ใบงานการทดลองที่ 4

เรื่อง การกำหนดวัตถุ การใช้วัตถุ การสืบทอด และการหอหุ้ม

1. จุดประสงค์ทั่วไป

1.1. รู้และเข้าใจหลักการเขียนโปรแกรมเชิงวัตถุ คลาส การกำหนด และการใช้วัตถุ

1.2. รู้และเข้าใจหลักการสืบทอด และการหอหุ้มวัตถุ

2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

3. ทฤษฎีการทดลอง

3.1. คลาสคืออะไร? มีลักษณะเด่นเป็นอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

มันคือวัตถุ ตัวอย่างpublic class

3.2. วัตถุคืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

ตัวแปลที่อยู่ในคลาส เช่นnum1()

3.3. คุณสมบัติ(Properties/Attributes) ควรมีลักษณะการประกาศค่าอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

Attributes/Propertiesคือชื่อตัวแปล

```
public class Person {
    String name;
    int age;
    String gender;
```

}

Person person1 = new Person();

person1.name = "Alice";

person1.age = 24;

person1.gender = "female";

Person person2 = new Person();

person2.name = "Bob";

person2.age = 32;

person2.gender = "male";

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

```
public static void  MyClassicalArray() {
  int num1[] = {10,45,22,115,66,33,34,55,66,99 };
  int NumberMax = Arrays.stream(num1).max().getAsInt();
  System.out.println(NumberMax);

        }
```

3.5. เพราะเหตุใดจึงควรสราง 1 คลาสตอ 1 ไฟล ?

เพื่อให้ไม่ซับซ้อน

3.6. เมื่อสร้างวัตถุขึ้นมาแล้ว วัตถุจะสามารถอ้างอิง Properties หรือ Method ได้ด้วยวิธีการใด ?

```
public static void main(String[] args) {
        // TODO Auto-generated method stub
        MyClassicalArray();
        min();
        Person art = new Person();
        art.Mysys();
        System.out.println("");
        art.sh();

            }
```

3.7. คำสั่ง this มีหน้าที่อย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

การอ้างถึงอินสแตนซ์ปัจจุบันของคลาสนั้น

```
public class Person {
    private String name;
    private int age;
    private String gender;

    public Person(String name, int age, String gender) {
        this.name = name;
        this.age = age;
        this.gender = gender;
    }

    public String getName() {
        return this.name;
    }

    public int getAge() {
        return this.age;
    }
```

```
    public String getGender() {
        return this.gender;
    }
}
```

3.8. Constructor Method มีหนาที่อยางไร? อธิบายพรอมยกตัวอยางประกอบ

คือตัวเริ่มต้นสถานะของวัตถุ

```
public class Person {
    private String name;
    private int age;
    private String gender;

    public Person(String name, int age, String gender) {
        this.name = name;
        this.age = age;
        this.gender = gender;
    }
}
```

3.9. Destructor Method มีหนาที่อยางไร? อธิบายพรอมยกตัวอยางประกอบ

ดำเนินการล้างข้อมูล

เช่นอาจใช้วิธีการปิดท้ายเพื่อปิดตัวจัดการไฟล์ที่เปิดอยู่หรือการเชื่อมต่อฐานข้อมูลที่วัตถุนั้นใช้อยู่ เช่นเดียวกับในภาษาการเขียนโปรแกรมอื่นๆ เมธอดสุดท้ายจะถูกเรียกใช้โดยอัตโนมัติ

3.10. การสืบทอด(Inheritance) คืออะไร? มีข้อดีและข้อเสียอย่างไร?

การสืบทอดเป็นแนวคิดพื้นฐานในการเขียนโปรแกรมเชิงวัตถุและได้รับการสนับสนุนในภาษาจาวา การสืบทอดทำให้คลาส (หรือที่เรียกว่า "คลาสย่อย") สืบทอดคุณสมบัติและวิธีการของคลาสอื่น (เรียกว่า "ซูเปอร์คลาส") ซึ่งหมายความว่าคลาสย่อยสามารถใช้เมธอดและคุณสมบัติที่กำหนดไว้ในซูเปอร์คลาสได้ และยังสามารถเพิ่มเมธอดและคุณสมบัติของตัวเองได้อีกด้วย ข้อดีหลักอย่างหนึ่งของการสืบทอดคือการอนุญาตให้ใช้โค้ดซ้ำได้ ด้วยการสร้างซูเปอร์คลาสที่กำหนดเมธอดและคุณสมบัติทั่วไป จากนั้นจึงสร้างคลาสย่อยที่สืบทอดมาจากซูเปอร์คลาส คุณจะสามารถหลีกเลี่ยงการเขียนโค้ดเดียวกันหลายๆ ครั้งได้ วิธีนี้จะช่วยประหยัดเวลาและทำให้โค้ดของคุณดูแลรักษาได้ง่ายขึ้น ข้อดีอีกประการของการสืบทอดคือช่วยให้มีลำดับชั้นที่ชัดเจนและเป็นระเบียบ ด้วยการใช้การสืบทอด คุณสามารถสร้างลำดับชั้นของคลาสโดยที่แต่ละคลาสย่อยจะสืบทอดมาจากซูเปอร์คลาส และคลาสย่อยยังสามารถจัดคลาสย่อยเพิ่มเติมเพื่อสร้างคลาสพิเศษเพิ่มเติมได้ ซึ่งจะทำให้เข้าใจและสำรวจโค้ดของคุณได้ง่ายขึ้น อย่างไรก็ตาม มรดกก็มีข้อเสียเช่นกัน ข้อเสียเปรียบหลักประการหนึ่งคือทำให้โค้ดของคุณมีความยืดหยุ่นน้อยลง เมื่อคุณสร้างซูเปอร์คลาสและคลาสย่อยแล้ว อาจเป็นเรื่องยากที่จะทำการเปลี่ยนแปลงกับซูเปอร์คลาสโดยไม่ทำลายคลาสย่อยที่สืบทอดมา ซึ่งอาจทำให้แก้ไขโค้ดได้ยากขึ้นเมื่อโปรเจกต์ของคุณพัฒนาขึ้น นอกจากนี้ การใช้การสืบทอดสามารถทำให้โค้ดของคุณซับซ้อนขึ้นได้ ในการใช้การสืบทอดอย่างเหมาะสม คุณต้องมีความเข้าใจที่ชัดเจนเกี่ยวกับความสัมพันธ์ระหว่างคลาสของคุณ และคุณต้องออกแบบลำดับชั้นของคลาสอย่างระมัดระวัง สิ่งนี้สามารถเพิ่มภาระทางความคิดในการทำงานกับโค้ดของคุณ และทำให้ผู้อื่นเข้าใจและทำงานกับโค้ดของคุณได้ยากขึ้น โดยรวมแล้ว การสืบทอดเป็นแนวคิดที่ทรงพลังซึ่งมีประโยชน์ในหลาย ๆ สถานการณ์ แต่ก็ไม่ใช่เครื่องมือที่เหมาะสมสำหรับงานเสมอไป สิ่งสำคัญคือต้องพิจารณาข้อดีและข้อเสียของการใช้การสืบทอดอย่างรอบคอบก่อนที่จะนำไปใช้ในโค้ดของคุณ

3.11. จงยกตัวอย่างการสรางคลาสรองเพื่อทำการสืบทอดจากคลาสหลัก

```
public class Parent {
    // fields and methods go here
}

public class Child extends Parent {
    // fields and methods go here
}
public class Parent {
    public void doSomething() {
        // implementation goes here
    }
}

public class Child extends Parent {
    public void doSomething() {
        // new implementation goes here
    }
}
```

3.12. จงยกตัวอย่างการสร้างวัตถุของคลาสหลักและคลาสรอง พร้อมกับยกตัวอย่างการเรียกใช้งานวัตถุในแต่ละคลาส ที่ให้เห็น ภาพการสืบทอดการทำงานซึ่งกันและกัน

```
public class Parent {
    // fields and methods go here
}

public class Child extends Parent {
    // fields and methods go here
}
public class Parent {
    public void doSomething() {
        // implementation goes here
    }
}

public class Child extends Parent {
    public void doSomething() {
        // new implementation goes here
    }
}
```

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

ใน Java ตัวดัดแปลงการเข้าถึงคือคีย์เวิร์ดที่ใช้เพื่อตั้งค่าระดับการเข้าถึงสำหรับคลาส เมธอด หรือตัวแปร ตัวแก้ไขการเข้าถึงสามตัวใน Java เป็นแบบสาธารณะ ปกป้อง และเป็นส่วนตัว สาธารณะ: ตัวแปรหรือวิธีการสาธารณะสามารถเข้าถึงได้จากทุกที่ในโปรแกรม ป้องกัน: ตัวแปรหรือเมธอดที่ได้รับการป้องกันสามารถเข้าถึงได้จากภายในแพ็คเกจเดียวกัน หรือจากคลาสย่อยของคลาสที่มีการประกาศตัวแปรหรือเมธอด ส่วนตัว: ตัวแปรหรือเมธอดส่วนตัวสามารถเข้าถึงได้จากภายในคลาสที่มีการประกาศเท่านั้น ตัวดัดแปลงการเข้าถึงเหล่านี้ใช้เพื่อควบคุมการมองเห็นและการเข้าถึงของสมาชิกชั้นเรียน ตามค่าดีฟอลต์ สมาชิกคลาสทั้งหมดใน Java เป็นแบบส่วนตัว ซึ่งหมายความว่าสามารถเข้าถึงได้ภายในคลาสที่มีการประกาศเท่านั้น อย่างไรก็ตาม คุณสามารถใช้ตัวแก้ไขการเข้าถึงเพื่อเปลี่ยนการมองเห็นและการเข้าถึงของสมาชิกชั้นเรียนได้ตามต้องการ ตัวอย่างเช่น คุณอาจต้องการทำให้ตัวแปรหรือเมธอดเป็นแบบสาธารณะหากคุณต้องการให้คลาสอื่นๆ ในโปรแกรมของคุณพร้อมใช้งาน หรือคุณอาจต้องการทำให้ตัวแปรหรือเมธอดมีการป้องกันหากคุณต้องการให้พร้อมใช้งานสำหรับคลาสย่อยของคลาสนั้น มีการประกาศ

3.14. การหอหุม(Encapsulation) คืออะไร? อธิบายพรอมยกตัวอยางประกอบ

Encapsulation เป็นแนวคิดในการเขียนโปรแกรมเชิงวัตถุ (OOP) ที่อ้างถึงการรวมข้อมูลด้วยวิธีการที่ดำเนินการกับข้อมูลนั้น เป็นกลไกในการซ่อนรายละเอียดภายในหรือสถานะของวัตถุจากโลกภายนอก และอนุญาตให้เข้าถึงคุณสมบัติและพฤติกรรมของวัตถุผ่านอินเทอร์เฟซที่กำหนดไว้อย่างดีเท่านั้น ตัวอย่างเช่น พิจารณาคลาสที่แสดงถึงบัญชีธนาคาร ชั้นเรียนอาจมีฟิลด์สำหรับจัดเก็บยอดเงินในบัญชีและชื่อเจ้าของบัญชี ตลอดจนวิธีการฝากเงิน ถอนเงิน และตรวจสอบยอดเงินคงเหลือ

```
class BankAccount {
  private double balance;  // The account balance
  private String name;     // The account holder's name
```

```
  // Constructor to initialize the account
  public BankAccount(double initialBalance, String accountHolderName) {
    balance = initialBalance;
    name = accountHolderName;
  }

  // Method to deposit money into the account
  public void deposit(double amount) {
    balance += amount;
  }

  // Method to withdraw money from the account
  public void withdraw(double amount) {
    balance -= amount;
  }

  // Method to check the account balance
  public double checkBalance() {
    return balance;
  }
}
```

4. ลำดับขั้นการปฏิบัติการ

4.1. จงเขียนโปรแกรมสรางคลาสในการจัดการอาเรยดังตอไปนี้

4.1.1. สรางคลาสชื่อวา MyClassicalArray

มี Properties ชื่อวา MyArray[ ]

พรอมกับสุมคาไวภายในตัวแปรทั้งหมด 10คา มี Method

ชื่อวา FindMax( ) ;

เพื่อหาคาที่มากที่สุดที่อยในตัวแปร MyArray มี

Method ชื่อวา FindMin( ) ;

เพื่อหาคาที่นอยที่สุดที่อยูในตัวแปร MyArray

Method : FindMax() ;

ผังงาน	โคดโปรแกรม

```
1  package Array;
2  import java.util.Arrays;
3  public class main {
4
5      public main() {
6          // TODO Auto-generated co
7
8      }
9
10     public static void main(Strin
11         // TODO Auto-generated me
12         MyClassicalArray();
13     }
14     public static void  MyClassic
15         int num1[] = {10,45,22,11
16         int NumberMax = Arrays.st
17         System.out.println(Number
18         int NumberMin = Arrays.st
19         System.out.println(Number
20     }
21 }
22
```

Method : FindMin() ;

ผังงาน	โคดโปรแกรม

4.1.2. สรางคลาสชื่อวา MyCurrentArray ที่สืบทอดคลาส MyClassicalArray มี Method ชื่อวา Sort( ) ; เพื่อเรียงคาภายในตัวแปร MyArray จากนอยไปมาก และมี Method ชื่อวา Search(Find) ; เพื่อคนหาคาที่อยูภายในตัวแปร MyArray

Method : Sort() ;

ผังงาน	โคดโปรแกรม

```
package Array;
import java.util.Arrays;
public class main {

    public main() {
        // TODO Auto-generated const
    }
    public static void main(String[]
        // TODO Auto-generated metho
        MyClassicalArray();
        min();
        Person art = new Person();
        art.Mysys();
        System.out.println("");
        art.sh();

    }
    public static void  MyClassicalA
        int numl[] = {10,45,22,115,6
        int NumberMax = Arrays.strea
        System.out.println(NumberMax
    }
    public static void  min() {
        int numl[] = {10,45,22,115,6
        int NumberMin = Arrays.strea
        System.out.println(NumberMin
    }
}
class Person {
    public static void  Mysys() {
        int numl[] = {10,45,22,115,6
        int NumberMax = Arrays.strea
        System.out.println(NumberMax
        System.out.println("The orig
        for (int num : numl) {
            System.out.print(num + "
        }
        Arrays.sort(numl);
        System.out.println("\nThe so
        for (int num : numl) {
```

	System.out.print(num +
	}
	
	}
	public static void sh() {
	int numl[] = {10,45,22,115,6
	for (int i = 0; i < numl.ler
	if(numl[i] == 99){
	System.out.println(r
	}
	}
	}
	}

Method : Search(Find) ;

ผังงาน	โคดโปรแกรม

4.1.3. ในฟงกชันหลัก สรางวัตถุจากคลาส MyClassicalArray ขึ้นมา และทดสอบการใชงานคำสั่ง FindMax() ; และคำสั่ง FindMin() ;

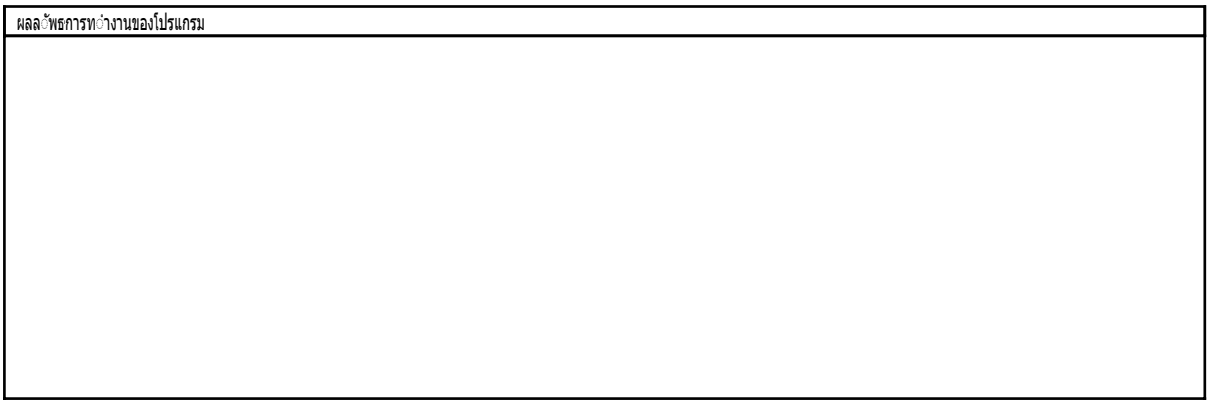
4.1.4. ในฟงกชันหลัก สรางวัตถุจากคลาส MyCurrentArray ขึ้นมา และทำการทดสอบการใชงานคำสั่ง FindMax() ; คำสั่ง FindMin() ; คำสั่ง Sort() ; และคำสั่ง Search(Find) ;

โคดโปรแกรมภายในฟงกชันหลัก

```java
package Array;
import java.util.Arrays;
public class main {

    public main() {
        // TODO Auto-generated constructor stub
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        MyClassicalArray();
        min();
        Person art = new Person();
        art.Mysys();
        System.out.println("");
        art.sh();

    }
    public static void  MyClassicalArray() {
        int num1[] = {10,45,22,115,66,33,34,55,66,99 };
        int NumberMax = Arrays.stream(num1).max().getAsInt();
        System.out.println(NumberMax);
    }
    public static void  min() {
        int num1[] = {10,45,22,115,66,33,34,55,66,99 };
        int NumberMin = Arrays.stream(num1).min().getAsInt();
        System.out.println(NumberMin);
    }
}
class Person {
    public static void  Mysys() {
        int num1[] = {10,45,22,115,66,33,34,55,66,99 };
        int NumberMax = Arrays.stream(num1).max().getAsInt();
        System.out.println(NumberMax);
        System.out.println("The original array is: ");
        for (int num : num1) {
            System.out.print(num + " ");
        }
        Arrays.sort(num1);
        System.out.println("\nThe sorted array is: ");
        for (int num : num1) {
```

```
            System.out.print(num + " ");
        }

    }
    public static void sh() {
        int num1[] = {10,45,22,115,66,33,34,55,66,99 };
        for (int i = 0; i < num1.length; i++){
            if(num1[i] == 99){
                System.out.println(num1[i]);
            }
        }
    }
}
```

ผลลัพธการทำงานของโปรแกรม



```
115
10
115
The original array is:
10 45 22 115 66 33 34 55 66 99
The sorted array is:
10 22 33 34 45 55 66 66 99 115
99
```

5. สรุปผลการปฏิบัติการ

ให้ได้รู้เกี่ยวกับการใช้ฟังชั่นตัวแปล

6. คำถามทายการทดลอง

6.1. การสืบทอดในภาษาจาวาสามารถทำได้โดยใช้คำสั่งใด ?

```
class Car extends Vehicle {
  // Code for the Car class goes here
}
```

6.2. จงอธิบายข้อควรระวังในการใช้งาน public, private และ protected

ในคลาส ตัวแก้ไขการเข้าถึงเหล่านี้จะกำหนดการมองเห็นและการเข้าถึงของสมาชิกคลาส (ฟิลด์ เมธอด ฯลฯ) ที่นำไปใช้

6.3. วัตถุ และ คลาส มีความเหมือนหรือแตกต่างกันอย่างไร ?

คลาสคือแม่แบบหรือพิมพ์เขียวที่ใช้สร้างวัตถุ กำหนดคุณสมบัติและพฤติกรรมที่วัตถุของคลาสนั้นจะมี ด้วยวิธีนี้ คลาสสามารถถูกมองว่าเป็นประเภทที่อธิบายลักษณะและพฤติกรรมของวัตถุประเภทนั้น ในทางกลับกัน วัตถุเป็นตัวอย่างเฉพาะของคลาส มันมีเอกลักษณ์เฉพาะของตัวเองและมีค่าของตัวเองสำหรับคุณสมบัติที่กำหนดโดยคลาสของมัน กล่าวอีกนัยหนึ่ง วัตถุเป็นตัวอย่างที่เป็นรูปธรรมและเป็นจริงของคลาส

6.4. ในฐานะที่เปนผูพัฒนาระบบ คุณจะเลือกใชการสืบทอดคลาสเมื่อใด? เพราะเหตุใด ?

ในฐานะผู้พัฒนาระบบ ฉันจะเลือกใช้การสืบทอดคลาสเมื่อฉันต้องการสร้างคลาสใหม่ที่อิงจากคลาสที่มีอยู่ สิ่งนี้มีประโยชน์เพราะช่วยให้ฉันสามารถนำโค้ดและพฤติกรรมของคลาสที่มีอยู่กลับมาใช้ใหม่ได้ ในขณะเดียวกันก็เพิ่มฟีเจอร์และฟังก์ชันใหม่ให้กับคลาสใหม่ สิ่งนี้สามารถช่วยประหยัดเวลาและความพยายาม เนื่องจากฉันไม่ต้องเริ่มต้นใหม่ตั้งแต่ต้นกับชั้นเรียนใหม่ นอกจากนี้ การใช้การสืบทอดยังช่วยส่งเสริมการใช้โค้ดซ้ำและโมดูลาร์ ซึ่งจะทำให้ระบบโดยรวมสามารถบำรุงรักษาและปรับขนาดได้มากขึ้น