

Unit - 5

Layout managers

These are essential part of creating graphical user interface (GUI) for Java applications. They are used to control the arrangement and positions of components (such as buttons, labels, and text fields) within a container like (JFrame or JPanel).

various layout managers

① Border layout:-

Divide the Container into 4 directions: North, south, east, west.

② Grid layout:-

Components are arranged in a grid with fixed number of rows and columns.

grid layout

Arrange components in a single row or column.

④ spring layout :-

Allows you to define relationship between components using spring constraints.

⑤ card layout :-

Allows you to stack components on top of each other.

⑥ group layout :-

Provides a powerful and flexible way to create complex and nested layouts.

⑦ flow layout :-

Components are arranged in a row from left to right.

Awt Components

Awt (Abstract window Toolkit) is a set of classes and libraries in Java for building GUI for desktop application. Awt provides various components to create windows, buttons, text field and other elements.

Some components

1. Frame :-

A top-level window for creating a basic application window.

2. Dialos :-

A pop-up window typically used for secondary interaction or user input.

3. Button :-

A clickable component that triggers an action when pressed.

Label

A non-editable text input field
which can display used to
provide descriptions or information.

5. Text Field:-

A single-line text input field where
user can enter text.

6. Text Area:-

A multi-line text field input field
allow user to enter long text
messages.

7. checkbox:-

allows user to select or de-select
an option.

8. Scrollbar:-

used for scrolling.

9. Panel:-

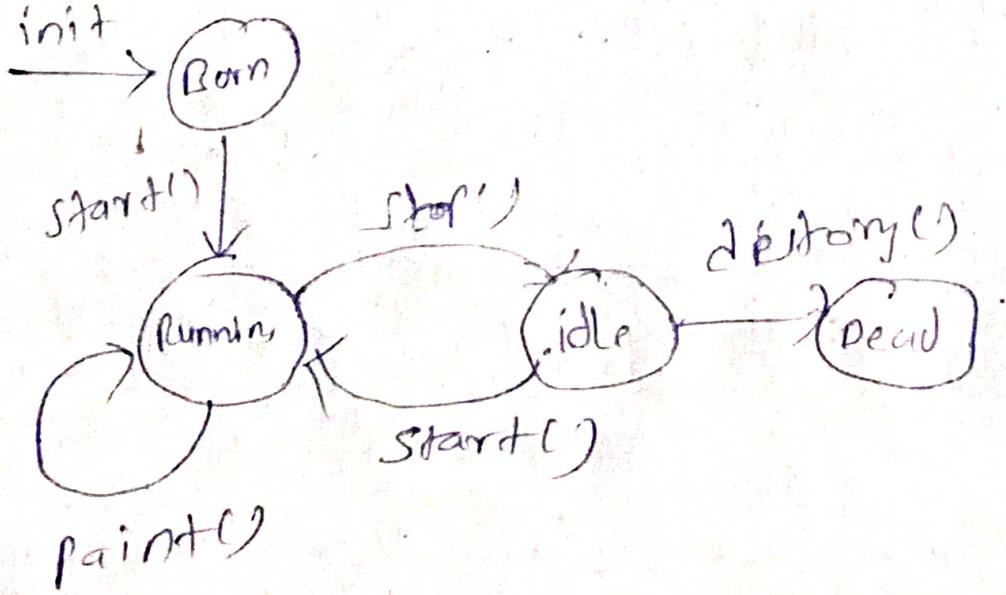
A light-weight container that
can hold other AWT components.

Applet life cycle

An applet is a special type of program embedded in the web page to generate dynamic content. The Applet life cycle can be defined as the process of how the object is created, started, stopped and destroyed during the entire ~~of~~ execution.

It basically has five core methods namely init(), start(), stop(), paint(), destroy().

Methods of Applet Life Cycle



`init()` → That initializes the applet

`start()` → The applet starts here

`stop()` → method stops the execution
of applet

`destroy()` → method destroys the applet
after work done

`paint()` → method belongs to the `Graphics`
class in Java. It is used to
draw shapes like circles, squares

* In `init()`, `start()`, `stop()`, `destroy`
belongs to `Applet` class

* `paint()` belongs to `awt.Component`
class.

Flow of Applet Lifecycle

init() method



start() method



paint() method



stop() method



destroy() method

Syntax for entire applet
lifecycle

class TestApplet extends
Applet {

 public void init() {

 // initialized objects

}

 public void start() {

 // code to start the applet

}

1 public void paint(Graphics graphics){
 //Draw the shapes}

2 public void stop() {
 //Code to stop the applet}

3 public void destroy() {
 //Code to destroy the applet}

Swing Components

Swing is a powerful and flexible GUI framework for Java. It provides a wide range of components and part of the Java X.Swing package and are designed to be platform independent and highly customizable.

1. JFrame :-

A top-level window that represents the main application window.

2. JPanel

A light-weight container used to group and organize other swing components.

3. JButton

A clickable button that misses an action when pressed.

4. JLabel:-

(1) A non-editable text or image display used for providing descriptive or information.

5. JTextField:-

A single-line text input field.

6. JTextArea :-

A multi-line text input field.

7. JCheckBox :-

A component that allows users to select or deselect an option.

8. JTable :-

A component for displaying and editing tabular data.

Applet Program for handle all mouse events

```
import java.applet.*;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
public class mouseEvents extends Applet
```

```
Applet implements MouseListener, MouseMotionListener
```

```
{}
```

private String message;

public void init() {

add mouseListener(this);

add mouse motionListener(this);

}

public void paint (Graphics g) {

g.drawString (message, 20, 20);

}

public void mouseClicked (MouseEvent

e) {

message = "Mouse clicked at (" +

e.getX() + ", " + e.getY() + ")";

repaint();

}

public void mousePressed (MouseEvent)

message = "Mouse pressed at "

repaint();

public void mouseReleased(MouseEvent e){

message = _____

repaint();

3
public void mouseMoved(MouseEvent e){

message = _____

repaint();

3
public void mousePressed(MouseEvent e){

message = _____

repaint();

3
public void mouseMoved(MouseEvent e){

it called when mouse is moved

3

3

Event listeners :-

Java supports wide range of event listeners through its Event listener interface and event-handling mechanism. This listeners enable Java applications to respond to various user interactions and system events.

① Action Listener:-

Used for handling action events, buttons or menu items.

② Mouse Listener:-

Consists of several methods
mouseClicked
Pressed

Released

Entered

MouseExited

3. Mouse motion Listener

provides methods
mouse moved
mouse pressed

4. Key Listener

Allows classes to respond to
key board events

5. windowListener

used to handle window-
related events like opening,
closing

JButton subclass for swing

- ① JToggleButton
- ② JCheckBox
- ③ JRadioButton
- ④ mixed toggle button UI
- ⑤ pluggable look

Event handling methods

Same for this

Event listeners.

Adapter class

In Java is a class that provides default implementations for all the methods of an interface allowing selectively override only the methods you need.

How adapter class are used in Event Handl.

(1)

- ① Providing Default implementation
- ② Selective override.

③ Cleaner code

By using adapter class, you can streamline the event handling code in your Java application and focus on the specific events that are relevant to your application's functionality.

Delegation event model

It is an event-handling mechanism used in Java to manage and respond to events in GUI. It is a fundamental part of Java AWT and Swing framework.

① Event Source:-

GUI are considered as event source.

2. Event Listener

are objects that listen for
and respond to events. Started
by event source.

3. Event object

an event object is created to
encapsulate information about
the event

4. Event handling methods

event listeners implement
specific event-handling methods
corresponding to the events they
want to handle

By the delegation event model
Java developer can create desktop
and interactive GUI applications
where components communicate through