

Java GUI 【3】

- サンプルプログラム(1) -

2011-09-25 Version 1.00

K. Yanai

目 次

1	概要	1
2	クラス構成	2
3	ソースコード例	3
4	課題	7

1 概要

まずは簡単なサンプルプログラムをみながら、Java GUI の基本的なことを学びましょう。

本サンプルは、図に示すようなひとつのメイン画面を使用します。おおまかな流れを説明します。

- (1) メイン画面を表示します。
- (2) メイン画面で、[参照]ボタンを選択します。
- (3) ファイル選択のウィンドウが表示されますので、ファイルを選択して、[開く]を選択します。
- (4) メイン画面のテキスト領域に、選択したファイル名が表示されます。
- (5) [実行する]ボタンを選択すると、何かの処理をします。(ここでは何もしません)
- (6) [終了する]ボタンを選択すると本サンプルプログラムを終了します。

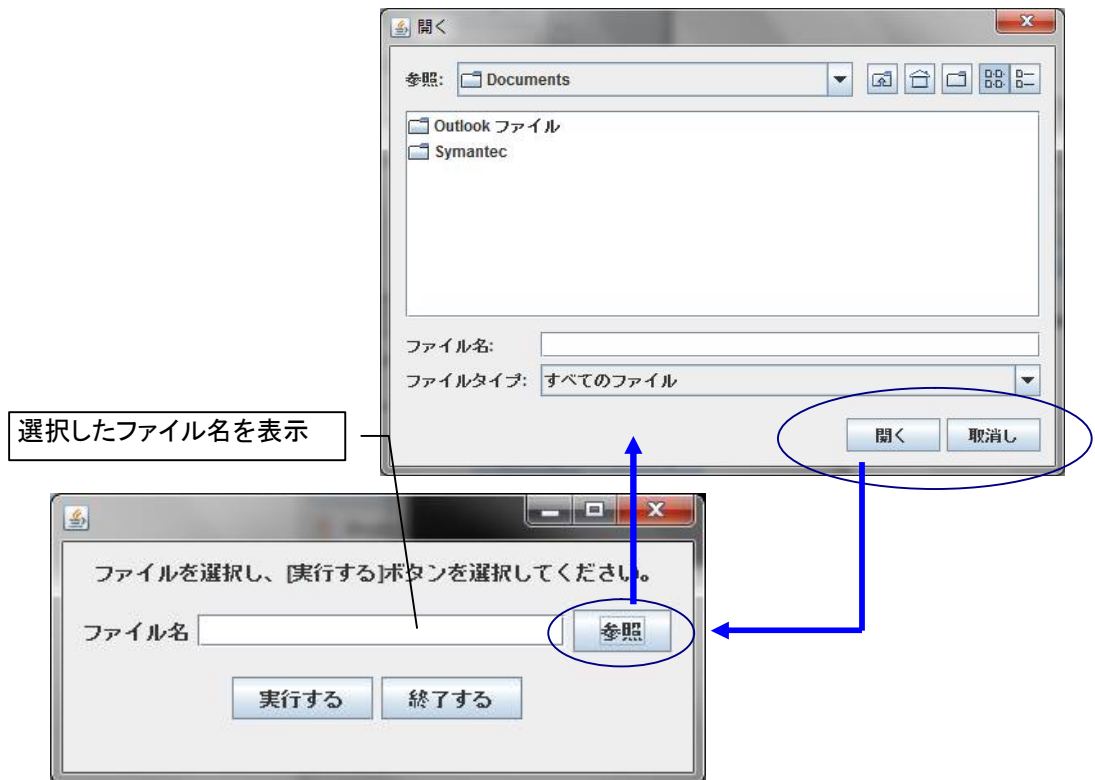


図 1-1 サンプルアプリケーションの画面(概略)

また、ファイルを選択していない状態で、[実行する]ボタンを選択すると、図のようなウィンドウ(メッセージダイアログ)を表示します。



図 1-2 メッセージダイアログ

2 クラス構成

本サンプルプログラムのクラス図を載せます。実際にコーディングするのは、FileApplication クラスだけです。

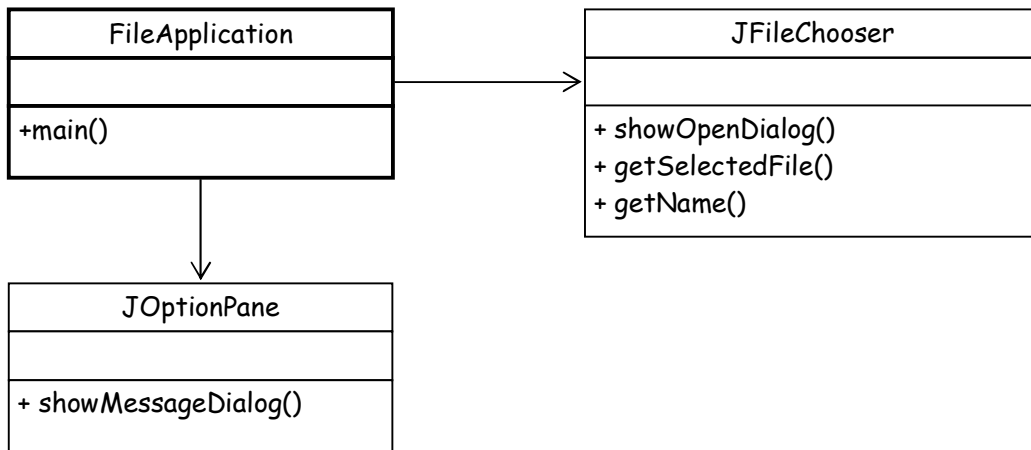


図 2-1 サンプルアプリケーションのクラス図(概略)

JFileChooser、JOptionPane については、下記のサイトを参照してください。

<http://java.sun.com/javase/ja/6/docs/ja/api/index.html?javax/swing/package-summary.html>

3 ソースコード例

サンプルのソースコードを載せておきます。

リスト 3-1 FileApplication

line	FileApplication
1	package jp.lites.study.java.gui;
2	
3	import java.awt.EventQueue;
4	
5	import javax.swing.JFrame;
6	import java.awt.GridBagLayout;
7	import javax.swing.JLabel;
8	import java.awt.GridBagConstraints;
9	import java.awt.Insets;
10	import java.awt.Window;
11	
12	import javax.swing.JFileChooser;
13	import javax.swing.JOptionPane;
14	import javax.swing.JPanel;
15	import javax.swing.BoxLayout;
16	import javax.swing.JTextField;
17	import javax.swing.JButton;
18	import javax.swing.JSplitPane;
19	import java.awt.FlowLayout;
20	import java.awt.event.ActionEvent;
21	import java.awt.event.ActionListener;
22	import java.io.File;
23	
24	public class FileApplication {
25	
26	private JFrame frame;
27	private JTextField textField;
28	private File file;
29	
30	/**
31	* Launch the application.
32	*/
33	public static void main(String[] args) {
34	EventQueue.invokeLater(new Runnable() {
35	public void run() {
36	try {
37	FileApplication window = new FileApplication();
38	window.frame.setVisible(true);
39	} catch (Exception e) {
40	e.printStackTrace();
41	}
42	}

line	FileApplication
43	});
44	}
45	
46	/**
47	* Create the application.
48	*/
49	public FileApplication() {
50	initialize();
51	}
52	
53	/**
54	* ウィンドウを初期化します.
55	*/
56	private void initialize() {
57	frame = new JFrame();
58	frame.setBounds(100, 100, 400, 178);
59	frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
60	frame.getContentPane().setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));
61	
62	JPanel panel_top = new JPanel();
63	FlowLayout fl_panel_top = (FlowLayout) panel_top.getLayout();
64	fl_panel_top.setAlignment(FlowLayout.LEFT);
65	frame.getContentPane().add(panel_top);
66	
67	
68	// ラベル: ガイドメッセージ
69	JLabel lblNewLabel = new JLabel("ファイルを選択し、[実行する]ボタンを選択してくださ
70	い。");
71	panel_top.add(lblNewLabel);
72	
73	JPanel panel_middle = new JPanel();
74	frame.getContentPane().add(panel_middle);
75	
76	// ラベル: ファイル名
77	JLabel label = new JLabel("ファイル名");
78	panel_middle.add(label);
79	
80	
81	// テキスト入力領域(選択したファイル名を入れます)
82	JTextField textField = new JTextField();
83	panel_middle.add(textField);
84	textField.setColumns(20);
85	
86	// [参照] ボタン
87	JButton button = new JButton("参照");
88	panel_middle.add(button);
89	

line	FileApplication
90	
91	// [参照] ボタンが選択されたときの処理
92	button.addActionListener(new ActionListener() {
93	public void actionPerformed(ActionEvent arg0) {
94	JFileChooser filechooser = new JFileChooser();
95	int selected = filechooser.showOpenDialog(textField);
96	if (selected == JFileChooser.APPROVE_OPTION) {
97	file = filechooser.getSelectedFile();
98	String name = filechooser.getName(file);
99	System.out.println("ファイル名 = " + name);
100	textField.setText(name);
101	} else {
102	System.out.println("ファイルが選択されませんでした。");
103	}
104	}
105	});
106	
107	
108	JPanel panel_button = new JPanel();
109	frame.getContentPane().add(panel_button);
110	panel_button.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));
111	
112	
113	// [実行する] ボタン
114	JButton exec_button = new JButton("実行する");
115	panel_button.add(exec_button);
116	
117	
118	// [実行する] ボタンが選択されたときの処理
119	exec_button.addActionListener(new ActionListener() {
120	public void actionPerformed(ActionEvent arg0) {
121	System.out.println("[実行する] selected");
122	
123	// TODO: ダイアログを表示してください。
124	if (file == null) {
125	JOptionPane.showMessageDialog(frame, "ファイルを選択してください。");
126	return;
127	}
128	JOptionPane.showConfirmDialog(frame, file + "を処理しますか?");
129	
130	// TODO: [はい] [いいえ] [取り消し] のうち、
131	// どのボタンが選択されたかを判断してお好みの処理をしてください。
132	
133	// TODO: なんらかの処理をします。
134	System.out.println("**** TODO: (Dummy)なんらかの処理をします。");
135	
136	}
137	});

line	FileApplication
138	
139	// [終了する] ボタン
140	JButton terminate_button = new JButton("終了する");
141	panel_buttonom.add(terminate_button);
142	
143	// [終了する] ボタンが選択されたときの処理
144	terminate_button.addActionListener(new ActionListener() {
145	public void actionPerformed(ActionEvent arg0) {
146	System.out.println("[終了] selected");
147	frame.dispose();
148	}
149	});
150	
151	}
152	
132	}

4 課題

ファイルを選択した状態で、[実行する]ボタンを選択すると、図のようなウィンドウ(ダイアログ)を表示し、どのボタンが選択されたかを判断し、適切な処理を行ってください。

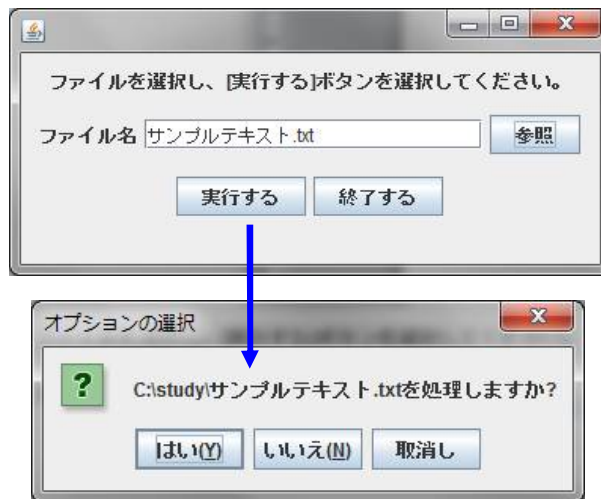


図 4-1 課題: 処理選択ウィンドウの表示と、選択したボタンの判断処理

// End of the document.