

ใบงานการทดลองที่ 4

เรื่อง การกำหนดวัตถุ การใช้วัตถุ การสืบทอด และการห่อหุ้ม

1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจหลักการเขียน โปรแกรมเชิงวัตถุ คลาส การกำหนด และการใช้วัตถุ
- 1.2. รู้และเข้าใจหลักการสืบทอด และการห่อหุ้มวัตถุ

2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

3. ทฤษฎีการทดลอง

- 3.1. คลาสคืออะไร? มีลักษณะเด่นเป็นอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

Class คือ กลุ่มของ Object ที่มีโครงสร้างพื้นฐานพฤติกรรมเดียวกัน ดังนั้น Object ที่มีคุณสมบัติลักษณะเดียวกันนี้ จะรวมกลุ่มอยู่ใน Class เดียวกัน จึงสามารถสรุปได้ว่า Class คือ ต้นแบบข้อมูล ที่มีไว้เพื่อสร้าง Object นั้นเอง Class นอกจากจะมีชื่อ Class ที่บอกคุณสมบัติของ Class นั้นแล้ว ยังมี Attribute และ Operations ต่างๆ ซึ่งเป็นตัวอธิบายรายละเอียด และหน้าที่ต่างๆด้วย

ตัวอย่าง Class เช่น

Class คือ แบบรูปดาวที่สร้างจากแม่แบบ (Template) และ Object คือ ดาวที่สร้างจากแม่แบบรูปดาว หรือ Class นั้นเอง ซึ่ง Object ที่ถูกสร้างขึ้นมาจะมีลักษณะเหมือนกับ Class ที่เป็นต้นแบบ แต่โดยคุณสมบัติของ Object แล้ว จะสามารถเพิ่มคุณสมบัติเฉพาะของตัวเองขึ้นมาได้ เช่น เพิ่มสีขึ้นมาเป็นคุณสมบัติเฉพาะตัวเองขึ้นมาได้ ทำให้ดาวแต่ละรูปมีสีที่ต่างกันไป แต่คุณสมบัติพื้นฐานจะได้รับการสืบทอดมาจาก class ที่เป็นต้นแบบยังเหมือนเดิม นั่นคือขนาดของรูปดาวจะเท่ากันทุกรูป เพราะได้จากแม่พิมพ์ตัวเดียวกัน หรือ มาจาก class เดียวกันนั่นเอง

3.2. วัตถุคืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

Object หรือวัตถุนั้นถูกสร้างขึ้นจากคลาส(Class) โดยใช้ keyword ว่า new ซึ่งจะมีคุณสมบัติ (Property) และพฤติกรรม(Method) เหมือนกันกับคลาส(Class)ทุกประการ

```
<?php
class MyClass
{
    //กำหนดคุณสมบัติ(Property)
    public $myvar = 'สวัสดีชาวโลก';

    //กำหนดพฤติกรรม(Method)
    public function showMyvar(){
        echo $this->myvar;
    }
}

//สร้าง Object
$obj = new MyClass; //สร้าง object $obj โดยการ new
$obj->showMyvar(); //call หรือเรียก method showMyvar()
// output สวัสดีชาวโลก
```

3.3. คุณสมบัติ(Properties/Attributes) ควรมีลักษณะการประกาศอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

คุณสมบัติ (properties) คือสิ่งที่สามารถบ่งบอกถึงความเป็นวัตถุ และอยู่ภายในตัววัตถุซึ่งสามารถเปลี่ยนแปลงได้เช่น รูปร่าง ลักษณะ ความกว้าง ความยาว ฯลฯ สำหรับในแต่ละคอนโทรล หรือออบเจกต์ อาจจะมีคุณสมบัติที่เหมือนกัน

คุณสมบัติ (attribute) หมายความว่า คุณลักษณะหรือคุณสมบัติ เฉพาะของใครคนใดหรือสิ่งใดที่มี ในภาษาทางคอมพิวเตอร์ attribute จะใช้เป็นข้อมูลอธิบายคุณสมบัติของสิ่งที่พูดถึงนั้น เช่น ไฟล์ฐานข้อมูล เป็นต้น สามารถกำหนดได้ 3 ลักษณะ คือ

1. กำหนดโดยเครื่องหมาย "___" เช่น
2. กำหนดโดยเครื่องหมาย '___' เช่น
3. กำหนดโดยไม่ใช้เครื่องหมายใดๆ หรือ การใส่ค่าลงไปเลย เช่น ก็ได้เหมือนกัน

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

เมธอด เป็นกลุ่มของคำสั่งที่สร้างขึ้นมาเพื่อทำหน้าที่บางอย่าง การสร้างเมธอดจะสามารถทำให้เราใช้โค้ดนั้นซ้ำๆ โดยที่ไม่ต้องเขียนโปรแกรมใหม่เมื่อสร้างเมธอดในภาษา Java มันสามารถที่จะเรียกใช้งานได้จากส่วนใดๆ ของโปรแกรม ขึ้นกับขอบเขตและระดับการเข้าถึงที่ได้กำหนดขึ้น โดยรูปแบบในการสร้างเมธอดในภาษา Java เป็นดังนี้

```
type name ( parameter1, parameter2, ... ) {
    statements
}

access_modifier type name ( parameter1, parameter2, ... ) {
    statements
}
```

3.5. เพราะเหตุใดจึงควรสร้าง 1 คลาสต่อ 1 ไฟล์?

Class คือ แบบรูปดาวที่สร้างจากแม่แบบ (Template) และ Object คือ ดาวที่สร้างจากแม่แบบรูปดาว หรือ Class นั้นเอง ซึ่ง Object ที่ถูกสร้างขึ้นมามีลักษณะเหมือนกับ Class ที่เป็นต้นแบบ แต่โดยคุณสมบัติของ Object แล้ว จะสามารถเพิ่มคุณสมบัติเฉพาะของตัวเองขึ้นมาได้ ซึ่ง Class เป็นแม่แบบจึงต้องสร้าง 1 คลาสต่อ 1 ไฟล์

3.6. เมื่อสร้างวัตถุขึ้นมาแล้ว วัตถุจะสามารถอ้างอิง Properties หรือ Method ได้ด้วยวิธีการใด ?

โดยปกติแล้ว object จะถือแค่ properties ของใครของมัน ใช้แยกกัน ไม่มีการใช้ร่วมกับผู้อื่น แต่เนื่องจาก method นั้นเป็นโค้ดที่ไม่สามารถเปลี่ยนแปลงค่าได้ ก็ไม่ใช่ตัวแปรนี่นะ object แต่ละตัวจึงมักจะมีการ refer (อ้างอิง) method ที่มันทำงานได้กลับไปหาโค้ดต้นฉบับที่เก็บอยู่ที่คลาสแทน

3.7. คำสั่ง this มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

คำสั่ง this มีความสำคัญกับการเขียนโปรแกรมในภาษา Java แน่นนอนเนื่องจาก ภาษา Java มีโครงสร้างภาษาแบบ OOP ซึ่งจุดเด่นที่ขาดไม่ได้คือ เรื่องของการสืบทอดคุณสมบัติแล้วการสืบทอดคุณสมบัติ หรือ Inherit มันเกี่ยวกับ this ยังไง เรามาดูกันแน่นอนเมื่อมีการสืบทอดมันก็ย่อมมีเรื่องของ การทับซ้อนของ method ต่างๆ ของ Class แม้กับ Class ลูก อาจเนื่องมาจากการทำ Override Method เมื่อเกิดเหตุการณ์แบบนี้ขึ้น คำถามก็มีอยู่ว่าเราจะแยกแยะ Method ที่เหมือนกันของ Class แม้กับ class ปัจจุบันยังงี้ นี่คือการที่ทำให้เราต้องเรียกใช้ คำสั่ง this ใช้เพื่อเรียก ตัวแปร หรือ method ของ Class เราย่อย

3.8. Constructor Method มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

Constructor Method คือ Method ที่มีชื่อเดียวกับชื่อ Class โดยจะ ไม่มีการ Return หรือส่งค่าใด ๆ กลับออกไป โดยมีหน้าที่กำหนดค่าเริ่มต้นตอนสร้าง Class ภาพรวมของ Constructor Method

1. เป็น Method ที่มีชื่อเดียวกับชื่อ Class
2. เป็น Method สำหรับกำหนดค่าเริ่มต้นตอนสร้าง Class

ตัวอย่างโปรแกรม

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Data;
6  using System.Data.SqlClient;
7
8  namespace ConsoleApplication2
9  {
10     class Program
11     {
12         static void Main(string[] args)
13         {
14             ParentClass p = new ParentClass("AmlySoft");
15             p.sayHello();
16             Console.ReadLine();
17         }
18     }
19
20     class ParentClass
21     {
22         string name;
23
24         public ParentClass(string n)
25         {
26             name = n;
27         }
28
29         public void sayHello()
30         {
31             Console.WriteLine("Hello, {0} with Constructor Method",
32                             name.ToString());
33         }
34     }
35 }

```

3.9. Destructor Method มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

destructor เป็น method ที่ให้ class ทำอะไรบางอย่างก่อนที่ class นั้นจะถูกทำลายลง ผมขอเริ่มจากการสร้าง constructor ก่อนละกันครับ การสร้าง constructor นั้นจะมีข้อกำหนดว่าชื่อ constructor จะต้อง มีชื่อเดียวกับ class เสมอ

3.10. การสืบทอด(Inheritance) คืออะไร? มีข้อดีและข้อเสียอย่างไร?

Inheritance คือ การสืบทอดคุณสมบัติ และคุณลักษณะเช่น พ่อแม่ ถ่ายทอด DNA มาสู่ลูกเป็นต้น ซึ่งการเขียนโปรแกรมในภาษาจาวา และภาษาอื่นในเรื่องของการสืบทอดก็เป็นในลักษณะดังที่กล่าวมาเหมือนกัน โดยเรามาดูตัวอย่างการเขียนโปรแกรม และคำอธิบายต่าง ๆ ได้เลยครับ

ประโยชน์ของการทำ Inheritance

1. Class ลูกสามารถเรียกใช้งานคำสั่งต่าง ๆ จาก Class พ่อแม่ ได้เลยทันทีโดยไม่จำเป็นต้องเขียนโปรแกรมใหม่
2. ทำให้ Class ต่าง ๆ ที่มีลักษณะเหมือน ๆ กันทำการสืบทอดคุณสมบัติไปได้ และใช้งานได้โดยทันที
3. เพื่อจัดกลุ่ม และทำให้การออกแบบ Class มีประสิทธิภาพมากยิ่งขึ้น

3.11. จงยกตัวอย่างการสร้างคลาสรองเพื่อทำการสืบทอดจากคลาสหลัก

ในการสืบทอดคลาส นอกจากมันจะได้รับสมาชิกจากคลาสแม่แล้ว ยังสามารถเพิ่มสมาชิกและเมธอดของมันเองได้ ทำให้มันมีความสามารถเพิ่มมากขึ้น นี่เป็นรูปแบบการสืบทอดของคลาสในภาษา Java

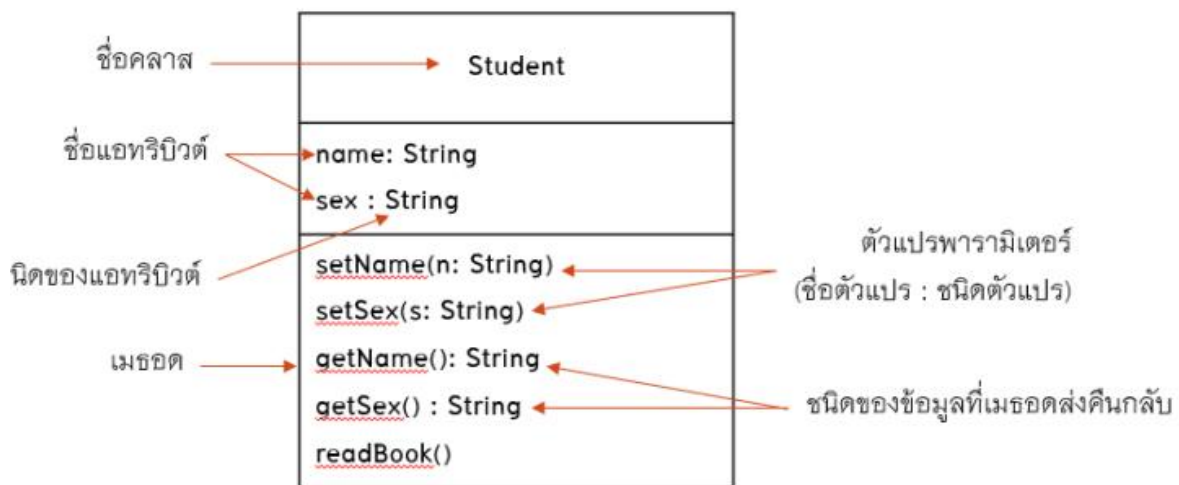
```
class DeliveredClass extends SuperClass {  
    ...  
}
```

3.12. จงยกตัวอย่างการสร้างวัตถุของคลาสหลักและคลาสรอง พร้อมกับยกตัวอย่างการเรียกใช้งานวัตถุนั้นแต่ละคลาส เพื่อให้เห็นภาพการสืบทอดการทำงานซึ่งกันและกัน

ขั้นตอนที่ 1 จำแนกออกแบบเจ็ท

ขั้นตอนที่ 2 ให้ Concept กับออกแบบเจ็ท

ขั้นตอนที่ 3 จัดกลุ่มออกแบบเจ็ท



รูปที่ 3.2 ตัวอย่างคลาสไดอะแกรม (Class Diagram) ของคลาส Student

ที่มา : Oesterich, B., 2002

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

public: คลาสหรือสมาชิกสามารถเข้าถึงได้จากที่ส่วนของโปรแกรม

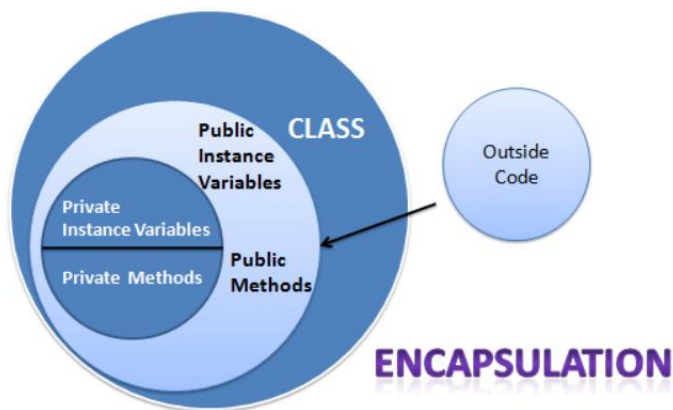
protected: คลาสหรือสมาชิกสามารถเข้าถึงได้ภายใน package เดียวกันและ sub class ของมัน

private: คลาสหรือสมาชิกสามารถเข้าถึงได้ภายในคลาสเดียวกันเท่านั้น

3.14. การห่อหุ้ม(Encapsulation) คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

Encapsulation คืออะไร

ในการเขียนโปรแกรมเชิงวัตถุ encapsulation คือการปกปิดหรือควบคุมการเข้าถึงข้อมูลของออบเจ็กต์จากภายนอก ซึ่งวิธีการที่ใช้ในการควบคุมคือการทำงานผ่านเมธอดเพื่อดำเนินการกับข้อมูลนั้น ในภาษา Java และภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุภาษาอื่นๆ นั้นสามารถใช้คุณสมบัตินี้ในการปกปิดส่วนประกอบภายในคลาสได้



4. ลำดับขั้นการปฏิบัติการ

4.1. จงเขียนโปรแกรมสร้างคลาสในการจัดการอาร์เรย์ดังต่อไปนี้

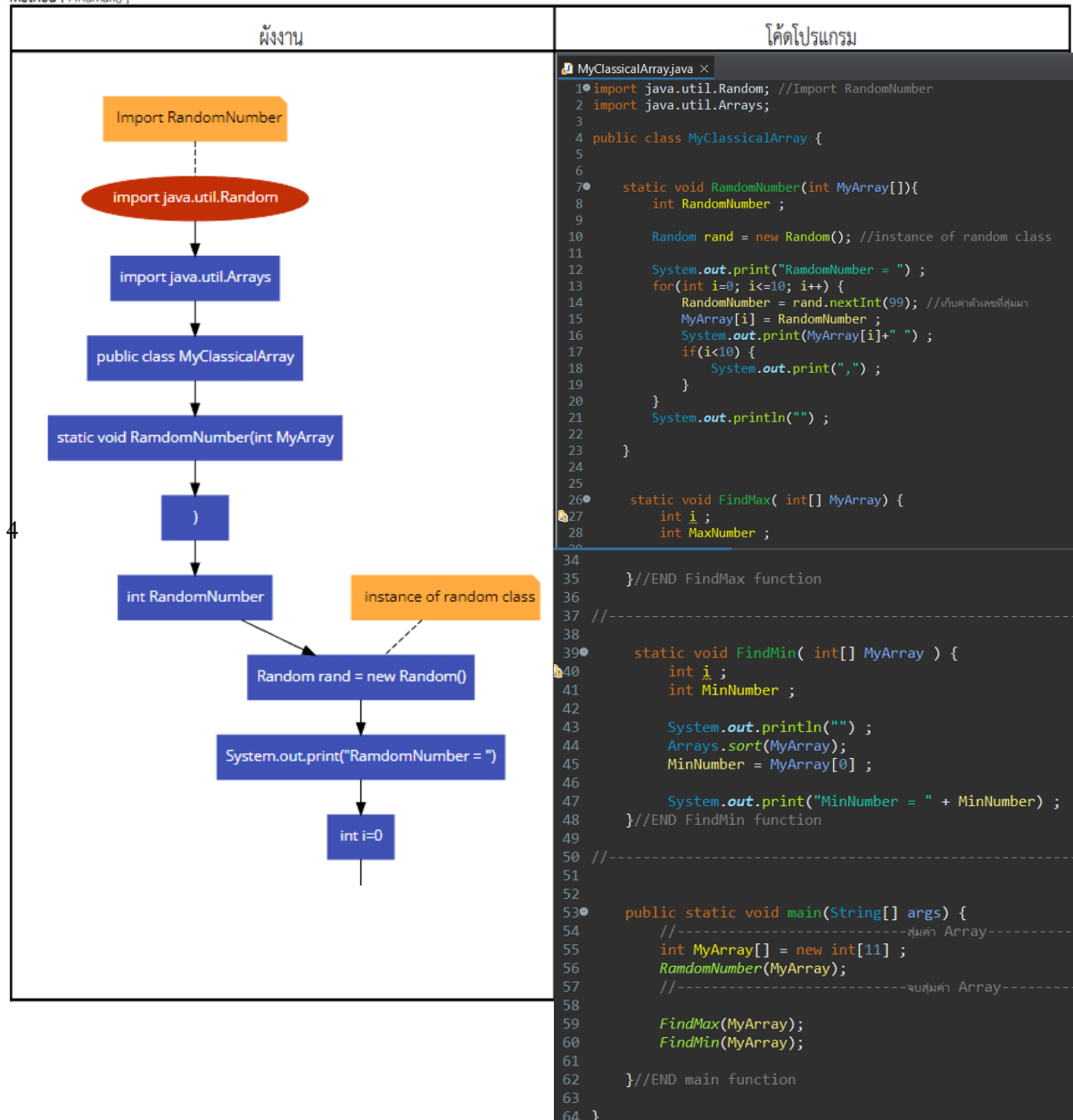
4.1.1. สร้างคลาสชื่อว่า MyClassicalArray

มี Properties ชื่อว่า MyArray[] พร้อมกับสุ่มค่าไว้ภายในตัวแปรทั้งหมด 10 ค่า

มี Method ชื่อว่า FindMax() ; เพื่อหาค่าที่มากที่สุดที่อยู่ในตัวแปร MyArray

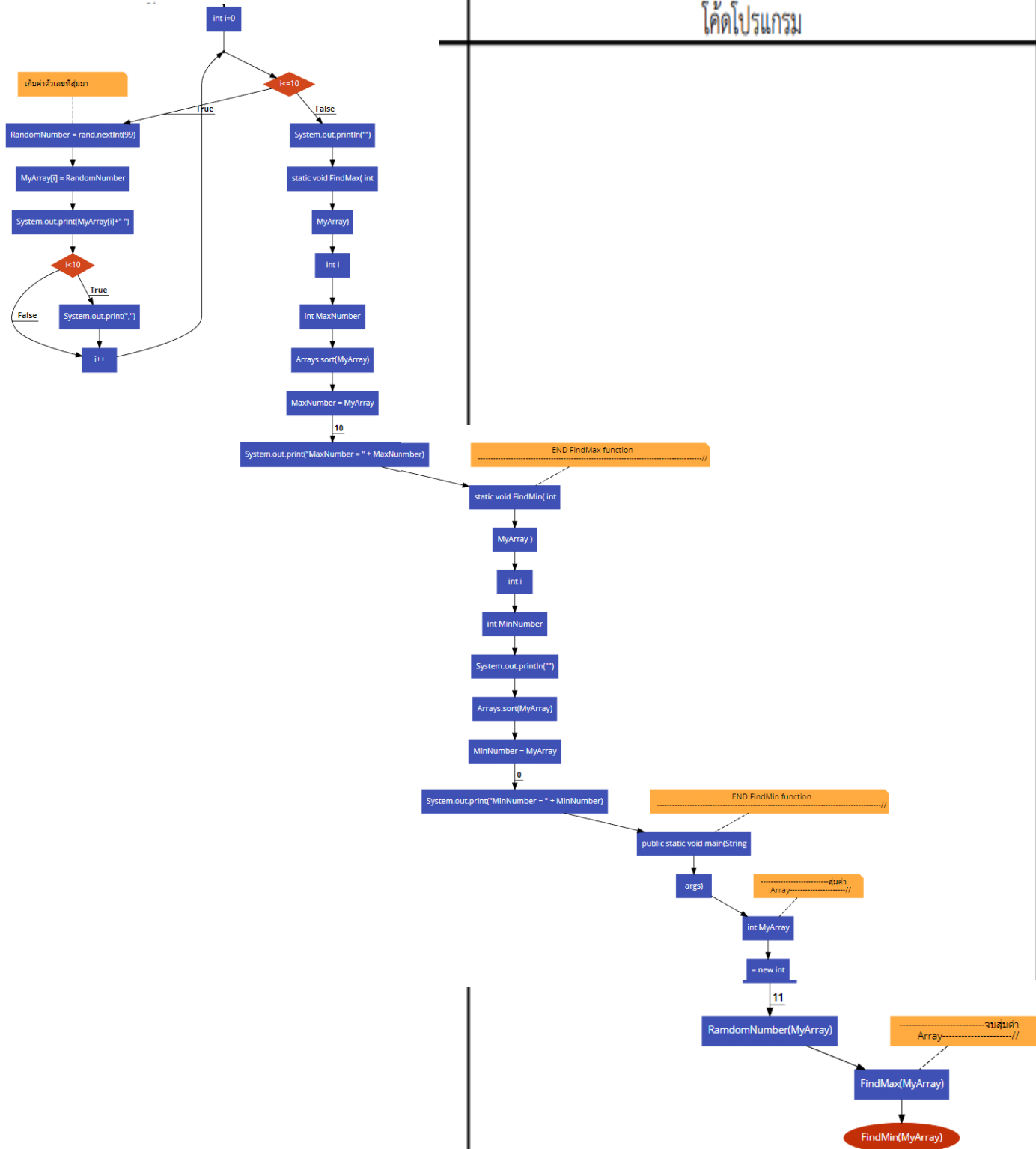
มี Method ชื่อว่า FindMin() ; เพื่อหาค่าที่น้อยที่สุดที่อยู่ในตัวแปร MyArray

Method : FindMax();



Method : FindMax();

โค้ดโปรแกรม



4.1.2. สร้างคลาสชื่อว่า MyCurrentArray ที่สืบทอดคลาส MyClassicalArray

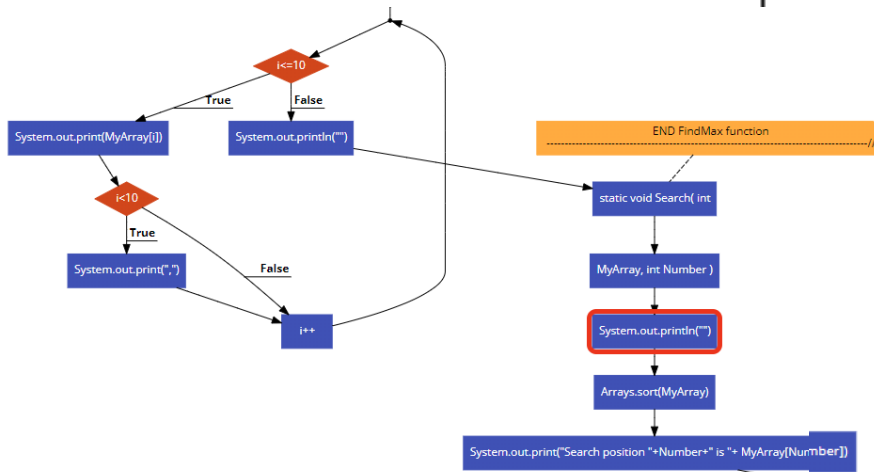
มี Method ชื่อว่า Sort() ; เพื่อเรียงค่าภายในตัวแปร MyArray จากน้อยไปมาก

มี Method ชื่อว่า Search(Find) ; เพื่อค้นหาค่าที่อยู่ในตัวแปร MyArray

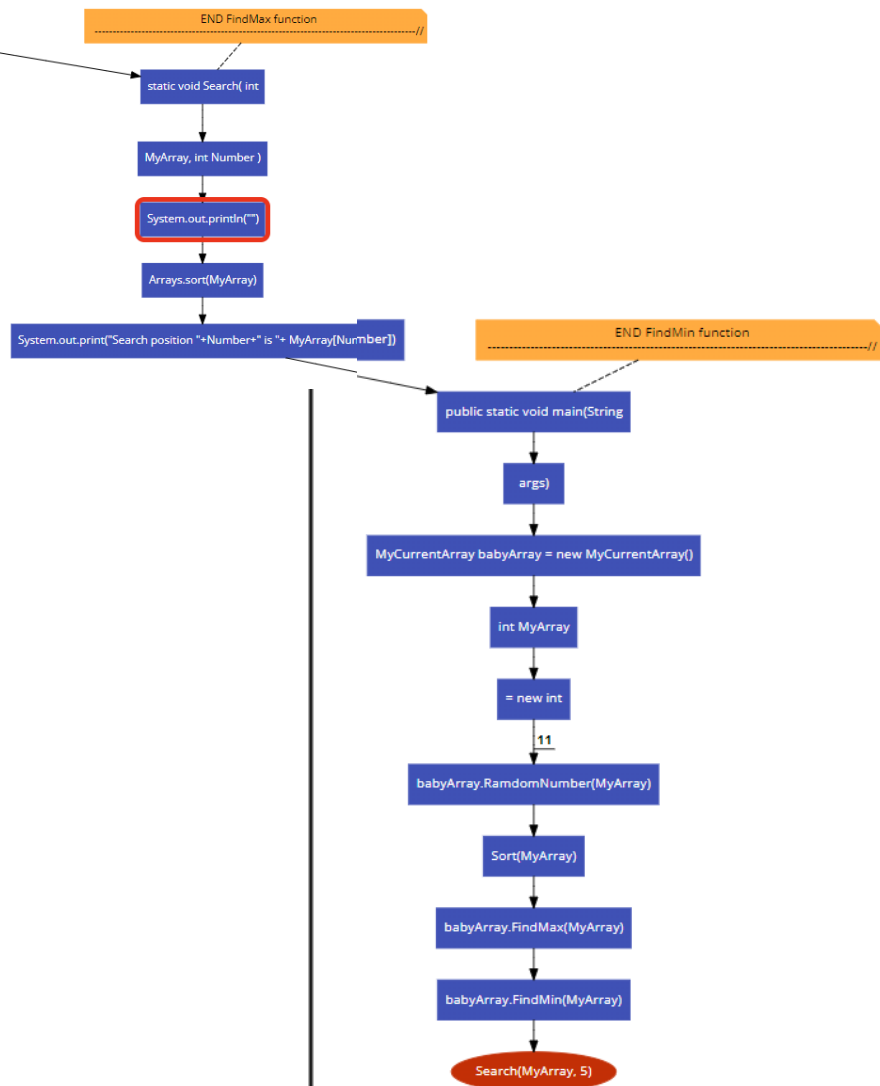
Method : FindMax();

ผังงาน	โค้ดโปรแกรม
<pre> graph TD A([import java.util.Arrays]) --> B[import java.util.Random] B --> C[public class MyCurrentArray extends MyClassicalArray] C --> D[static void Sort(int MyArray)] D --> E[int i] E --> F[int MaxNumber] F --> G[Arrays.sort(MyArray)] G --> H[System.out.print("Sort Array = ")] H --> I[i=0] </pre>	<pre> MyCurrentArray.java x 1 import java.util.Arrays; 2 import java.util.Random; 3 4 public class MyCurrentArray extends MyClassicalArray { 5 6 static void Sort(int[] MyArray) { 7 8 int i ; 9 int MaxNumber ; 10 11 Arrays.sort(MyArray); 12 13 System.out.print("Sort Array = ") ; 14 for(i=0; i<=10; i++) { 15 System.out.print(MyArray[i]) ; 16 if(i<10) { 17 System.out.print(",") ; 18 } 19 } 20 21 System.out.println("") ; 22 23 } //END FindMax function 24 25 //----- 26 27 static void Search(int[] MyArray, int Number) { 28 29 System.out.println("") ; 30 Arrays.sort(MyArray); 31 32 System.out.print("Search position "+Number+" is "+ MyArray[Number]) ; 33 34 } //END FindMin function 35 36 //----- 37 38 39 public static void main(String[] args) { 40 41 MyCurrentArray babyArray = new MyCurrentArray(); 42 int MyArray[] = new int[11] ; 43 44 babyArray.RandomNumber(MyArray); 45 Sort(MyArray); 46 babyArray.FindMax(MyArray); 47 babyArray.FindMin(MyArray); 48 Search(MyArray, 5); 49 50 } //END main function 51 52 } 53 </pre>

ผังงาน



โค้ดโปรแกรม



4.1.3. ในฟังก์ชันหลัก สร้างวัตถุจากคลาส MyClassicalArray ขึ้นมา และทดสอบการใช้งานคำสั่ง

FindMax() ; และคำสั่ง

FindMin() ;

```
static void FindMax( int[] MyArray) {  
    int i ;  
    int MaxNumber ;  
  
    Arrays.sort(MyArray);  
    MaxNumber = MyArray[10] ;  
  
    System.out.print("MaxNumber = " + MaxNumber) ;  
  
} //END FindMax function  
  
//-----  
  
static void FindMin( int[] MyArray ) {  
    int i ;  
    int MinNumber ;  
  
    System.out.println("") ;  
    Arrays.sort(MyArray);  
    MinNumber = MyArray[0] ;  
  
    System.out.print("MinNumber = " + MinNumber) ;  
  
} //END FindMin function
```

ผลลัพธ์การทำงานของโปรแกรม

```
<terminated> MyClassicalArray [Java Application] C:\Users\Augsupan tharawan\  
RamdomNumber = 51 ,25 ,93 ,27 ,67 ,5 ,14 ,11 ,81 ,68 ,59  
MaxNumber = 93  
MinNumber = 5
```

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

```

        Arrays.sort(MyArray);

        System.out.print("Sort Array = " ) ;
        for(i=0; i<=10; i++) {
            System.out.print(MyArray[i]) ;
            if(i<10) {
                System.out.print(",") ;
            }
        }

        System.out.println("") ;

    } //END FindMax function

//-----
static void Search( int[] MyArray, int Number ) {

    System.out.println("") ;
    Arrays.sort(MyArray);

    System.out.print("Search position "+Number+" is "+ MyArray[Number]) ;

} //END FindMin function
    
```

ผลลัพธ์การทำงานของโปรแกรม

```

<terminated> MyCurrentArray [Java Application] C:\Users\Augsupan tharawan\
RandomNumber = 29 ,52 ,58 ,0 ,45 ,12 ,19 ,57 ,15 ,16 ,79
Sort Array = 0,12,15,16,19,29,45,52,57,58,79
MaxNumber = 79
MinNumber = 0
Search position 5 is 29
    
```

5. สรุปผลการปฏิบัติการ

จากการทดลอง ในสร้างวัตถุจากคลาส MyClassicalArray ขึ้นมา และทดสอบการใช้งานคำสั่ง FindMax() ; และคำสั่ง FindMin() ; ได้ผลลัพธ์คือ

```
<terminated> MyClassicalArray [Java Application] C:\Users\Augsupan tharawan\
RamdomNumber = 51 ,25 ,93 ,27 ,67 ,5 ,14 ,11 ,81 ,68 ,59
MaxNumber = 93
MinNumber = 5
```

จากการทดลอง ในสร้างวัตถุจากคลาสสร้างวัตถุจากคลาส MyCurrentArray ขึ้นมา และทำการทดสอบการใช้งานคำสั่ง FindMax() ; คำสั่ง FindMin() ; คำสั่ง Sort() ; และคำสั่ง Search(Find) ; ได้ผลลัพธ์คือ

```
<terminated> MyCurrentArray [Java Application] C:\Users\Augsupan tharawan\
RamdomNumber = 29 ,52 ,58 ,0 ,45 ,12 ,19 ,57 ,15 ,16 ,79
Sort Array = 0,12,15,16,19,29,45,52,57,58,79
MaxNumber = 79
MinNumber = 0
Search position 5 is 29
```

6. คำถามท้ายการทดลอง

6.1. การสืบทอดในภาษาจาวาสามารถทำได้โดยใช้คำสั่งใด ?

การสืบทอด (Inheritance) ในภาษา JAVA

คือ การที่มีคลาสแม่ (Superclass) และคลาสลูก (Subclass) โดยคลาสลูกสามารถสืบทอด method และ attribute มาจากคลาสแม่ได้ แต่ไม่สามารถสืบทอดระดับการเข้าถึงที่เป็น private และ constructor นอกจากนี้ คลาสลูกเองก็สามารถมี mothod และ attribute นอกเหนือจากคลาสแม่ได้ และคลาสลูกสามารถมีคลาสอื่น ๆ มาสืบทอดต่อได้

CODE: SELECT ALL

```
class Subclass extends Superclass {
    ....
    ....
}
```

6.2. จงอธิบายข้อควรระวังในการใช้งาน public, private และ protected

ในภาษา Java นั้นมีคำสั่งในการควบคุมระดับการเข้าถึงอยู่ ระดับด้วยกัน คือ public, protected, private โดยแต่ละรูปแบบนั้นมีความหมายดังนี้

public: คลาสหรือสมาชิกสามารถเข้าถึงได้จากที่ส่วนของโปรแกรม

protected: คลาสหรือสมาชิกสามารถเข้าถึงได้ภายใน package เดียวกันและ sub class ของมัน

private: คลาสหรือสมาชิกสามารถเข้าถึงได้ภายในคลาสเดียวกันเท่านั้น

6.3. วัตถุ และ คลาส มีความเหมือนหรือแตกต่างกันอย่างไร ?

Object หรือวัตถุนั้นถูกสร้างขึ้นจากคลาส(Class) โดยใช้ keyword ว่า new ซึ่งจะมีคุณสมบัติ (Property) และพฤติกรรม(Method) เหมือนกันกับคลาส(Class)ทุกประการ

Class คือ ต้นแบบข้อมูล ที่มีไว้เพื่อสร้าง Object นั้นเอง Class นอกจากจะมีชื่อ Class ที่บอกคุณสมบัติของ Class นั้นแล้ว ยังมี Attribute และ Operations ต่างๆ ซึ่งเป็นตัวอธิบายรายละเอียด และหน้าที่ต่างๆด้วย

6.4. ในฐานะที่เป็นผู้พัฒนาระบบ คุณจะเลือกใช้การสืบทอดคลาสเมื่อใด? เพราะเหตุใด ?

เลือกใช้การสืบทอดคลาสเมื่อสร้างคลาสใหม่โดยสืบทอดจากคลาสเดิมได้ โดยคลาสที่ได้รับการสืบทอดจะได้รับตัวแปรและเมธอดจากคลาสหลักมาใช้งาน และสามารถเพิ่มเติมการทำงานหรือความสามารถเข้าไปได้ นี่เป็นแนวคิดที่สำคัญของออกแบบโปรแกรมให้นำโค้ดกลับมาใช้ใหม่ได้ (Reusable)

