UNIT 11: AWT

UNIT STRUCTURE

11.1 Learning Objectives

11.2 Introduction

11.3 AWT Basics

11.4 AWT Components

11.5 Containers

11.6 Event Handling

11.7 Menus

11.8 Let Us Sum Up

11.9 Further Readings

11.10 Answers to Check Your Progress

11.11 Model Questions

11.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- describe AWT and its components
- describe event handling
- describe Swing and its components
- describe how to caeate menus

11.2 INTRODUCTION

In the previous unit, we learnt about applets in detail. In this unit, we will learn about the **AWT** package of the Java. The AWT is Java's original platform-independent windowing, graphics, and user-interface widget toolkit. The AWT is now part of the J**ava Foundation Classes** (JFC) — the standard API for providing a graphical user interface (GUI) for a Java program. Here we will provide a brief description about different AWT components, event handling in AWT. At the end, we will learn about menus in AWT. In the next unit, we will learn different concepts related to swings.

11.3 AWT BASICS

AWT (Abstract Window Toolkit) is an API to develop GUI or window-based applications in java. It contains all classes to write the program that interface between the user and different windowing toolkits. We can use the AWT package to develop user interface objects like buttons, checkboxes, radio buttons and menus etc.The java.awt package provides classes for AWT api such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

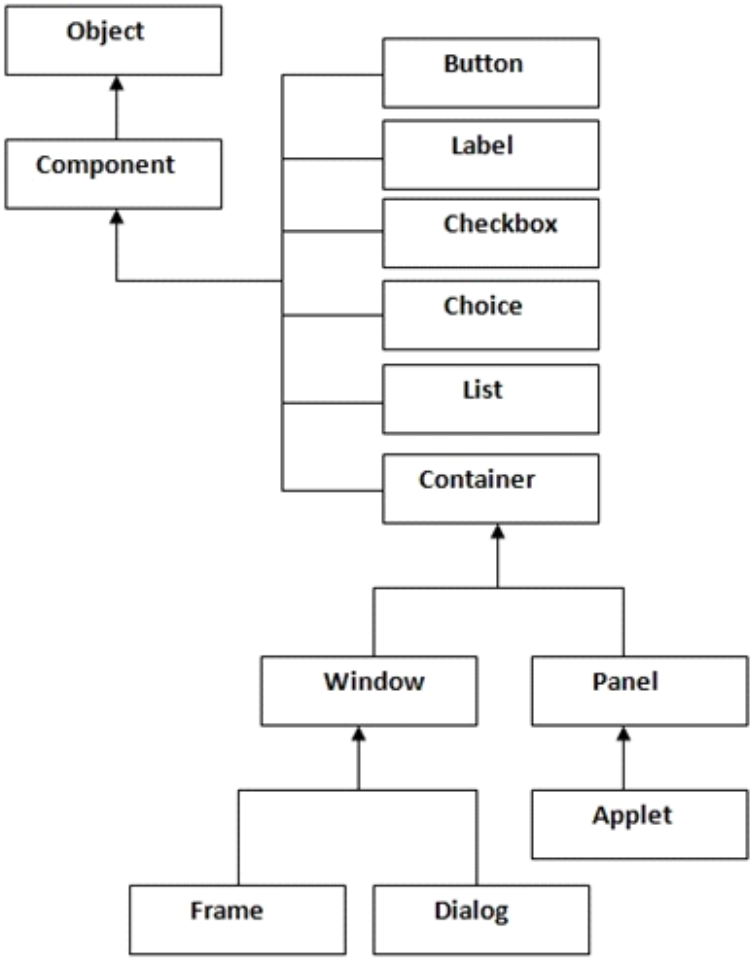


Figure 11.1: Java AWT Hierarchy

Basic Terminologies:

Component: Component is an object having a graphical representation that can be displayed on the screen and that can interact with the user like buttons, checkboxes, list etc.

Container:Container object is a component in AWT that can contain one or more components like buttons, textfields, labels etc. The classes that extends Container class are known as container such as Frame, Dialog and Panel.

Window: Window is a rectangular area which is displayed on the screen.Window provide us multitasking environment. A window must have either a frame, dialog or another window defined as its owner when it is constructed.

Panel: The Panel is the container that doesn't contain title bar and menu bars. Panel provides space in which an application can attach any other components, including other panels. It can have components like button, textfield etc.

Frame: The Frame is the container that contain title bar,menu bar, borders, and resizing corners. It can have other components like button, textfield etc.

11.4 AWT COMPONENTS

In this section, we will learn about the different components available in the Java AWT package for developing user interface for our program. Following are some of the components of Java AWT:

Labels : This is the simplest component of Java Abstract Window Toolkit. This component is generally used to show the text or string in our application label never perform any type of action. Syntax for defining the label is:

Label label_name = new Label ("This is the label text");

Above code simply represents the text for the label.

Label label_name = new Label ("This is the label text.", Label.CENTER);

label can be left, right or centered. Above declaration used the center justification of the label using the **Label.CENTER**

Buttons : This is the component of Java Abstract Window Toolkit and is used to trigger actions and other events required for our application. The syntax of defining the button is as follows:

Button button_name = new Button ("This is the label of the button.");

We can change the Button's label or get the label's text by using the Button.setLabel(String) and Button.getLabel() method. Buttons are added to its container using the add (button_name) method.

Check Boxes : This component of Java AWT allows us to create check boxes in our applications. The syntax for declaring a Checkbox is as follows :

CheckBox checkbox_name = new Checkbox ("Optional check box 1", false);

Above code constructs the unchecked checkbox by passing the Boolean valued argument false with the checkbox label through the checkbox() constructor. Defined Checkbox is added to its container using add (checkbox_name) method. We can change and get the checkbox's label using the **setLabel (String)** and **getLabel()** method. We can also set and get the state of the checkbox using the **setState(boolean)** and **getState()** method provided by the Checkbox class.

Radio Button : This is the special case of the Checkbox component of Java AWT package. This is used as a group of checkboxes which group name is same. Only one Checkbox from a Checkbox Group can be selected at a time. Syntax for creating radio buttons is as follows:

CheckboxGroup chkgp = new CheckboxGroup();
add (new Checkbox ("One", chkgp, false);
add (new Checkbox ("Two", chkgp, false);
add (new Checkbox ("Three",chkgp, false);

In the above code we are making three check boxes with the label "One", "Two" and "Three". If we mention more than one true valued for checkboxes then our program takes the last true and show the last check box as checked.

Text Area : This is the text container component of Java AWT package. The Text Area contains plain text. TextArea can be declared as follows:

TextArea txtArea_name = new TextArea();

We can make the Text Area editable or not using the setEditable (Boolean) method. If we pass the Boolean valued argument false then the text area will be non-editable otherwise it will be editable. The text area is by default in editable mode. Text are set in the text area using the setText(string) method of the TextArea class.

Text Field : This is also the text container component of Java AWT package. This component contains single line and limited text information. This is declared as follows:

TextField txtfield = new TextField(20);

We can fix the number of columns in the text field by specifying the number in the constructor. In the above code we have fixed the number of columns to 20.

As shown in the example below, a button is represented by a single label. That is the label shown in the example can be pushed with a click of a mouse.

//Program 11.1 : Program to show button represented by label.

```
import java.awt.*;
import java.applet.Applet;
public class MyButton extends Applet
     {
          public void init()
          {
               Button button = new Button("SUBMIT");
               add(button);
          }
     }
```

Here is the HTML code:

```
<HTML>
<HEAD>
</HEAD>
```