20.UIとAPI

JavaのUI（ユーザーインターフェース）には、JavaFXやSwingなどのライブラリがあります。これらのライブラリを使用することで、GUI（グラフィカルユーザーインターフェース）を作成することができます。

JavaFXは、Java SE 8以降で標準で提供されているライブラリで、豊富なUIコントロールやアニメーション機能などを提供しています。JavaFXは、FXMLというXMLベースのマークアップ言語を使ってUIを記述することもできます。

Swingは、Javaの古いバージョンから存在するUIライブラリで、標準で提供されています。Swingは、豊富なUIコントロールやレイアウトマネージャーなどを提供しています。Swingは、JavaFXよりも軽量であり、レガシーシステムやリソースが限られている環境で使用されることがあります。

JavaのAPI（Application Programming Interface）には、標準のJava APIや、サードパーティ製のAPIがあります。標準のJava APIには、java.langやjava.utilなどのパッケージがあり、基本的なデータ型、コレクション、スレッド、入出力などの機能を提供しています。

また、Javaには、さまざまな分野で使用されるAPIが存在します。例えば、データベースとのやりとりに使用されるJDBC（Java Database Connectivity）APIや、ネットワーク通信に使用されるJava Networking APIなどがあります。

APIを使用することで、開発者は簡単にコードを再利用することができます。さまざまなAPIを組み合わせることで、複雑なアプリケーションを作成することができます。

例）

以下は、JavaFXを使用して作成されたシンプルなUIのサンプルコードです。

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.layout.StackPane;

import javafx.stage.Stage;

public class MyApplication extends Application {

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage primaryStage) {

}

上記の例では、StackPaneとButtonを使用して、ウィンドウにボタンを配置しています。また、Buttonには、アクションが発生したときにコンソールにメッセージを表示するイベントリスナーが設定されています。

以下は、JavaのAPIのサンプルコードです。例として、ArrayListクラスを使用して、文字列を格納してから、その要素を表示するプログラムを示します。

import java.util.ArrayList;

public class MyApplication {

public static void main(String[] args) {

ArrayList<String> list = new ArrayList<>(); list.add("Hello"); list.add("World");

for (String s : list) {

System.out.println(s); }

}

}

上記の例では、ArrayListクラスを使用して、文字列を格納しています。そして、forループを使用して、格納された要素を表示しています。ここでは、ジェネリックスを使用して、ArrayListが格納する要素の型を指定しています。

例）APIサンプル

以下は、JavaのAPIのサンプルコードです。例として、Mathクラスを使用して、円周率と正弦関数の値を計算するプログラムを示します。

public class MyApplication {

public static void main(String[] args) {

double pi = Math.PI;

double sin = Math.sin(pi / 2);

System.out.println("円周率: " + pi);

System.out.println("正弦関数(π/2): " + sin); }

}

上記の例では、Mathクラスを使用して、円周率と正弦関数の値を計算しています。Math.PIは円周率の値を返し、Math.sinメソッドは、指定された角度の正弦関数の値を返します。

以下は、JavaのAPIの別のサンプルコードです。例として、Stringクラスを使用して、文字列を操作するプログラムを示します。

public class MyApplication {

public static void main(String[] args) {

String str1 = "Hello";

String str2 = "World";

String str3 = str1 + " " + str2;

System.out.println(str3);

System.out.println("文字列の長さ: " + str3.length());

System.out.println("先頭の文字: " + str3.charAt(0));

System.out.println("部分文字列: " + str3.substring(6, 11));

}

}

上記の例では、Stringクラスを使用して、文字列を操作しています。文字列の連結には、+演算子を使用し、文字列の長さや先頭の文字、部分文字列の取得には、Stringクラスのメソッドを使用しています。これらのメソッドには、使用方法や引数、戻り値などの詳細な説明がJava API仕様書に含まれています。