## デザイナーを使った実装4(ラジオボックスの選択表示)

ボタンをクリックするとラジオ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)

※ラジオボタンはグループ化してグループ内でただ一つだけチェックできるようにしないといけません。

1. クラスファイル作成

メニューバー  $\rightarrow$  ファイル  $\rightarrow$  新規  $\rightarrow$  その他  $\rightarrow$  WindowBuilder

→ Swing  $\vec{r}$   $\vec{r}$   $\vec{r}$   $\vec{r}$   $\vec{r}$   $\vec{r}$   $\vec{r}$   $\vec{r}$ 

→ パッケージ名:jp.ict.aso.swing クラス名:Exec321

2. GUI デザイン作成(デザインタブ)

【パレット】		【配置場所】	▲ === 構造 ====					
① BorderLayout	$\rightarrow$	FRAME 全体	<sup>1</sup> <sup>1</sup>					
② JPanel	$\rightarrow$	NORTH 位置	frame getContentPane()					
③ JLabel	$\rightarrow$	NORTH 位置	v 🗌 panel					
4 JRadioButton	$\rightarrow$	NORTH 位置	CallblNewLabel - "New label"					
(5) JRadioButton	$\rightarrow$	NORTH 位置	<ul> <li>rdbtnNewRadioButton - "New radio button"</li> <li>rdbtnNewRadioButton 1 - "New radio button"</li> </ul>					
6 JRadioButton		→ NORT	<ul> <li>rdbtnNewRadioButton_2 - "New radio button"</li> </ul>					
⑦ JPanel	$\rightarrow$	SOUTH 位置	✓ □ panel_1					
⑧ JPanel	$\rightarrow$	CENTER 位置	btnNewButton - "New button"					
④ JTextArea	$\rightarrow$	CENTER 位置	✓ panel_2					
プロパティ	$\rightarrow$	colums 30	🔛 textArea					
		rows 10	✓ i (actions)					
1 JButton	$\rightarrow$	SOUTH 位置	⇔ action_1					
① SwingActions 新規	$\rightarrow$	JButton 位置						
12 JButton	$\rightarrow$	SOUTH 位置						
① SwingActions 新規	$\rightarrow$	JButton 位置						
New labe O New radio buttor O New radio buttor								
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	南北中央(East, West, South, North, Center)で示されます。					
SwingAction	SwingAction_	1						

```
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 (rdbtnNewRadioButton_getText()+"["); //koko
textArea. append (rdbtnNewRadioButton_isSelected()+"]¥n"); //koko
textArea. append (rdbtnNewRadioButton_1.getText()+"["); //koko
textArea. append (rdbtnNewRadioButton_1.isSelected()+"]¥n"); //koko
textArea. append (rdbtnNewRadioButton_2.getText()+"["); //koko
textArea. append (rdbtnNewRadioButton_2.getText()+"["); //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 ↓
       }
} ↓
```

```
    //kokoの部分を追加(インスタンス変数の宣言追加)
```

```
public class Exec321 {
    private JFrame frame; |
    private final Action action = new SwingAction(); |
    private final Action action_1 = new SwingAction_1(); |
    private JRadioButton rdbtnNewRadioButton; //koko
    private JRadioButton rdbtnNewRadioButton_1; //koko
    private JRadioButton rdbtnNewRadioButton_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 |b|NewLabe| = new JLabel("New |abe|");↓ panel.add(|b|NewLabe|);↓

rdbtnNewRadioButton = new JRadioButton("New radio button"); //koko |
panel.add(rdbtnNewRadioButton); |

rdbtnNewRadioButton\_1 = new JRadioButton("New radio button");///koko |
panel.add(rdbtnNewRadioButton\_1); |

rdbtnNewRadioButton\_2 = new JRadioButton("New radio button");//koko |
panel.add(rdbtnNewRadioButton\_2); |

JPanel **panel\_1 = new** JPanel(); frame.getContentPane().add(panel\_1, BorderLayout.**SOUTH**);



この状態では横幅が狭く3つのラジオボタンがすべて表示されていません。また、ラジ オボタンが複数選択されてしまいます。

よって、Windowの横幅を広げ、ラジオボタンをグループ化する処理を追加します。

```
【横幅の変更】
```

```
/** |
* Initialize the contents of the frame. |
*/ |
private void initialize() { |
frame = new JFrame(); |
frame.setBounds(100, 100, 500, 300); | //koko |
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); |
frame.getContentPane().setLayout(new BorderLayout(0, 0)); |
```

## 【ラジオボタンのグループ化】

rdbtnNewRadioButton = new JRadioButton("New radio button"); //koko |
panel.add(rdbtnNewRadioButton); |

rdbtnNewRadioButton\_1 = new JRadioButton("New radio button");///koko |
panel.add(rdbtnNewRadioButton\_1); |

rdbtnNewRadioButton\_2 = new JRadioButton("New radio button");//koko |
panel.add(rdbtnNewRadioButton\_2); |

```
ButtonGroup buttongp = new ButtonGroup(); //koko
buttongp.add(rdbtnNewRadioButton); //koko
buttongp.add(rdbtnNewRadioButton_1); //koko
buttongp.add(rdbtnNewRadioButton_2); //koko
```

以上でとりあえず問題ないようです。

2				_		×
New label	I 🔘 New radio button	۲	New radio button	New	radio	button
	New radio button[false] New radio button[true] New radio button[false]					
	SwingActio	n	SwingAction	_1		

5. 表示名の変更	🛃 カラーサイン				-	×
内容を変更してください		サインパネル	○ 赤	○ 緑	• 🕈	
・タイトルを付けます	赤[true] 緑[false] 末[false]					
・ボタンの表示を変更します	音[laise] 赤[faise] 緑[faise]					
あわせてツールチップ	- 書[true]					
(注称補足有報)も設定しより						
ソースタブでもデザインタブでも						
変更可能で9か、テサインタンの プロパティだけでは設定できない		選扔	」」 マ状況を表	, 沅		

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

チェックした色がラジオボタンに反映されるように変更してください

ラジオボタンの色の指定処理は次のようにして実装できます ラジオボタンフィールド変数名.setForeground(Color.red);

🛃 カラーサイン				_	×
	サインパネル	● <mark>赤</mark>	○ 縁	○ 🕈	
赤[true] 緑[false] 青[false]					
	表示	2	リア		

【ヒント】 赤を指定するとき

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