15.6.2 Button

Button class is used to create pushbutton or command button. A Button is the most common GUI control that lets you click on it do some action. You might already have seen it a lot of times, in **OK** and **Cancel** buttons.

Constructors of button class are :

Constructor	Description
Button ( )	Constructs a Button with no label
Button (string str)	Constructs a Button with the specified label str

Methods of button Class are :

Methods	Description
setLabel(String str)	To Add or Change the Label of button. with text str.
getLabel( )	To Obtain the label of the Button

Example

This program illustrates how to add buttons in a frame.

```
import java.awt.*;
class demobutton extends Frame
{
    public static void main(String args[ ])
    {
            Button b1=new Button("First Button");
            Button b2=new Button("Second Button");
            Button b3=new Button("Third Button");
            demobutton f1 = new demobutton();   //Create a frame
            f1.setTitle("Adding buttons in a Frame");   //Set title of frame
            f1.setLayout(new FlowLayout());          // Set a layout manager
            f1.add(b1);   //adding a component (button) to the frame
            f1.add(b2);
            f1.add(b3);
            f1.setSize(600, 300);   //set size of frame
            f1.setVisible(true);
    }
}
```

Output :

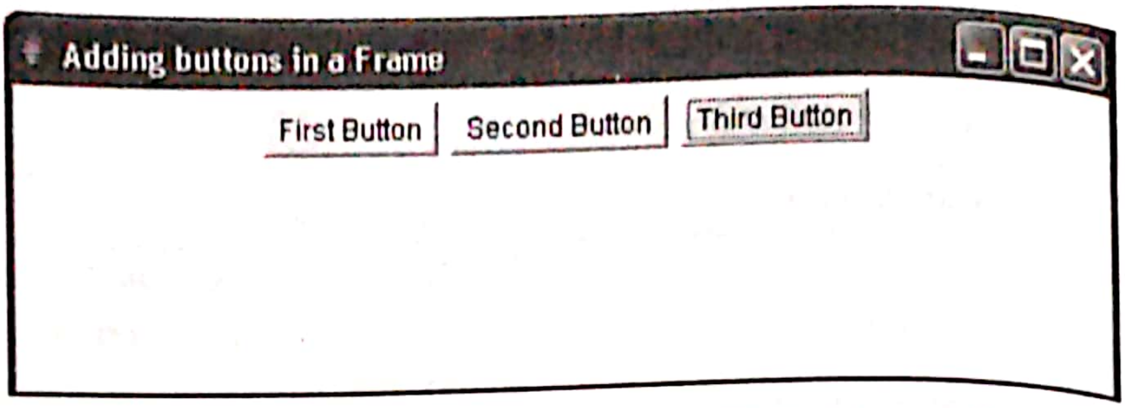


FIGURE 15.8

In this example, FlowLayout is a layout manager and it is used to arrange the components in a row. The **setLayout( )** method is used to set a new layout manager.

15.6.3 TextField

TextField provides an area where user can enter and edit a single line of text. TextFields are generally used for getting text input from a user. It can be created with any of the following constructors.

Constructor	Description
TextField ( )	To Create an empty text field
TextField (string str)	To Create a text field with the specified string
TextField (string str, int size)	To Create a text field with specified string and specified size (width)

Methods of TextField Class are :

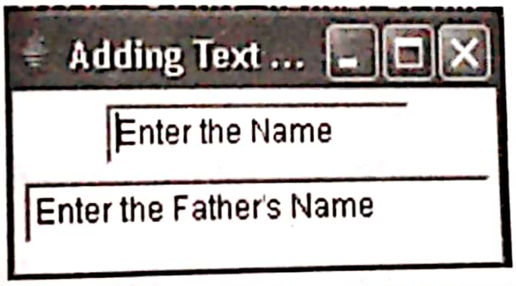
Methods	Description
getText( )	To Return the text that the TextField contain
setText (String str)	To Put the given String into the field
setColumns ( )	To Return the width of TextField
selectAll ( )	To Select all the Text in the Field
isEditable( )	Return True or False based on the Text is editable
setEditable (boolean)	True enabled text to be edited, False Freeze the Text.

Example

This program illustrates how to add text field in a frame.

```
import java.awt.*;
class demotextfield extends Frame
{
    public static void main(String args[ ])
    {
            TextField t1=new TextField("Enter the Name");
            TextField t2=new TextField("Enter the Father's Name");
            demotextfield f1 = new demotextfield();  //Create a frame
            f1.setTitle("Adding Text Field in Frame");  //Set title of frame
            f1.setLayout(new FlowLayout());
            f1.add(t1);  //adding a component to the frame
            f1.add(t2);
            f1.setSize(200, 100);  //set size of frame
            f1.setVisible(true);
    }
}
```

Output :



Adding Text ...

Enter the Name

Enter the Father's Name

FIGURE 15.9

15.6.4 TextArea

TextArea component is useful for getting a large data from the user. TextArea is similar as TextField but there is multiple line input. A TextArea is a multi-line area that displays text. As it contains multiple rows and columns, therefore, it can contain scrollbars. It can be set to allow editing text or read only modes.

Constructors of TextArea class are :

Constructor	Description
TextArea( )	Creates an empty Text Area
TextArea(int rows, int cols)	Creates a TextArea with specified rows and columns or width
TextArea (String str)	Creates a TextArea with string str
TextArea (String str, int rows, int cols)	Creates a TextArea with string str, specified rows and columns

Methods of TextArea Class are :

Methods	Description
setText(String s)	To Set the text of the TextArea
getText( )	To Retrieve the text of a TextArea
setEditable(boolean)	True enabled text to be edited, False Freeze the Text
isEditable( )	Return True or False based on the Text is editable

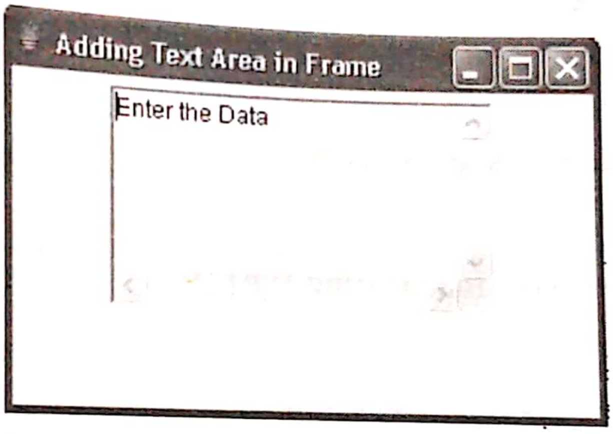
Example

This program illustrates how to add text area in a frame.

```
import java.awt.*;
class demotextarea extends Frame
{
    public static void main(String args[ ])
    {
                                                          Rows columns
        TextArea t1=new TextArea("Enter the Data" , 6,25);

        demotextarea f1 = new demotextarea();   //Create a frame
        f1.setTitle("Adding Text Area in Frame"); //Set title of frame
        f1.setLayout(new FlowLayout());
        f1.add(t1);                 //adding a component to the frame
        f1.setSize(300, 200);       //set size of frame
        f1.setVisible(true);
    }
}
```

Output :



Enter the Data

FIGURE 15.10