

## การทดลองที่ 4 คลาสและอ็อบเจกต์

ให้นักศึกษาสร้าง Class file ชื่อ Lab4 แล้วเขียนโปรแกรมดังต่อไปนี้

## 4.1 ให้พิมพ์โปรแกรมดังตัวอย่างต่อไปนี้

```
class Car{  
    public int v;  
    public int s;  
}  
  
public class Lab4 {  
    public static void main(String[] args) {  
        Car c = new Car();  
        Car c2 = new Car();  
        c.v = 3;  
        c2.v = 4;  
        c.s = 10;  
        c2.s = 20;  
        System.out.println("V of 1st car : "+c.v+" S of 1st car : "+c.s);  
        System.out.println("V of 2nd car : "+c2.v+" S of 2nd car : "+c2.s);  
    }  
}
```

Compile และเขียนผลลัพธ์ที่ได้

Output

## 4.2 จากข้อ 4.1 เปลี่ยน public int v; และ public int s; เป็น private int v; และ private int s; ตามลำดับ

Compile และเขียนผลลัพธ์ที่ได้

Output

4.3 จากข้อ 4.1 และ 4.2 ให้เติม Method ต่อไปนี้ลงใน Class Car

```
public int getV() {  
    return v;  
}
```

```
public void setV(int v) {  
    this.v = v;  
}
```

```
public int getS() {  
    return s;  
}
```

```
public void setS(int s) {  
    this.s = s;  
}
```

Compile และเขียนผลลัพธ์ที่ได้

Output

4.4 จากข้อ 4.1, 4.2, 4.3 ให้แก้ไข Method main() ดังต่อไปนี้

```
public static void main(String[] args) {  
    Car c = new Car();  
    Car c2 = new Car();  
    c.setV(3);  
    c2.setV(4);  
    c.setS(10);  
    c2.setS(20);  
    System.out.println("V of 1st car : "+c.getV()+" S of 1st car : "+c.getS());  
    System.out.println("V of 2nd car : "+c2.getV()+" S of 2nd car : "+c2.getS());  
}
```

Compile และเขียนผลลัพธ์ที่ได้

Output

4.5 จงอธิบายหลักการของคุณสมบัติของ Encapsulation จากข้อ 4.1-4.4 ว่าเกิดอะไรขึ้น และเหตุใดจึงมีผลลัพธ์เป็นเช่นนั้น

คำอธิบาย

4.6 ให้พิมพ์โปรแกรมดังตัวอย่างต่อไปนี้

```
class Car{
    private String brand;
    private int v;
    private int x;
    private int y;
    Car(){
        this.brand = "Toyota";
        this.v = 0;
        this.x = 0;
        this.y = 0;
    }
    Car(String brand,int v,int x,int y){
        this.brand = brand;
        this.v = v;
        this.x = x;
        this.y = y;
    }
    public String getBrand() {
        return brand;
    }
    public void setBrand(String brand) {
        this.brand = brand;
    }
}
```

```
public int getV() {  
    return v;  
}  
public void setV(int v) {  
    this.v = v;  
}  
public int getX() {  
    return x;  
}  
public void setX(int x) {  
    this.x = x;  
}  
public int getY() {  
    return y;  
}  
public void setY(int y) {  
    this.y = y;  
}  
}  
public class Lab4 {  
    public static void main(String[] args) {  
        Car c1 = new Car();  
        Car c2 = new Car("Honda",10,100,100);  
        System.out.println("Brand : "+c1.getBrand()+" v : "+c1.getV());  
        System.out.println("Brand : "+c2.getBrand()+" v : "+c2.getV());  
    }  
}
```

Compile และเขียนผลลัพธ์ที่ได้

Output

4.7 จาก 4.6 ให้นักศึกษา เพิ่มโค้ดต่อไปนี้ลงไป ใน Method main()

```
Car c3 = new Car("BMW",60,200,100);  
System.out.println("Brand : "+c3.getBrand()+" v : "+c3.getV());
```

Compile และเขียนผลลัพธ์ที่ได้

Output

4.8 ให้พิมพ์โปรแกรมดังตัวอย่างต่อไปนี้

```
class knight{
    private String name;
    private int HP;
    private int atk;
    knight(String name,int HP,int atk){
        this.name = name;
        this.HP = HP;
        this.atk = atk;
    }
    public void atk(knight target){
        System.out.println(this.name + " attack "+target.name);
        target.HP = target.HP-this.atk;
        if(target.HP<=0){
            System.out.println(target.name+" knockout");
        }
        else{
            System.out.println(target.name+" HP : "+target.HP);
        }
    }
    public int getHP() {
        return HP;
    }
    public void setHP(int HP) {
        this.HP = HP;
    }
    public int getAtk() {
        return atk;
    }
    public void setAtk(int atk) {
        this.atk = atk;
    }
}
```

```
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
}  
public class Lab4 {  
    public static void main(String[] args) {  
        knight rav = new knight("Rav",1000,200);  
        knight jak = new knight("Jak",2000,999);  
        rav.atk(jak);  
        jak.atk(rav);  
        jak.atk(rav);  
    }  
}
```

Compile และเขียนผลลัพธ์ที่ได้

Output

4.9 จากข้อ 4.8 ให้แก้ไข `knight rav = new knight("Rav",1000,200);` เป็น `knight rav = new knight();`

Compile และเขียนผลลัพธ์ที่ได้

Output

## 4.10 ให้พิมพ์โปรแกรมดังต่อไปนี้

```
class knight{
    private String name;
    private int HP;
    private int atk;
    knight(String name,int HP,int atk){
        this.name = name;
        this.HP = HP;
        this.atk = atk;
    }

    public int getHP() {
        return HP;
    }
    public void setHP(int HP) {
        this.HP = HP;
    }
    public int getAtk() {
        return atk;
    }
    public void setAtk(int atk) {
        this.atk = atk;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void checkHP(knight target){
        if(target.HP<=0){
            System.out.println(target.name+" knockout");
        }
        else{
            System.out.println(target.name+" HP : "+target.HP);
        }
    }
    public void atk(knight target){
        System.out.println(this.name + " attack "+target.name);
        target.HP = target.HP-this.atk;
    }
}
```

```
        this.checkHP(target);
    }
    public void bash(knight target,int bashlv){
        System.out.println(this.name + " Bash "+target.name);
        target.HP = target.HP-(this.atk*bashlv);
        this.checkHP(target);
    }
}
public class Lab4 {
    public static void main(String[] args) {
        knight rav = new knight("Ravkung",10000,100);
        knight jak = new knight("Jakkung",200000,20000);
        rav.atk(jak);
        rav.bash(jak, 10);
        rav.bash(jak, 10);
        rav.bash(jak, 10);
        for (int i = 0; i < 10; i++) {
            rav.bash(jak, 10);
        }
        jak.bash(rav, 10);
    }
}
```

Compile และเขียนผลลัพธ์ที่ได้

Output



4.11 จาก 4.10 ให้เพิ่ม Class redpotion ดังนี้

```
class redpotion{
    private int recovery;
    redpotion(){
        this.recovery = 100;
    }
    public int getRecovery() {
        return recovery;
    }
}
```

เพิ่ม Method drink potion ลงใน Class knight

```
public void drinkpotion(redpotion red){
    this.HP = this.HP+red.getRecovery();
    this.checkHP(this);
}
```

และเขียนคำสั่งลงใน Method main ดังนี้

```
public static void main(String[] args) {
    knight rav = new knight("Ravkung",10000,100);
    knight jak = new knight("Jakkung",200000,9999);
    redpotion red = new redpotion();
    rav.atk(jak);
    for (int i = 0; i < 10; i++) {
        rav.bash(jak, 10);
    }
    jak.atk(rav);
    for (int i = 0; i < 10; i++) {
        rav.drinkpotion(red);
    }
    for (int i = 0; i < 10; i++) {
        rav.bash(jak, 10);
    }
    jak.bash(rav, 10);
}
```

ชื่อ-สกุล \_\_\_\_\_

รหัสนักศึกษา \_\_\_\_\_

Compile และเขียนผลลัพธ์ที่ได้

Output

Assignments

1. ให้นักศึกษาออกแบบคลาสตัวละคร RPG และเขียนโปรแกรม โดยให้ออกแบบอาชีพ knight ซึ่งมี Instance และ Method ดังต่อไปนี้

- HP (พลังชีวิต)
- MP (พลังเวทย์)
- Atk (พลังโจมตี)
- Def (พลังป้องกัน)
- Name (ชื่อตัวละคร)

+ โจมตี (เป้าหมาย) โดยการโจมตีจะต้องมีการคำนวณค่า Atk ของเรา และ HP, Def ของศัตรู

+ ทำพิเศษ (เป้าหมาย) โดยทำพิเศษจะต้องมีการคำนวณค่า Atk และ MP ของเรา และ HP, Def ของศัตรู

ให้แสดงว่าชื่อละครที่โจมตีและแสดงตัวละครของเป้าหมายที่ถูกโจมตี และเป้าหมายเหลือ HP อยู่เท่าใดหลังการโจมตี โดยให้ออกแบบคลาส ให้คลาสมีคุณสมบัติ Encapsulation (การห่อหุ้ม) คือมีฟังก์ชัน get และ set ด้วย จากนั้นให้สร้าง Object knight ขึ้นมาอย่างน้อย 2 Objects และมีการใช้ฟังก์ชันโจมตี หรือทำพิเศษ อย่างน้อย 5 ครั้ง

Instructor

2. ให้นักศึกษาออกแบบคลาสรถยนต์ และเขียนโปรแกรม โดยรถยนต์มี Instance และ Method ดังนี้

- v (ความเร็ว)
- Brand (ยี่ห้อรถ)

+ เร่งความเร็ว จะทำให้ความเร็วรถเพิ่มขึ้น โดยให้เพิ่มขึ้นทีละ 1

+ เบรก จะทำให้ความเร็วรถลดลง โดยให้ลดลงทีละ 1

> ตอนเริ่มต้นรถทุกคันจะมีความเร็ว = 1

> ตอนเริ่มต้นรถทุกคันจะมีตำแหน่ง = 0

ให้ออกแบบคลาส โดยคลาสมีคุณสมบัติ Encapsulation คือมีฟังก์ชัน get และ set ด้วย

จากนั้นสร้าง Object Car มาอย่างน้อย 3 Objects และมีการเร่งความเร็ว หรือเบรก อย่างน้อย 10 ครั้ง ในการสั่งคำสั่งแต่ละครั้ง ให้มีการแสดงยี่ห้อรถ ความเร็ว ด้วย

Instructor