

Introduction to GUI Programming



BY
M. BABY ANUSHA,
ASST.PROF IN CSE DEPT.,
RGUKT,NUZVID

Introduction :



- JAVA provides a rich set of libraries to create Graphical User Interface in platform independent way.
- Graphical User Interface (GUI) offers user interaction via some graphical components.
- For example our underlying Operating System also offers GUI via window, frame, Panel, Button, Textfield, TextArea, Listbox, Combobox, Label, Checkbox etc.

Graphical User Interface :



- These all are known as components. Using these components we can create an interactive user interface for an application.
- GUI provides result to end user in response to raised events. GUI is entirely based on events.
- For example clicking over a button, closing a window, opening a window, typing something in a textarea etc. These activities are known as events.
- GUI makes it easier for the end user to use an application. It also makes them interesting.

Basic Terminologies : GUI



Component

- Component is an object having a graphical representation that can be displayed on the screen and that can interact with the user.
- For examples buttons, checkboxes, list and scrollbars of a graphical user interface.

Container

- Container object is a component that can contain other components. Components added to a container are tracked in a list.

Basic Terminologies : GUI



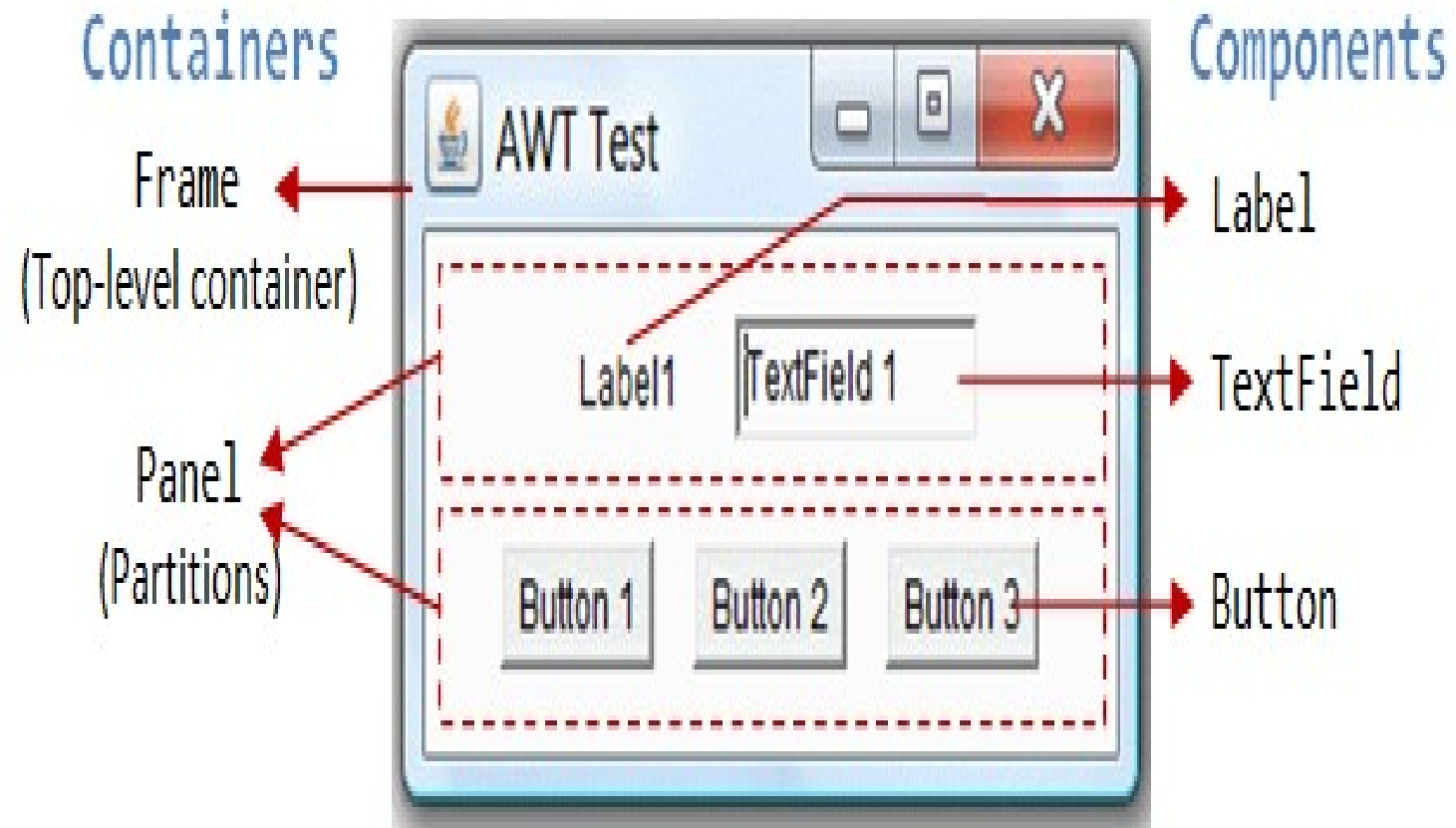
- The order of the list will define the components' front-to-back stacking order within the container. If no index is specified when adding a component to a container, it will be added to the end of the list.

Types of containers:

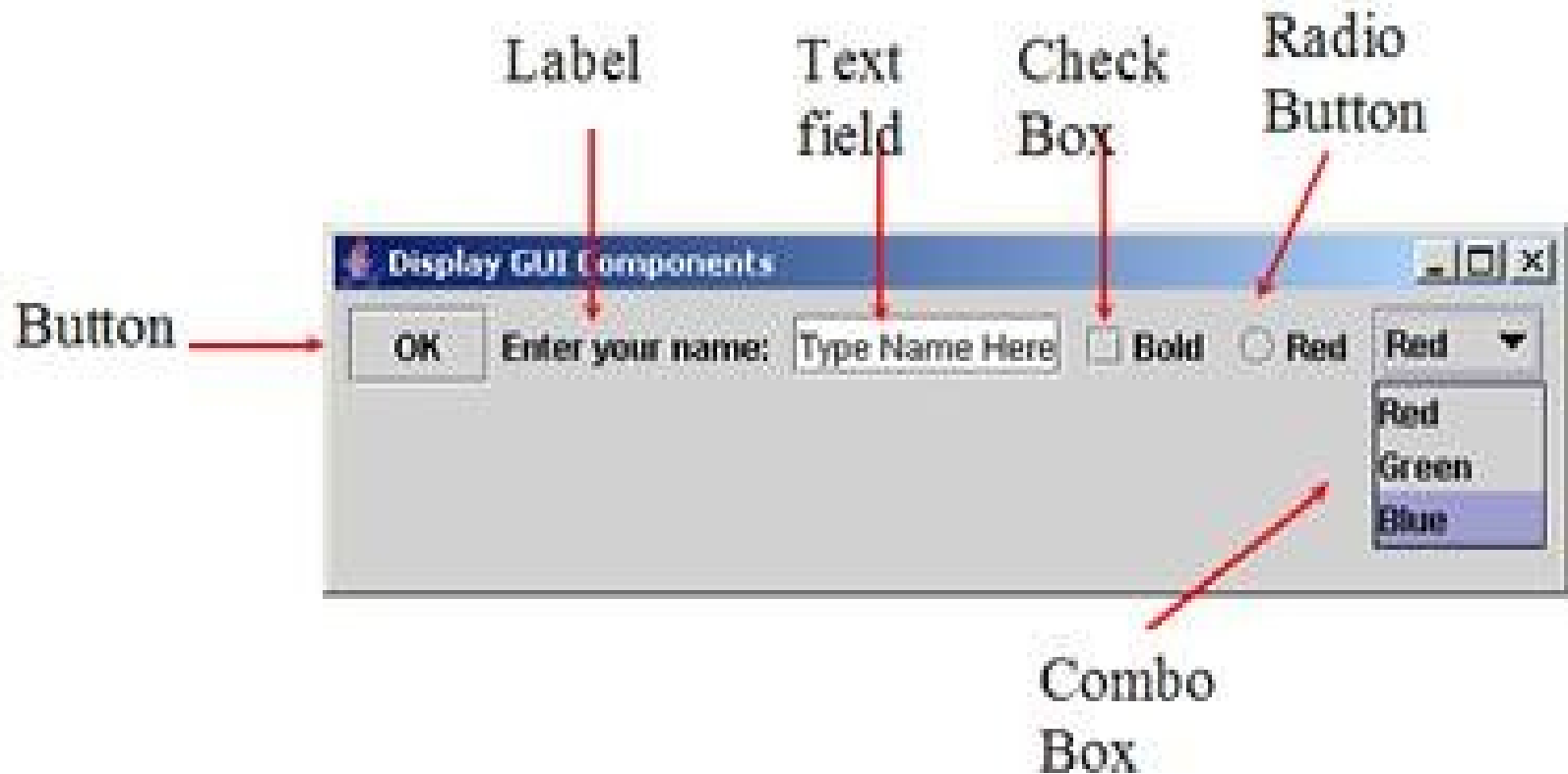
There are four types of containers in Java AWT:

- Window
- Panel
- Frame
- Dialog

Basic Terminologies : GUI



Basic Terminologies : GUI



Basic Terminologies : GUI



Panel

- Panel provides space in which an application can attach any other components, including other panels.

Window

- Window is a rectangular area which is displayed on the screen. In different window we can execute different program and display different data. Window provide us with multitasking environment.
- A window must have either a frame, dialog, or another window defined as its owner when it's constructed.

Basic Terminologies : GUI



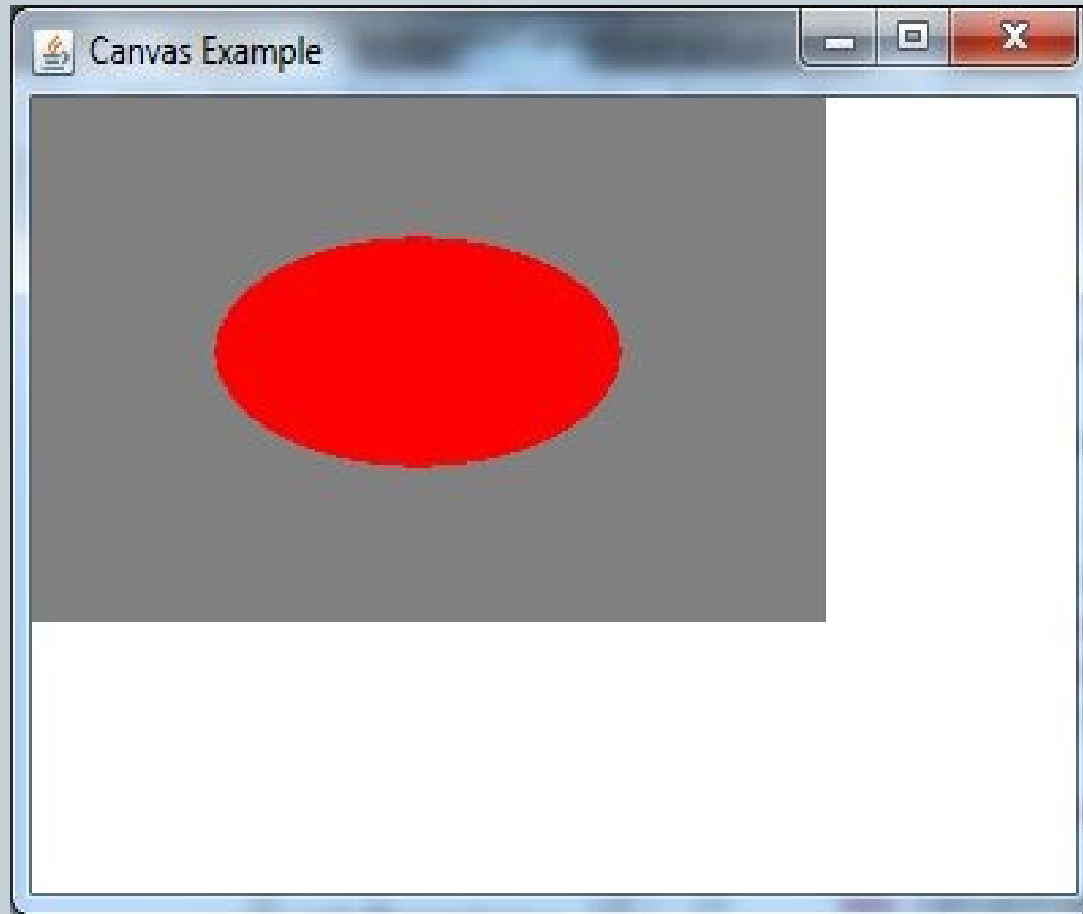
Frame

- A Frame is a top-level window with a title and a border. The size of the frame includes any area designated for the border.
- Frame encapsulates **window**. It and has a title bar, menu bar, borders, and resizing corners.

Canvas

- Canvas component represents a blank rectangular area of the screen onto which the application can draw. Application can also trap input events from the use from that blank area of Canvas component.

Basic Terminologies : GUI



Useful Methods of Component Class :



public void add(Component c)

- Inserts a component on this component.

public void setSize(int width,int height)

- Sets the size (width and height) of the component.

public void setLayout(LayoutManager m)

- Defines the layout manager for the component.

public void setVisible(boolean status)

- Changes the visibility of the component, by default false.

GUI based Applications : Examples



Following are some of the examples for GUI based applications.

- Automated Teller Machine (ATM)
- Airline Ticketing System
- Information Kiosks at railway stations
- Mobile Applications
- Navigation Systems

Advantages of GUI over CUI :



- GUI provides graphical icons to interact while the CUI (Character User Interface) offers the simple text-based interfaces.
- GUI makes the application more entertaining and interesting on the other hand CUI does not.
- GUI offers click and execute environment while in CUI every time we have to enter the command for a task.
- New user can easily interact with graphical user interface by the visual indicators but it is difficult in Character user interface.

Advantages of GUI over CUI :



- GUI offers a lot of controls of file system and the operating system while in CUI you have to use commands which is difficult to remember.
- Windows concept in GUI allow the user to view, manipulate and control the multiple applications at once while in CUI user can control one task at a time.
- GUI provides multitasking environment so as the CUI also does but CUI does not provide same ease as the GUI do.
- Using GUI it is easier to control and navigate the operating system which becomes very slow in command user interface. GUI can be easily customized.

Java AWT :



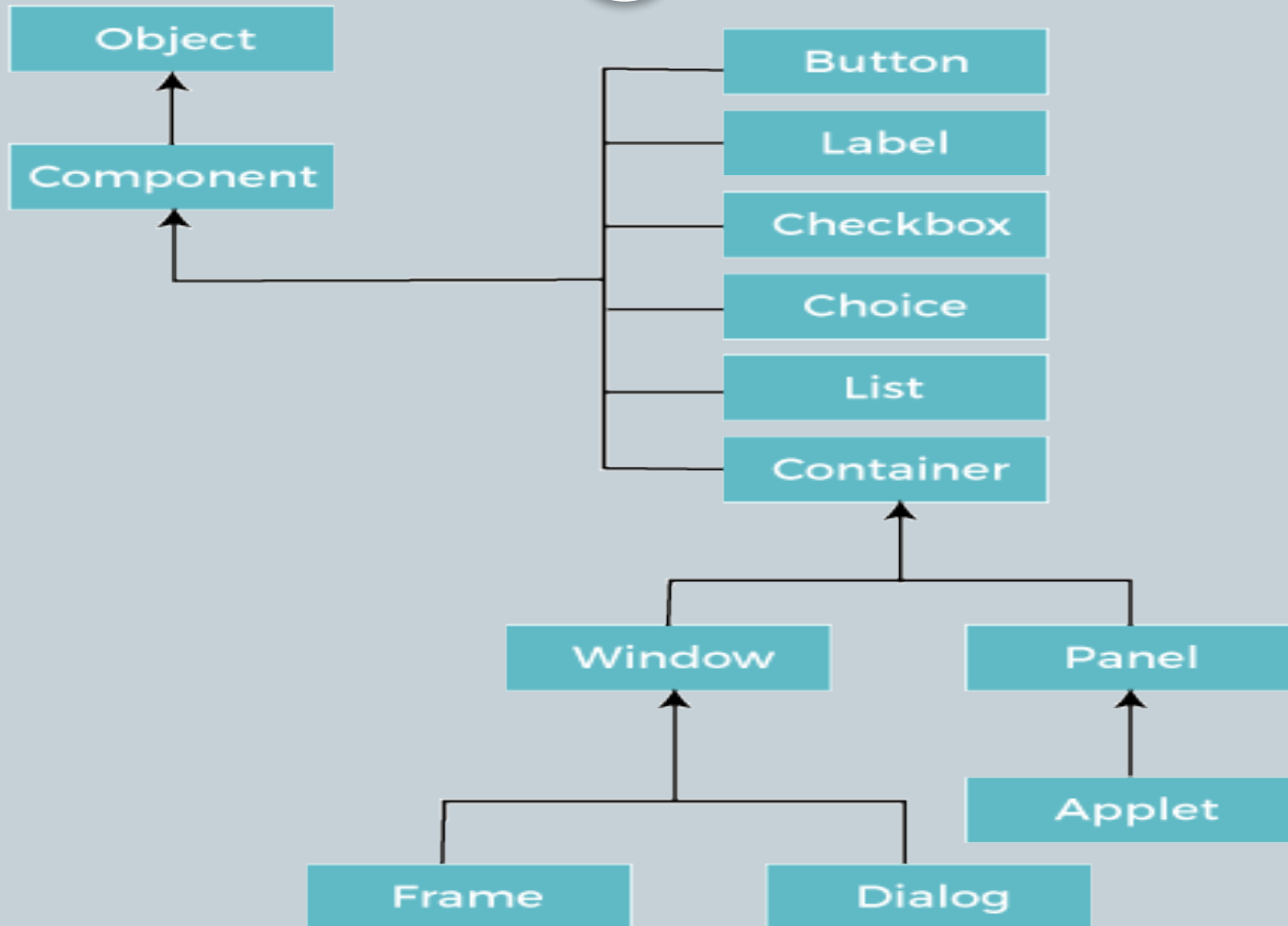
- **Java AWT** (Abstract Window Toolkit) is *an API to develop Graphical User Interface (GUI) or windows-based applications* in Java.
- Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system.
- AWT is heavy weight i.e. its components are using the resources of underlying operating system (OS).
- The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

Why AWT is platform independent?



- Java AWT calls the native platform calls the native platform (operating systems) subroutine for creating API components like TextField, CheckBox, button, etc.
- For example, an AWT GUI with components like TextField, label and button will have different look and feel for the different platforms like Windows, MAC OS, and Unix.
- The reason for this is the platforms have different view for their native components and AWT directly calls the native subroutine that creates those components.

Java AWT Hierarchy :



AWT : Example



```
import java.awt.*;
public class AWTExample1 extends Frame
{
    // initializing using constructor
    AWTExample1()
    {
        // creating a button
        Button b = new Button("Click Me!!");
        // setting button position on screen
        b.setBounds(30,100,80,30);
        // adding button into frame
        add(b);
        // frame size 300 width and 300 height
        setSize(300,300);
    }
}
```

AWT : Example



```
// setting the title of Frame
setTitle("This is our basic AWT example");

// no layout manager
setLayout(null);

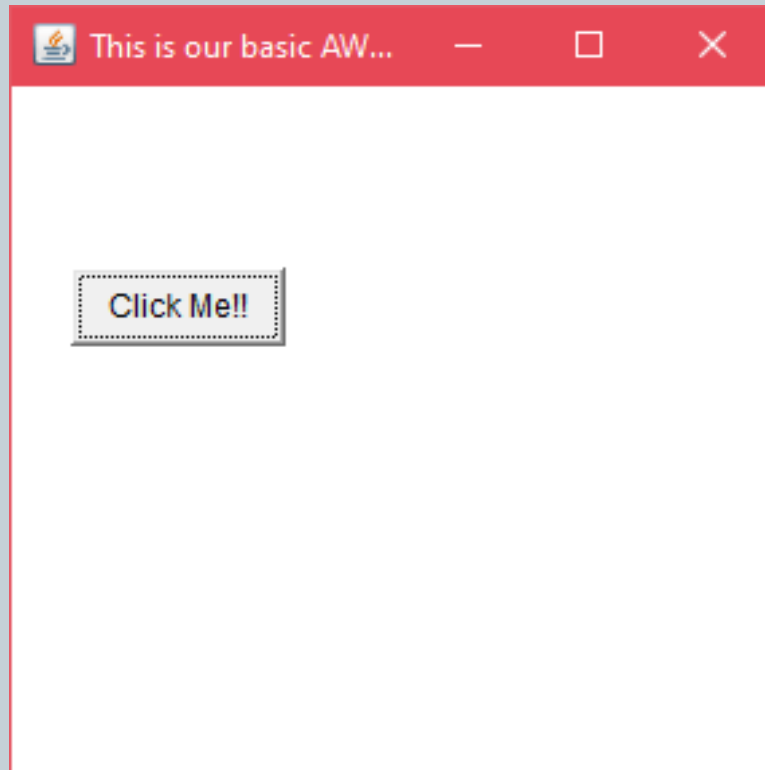
// now frame will be visible, by default it is not visible
setVisible(true);
}

// main method
public static void main(String args[])
{
    // creating instance of Frame class
    AWTEExample1 f = new AWTEExample1();
}
}
```

AWT : Example



Output:





Thank you!

shutterstock.com • 567687052