

デザイナーを使った実装3(チェックボックスの選択表示)

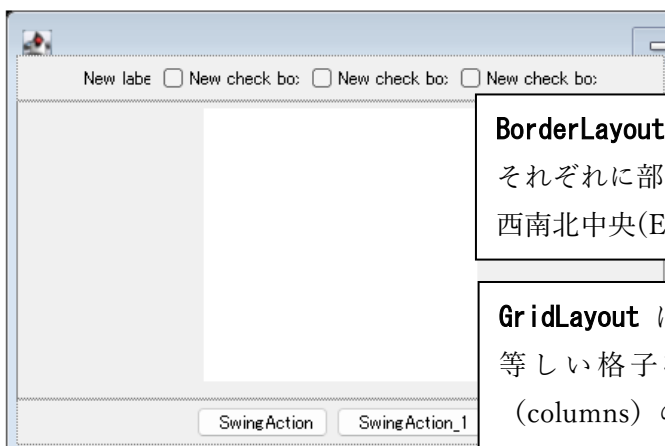
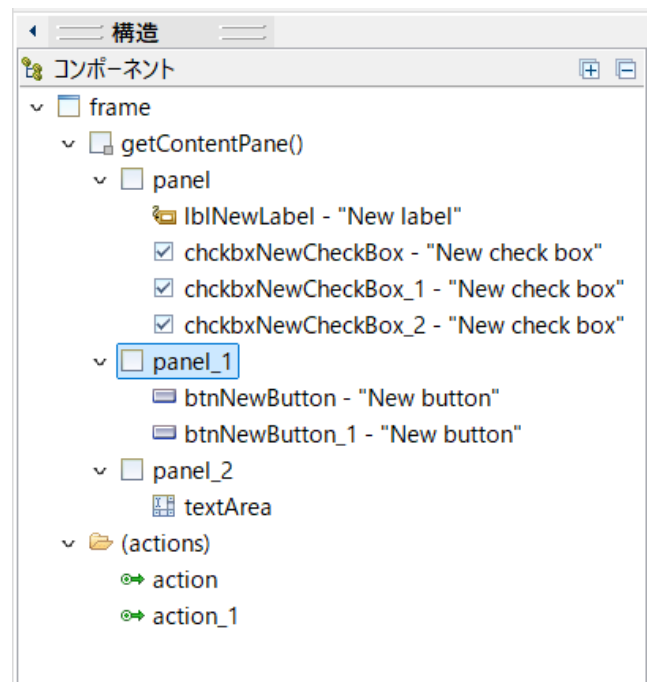
ボタンをクリックするとチェックボックスのチェックした内容がテキストエリアに出力される GUI アプリケーションを作成する(プロジェクト名:Swing)

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 → アプリケーション・ウィンドウ
→ パッケージ名 :jp.ict.aso.swing クラス名 :Exec320

2. GUI デザイン作成(デザインタブ)

【パレット】		【配置場所】
① BorderLayout	→	FRAME 全体
② JPanel	→	NORTH 位置
③ JLabel	→	NORTH 位置
④ JCheckBox	→	NORTH 位置
⑤ JCheckBox	→	NORTH 位置
⑥ JCheckBox	→	NORTH 位置
⑦ JPanel	→	SOUTH 位置
⑧ JPanel	→	CENTER 位置
⑨ JTextArea	→	CENTER 位置
プロパティ	→	columns 30 rows 10
⑩ JButton	→	SOUTH 位置
⑪ SwingActions 新規	→	JButton 位置
⑫ JButton	→	SOUTH 位置
⑬ SwingActions 新規	→	JButton 位置



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。

GridLayout は、格子状にレイアウトを組む手法です。サイズの等しい格子状の升目に部品を配置します。設定時は列 (columns) の個数を指定するのが効率的です。

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(chckbxNewCheckBox.getText()+"[""); //koko  
        textArea.append(chckbxNewCheckBox.isSelected()+"¥n"); //koko  
        textArea.append(chckbxNewCheckBox_1.getText()+"[""); //koko  
        textArea.append(chckbxNewCheckBox_1.isSelected()+"¥n"); //koko  
        textArea.append(chckbxNewCheckBox_2.getText()+"[""); //koko  
        textArea.append(chckbxNewCheckBox_2.isSelected()+"¥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 Exec320 {  
    private JFrame frame;  
    private final Action action = new SwingAction();  
    private final Action action_1 = new SwingAction_1();  
    private JCheckBox chckbxNewCheckBox; //koko  
    private JCheckBox chckbxNewCheckBox_1; //koko  
    private JCheckBox chckbxNewCheckBox_2; //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)

```
JLabel lblNewLabel = new JLabel("New label"); ↓
panel.add(lblNewLabel); ↓

chckbxNewCheckBox = new JCheckBox("New check box"); //koko ↓
panel.add(chckbxNewCheckBox); ↓

chckbxNewCheckBox_1 = new JCheckBox("New check box"); //koko ↓
panel.add(chckbxNewCheckBox_1); ↓

chckbxNewCheckBox_2 = new JCheckBox("New check box"); //koko ↓
panel.add(chckbxNewCheckBox_2); ↓

JPanel panel_1 = new JPanel(); ↓
frame.getContentPane().add(panel_1, BorderLayout.SOUTH); ↓
```

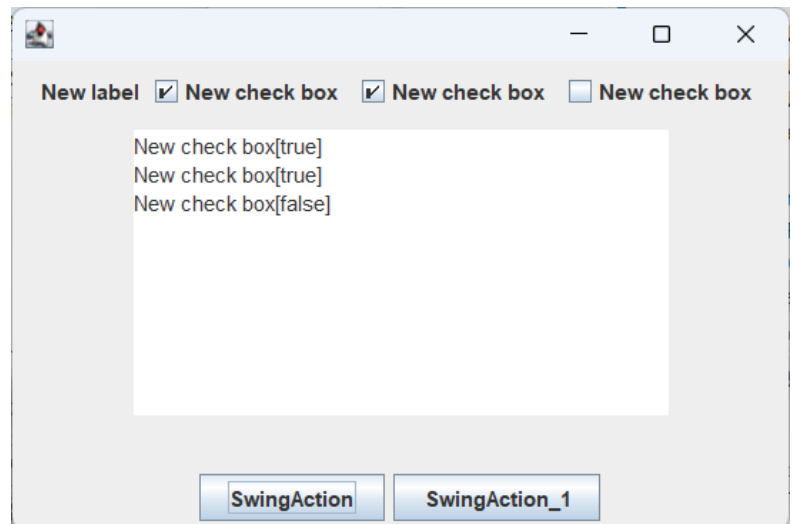
4. 実行

右は実行結果の例

上記①～③までの実装
ではテキストエリアへの
書き込み段階でエラー
となります。
適切に対処してください。

【ヒント】

ローカル変数の宣言削除

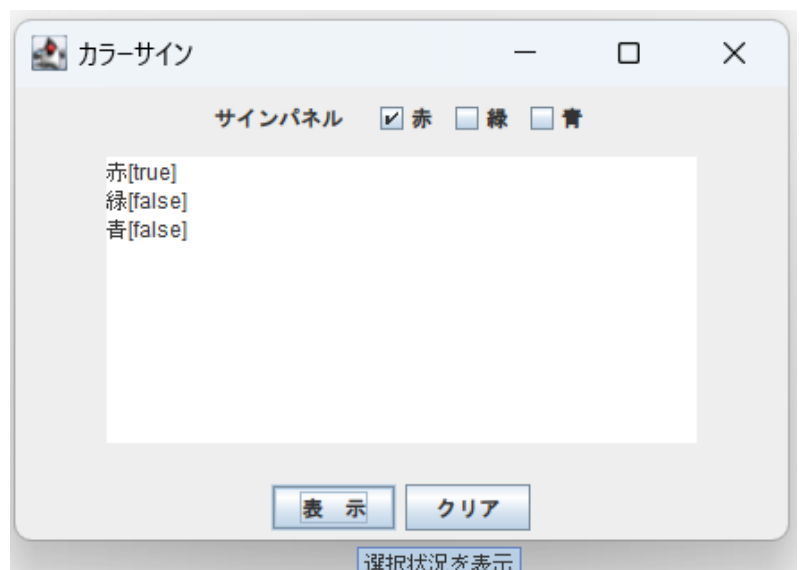


5. 表示名の変更

以下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。



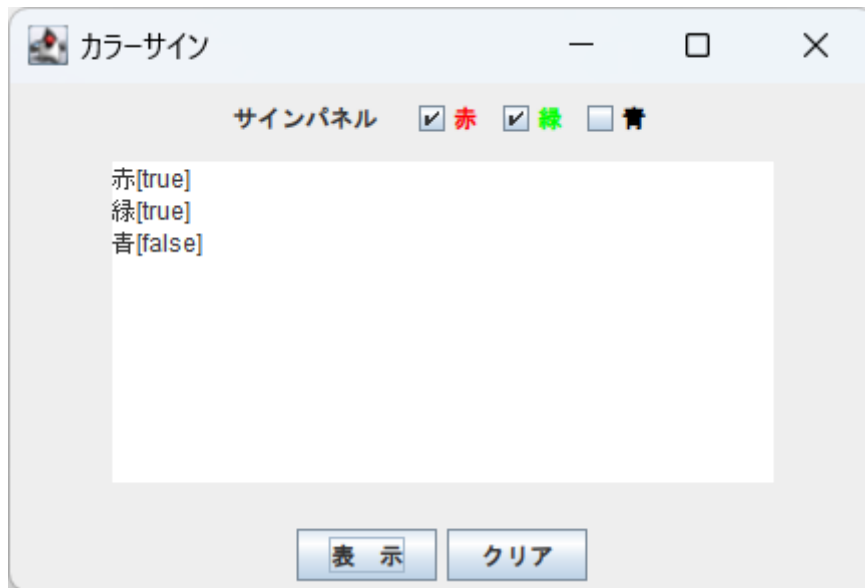
【演習1】

上記プログラムを以下のように

チェックした色がチェックボックスに反映されるように変更してください

チェック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

チェックボックスフィールド変数名.setForeground(Color.red);



【ヒント】

赤を指定するとき

```
//赤色
if(chckbxNewCheckBox.isSelected()) {
    chckbxNewCheckBox.setForeground(Color.red);
    textArea.append(chckbxNewCheckBox.getText()+"[");
    textArea.append(chckbxNewCheckBox.isSelected()+""]\n");
} else {
    chckbxNewCheckBox.setForeground(Color.black);
    textArea.append(chckbxNewCheckBox.getText()+"[");
    textArea.append(chckbxNewCheckBox.isSelected()+""]\n");
}
```

【演習2】

SOUTH 位置の JPanel のレイアウトマネージャを GridLayout に変更して、以下のようなボタンに表示が変更されるか確認してください。



以下のような部品配置に変更してください。

