

デザイナーを使った実装5(コンボボックスの選択表示)

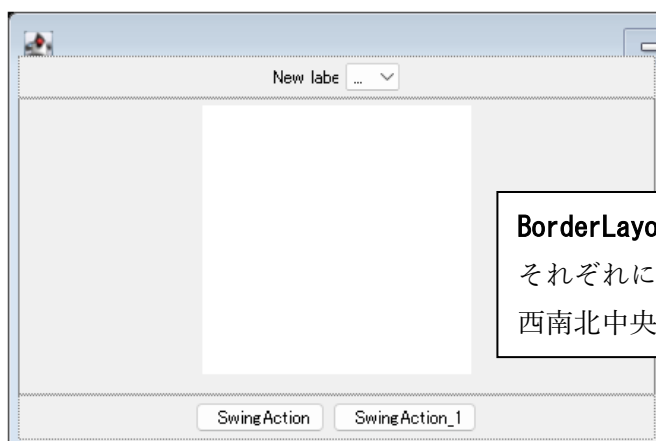
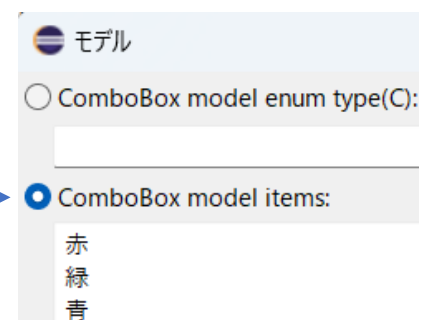
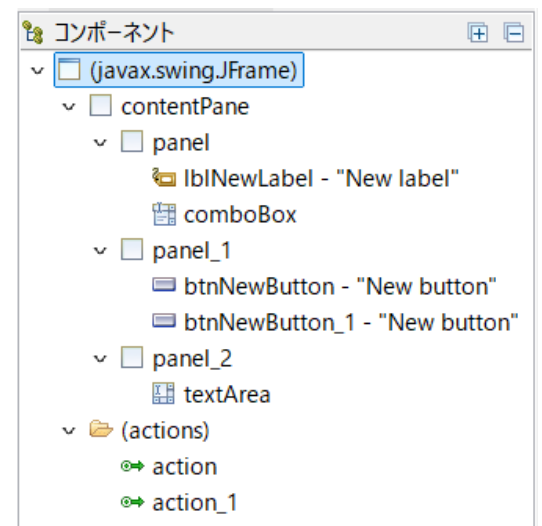
ボタンをクリックするとコンボボックスの選択したインデックスと内容がテキストエリア
に出力される GUI アプリケーションを作成する(プロジェクト名:SwingChapter03)

1. クラスファイル作成

メニューバー → ファイル → 新規 → その他 → WindowBuilder
→ Swing デザイナー → JFrame
→ パッケージ名 :jp.ict.swing クラス名 :Exec340

2. GUI デザイン作成(デザインタブ)

【パレット】	【配置場所】
① BorderLayout	→ contentPane
② JPanel	→ contentPane (NORTH 位置)
③ JLabel	→ panel
④ JComboBox プロパティ	→ panel model [赤,緑,青]
⑤ JPanel	→ contentPane (SOUTH 位置)
⑧ JButton	→ panel_1
⑩ JButton	→ panel_1
⑨ SwingActions 新規	→ JButton 位置(SwingAction)
⑪ SwingActions 新規	→ JButton 位置(SwingAction_1)
⑥ JPanel	→ contentPane (CENTER 位置)
⑦ JTextArea プロパティ	→ panel_2 columns 30 rows 10



BorderLayout は、画面を上下左右中の 5つのブロックに分け、それぞれに部品を配置します。5つのブロックはそれぞれ、東西南北中央(East, West, South, North, Center)で示されます。

3. ソース変更(ソースタブ)

① //koko の部分を追加(イベント処理の記述)

```
private class SwingAction extends AbstractAction {  
    public SwingAction() {  
        putValue(NAME, "SwingAction");  
        putValue(SHORT_DESCRIPTION, "Some short description");  
    }  
    public void actionPerformed(ActionEvent e) {  
        textArea.append(comboBox.getSelectedIndex() + " "); //koko  
        textArea.append(comboBox.getSelectedItem() + "¥n"); //koko  
    }  
}  
  
private class SwingAction_1 extends AbstractAction {  
    public SwingAction_1() {  
        putValue(NAME, "SwingAction_1");  
        putValue(SHORT_DESCRIPTION, "Some short description");  
    }  
    public void actionPerformed(ActionEvent e) {  
        textArea.setText(null); //koko  
    }  
}
```

② //koko の部分を追加(インスタンス変数の宣言追加)

```
public class Exec340 extends JFrame {  
  
    private JPanel contentPane;  
    private final Action action = new SwingAction();  
    private final Action action_1 = new SwingAction_1();  
  
    private JComboBox comboBox; //koko  
    private JTextArea textArea; //koko  
  
    /**  
     * Launch the application.  
     */  
}
```

③ //koko の部分を変更(ローカル変数の宣言削除)

```
comboBox = new JComboBox(); //koko  
comboBox.setModel(new DefaultComboBoxModel(new String[] { "赤", "緑", "青" }));  
panel.add(comboBox);  
  
textArea = new JTextArea(); //koko  
textArea.setRows(10);  
textArea.setColumns(30);  
panel_2.add(textArea);
```

4. 実行

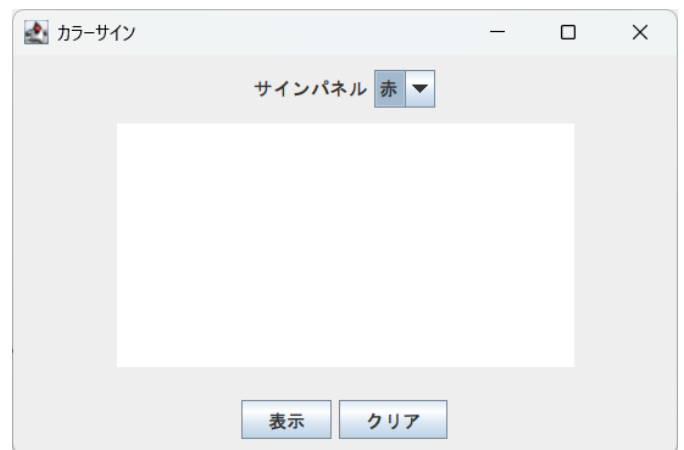
以下実行結果の例



5. 表示名の変更

右のように各オブジェクトの内容を変更してください

- ・タイトルを付けます
 - ・ラベルの表示を変更します
 - ・ボタンの表示を変更します
- あわせてツールチップ(注釈補足情報)も設定します



ソースタブでもデザインタブでも変更可能ですが、**デザインタブのプロパティだけでは設定できない項目がありますのでその時はソースを直接変更します。**

【演習1】

上記プログラムを以下のように
選択した色がコンボボックスに反映されるようにしてください

コンボボックスの項目選択は index で判断できます
コンボボックスフィールド変数名.getSelectedIndex();

コンボボックスの色の指定処理は次のようにして実装できます
コンボボックスフィールド変数名.setForeground(Color.red);
※この場合 combobox 全体の色が変わります



【ヒント】

赤を指定するとき

```
if (comboBox.getSelectedIndex() == 0) { // 赤 ↓
    comboBox.setForeground(Color.red);
    textArea.setForeground(Color.red);
}
```