

パッケージ javax.swing

すべてのプラットフォーム上で可能なかぎり同じように機能する「軽量」(Java共通言語)コンポーネントのセットを提供します。

参照: [説明](#)

インターフェースのサマリー

インターフェース	説明
Action	Actionインターフェースは、同じ機能が複数のコントロールによってアクセスされる場合に、ActionListenerインターフェースに対する便利な拡張機能を提供します。
BoundedRangeModel	SliderやProgressBarなどのコンポーネントが使用するデータ・モデルを定義します。
ButtonModel	ボタンの状態モデルです。
CellEditor	すべての汎用エディタが実装可能なメソッドを定義します。
ComboBoxEditor	JComboBoxコンポーネントに使われるエディタ・コンポーネントです。
ComboBoxModel<E>	コンボボックスのデータ・モデルです。
DesktopManager	DesktopManagerオブジェクトは、JDesktopPaneオブジェクトによって所有されます。
Icon	通常はコンポーネントを装飾するために使う、固定サイズの小型イメージです。
JComboBox.KeySelectionManager	KeySelectionManagerを定義するインターフェースです。
ListCellRenderer<E>	JList内のセルをペイントする「ゴム印」として使用できるコンポーネントを識別します。
ListModel<E>	JListなどのコンポーネントがリスト内の各セルの値およびリストの長さを取得するために使用するメソッドを定義します。
ListSelectionModel	固定のインデックスを持つ値のリストを表示するコンポーネントの、現在の選択状態を表します。
MenuItem	メニューに配置できるコンポーネントはすべて、このインターフェースを実装する必要があります。
MutableComboBoxModel<E>	ComboBoxModelの可変バージョンです。
Painter<T>	ペイント委譲。

Renderer	値の描画(表示)を行うオブジェクトの必要条件を定義します。
RootPaneContainer	このインターフェースは、JRootPaneの子を1つ持つJDialog、JFrame、JWindow、JApplet、JInternalFrameなどのコンポーネントによって実装されます。
Scrollable	JScrollPaneなどのスクロール・コンテナに情報を提供するインターフェースです。
ScrollPaneConstants	JScrollPaneコンポーネントとともに使われる定数です。
SingleSelectionModel	インデックス付き選択範囲を1つだけサポートするモデルです。
SpinnerModel	要素を無限に作成できるオブジェクト値シーケンスを表すモデルです。
SwingConstants	一般的には画面上でコンポーネントの配置および方向指定を行うために使用する定数のコレクションです。
UIDefaults.ActiveValue	このクラスを使うと、getXXX(key)メソッドで検索されるたびに構築されるデフォルトのテーブルにエントリを格納できます。
UIDefaults.LazyValue	このクラスを使うと、getXXX(key)メソッドではじめて検索されるまで構築されないデフォルトのテーブルにエントリを格納できます。
WindowConstants	ウィンドウのクローズ操作を制御する定数です。

クラスのサマリー

クラス	説明
AbstractAction	このクラスは、JFCのActionインターフェースのデフォルト実装を提供します。
AbstractButton	ボタンおよびメニュー項目の共通動作を定義します。
AbstractCellEditor	CellEditorsの基底クラスであり、getCellEditorValue()を除くCellEditorインターフェースのメソッドのデフォルトの実装を提供します。
AbstractListModel<E>	Listにその内容を提供するデータ・モデルの抽象定義です。
AbstractSpinnerModel	このクラスでは、ほとんどの具象SpinnerModel実装に適するSpinnerModelインターフェースのChangeListener部分を提供します。
ActionMap	ActionMapは、キーまたはAction名と呼ばれるObjectからActionへのマップを提供します。
BorderFactory	標準的なBorderオブジェクトを提供するためのファクトリ・クラスです。
Box	BoxLayoutオブジェクトをレイアウト・マネージャとして使う軽量なコンテナです。

Box.Filler	レイアウトに含まれるがビューを持たない軽量コンポーネントの実装です。
BoxLayout	複数のコンポーネントを、垂直方向にも水平方向にも配置できるようにする、レイアウト・マネージャです。
ButtonGroup	複数のボタンのセットに多重排他スコープを設定するために使用します。
CellRendererPane	このクラスは、セル・レンダラとそれを使うコンポーネントの間に挿入されます。
ComponentInputMap	ComponentInputMapは、特定のJComponentに関連付けられているInputMapです。
DebugGraphics	グラフィックスのデバッグをサポートするGraphicsのサブクラスです。
DefaultBoundedRangeModel	BoundedRangeModelのジェネリック実装です。
DefaultButtonModel	Buttonコンポーネントのデータ・モデルのデフォルト実装です。
DefaultCellEditor	テーブルとツリー・セルのデフォルト・エディタです。
DefaultComboBoxModel<E>	コンボボックスのデフォルト・モデルです。
DefaultDesktopManager	DesktopManagerの実装です。
DefaultFocusManager	このクラスは1.4 focus APIでは現在使用されません。
DefaultListCellRenderer	リスト内の項目をレンダリングします。
DefaultListCellRenderer.UIResource	UIResourceを実装するDefaultListCellRendererのサブクラスです。
DefaultListModel<E>	このクラスはjava.util.Vector APIを柔軟に実装します。1.1.xのjava.util.Vectorを実装しますが、コレクション・クラスはサポートせず、変更発生時にはListDataListenerに通知します。
DefaultListSelectionModel	リスト選択のデフォルト・データ・モデルです。
DefaultRowSorter<M,I>	グリッド・ベースのデータ・モデルのソートやフィルタの適用を行うRowSorterの実装です。
DefaultRowSorter.ModelWrapper<M,I>	DefaultRowSorter.ModelWrapperは、DefaultRowSorterでソートされるデータを提供します。
DefaultSingleSelectionModel	SingleSelectionModelのジェネリック実装です。

FocusManager

このクラスは1.4 focus APIでは現在使用されていません。

GrayFilter

イメージをグレー・スケールに変換し、イメージ内のピクセルの明度を上げることによって、イメージを「無効にする」イメージ・フィルタです。

GridLayout

GridLayoutは、Container内に配置するためにコンポーネントを階層的にグループ化するLayoutManagerです。

ImageIcon

アイコンをイメージからペイントするIconインターフェースの実装です。

InputMap

InputMapは、入力イベント(現在はKeyStrokeのみ使用)とObjectの間のバインディングを提供します。

InputVerifier

このクラスの目的は、クライアントがテキスト・フィールドとGUIを使って、スムーズなフォーカス・ナビゲーションを行えるようにサポートすることです。

InternalFrameFocusTraversalPolicy

JInternalFrameの初期Componentを判定するためのアルゴリズムを任意に提供できるFocusTraversalPolicyです。

JApplet

JFC/Swingコンポーネント・アーキテクチャのサポートを追加するjava.applet.Appletの拡張バージョン。

JButton

「プッシュ」ボタンの実装です。

JCheckBox

チェックボックス(選択または選択解除が可能で、その状態をユーザーに表示する項目)の実装です。

JCheckBoxMenuItem

選択または選択解除できるメニュー項目です。

JColorChooser

JColorChooserは、ユーザーが色を操作したり、選択したりできるように設計されたコントロールのペインを提供します。

JComboBox<E>

ボタンまたは編集可能フィールドとドロップ・ダウン・リストを組み合わせたコンポーネントです。

JComponent

トップ・レベルのコンテナを除くすべてのSwingコンポーネントの基底クラスです。

JDesktopPane

マルチドキュメント・インターフェースまたは仮想デスクトップを生成するコンテナです。

JDialog

ダイアログ・ウィンドウを作成するためのメイン・クラスです。

JEditorPane

さまざまな種類のコンテンツを編集するためのテキスト・コンポーネントです。

JFileChooser

JFileChooserは、ユーザーがファイルを選択するための単純なメカニズムを提供します。

JFormattedTextField

JFormattedTextFieldは JTextField を拡張して、任意の値をフォーマットしたり、ユーザーがテキストを編集したあとに特定のオブジェクトを取得したりするためのサポートを追加します。

JFormattedTextField.AbstractFormatter

AbstractFormatter のインスタンスは、Object から String および String から Object への変換を処理するために JFormattedTextField で使用されます。

JFormattedTextField.AbstractFormatterFactory

AbstractFormatterFactory のインスタンスは JFormattedTextField で使用され、値のフォーマットに使用される AbstractFormatter のインスタンスを取得します。

JFrame

JFC/Swing コンポーネント・アーキテクチャのサポートを追加する java.awt.Frame の拡張バージョン。

JInternalFrame

ネイティブなフレームに、ドラッグ、クローズ、アイコン化、サイズ変更、タイトル表示、およびメニュー・バーのサポートなど、多くの機能を提供する軽量オブジェクトです。

JInternalFrame.JDesktopIcon

このコンポーネントは、JInternalFrame のアイコン化されたバージョンを表します。

JLabel

短いテキスト文字列やイメージの表示領域です。両方と一緒に表示することもできます。

JLayer<V extends Component>

JLayer は、さまざまな高度なペイント効果を実装したり、そのボーダー内で生成されるすべての AWTEvent の通知を受信したりするために使用できる、 Swing コンポーネントのための汎用のデコレータです。

JLayeredPane

JLayeredPane は JFC/Swing コンテナに深さを追加し、コンポーネントが必要に応じて互いにオーバーラップできるようにします。

JList<E>

オブジェクトのリストを表示し、ユーザーが 1 つまたは複数のオブジェクトを選択できるようにするコンポーネントです。

JList.DropLocation

TransferHandler.DropLocation のサブクラスであり、JList のドロップ位置を表します。

JMenu

メニュー(ユーザーが JMenuBar の項目を選択したときに表示される、 JMenuItem を含む ポップアップ・ウィンドウ) の実装です。

JMenuBar

メニュー・バーの実装です。

JMenuItem

メニューの項目の実装です。

JOptionPane

JOptionPaneは、ユーザーに値の入力を求めたり、何らかの情報を通知したりするための標準のダイアログ・ボックスを容易にポップアップできるようにします。

JPanel

JPanelは、汎用の軽量コンテナです。

JPasswordField

JPasswordFieldは、1行のテキストの編集を可能にする軽量コンポーネントです。何かが入力されたことはビューに示されますが、元の文字は表示されません。

JPopupMenu

ポップアップ・メニュー(一連の選択肢をポップアップおよび表示する小さなウィンドウ)の実装です。

JPopupMenu.Separator

ポップアップ・メニュー固有のセパレータです。

JProgressBar

一部のタスクの進捗状況を視覚的に表示するコンポーネントです。

JRadioButton

ラジオ・ボタン(選択または選択解除が可能で、その状態をユーザーに表示する項目)の実装です。

JRadioButtonMenuItem

ラジオ・ボタン・メニュー項目の実装です。

JRootPane

JFrame、JDialog、JWindow、JApplet、およびJInternalFrameによって内部的に使用される軽量コンテナ。

JScrollBar

スクロール・バーの実装です。

JScrollPane

軽量コンポーネントのスクロール可能なビューを提供します。

JSeparator

JSeparatorは、区切り線を実装するための汎用的なコンポーネントを提供します。もっとも一般的には、メニュー項目を論理的なグループに分割するための項目間の区切り線として使用されます。

JSlider

ユーザーが指定された区間内でノブをスライドすることによりグラフィカルに値を選択できるようにするコンポーネントです。

JSpinner

ユーザーが、順序付けられたシーケンスから数またはオブジェクトを選択できるようにする單一行の入力フィールドです。

JSpinner.DateEditor

モデルがSpinnerDateModelであるJSpinnerのエディタ。

JSpinner.DefaultEditor

JFormattedTextFieldを使用してモデルの現在の値の読み取り専用ビューを表示する、より特殊なエディタの単純な基底クラス。

JSpinner.ListEditor

モデルがSpinnerListModelであるJSpinnerのエディタ。

JSpinner.NumberEditor	モデルがSpinnerNumberModelであるJSpinnerのエディタ。
JSplitPane	JSplitPaneは、2つ(2つだけ)のComponentを分割するために使用されます。
JTabbedPane	指定されたタイトルやアイコンを持つタブをクリックすることにより、ユーザーがコンポーネントのグループを切り替えられるようにするコンポーネントです。
JTable	JTableは、セルの通常の2次元テーブルを表示および編集するために使用されます。
JTable.DropLocation	TransferHandler.DropLocationのサブクラスであり、JTableのドロップ位置を表します。
JTextArea	JTextAreaは、プレーン・テキストを表示する複数行の領域です。
JTextField	JTextFieldは、1行のテキストの編集を可能にする軽量コンポーネントです。
JTextPane	グラフィックで表現される属性でマークアップできるテキスト・コンポーネントです。
JToggleButton	2つの状態を持つボタンの実装です。
JToggleButton.ToggleButtonModel	ToggleButtonモデルです
JToolBar	JToolBarは、一般的に使用されるActionまたはコントロールを表示するのに役立つコンポーネントを提供します。
JToolBar.Separator	ツールバー独自のセパレータです。
JToolTip	Componentの「ヒント」を表示するために使います。
JTree	階層データのセットをアウトラインとして表示するコントロール。
JTree.DropLocation	TransferHandler.DropLocationのサブクラスであり、JTreeのドロップ位置を表します。
JTree.DynamicUtilTreeNode	DynamicUtilTreeNodeは、ベクター、ハッシュ・テーブル、配列、および文字列をラップし、必要に応じて適切な子ツリー・ノードを作成できます。
JTree.EmptySelectionModel	EmptySelectionModelは、何の選択も許可しないTreeSelectionModelです。
JViewport	ベースとなる情報をるために使用する「ビュー・ポート(窓)」です。
JWindow	JWindowは、ユーザーのデスクトップ上のどこにでも表示できるコンテナです。

KeyStroke	KeyStrokeは、キーボードまたは同等の入力デバイスのキー・アクションを表します。
LayoutFocusTraversalPolicy	サイズ、位置、方向に基づいてComponentをソートするSortingFocusTraversalPolicyです。
LayoutStyle	LayoutStyleは、コンポーネントの配置に関する情報を提供します。
LookAndFeel	LookAndFeelは、その名前が示す通り、ルック・アンド・フィールをカプセル化します。
MenuSelectionManager	MenuSelectionManagerは、メニュー階層での選択を所有します。
OverlayLayout	コンポーネントを互いの上に重なり合うように配置するレイアウト・マネージャです。
Popup	Popupは、ユーザーにComponentを表示するために使用されます。通常、Componentは、特定の包含関係の階層に存在するほかのすべてのComponentの上に配置されます。
PopupFactory	PopupFactoryは、その名前が示す通り、Popupのインスタンスを取得するために使用します。
ProgressMonitor	処理の進捗を監視するクラスです。
ProgressMonitorInputStream	InputStreamからの読み込みの進捗を監視します。
RepaintManager	このクラスは再ペイント要求を管理し、コンポーネント・ツリーのメンバーに対する複数の要求を1回の再ペイントにまとめることなどにより、再ペイントの回数を最小限に抑えます。
RowFilter<M,I>	RowFilterは、エントリをモデルからフィルタ・リングして、ビューに表示されないようにするために使用します。
RowFilter.Entry<M,I>	EntryオブジェクトがRowFilterのインスタンスに渡されると、フィルタはエントリのデータ値を取得し、エントリを表示するかどうかを判断できます。
RowSorter<M>	RowSorterは、ソートとフィルタの基盤を提供します。
RowSorter.SortKey	SortKeyは、特定の列のソート順序を記述します。
ScrollPaneLayout	JScrollPaneが使用するレイアウト・マネージャです。
ScrollPaneLayout.UIResource	ScrollPaneLayoutのUIリソース・バージョンです。
SizeRequirements	レイアウト・マネージャの便宜のために、コンポーネントのサイズと位置についての情報を計算します。

SizeSequence	SizeSequenceオブジェクトは、サイズおよび対応する位置の、順序付けされたリストを効率的に維持します。
SortingFocusTraversalPolicy	指定されたComparatorに基づいてフォーカス・トラバーサル・サイクルのComponentをソートすることによって、トラバーサル順序を決定するFocusTraversalPolicyです。
SpinnerDateModel	DateのシーケンスのSpinnerModelです。
SpinnerListModel	配列またはListによって値を定義するSpinnerModelの単純な実装です。
SpinnerNumberModel	数値のシーケンスを表すSpinnerModelです。
Spring	Springクラスのインスタンスは、その動作を特徴付ける3つのプロパティ(値はminimum、preferred、およびmaximum)を保持します。
SpringLayout	SpringLayoutは、関連付けられているコンテナの子を一連の制約に従ってレイアウトします。
SpringLayout.Constraints	Constraintsオブジェクトは、SpringLayoutが制御するコンテナ内のコンポーネントのサイズ変更方法および位置変更方法を決定する制約を保持します。
SwingUtilities	Swingのユーティリティ・メソッドのコレクションです。
SwingWorker<T,V>	GUIとやりとりする時間のかかるタスクを、バックグラウンド・スレッドで実行するためのabstractクラスです。
Timer	指定された間隔で、1つ以上のActionEventをトリガーリします。
ToolTipManager	システム内のすべてのToolTipsを管理します。
TransferHandler	このクラスは、Swingコンポーネントに対するTransferableの転送処理に使用します。
TransferHandler.DropLocation	ドロップされたデータを挿入する位置を表します。
TransferHandler.TransferSupport	このクラスには、クリップボードによる転送またはドラッグ・アンド・ドロップによる転送に関する詳細情報をすべてカプセル化する機能と、ドラッグ・アンド・ドロップ操作をカスタマイズする機能があります。
UIDefaults	Swingコンポーネントのデフォルト値のテーブルです。
UIDefaults.LazyInputMap	LazyInputMapは、createValueメソッドでInputMapを作成します。

UIDefaults.ProxyLazyValue

このクラスは、生成されるインスタンスのClassのコードを遅延させるLazyValueの実装を提供します。

UIManager

UIManagerは、現在のルック・アンド・フィール、使用可能なルック・アンド・フィールのセット、ルック・アンド・フィールの変更時に通知を受けるPropertyChangeListeners、ルック・アンド・フィールのデフォルト値、およびさまざまなデフォルト値を取得するために使用する簡易メソッドを管理します。

UIManager.LookAndFeelInfo

メニューの設定またはアプリケーションの初期設定のための、インストール済みLookAndFeelに関する概要を説明します。

ViewportLayout

JViewportのデフォルトのレイアウト・マネージャです。

列挙型のサマリー**列挙型****説明****DropMode**

メソッドの特定に使用するドロップ・モードです。ドラッグ・アンド・ドロップのときに、コンポーネントは、このメソッドを使ってドロップの位置を追跡して示します。

GroupLayout.Alignment

ParallelGroupがその子を配置するときに選択できる方法の列挙です。

JTable.PrintMode

JTableの出力に使用する出力モードです。

LayoutStyle.ComponentPlacement

ComponentPlacementは、2つのコンポーネントを相互に関連付けて配置する場合に使用できる方法の列挙です。

RowFilter.ComparisonType

一部のデフォルトのRowFilterでサポートされる、値比較に使用可能な値の列挙です。

SortOrder

SortOrderは、可能なソート順序の列挙です。

SwingWorker.StateValue

stateバウンド・プロパティの値です。

例外のサマリー**例外****説明****UnsupportedLookAndFeelException**

要求されたルック・アンド・フィールの管理クラスがユーザーのシステム上に存在しないことを示す例外です。

パッケージjavax.swingの説明

すべてのプラットフォーム上で可能なかぎり同じように機能する「軽量」(Java共通言語)コンポーネントのセットを提供します。これらのコンポーネントの使用方法に関するプログラマ向けガイドについては、『The Java Tutorial』の「Creating a GUI with JFC/Swing」を参照してください。その他のリソースについては、「関連ドキュメント」を参照してください。

Swingのスレッド・ポリシー

通常、Swingはスレッドに対して安全ではありません。すべてのSwingコンポーネントと関連クラスには、特に説明がないかぎり、イベント・ディスパッチ・スレッド上でアクセスしてください。

通常のSwingアプリケーションは、ユーザーのジェスチャによって生成されたイベントに応じて、処理を実行します。たとえば、ユーザーが JButton をクリックすると、 JButton に追加されたすべての ActionListener に通知が送信されます。ユーザーのジェスチャによって生成されたすべてのイベントはイベント・ディスパッチ・スレッド上でディスパッチされるので、開発者が制限の影響を受けることは、ほとんどありません。

ただし、Swingアプリケーションの構築時と表示時には影響があります。イベント・ディスパッチ・スレッド上では、アプリケーションの main メソッドの呼び出しや Applet 内のメソッドの呼び出しは行われません。このため、アプリケーションやアプレットの構築時および表示時には、イベント・ディスパッチ・スレッドに制御を渡していることを確認してください。制御を渡し、Swing の処理を開始するためには、 invokeLater を使用することをお薦めします。 invokeLater メソッドは、 Runnable がイベント・ディスパッチ・スレッド上で処理されるようにスケジュールします。次に、Swing アプリケーションに制御を渡し、Swing アプリケーションを起動する例を 2 つ示します。どちらの方法も適切です。

```
import javax.swing.SwingUtilities;

public class MyApp implements Runnable {
    public void run() {
        // Invoked on the event dispatching thread.
        // Construct and show GUI.
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new MyApp());
    }
}
```

または:

```
import javax.swing.SwingUtilities;

public class MyApp {
    MyApp(String[] args) {
        // Invoked on the event dispatching thread.
        // Do any initialization here.
    }

    public void show() {
        // Show the UI.
    }

    public static void main(final String[] args) {
        // Schedule a job for the event-dispatching thread:
        // creating and showing this application's GUI.
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new MyApp(args).show();
            }
        });
    }
}
```

この制約は、Swing コンポーネントに接続されたモデルにも適用されます。たとえば、TableModel が JTable に接続されている場合、TableModel の変更は必ずイベント・ディスパッチ・スレッド上で行うようにしてください。別のスレッド上でモデルを変更した場合、例外が発生し、画面の表示が崩れる可能性があります。

すべてのイベントはイベント・ディスパッチ・スレッド上で配信されるので、イベント処理時に注意してください。特に、イベント・ディスパッチ・スレッド上で、ネットワーク入出力や計算量の多い処理など、

時間のかかるタスクを実行すると、その他のイベントのディスパッチがブロックされます。イベント・ディスパッチ・スレッドがブロックされている間、アプリケーションはユーザーの入力に一切反応しなくなります。Swingの使用時にこのような処理を実行する場合は、 [SwingWorker](#)の説明に従うことをお勧めします。

このトピックの詳細は、『The Swing Tutorial』(特に「Concurrency in Swing」のセクション)を参照してください。

関連項目

概要、チュートリアル、例、ガイド、その他のマニュアルについては、次を参照してください。

- ・ 「The Swing Connection」
- ・ 「The Java Tutorial」
- ・ Java Developer ConnectionSMの「Online Training」
- ・ 「Java Foundation Classes (JFC)」ホーム・ページ

概要 パッケージ クラス 使用 階層ツリー 非推奨 索引 ヘルプ

Java(tm) Platform
Standard Edition 8

前のパッケージ 次のパッケージ フレーム フレームなし すべてのクラス

[バグまたは機能を送信](#)

詳細なAPIリファレンスおよび開発者ドキュメントについては、Java SEのドキュメントを参照してください。そのドキュメントには、概念的な概要、用語の定義、回避方法、有効なコード例などの、開発者を対象にしたより詳細な説明が含まれています。

Copyright © 1993, 2018, Oracle and/or its affiliates. All rights reserved. Use is subject to license terms. Documentation Redistribution Policyも参照してください。Modify Cookieの設定. Modify Ad Choices.