Swing

[戻る] [前に] [次に]

Swingとは

Swing（スィング）は、JDK1.2 でサポートされた AWT よりも新しい GUIコンポーネントです。AWT に対して以下のような特徴があります。

- OSが変わっても見栄えが変わらない。
- AWTよりも重い。
- AWT での部品名に J をつけたものが多い。

AWT に対応する Swing の部品には次のようなものがあります。

カテゴリ	種類	AWT	Swing
ウィンドウ	フレーム	Frame	JFrame
	ダイアログ	Dialog	JDialog
部品	ボタン	Button	JButton
	トグルボタン	×	JToggleButton
	ラベル	Label	JLabel
	テキストフィールド	TextField	JTextField
	テキストエリア	TextArea	JTextArea
	チェックボックス	Checkbox	JCheckBox
	ラジオボタン	×	JRadioButton
	コンボボックス	Choice	JComboBox
	リスト	List	JList
	スクロールバー	Scrollbar	JScrollBar
	テーブル	×	JTable
	ツリー	×	JTree
	プログレスバー	×	JProgressBar
ペイン	パネル	Panel	JPanel
	スクロールペイン	ScrollPane	JScrollPane
	タブペイン	×	JTabbedPane
	キャンバス	Canvas	?
メニュー	メニューバー	MenuBar	JMenuBar
	ツールバー	×	JToolBar
	ステータスバー	×	※
	ポップアップメニュー	PopupMenu	JPopupMenu

フレーム（JFrame）

◆ サンプル

JFrameTest.java

◆ 説明

Swing では **フレーム** の作成に、Frame の代わりに **JFrame** を使用します。下記は、JFrame を表示するだけのシンプルなサンプルです。

JFrameTest.java

```
import java.awt.*;
import javax.swing.*;

class JFrameTest extends JFrame {
JFrameTest() {
getContentPane().setLayout(new FlowLayout());
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setTitle("JFrameTest");
setSize(200, 100);
setVisible(true);
}
public static void main(String [] args) {
new JFrameTest();
}
}
```

AWT の Frame サンプルと比べると下記のような違いがあります。

- javax.swing.* をインポートする。
- JFrame のサブクラスとして実装する。
- Button は JButton など、Swing 部品には J がつくものが多い。
- 部品配置やレイアウトは JFrame に直接ではなく、getContentPane() で取得したペインに対して行う。
- setDefaultCloseOperation()でウィンドウを閉じる処理を簡単に記述できる。

ボタン（JButton）

◆ サンプル

JButtonTest.java

◆ 説明

ボタン は **JButton** クラスを用います。

◆ 補足

- 押されたことを監視する → ActionListener
- ボタンにアクションコマンド名を登録する → setActionCommand()
- 押下時にアクションコマンド名を得る → e.getActionCommand()

トグルボタン（JToggleButton）

◆ サンプル

JToggleButtonTest.java

◆ 説明

トグルボタン は、1回ボタンを押すごとに、オン・オフの状態が切り替わるボタンです。**JToggleButton** クラスを用います。

◆ 補足

- オン・オフの状態変化を監視する → ChangeListener
- オンかオフか判断する → isSelected()

ラベル（JLabel）

◆ サンプル

JLabelTest.java

◆ 説明

ラベル には **Label** クラスを用います。

◆ 補足

- ラベルを変更する → setText()

テキストフィールド（JTextField）

◆ サンプル

JTextFieldTest.java

◆ 説明

テキストフィールド には **JTextField** クラスを用います。

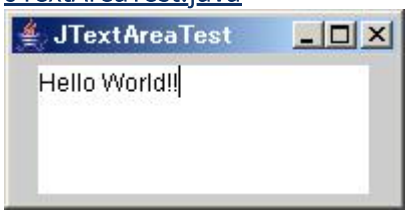
◆ 補足

- テキストを読み書きする → getText(), setText()

テキストエリア（JTextArea）

◆ サンプル

JTextAreaTest.java



◆ 説明

テキストエリア には **JTextArea** クラスを用います。

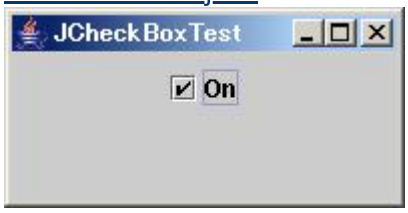
◆ 補足

- テキストを読み書きする → getText(), setText()

チェックボックス（JCheckBox）

◆ サンプル

JCheckBoxTest.java



◆ 説明

チェックボックス には **JCheckBox** クラスを用います。ActionListener でもチェックボックスをクリックされたことを監視可能ですが、チェックボックスのオン・オフ変化を確実に監視するには、ChangeListener（javax.swing.event）を用いるのが無難です。ButtonGroup によるグルーピングも可能です。

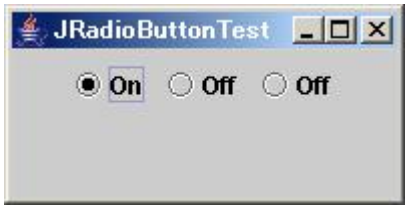
◆ 補足

- オン・オフの状態変化を監視する → ChangeListener
- オンかオフか調べる → isSelected()
- オン・オフを切り替える → setSelected()

ラジオボタン（JRadioButton）

◆ サンプル

JRadioButtonTest.java



◆ 説明

ラジオボタン には **JRadioButton** クラスを用います。ButtonGroup でグループ化することにより、グループ内のオン状態のボタンが常にひとつになるように制御することができます。

◆ 補足

- オン・オフの状態変化を監視する → ChangeListener
- オンかオフか調べる → isSelected()
- オン・オフを切り替える → setSelected()

コンボボックス（JComboBox）

◆ サンプル

JComboBoxTest.java

◆ 説明

コンボボックス には **JComboBox** クラスを用います。キーボードで選択操作する場合、キー移動する度に actionPerformed() が呼ばれるので注意してください。

◆ 補足

- アイテムを追加する → addItem()
- アイテムの個数を得る → getItemCount()
- 選択変更を監視する → ActionListener, ItemListener
- 選択中のインデックスを得る → getSelectedIndex()
- 選択中のアイテムを得る → getSelectedItem()
- 編集可能にする → setEditable()

リスト（JList）

◆ サンプル

JListTest.java

◆ 説明

リスト には **JList** クラスを用います。

◆ 補足

- 選択の変更を監視する → ListSelectionListener
- スクロールバーをつける → JScrollPane
- 選択されているインデックスを得る → getSelectedIndex()
- 選択されている値を得る → getSelectedValue()

スクロールバー（JScrollBar）

◆ サンプル

JScrollBarTest.java

◆ 説明

スクロールバー には **JScrollBar** クラスを用います。テキストエリアや、テーブルなどの部品にスクロールバーをつけるには、JScrollBar ではなく JScrollPane を用います。

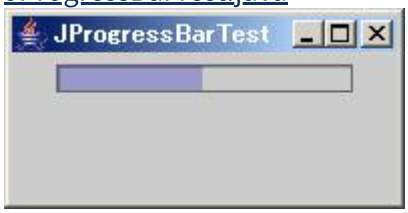
◆ 補足

- スクロールを監視する → AdjustmentListener
- 現在の値を得る → getValue()

プログレスバー（JProgressBar）

◆ サンプル

JProgressBarTest.java



◆ 説明

プログレスバー には **JProgressBar** クラスを用います。時間のかかる処理の経過状況などを表示するのに用いられます。

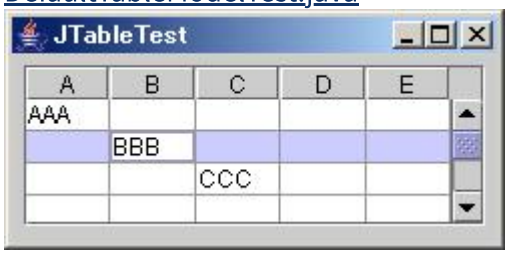
◆ 補足

- 値を設定する → setValue()

テーブル（JTable）

◆ サンプル

JTableTest.java
DefaultTableModelTest.java



◆ 説明

テーブル には **JTable** クラスを用います。通常のテーブルは、セルの編集は可能ですが、行や列を追加・削除することができません。行や列を追加削除するには TableModel を用います。基本的な TableModel として、DefaultTableModel を用いた例をサンプルに示しています。

◆ 補足

- 指定したセルに値を設定する → setValueAt()
- 指定したセルの値を読み出す → getValueAt()
- 列数を得る → getColumnCount()
- 行数を得る → getRowCount()
- スクロールバーをつける → JScrollPane
- 列や行を追加・削除する → DefaultTableModel

ツリー（JTree）

◆ サンプル

JTreeTest.java

◆ 説明

ツリー を作成するには **JTree** クラスを用います。

◆ 補足

- ツリーの各ノードを作成する → DefaultMutableTreeNode
- ノードに子ノードを追加する → add()
- ノードの選択状態を監視する → TreeSelectionListener
- 選択されているノードを得る → getLastSelectedPathComponent()

メニューバー（JMenuBar）

◆ サンプル

JMenuBarTest.java

◆ 説明

メニュー を作成するには、主に下記の 4つのクラスを用います。

クラス	説明
JMenuBar	**メニューバー**。
JMenu	**メニュー**。子アイテムを持つメニュー項目。
JMenuItem	**メニューアイテム**。子アイテムを持たないメニュー項目。
JCheckBoxMenuItem	チェックタイプのメニューアイテム。

MenuBar の場合は Menu と CheckBoxMenuItem にアクションリスナーを追加していましたが、JMenuBar の場合はそれぞれのメニューアイテムにアクションリスナーを登録します。

◆ 補足

- 実行を監視する → ActionListener
- ニーモニックを設定する → setMnemonic()

ツールバー（JToolBar）

◆ サンプル

JToolBarTest.java

◆ 説明

ツールバー を作成するには **JToolBar** クラスを用います。ツールバーの上には、JButton で作成したボタンなどを配置することができます。

ステータスバー（JStatusBar）

◆ サンプル

JStatusBarTest.java

◆ 説明

Swing ではステータスバー専用の部品はサポートされていません。代わりに、JLabel で生成したラベルを BorderLayout.SOUTH に貼り付けることで、ステータスバーを模倣することができます。

スクロールペイン（JScrollPane）

◆ サンプル

JScrollPaneTest.java

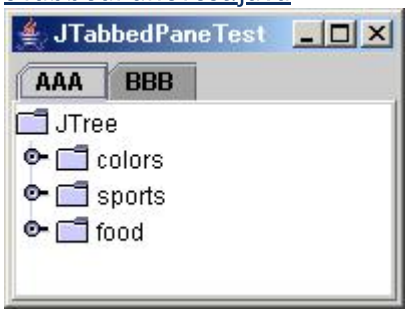
◆ 説明

JTextArea、JList、JTree などの部品に **スクロールバー** をつけるには、**スクロールペイン**（**JScrollPane**）の中にこれらの部品を貼り付けます。

タブペイン（JTabbedPane）

◆ サンプル

JTabbedPaneTest.java



◆ 説明

画面をタブ付きのペインで切り替えるには、**タブペイン**（**JTabbedPane**）を用います。

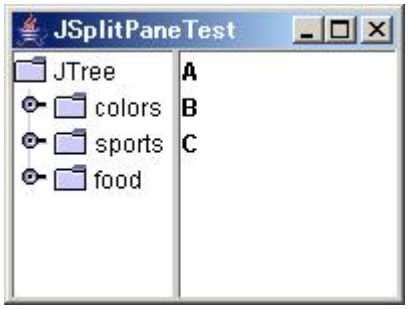
◆ 補足

- タブを加える → add()
- タブの切り替わりを監視する → ChangeListener
- 現在選択中のタブを得る → getSelectedComponent()

画面を分割する（JSplitPane）

◆ サンプル

JSplitPaneTest.java



◆ 説明

画面を左右や上下に分割するには、**スプリットペイン**（**JSplitPane**）を用います。

◆ 補足

- 左側のペインに部品を割り当てる → setLeftComponent()
- 右側のペインに部品を割り当てる → setRightComponent()
- 境界線の太さを指定する → setDividerSize()

ダイアログ（JDialog）

◆ サンプル

JDialogTest.java

◆ 説明

ダイアログ を表示するには **JDialog** を用います。通常、JDialog のサブクラスを作成し、そのサブクラスのインスタンスを生成してダイアログを表示します。

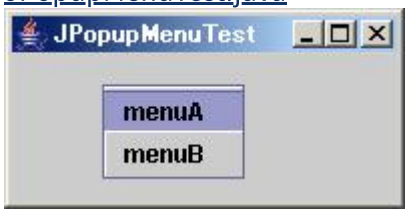
◆ 補足

- ダイアログを表示する → show()
- ダイアログを非表示にする → hide()

ポップアップメニュー（JPopupMenu）

◆ サンプル

JPopupMenuTest.java



◆ 説明

ポップアップメニュー を表示するには **JPopupMenu** を用います。

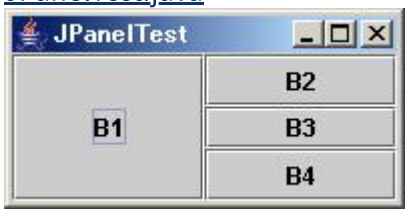
◆ 補足

- ポップアップメニューを表示する → show()
- ポップアップを表示するトリガーとなるイベントか調べる → isPopupTrigger()

パネル（JPanel）

◆ サンプル

JPanelTest.java



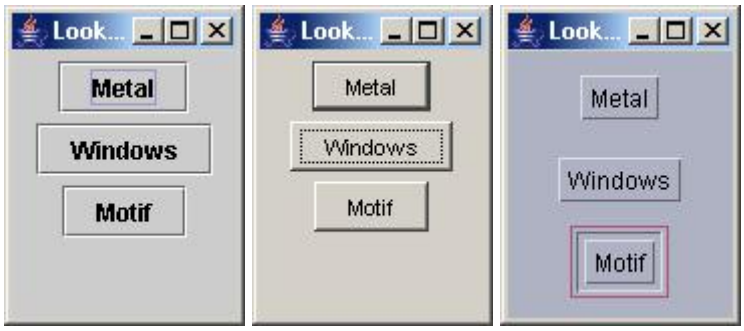
◆ 説明

パネル（JPanel）は、より柔軟なレイアウトを実現する際に用いられます。フレームやダイアログ上に複数のパネルを配置し、パネル内に部品を配置することにより、通常の XxxxLayout では表現できないレイアウトを実現します。「パネルによるレイアウト」を参照してください。

ルック＆フィールの変更

◆ サンプル

LookAndFeelTest.java



Metal **Windows** **Motif**

◆ 説明

Swing では、GUI のルック＆フィールを動的に変更することができます。現状では、Metalモード、Windowsモード、Motifモードがサポートされています。

---

http://www.tohoho-web.com/java/swing.htm
初版：2004年6月26日、最終更新：2004年6月26日
[戻る] [前に] [次に]