CS205 Object Oriented Programming in Java

Module 5 - **Graphical User Interface and Database support of Java**

(Part 1)

Prepared by

Renetha J.B.

AP

Dept.of CSE,

Lourdes Matha College of Science and Technology

Topics

☑ Swings

☑ Swings fundamentals

☑ Swing Key Features

Swing fundamentals

- Swing is **written entirely in Java** (platform-independent).
  - So swing components are **light-weight**
  - Swing components are NOT implemented by platform-specific code.
- Swing is a **set of classes**.
- Swing is built on the foundation of the AWT(Abstract Window Toolkit).
- Swing provides more **powerful and flexible** functionalities than standard AWT components.
- Swing classes are defined in javax.swing package and its subpackages.

Swing Key Features

- Two key features of Swing are
  - Swing components are **Lightweight**
  - Swing supports a **Pluggable Look and Feel**

Swing Key Features-**Swing components are Lightweight**

- Swing Components are **lightweight** because
  - they are **written entirely in Java**
  - they do **not** map directly to **platform-specific peers.**
    - Peer classes are written by java API developers to interface with native objects
  - Lightweight components <u>do not call the native operating system for drawing the graphical user interface(GUI) components</u>
  - They are **rendered using graphics primitives**
    - they can be transparent, which enables nonrectangular shapes.
  - lightweight components are more efficient and more flexible.

Swing Key Features-**Swing components are Lightweight**(contd.)

- Lightweight components
  - **do not translate into native peers,**
  - the **look and feel of each component is determined by Swing**, not by the underlying operating system.
- This means that **each component will work in a consistent manner across all platforms.**

Swing Key Features-**Swing Supports a Pluggable Look and Feel**

- Swing supports a **pluggable look and feel (PLAF).**
  - Because each Swing component is **rendered by Java code** not by native peers, the look and feel of a component is under the control of Swing.
- It is possible to separate the **look and feel of a component** from the **logic of the component.**

❑ Advantage:

- It is possible to change the way that a component is rendered *without affecting any of its other aspects*
  - it is possible to **"plug in" a new look and feel for any given component** without creating any side effects in the code that uses that component.

Swing Key Features-**Swing Supports a Pluggable Look and Feel**

- It is possible to define entire sets of look-and-feels that represent *different GUI styles*
- To use a specific style, its look and feel is simply “plugged in.”
  - Once this is done, all components are automatically rendered using that style.
- Pluggable look-and-feels offer several important advantages.
  - It is possible to define a look and feel that is **consistent across all platforms.**
  - it is possible to create a look and feel that acts like a specific platform.
    - For example, if you know that an application will be running only in a Windows environment, it is possible to specify the Windows look and feel.
    - It is also possible to design a custom look and feel. Finally, the look and feel can be changed dynamically at run time.

Difference between Swing and AWT

Swing	**AWT**
Swing components are **not platform-dependent.**	AWT components are **platform-dependent**
Swing provides several **additional components** such as scroll panes, trees etc in addition to other standard components	The AWT defines a basic set of controls, windows, and dialog boxes that support a usable, but limited graphical interface.
Swing is **written entirely in Java**. So swing components are **light-weight**	AWT components use native code. So they are heavy weight
Swing supports a **pluggable look and feel** (PLAF) that can be dynamically changed at run-time depending on environment	In AWT look and feel of each component is fixed and it is difficult to change its look and feel.
Swing follow MVC	AWT does not follow MVC

Swing

- Swing is **light weight** Component.
- Swing needs **main method** to execute the program.
- Swing **follows MVC**(Model view Controller).
- Swing have its own Layout like most popular **Box Layout.**
- Swing uses for stand alone Applications.
- To execute Swing, the browser is not needed.

Applet

- Applet is heavy weight Component.
- Applet does not need main method to execute.
- Applet does not follow MVC.
- Applet uses AWT Layouts like Flowlayout.
- Applet need HTML code for Run.
- To execute Applet program we need browsers like Appletviewer, web browser etc.

Swing

- Swing is a set of program components for **Java** programmers that provide the **ability** to **create graphical user interface (GUI) components**, such as buttons and scroll bars, that are independent of the windowing system for specific operating system .

Reference

- Herbert Schildt, Java: **The Complete Reference, 8/e, Tata McGraw Hill, 2011**.