

แบบฝึกปฏิบัติ ครั้งที่ 8

เรื่อง

การจัดการเหตุการณ์และการเขียนโปรแกรมโดยอาศัยสถาปัตยกรรม **MVC**

วัตถุประสงค์

1. เพื่อฝึกฝนการจัดการเหตุการณ์ร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้
2. เพื่อทบทวนการเขียนโปรแกรมโดยอาศัยสถาปัตยกรรม **MVC**

1. ให้นักศึกษาตอบคำถามจากโปรแกรมต่อไปนี้

```

1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4
5  public class App01 implements ActionListener {
6
7      private JFrame fr;
8      private JPanel p1,p2;
9      private JTextField txt1, txt2, txt3;
10     private JButton btn1, btn2, btn3, btn4;
11
12     public App01() {
13
14         fr = new JFrame("เครื่องคิดเลข");
15         p1 = new JPanel();
16         p2 = new JPanel();
17         txt1 = new JTextField();
18         txt2 = new JTextField();
19         txt3 = new JTextField();
20         btn1 = new JButton("บวก");
21         btn2 = new JButton("ลบ");
22         btn3 = new JButton("คูณ");
23         btn4 = new JButton("หาร");
24
25         // Add Listener
26         btn1.addActionListener(this);
27         btn2.addActionListener(this);
28         btn3.addActionListener(this);
29         btn4.addActionListener(this);
30
31         // Set Layout
32         p1.setLayout(new GridLayout(4,1));
33         p1.add(txt1);
34         p1.add(txt2);
35         p1.add(p2);
36         p1.add(txt3);
37
38         p2.setLayout(new FlowLayout());
39         p2.add(btn1);
40         p2.add(btn2);
41         p2.add(btn3);
42         p2.add(btn4);
43         fr.getContentPane().add(p1);
44     }

```

```

45
46     // Set JFrame Property
47     fr.setFont(new Font("TimesRoman",Font.BOLD,60));
48     fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
49     fr.pack();
50     fr.setVisible(true);
51
52 }
53
54 public static void main(String[] args) {
55     new App01();
56 }
57
58 @Override
59 public void actionPerformed(ActionEvent ae) {
60     if(ae.getSource().equals(btn1)){
61         System.out.println("btn1");
62     }else if(ae.getSource().equals(btn2)){
63         System.out.println("btn2");
64     }else if(ae.getSource().equals(btn3)){
65         System.out.println("btn3");
66     }else if(ae.getSource().equals(btn4)){
67         System.out.println("btn4");
68     }
69 }
70 }
```

- 1.1. จากโปรแกรมบรรทัดที่ 1 – 3 ให้นักศึกษาอธิบายว่า **Package** ต่อไปนี้จะถูกเรียกใช้เมื่อใด

Package	อธิบาย
java.awt.*;	
java.awt.event.*;	
javax.swing.*;	

- 1.2. จากโปรแกรมบรรทัดที่ 5 ให้นักศึกษาอธิบายหน้าที่ของคำสั่ง `implements ActionListener`

- 1.3. จากโปรแกรมบรรทัดที่ 7 – 10 นักศึกษาคิดว่ามีจำนวน component และ container กี่อัน

- 1.4. จากโปรแกรมบรรทัดที่ 26 – 29 ให้นักศึกษาอธิบายว่าโปรแกรมดังกล่าวมีหน้าที่อะไร

- 1.5. จากโปรแกรมบรรทัดที่ 32 ให้นักศึกษาอธิบายหน้าที่ของคำสั่ง `p1.setLayout (new GridLayout (4,1))` และ Panel ของ p1 จะถูกแบ่งออกเป็นรูปแบบใดพร้อมวาดรูป

- 1.6. จากโปรแกรมบรรทัดที่ 38 ให้นักศึกษาอธิบายหน้าที่ของคำสั่ง `p2.setLayout (new FlowLayout ()) ;`

- 1.7. จากโปรแกรมบรรทัดที่ 48 – 50 ให้นักศึกษาอธิบายหน้าที่ของคำสั่งต่อไปนี้

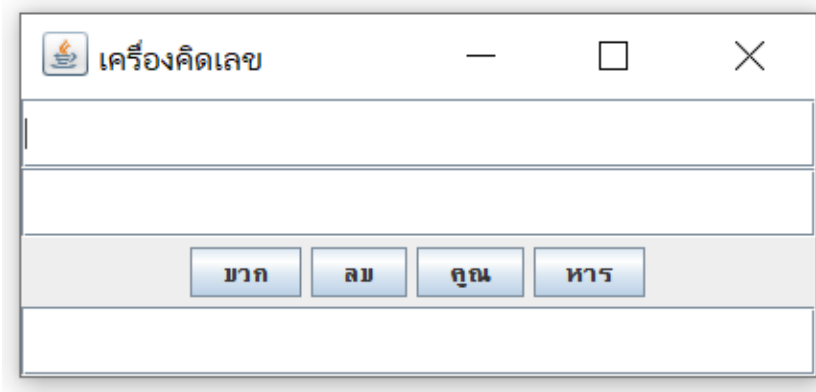
คำสั่ง	อธิบาย
<code>fr.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE)</code>	
<code>fr.pack ()</code>	
<code>fr.setVisible (true)</code>	

- 1.8. จากโปรแกรมบรรทัดที่ 59 ให้นักศึกษาอธิบายหน้าที่ของเมธอด
`public void actionPerformed (ActionEvent ae)`

นอกจากนี้ จากโปรแกรมบรรทัดที่ 60 ให้นักศึกษาอธิบายหน้าที่ของเมธอด `ae.getSource ()`

- 1.9. ให้นักศึกษาวาดรูปหน้าต่างส่วนติดต่อผู้ใช้ที่เป็นผลลัพธ์ของโปรแกรมจากโปรแกรมข้างต้น พร้อมอธิบายหลักการ
ทำงาน

2. ให้นิสิตแก้ไขโปรแกรมจากข้อที่ 1 ให้กลายเป็นโปรแกรมเครื่องคิดเลขอย่างง่าย

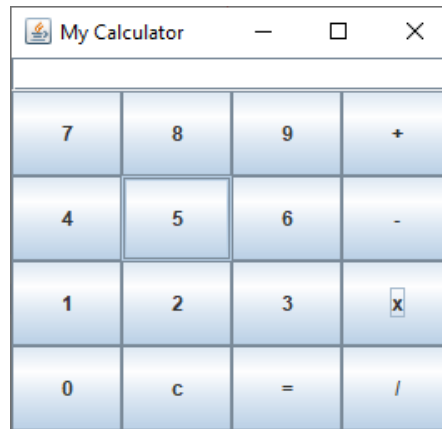


```

57
58     @Override
59     public void actionPerformed(ActionEvent ae) {
60         if(ae.getSource().equals(btn1)) {
61
62
63
64
65
66
67         }else if(ae.getSource().equals(btn2)) {
68
69
70
71
72
73
74         }else if(ae.getSource().equals(btn3)) {
75
76
77
78
79
80
81         }else if(ae.getSource().equals(btn4)) {
82
83
84
85
86
87         }
88     }
89
  
```

3.ให้นักศึกษาร่างโปรแกรมเครื่องคิดเลข โดยอาศัยส่วนต่อประสานกราฟิกกับผู้ใช้งานของคลาส "Calculator

Sample" จาก Lab Sheet 07 มาใช้งาน



โดยมีรายละเอียดดังต่อไปนี้

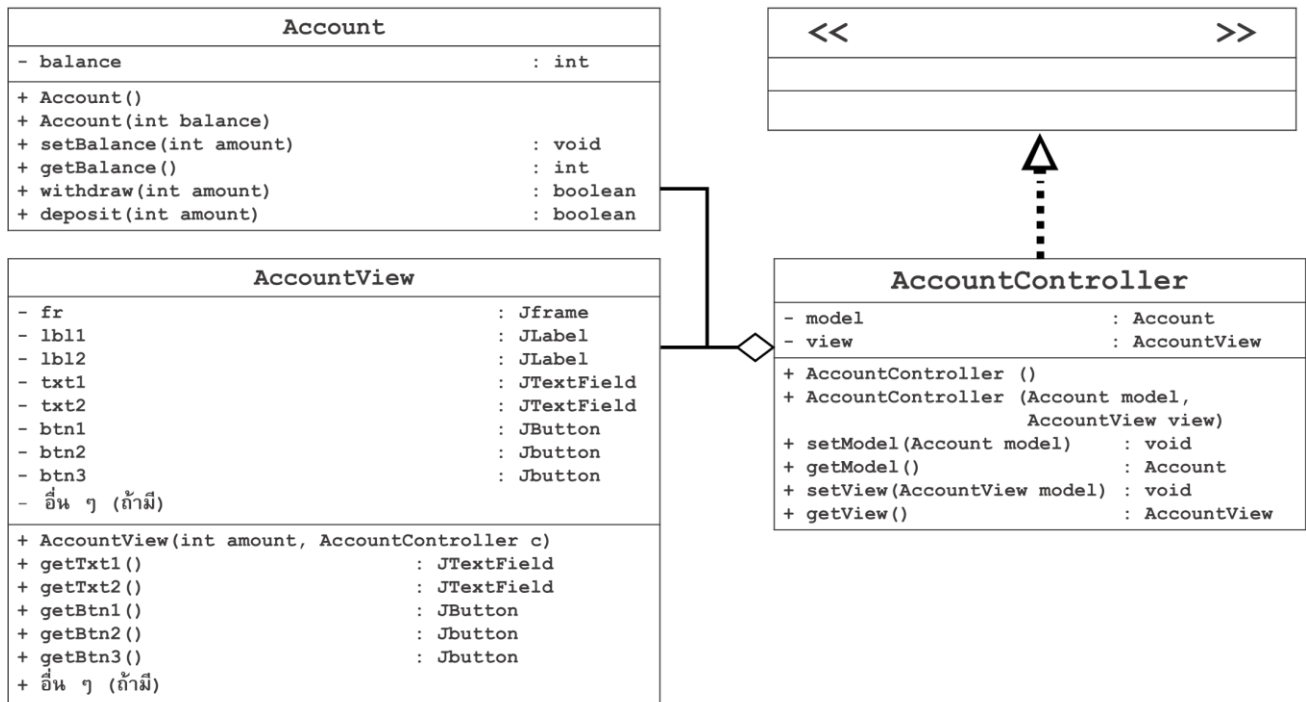
- 1) ถ้าผู้ใช้กดปุ่ม "+" หรือ "-" หรือ "x" หรือ "/" ค่าที่ปรากฏใน **JTextField** จะถูกกำหนดให้เป็นช่องว่าง
- 2) ถ้าผู้ใช้กดปุ่ม "0" ถึง "9" ค่าที่ปรากฏใน **JTextField** จะถูกเพิ่มต่อจากของเดิมทางด้านขวาไปเรื่อย ๆ
- 3) ถ้าผู้ใช้กดปุ่ม "c" ค่าที่ปรากฏใน **JTextField** จะถูกกำหนดให้เป็นช่องว่าง
- 4) ถ้าผู้ใช้กดปุ่ม "=" ค่าที่ปรากฏใน **JTextField** จะเป็นผลลัพธ์

กำหนดโค้ดสำหรับทดสอบความถูกต้องของอินเทอร์เฟซที่นักศึกษาได้พัฒนาขึ้นดังนี้
 โค้ด

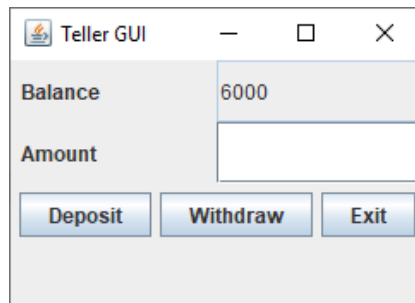
```

public class Main {
    public static void main(String[] args) {
        new CalculatorSample().init();
    }
}
  
```

4. ให้นักศึกษาร่างโปรแกรม **AccountSystem** โดยกำหนดคลาสไดอะแกรมต่อไปนี้



โดยที่ กำหนดให้นักศึกษานำส่วนต่อประสานกราฟิกกับผู้ใช้งานของคลาสชื่อ **"TellerGUI"** จาก **Lab Sheet 07** มาเปลี่ยนชื่อเป็นคลาส **"AccountView"** และแก้ไข



นอกจากนี้ ยังกำหนดรายละเอียดเพิ่มเติมดังต่อไปนี้

1. คลาส **AccountController** สำหรับ **default constructor** จะทำการสร้างอ็อบเจ็กต์จากคลาส **Account** และ **AccountView** ใหม่และกำหนดให้แอตทริบิวต์ **model** และ **view** ตามลำดับ
2. คลาส **AccountView** สำหรับ **constructor** จะนำค่าพารามิเตอร์ **amount** ไปให้แสดงใน **JTextField** ของช่อง **Balance** ใน **GUI** และกำหนดให้พารามิเตอร์ **c** เป็นตัวจัดการเหตุการณ์ (**Event Handle**)
3. คลาส **Account** เมธอด **withdraw()** และเมธอด **deposit()** ต้องใส่พารามิเตอร์ **amount** มีค่าเป็นตัวเลขตั้งแต่ 0 ขึ้นไป นอกจากนี้
 - 3.1. ถ้าผู้ใช้กดปุ่ม **"Deposit"** โปรแกรมจะหักยอดเงินตามค่าที่ปรากฏใน **JTextField** ของช่อง **Amount** ออกแอตทริบิวต์ **Balance** ของอ็อบเจ็กต์จากคลาส **Account** จากนั้น จึงอัปเดตค่าที่แสดงใน **JTextField** ของช่อง **Balance** ให้สอดคล้องกับค่าในแอตทริบิวต์ **Balance** สำหรับกรณีที่ยอดเงินแอตทริบิวต์ **Balance** ของอ็อบเจ็กต์จากคลาส **Account** มีเพียงพอ ถ้าไม่ใช่โปรแกรมจะไม่ดำเนินการใด ๆ

- 3.2. ถ้าผู้ใช้กดปุ่ม **"Withdraw"** โปรแกรมจะเพิ่มยอดเงินตามค่าที่ปรากฏใน **JTextField** ของช่อง **Amount** ลงในแอททริบิวต์ **Balance** ของอ็อบเจกต์จากคลาส **Account** จากนั้น จึงอัปเดตค่าที่แสดงใน **JTextField** ของช่อง **Balance** ให้สอดคล้องกับค่าในแอททริบิวต์ **Balance**

กำหนดโค้ดสำหรับทดสอบความถูกต้องของอินเทอร์เฟซที่นักศึกษาได้พัฒนาขึ้นดังนี้
โค้ด

```
public class Main {
    public static void main(String[] args) {
        AccountController c = new AccountController();
    }
}
```