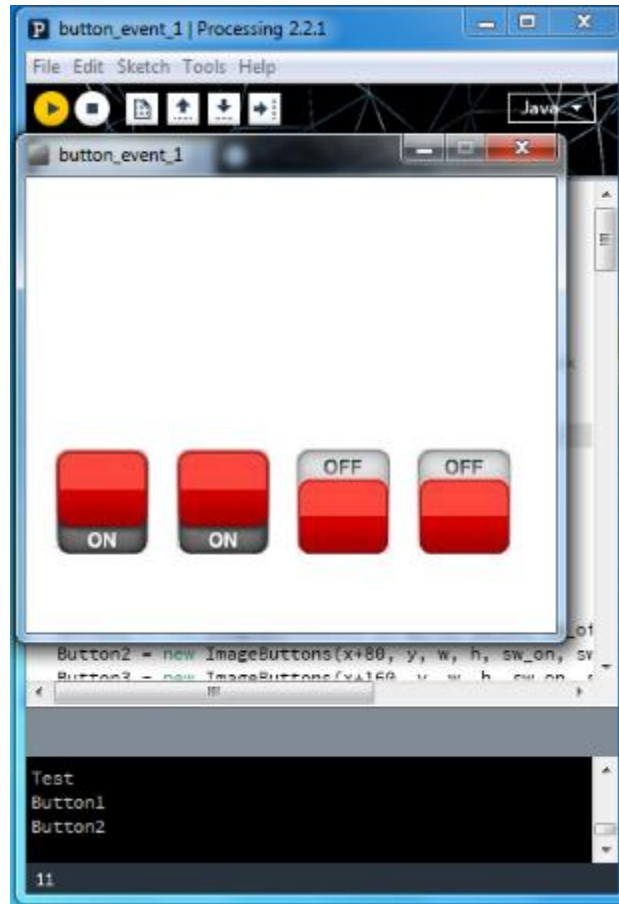


## Object-Oriented Programming in Processing



### สารบัญ

1	การโปรแกรมแบบเชิงวัตถุ Object Oriented Programming (OOP) คืออะไร.....	2
2	ความหมายของวัตถุในโปรแกรมแบบ OOP.....	2
3	การทดลอง .....	2
3.1	โปรแกรมที่ 1 รถแข่ง .....	3
3.2	โปรแกรมที่ 2.....	5
3.3	งานมอบหมาย .....	9

## 1 การโปรแกรมแบบเชิงวัตถุ Object Oriented Programming (OOP) คืออะไร

การเขียนโปรแกรมเชิงวัตถุ เป็นเทคนิคการเขียนโปรแกรมที่เน้นความคิดเชิงวัตถุ (object) ที่สร้างขึ้นใช้งานในโปรแกรม โดยคำว่า "วัตถุ" ในที่นี้คือ ส่วนย่อยๆ ของโปรแกรมที่ผู้พัฒนาโปรแกรมสร้างขึ้นเพื่อทำงานเฉพาะอย่าง แล้วจึงนำวัตถุย่อยๆ เหล่านั้นมาประกอบกันเป็น โปรแกรมใหญ่ อีกทั้งวัตถุที่สร้างขึ้นมาแล้วสามารถนำกลับไปใช้กับโปรแกรมอื่นได้อีก โดยบางครั้งผู้เขียนโปรแกรมไม่จำเป็นต้องสร้างวัตถุเองทุกชิ้น สามารถนำวัตถุที่ผู้อื่นสร้างไว้มาใช้ใหม่ได้ เพียงแค่รู้ว่าวัตถุนั้นทำหน้าที่อะไร เรียกใช้งานอย่างไร ทำให้ โปรแกรมเมอร์ที่เขียนโปรแกรมประเภทนี้สามารถปรับปรุงเปลี่ยนแปลงโปรแกรมได้สะดวก การเขียนโปรแกรมแบบนี้จึงเข้ามาแทนที่การเขียนโปรแกรมแบบโครงสร้างอย่างรวดเร็ว โดยเฉพาะอย่างยิ่งกับงานที่เกี่ยวข้องกับอินเทอร์เน็ต นอกจากนี้ การโปรแกรมแบบเชิงวัตถุยังสามารถทำงานกับข้อมูลได้หลายรูปแบบ ทั้งที่เป็นรูปภาพ (image) วิดีทัศน์ (video) หรือเสียง (sound) ตัวอย่าง ภาษาคอมพิวเตอร์ที่ใช้เทคนิคการโปรแกรมเชิงวัตถุ เช่น ภาษาจาวา ภาษาซีชาร์ป รวมถึง Processing ก็สามารถใช้นิวคิดการเขียนโปรแกรมเชิงวัตถุได้อย่างดี

## 2 ความหมายของวัตถุในโปรแกรมแบบ OOP

เพื่อให้เห็นภาพของวัตถุได้ชัดเจนยิ่งขึ้นขอให้พิจารณาตัวอย่างดังต่อไปนี้

ส่วนประกอบของวัตถุมี 2 ส่วน คือ Data และ Function

- Data ลักษณะเฉพาะของวัตถุ เช่น Datta ของ มนุษย์ ได้แก่ ความสูง น้ำหนัก สีผม สีตา หรือ ถ้าเป็น Data ของรถยนต์ก็ได้แก่ สี ขนาด ความเร็ว
- Function หรือความสามารถในการทำงาน ของมนุษย์ได้แก่ นอน ตื่น กิน วิ่ง เดิน ของรถยนต์ก็ได้แก่ การเคลื่อนที่

## 3 การทดลอง

ตัวอย่างการทดลองมีอยู่ 2 โปรแกรม โปรแกรมแรก เป็นโปรแกรมให้แสดงรถแข่ง ซึ่งเขียนเป็นวัตถุรูปรถที่สามารถเคลื่อนที่จากด้านซ้ายของจอภาพไปยังด้านขวาของจอภาพ โดยสามารถกำหนดสีรถ ตำแหน่งรถในแนวแกน Y ความเร็วของรถแต่ละคัน รวมถึงจำนวนรถได้หลายคัน โปรแกรมเพียงแต่สร้างวัตถุที่เป็นรถ หรือเรียกว่า Class ของรถไว้ให้ แล้วเราก็กำหนดรถยนต์แต่ละคันขึ้นมาใหม่ ด้วยคำสั่ง new เท่านั้น ส่วนโปรแกรมที่ 2 แสดงการสร้าง วัตถุหลักหรือ Class หลักแล้วจะสร้าง Class ย่อย จาก Class หลักนี้ โดยเขียนตัวอย่างเป็นภาพของสวิทช์ที่สามารถปิด-เปิด ได้ด้วยการคลิกเมาส์ โปรแกรมนี้ใช้รูปภาพของสวิทช์เป็นการแสดงภาพ ในโปรแกรมจะสร้างวัตถุที่เรียกว่า Button เป็น Classs หลักและสร้าง Class ImageButton เป็น Class ที่ขยายจาก Class Button ด้วยคำสั่ง class ImageButtons extends Button

### 3.1 โปรแกรมที่ 1 รถแข่ง

โปรแกรมนี้นำมาจาก Learning Processing โดย Daniel Shiffman

เว็บ <http://www.learningprocessing.com>

เป็นตัวอย่างการเขียนโปรแกรมเชิงวัตถุ แบบง่ายๆ โดยใช้สัญลักษณ์กล่องสี่เหลี่ยม แทนรถแข่ง // การสร้าง วัตถุ (Object) ใช้คำสั่ง class โดยระบุ Data (คุณลักษณะประจำตัวของวัตถุ) และ Function (ความสามารถ ของวัตถุ)

การสร้าง Class car (object car)

```
class Car
{
  color c;           //กำหนดค่าตัวแปรที่ใช้เป็น Data
  float xpos;
  float ypos;
  float xspeed;

  Car(color tempC, float tempXpos, float tempYpos, float tempXspeed)
  { // The Constructor is defined with arguments.
    c = tempC;
    xpos = tempXpos;           //ระบุ data หรือคุณลักษณะของ car
    ypos = tempYpos;
    xspeed = tempXspeed;      //เพื่อใช้ในการสร้างรูปรถ
  }

  void display()              //ฟังก์ชันการวาดรูปรถ
  {
    stroke(0);                //วาดรูปรถด้วยกล่องสี่เหลี่ยมพื้นดำ
    fill(c);                   //แล้วใส่สี รถ ตำแหน่งรถ ตาม data
    rectMode(CENTER);
    rect(xpos,ypos,20,10);
  }

  void move()                  //ฟังก์ชันแสดงการเคลื่อนที่ของรถ
  {
    xpos = xpos + xspeed;     //โดยใช้ data xpos กับ xspeed
    if (xpos > width)
    {
      xpos = 0;
    }
  }
}
```

การเขียนโปรแกรมเพื่อสร้าง car จาก class ที่สร้างขึ้นมา มีขั้นตอนดังนี้

ขั้นตอนที่ 1 Declare an object. บอกว่าตัวแปรต่อไปนี้มี class เป็น car (เป็น object car)

Car myCar;

ขั้นตอนที่ 2 **Initialize object.** บอกว่าตัวแปรที่ declare ขึ้นมา เป็น car กันใหม่ ใช้คำสั่ง new

```
myCar = new Car();
```

ขั้นตอนนี้อยู่ใน setup

ขั้นตอนที่ 3 เรียกใช้ methods (ฟังก์ชัน) ต่างๆ ของ object นั้นๆ ขั้นตอนนี้ส่วนใหญ่อยู่ใน draw

โปรแกรมสมบูรณ์เขียนดังนี้

```
//สร้างรถ 4 คัน
Car myCar1;
Car myCar2;
Car myCar3;
Car myCar4;

void setup() {
  size(200,200);
  //รถแต่ละคันมีลักษณะไม่เหมือนกัน
  myCar1 = new Car(color(255,0,0),0,30,2); //คันนี้ สีแดง อยู่ที่ตำแหน่ง x = 0 y = 30 ความเร็ว 2
  myCar2 = new Car(color(0,0,255),0,60,1); //คันนี้ สีน้ำเงิน อยู่ที่ตำแหน่ง x = 0 y = 60 ความเร็ว 1
  myCar3 = new Car(color(0,255,0),0,90,4); //คันนี้ สีเขียว อยู่ที่ตำแหน่ง x = 0 y = 90 ความเร็ว 4
  myCar4 = new Car(color(255,255,0),0,120,3); //คันนี้ สีเหลือง อยู่ที่ตำแหน่ง x = 0 y = 120 ความเร็ว 3
}

void draw() {
  background(255); //สีพื้นเป็นสีขาว
  myCar1.move(); // car1 เคลื่อนที่ ใช้ method move
  myCar1.display(); // แสดงรูปรถ car1
  myCar2.move(); // car2 เคลื่อนที่ ใช้ method move
  myCar2.display(); // แสดงรูปรถ car2
  myCar3.move(); // car3 เคลื่อนที่ ใช้ method move
  myCar3.display(); // แสดงรูปรถ car3
  myCar4.move(); // car4 เคลื่อนที่ ใช้ method move
  myCar4.display(); // แสดงรูปรถ car4
}

class Car {
  color c;
  float xpos;
  float ypos;
  float xspeed;

  Car(color tempC, float tempXpos, float tempYpos, float tempXspeed) { // The Constructor is defined
  with arguments.
    c = tempC;
    xpos = tempXpos;
    ypos = tempYpos;
    xspeed = tempXspeed;
  }

  void display() {
    stroke(0);
    fill(c);
    rectMode(CENTER);
  }
}
```

```

    rect(xpos,ypos,20,10);
}

void move() {
    xpos = xpos + xspeed;
    if (xpos > width) {
        xpos = 0;
    }
}
}
}

```

### 3.2 โปรแกรมที่ 2

วาดรูปภาพสวิทช์ เมื่อคลิกเมาส์ที่สวิทช์จะแสดงสถานการณ์ปิดหรือเปิด บนพื้นที่ console ของโปรแกรม สำหรับโปรแกรมนี้อาจวาดรูปภาพสวิทช์และหลอดไฟให้ไปดาวน์โหลดได้ที่

<http://narong.ece.engr.tu.ac.th/microlab/document/index.php> รายการที่ 15

โปรแกรมนี้อเริ่มต้นด้วยการสร้าง class Button เป็น class หลัก แล้วจะมีการขยาย class เป็น class ลูกชื่อ ImageButtons ด้วยการใส่คำสั่ง class ImageButtons extends Button

#### Class หลัก หรือวัตถุหลัก

```

class Button //สร้างวัตถุใหม่ ชื่อ Button
{
    int x, y; //Data ของ class นี้ประกอบด้วย ตำแหน่งที่จะวางรูปสวิทช์บนหน้าต่าง
    int w, h; //ขนาดของรูปสวิทช์
    boolean over = false; //สถานะของเมาส์เหนือรูป เริ่มต้นกำหนดให้เป็นเท็จ
    boolean pressed = false; //สถานะการกดเมาส์ที่รูป เริ่มต้นกำหนดให้เป็นเท็จ

    void pressed() { //ฟังก์ชันตรวจสอบการกดเมาส์ที่รูปสวิทช์
        if(over && mousePressed) { //ถ้าเมาส์อยู่เหนือรูปและกด ให้สถานะ pressed เป็นจริง
            pressed = true;
        } else { //ถ้าไม่ใช่ ให้สถานะ pressed เป็นเท็จ
            pressed = false;
        }
    }
}

boolean overRect(int x, int y, int width, int height) { //วิธีการตรวจสอบว่าเมาส์อยู่เหนือรูป
    if (mouseX >= x && mouseX <= x+width &&
        mouseY >= y && mouseY <= y+height) {
        return true;
    } else {
        return false;
    }
}
}
}

```

สร้าง Class ใหม่ หรือวัตถุใหม่ชื่อ ImageButtons ซึ่งขยายมาจาก วัตถุเก่า Button

```

class ImageButtons extends Button
{
    PImage img_on;
    PImage img_off;
}

```

```

PImage down;
PImage currentimage;
String textoff;
String texton;

ImageButtons(int ix, int iy, int iw, int ih, PImage button_on, PImage button_off,String t_off, String
t_on)
{
  x = ix;           //ตำแหน่งภาพสวิตช์ แนวแกน x
  y = iy;           //ตำแหน่งภาพสวิตช์ แนวแกน y
  w = iw;           //ขนาดความกว้างของภาพสวิตช์
  h = ih;           //ขนาดความสูงของภาพสวิตช์
  img_on = button_on; //ภาพสวิตช์ on
  img_off = button_off; //ภาพสวิตช์ off
  currentimage = img_off; //ภาพสวิตช์ในปัจจุบันเริ่มต้นให้เป็นภาพ off
  textoff = t_off; //ข้อความที่จะให้พิมพ์เมื่อสวิตช์ off
  texton = t_on; //ข้อความที่จะให้พิมพ์เมื่อสวิตช์ on
}

void update(){ //ฟังก์ชัน update ภาพ
  over(); //ตรวจสอบว่าเมาส์อยู่เหนือภาพหรือไม่
  pressed(); //ตรวจสอบว่ากดเมาส์หรือไม่

  if(pressed) //ถ้ากดให้แสดงการกระทำโดยเปลี่ยนรูปสวิตช์และพิมพ์ข้อความ
  {
    if(currentimage==img_on)
    {
      currentimage=img_off;
      println(textoff);
    }
    else
    {
      currentimage=img_on;
      println(texton);
    }
    delay(200);
  }
}

void over(){ //ฟังก์ชันตรวจสอบว่าเมาส์อยู่เหนือภาพหรือไม่
  if( overRect(x, y, w, h) ) {
    over = true;
  } else {
    over = false;
  }
}

void display() { //ตรวจสอบแสดงภาพ
  image(currentimage, x, y);
}
}

```

โปรแกรมทั้งหมด

```

//--- define variable and constant
ImageButtons Button1,Button2;

```

```

void setup() {
  size(300, 300);
  smooth();
  background(255); // Set the background to black
  // Define and create image button
  PImage sw_on = loadImage("sw_on.JPG");
  PImage sw_off = loadImage("sw_off.JPG");

  int x = width/20;
  int y = height - (sw_on.height)-50;
  int w = sw_on.width;
  int h = sw_on.height;

  Button1 = new ImageButtons(x, y, w, h, sw_on, sw_off,"button1_off","button1_on");
  Button2 = new ImageButtons(x+80, y, w, h, sw_on, sw_off,"button2_off","button2_on");

  stroke(255); // Set stroke color to white
  println("Test");
}

void draw()
{
  Button1.update(); Button1.display();
  Button2.update(); Button2.display();
}

class Button
{
  int x, y;
  int w, h;
  boolean over = false;
  boolean pressed = false;

  void pressed() {
    if(over && mousePressed) {
      pressed = true;
    } else {
      pressed = false;
    }
  }
}

boolean overRect(int x, int y, int width, int height) {
  if (mouseX >= x && mouseX <= x+width &&
      mouseY >= y && mouseY <= y+height) {
    return true;
  } else {
    return false;
  }
}

//-----
class ImageButtons extends Button
{
  PImage img_on;
  PImage img_off;
  PImage down;
}

```

```

PImage currentimage;
String textoff;
String texton;

ImageButtons(int ix, int iy, int iw, int ih, PImage button_on, PImage button_off,String t_off, String
t_on)
{
  x = ix;
  y = iy;
  w = iw;
  h = ih;
  img_on = button_on;
  img_off = button_off;
  currentimage = img_off;
  textoff = t_off;
  texton = t_on;
}

void update(){
  over();
  pressed();
  if(pressed)
  {
    if(currentimage==img_on)
    {
      currentimage=img_off;
      println(textoff);
    }
    else
    {
      currentimage=img_on;
      println(texton);
    }
    delay(200);
  }
}

void over(){
  if( overRect(x, y, w, h) ) {
    over = true;
  } else {
    over = false;
  }
}

void display() {
  image(currentimage, x, y);
}
}

```



#### 4 งานมอบหมาย

- 1) ให้เปลี่ยนจากรูปสวิทช์เป็นหลอดไฟ เมื่อแตะหลอดไฟแล้วให้เปลี่ยนเป็นหลอดไฟติด หลอดไฟดับสลับกันไป
- 2) เปลี่ยนจากรถแข่ง เป็นเต่าแข่งกันหรือภาพสัตว์อะไรก็ได้แข่งกัน
- 3) ให้เขียน Obj muj เป็นอุปกรณ์อินพุทหรือเอาต์พุทมาคนละ 1 อย่าง ตัวอย่างของ Labview

