Graphical User Interface Development in JAVA Programming

Java is Object-Oriented Programming (OOP) language which permits higher level of abstraction than traditional procedural-oriented languages (such as C and Pascal). Java allows high-level abstract data types called classes to mimic real-life things. These classes are self-contained and are reusable. One can reuse the graphics classes provided in Java Development Kit (JDK) of the compiler been used for the development of Graphical User Interface (GUI) applications.

There are currently three sets of Java APIs for graphics programming: AWT (Abstract Windowing Toolkit), Swing and JavaFX.

1. AWT API was introduced in JDK 1.0. some of the AWT components have been replaced by newer Swing components.
2. Swing API, a much more comprehensive set of graphics libraries that enhances the AWT, was introduced as part of Java Foundation Classes (JFC) after the release of JDK 1.1. JFC consists of Swing, Java2D, Accessibility, Internationalization, and Pluggable Look-and-Feel Support APIs. JFC has been integrated into core Java since JDK 1.2.
3. The latest JavaFX, which was integrated into JDK 8, is meant to replace Swing.

Programming GUI with AWT

1. Only 2 packages - java.awt and java.awt.event are commonly used out of several packages and classes in the AWT API. The java.awt package contains the core AWT graphics classes:
    - GUI Component classes, such as Button, TextField, and Label.
    - GUI Container classes, such as Frame and Panel.
    - Layout managers, such as FlowLayout, BorderLayout and GridLayout.
    - Custom graphics classes, such as Graphics, Color and Font.
2. The java.awt.event package supports event handling:
    - Event classes, such as ActionEvent, MouseEvent, KeyEvent and WindowEvent,
    - Event Listener Interfaces, such as ActionListener, MouseListener, MouseMotionListener, KeyListener and WindowListener,
    - Event Listener Adapter classes, such as MouseAdapter, KeyAdapter, and WindowAdapter.

AWT provides a platform-independent *and* device-independent interface to develop graphic programs that runs on all platforms, including Windows, Mac OS etc.

Containers & Components

There are two types of GUI elements:

1. Component: Components are elementary GUI entities, such as push Button, Label, dialogbox and TextField.
2. Container: Containers, such as Frame and Panel, are used to hold components in a specific layout (such as FlowLayout or GridLayout). A container can also hold sub-containers.

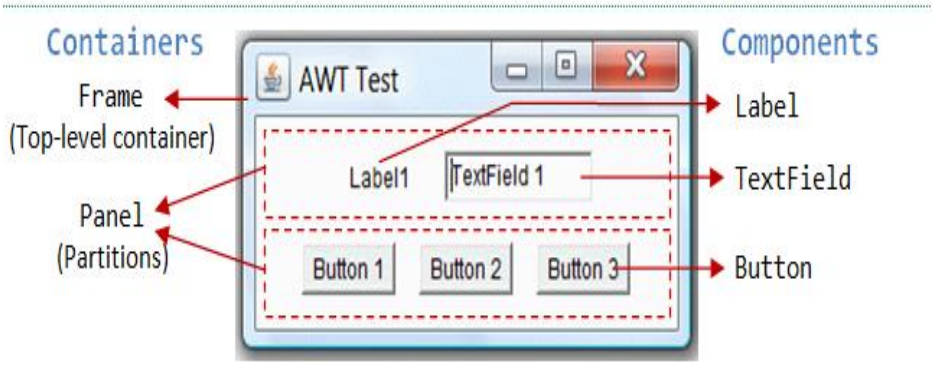

Figure 1: Illustration of types of containers and components

In figure 1, there are three containers: a Frame and two Panels. A Frame is the top-level container of an AWT program. A Frame has a title bar (containing an icon, a title, and the minimize/maximize/close buttons), an optional menu bar and the content display area. A Panel is a rectangular area used to group related GUI components in a certain layout. In the figure 1, the top-level Frame contains two Panels. There are five components: a Label (providing description), a TextField (for users to enter text), and three Buttons (for user to trigger certain programmed actions).

In a GUI program, a component must be kept in a container. You need to identify a container to hold the components. Every container has a method called add (Component c). A container (say c) can invoke c.add (aComponent) to add aComponent into itself. For example,

```
Panel pnl = new Panel();          // Panel is a container
Button btn = new Button("Press"); // Button is a component
```

```
pnl.add(btn);                    // The Panel container adds a Button component
```

With the support of the following libraries:

import java.awt.Image; import java.awt.Frame; import java.awt.Button; import java.awt.Panel; import java.awt.Graphics; import java.awt.Graphics2D;

AWT Container Classes

Top-Level Containers: Frame, Dialog and Applet

A Frame provides the "main window" for your GUI application. It has a title bar (containing an icon, a title, the minimize, maximize/restore-down and close buttons), an optional menu bar, and the content display area.

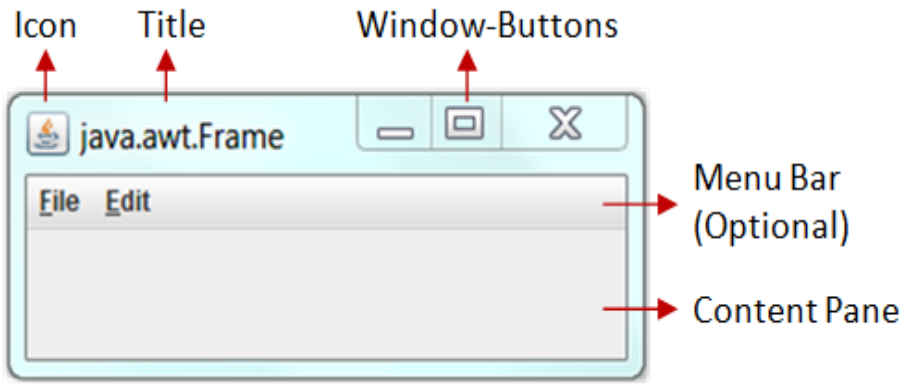

Figure 2: Top Level Containers

Secondary Containers: Panel and Scroll Panel

Secondary containers are placed inside a top-level container or another secondary container. AWT provides these secondary containers:

- Panel: a rectangular box used to layout a set of related GUI components in pattern such as grid or flow.
- ScrollPane: provides automatic horizontal and/or vertical scrolling for a single child component.
- others.

AWT Component Classes

AWT provides many ready-made and reusable GUI components in package java.awt. The frequently-usedare: Button, TextField, Label, Checkbox, CheckboxGroup (radio buttons), List, and Choice, as illustrated below.

Figure 3: Component classes /types

AWT GUI Component: java.awt.Label

A java.awt.Label provides a descriptive text string. Take note that System.out.println() prints to the system console, NOT to the graphics screen. A Label can be used to label another component (such as text field) to provide a text description.

```
public Label(String strLabel, int alignment); // Construct a Label with the given text String, of the text alignment
public Label(String strLabel);                // Construct a Label with the given text String
public Label();                               // Construct an initially empty Label
```

Constructing a Component and Adding the Component into a Container

Three steps are necessary to create and place a GUI component:

1. Declare the component with an identifier (name);
2. Construct the component by invoking an appropriate constructor via the new operator;
3. Identify the container (such as Frame or Panel) designed to hold this component. The container can then add this component onto itself via aContainer.add(aComponent) method. Every container has a add(Component) method. Take note that it is the container that actively and explicitly adds a component onto itself, NOT the other way.

```
Label lblInput;                     // Declare an Label instance called lblInput
lblInput = new Label("Enter ID");   // Construct by invoking a constructor via the
new operator
add(lblInput);                       // this.add(lblInput) - "this" is typically a
subclass of Frame
lblInput.setText("Enter password"); // Modify the Label's text string
lblInput.getText();
```

AWT GUI Component: java.awt.Button

A java.awt.Button is a GUI component that triggers a certain programmed action upon clicking.

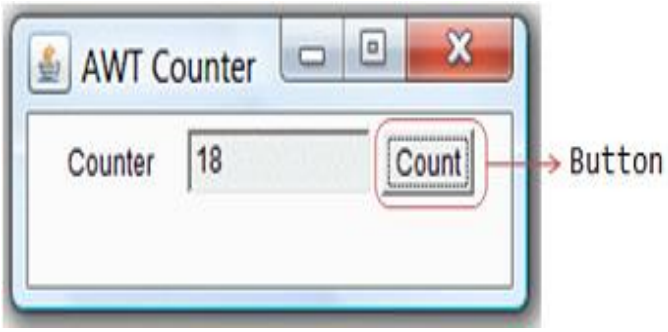

Figure 4: push button component

Event

Clicking a button fires a so-called ActionEvent and triggers user-defined programmed action.