

6.1 Design philosophy of Swing

Java provides a rich set of libraries to create graphical user interface in a platform independent way. Swing GUI controls is part of java standard Library. It belongs to the JFC (Java foundation classes) and provides a set of components (also called Controls) for building desktop applications.

Swing is a Graphical User Interface (GUI) toolkit for java that is used to create rich desktop applications. It is a part of the java foundation classes (JFC) and serves as an extension of the Abstract Window Toolkit (AWT).

Swing provides a set of lightweight, platform-independent, and highly customizable GUI components like buttons, text fields, tables, trees, menus, and more. It is written entirely in java, making it portable across different platforms.

Swing components are found in the package: `javax.swing.*`

Design philosophy of Swing are:

1. platform independence :-

Swing is written entirely in java, so it runs the same on all OS (

windows, macOS, Linux, etc.). Unlike AWT, it does not depend on native OS components, which ensures consistent look and behavior across platforms.

2. Lightweight Components :-

Swing components are lightweight, meaning they do not rely on the native window system. Swing lightweight components are more customizable and portable. `JButton`, `JLabel` are lightweight, while AWT's `Button`, `Label` are heavyweight. Platform-independent (same look on all OS) because written entirely in Java.

3. Pluggable Look and Feel :-

Pluggable look and feel is a feature of Java Swing that allows developers to change the appearance and behavior of GUI components (like buttons, windows, menus) without changing the program's logic. It means you can "plug-in" different styles or themes for your application's interface at runtime.

Common Built in Look and feels:

metal → Default Java look (cross-platform)

Nimbus → Modern-looking swing theme

Motif → Unix-style look

Windows → matches native windows UI

mac → matches native macOS UI

`UIManager.setLookAndFeel()` method sets the desired theme.

4. MVC Architecture (Model - View - Controller) :-

Swing follows the MVC design pattern:

- ① Model :- Holds the data components (eg text in a text field).
- ② View :- Represents the UI (how it looks)
- ③ Controller :- Handles user actions (like button clicks).

5. Extensibility :- Swing components are easy to extend and customize. You can create new components or modify existing ones.

6. Event-Driven Programming :- Swing makes the application interactive and dynamic because it uses event listeners to respond to user actions like clicking, typing, etc. Common event listeners are:
ActionListener
MouseListener
KeyListener - etc

7. Double Buffering :- Swing uses double buffering to draw graphics smoothly. It draws on a hidden buffer first, then shows the final image, reducing flickering. Useful in animations or fast GUI updates.

8. Custom Painting :- Swing allows developers to draw custom graphics on components using the `paintComponent (Graphics g)` method.