INTERVIEW QUESTIONS AND ANSWERS

Q1. What is Swing?

Swing is a Java GUI (Graphical User Interface) framework built on top of AWT (Abstract Window Toolkit). It provides a set of lightweight, platform-independent components for building desktop applications. Swing components are written entirely in Java and use a pluggable look-and-feel architecture, allowing applications to have different appearances while maintaining the same code. Key features include support for complex UI components, better performance, and easier customization compared to AWT.

Q2. Difference between AWT and Swing?

AWT (Abstract Window Toolkit) and Swing have several key differences:

- **Lightweight vs Heavyweight**: AWT components are heavyweight (depend on native OS components), while Swing components are lightweight (written entirely in Java).
- **Platform Dependency**: AWT looks and behaves differently across platforms, while Swing provides consistent look and feel across all platforms.
- **Functionality**: Swing offers more advanced components like JTable, JTree, JTabbedPane, while AWT has limited components.
- **Performance**: Swing is generally slower due to being Java-based, but AWT can have platform-specific issues.
- **Customization**: Swing allows easier customization through look-and-feel architecture.
- **Threading**: Swing requires all GUI updates to be done on the Event Dispatch Thread.

Q3. What is ActionListener?

ActionListener is an interface in Java that listens for action events generated by user interactions with components like buttons, text fields, and menu items. When an action event occurs (e.g., button click or pressing Enter), the actionPerformed() method is called. Implementation:

```
button.addActionListener(new ActionListener() {
```

```
    @Override
    public void actionPerformed(ActionEvent e) {
        // Handle the action
    }
});
```

Q4. How to manage layouts in Java?

Java provides several layout managers to arrange components in a container:

- **BorderLayout**: Divides container into five regions (NORTH, SOUTH, EAST, WEST, CENTER).
- **FlowLayout**: Places components in a row, wrapping to next row if needed.
- **GridLayout**: Arranges components in equal-sized grid cells.
- **BoxLayout**: Arranges components in a single row or column.
- **GridBagLayout**: Complex layout with fine-grained control over component placement.
- **CardLayout**: Shows one component at a time from multiple panels.
- **null Layout**: Manual positioning using setLayout(null) and setBounds().

Q5. What is the Event Dispatch Thread?

The Event Dispatch Thread (EDT) is a special thread in Swing responsible for processing all GUI events and updates. All Swing components and GUI modifications must be executed on the EDT to ensure thread safety and prevent race conditions. Methods like SwingUtilities.invokeLater() and SwingUtilities.invokeAndWait() are used to submit tasks to the EDT. This single-threaded model prevents multiple threads from accessing GUI components simultaneously.

Q6. What are the GUI components in Java?

Common Swing GUI components include:

- **JFrame**: Top-level container window

- **JPanel**: Container for grouping components
- **JButton**: Clickable button
- **JLabel**: Non-editable text display
- **JTextField**: Single-line text input
- **JTextArea**: Multi-line text input
- **JList**: Selectable list of items
- **JComboBox**: Dropdown selection menu
- **JCheckBox**: Checkbox for binary selection
- **JRadioButton**: Radio button for grouped selections
- **JTable**: Tabular data display
- **JTree**: Hierarchical tree structure
- **JMenuBar**: Menu bar container
- **JScrollPane**: Scrollable container
- **JDialog**: Modal or non-modal dialog window

Q7. How to handle multiple events?

Multiple events can be handled in several ways:

- **Single Listener for Multiple Components**: One listener can listen to events from multiple components by adding it to all of them.
- **Multiple Listeners**: Different listeners can be added to the same component for different event types.
- **Conditional Logic**: Use if-else or switch statements inside actionPerformed() to differentiate between event sources.
- **Separate Handler Classes**: Create separate listener classes for different event types.
- **Anonymous Inner Classes**: Create different anonymous listeners for different components.

Example:

```
if (e.getSource() == button1) {
```

```
    // Handle button1
} else if (e.getSource() == button2) {
    // Handle button2
}
```

Q8. What is JPanel vs JFrame?

- **JFrame**: Top-level container window that appears as a window on screen. It can stand alone and requires explicit setVisible(true) to display. Used as the main application window.
- **JPanel**: A lightweight container that cannot exist independently. It must be added to another container like JFrame. Used as a sub-container for grouping related components.

```
JFrame frame = new JFrame();
JPanel panel = new JPanel();
frame.add(panel); // Panel added to frame
```

Q9. How to add scroll bar in GUI?

Scroll bars are added using the JScrollPane wrapper component:

```
JList<String> list = new JList<>(model);
JScrollPane scrollPane = new JScrollPane(list);
scrollPane.setPreferredSize(new Dimension(300, 200));
panel.add(scrollPane);
```

JScrollPane automatically handles both horizontal and vertical scrolling. Other components like JTextArea and JTable can also be wrapped in JScrollPane for scrolling functionality.

Q10. What is MVC architecture?

MVC (Model-View-Controller) is a design pattern that separates an application into three components:

- **Model**: Contains the business logic and data (TodoModel). Manages the state and behavior of the application data.
- **View**: Displays data to the user (TodoApp GUI). Responsible for rendering the user interface.
- **Controller**: Handles user input and updates the model (Event handlers in TodoApp). Translates user interactions into model changes.

Advantages: Separation of concerns, easier testing, code reusability, and maintainability. In this To-Do app, Task and TodoModel form the Model layer, TodoApp is the View and Controller combined.