



แบบฝึกปฏิบัติ ครั้งที่ 5

เรื่อง

การเขียนโปรแกรมเชิงวัตถุ

วัตถุประสงค์

1. เพื่อฝึกฝนการใช้คุณสมบัติการห่อหุ้ม (Encapsulation)

2. เพื่อฝึกฝนการใช้คุณสมบัติการสืบทอด (Inheritance)

1. ให้นักศึกษาตอบคำถามจากคลาส Person ต่อไปนี้

```
public class Person {  
    private String name;  
    private int age;  
  
    public void setName(String name) {  
        this.name = name;  
    }  
    public String getName() {  
        return name;  
    }  
    public void setAge(int age) {  
        this.age = age;  
    }  
    public int getAge() {  
        return age;  
    }  
}
```

ให้นักศึกษาเขียนโปรแกรมในช่องว่างเพื่อสร้างออบเจกต์ p จากคลาส Person และกำหนดค่า name = "John Doe" และ age = 25 จากนั้น แสดงผลข้อมูลของออบเจกต์ p ดังภาพ

โค้ด

```
public class Main {  
    public static void main(String[] args) {  
        Person p = new Person();  
        p.setName("John Doe");  
        p.setAge(25);  
        System.out.println("This person's name is " + p.getName() + " and they are " + p.getAge() + " years old.");  
    }  
}
```

ตัวอย่างผลลัพธ์

This person's name is John Doe and they are 25 years old.

2. ให้นักศึกษาตอบคำถามจากคลาส Book ต่อไปนี้

```
public class Book {  
    private String title;  
    private String author;  
    private double price;  
  
    public void setTitle(String title) {  
        this.title = title;  
    }  
    public String getTitle() {  
        return title;  
    }  
    public void setAuthor(String author) {  
        this.author = author;  
    }  
    public String getAuthor() {  
        return author;  
    }  
    public void setPrice(double price) {  
        this.price = price;  
    }  
    public double getPrice() {  
        return price;  
    }  
}
```

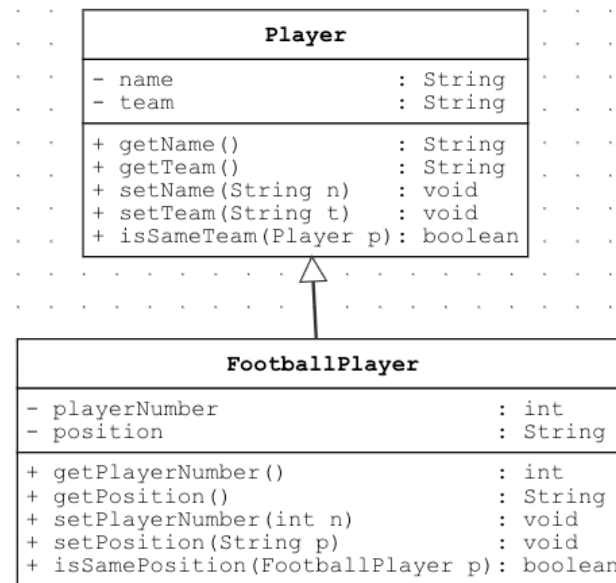
ให้นักศึกษาเขียนโปรแกรมในช่องว่างเพื่อสร้างออบเจกต์ b จากคลาส Book โดยกำหนดค่า title = "The Great Gatsby", author = "F. Scott Fitzgerald", และ price = 10.99 จากนั้น แสดงผลข้อมูลของออบเจกต์ b ดังภาพ
โค้ด

```
public class Main {  
    public static void main(String[] args) {  
        Book b = new Book();  
        b.setTitle("The Great Gatsby");  
        b.setAuthor("F. Scott Fitzgerald");  
        b.setPrice(10.99);  
        System.out.println("\n" + b.getTitle() + "\n by " + b.getAuthor() + " costs " + b.getPrice() +  
            " USD. ");  
    }  
}
```

ตัวอย่างผลลัพธ์

```
"The Great Gatsby" by F. Scott Fitzgerald costs 10.99 USD.
```

3. ให้นักศึกษาร่างคลาส Player และ FootballPlayer ตามคลาสไดอะแกรมต่อไปนี้ โดยอาศัยหลักการห่อหุ้มและการสืบทอดตามโครงสร้างภาษาจาวา



โดยกำหนดให้คลาส Player มีรายละเอียดแต่ละเมธอดดังต่อไปนี้

- เมธอด setName() จะนำค่าพารามิเตอร์ n ไปกำหนดให้แอททริบิวต์ name ขณะที่ เมธอด setTeam() จะนำค่าพารามิเตอร์ t ไปกำหนดให้แอททริบิวต์ team
- เมธอด getName() จะคืนค่าเป็นแอททริบิวต์ name และเมธอด getTeam() จะคืนค่าเป็นแอททริบิวต์ team
- เมธอด isSameTeam() จะคืนค่าเป็น true ก็ต่อเมื่อ ออปเจกต์ของคลาส Player ที่เรียกใช้มีค่าของแอททริบิวต์ team เหมือนกับของออปเจกต์ p ถ้าไม่เหมือนกันจะคืนค่าเป็น false

โดยกำหนดให้คลาส FootballPlayer มีรายละเอียดแต่ละเมธอดดังต่อไปนี้

- เมธอด setPlayerNumber() จะนำค่าพารามิเตอร์ n ไปกำหนดให้แอททริบิวต์ playerNumber ขณะที่ เมธอด setPosition() จะนำค่าพารามิเตอร์ p ไปกำหนดให้แอททริบิวต์ position
- เมธอด getPlayerNumber() จะคืนค่าเป็นแอททริบิวต์ playerNumber และเมธอด getPosition() จะคืนค่าเป็นแอททริบิวต์ position
- เมธอด isSamePosition() จะคืนค่าเป็น true ก็ต่อเมื่อ ออปเจกต์ของคลาส FootballPlayer ที่เรียกใช้มีค่าของแอททริบิวต์ team และ position เหมือนกับของออปเจกต์ p ที่รับเข้ามา ถ้าไม่เหมือนกันจะคืนค่าเป็น false

กำหนดโค้ดสำหรับทดสอบความถูกต้องของคลาส **Player** ที่นักศึกษาได้พัฒนาขึ้น

กรณีที่ 1

```
public class Main {  
    public static void main(String[] args) {  
        Player p1 = new Player();  
        p1.setName("Bank");  
        p1.setTeam("Gate OR");  
  
        Player p2 = new Player();  
        p2.setName("Khim");  
        p2.setTeam("Gate OR");  
  
        if(p1.isSameTeam(p2))  
            System.out.println(p1.getName() + " is a same team with " + p2.getName());  
        else  
            System.out.println(p1.getName() + " is not a same team with " + p2.getName());  
    }  
}
```

ตัวอย่างผลลัพธ์

Bank is a same team with Khim

กรณีที่ 2

```
public class Main {  
    public static void main(String[] args) {  
        Player p1 = new Player();  
        p1.setName("Bank");  
        p1.setTeam("Gate OR");  
        System.out.println(p1.name);  
    }  
}
```

ตัวอย่างผลลัพธ์

Main.java:6: error: name has private access in Player

เนื่องจากสาเหตุใดทำให้จึงเกิดข้อความ Error ดังกล่าว

เพราะเนื่องจาก Attribute ของ Class ไม่ได้เป็น Public แต่เป็น Private
ซึ่งถ้าเป็น Public (ทั้งตัว Class และ Method) ก็สามารถใช้เรียกชื่อได้

กำหนดโค้ดสำหรับทดสอบความถูกต้องของคลาส **FootballPlayer** ที่นักศึกษาได้พัฒนาขึ้น

กรณีที่ 1

```
public class Main {
    public static void main(String[] args) {
        FootballPlayer p1 = new FootballPlayer();
        p1.setName("Harry");
        p1.setTeam("Gryffindor");
        p1.setPlayerNumber(1);
        p1.setPosition("keeper");
        FootballPlayer p2 = new FootballPlayer();
        p2.setName("Jame");
        p2.setTeam("Gryffindor");
        p2.setPlayerNumber(1);
        p2.setPosition("keeper");

        System.out.println("We are same position : " + p1.isSamePosition(p2));
        System.out.println("We are same team : "+ p1.isSameTeam(p2));
    }
}
```

ตัวอย่างผลลัพธ์

```
We are same position : true
We are same team : true
```

กรณีที่ 2

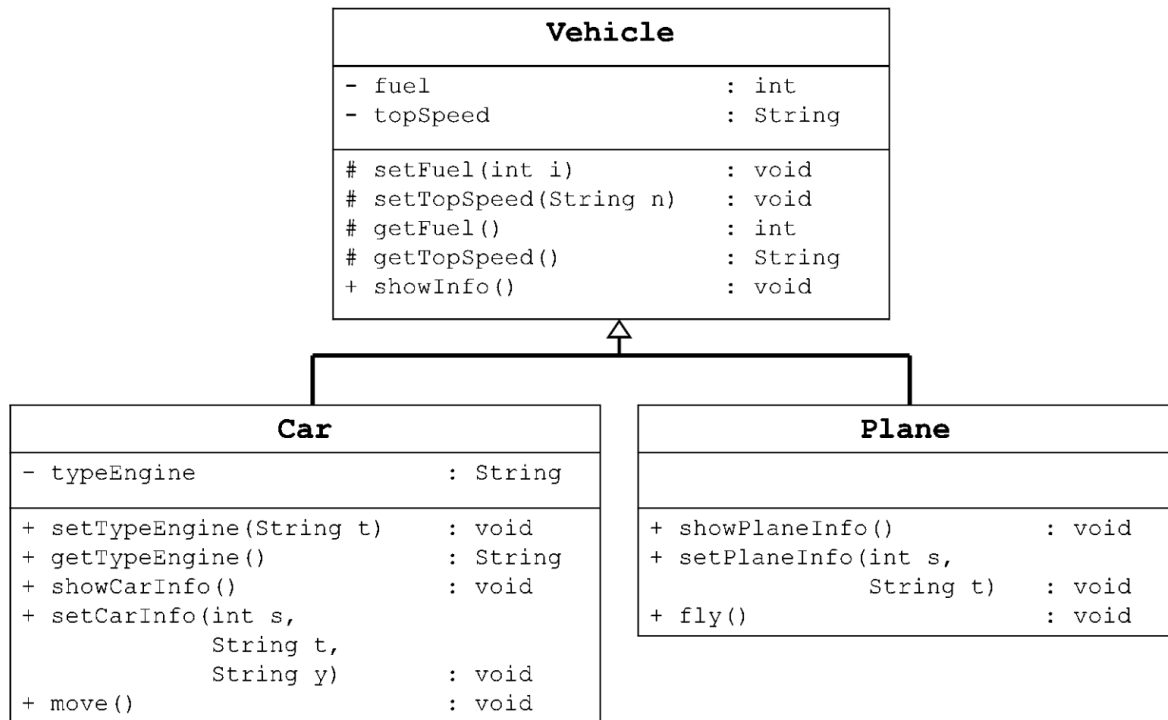
```
public class Main {
    public static void main(String[] args) {
        FootballPlayer p1 = new FootballPlayer();
        p1.setName("Harry");
        p1.setTeam("Gryffindor");
        p1.setPlayerNumber(1);
        p1.setPosition("keeper");
        FootballPlayer p2 = new FootballPlayer();
        p2.setName("Jame");
        p2.setTeam("Gryffindor");
        p2.setPlayerNumber(1);
        p2.setPosition("fullback");

        System.out.println("We are same position : " + p1.isSamePosition(p2));
        System.out.println("We are same team : "+ p1.isSameTeam(p2));
    }
}
```

ตัวอย่างผลลัพธ์

```
We are same position : false
We are same team : true
```

4. ให้นักศึกษาร่างคลาส Vehicle, Car และ Plane ตามคลาสไดอะแกรมต่อไปนี้ โดยอาศัยหลักการห่อหุ้มและการสืบทอดตามโครงสร้างภาษาจาวา



หมายเหตุ เครื่องหมาย # ในคลาสไดอะแกรมบ่งบอกถึง Access Modifier ประเภท **protected** โดยกำหนดให้แต่ละเมธอดมีรายละเอียดดังต่อไปนี้

คลาส **Vehicle**

- เมธอด `setFuel(int i)` จะนำค่าพารามิเตอร์ `i` ไปกำหนดให้แอตทริบิวต์ `fuel` ขณะที่ เมธอด `setTopSpeed(String n)` จะนำค่าพารามิเตอร์ `n` ไปกำหนดให้แอตทริบิวต์ `topSpeed`
- เมธอด `getFuel()` จะคืนค่าเป็นแอตทริบิวต์ `fuel` ขณะที่ เมธอด `getTopSpeed()` จะคืนค่าเป็นแอตทริบิวต์ `topSpeed`
- เมธอด `showInfo()` จะแสดงค่าของแต่ละแอตทริบิวต์ดังตัวอย่าง

Fuel is [ค่าจากแอตทริบิวต์ `fuel`] litre and Top Speed is [ค่าจากแอตทริบิวต์ `topSpeed`] m/s.

คลาส **Car**

- เมธอด `setTypeEngine(String t)` จะนำค่าพารามิเตอร์ `t` ไปกำหนดให้แอตทริบิวต์ `typeEngine` ขณะที่ เมธอด `getTypeEngine()` จะคืนค่าเป็นแอตทริบิวต์ `typeEngine`
- `setCarInfo(int s, String t, String y)` จะนำค่าพารามิเตอร์ `s` ไปกำหนดให้แอตทริบิวต์ `fuel`, ค่าพารามิเตอร์ `t` ไปกำหนดให้แอตทริบิวต์ `topSpeed` และค่าพารามิเตอร์ `y` ไปกำหนดให้แอตทริบิวต์ `typeEngine`
- เมธอด `move()` จะดำเนินการพิมพ์ข้อความต่อไปนี้ทางจอภาพ จากนั้น ลดค่าแอตทริบิวต์ `fuel` ลง 50

Move.

กรณีค่าแอมเพริส fuel ไม่เพียงพอให้ห้จะแสดงข้อความต่อไปนี้แทน

Please add fuel.

- เมธอด showCarInfo() จะแสดงค่าของแต่ละแอมเพริสดังตัวอย่าง

Car engine is [ค่าจากแอมเพริส typeEngine].

Fuel is [ค่าจากแอมเพริส fuel] litre and Top Speed is [ค่าจากแอมเพริส topSpeed] m/s.

คลาส **Plane**

- setPlaneInfo(int s, String t) จะนำค่าพารามิเตอร์ s ไปกำหนดให้แอมเพริส fuel และค่าพารามิเตอร์ t ไปกำหนดให้แอมเพริส topSpeed
- เมธอด fly() จะดำเนินการพิมพ์ข้อความต่อไปนี้ทางจอภาพ จากนั้น ลดค่าแอมเพริส fuel ลง 200

Fly.

กรณีค่าแอมเพริส fuel ไม่เพียงพอให้ห้จะแสดงข้อความต่อไปนี้แทน

Please add fuel.

- เมธอด showPlaneInfo() จะแสดงค่าของแต่ละแอมเพริสดังตัวอย่าง

Plane detail is, Fuel is [ค่าจากแอมเพริส fuel] litre and Top Speed is [ค่าจากแอมเพริส topSpeed] m/s.

กำหนดโค้ดสำหรับทดสอบความถูกต้องของคลาสข้างต้นที่นักศึกษาได้พัฒนาขึ้น

กรณีที่ 1

```
class Main {
    public static void main(String[] args) {
        Plane p1 = new Plane();
        p1.setPlaneInfo(500, "High");
        p1.showPlaneInfo();
        Car c1 = new Car();
        c1.setCarInfo(500, "High", "Diesel");
        c1.showCarInfo();
    }
}
```

ตัวอย่างผลลัพธ์

Plane detail is, Fuel is 500 litre and Top Speed is High m/s.
Car engine is Diesel.
Fuel is 500 litre and Top Speed is High m/s.

กรณีที่ 2

```
public class Main {  
    public static void main(String[] args) {  
        Plane p1 = new Plane();  
        p1.setPlaneInfo(300, "High");  
        p1.showPlaneInfo();  
        p1.fly();  
        p1.showPlaneInfo();  
        p1.fly();  
        p1.showPlaneInfo();  
    }  
}
```

ตัวอย่างผลลัพธ์

```
Plane detail is, Fuel is 300 litre and Top Speed is High m/s.  
Fly.  
Plane detail is, Fuel is 100 litre and Top Speed is High m/s.  
Please add fuel.  
Plane detail is, Fuel is 100 litre and Top Speed is High m/s.
```

กรณีที่ 3

```
public class Main {  
    public static void main(String[] args) {  
        Car c1 = new Car();  
        c1.setCarInfo(60, "High", "Diesel");  
        c1.showCarInfo();  
        c1.move();  
        c1.showCarInfo();  
        c1.move();  
        c1.showCarInfo();  
    }  
}
```

ตัวอย่างผลลัพธ์

```
Car engine is Diesel.  
Fuel is 60 litre and Top Speed is High m/s.  
Move.  
Car engine is Diesel.  
Fuel is 10 litre and Top Speed is High m/s.  
Please add fuel.  
Car engine is Diesel.  
Fuel is 10 litre and Top Speed is High m/s.
```