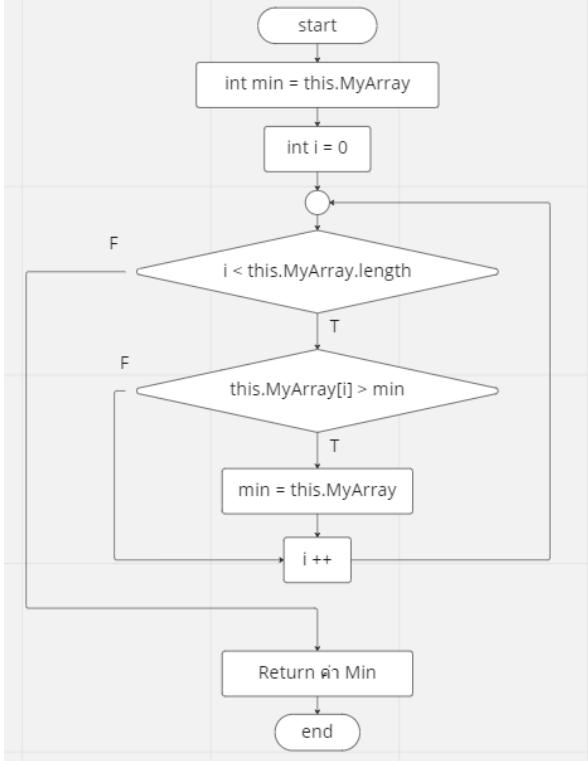
มี Method ชื่อว่า FindMax() ; เพื่อหาค่าที่มากที่สุดที่อยู่ในตัวแปร MyArray

มี Method ชื่อว่า FindMin() ; เพื่อหาค่าที่น้อยที่สุดที่อยู่ในตัวแปร MyArray

Method : FindMax();

ผังงาน	โค้ดโปรแกรม
<pre> graph TD Start([start]) --> InitMax[int max = this.MyArray] InitMax --> InitI[int i = 0] InitI --> LoopStart(()) LoopStart --> Cond1{i < this.MyArray.length} Cond1 -- F --> ReturnMax[Return คำ Max] ReturnMax --> End([end]) Cond1 -- T --> Cond2{this.MyArray[i] > max} Cond2 -- F --> IncI[i++] IncI --> LoopStart Cond2 -- T --> AssignMax[max = this.MyArray[i]] AssignMax --> IncI </pre>	<pre> public int FindMax() { int max = this.MyArray[0]; for(int i = 0 ; i < this.MyArray.length ; i++) if(this.MyArray[i] > max) { max = this.MyArray[i]; } return max; } </pre>

Method : FindMin() ;

ผังงาน	โค้ดโปรแกรม
 <pre> graph TD Start([start]) --> InitMin[int min = this.MyArray[0]] InitMin --> InitI[int i = 0] InitI --> LoopStart(()) LoopStart --> Cond1{i < this.MyArray.length} Cond1 -- F --> ReturnMin[Return ค่า Min] ReturnMin --> End([end]) Cond1 -- T --> Cond2{this.MyArray[i] > min} Cond2 -- T --> UpdateMin[min = this.MyArray[i]] UpdateMin --> IncI[i++] Cond2 -- F --> ReturnMin IncI --> LoopStart </pre>	<pre> public int FindMin() { int min = this.MyArray[0]; for(int i = 0 ; i < this.MyArray.length ; i++) if(this.MyArray[i] < min) { min = this.MyArray[i]; } return min; } </pre>

4.1.2. สร้างคลาสชื่อว่า MyCurrentArray ที่สืบทอดคลาส MyClassicalArray

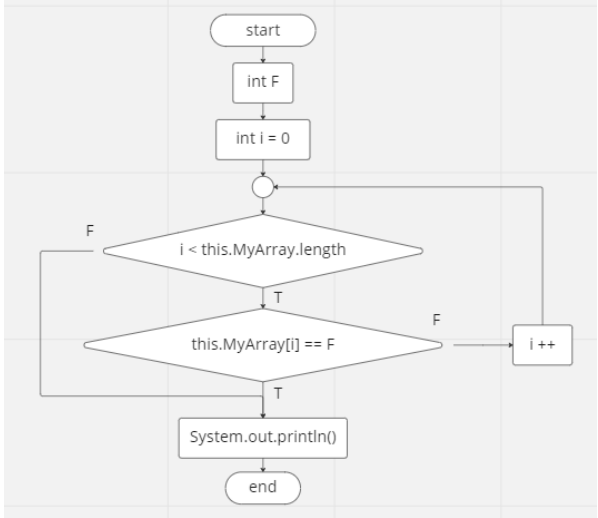
มี Method ชื่อว่า Sort() ; เพื่อเรียงค่าภายในตัวแปร MyArray จากน้อยไปมาก

มี Method ชื่อว่า Search(Find) ; เพื่อค้นหาค่าที่อยู่ในตัวแปร MyArray

Method : Sort() ;

ผังงาน	โค้ดโปรแกรม
<pre> graph TD Start([start]) --> SortMethod[Arrays.sort(this.MyArray)] SortMethod --> PrintHeader[System.out.print("Sort Array : ")] PrintHeader --> InitI[int i = 0] InitI --> LoopStart(()) LoopStart --> Cond1{i < this.MyArray.length} Cond1 -- F --> End([end]) Cond1 -- T --> PrintVal[System.out.print(this.MyArray[i])] PrintVal --> Cond2{i < this.MyArray.length - 1} Cond2 -- F --> IncI[i++] Cond2 -- T --> PrintComma[System.out.print(", ")] PrintComma --> IncI IncI --> LoopStart LoopStart --> PrintNewline[System.out.println()] PrintNewline --> End </pre>	<pre> public void sort() { Arrays.sort(this.MyArray); System.out.print("Sort Array : "); for(int i = 0 ; i < this.MyArray.length; i++) { System.out.print(this.MyArray[i]); if(i < this.MyArray.length - 1) { System.out.print(", "); } } System.out.println(); } </pre>

Method : Search(Find) ;

ผังงาน	โค้ดโปรแกรม
 <pre> graph TD Start([start]) --> FVar[int F] FVar --> iVar[int i = 0] iVar --> LoopStart(()) LoopStart --> Cond1{i < this.MyArray.length} Cond1 -- F --> End([end]) Cond1 -- T --> Cond2{this.MyArray[i] == F} Cond2 -- F --> Inc[i++] Inc --> LoopStart Cond2 -- T --> Print[System.out.println()] Print --> End </pre>	<pre> public boolean Search(int F) { for(int i = 0 ; i < this.MyArray.length; i++) { if (this.MyArray[i] == F) { return true; } } return false; } </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) ;

โค้ดโปรแกรมภายในฟังก์ชันหลัก

```
public static void main(String[] args) {
    MyCurrentArray arr = new MyCurrentArray();
    arr.sort();
    int F = 33;
    int max = arr.FindMax();
    System.out.println("Max value is : "+max);
    int min = arr.FindMin();
    System.out.println("Min value is : "+min);

    System.out.print("Search value result : "+F+" = ");
    System.out.print(arr.Search(F));
}
```

ผลลัพธ์การทำงานของโปรแกรม

```
24
89
49
78
84
16
66
88
42
87
Sort Array : 16, 24, 42, 49, 66, 78, 84, 87, 88, 89
Max value is : 89
Min value is : 16
Search value result : 33 = false
```


5. สรุปผลการปฏิบัติการ

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

6. คำถามท้ายการทดลอง

6.1. การสืบทอดในภาษาจาวาสามารถทำได้โดยใช้คำสั่งใด ?

คำสั่ง Extends

6.2. จงอธิบายข้อควรระวังในการใช้งาน public, private และ protected

Public class ทุก class สามารถใช้ได้ หาก class นั้นมี property ชื่อเดียวกันอาจจะทำให้คำผิดพลาดได้ Private class ตัวเองเท่านั้นที่ใช้ได้ จะมีปัญหาเวลาต้องการใช้ property นี้ใน class อื่น Protected class ตัวเองกับ class ที่สืบทอดไปเท่านั้นใช้ได้

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