แนวคิดและหลักการพื้นฐานการโปรแกรมเชิงวัตถุ

การเขียนโปรแกรมเชิงวัตถุคืออะไร

(Object-Oriented Programming หรือ OOP)

- คือ รูปแบบหนึ่งของการเขียนโปรแกรมที่ให้น้ำหนักกับการจำลองสิ่งต่างๆ ในโลกจริงให้อยู่ในรูปแบบของ "วัตถุ" (**Object**) ซึ่งมีทั้ง "สถานะ" (**State**) และ "พฤติกรรม" (**Behavior**) เป็นของตนเอง ผ่านการใช้ประโยชน์จากหลักการหลักๆ 4 ประการ ดังต่อไปนี้:

1. Encapsulation (การห่อหุ้ม)
2. Abstraction (นามธรรม)
3. Inheritance (การสืบทอด)
4. Polymorphism (การมีหลายรูปแบบ)

1.Encapsulation (การห่อหุ้ม)

การห่อหุ้มเป็นการซ่อนรายละเอียดกระบวนการภายใน ปกป้องข้อมูลและการทำงานของวัตถุ โดยอนุญาตให้ส่วนของวัตถุที่ "เปิดเผย" (Public Interface) ให้สามารถเข้าถึงได้จากภายนอก นี่เปรียบเสมือนกับการใช้รีโมตควบคุมโทรทัศน์ เราไม่จำเป็นต้องรู้ว่ามันทำงานอย่างไรภายใน แต่เรารู้ว่าเมื่อกดปุ่มเปิด/ปิด โทรทัศน์ก็จะทำงานตามที่เราต้องการ

```
public class Television {
    private boolean isOn = false;

    public void togglePower() {
        isOn = !isOn;
    System.out.println("Television is " + (isOn ? "on" :
    "off"));
    }
}
```

2. Abstraction (นามธรรม)

นามธรรมเป็นการย่อยส่วนที่ซับซ้อนของวัตถุเป็นสิ่งที่เราสามารถจัดการได้ง่ายขึ้น การทำให้นามธรรมคือการระบุอย่างจำเพาะว่า "อะไร" คือสิ่งที่วัตถุนั้นทำได้ โดยไม่ต้องระบุว่า "อย่างไร" ในการทำงานนั้นๆ ซึ่งเหมือนกับการขับรถ เราไม่จำเป็นต้องรู้ทุกอย่างที่เกิดขึ้นภายในเครื่องยนต์ แต่เรารู้ว่าเมื่อเราหมุนพวงมาลัยไปทางไหน รถก็จะเคลื่อนที่ไปในทิศทางนั้น

```
public interface Vehicle {
    void turnLeft();
    void turnRight();
    void accelerate();
    void brake();
}
```

3. Inheritance (การสืบทอด)

การสืบทอดเปรียบเสมือนการรับมรดกในครอบครัว โดย subclass สามารถรับพฤติกรรมและสถานะของ superclass มา ซึ่งทำให้เราไม่จำเป็นต้องเขียนโค้ดซ้ำซ้อน และยังสามารถขยายหรือแก้ไขพฤติกรรมเฉพาะที่จำเป็นได้

```
public class Car extends Vehicle {
// Car inherits all behaviors from Vehicle and can add its own specific behaviors.
}
public class Bicycle extends Vehicle {
// Bicycle also inherits from Vehicle but will have different implementations for some behaviors.
}
```

4. Polymorphism (การมีหลายรูปแบบ)

Polymorphism คือความสามารถของโค้ดในการมีหลายรูปแบบ โดยวัตถุต่างๆ ที่มาจากคลาสเดียวกันหรือคลาสที่มีสัมพันธ์กัน สามารถถูกใช้งานอย่างสลับโค้ดกันได้ ทำให้โปรแกรมมีความยืดหยุ่นและสามารถนำไปใช้ในสถานการณ์ที่มีความต้องการที่หลากหลาย

```
Vehicle myCar = new Car();
Vehicle myBike = new Bicycle();
```

สืบค้นที่

http://m.me/Expert.Programming.Tutor