

Java Programming (0409)

Swing Cheet Sheet 2

Yulwon Rhee (202211342)

Department of Computer Science and Engineering, Konkuk University

1 Method related with Components

1.1 Components' look

```
1 void setForeground(Color)
2 void setBackground(Color)
3 void setOpaque(boolean)
4 void setFont(Font)
5 Font getFont()
```

1.2 Components' Location and Size

```
1 int getWidth()
2 int getHeight()
3 int getX()
4 int getY()
5 Point getLocationOnScreen()
6 void setLocation(int, int)
7 void setSize(int, int)
```

1.3 Components' State

```
1 void setEnabled(boolean)
2 void setVisible(boolean)
3 boolean isVisible()
```

1.4 Methods for Containers

```
1 Component add(Component)
2 void remove(Component)
3 void removeAll()
4 Component[] getComponents()
5 Container getParent()
6 Container getTopLevelAncestor()
```

2 Draw Image with JLabel

```
1 ImageIcon image = new ImageIcon("images/apple.jpg"); JLabel label = new JLabel(image);
2 panel.add(label);
```

3 JCheckBox

```

1 public class CheckBoxItemEventEx extends JFrame {
2     ^IContainer contentPane;
3     ^IJCheckBox [] fruits = new JCheckBox [3];
4     ^IString [] names = {"사과", "배", "체리"};
5     ^IJLabel sumLabel;
6     ^Iint sum = 0;
7
8     ^ICheckBoxItemEventEx() {
9         ^I^IsetTitle("체크박스와 ItemEvent 예제");
10        ^I^IsetDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11        ^I^IcontentPane = getContentPane();
12        ^I^IcontentPane.setLayout(new FlowLayout());^I^I^I^I^I^I^I^I
13        ^I^IcontentPane.add(new JLabel("사과 100원, 배 500원, 체리 20000원"));
14        ^I^Ifor(int i = 0; i < fruits.length; i++) {
15            ^I^I^Ifruits[i] = new JCheckBox(names[i]);
16            ^I^I^Ifruits[i].setBorderPainted(true);
17            ^I^I^IcontentPane.add(fruits[i]);
18            ^I^I^Ifruits[i].addItemListener(new MyItemListener());
19        ^I^I}
20        ^I^IsumLabel = new JLabel("현재 0원 입니다.");
21        ^I^IcontentPane.add(sumLabel);
22        ^I^IsetSize(250, 200);
23        ^I^IsetVisible(true);
24        ^I}
25
26        ^Iclass MyItemListener implements ItemListener {
27            ^I^Ipublic void itemStateChanged(ItemEvent e) {
28                ^I^I^Iint selected = 1;
29                ^I^I^Iif(e.getStateChange() == ItemEvent.SELECTED)
30                    ^I^I^Iselected = 1;
31                ^I^I^Ielse
32                    ^I^I^Iselected = -1;
33                ^I^I^I
34                ^I^I^Iif(e.getItem() == fruits[0])
35                    ^I^I^Isum = sum + selected * 100;
36                ^I^I^Ielse if(e.getItem() == fruits[1])
37                    ^I^I^Isum = sum + selected * 500;
38                ^I^I^Ielse
39                    ^I^I^Isum = sum + selected * 20000;
40
41                ^I^I^IsumLabel.setText("현재 " + sum + "원 입니다.");
42            ^I^I}
43        ^I}
44    }

```

4 JSlider

```

1 slider = new JSlider();
2 slider.setBorder(BorderFactory.createEmptyBorder(0, 20, 0, 20));
3 slider.setMinimum(0);
4 slider.setMaximum(15000);
5 slider.addChangeListener(e -> {
6     weightLabel.setText(weightFormatter(slider.getValue()));
7
8     if (slider.getValue() >= MAXIMUM_WEIGHT * 100) {
9         logArea.setText(logArea.getText().trim() + "\nWeight Over " + weightFormatter((int)
10             ↪ MAXIMUM_WEIGHT * 100) + " Detected");

```

```

10     weightLabel.setForeground(Color.decode("#FF3B30"));
11     stopEngine();
12 }
13 });
14 slider.setMinorTickSpacing(100);
15 slider.setMajorTickSpacing(1000);
16 slider.setPaintTicks(true);
17
18 Hashtable<Integer, JLabel> sliderLabel = new Hashtable<>();
19 for (int i = 0; i <= 15000; i += 5000) {
20     sliderLabel.put(i, new JLabel(String.valueOf(i / 100)));
21 }
22 sliderLabel.put((int) (MAXIMUM_WEIGHT * 100), new JLabel("<html><font color='#FF3B30'><b>" +
23     ↪ (int) (MAXIMUM_WEIGHT) + "</b></font></html>"));
24 slider.setLabelTable(sliderLabel);
25 slider.setPaintLabels(true);
26 leftPanel.add(slider, BorderLayout.CENTER);

```

5 Draw Image with JLabel

```

1 ImageIcon image = new ImageIcon("images/apple.jpg"); JLabel label = new JLabel(image);
2 panel.add(label);

```

6 Image JButton

```

1 ImageIcon normalIcon = new ImageIcon("images/normalIcon.gif");
2 ImageIcon rolloverIcon = new ImageIcon("images/rolloverIcon.gif");
3 ImageIcon pressedIcon = new ImageIcon("images/pressedIcon.gif");
4
5 JButton button = new JButton("테스트버튼", normalIcon);
6 button.setRolloverIcon(rolloverIcon);
7 button.setPressedIcon(pressedIcon);

```

7 Image JCheckBox

```

1 ImageIcon cherryIcon = new ImageIcon("images/cherry.jpg");
2 ImageIcon selectedCherryIcon = new ImageIcon("images/selectedCherry.jpg");
3 JCheckBox cherry = new JCheckBox("체리", cherryIcon);
4 cherry.setSelectedIcon(selectedCherryIcon);

```

8 JTextField

```

1 JTextField(String text, int cols)
2 JTextField.select(0, 5); // Select from index 0 letter to index 5 letter

```

9 JTextArea

```

1 JTextArea(String text, int rows, int cols);
2
3 public class TextAreaEx extends JFrame {
4     JTextField tf = new JTextField(20);

```

```

5      JTextArea ta = new JTextArea(7, 20);
6
7      TextAreaEx() {
8          setTitle("텍스트영역 만들기 예제");
9          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10         Container c = getContentPane();
11         c.setLayout(new FlowLayout());
12         c.add(new JLabel("입력 후 <Enter> 키를 입력하세요"));
13         c.add(tf);
14         c.add(new JScrollPane(ta));
15
16         tf.addActionListener(new ActionListener() {
17             public void actionPerformed(ActionEvent e) {
18                 JTextField t = (JTextField)e.getSource();
19                 ta.append(t.getText() + "\n"); t.setText("");
20             }
21         });
22         setSize(300,300);
23         setVisible(true);
24     }
25 }

```

10 JComboBox

```

1  JComboBox()
2  JComboBox(ComboBoxModel model)
3  JComboBox(Object[] items)
4  JComboBox(Vector items)

```

콤보박스의 아이тем 선택시 Action 이벤트 발생

ActionListener 이용

콤보박스의 아이тем의 선택시 Item 이벤트 발생

ItemListener 이용

새로운 아이тем이 선택되면 두 번의 Item 이벤트 발생

새로 아이тем이 선택되었음을 알리는 Item 이벤트 발생

이전에 선택된 아이тем이 해제됨을 알리는 Item 이벤트 발생

선택 상태인 아이тем을 선택하는 경우 Item 이벤트 발생 않음

현재 선택된 아이тем 알아내기

`int JComboBox.getSelectedIndex()` : 선택 상태인 아이тем의 인덱스 번호 리턴

`Object JComboBox.getSelectedItem()` : 선택 상태인 아이тем 객체의 레퍼런스 리턴

```

1  public class ComboActionEx extends JFrame {
2      String [] fruits = {"apple", "banana", "mango"};
3      ImageIcon [] images = { new ImageIcon("images/apple.jpg"),
4                              new ImageIcon("images/banana.jpg"),
5                              new ImageIcon("images/mango.jpg") };
6      JLabel imgLabel = new JLabel(images[0]);
7
8      ComboActionEx() {
9          setTitle("콤보박스 활용 예제");
10         Container c = getContentPane();
11         c.setLayout(new FlowLayout());
12         JComboBox combo = new JComboBox(fruits);

```

```

13     c.add(combo);
14     c.add(imgLabel);
15
16     // 콤보박스에 Action 리스너 등록. 선택된 아이템의 이미지 출력
17     combo.addActionListener(new ActionListener() {
18         public void actionPerformed(ActionEvent e) {
19             JComboBox cb = (JComboBox)e.getSource();
20             int index = cb.getSelectedIndex();
21             imgLabel.setIcon(images[index]);
22         }
23     });
24     setSize(300, 250);
25     setVisible(true);
26 }
27 }

```

11 JMenu

```

1  public class MenuActionEventEx extends JFrame {
2      Container contentPane;
3      JLabel label = new JLabel("Hello");
4
5      MenuActionEventEx() {
6          setTitle("Menu 만들기 예제");
7          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
8          contentPane = getContentPane();
9          label.setHorizontalAlignment(SwingConstants.CENTER);
10         contentPane.add(label, BorderLayout.CENTER);
11         createMenu();
12         setSize(250, 200);
13         setVisible(true);
14     }
15
16     void createMenu() {
17         JMenuBar mb = new JMenuBar();
18         JMenuItem [] menuItem = new JMenuItem [4];
19         String[] itemTitle = {"Color", "Font", "Top", "Bottom"};
20         JMenu textMenu = new JMenu("Text");
21         for(int i=0; i<menuItem.length; i++) {
22             menuItem[i] = new JMenuItem(itemTitle[i]);
23             menuItem[i].addActionListener(new MenuActionListener());
24             textMenu.add(menuItem[i]);
25         }
26         mb.add(textMenu);
27         this.setJMenuBar(mb);
28     }
29
30     class MenuActionListener implements ActionListener {
31         public void actionPerformed(ActionEvent e) {
32             String cmd = e.getActionCommand();
33             if(cmd.equals("Color"))
34                 label.setForeground(Color.BLUE);
35             else if(cmd.equals("Font"))
36                 label.setFont(new Font("Ravie", Font.ITALIC, 30));
37             else if(cmd.equals("Top"))
38                 label.setVerticalAlignment(SwingConstants.TOP);
39             else
40                 label.setVerticalAlignment(SwingConstants.BOTTOM);
41         }
42     }
43 }

```

```

42     }
43 }

```

12 JToolBar

툴바를 구현한 컴포넌트

여러 컴포넌트를 담을 수 있는 컨테이너

툴바의 모양

한 행 혹은 한 열로만 표현

기능

버튼이나 이미지 등 모든 컴포넌트를 부착하여 이들을 메뉴처럼 사용

툴바가 부착되는 위치

툴바는 BorderLayout 배치 관리자를 가진 컨테이너에만 부착

상(NORTH), 하(SOUTH), 좌(WEST), 우(EAST) 측의 모서리 중 선택 부착

사용자의 드래그에 의해 위의 4가지 위치에 이동 부착 가능

사용자의 드래그에 의해 독립적인 다이얼로그 형태로 떨어져서 존재할 수 있음

사용자의 드래그에 의한 이동이 불가능하게 할 수 있음

```

1 public class ToolBarEx extends JFrame {
2     Container contentPane; ToolBarEx() {
3         setTitle("툴바 만들기 예제");
4         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
5         contentPane = getContentPane();
6         createToolBar();
7         setSize(400, 200);
8         setVisible(true);
9     }
10
11     void createToolBar() {
12         JToolBar toolBar = new JToolBar("Kitae Menu");
13         bar.setBackground(Color.LIGHT_GRAY);
14         toolBar.add(new JButton("New"));
15         toolBar.add(new JButton(new ImageIcon("images/open.jpg")));
16         toolBar.addSeparator();
17         toolBar.add(new JButton(new ImageIcon("images/save.jpg")));
18         toolBar.add(new JLabel("search"));
19         toolBar.add(new JTextField("text field"));
20         JComboBox combo = new JComboBox();
21         combo.addItem("Java");
22         combo.addItem("C#");
23         combo.addItem("C");
24         combo.addItem("C++");
25         toolBar.add(combo);
26         contentPane.add(toolBar, BorderLayout.NORTH);
27     }
28 }

```

13 Dialog

13.1 Input Dialog

```
String JOptionPane.showInputDialog(String msg)
```

13.2 Confirm Dialog

```

1  int result = JOptionPane.showConfirmDialog(null, "계속할 것입니까?", "Confirm",
    ↳ JOptionPane.YES_NO_OPTION);
2  if(result == JOptionPane.CLOSED_OPTION) {
3      // 사용자가 "예"나 "아니오"의 선택없이 다이얼로그 창을 닫은 경우
4  } else if(result == JOptionPane.YES_OPTION) {
5      // 사용자가 "예"를 선택한 경우
6  } else {
7      // 사용자가 "아니오"를 선택한 경우
8  }

```

13.3 Message Dialog

```

1  JOptionPane.showMessageDialog(null, "조심하세요", "Message", JOptionPane.ERROR_MESSAGE);
2  JOptionPane.showMessageDialog(null, "조심하세요", "Message", JOptionPane.INFORMATION_MESSAGE);
3  JOptionPane.showMessageDialog(null, "조심하세요", "Message", JOptionPane.WARNING_MESSAGE);
4  JOptionPane.showMessageDialog(null, "조심하세요", "Message", JOptionPane.PLAIN_MESSAGE);

```

14 JColorChooser

```

1  public class MenuAndColorChooserEx extends JFrame {
2      Container contentPane;
3      JLabel label = new JLabel("Hello");
4      MenuAndColorChooserEx() {
5          setTitle("JColorChooser 예제");
6          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
7          contentPane = getContentPane();
8          label.setHorizontalAlignment(SwingConstants.CENTER);
9          label.setFont(new Font("Ravie", Font.ITALIC, 30));
10         contentPane.add(label, BorderLayout.CENTER);
11         createMenu();
12         setSize(250, 200);
13         setVisible(true);
14     }
15     void createMenu() {
16         JMenuBar mb = new JMenuBar();
17         JMenuItem colorMenuItem = new JMenuItem("Color");
18         JMenu fileMenu = new JMenu("Text");
19         colorMenuItem.addActionListener(new MenuActionListener());
20         fileMenu.add(colorMenuItem);
21         mb.add(fileMenu);
22         this.setJMenuBar(mb);
23     }
24
25     class MenuActionListener implements ActionListener {
26         JColorChooser chooser = new JColorChooser();
27
28         public void actionPerformed(ActionEvent e) {
29             String cmd = e.getActionCommand();
30             if(cmd.equals("Color")) {
31                 Color selectedColor = chooser.showDialog(null, "Color", Color.YELLOW);
32
33                 if(selectedColor != null)
34                     label.setForeground(selectedColor);
35             }
36         }
37     }
38 }

```