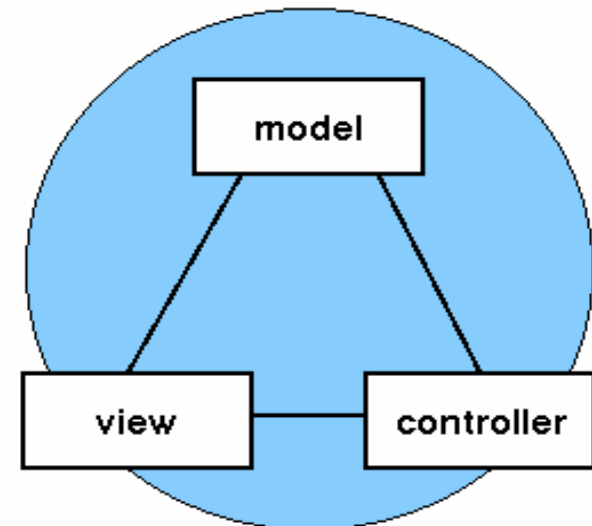

제8장 *MVC Model*

- **MVC (Model, View, Controller) 모델**
 - 스윙은 **MVC** 모델에 기초를 두고 있다.
 - **MVC**란 **Xerox**의 연구소에서 **Smalltalk** 언어를 바탕으로 사용자 인터페이스를 개발하기 위한 방법
 - **MVC**는 3개의 구성 요소로 구성
 - **Model** : 응용 프로그램의 자료를 표현하기 위한 모델
 - **View** : 자료를 시각적으로 (**GUI** 방식으로) 표현하는 것
 - **Controller**: 사용자가 **View**를 통해 입력을 하면 **Model**을 변경해주는 것

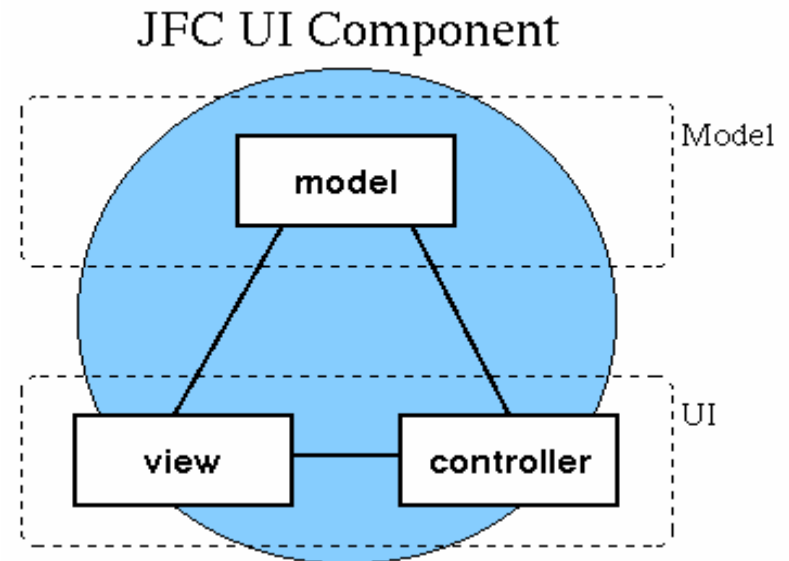


- MVC 모델

- 모델, 뷰, 컨트롤러를 분리한 MVC 모델은 상당히 유용하고 강력한 프로그래밍 모델이기 때문에 사용자 인터페이스 개발에 많이 적용
- 스윙을 개발하면서 MVC 모델을 그대로 적용하기에는 실제적인 어려움이 많아서 MVC를 변형한 모델을 개발.
- 스윙에서는 MVC의 뷰와 컨트롤러를 합쳐서 "UI delegate"라는 객체를 정의.

- MVC

- 모델 - 내부 상태, 자료 관리
- UI delegate - 모양 결정, 이벤트



- 스윙의 모든 **GUI** 컴포넌트들은 **UI delegate** 부분과 **Model** 부분으로 구성
- **UI delegate** 부분은 **GUI** 컴포넌트의 모양과 사용자의 입력에 따른 이벤트를 처리

- 모델의 역할
 - 내부 상태를 알아본다.
 - 내부 상태를 관리한다.
 - 이벤트 리스너를 추가/삭제 한다.
 - 이벤트를 발생시킨다.

- **UI delegate**의 역할
 - **GUI** 컴포넌트를 그려준다.
 - **GUI** 컴포넌트의 위치 정보를 리턴한다.
 - **AWT** 이벤트를 처리한다.



8.1 MVC 모델 (4/7)

- 각 컴포넌트의 모델 인터페이스와 모델 타입(I)

Component	Model Interface	Model Type
<u>JButton</u>	<u>ButtonModel</u>	GUI
<u>JToggleButton</u>	<u>ButtonModel</u>	GUI/data
<u>JCheckBox</u>	<u>ButtonModel</u>	GUI/data
<u>JRadioButton</u>	<u>ButtonModel</u>	GUI/data
<u>JMenu</u>	<u>ButtonModel</u>	GUI
<u>JMenuItem</u>	<u>ButtonModel</u>	GUI
<u>JCheckBoxMenuItem</u>	<u>ButtonModel</u>	GUI/data
<u>JRadioButtonMenuItem</u>	<u>ButtonModel</u>	GUI/data
<u>JComboBox</u>	<u>ComboBoxModel</u>	data
<u>JProgressBar</u>	<u>BoundedRangeModel</u>	GUI/data
<u>JScrollBar</u>	<u>BoundedRangeModel</u>	GUI/data
<u>JSlider</u>	<u>BoundedRangeModel</u>	GUI/data



8.1 MVC 모델 (5/7)

- 각 컴포넌트의 모델 인터페이스와 모델 타입(II)

Component	Model Interface	Model Type
<u>JTabbedPane</u>	<u>SingleSelectionModel</u>	GUI
<u>JList</u>	<u>ListModel</u>	data
<u>JList</u>	<u>ListSelectionModel</u>	GUI
<u>JTable</u>	<u>TableModel</u>	data
<u>JTable</u>	<u>TableColumnModel</u>	GUI
<u>JTree</u>	<u>TreeModel</u>	data
<u>JTree</u>	<u>TreeSelectionModel</u>	GUI
<u>JEditorPane</u>	Document	data
<u>JTextPane</u>	Document	data
<u>JTextArea</u>	Document	data
<u>JTextField</u>	Document	data
<u>JPasswordField</u>	Document	data

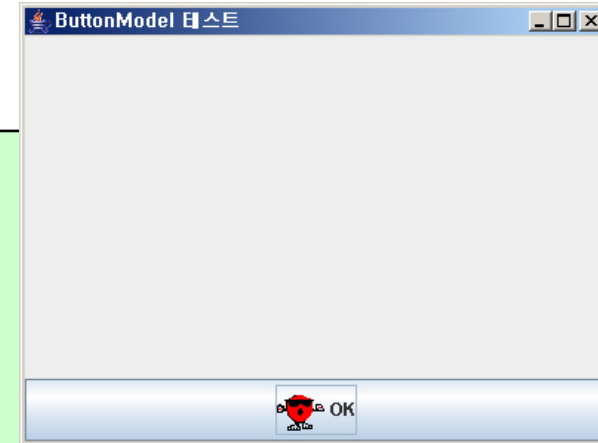


- 예제: `CountButtonModel.java`

```
4 public class CountButtonModel extends DefaultButtonModel {
5     private int count;
6     private JButton btn;
7
8     public CountButtonModel(JButton btn) {
9         this.btn = btn;
10        btn.setModel(this);
11    }
12
13    public void setPressed(boolean b) {
14        if(b) {
15            count = ++count % 4;
16            switch(count) {
17                case 0:
18                    btn.setBackground(Color.lightGray);
```

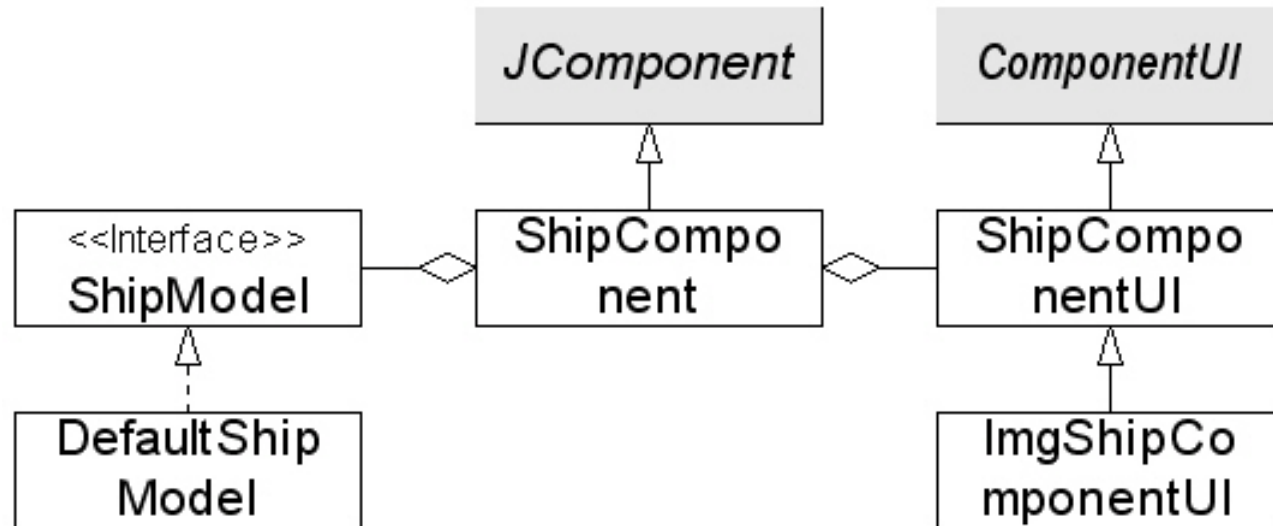
- 예제: `ButtonModelTest.java`

```
8     ImageIcon rai = new ImageIcon("rai.gif");
9     ok = new JButton("OK", rai);
10    ok.setRolloverIcon(rai);
11    ok.setPressedIcon(rai);
12    CountButtonModel model = new CountButtonModel(ok);
13
14    getContentPane().add("South", ok);
15    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
16    setSize(400, 300);
17    setVisible(true);
```



8.2 MVC 모델 예제(1/12)

- 새로운 컴포넌트(ShipComponent) 만들기



- MVCtest.java
 - ShipModel.java
 - ShipComponent.java
 - ShipComponentUI.java
 - DefaultShipModel.java
 - ImgShipComponent.java



- 예제: `ShipComponent.java`

```
5 public class ShipComponent extends JComponent {
6     private ShipModel      model;
7     private ShipComponentUI ui;
8     private ActionListener  actionListener;
9     private boolean        pressed;
10
11     public ShipComponent(String s, String d) {
12         enableEvents(AWTEvent.MOUSE_EVENT_MASK);
13         model = new DefaultShipModel(s, d);
14         setUI(new ShipComponentUI());
15     }
```



- 예제: `ShipComponent.java`(계속)

```
22     public void paint(Graphics g) {
23         ui.paint(this, g);
24     }
...
35     public void setUI(ShipComponentUI ui) {
36         if(ui != null && this.ui != null) {
37             this.ui.uninstallUI(this);
38         }
40         if(ui != null) {
41             this.ui = ui;
42             this.ui.installUI(this);
43         }
44         updateShipComponent();
```

8.2 MVC 모델 예제(5/12)

- 예제: `ShipComponent.java`(계속)

```
51     public Dimension getPreferredSize() {
52         return ui.getPreferredSize(this);
53     }
...
67     public boolean isPressed() {
68         return pressed;
69     }
70
71     public void addActionListener(ActionListener listener) {
72         actionListener =
73             AWTEventMulticaster.add(actionListener, listener);
74     }
```

- 예제: `ShipComponent.java`(계속)

```
81     public void processMouseEvent(MouseEvent e) {
82         Graphics g;
83         switch(e.getID()) {
...
88             case MouseEvent.MOUSE_RELEASED:
89                 if(actionListener != null) {
90                     actionPerformed(new ActionEvent(this,
91                         ActionEvent.ACTION_PERFORMED, model.getData()));
92                 }
93                 if(pressed == true) {
94                     pressed = false;
95                     repaint();
```

8.2 MVC 모델 예제(7/12)

- 예제: `ShipComponentUI.java`

```
5 public class ShipComponentUI extends ComponentUI {
6     private Dimension dim;
...
32     public void paint(ShipComponent ship, Graphics g) {
...
37         int xs[] = {2, d.width-2, 2*d.width/3, d.width/3, 2};
38         int ys[] = { 2, 2, d.height -2, d.height - 2, 2};
39         Color fg = g.getColor();
40         g.setColor(new Color(200, 200, 200));
41         g.fillPolygon(xs, ys, xs.length);
42         g.setColor(fg);
43         if(ship.isPressed()) {
44             g.drawPolygon(xs, ys, xs.length);
```

8.2 MVC 모델 예제(8/12)

- 예제: `ImgShipComponentUI.java`

```
3 public class ImgShipComponentUI extends ShipComponentUI {
4     private Image    img;
...
37     public void paint(ShipComponent ship, Graphics g) {
38         Dimension d = ship.getSize();
39         ShipModel model = ship.getModel();
40         ship.setToolTipText(model.getData());
41         if(img != null) {
42             int x = (d.width - img.getWidth(ship))/2;
43             int y = (d.height - img.getHeight(ship))/2;
44             g.drawImage(img, x, y, ship);
45         }
46         if(ship.isPressed()) {
47             g.drawRect(1,1,d.width-2, d.height-2);
```


8.2 MVC 모델 예제(9/12)

- 예제: ShipModel.java

```
1 public interface ShipModel {  
2     public String getData();  
3 }
```



8.2 MVC 모델 예제(10/12)

- 예제: DefaultShipModel.java

```
1 public class DefaultShipModel implements ShipModel {
2     private String dest;
3     private String start;
4
5     public DefaultShipModel(String s, String d) {
6         start = s;
7         dest = d;
8     }
9
10    public String getData() {
11        return start + " -> " + dest;
```



8.2 MVC 모델 예제(11/12)

- 예제: **MVCTest.java**

```
4 public class MVCTest extends JFrame implements ActionListener {
6     private ShipComponent toLA, toTokyo;
...
12     toLA = new ShipComponent("Seoul", "LA");
13     toLA.addActionListener(this);
14     map.add(toLA);
...
19     toTokyo = new ShipComponent("Seoul", "Tokyo");
20     toTokyo.addActionListener(this);
21     toTokyo.setUI(new ImgShipComponentUI(this, "ship.jpg"));
22     map.add(toTokyo);
23     toTokyo.setLocation(250, 130);
24     toTokyo.setSize(toTokyo.getPreferredSize());
```

8.2 MVC 모델 예제(12/12)

- 결과

