



รายงาน

เรื่อง เอกสารประกอบการออกแบบและแนวคิดในการพัฒนาโครงการ

คณะผู้จัดทำ

1. นาย อภิวัฒน์ เกษสาวก์ รหัสนักศึกษา 6506021620211
2. นายธีรวัช องค์กรีชา รหัสนักศึกษา 6506021610053
3. นาย ปุณณที ปิ่นวิเศษ รหัสนักศึกษา 6506021630038

เสนอ

ผู้ช่วยศาสตราจารย์ สมชัย เชียงพงศ์พันธุ์

รายงานนี้เป็นส่วนหนึ่งของการศึกษารายวิชา OBJECT ORIENTED PROGRAMMING

ภาคเรียนที่ 2 ปี การศึกษา 2565

ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีและการจัดการอุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

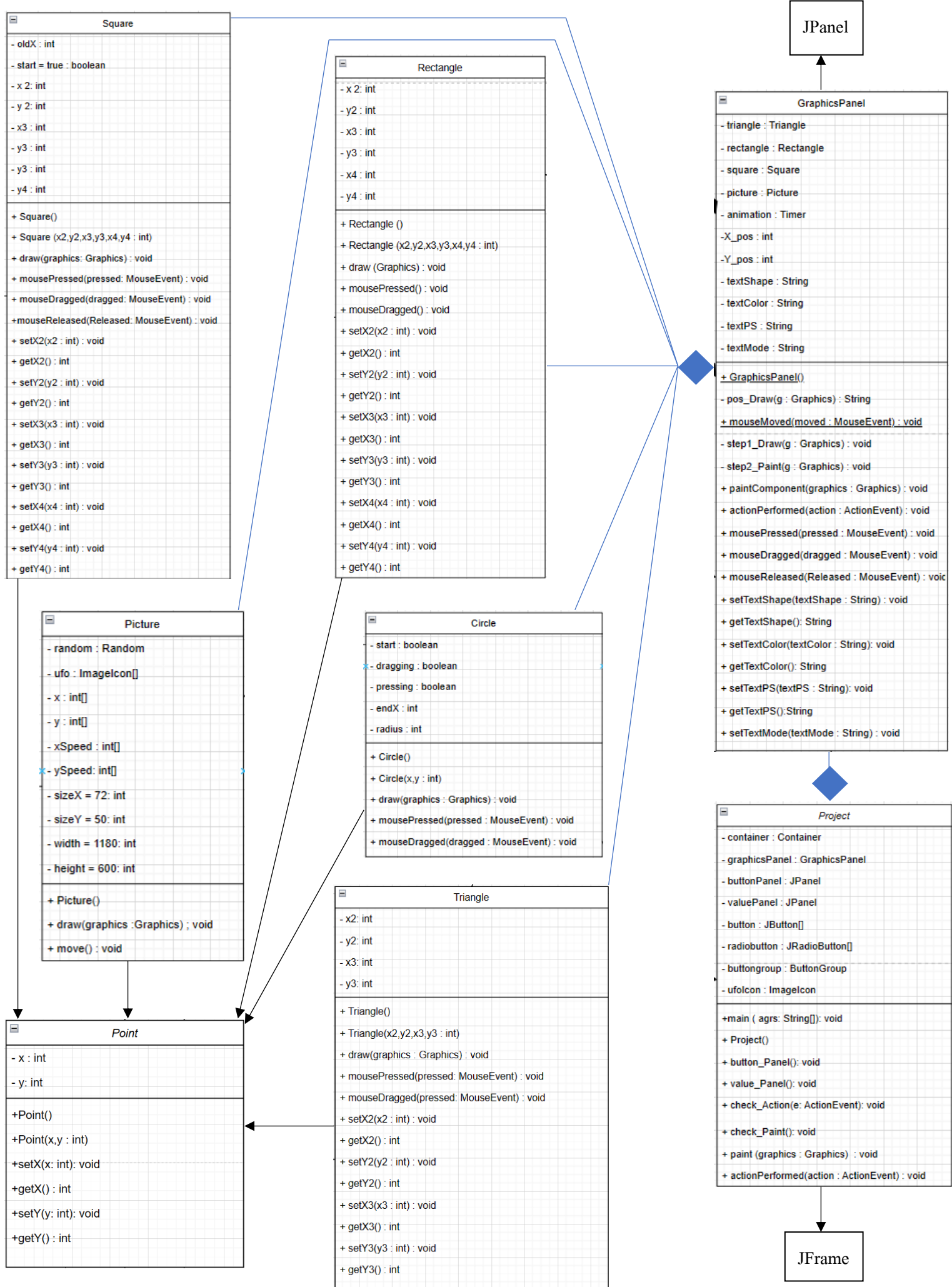
คำนำ

รายงาน เอกสารประกอบการออกแบบและแนวคิดในการพัฒนาโครงการ เล่มนี้เป็นส่วนหนึ่งของวิชา OBJECT ORIENTED PROGRAMMING ซึ่งมีจุดประสงค์เพื่อศึกษาอธิบายเนื้อหาหลักการทำงานของโปรแกรม Java ที่นำมาใช้ในการพัฒนาโปรแกรม “Java Application” โดยใช้การทำงานและรูปแบบของ ตัวโค้ด มาใช้ในการทำงาน

กลุ่มข้าพเจ้าขอกล่าวขอบคุณอาจารย์ สมชัย เชียงพงศ์พันธุ์ ที่ได้ทำการสอนเรื่องนี้ให้แก่กลุ่มข้าพเจ้าและกล่าวขอบคุณเพื่อนๆ ที่ให้การสนับสนุนแก่กลุ่มข้าพเจ้า จนทำให้รายงานเล่มนี้เสร็จสมบูรณ์ และหวังว่าคงเป็นประโยชน์ต่อผู้ใดในการศึกษา

นายอภิวัฒน์ เกษสาวงค์ และ คณะผู้จัดทำ

4 เมษายน 2566



สารบัญ

Class Project	1
Method Project	2
Method button_Panel	3
Method value_Panel	4
Method check_Action	5
Method check_Paint	6
Method paint และ actionPerformed	7
Class GraphicsPanel	8
Method GraphicsPanel	9
Method pos_Draw	9
Method mouseMoved	9
Method step1_Draw	10
Method step2_Paint	11
Method paintComponent และ actionPerformed	12
Method check_mousePressed	12
Method mousePressed	13
Method check_mouseDragged	13
Method mouseDragged	13
Method setter and getter	13
Class Triangle	15
Method Triangle	15
Method draw	15
Method mousePressed	17
Method mouseDragged	18

Class Square	19
Method Square	19
Method draw	19
Method mousePressed	22
Method mouseDragged	23
Class Rectangle	24
Method Rectangle	24
Method draw	24
Method mousePressed	26
Method mouseDragged	27
Class Circle	28
Method Circle	28
Method draw	28
Method mousePressed	29
Method mouseDragged	30
Class Picture	31
Method Picture	32
Method draw	32
Method move	33
Class Point	34
Method setter and getter	35
หน้าจอโปรแกรม	36

Class Project

```

1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4
5  public class Project extends JFrame implements ActionListener {
6      private Container container;
7      private GraphicsPanel graphicsPanel;
8
9      private JPanel buttonPanel, valuePanel;
10     private JButton[] button = new JButton[12];
11     private JRadioButton[] radiobutton = new JRadioButton[4];
12     private ButtonGroup buttongroup;
13     private ImageIcon ufoIcon;
14
15     Run | Debug
16     public static void main(String args[]) {new Project();}

```

implements ActionListener เข้ามาเพื่อเช็คการทำงานหลังจากการคลิกปุ่ม

Attribute จะถูกสร้างอยู่ในช่วงบรรทัดที่ 6 – 13

ตัวแปรและชนิดตัวแปร จะถูกสร้างเป็น private โดยจะมีทั้งหมด ดังนี้

บรรทัดที่ 6 กำหนดวัตถุที่สามารถเก็บวัตถุอื่นๆไว้ภายในได้ : ชื่อตัวแปร container

บรรทัดที่ 7 กำหนดคลาส GraphicsPanel เพื่อเตรียมเรียกใช้งาน

บรรทัดที่ 9 กำหนด Panel ไว้เก็บวัตถุต่างๆ : ชื่อตัวแปร buttonPanel และ valuePanel

บรรทัดที่ 10 กำหนด Button เป็นแบบอาร์เรย์ โดยเก็บทั้งหมด 12 ปุ่ม : ชื่อตัวแปร button

บรรทัดที่ 11 กำหนด RadioButton เป็นแบบอาร์เรย์ โดยเก็บทั้งหมด 4 ปุ่ม : ชื่อตัวแปร radiobutton

บรรทัดที่ 12 กำหนด ButtonGroup ไว้เก็บวัตถุที่เป็นปุ่มของ RadioButton : ชื่อตัวแปร buttongroup

บรรทัดที่ 13 กำหนด ImageIcon ไว้เก็บรูปภาพ : ชื่อตัวแปร ufoIcon

Method main อยู่บรรทัดที่ 15 และสร้าง object ของ method หลักเข้ามา

Method public Project จะเป็น method หลักในการทำงาน

```

16
17 public Project() {
18     super("Project : 060243104 Object-oriented Programming");
19     container = getContentPane();
20     container.setLayout(new FlowLayout());
21
22     graphicsPanel = new GraphicsPanel();
23     ufoIcon = new ImageIcon("images/ufo.png");
24
25     var check = JOptionPane.showConfirmDialog(null,"Do you want to use the project?","project",1,EXIT_ON_CLOSE,ufoIcon);
26
27     if (check == 0) {
28         button_Panel();
29         container.add(graphicsPanel);
30         value_Panel();
31         this.setVisible(true);
32     } else if (check == 1) {
33         JOptionPane.showMessageDialog(null,"You don't want to use the project.","project",JOptionPane.NO_OPTION);
34         System.exit(0);
35     } else if (check == 2) System.exit(0);
36
37     this.setSize(1200,750);
38     this.setLocation(160,30);
39     this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
40 }

```

container จะกำหนด Layout ของการจัดเรียงวัตถุเป็นแบบ FlowLayout

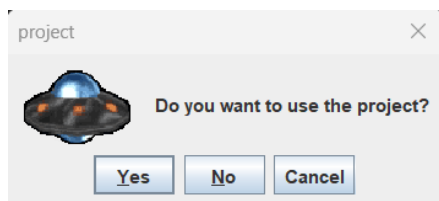
สร้าง object ของ graphicsPanel และ ufoIcon ขึ้นมา

กำหนดตัวแปรชนิด var ชื่อตัวแปร check และสร้างกราฟิกของ class JOptionPane ขึ้นมา เพื่อให้ผู้ใช้งานตัดสินใจก่อนเปิดโปรแกรม โดยจะสร้างเงื่อนไข if else ขึ้นมาเช็คการคลิกปุ่มของผู้ใช้งาน

บรรทัดที่ 27 – 31 เช็คค่า check ที่ปุ่ม Yes หรือไม่ ถ้าใช่จะทำการเปิดโปรแกรมขึ้นมา

บรรทัดที่ 32 – 34 เช็คค่า check ที่ปุ่ม No หรือไม่ ถ้าใช่จะทำการแสดงกราฟิก MessageDialog และปิดโปรแกรม

บรรทัดที่ 35 เช็คค่า check ที่ปุ่ม Cancel หรือไม่ ถ้าใช่จะทำการปิดโปรแกรม



โดยขนาดของ container จะกำหนดขนาดของตัวโปรแกรมไว้เก็บวัตถุอยู่ที่ ความกว้าง 1200 ความสูง 750

และเช็คตำแหน่งของโปรแกรมให้อยู่ในตำแหน่งหน้าจอที่กำหนด ที่ x 160 , y 30

Method private void button_Panel จะสร้างวัตถุที่เป็นกล่องและเก็บวัตถุของปุ่มต่างๆ

```

41
42     private void button_Panel() {
43         buttonPanel = new JPanel();
44         buttonPanel.setPreferredSize(new Dimension(1180,45));
45         buttonPanel.setBorder(BorderFactory.createLineBorder(Color.gray,1));
46         buttonPanel.setBackground(Color.lightGray);
47
48         String[] textbutton = {"Triangle","Square","Rectangle","Circle","Red","Green",
49                                "Blue","Magenta","Picture","Play","Stop","Exit"};
50
51         for (int n = 0 ; n < 12 ; n++) {
52             button[n] = new JButton(textbutton[n]);
53             button[n].setFont(new Font("SERIF",Font.BOLD,17));
54             button[n].addActionListener(this);
55
56             JLabel empty = new JLabel(" | ");
57             empty.setFont(new Font("SERIF",Font.BOLD,20));
58
59             if (n == 4) buttonPanel.add(empty);
60             else if (n == 8) buttonPanel.add(empty);
61             else if (n == 11) buttonPanel.add(empty);
62
63             buttonPanel.add(button[n]);
64         }
65
66         container.add(buttonPanel);
67     }
68

```

สร้าง object ของ buttonPanel และเซตขนาดความกว้าง 1180 ความสูง 45 สร้างสีขอบกล่องและใส่สีพื้นหลัง

Data type ของ textbutton เป็นชนิด String แบบอาร์เรย์ จะเก็บข้อความ 12 ข้อความ เอาไว้ใส่ข้อความให้กับปุ่ม

สร้าง object ของ button ด้วย forloop และใส่ textbutton ที่เก็บชุดข้อความเอาไว้ใส่ลงไป และ ActionListener

ให้ทุกปุ่ม และนำ button ใส่ลงไปในกลุ่ม buttonPanel และใส่ลงไป container ที่เป็นตัวเก็บวัตถุหลัก

Method private void value_Panel จะสร้างวัตถุที่เป็นกล่องและเก็บวัตถุของปุ่มต่างๆ

```

69     private void value_Panel() {
70         valuePanel = new JPanel();
71         valuePanel.setPreferredSize(new Dimension(1180,35));
72         valuePanel.setBorder(BorderFactory.createLineBorder(Color.gray,1));
73         valuePanel.setBackground(Color.lightGray);
74
75         String[] textradiobutton = {"Bright","Dark","Orange","Pink"};
76
77         for (int n = 0 ; n < 4 ; n++) {
78             radiobutton[n] = new JRadioButton(textradiobutton[n]);
79             radiobutton[n].setBackground(Color.lightGray);
80             radiobutton[n].addActionListener(this);
81
82             buttongroup = new ButtonGroup();
83             buttongroup.add(radiobutton[0]);
84             buttongroup.add(radiobutton[1]);
85             buttongroup.add(radiobutton[2]);
86             buttongroup.add(radiobutton[3]);
87
88             valuePanel.add(radiobutton[n]);
89         }
90
91         container.add(valuePanel);
92     }
93

```

สร้าง object ของ valuePanel และเซตขนาดความกว้าง 1180 ความสูง 35 สร้างสีขอบกล่องและใส่สีพื้นหลัง

Data type ของ textradiobutton เป็น String แบบอาร์เรย์ จะเก็บข้อความ 4 ข้อความ เอาไว้ใส่ข้อความให้กับปุ่ม

สร้าง object ของ radiobutton ด้วย forloop และใส่ textradiobutton ที่เก็บชุดข้อความเอาไว้ใส่ลงไป

และ ActionListener ให้ทุกปุ่ม สร้าง object ของ buttongroup และทำการเก็บวัตถุ ของ radiobutton เข้ามา

และนำ buttongroup ใส่ลงไปในกลุ่ม valuePanel และใส่ลงไป container ที่เป็นตัวเก็บวัตถุหลัก

Method private void check_Action จะเป็น method ใช้ในการจัดการคลิกปุ่มต่างๆ

```

94 private void check_Action(ActionEvent e) {
95     // Button Shape in button_Panel
96     if (e.getSource() == button[0]) graphicsPanel.setTextShape("Triangle");
97     if (e.getSource() == button[1]) graphicsPanel.setTextShape("Square");
98     if (e.getSource() == button[2]) graphicsPanel.setTextShape("Rectangle");
99     if (e.getSource() == button[3]) graphicsPanel.setTextShape("Circle");
100    if (e.getSource() == button[8]) graphicsPanel.setTextShape("Picture");
101
102    // Button Color in button_Panel
103    if (e.getSource() == button[4]) graphicsPanel.setTextColor("Red");
104    if (e.getSource() == button[5]) graphicsPanel.setTextColor("Green");
105    if (e.getSource() == button[6]) graphicsPanel.setTextColor("Blue");
106    if (e.getSource() == button[7]) graphicsPanel.setTextColor("Magenta");
107
108    // Button Play,Stop in button_Panel
109    if (e.getSource() == button[9]) graphicsPanel.setTextPS("Play");
110    if (e.getSource() == button[10]) graphicsPanel.setTextPS("Stop");
111
112    // Button Exit in button_Panel
113    if (e.getSource() == button[11]) {
114        JOptionPane.showMessageDialog(null,"Thank you for using","project",EXIT_ON_CLOSE,ufoIcon);
115        System.exit(0);
116    }
117
118    // RadioButton Mode in value_Panel
119    if (radiobutton[0].isSelected() == true) graphicsPanel.setTextMode("Bright");
120    if (radiobutton[1].isSelected() == true) graphicsPanel.setTextMode("Dark");
121    if (radiobutton[2].isSelected() == true) graphicsPanel.setTextMode("Orange");
122    if (radiobutton[3].isSelected() == true) graphicsPanel.setTextMode("Pink");
123    repaint();
124 }
125

```

หลังจากเช็คด้วยเงื่อนไข getSource ของปุ่มต่างๆแล้ว จะเก็บข้อมูลแบบ properties โดยจะประกาศคลาสของ GraphicsPanel และประกาศ method ที่เป็น setter ของ GraphicsPanel เข้ามา และส่งค่าเป็น String กลับไป เพื่อใช้ในการแสดงหน้าจอการวาดภาพออกมาหลังจากกดปุ่ม โดยจะมี method setter ทั้งหมด ดังนี้

setTextShape

setTextColor

setTextPS

setTextMode

(จะถูกอธิบายการทำงานอยู่ในส่วนของคลาส GraphicsPanel)

Method private void check_Paint จะเป็น method ใช้ในการวาดภาพและเปลี่ยนสีพื้นหลังของปุ่มต่างๆ

```

126 private void check_Paint() {
127     // Set a color for the default button.
128     for (int n = 0 ; n < 11 ; n++) button[n].setBackground(Color.orange);
129
130     // Button Shape in button_Panel
131     if (graphicsPanel.getTextShape() == "Triangle") button[0].setBackground(Color.green);
132     if (graphicsPanel.getTextShape() == "Square") button[1].setBackground(Color.green);
133     if (graphicsPanel.getTextShape() == "Rectangle") button[2].setBackground(Color.green);
134     if (graphicsPanel.getTextShape() == "Circle") button[3].setBackground(Color.green);
135     if (graphicsPanel.getTextShape() == "Picture") button[8].setBackground(Color.green);
136
137     // Button Color in button_Panel
138     if (graphicsPanel.getTextColor() == "Red") button[4].setBackground(Color.green);
139     if (graphicsPanel.getTextColor() == "Green") button[5].setBackground(Color.green);
140     if (graphicsPanel.getTextColor() == "Blue") button[6].setBackground(Color.green);
141     if (graphicsPanel.getTextColor() == "Magenta") button[7].setBackground(Color.green);
142
143     // Button Play,Stop in button_Panel
144     if (graphicsPanel.getTextPS() == "Play") button[9].setBackground(Color.yellow);
145     if (graphicsPanel.getTextPS() == "Stop") button[9].setBackground(Color.orange);
146
147     // Button Exit in button_Panel
148     button[11].setForeground(Color.white);
149     button[11].setBackground(Color.red);
150
151     // Message Shap,Color in value_Panel
152     Graphics g = valuePanel.getGraphics();
153     g.setFont(new Font("SERIF",Font.PLAIN,18));
154     g.drawString("Shap : " + graphicsPanel.getTextShape(), 10, 25);
155     g.drawString("Color : " + graphicsPanel.getTextColor(), 1050, 25);
156 }
157

```

กำหนดสีพื้นหลังของปุ่ม button ทั้งหมด เป็นสีส้ม โดยใช้ forloop และเช็คเงื่อนไขด้วยข้อมูลที่เป็น properties โดยจะประกาศคลาสของ GraphicsPanel และประกาศ method ที่เป็น getter ของ GraphicsPanel เข้ามาทำการเช็คข้อความ หลังจากเช็คด้วยเงื่อนไข จะทำการเปลี่ยนสีปุ่ม button เป็นสีเขียวหลังจากกดปุ่มนั้นๆ

ทำการวาดกราฟิกแบบข้อความไว้ในส่วนวัตถุของกล่อง valuePanel แสดงข้อความเมื่อเรากดปุ่มนั้นๆ

โดยจะมี method getter ทั้งหมด ดังนี้

getTextShape

getTextColor

getTextPS

(จะถูกอธิบายการทำงานอยู่ในส่วนของคลาส GraphicsPanel)

Method Override paint และ actionPerformed จะเป็น method หลักในการทำงาน graphics และ action

```

158      @Override // main paint checker
159      public void paint(Graphics graphics) {
160          super.paint(graphics);
161          check_Paint();
162      }
163
164      @Override // main actionPerformed checker
165      public void actionPerformed(ActionEvent action) {
166          check_Action(action);
167          repaint();
168      }
169  }

```

โดยจะเรียกใช้ method

check_Paint

check_Action

เข้ามาทำงานในส่วนของ method หลักของตัวเอง

Class GraphicsPanel

```

1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4
5  class GraphicsPanel extends JPanel implements ActionListener, MouseListener, MouseMotionListener {
6      private Triangle triangle;
7      private Rectangle rectangle;
8      private Square square;
9      private Circle circle;
10     private Picture picture;
11
12     private Timer animation;
13     private int X_pos, Y_pos;
14     private String textShape = "null";
15     private String textColor, textPS, textMode;
16

```

เป็นคลาสที่ใช้แสดงรูปภาพและจำลองการวาดเส้นในรูปแบบต่างๆ หรือเป็นวัตถุกล่องหน้าจอแสดงผล

implements ActionListener เข้ามาเพื่อเช็คการทำงานหลังจากการคลิกปุ่ม

implements MouseListener เข้ามาเพื่อเช็คการทำงานหลังจากมีการคลิกเมาส์ที่หน้าจอแสดงผล

implements MouseMotionListener เข้ามาเพื่อเช็คการทำงานหลังจากมีเมาส์เคลื่อนที่ ที่หน้าจอแสดงผล

Attribute จะถูกสร้างอยู่ในช่วงบรรทัดที่ 6 – 15

ตัวแปรและชนิดตัวแปร จะถูกสร้างเป็น **private** โดยจะมีทั้งหมด ดังนี้

บรรทัดที่ 6 – 10 กำหนดคลาสที่จะนำมาวาดที่หน้าจอแสดงผลทั้งหมด เพื่อเตรียมเรียกใช้งาน

โดยมี Triangle , Rectangle , Square , Circle และ Picture

บรรทัดที่ 12 กำหนด Timer เพื่อเอาไว้สร้างการ movement ของรูปภาพ : ชื่อตัวแปร animation

บรรทัดที่ 13 กำหนด Integer เก็บค่าตำแหน่ง x , y ของเมาส์ที่อยู่ในหน้าจอแสดงผล : ชื่อตัวแปร X_pos , Y_pos

บรรทัดที่ 14 กำหนด String เก็บค่าข้อมูลและส่งค่าข้อมูลแบบ properties : ชื่อตัวแปร textShape

บรรทัดที่ 15 กำหนด String ทำงานเหมือนบรรทัดที่ 14 : ชื่อตัวแปร textColor , textPS , textmode

Method public GraphicsPanel จะเป็น method หลักในการทำงาน

```

17     public GraphicsPanel() {
18         picture = new Picture();
19         rectangle = new Rectangle();
20         triangle = new Triangle();
21         square = new Square();
22         circle = new Circle();
23
24         animation = new Timer(30,this);
25         addMouseListener(this);
26         addMouseMotionListener(this);
27
28         this.setPreferredSize(new Dimension(1180,613));
29         this.setBorder(BorderFactory.createLineBorder(Color.gray,1));
30     }
31

```

สร้าง object ของทุกคลาสที่จะนำมาแสดงที่หน้าจอแสดงผลทั้งหมด และ animation โดยจะปรับความเร็ว อยู่ที่ 30 มิลลิวินาที และ addMouseListener กับ addMouseMotionListener เพื่อใช้ในการวาดเส้นทุกรูปแบบ กำหนดขนาดของหน้าจอแสดงผลอยู่ที่ ความกว้าง 1180 ความสูง 613 และใส่สีขอบของวัตถุกล่องแสดงผล

Method private void pos_Draw ทำการวาดกราฟิกข้อความของตำแหน่งเมาส์ที่อยู่ในจอแสดงผล

```

32     private void pos_Draw(Graphics g) {
33         String pos; // Draw Pos
34         pos = "[ x " + X_pos + " , y " + Y_pos + " ]";
35         g.setFont(new Font("SERIF",Font.PLAIN,15));
36         g.drawString(pos, 15, 25);
37     }
38

```

Method Override mouseMoved จะให้ตัวแปร X_pos และ Y_pos เก็บค่าตำแหน่งเมาส์ที่อยู่ในหน้าจอแสดงผล

```

39     @Override // main mouseMoved checker
40     public void mouseMoved(MouseEvent moved) {
41         X_pos = moved.getX();
42         Y_pos = moved.getY();
43         repaint();
44     }
45

```

Method private void step1_Draw เป็น method ใ้วาดรูปและเส้นของรูปแบบต่างๆ

```

46 private void step1_Draw(Graphics g) {
47     // Draw Shape
48     if (textShape == "null") {
49         g.setFont(new Font("MONOSPACED",Font.HANGING_BASELINE,50));
50         g.drawString("Please select a shape",260,300);
51     }
52     if (textShape == "Triangle") {
53         triangle.draw(g);
54         textPS = "Stop";
55     }
56     if (textShape == "Square") {
57         square.draw(g);
58         textPS = "Stop";
59     }
60     if (textShape == "Rectangle") {
61         rectangle.draw(g);
62         textPS = "Stop";
63     }
64     if (textShape == "Circle") {
65         circle.draw(g);
66         textPS = "Stop";
67     }
68     if (textShape == "Picture") picture.draw(g);
69
70     // animation
71     if (textPS == "Play") animation.start();
72     if (textPS == "Stop") animation.stop();
73 }
74

```

ต่อจากคลาส Project ที่มีการ setter ให้กับ method ของคลาส GraphicsPanel ข้อความที่ส่งมาจะถูกเก็บมาไว้กับตัวแปร ที่สร้างอยู่ส่วนของ Attribute ที่กำหนดไว้ และตัวแปรนั้นจะเก็บค่าที่ถูกส่งมาจาก setter เรียบร้อยแล้ว

โดยใน method นี้ จะมีการเช็คเงื่อนไขของตัวแปรที่เก็บข้อมูลแบบ properties ดังนี้

textShape ทำการวาดรูปและเส้นของรูปแบบต่างๆออกมาตามเงื่อนไขที่ถูกต้อง

textPS ทำการเริ่มและหยุดการ movement ของรูปภาพตามเงื่อนไขที่ถูกต้อง

Method private void step2_Paint เป็น method วาดสีของเส้นในรูปแบบต่างๆและวาดสีพื้นหลังหน้าจอแสดงผล

```

75     private void step2_Paint(Graphics g) {
76         // step2_Paint gets shape1_Paint and pos_Draw into work.
77         pos_Draw(g);
78         step1_Draw(g);
79
80         // Draw Color
81         if (textColor == "Red") {
82             g.setColor(Color.red);
83             step1_Draw(g);
84         }
85         if (textColor == "Green") {
86             g.setColor(Color.green);
87             step1_Draw(g);
88         }
89         if (textColor == "Blue") {
90             g.setColor(Color.blue);
91             step1_Draw(g);
92         }
93         if (textColor == "Magenta") {
94             g.setColor(Color.magenta);
95             step1_Draw(g);
96         }
97
98         // Draw screen color
99         if (textMode == "Bright") this.setBackground(Color.white);
100        if (textMode == "Dark") this.setBackground(Color.lightGray);
101        if (textMode == "Orange") this.setBackground(Color.orange);
102        if (textMode == "Pink") this.setBackground(Color.pink);
103    }
104

```

ต่อจากคลาส Project ที่มีการ setter ให้กับ method ของคลาส GraphicsPanel ข้อความที่ส่งมาจะถูกเก็บมาไว้กับตัวแปร ที่สร้างอยู่ส่วนของ Attribute ที่กำหนดไว้ และตัวแปรนั้นจะเก็บค่าที่ถูกส่งมาจาก setter เรียบร้อยแล้ว

โดยใน method นี้ จะมีการเช็คเงื่อนไขของตัวแปรที่เก็บข้อมูลแบบ properties ดังนี้

textColor ทำการวาดสีให้กับเส้นของรูปแบบต่างๆออกมาตามเงื่อนไขที่ถูกต้อง

textMode ทำการวาดสีให้กับพื้นหลังหน้าจอแสดงผลออกมาตามเงื่อนไขที่ถูกต้อง

Method Override paintComponent และ actionPerformed เป็น method หลักในการทำงาน graphics และ action

```

105      @Override // main paintComponent checker
106  ✓    public void paintComponent(Graphics graphics) {
107          super.paintComponent(graphics);
108          step2_Paint(graphics);
109      }
110
111      @Override // main actionPerformed checker
112  ✓    public void actionPerformed(ActionEvent action) {
113          picture.move(); // call method move from class picture
114          repaint();
115      }
116

```

โดยจะเรียกใช้ method

step2_paint

picture.move

เข้ามาทำงานในส่วนของ method หลักของตัวเอง

Method private void check_mousePressed เป็น method ไว้วาดเส้นในตำแหน่งที่เมาส์คลิกในหน้าจอแสดงผล

```

117      private void check_mousePressed(MouseEvent m) {
118          if (textShape == "Triangle") triangle.mousePressed(m);
119          if (textShape == "Rectangle") rectangle.mousePressed(m);
120          if (textShape == "Square") square.mousePressed(m);
121          if (textShape == "Circle") circle.mousePressed(m);
122          repaint();
123      }
124

```

โดยที่จะเรียก method ที่มีชื่อว่า mousePressed ที่ถูกสร้างอยู่ในคลาสอื่นๆมาใช้งาน

Method Override mousePressed เรียก method check_mousePressed เข้ามาทำงาน

```
125     @Override // main mousePressed checker
126     public void mousePressed(MouseEvent pressed) {check_mousePressed(pressed);}
127
```

Method private void check_mouseDragged เป็น method เพื่อเพิ่มและลดขนาดของเส้นเมื่อถูกลากเมาส์

```
128     private void check_mouseDragged(MouseEvent m) {
129         if (textShape == "Triangle") triangle.mouseDragged(m);
130         if (textShape == "Rectangle") rectangle.mouseDragged(m);
131         if (textShape == "Square") square.mouseDragged(m);
132         if (textShape == "Circle") circle.mouseDragged(m);
133         repaint();
134     }
135
```

โดยที่จะเรียก method ที่มีชื่อว่า mouseDragged ที่ถูกสร้างอยู่ในคลาสอื่นๆมาใช้งาน

Method Override mouseDragged เรียก method check_mouseDragged เข้ามาทำงาน

```
136     @Override // main mouseDragged checker
137     public void mouseDragged(MouseEvent dragged) {check_mouseDragged(dragged);}
138
```

Method setter and getter

```
139     // Set a string at the project class.
140     public void setTextShape(String textShape) {this.textShape = textShape;}
141     public String getTextShape() {return this.textShape;}
142
143     public void setTextColor(String textColor) {this.textColor = textColor;}
144     public String getTextColor() {return this.textColor;}
145
146     public void setTextPS(String textPS) {this.textPS = textPS;}
147     public String getTextPS() {return this.textPS;}
148
149     public void setTextMode(String textMode) {this.textMode = textMode;}
150
```

เป็นการเก็บค่าใหม่และส่งค่าของตัวแปรที่ประกาศในส่วนของ Attribute แบบ private ในคลาส GraphicsPanel

โดยจะทำงานแบบนี้ได้โดยการผ่าน accessor ที่มีชื่อว่า set และ get

set จะเป็นการเขียนหรือเปลี่ยนแปลงข้อมูลจากตัวแปร private ได้

get จะเป็นการเรียกดูหรือเรียกใช้ข้อมูลจากตัวแปร private ได้

properties ทั้งหมดของคลาส GraphicsPanel จะมีทั้งหมดตามภาพ source code

Class Triangle

Method public Triangle จะเป็น constructor และทำการsetค่าของ x1,y1,x2,y2,x3,y3

```

1  import java.awt.*;
2  import java.awt.event.*;
3
4  class Triangle extends Point implements MouseListener, MouseMotionListener {
5      private int x2=75 , y2=400;
6      private int x3=325 , y3=400;
7
8      public Triangle() {
9      }
10
11     public Triangle(int x, int y, int x2, int y2, int x3, int y3) {
12         super(x, y);
13         this.x2 = x2;
14         this.y2 = y2;
15         this.x3 = x3;
16         this.y3 = y3;
17     }

```

implements MouseListener, MouseMotionListener เข้ามาเพื่อเช็คการทำงานต่างๆของเมาส์

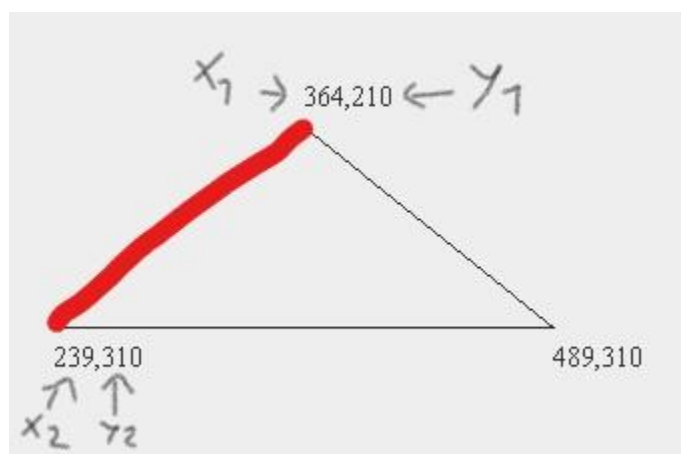
Method public void draw จะทำหน้าที่วาดรูปสามเหลี่ยม

```

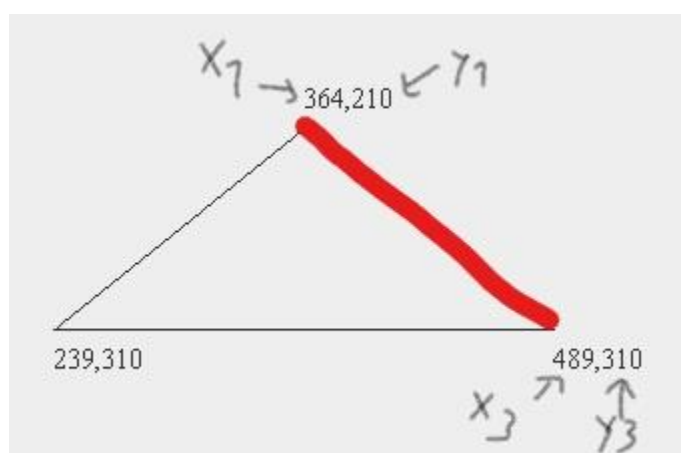
12
13     public void draw(Graphics graphics) {
14         String str,str2,str3;
15
16         // Draw Triangle
17         graphics.drawLine(getX(),getY(),getX2(),getY2());
18         str = getX() + "," + getY();
19         graphics.drawString(str, getX(), getY() - 10);
20
21         graphics.drawLine(getX(),getY(),getX3(),getY3());
22         str2 = getX2() + "," + getY2();
23         graphics.drawString(str2, getX2(), getY2() + 20);
24
25         graphics.drawLine(getX2(),getY2(),getX3(),getY3());
26         str3 = getX3() + "," + getY3();
27         graphics.drawString(str3, getX3(), getY3() + 20);
28     }
29

```

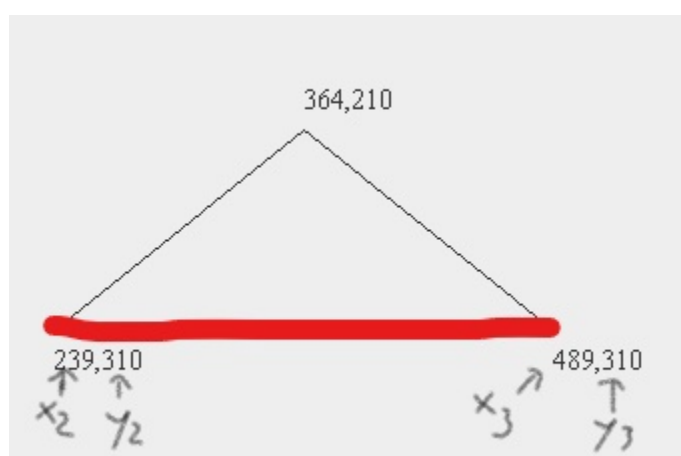
บรรทัดที่ 17-19 จะเป็นการวาดเส้นด้านซ้ายของสามเหลี่ยม



บรรทัดที่ 21-23 จะเป็นการวาดเส้นด้านขวาของสามเหลี่ยม



บรรทัดที่ 25-27 จะเป็นการวาดเส้นด้านล่างของสามเหลี่ยม



Method public void mousePressed จะเปลี่ยนตำแหน่งรูปสามเหลี่ยมหลังจากกดเมาส์

```

30     public void mousePressed(MouseEvent pressed) {
31         int x,y;
32         x = getX() - getX2();
33         y = getY2() - getY();
34
35         // Draw a new point triangle.
36         setX(pressed.getX());
37         setY(pressed.getY());
38
39         setX2(getX() - x);
40         setY2(getY() + y);
41
42         setX3(getX() + x);
43         setY3(getY2());
44     }

```

บรรทัดที่ 31-33 จะคำนวณตัวแปร x ซึ่งเป็นระยะห่างระหว่างจุด x1 กับ x2 และ จะคำนวณตัวแปร y ซึ่งเป็นระยะห่างระหว่างจุด y1 กับ y2 ซึ่งก็คือความยาวของเส้นด้านซ้าย

บรรทัดที่ 36-37 จะทำการsetค่า x1,y1 ที่เป็นตำแหน่งเมาส์ที่เรากดลงไป

บรรทัดที่ 39-40 จะทำการsetค่า x2,y2 โดยค่าของ x2 จะเท่ากับค่า x1 ลบกับ ค่า x ที่เราคำนวณไว้ก่อนหน้านี้ และค่าของ y2 จะเท่ากับค่า y1 บวกกับ ค่า y

บรรทัดที่ 42-43 จะทำการsetค่า x3,y3 โดยค่าของ x3 จะเท่ากับค่า x1 บวกกับ ค่า x ที่เราคำนวณไว้ก่อนหน้านี้ และค่าของ y3 จะเท่ากับค่า y2

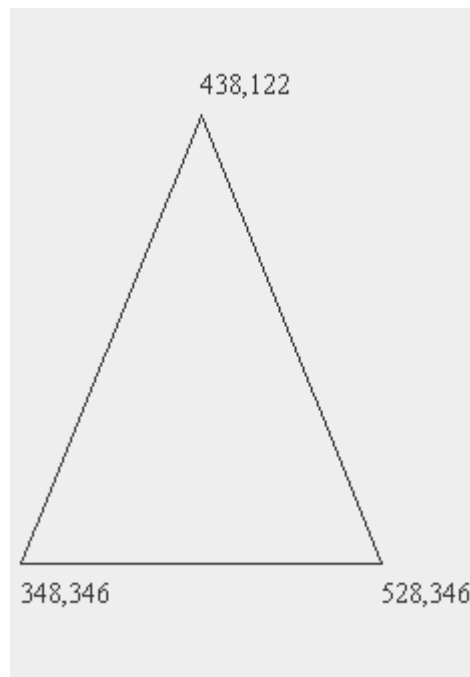
Method public void mouseDragged จะทำการวาดสามเหลี่ยมโดยการลากเมาส์

```

46     public void mouseDragged(MouseEvent dragged) {
47         // draw a triangle at the same point But when you drag the mouse It increases and decreases in size.
48         setX2(dragged.getX());
49         setY2(dragged.getY());
50
51         setX3((getX() - dragged.getX()) + getX());
52         setY3(dragged.getY());
53     }
54

```

บรรทัดที่ 48-52 เมื่อมีการลากเมาส์เกิดขึ้น จุดx2,y2 จะอยู่ตรงจุดที่ cursor อยู่ และทำการ set ค่าของ x3,y3 ให้
อยู่ตรงข้ามกัน



Class Square

Method public Square จะเป็น constructor และทำการsetค่าของ x1,y1,x2,y2,x3,y3,x4,y4

```

1  import java.awt.*;
2  import java.awt.event.*;
3
4  class Square extends Point implements MouseListener, MouseMotionListener {
5      private int oldX = 0;
6      private boolean start = true;
7      private int x2=350,y2=300;
8      private int x3=200,y3=450;
9      private int x4=350,y4=450;
10
11     public Square() {
12     }
13
14     public Square(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4) {
15         super(x, y);
16         this.x2 = x2;
17         this.y2 = y2;
18         this.x3 = x3;
19         this.y3 = y3;
20         this.x4 = x4;
21         this.y4 = y4;
22     }

```

implements MouseListener, MouseMotionListener เข้ามาเพื่อเช็คการทำงานต่างๆของเมาส์

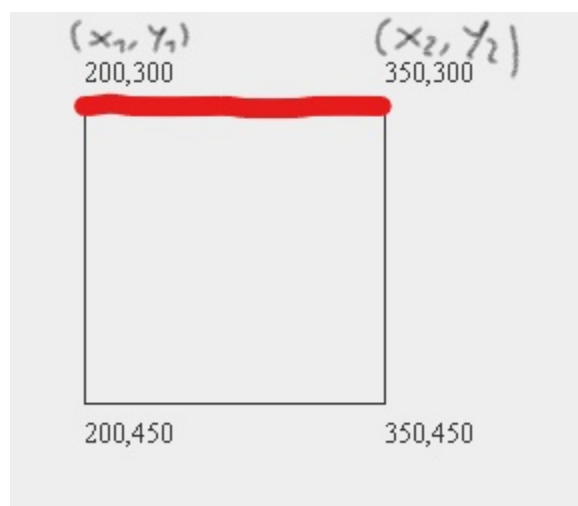
Method public void draw จะทำหน้าที่วาดรูปสี่เหลี่ยมจัตุรัส

```

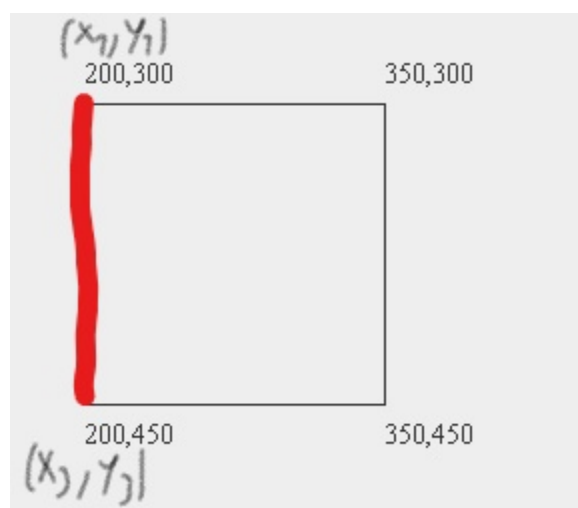
15     public void draw(Graphics graphics) {
16         String str,str2,str3,str4;
17
18         // Draw square
19         graphics.drawLine(getX(),getY(),getX2(),getY2());
20         str = getX() + "," + getY();
21         graphics.drawString(str, getX(), getY() - 10);
22
23         graphics.drawLine(getX(),getY(),getX3(),getY3());
24         str2 = getX2() + "," + getY2();
25         graphics.drawString(str2, getX2(), getY2() - 10);
26
27         graphics.drawLine(getX2(),getY2(),getX4(),getY4());
28         str3 = getX3() + "," + getY3();
29         graphics.drawString(str3, getX3(), getY3() + 20);
30
31         graphics.drawLine(getX4(),getY4(),getX3(),getY3());
32         str4 = getX4() + "," + getY4();
33         graphics.drawString(str4, getX4(), getY4() + 20);
34     }
35

```

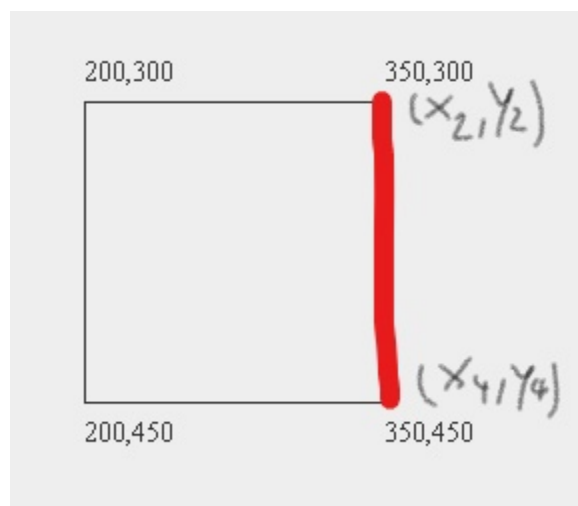

บรรทัดที่ 19-21 จะเป็นการวาดเส้นด้านบนของสี่เหลี่ยมจัตุรัส



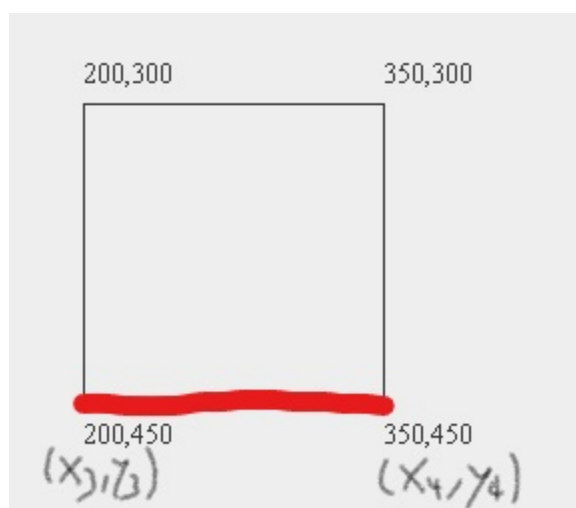
บรรทัดที่ 23-25 จะเป็นการวาดเส้นด้านซ้ายของสี่เหลี่ยมจัตุรัส



บรรทัดที่ 27-29 จะเป็นการวาดเส้นด้านขวาของสี่เหลี่ยมจัตุรัส



บรรทัดที่ 31-33 จะเป็นการวาดเส้นด้านล่างของสี่เหลี่ยมจัตุรัส



Method public void mousePressed จะเปลี่ยนตำแหน่งรูปสี่เหลี่ยมจัตุรัสหลังจากกดเมาส์

```

36     public void mousePressed(MouseEvent pressed) {
37         int n;
38         n = getX2() - getX();
39
40         // Draw a new point square.
41         setX(pressed.getX());
42         setY(pressed.getY());
43
44         setX2(getX() + n);
45         setY2(pressed.getY());
46
47         setX3(pressed.getX());
48         setY3(pressed.getY() + n);
49
50         setX4(getX2());
51         setY4(getY3());
52     }
53

```

บรรทัดที่ 38 จะนำจุด $x_2 - x_1$ เพื่อหาค่าความยาวของสี่เหลี่ยมแล้วเก็บลงตัวแปร n

บรรทัดที่ 41-42 จะทำการเปลี่ยนตำแหน่ง x_1, y_1 เป็นตำแหน่ง x, y ที่เมาส์ได้ทำการกด

บรรทัดที่ 44-45 กำหนดค่า x_2 ให้เท่ากับ $x_1 + n$ และกำหนด y_2 ให้เท่ากับ y_1

บรรทัดที่ 47-48 กำหนดค่า x_3 ให้เท่ากับ x_1 และกำหนด y_3 ให้เท่ากับ $y_1 + n$

บรรทัดที่ 50-51 กำหนดค่า x_4 ให้เท่ากับ x_2 และกำหนด y_4 ให้เท่ากับ y_3

Method public void mouseDragged จะทำการวาดสี่เหลี่ยมจัตุรัสโดยการลากเมาส์

```

55 public void mouseDragged(MouseEvent dragged) {
56     // draw a square at the same point But when you drag the mouse It increases and decreases in size.
57     if(start == true) {
58         setX2(getX());
59         setY2(getY());
60
61         setX3(getX());
62         setY3(getY());
63
64         setX4(getX());
65         setY4(getY());
66     }
67     if (oldX < dragged.getX()) {
68         setX2(getX2() + 3);
69         setY3(getY3() + 3);
70
71         setX4(getX4() + 3);
72         setY4(getY4() + 3);
73     }
74     else if (oldX > dragged.getX()) {
75         setX2(getX2() - 3);
76         setY3(getY3() - 3);
77
78         setX4(getX4() - 3);
79         setY4(getY4() - 3);
80     }
81     oldX = dragged.getX();
82     start = false;
83 }

```

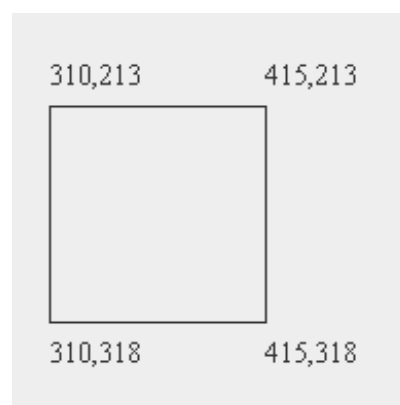
บรรทัดที่ 57-65 หาก start เป็น true จะทำการกำหนดค่า x2,y2,x3,y3,x4,y4 ให้เป็นจุดเดียวกับ x1,y1

บรรทัดที่ 82 กำหนด start ให้เป็น false เพื่อที่จะไม่ต้องให้คำสั่งในบรรทัด 57-65 ทำงานตลอดทุกครั้งขณะลากเมาส์

บรรทัดที่ 67-72 จะทำการเช็คเมาส์ถูกลากไปทางขวาหรือไม่หากใช่ สี่เหลี่ยมจัตุรัสหากใช่จะทำการขยายขนาดมาทางขวา

บรรทัดที่ 74-79 จะทำการเช็คเมาส์ถูกลากไปทางซ้ายหรือไม่หากใช่ สี่เหลี่ยมจัตุรัสหากใช่จะทำการขยายมาทางซ้ายแทน

บรรทัดที่ 81 ให้พิกัด x ที่เมาส์ลากไปเป็น oldX เพื่อไปเปรียบเทียบกับ



Class Rectangle

Method public Rectangle จะเป็น constructor และทำการsetค่าของ x1,y1,x2,y2,x3,y3,x4,y4

```

1  import java.awt.*;
2  import java.awt.event.*;
3
4  class Rectangle extends Point implements MouseListener, MouseMotionListener {
5      private int x2 = 600;
6      private int y2 = 300;
7      private int x3 = 200;
8      private int y3 = 500;
9      private int x4 = 600;
10     private int y4 = 500;
11
12     public Rectangle() {
13     }
14
15     public Rectangle(int x, int y, int x2, int y2, int x3, int y3, int x4, int y4) {
16         super(x, y);
17         this.x2 = x2;
18         this.y2 = y2;
19         this.x3 = x3;
20         this.y3 = y3;
21         this.x4 = x4;
22         this.y4 = y4;
23     }

```

implements MouseListener, MouseMotionListener เข้ามาเพื่อเช็คการทำงานต่างๆของเมาส์

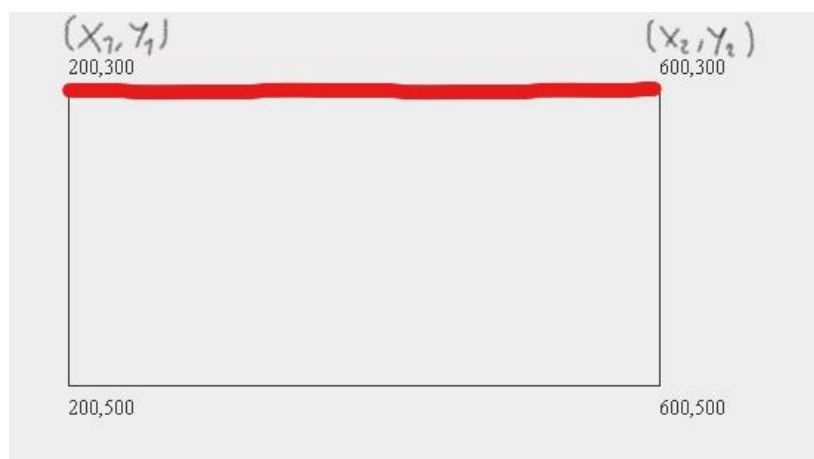
Method public void draw จะทำหน้าที่วาดรูปสี่เหลี่ยมผืนผ้า

```

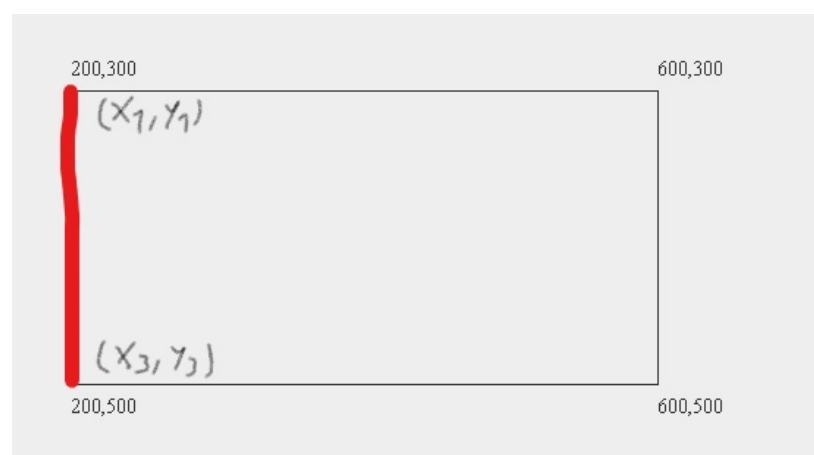
14     public void draw(Graphics graphics) {
15         String str,str2,str3,str4;
16
17         // Draw rectangle
18         graphics.drawLine(getX(), getY(), getX2(), getY2());
19         str = getX() + "," + getY();
20         graphics.drawString(str, getX(), getY() - 10);
21
22         graphics.drawLine(getX(), getY(), getX3(), getY3());
23         str2 = getX2() + "," + getY2();
24         graphics.drawString(str2, getX2(), getY2() - 10);
25
26         graphics.drawLine(getX2(), getY2(), getX4(), getY4());
27         str3 = getX3() + "," + getY3();
28         graphics.drawString(str3, getX3(), getY3() + 20);
29
30         graphics.drawLine(getX3(), getY3(), getX4(), getY4());
31         str4 = getX4() + "," + getY4();
32         graphics.drawString(str4, getX4(), getY4() + 20);
33     }
34

```

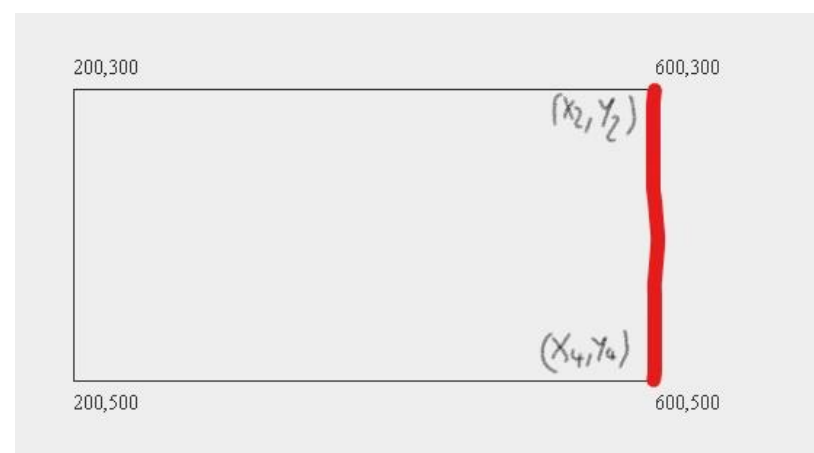
บรรทัดที่ 18-20 จะเป็นการวาดเส้นด้านบนของสี่เหลี่ยมผืนผ้า



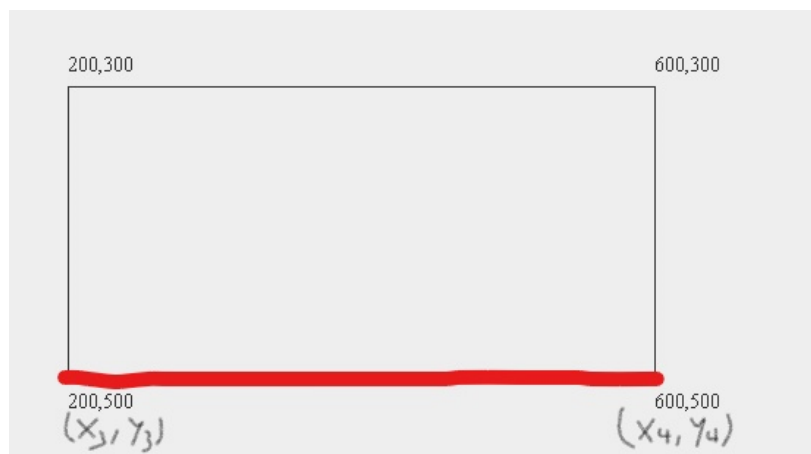
บรรทัดที่ 22-24 จะเป็นการวาดเส้นด้านซ้ายของสี่เหลี่ยมจัตุรัส



บรรทัดที่ 26-28 จะเป็นการวาดเส้นด้านขวาของสี่เหลี่ยมจัตุรัส



บรรทัดที่ 30-32 จะเป็นการวาดเส้นด้านล่างของสี่เหลี่ยมจัตุรัส



Method `public void mousePressed` จะเปลี่ยนตำแหน่งรูปสี่เหลี่ยมผืนผ้าหลังจากกดเมาส์

```

35     public void mousePressed(MouseEvent pressed) {
36         int w,h;
37         w = getX2() - getX();
38         h = getY3() - getY();
39
40         // Draw a new point rectangle.
41         setX(pressed.getX());
42         setY(pressed.getY());
43
44         setX2(pressed.getX() + w);
45         setY2(pressed.getY());
46
47         setX3(pressed.getX());
48         setY3(pressed.getY() + h);
49
50         setX4(pressed.getX2());
51         setY4(pressed.getY3());
52     }
53

```

บรรทัดที่ 37 จะทำการหาความกว้างโดยการนำ $x_2 - x_1$ แล้วเก็บในตัวแปร w

บรรทัดที่ 38 จะทำการหาความสูงโดยการนำ $y_3 - y_1$ แล้วเก็บในตัวแปร h

บรรทัดที่ 41-42 จะทำการเปลี่ยนตำแหน่ง x_1, y_1 เป็นตำแหน่ง x,y ที่เมาส์ได้ทำการกด

บรรทัดที่ 44-45 กำหนดค่า x_2 ให้เท่ากับ $x_1 + w$ และกำหนด y_2 ให้เท่ากับ y_1

บรรทัดที่ 47-48 กำหนดค่า x_3 ให้เท่ากับ x_1 และกำหนด y_3 ให้เท่ากับ $y_1 + h$

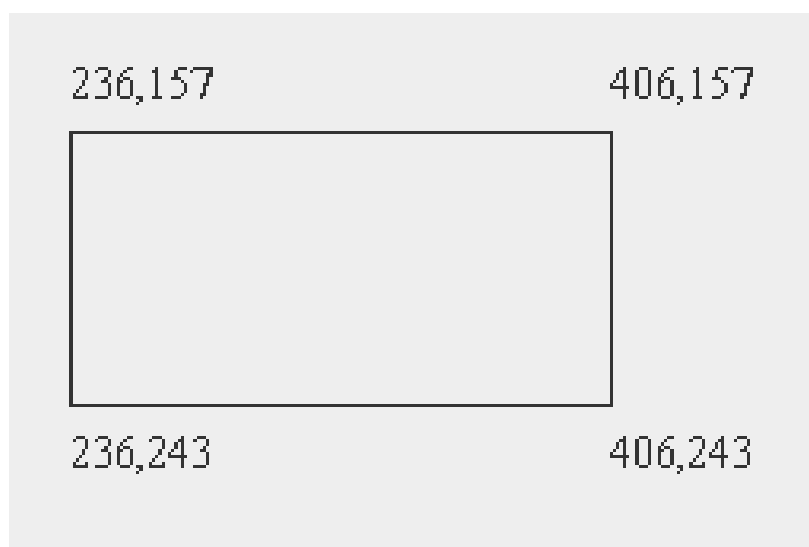
บรรทัดที่ 50-51 กำหนดค่า x_4 ให้เท่ากับ x_2 และกำหนด y_4 ให้เท่ากับ y_3

Method public void mouseDragged จะทำการวาดสี่เหลี่ยมผืนผ้าโดยการลากเมาส์

```
54 public void mouseDragged(MouseEvent dragged) {
55     // draw a rectangle at the same point But when you drag the mouse It increases and decreases in size
56     setX4(dragged.getX());
57     setY4(dragged.getY());
58
59     setX2(dragged.getX());
60     setY3(dragged.getY());
61 }
```

บรรทัดที่ 56-57 เมื่อมีการลากเมาส์เกิดขึ้น จุด x_4, y_4 จะอยู่ตรงจุดที่ cursor อยู่

บรรทัดที่ 59-60 กำหนดค่า x_2 ให้เท่ากับ x_1 และ y_3 ให้เท่ากับ y_1



Class Circle

Method public Circle จะเป็น constructor และทำการsetค่าของ x และ y

```

1  import java.awt.*;
2  import java.awt.event.*;
3
4  class Circle extends Point implements MouseListener, MouseMotionListener {
5      private boolean start = true;
6      private boolean dragging, pressing;
7
8      public Circle() {
9          // Set values for the x y axis.
10         setX(x:200); setY(y:300);
11     }
12

```

implements MouseListener, MouseMotionListener เข้ามาเพื่อเช็คการทำงานต่างๆของเมาส์

บรรทัดที่ 5 ตัวแปรที่ใช้กำหนดค่า start เป็น boolean

บรรทัดที่ 6 ตัวแปรกำหนดค่า dragging กับ pressing เป็น Boolean

Method public void draw จะทำหน้าที่วาดวงกลม

```

13  public void draw(Graphics graphics) {
14      // Draw circle
15      graphics.drawOval(getX(),getY(),2,2);
16      if (start == true) graphics.drawOval(getX()-100, getY()-100,200,200);
17
18      String str;
19      str = getX() + "," + getY();
20      graphics.drawString(str, getX(), getY() - 10);
21
22      // draw a circle at the same point But when you drag the mouse It increases and decreases in size.
23      if (dragging) {
24          int radius = (int) (getEndX() - getX());
25          setRadius(radius);
26          graphics.drawOval(getX() - getRadius(), getY() - getRadius(), getRadius() * 2, getRadius() * 2);
27      }
28
29      // Draw a new point circle.
30      if (pressing) {
31          graphics.drawOval(getX() - getRadius(), getY() - getRadius(), getRadius() * 2, getRadius() * 2);
32      }
33  }

```

บรรทัดที่ 15 วาดวงกลมเล็กๆที่เป็นจุดตรงกลาง

บรรทัดที่ 16 หาก start = true จะทำการวาดวงกลมขึ้นมาซึ่งวงกลมนี้เป็นวงกลมตอนเริ่มต้น

บรรทัดที่ 23-26 หากมีการ dragging จะทำการหา radius โดยนำจุดx มาลบกับ จุดEndX ซึ่งก็คือบริเวณที่cursor เมาส์อยู่ จากนั้นกำหนดค่า radius ลงไปแล้วทำการวาดวงกลมขึ้นมาโดย ตรงค่า x,y ตอนวาดจะมีการลบกับค่า radius ด้วยเพราะคำสั่ง drawoval จะมีจุด x,y เริ่มต้นตอนวาดอยู่ตรงมุมซ้ายบนของวงกลม

บรรทัดที่ 30-31 หากมีการ pressing จะทำการวาดวงกลมขึ้นใหม่โดยมีขนาด radius เท่าเดิมแต่เปลี่ยนเพียงแค่ ตำแหน่ง x,y

Method public void mousePressed จะเปลี่ยนตำแหน่งวงกลมหลังจากกดเมาส์

```

35     public void mousePressed(MouseEvent pressed) {
36         dragging = false;
37         pressing = true;
38         setX(pressed.getX());
39         setY(pressed.getY());
40     }
41

```

บรรทัดที่ 36 เปลี่ยน dragging เป็น false

บรรทัดที่ 37 เปลี่ยน pressing เป็น true เพื่อสั่ง drawoval ให้ทำงาน

บรรทัดที่ 38-39 กำหนดค่า x,y ที่ใช้ในการวาดวงกลมโดย x,y จะมาจากตำแหน่งที่เรากดเมาส์

Method public mouseDragged จะทำการวาดวงกลมโดยการลากเมาส์

```

42     public void mouseDragged(MouseEvent dragged) {
43         dragging = true;
44         start = false;
45         pressing = false;
46         setEndX(dragged.getX());
47         setEndY(dragged.getY());
48     }
49

```

บรรทัดที่ 43 เปลี่ยน dragging เป็น true เพื่อสั่ง drawoval ให้ทำงาน

บรรทัดที่ 44-45 เปลี่ยน pressing,start เป็น false

บรรทัดที่ 46-47 กำหนดค่า EndX,EndY ที่ใช้ในคิค่า radius และนำไปวาดวงกลมโดย EndX,EndY จะมาจากตำแหน่งที่เราลากเมาส์ไป



Class Picture

```

1  import java.awt.*;
2  import javax.swing.*;
3  import java.util.Random;
4
5  class Picture extends Point {
6      private Random random = new Random();
7      private ImageIcon[] ufo = new ImageIcon[10];
8      private int[] x = new int[10];
9      private int[] y = new int[10];
10     private int[] xSpeed = new int[10];
11     private int[] ySpeed = new int[10];
12     private int sizeX = 72, sizeY = 50;
13     private int width = 1180, height = 600;

```

Attribute จะถูกสร้างอยู่ในช่วงบรรทัดที่ 6 – 11

ตัวแปรและชนิดตัวแปร จะถูกสร้างเป็น private โดยจะมีทั้งหมด ดังนี้

บรรทัดที่ 6 กำหนด Random และสร้าง object เพื่อเตรียมเรียกใช้งาน : ชื่อตัวแปร random

บรรทัดที่ 7 กำหนด ImageIcon เป็นแบบอาร์เรย์ จะเก็บรูปทั้งหมด 10 รูป : ชื่อตัวแปร ufo

บรรทัดที่ 8 กำหนด Integer เป็นแบบอาร์เรย์ เก็บค่าแกน x ของรูปภาพ 10 ค่า : ชื่อตัวแปร x

บรรทัดที่ 9 กำหนด Integer เป็นแบบอาร์เรย์ เก็บค่าแกน y ของรูปภาพ 10 ค่า : ชื่อตัวแปร y

บรรทัดที่ 10 กำหนด Integer เป็นแบบอาร์เรย์ เก็บค่าความเร็วแกน x ของรูปภาพ 10 ค่า : ชื่อตัวแปร xSpeed

บรรทัดที่ 11 กำหนด Integer เป็นแบบอาร์เรย์ เก็บค่าความเร็วแกน y ของรูปภาพ 10 ค่า : ชื่อตัวแปร ySpeed

บรรทัดที่ 12 กำหนด Integer เก็บค่าขนาดความกว้างและความสูงของรูปภาพ : ชื่อตัวแปร sizeX, sizeY

บรรทัดที่ 13 กำหนด Integer เก็บค่าขนาดความกว้างและความสูงของกล่อง graphics : ชื่อตัวแปร width, height

Method public Picture จะเป็น method หลักในการทำงาน

```

13     public Picture() {
14         // random x , y , xSpeed , ySpeed
15         for (int n = 0 ; n < 10 ; n++) {
16             x[n] = random.nextInt(1000);
17             y[n] = random.nextInt(500);
18             xSpeed[n] = random.nextInt(6) + 1;
19             ySpeed[n] = random.nextInt(6) + 1;
20         }
21     }
22

```

จะสุ่มค่าแกน x , y , xSpeed และ ySpeed โดยใช้ forloop ในการเก็บค่าลงไปในแต่ละ index

สุ่ม x จาก 20 – 1000 โดยจะให้การสุ่มไม่เกินความกว้างของหน้าจอแสดงผล

สุ่ม y จาก 20 – 500 โดยจะให้การสุ่มไม่เกินความสูงของหน้าจอแสดงผล

สุ่ม xSpeed จาก 1 – 6 ความเร็วของรูปภาพ

สุ่ม ySpeed จาก 1 – 6 ความเร็วของรูปภาพ

Method public void draw เป็น method วาดรูปภาพ ufo

```

23     public void draw(Graphics graphics) {
24         // Draw ufo 10 ships
25         for (int n = 0 ; n < 10 ; n++) {
26             ufo[n] = new ImageIcon("images/ufo.png");
27             ufo[n].paintIcon(null, graphics, x[n], y[n]);
28         }
29     }
30

```

สร้าง object ของ ufo ด้วย forloop และทำการวาด ufo โดยเอาตัวแปร x และ y ที่ทำการสุ่มค่ามาใส่ (วาด 10 รูป)

Method public void move เป็น method ที่ใช้ในการ movement ของรูปภาพ ufo

```

33     public void move() {
34         // set the ufo to move
35         for (int n = 0 ; n < 5 ; n++) { // Run right 5 ships
36             x[n] += xSpeed[n];
37             y[n] += ySpeed[n];
38             if (x[n] < 1) xSpeed[n] = -xSpeed[n];
39             else if (x[n] + sizeX > width) xSpeed[n] = -xSpeed[n];
40             if (y[n] < 1) ySpeed[n] = -ySpeed[n];
41             else if (y[n] + sizeY > height) ySpeed[n] = -ySpeed[n];
42         }
43         for (int n = 5 ; n < 10 ; n++) { // Run left 5 ships
44             x[n] -= xSpeed[n];
45             y[n] -= ySpeed[n];
46             if (x[n] < 1) xSpeed[n] = -xSpeed[n];
47             else if (x[n] + sizeX > width) xSpeed[n] = -xSpeed[n];
48             if (y[n] < 1) ySpeed[n] = -ySpeed[n];
49             else if (y[n] + sizeY > height) ySpeed[n] = -ySpeed[n];
50         }
51     }
52 }

```

โดยจะกำหนดให้ ufo 5 รูป เริ่มต้นวิ่งไปทางขวา และ ufo อีก 5 รูป เริ่มต้นวิ่งไปทางซ้าย

ถ้า x บวก xSpeed จะทำการวิ่งไปทางขวา, ถ้า y บวก ySpeed จะทำการวิ่งไปทางขวา

ถ้า x ลบ xSpeed จะทำการวิ่งไปทางซ้าย, ถ้า y ลบ ySpeed จะทำการวิ่งไปทางซ้าย

เช็คด้วยเงื่อนไข if else

โดยที่เงื่อนไขแรก ถ้า x น้อยกว่า 1 หรือค่าความกว้างของหน้าจอแสดงผล รูปจะทำการเคลื่อนออกมาจากด้านซ้าย

เงื่อนไขที่สอง ถ้า x บวก ค่าความกว้างของรูป มากกว่า ค่าความกว้างของหน้าจอแสดงผล

รูปจะทำการเคลื่อนออกมาจากด้านขวา

เงื่อนไขที่สาม ถ้า y น้อยกว่า 1 หรือค่าความสูงของหน้าจอแสดงผล รูปจะทำการเคลื่อนออกมาจากด้านบน

เงื่อนไขที่สี่ ถ้า y บวก ค่าความสูงของรูป มากกว่า ค่าความสูงของหน้าจอแสดงผล

รูปจะทำการเคลื่อนออกมาจากด้านล่าง

Class Point

```
1  class Point {  
2      private int x = 200;  
3      private int y = 300;  
4  }
```

คลาส Point จะเป็นคลาสเก็บข้อมูลแบบ properties โดยเฉพาะ Attribute จะถูกสร้างอยู่ในช่วงบรรทัดที่ 2 – 6

ตัวแปรและชนิดตัวแปร จะถูกสร้างเป็น private โดยจะมีทั้งหมด ดังนี้

บรรทัดที่ 2 กำหนด Integer ใช้เก็บตำแหน่งของการวาดเส้นในรูปแบบต่างๆ : ชื่อตัวแปร x

บรรทัดที่ 3 กำหนด Integer ใช้เก็บตำแหน่งของการวาดเส้นในรูปแบบต่างๆ : ชื่อตัวแปร y

Method setter and getter

```

5      public Point() {}
6
7      public Point(int x, int y) {
8          this.x = x;
9          this.y = y;
10     }
11
12     public void setX(int x) {this.x = x;}
13     public int getX() {return this.x;}
14
15     public void setY(int y) {this.y = y;}
16     public int getY() {return this.y;}
17 }

```

บรรทัดที่ 7 เป็น constructor เก็บข้อมูลของ x และ y

ในบรรทัดที่ 11-16

เป็นการเก็บค่าใหม่และส่งค่าของตัวแปรที่ประกาศในส่วนของ Attribute แบบ private ในคลาส Point

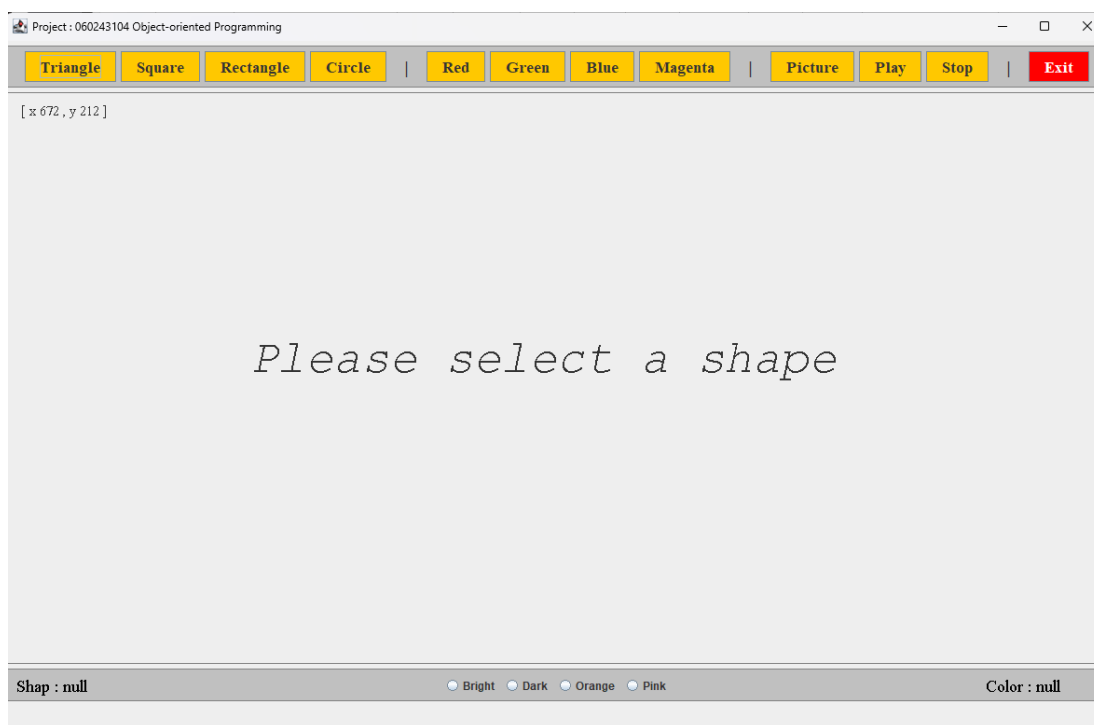
โดยจะทำงานแบบนี้ได้โดยการผ่าน accessor ที่มีชื่อว่า set และ get

set จะเป็นการเขียนหรือเปลี่ยนแปลงข้อมูลจากตัวแปร private ได้

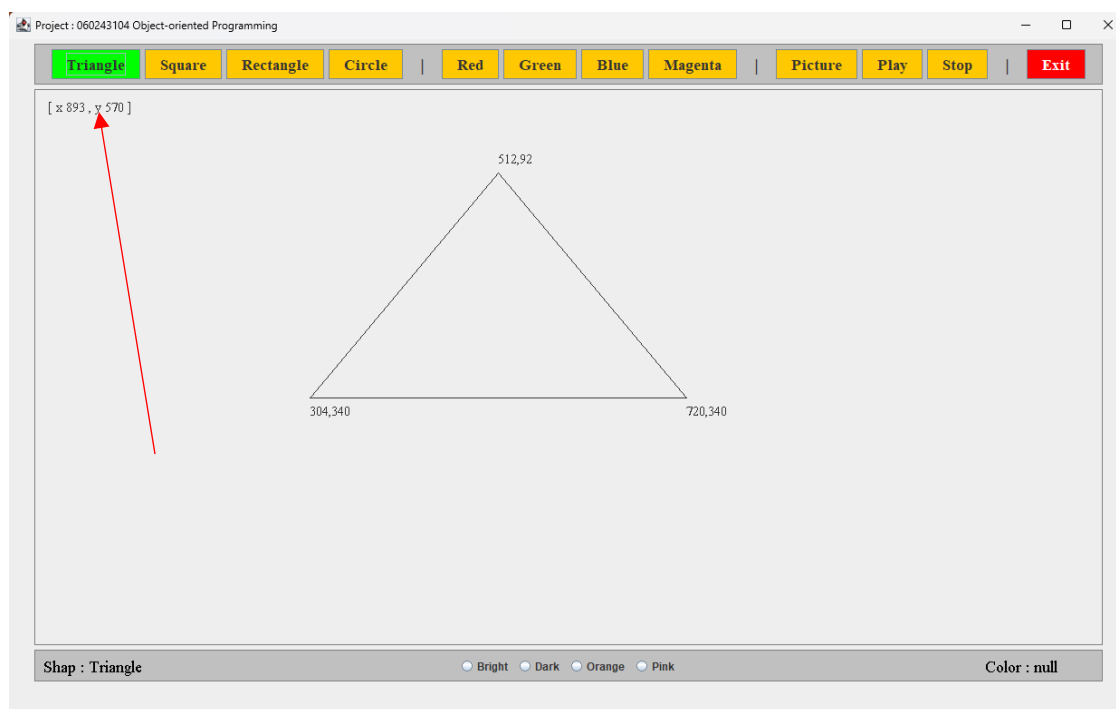
get จะเป็นการเรียกดูหรือเรียกใช้ข้อมูลจากตัวแปร private ได้

properties ทั้งหมดของคลาส Point จะมีทั้งหมดตามภาพ source code

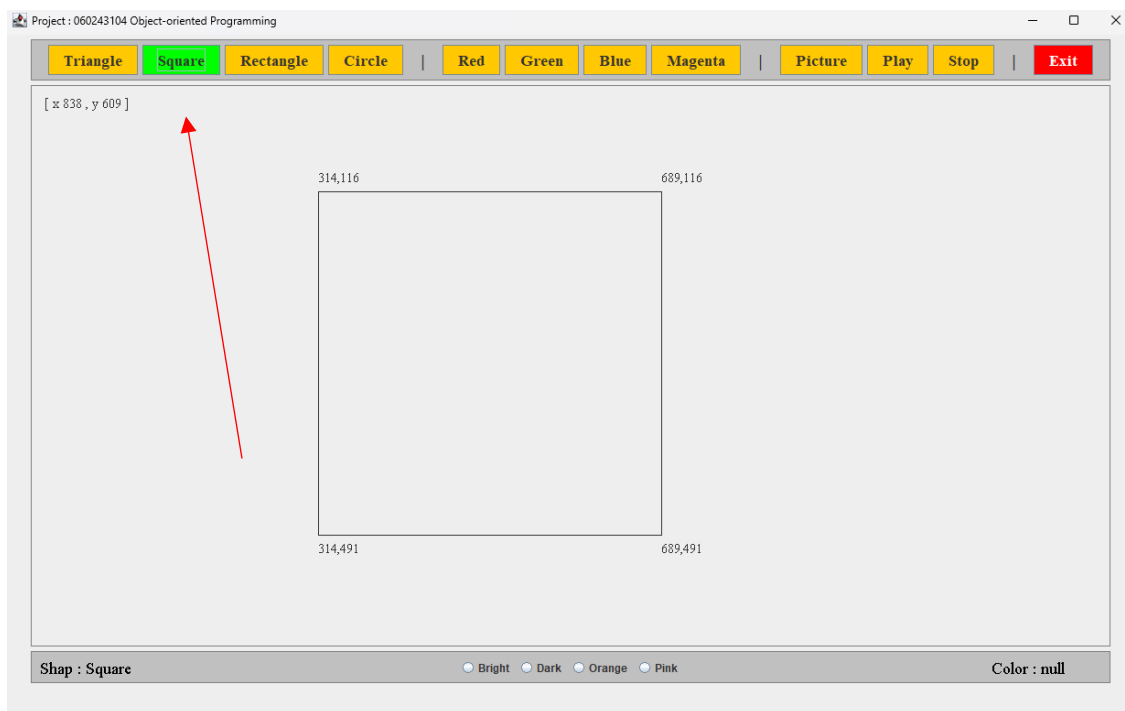
หน้าของโปรแกรม



หน้าของโปรแกรม เมื่อกดปุ่ม Triangle



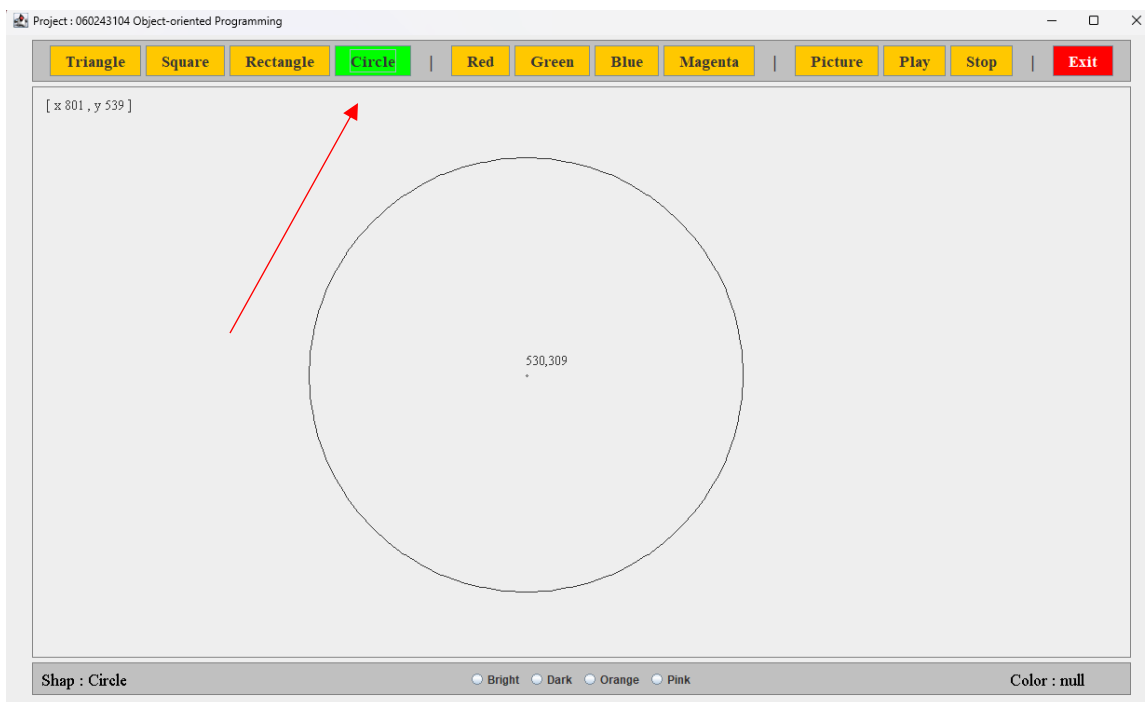
หน้าจอโปรแกรม เมื่อกดปุ่ม Square



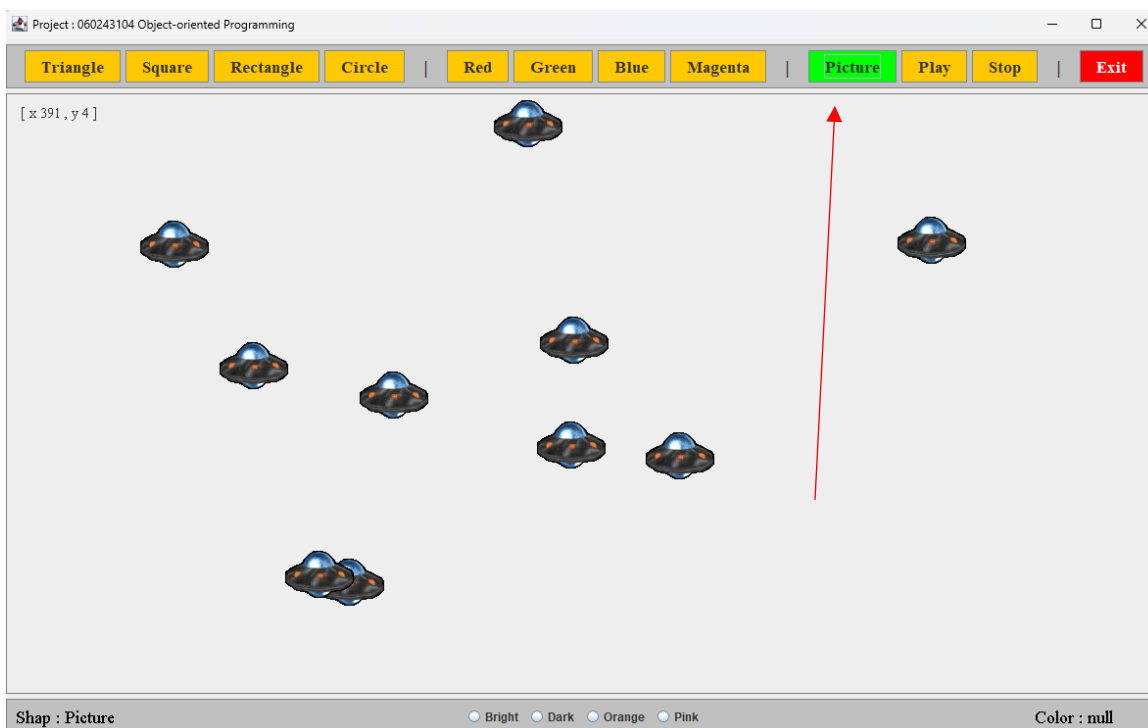
หน้าจอโปรแกรม เมื่อกดปุ่ม Rectangle



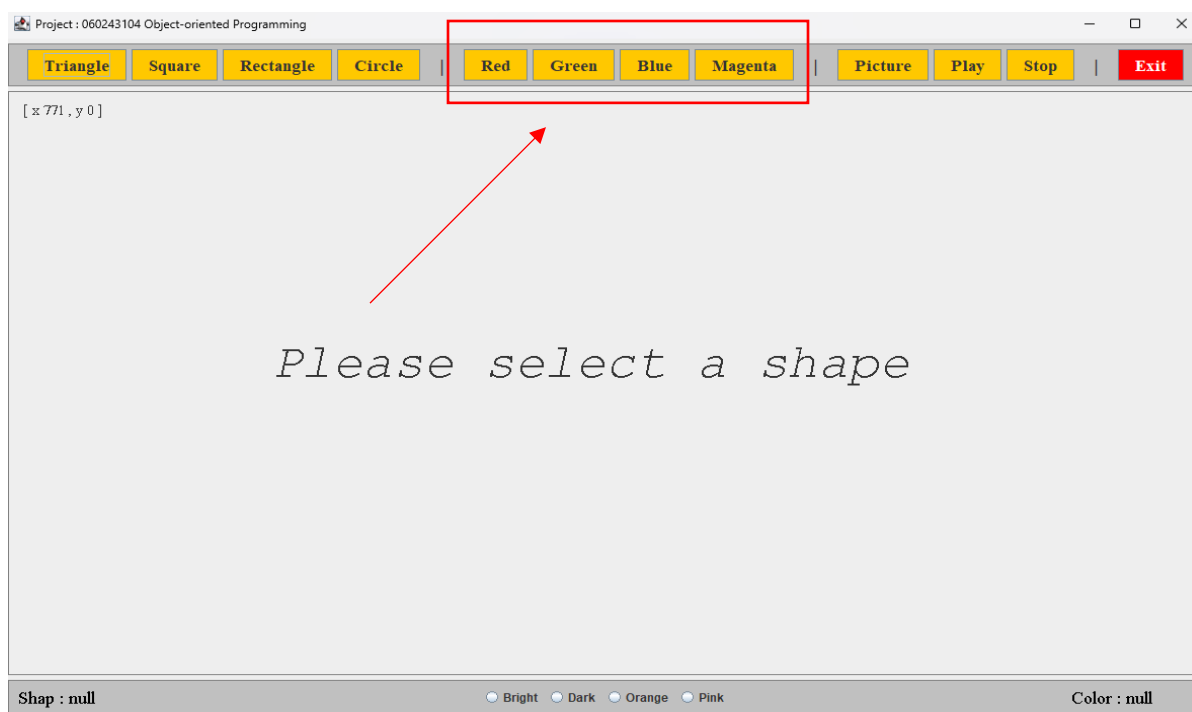
หน้าของโปรแกรม เมื่อกดปุ่ม Circle



หน้าของโปรแกรม เมื่อกดปุ่ม Picture



ปุ่มเปลี่ยนสีเส้น ของ Triangle, Square, Rectangle, Circle



ปุ่มเปลี่ยนสี background

