

# Java GUI

## プログラミング

日向俊二●著

Swing を使った  
今どきのアプリ開発



### ■ サンプルファイルのダウンロードについて

本書掲載のサンプルファイルは、一部を除いてインターネット上のダウンロードサービスからダウンロードすることができます。詳しい手順については、本書の巻末にある袋とじの内容をご覧ください。

なお、ダウンロードサービスのご利用にはユーザー登録と袋とじ内に記されている番号が必要です。そのため、本書を中古書店から購入されたり、他者から貸与、譲渡された場合にはサービスをご利用いただけないことがあります。あらかじめご承知おきください。

- 本書の内容についてのご意見、ご質問は、お名前、ご連絡先を明記のうえ、小社出版部宛文書（郵送またはE-mail）でお送りください。
- 電話によるお問い合わせはお受けできません。
- 本書の解説範囲を越える内容のご質問や、本書の内容と無関係なご質問にはお答えできません。
- 匿名のフリーメールアドレスからのお問い合わせには返信しかねます。

本書で取り上げられているシステム名／製品名は、一般に開発各社の登録商標／商品名です。本書では、™ および® マークは明記していません。本書に掲載されている団体／商品に対して、その商標権を侵害する意図は一切ありません。本書で紹介している URL や各サイトの内容は変更される場合があります。

# はじめに

---

Java はプログラミング言語の中では比較的歴史のある言語で、さまざまな用途に使われています。

Java で GUI アプリを作成する方法はいくつかありますが、Java の SDK に含まれている Swing というコンポーネントを使って、Windows、Linux、macOS で実行できる GUI アプリを作成することができます。本書では主に Swing コンポーネントを使った Java の GUI プログラミングについて解説します。

かつては Swing コンポーネントを使った Java プログラムは、実行時の動作が遅いといわれたことがありましたが、さまざまな面で改良や改善が行われて、現在では実用的な速度が実現されています。そして、JavaFX が Oracle JDK に含まれずに独立したライブラリとなった現在では、Swing コンポーネントを使った GUI プログラミングが Java では GUI プログラミングの標準といってよいでしょう。

本書では、最近の Java のバージョン（Oracle 用語ではリリース）で Swing を使って GUI アプリを作成する方法を、簡潔なサンプルプログラムを使って説明します。Java では、同じ目的であっても GUI アプリを作成する方法はいくつかあるので、利用可能ないくつかの異なるアプローチも示します。

本書を活用して Java の GUI プログラミングを楽しんでください。

2020 年初夏 著者しるす

## ■ 本書の表記

- xxx 斜体文字は、そこに具体的な文字や数値が入ることを表します。
- [ ] 書式の説明で [ ] で囲まれている場合は、その項目を省略可能であることを示します。
- > Windows の場合のコマンドプロンプトを表します。
- \$ Linux など UNIX 系 OS の場合のプロンプトを表します。



補足説明や知っておくと良い事柄です。

## ■ ご注意

- 本書の内容は本書執筆時の状態で記述しています。執筆時の Java の最新のリリース（バージョン）は 14 ですが、本書は 2023 年までの長期サポート（Long-Term-Support : LTS）リリースである Java 11 以降に対応するように配慮してあります。また掲載プログラムのほとんどは、Java 8 以降のどの環境でもコンパイルして実行できます。なお、将来、Java のバージョンが変わるなど、何らかの理由で記述と実際とが異なる結果となる可能性があります。
- 本書は Java や Java で使うことができるパッケージやクラスについてすべて完全に解説するものではありません。必要に応じて Java のドキュメントなどを参照してください。
- 本書のサンプルは、プログラミングを理解するために掲載するものです。実用的なアプリとして提供するものではありませんので、ユーザーのエラーへの対処やその他の面で省略してあるところがあります。
- 掲載しているプログラムコードの断片はそのままでは実行できません。本書掲載のプログラムコードを実行するときには、本書の内容をよく読んで理解してから実行してください。

## ■ 本書に関するお問い合わせについて

本書に関するお問い合わせは、sales@cutt.co.jp にメールでご連絡ください。

なお、お問い合わせは本書に記述されている範囲に限らせていただきます。特定の環境や特定の目的に対するお問い合わせ等にはお答えできませんので、あらかじめご了承ください。

お問い合わせ際には下記事項を明記していただきますようお願いいたします。

- 氏名
- 連絡先メールアドレス
- 書名
- 記載ページ
- お問い合わせ内容
- 実行環境

# 目次

はじめに .....	iii
<b>■ 第1章 GUIプログラミングの基礎 .....</b>	<b>1</b>
1.1 GUIプログラミングの基礎 .....	2
● GUIアプリの構造 / 2	
● イベントとその処理 / 3	
● イベントリスナー / 4	
1.2 JavaのGUIプログラミング .....	5
● GUIライブラリ / 5	
● Javaのバージョン / 6	
● バージョンの違いによる影響 / 7	
<b>■ 第2章 GUIアプリの基本 .....</b>	<b>9</b>
2.1 単純なウィンドウ .....	10
● ウィンドウの作成 / 10	
● プログラムのコンパイルと実行 / 13	
● JFrameの拡張 / 13	
● イベント処理 / 15	
● イベント処理のクラス / 18	
2.2 基本的なコンポーネント .....	20
● ボタン / 20	
● ラベルとの連携 / 24	
● グローバル変数を使う例 / 27	
<b>■ 第3章 ウィンドウ .....</b>	<b>31</b>
3.1 ウィンドウ .....	32
● JFrame / 32	
● JFrameの拡張 / 35	
● ウィンドウの位置とサイズ / 37	
● 装飾のないウィンドウ / 41	
3.2 ダイアログボックス .....	42
● JDialog / 42	
● JOptionPane / 49	
3.3 スプラッシュスクリーン .....	53
● SplashScreen / 53	
● JFrame スプラッシュ / 56	
3.4 パネル .....	58
● ウィンドウの背景色 / 58	
● コンテナとしてのパネル / 59	
<b>■ 第4章 ラベルとボタン .....</b>	<b>63</b>
4.1 ラベル .....	64
● 基本的なラベルの使い方 / 64	
● イメージの表示 / 68	
● スクロールできるラベル / 70	

4.2	ボタン .....	72
	●基本的なボタンの使い方 / 72	
	●イメージの表示 / 76	
■	<b>第5章 テキスト .....</b>	<b>79</b>
5.1	文字列の表示と編集 .....	80
	●テキスト表示のコンポーネント / 80	
	●共通メソッド / 81	
	●フォント / 81	
5.2	テキストの表示と編集 .....	85
	●テキストフィールド / 85	
	●テキストエリア / 87	
	●パスワードフィールド / 90	
5.3	テキスト用ペイン .....	94
	●テキストペイン / 94	
■	<b>第6章 レイアウト .....</b>	<b>99</b>
6.1	コンポーネントのレイアウト .....	100
	●コンテナとレイアウトマネージャ / 100	
	●レイアウトマネージャの特性 / 100	
6.2	レイアウトマネージャ .....	102
	●フローレイアウト / 102	
	●ボーダーレイアウト / 104	
	●ボックスレイアウト / 107	
	●グリッドレイアウト / 109	
	●グリッドバッグレイアウト / 111	
	●グループレイアウト / 115	
	●スプリングレイアウト / 118	
	●カードレイアウト / 121	
	●オーバーレイレイアウト / 124	
6.3	レイアウトの技法 .....	127
	●複数のパネルの利用 / 127	
	●分割パネル / 129	
	●レイアウトの変更 / 132	
■	<b>第7章 メニューとツールバー .....</b>	<b>135</b>
7.1	メニュー .....	136
	●メインメニュー / 136	
	●ポップアップメニュー / 140	
	●メニュー項目のアイコン / 144	
	●メニューの変更 / 145	
7.2	ツールバー .....	148
	●ツールバー / 148	
7.3	ステータスバー .....	151
	●ステータスバー / 151	

---

■ 第 8 章	マウスとキーボード .....	155
8.1	マウス .....	156
	●マウス操作の検出 / 156	
	●マウスボタンと修飾キーの検出 / 159	
	●マウスポインタの位置の取得 / 162	
	●マウスの移動の検出 / 165	
	●マウスポインタの形状変更 / 168	
	●マウスによる数値入力 / 170	
8.2	キーボード .....	174
	●キーの押し下げの検出 / 174	
	●矢印キーによる数値入力 / 177	
	●Enter キーによるフォーカス移動 / 177	
	●IME の制御 / 180	
■ 第 9 章	チェックボックスとラジオボタン .....	185
9.1	チェックボックス .....	186
	●チェックボックス / 186	
	●チェックボックスグループ / 189	
9.2	ラジオボタン .....	192
	●ラジオボタングループ / 192	
■ 第 10 章	リストボックスとコンボボックス .....	197
10.1	リストボックス .....	198
	●リストボックスの使い方 / 198	
	●イメージの表示 / 201	
10.2	コンボボックス .....	206
	●コンボボックスの使い方 / 206	
■ 第 11 章	テーブルとツリー .....	209
11.1	テーブル .....	210
	●JTable クラス / 210	
11.2	ツリー .....	221
	●ツリー / 221	
■ 第 12 章	スライダーとプログレスバー .....	231
12.1	スライダーとスクロールバー .....	232
	●スライダー / 232	
	●スクロールバー / 236	
12.2	プログレスバー .....	239
	●JProgressBar クラス / 239	



■ 第 13 章	さまざまなペイン .....	243
13.1	タブ付きペイン.....	244
	●タブ付きのページ / 244	
13.2	スクロールペイン .....	247
	●スクロール可能なコンポーネント / 247	
13.3	エディターペイン.....	250
	●Web ページの表示 / 250	
	●Web ページのソース表示 / 253	
■ 第 14 章	グラフィックス .....	257
14.1	イメージの表示.....	258
	●イメージを表示するには / 258	
	●ボタンへのイメージ表示 / 260	
14.2	グラフィックスの描画 .....	262
	●図形の描画 / 262	
	●色の作成 / 267	
	●色の選択ダイアログボックス / 270	
	●文字列の描画 / 272	
	●フォントの指定 / 274	
14.3	グラフ .....	277
	●棒グラフ / 277	
	●折れ線グラフ / 279	
	●円グラフ / 280	
■ 付 録	.....	283
付録 A	JDK のインストールと環境設定 .....	284
付録 B	トラブル対策.....	286
付録 C	参考リソース.....	292
索引	.....	293



# 第 1 章

---

## GUI プログラミングの 基礎



この章では GUI アプリのプログラミングの基礎について  
解説します。

## 1.1 GUI プログラミングの基礎

ここではウィンドウを使うアプリである GUI アプリのプログラムが動作する仕組みについて解説します。

### ■ GUI アプリの構造

コンソールウィンドウ（コマンドプロンプトウィンドウともいう）で数値を含むさまざまな文字列をキーボードから入力したり結果を画面に出力したりするようなプログラムを CUI（Character-based User Interface）プログラムといいます。

Java のプログラムをコンパイルする `javac` は代表的な CUI プログラムのひとつです。また、`java` にオプション `-version` を付けて実行したときにも CUI プログラムとして動作します。

次の例は `javac` で Java のプログラムをコンパイルする例と、「`java -version`」を実行してバージョン情報を出力する例です。

```
>javac SmplWnd.java
```

```
>java -version
```

```
openjdk version "11.0.5" 2019-10-15
```

```
OpenJDK Runtime Environment AdoptOpenJDK (build 11.0.5+10)
```

```
OpenJDK 64-Bit Server VM AdoptOpenJDK (build 11.0.5+10, mixed mode)
```

それに対して、ウィンドウを使うアプリを GUI（Graphical User Interface）アプリといいます。

GUI アプリは CUI のプログラムとは少し異なる考え方で作成します。その中心となるのが、アプリのイベントメッセージを処理するメインループです。

GUI アプリは、ウィンドウを作成する準備ができてウィンドウが表示されると、ウィンドウに送られるイベントメッセージを待ち続けます。イベントとは、マウスのクリックであったり、ユーザーからの入力であったり、あるいは OS や他のプログラムからの要求であったりしますが、いずれにしてもアプリのイベントメッセージを処理するメインループがイベントを待ち続けます（ボタンやリストボックスのようなコンポーネントも技術的には広い意味のウィンドウとしてとらえ、イベントを受け取ります）。

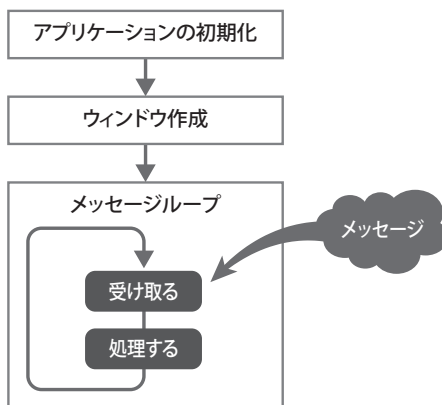


図 1.1 ● GUI アプリの構造

従って、アプリを終了するためのメッセージが送られない限り、プログラムが終了することはありません（CUI アプリは、無限ループがない限り、プログラムの最後のコードが実行されると終了します）。

CUI アプリと GUI アプリはこのように動作が異なりますが、プログラムを作成する立場からいえば、アプリがメッセージを受け取って、それに対応した動作をするようにコードを記述することで、GUI アプリを作成できます。

このような、イベントの発生を待ってそれを処理するという考え方でプログラミングすることを、イベント駆動型プログラミングともいいます。この考え方は、さまざまな GUI ベースのプログラミングでも共通して使われています。

## ■ イベントとその処理

アプリのメインウィンドウやコンポーネントが受け取るイベントには例えば次のようなものがあります（コンポーネントによって受け取るイベントの種類は異なります）。

表 1.1 ● GUI アプリのイベント

イベントの種類	イベントが発生する機会
ウィンドウ	ウィンドウが表示される、ウィンドウが閉じられようとしているなど。
アクション	ボタンがクリックされる、項目が選択されるなど。
チェンジ	コンポーネントが持つ値の状態が変わったなど。

イベントの種類	イベントが発生する機会
マウス	マウスのボタンが押されたりマウスが動かされたなど。
キー	キーボードのキーが押されたりキーから指が離されたなど。

GUI アプリでこれらのイベントの発生を待ってそれを処理するためのプログラムコードの詳細は、C 言語やアセンブリ言語で低レベルのプログラムを書く場合を除いて、通常は、プログラミング言語と実行時ライブラリの中に隠されています。そのため、プログラマーは、特定のイベントを処理するためのクラスやプロシージャ（イベントリスナーやイベントハンドラと呼ぶもの）に、そのイベントが発生したときに実行したいことをプログラムコードとして記述するだけで済みます（具体的な例は第2章以降で説明します）。

Java の GUI プログラミングでも、イベントの発生を待ってそれを処理するためのプログラムコードの詳細をプログラマーが直接扱う必要はなく、ただ次に説明するイベントリスナーとイベントを処理するメソッドについて理解するだけでかまいません。

## ■ イベントリスナー

さまざまなイベントを受け取るものを Java では特にイベントリスナー（EventListener）といいます。特定のイベントの処理に関連するクラスはそのイベントのリスナーインタフェースを実装します。

表 1.2 ● 主なイベントとイベントリスナー

イベント	リスナー
ウィンドウ	ウィンドウリスナー（WindowListener）
アクション	アクションリスナー（ActionListener）
チェンジ	チェンジリスナー（ChangeListener）
マウス	マウスリスナー（MouseListener）
キー	キーリスナー（KeyListener）

リスナーインタフェースを実装するクラスでは、イベントを処理する特定のメソッドを作成して（正確にはオーバーライドして）、そのメソッドでイベントに対応した処理や操作を行います。

作成したイベントリスナーは、コンポーネントの `addXxxListener()` メソッドを使ってコンポーネントに登録します。これでイベントが発生すると、オブジェクトの特定のメソッドが

呼び出されるようになります（具体的な例は第 2 章以降で説明します）。

ここで重要な点は、コンポーネントの種類にかかわらず、イベントはイベントリスナーが受け取って特定のイベント処理メソッドが呼び出されるという点と、そのメカニズムは Java の実行時ライブラリに隠されているという点です。そのため、Java の Swing による GUI アプリを作成するプログラマーがイベント処理のためにすることは、処理したいイベントのリスナーを実装するクラスを定義して、イベントを処理するメソッドを作成することです（具体的な例は第 2 章以降で説明します）。

この方法の基本的な考え方は、コンポーネントやイベントの種類にかかわらず基本的に同じです。

## 1.2 Java の GUI プログラミング

Java で GUI アプリを作成する方法はいくつかあります。

### ■ GUI ライブラリ

Java で GUI プログラミングを実現するためのライブラリ（パッケージ）にはさまざまな種類があります。通常、Java の GUI プログラミングで使うライブラリは、次のいずれかです。

- awt

基本的なユーザーインターフェースの作成およびグラフィックスとイメージの描画用のすべてのクラスを含むパッケージです。GUI コンポーネントとして awt を使った場合、デザインが環境に依存する（実行する環境によって異なった外観になる）という問題が発生することがあります。

しかし、awt の基本的なユーザーインターフェースについては、他のパッケージを併用したプログラミングでも問題なく使用できるため、コンポーネントのクラスに次に説明する Swing を使ったプログラミングでもこの awt のパッケージの要素を直接利用することができます。また、Swing と awt は基本的には併用可能です。

- 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一連の「軽量」コンポーネントを使います。以前は Swing は実行時の速度が遅いといわれたことがありますが、さまざまな面で改良や改善が行われて、現在では実用的な速度が実現されています。本書では主に Swing を使ったプログラミングについて説明します。



Swing を使ったプログラムの実行速度が速くなったからといっても、Java が完全なコンパイラ言語ではないので、特に速さを要求されるケースにはふさわしくありません。プログラムの速度が特に重要な場合は、C/C++ や Go 言語などの、ソースファイルから実行可能ファイルを生成するコンパイラ言語を使うべきです。

- SWT

Eclipse Foundation が Eclipse IDE と共に管理保守している Java 用 GUI 部品のツールキットのひとつです。開発用の統合ツールである Eclipse IDE と共に利用されることを前提としています。環境に依存する部分が多く、作成したプログラムが Java の動作するプラットフォームすべてで動作するとは限りません。またリソースをプログラムで必ず解放しなければならぬなど、やや難しい面があります。

- JavaFX

グラフィックスやアニメーション、ビデオなどを容易に扱うことが可能になるように設計されたもので、XML と CSS を併用してデザインを記述します。Java 11 以降、JavaFX は Oracle JDK に含まれておらず、スタンドアロンライブラリとして利用するようになりました。視覚的要素が特に重要なアプリの開発に適しています。

## ■ Java のバージョン

最近の Java のバージョン (Oracle の用語ではリリース) は、6 か月ごとに更新されています。そのうち、Java 11 は長期サポート (Long-Term-Support:LTS) リリースで、長期間のサポートが約束されています。

次の表に Oracle Java SE サポート・ロードマップの概要を示します (詳細は Oracle のサイトを参照してください)



表 1.3 ● Oracle Java SE サポート・ロードマップ (概要)

リリース	利用開始日	Premier Support 期限	Extended Support 期限
7	2011 年 7 月	2019 年 7 月	2022 年 7 月
8	2014 年 3 月	2022 年 3 月	2030 年 12 月
9	2017 年 9 月	2018 年 3 月	設定なし
10	2018 年 3 月	2018 年 9 月	設定なし
11 (LTS)	2018 年 9 月	2023 年 9 月	2026 年 9 月
12	2019 年 3 月	2019 年 9 月	設定なし
13	2019 年 9 月	2020 年 3 月	設定なし
14	2020 年 3 月	2020 年 9 月	設定なし
15	2020 年 9 月	2021 年 3 月	設定なし

本書では主にバージョン 11 以降の環境をターゲットとして説明しますが、本書のほとんどのプログラムは Java 8 以降のどの環境でも問題なく動作します。

## ■ バージョンの違いによる影響

Java の基本的な部分の多くは以前のバージョンを引き継いでいるため、バージョン多少が違ってても基本的な部分はあまり変わりません。そのため、一般的には次のようなことがいえま  
す (特定のバージョン間ではこれと異なることがあります)。

- Java のバージョンが異なっても、ほとんどの場合、プログラムをコンパイルして実行することが可能です。ただし、新しいバージョンに追加された機能を使っているプログラムを古いバージョンでコンパイルして実行するときや、新しいバージョンでは削除された機能を使っている古いプログラムをコンパイルして実行する場合は、多少の変更が必要になる場合があります。
- 特定のバージョン向けに書かれた Java プログラムを特定のバージョンの Java コンパイラ (javac) でコンパイルすると、そのバージョンで推奨されない事項はメッセージが出力され、削除された機能についてはコンパイルエラーとなります。
- 以前のバージョン向けに書かれた Java プログラムは、ほとんどの場合、そのまま新しいバージョンの Java でコンパイルして実行できます。しかし、実装や仕様の詳細が変

わっているために、実行できても動作や結果（アプリの外観などを含む）が異なる場合があります。

- ほとんどの場合、より以前の javac でコンパイルしたクラスファイルは、より新しい java で実行可能ですが、その逆はできない可能性があります。そのため、たとえば Java 8、Java 11、Java 14 のいずれかの環境（java）で実行される可能性があるプログラムは、Java 8 の javac でコンパイルする必要があります。
- ドキュメントの大半の部分はバージョンが異なっても同じです。そのため、使用するバージョンのドキュメントが見当たらないときには、多くの場合に他の近いバージョンのドキュメントを利用することができます。

# 第2章

---

## GUI アプリの基本



この章では、単純な GUI アプリの作り方を通して、Swing を使った GUI アプリの基本を解説します。ここで説明する基本的なプログラミング方法は、第3章以降のすべてのプログラムに共通します。

## 2.1 単純なウィンドウ

ここでは何も無い（空の）ウィンドウを作成する方法を説明します。

### ■ ウィンドウの作成

最初に説明するのは、何も無い（空の）ウィンドウを表示するプログラム `SmplWnd` です。このプログラムは、タイトルが `SmplWnd` の空のウィンドウを表示し、ウィンドウのクローズボックスがクリックされるとウィンドウを閉じてプログラムを終了します。

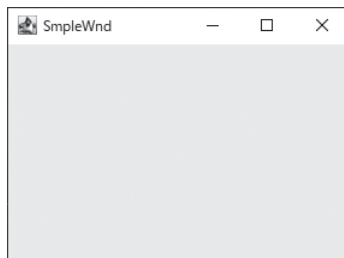


図 2.1 ● `SmplWnd`

最初にウィンドウを表示するだけの最も単純な GUI アプリのプログラム全体を次に示します。説明はプログラムリストのあとにあります。

#### リスト 2.1 ● `SmplWnd.java`

```
/*
 * SmplWnd.java
 */
import javax.swing.JFrame;

public class SmplWnd {

    public static void main(String[] args) {
        final JFrame frame = new JFrame("SmplWnd");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 220);
    }
}
```