Introduction to Swing

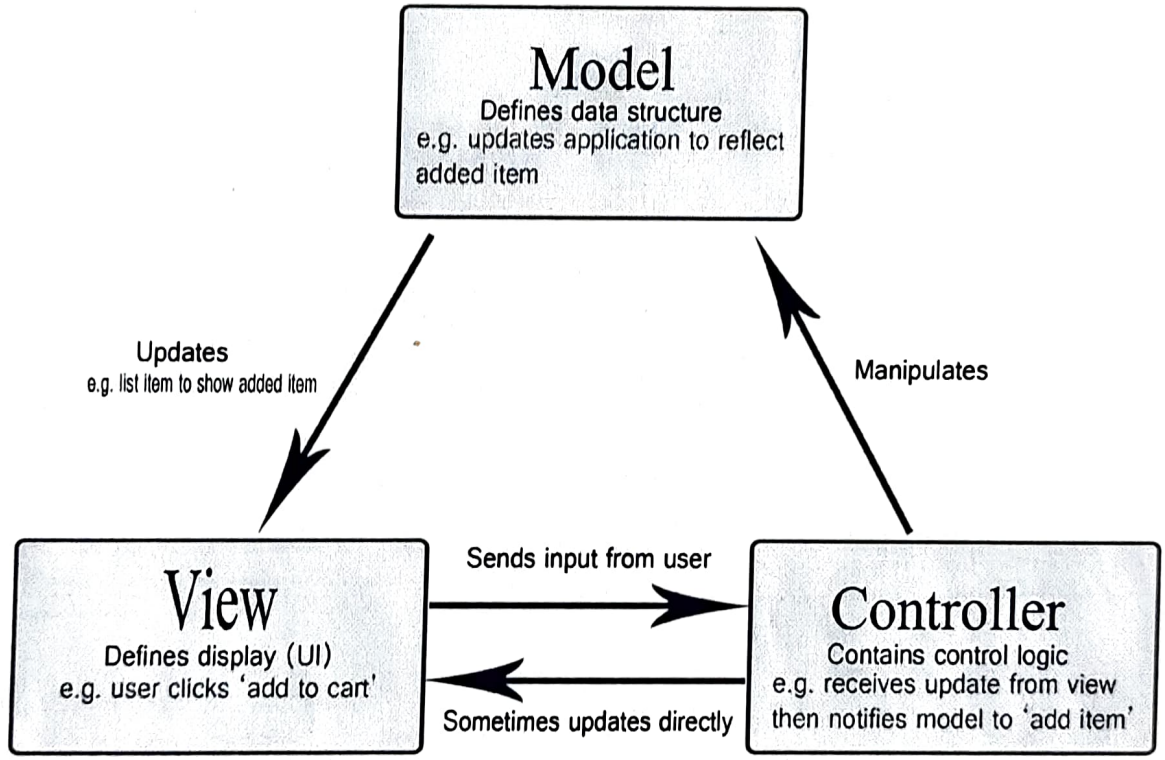
Swing Origins

- The original Java GUI subsystem was the Abstract Window Toolkit (AWT).
- AWT translates it visual components into platform-specific equivalents (peers).
- Under AWT, the look and feel of a component was defined by the platform.
- AWT components are referred to as **heavyweight**.
- Swing was introduced in 1997 to fix the problems with AWT.
- Swing offers two key features:
    1. Swing components are **lightweight** and don't rely on peers.
    2. Swing supports a pluggable look and feel. The three PLAFs available to all users are Metal (default), Windows, and Motif.
- Swing is built on AWT.

Model-View-Controller

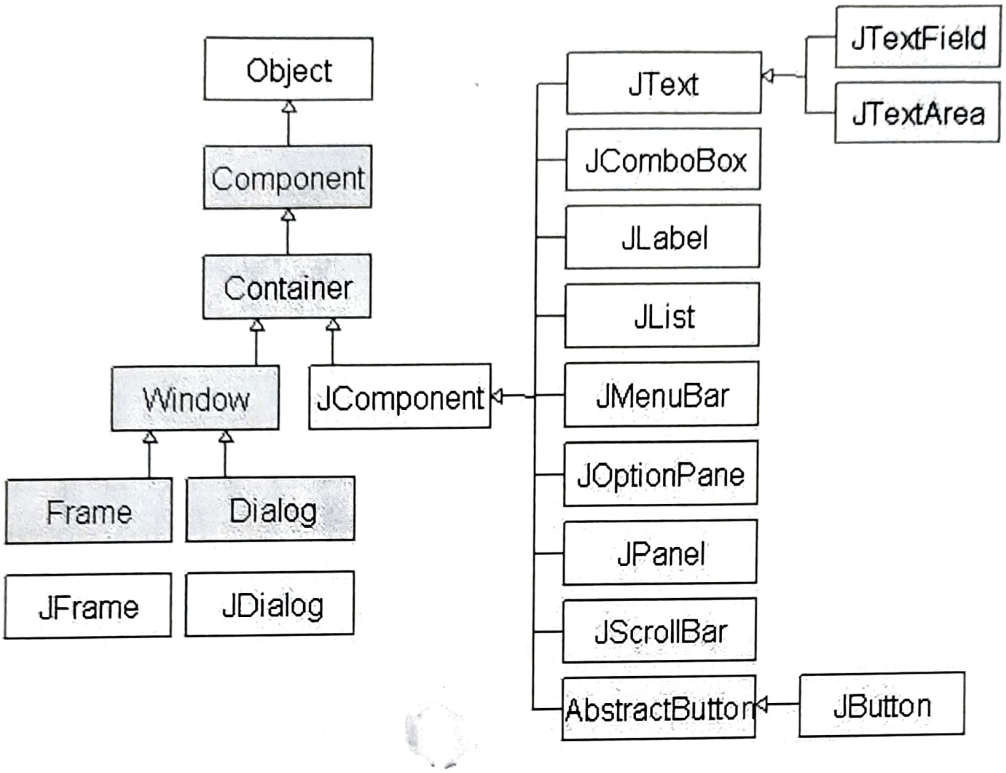
- One component architecture is MVC - Model-View-Controller.
- The **model** corresponds to the state information associated with the component.

- The **view** determines how the component is displayed on the screen.
- The **controller** determines how the component responds to the user.



Components

- Swing components are derived from the **JComponent** class. The only exceptions are the four top-level containers: JFrame, JApplet, JWindow, and JDialog.



Limitations of AWT:

The AWT defines a basic set of controls, windows, and dialog boxes that support a usable, but limited graphical interface.

One reason for the limited nature of the AWT is that it translates its various visual components into their corresponding, platform-specific equivalents or peers.

This means that the look and feel of a component is defined by the platform, not by java. Because the AWT components use native code resources, they are referred to as heavy weight.

- JComponent inherits AWT classes Container and Component.
- All the Swing components are represented by classes in the javax.swing package.
- All the component classes start with **J**: JLabel, JButton, JScrollbar, ...

Containers

- There are two types of containers: 1) top-level which do not inherit JComponent, and 2) Lightweight containers that do inherit JComponent.
- Lightweight components are often used to organize groups of components.
- Containers can contain other containers.
- All the component classes start with **J**: JLabel, JButton, JScrollbar, ...

The use of native peers led to several problems. First, because of variations between operating systems, a component might look, or even act, differently on different platforms. This variability threatened java's philosophy: write once, run anywhere.

Second, the look and feel of each component was fixed and could not be changed.

Third, the use of heavyweight components caused some frustrating restrictions.

Due to these limitations Swing came and was integrated to java. Swing is built on the AWT.

Two key Swing features are:

Swing components are light weight,

Swing supports a pluggable look and feel.