**Creating sources :--**

**Creating Labels:--**

AWT Class:-- Label

**Constructors of 'Label' class:--**

Label()

Label(String str)

Label(String str, int alignment)

|

|

|

|--------> Label.RIGHT

Label.LEFT

Label.CENTER

**Methods of 'Label' class:--**

i) setAlignment():--

void setAlignment(int alignment)

ii) getAlignment():--

int getAlignment()

iii) setText():--

void setText(String str)

iv) getText():--

String getText()

**Adding sources to Applet window:--**

Component

|----------------------|----------------------

| | |

| | |

Label Button Choice

Component add(Component obj)

Eg:--

import java.awt.\*;

import java.applet.\*;

/\*

<Applet code="LabelDemo" width="300" height="300">

</Applet>

\*/

public class LabelDemo extends Applet

{

Label L1, L2, L3;

public void init()

{

L1=new Label("One");

L2=new Label("Two");

L3=new Label("Three", Label.RIGHT);

add(L1);

add(L2);

add(L3);

L1.setBackground(Color.cyan);

L2.setBackground(Color.cyan);

L3.setBackground(Color.cyan);

L2.setAlignment(Label.CENTER);

}

}

**Creating Butttons:--**

**AWT Class** :-- Button

**Constructors:--**

Button()

Button(String str)

**Methods of ‘Button’ class:--**

1. setLabel() :-- void setLabel(String str)
2. getLabel():-- String getLabel()

**Event class:--**

ActionEvent

**Listener:**--

ActionListener

**Abstract method of ‘ActionListener’ interface :--**

public abstract void actionPerformed(ActionEvent ae)

**’Method of ‘ActionEvent’ class :--**

getActionCommand():-- String getActionCommand()

This method returns label of pressed button

Eg:--

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*

<Applet code="ButtonDemo" width="300" height="300">

</Applet>

\*/

public class ButtonDemo extends Applet implements ActionListener

{

Button b1,b2,b3;

public void init()

{

b1=new Button("Red");

b2=new Button("Green");

b3=new Button("Blue");

add(b1);

add(b2);

add(b3);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

String str=ae.getActionCommand();

Button b=(Button)ae.getSource();

if(b==b1)

setBackground(Color.red);

else if(b==b2)

setBackground(Color.green);

else if(b==b3)

setBackground(Color.blue);

}

}

**Creating Checkboxes:--**

**AWT Class** :-- Checkbox

**Constructors:--**

Checkbox()

Checkbox(String str)

Checkbox(String str, boolean on, CheckboxGroup cg)

Checkbox(String str, CheckboxGroup cg ,boolean on)

**Methods of ‘Checkbox’ class:--**

1. setLabel() :-- void setLabel(String str)
2. getLabel():-- String getLabel()
3. setState() :-- void setState(Boolean on)
4. getState() :-- boolean getState()

**Event class:--**

ItemEvent

**Listener:**--

ItemListener

**Abstract method of ‘ItemListener’ interface :--**

public abstract void itemStateChanged(ItemEvent ie)

Eg:--

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*

<Applet code="CheckDemo" width="300" height="300">

</Applet>

\*/

public class CheckDemo extends Applet implements ItemListener

{

Label L;

Checkbox ch1, ch2, ch3, ch4;

public void init()

{

L=new Label("Choose Players:--");

ch1=new Checkbox("Sachin");

ch2=new Checkbox("Saurav");

ch3=new Checkbox("Rahul");

ch4=new Checkbox("Anil");

add(L);

add(ch1);

add(ch2);

add(ch3);

add(ch4);

ch1.addItemListener(this);

ch2.addItemListener(this);

ch3.addItemListener(this);

ch4.addItemListener(this);

}

public void itemStateChanged(ItemEvent ie)

{

repaint();

}

public void paint(Graphics g)

{

String str="";

if(ch1.getState())

str=str + ch1.getLabel() + " ";

if(ch2.getState())

str=str + ch2.getLabel() + " ";

if(ch3.getState())

str=str + ch3.getLabel() + " ";

if(ch4.getState())

str=str + ch4.getLabel();

g.drawString(str,20,100);

}

}

**Creating Radio Buttons:--**

Eg:--

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*

<Applet code="RadioDemo" width="300" height="300">

</Applet>

\*/

public class RadioDemo extends Applet implements ItemListener

{

Label L;

Checkbox ch1, ch2, ch3;

CheckboxGroup cg;

public void init()

{

L=new Label("Choose Color:--");

CheckboxGroup cg=new CheckboxGroup();

ch1=new Checkbox("Red",false,cg);

ch2=new Checkbox("Green",true,cg);

ch3=new Checkbox("Blue",false,cg);

add(L);

add(ch1);

add(ch2);

add(ch3);

ch1.addItemListener(this);

ch2.addItemListener(this);

ch3.addItemListener(this);

}

public void itemStateChanged(ItemEvent ie)

{

repaint();

}

public void paint(Graphics g)

{

if(ch1.getState())

setBackground(Color.red);

else if(ch2.getState())

setBackground(Color.green);

else if(ch3.getState())

setBackground(Color.blue);

}

}

**Creating Choice:--**

**AWT Class** :-- Choice

**Constructors:--**

Choice()

**Methods of ‘Choice’ class:--**

1. add() :-- void add(String str)
2. getSelectedItem() :-- String getSelectedItem()
3. getSelectedIndex() :-- int getSelectedIndex()
4. getItemCount() :-- int getItemCount()
5. getItem() :-- String getItem(int index)
6. select() :-- void select(int index)

**Event class:--**

ItemEvent

**Listener:**--

ItemListener

**Abstract method of ‘ItemListener’ interface :--**

public abstract void itemStateChanged(ItemEvent ie)

**Eg:--**

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*

<Applet code="ChoiceDemo" width="300" height="300">

</Applet>

\*/

public class ChoiceDemo extends Applet implements ItemListener

{

Label L;

Choice ch;

public void init()

{

L=new Label("Choose Color:--");

ch=new Choice();

ch.add("Red");

ch.add("Green");

ch.add("Blue");

add(L);

add(ch);

ch.addItemListener(this);

}

public void itemStateChanged(ItemEvent ie)

{

repaint();

}

public void paint(Graphics g)

{

String str=ch.getSelectedItem();

if(str.equals("Red"))

setBackground(Color.red);

else if(str.equals("Green"))

setBackground(Color.green);

else if(str.equals("Blue"))

setBackground(Color.blue);

}

}

**Creating List:--**

**AWT Class** :-- List

**Constructors:--**

List(int numrows, boolean multselect)

**Methods of ‘List’ class:--**

1. add() :-- void add(String str)
2. getSelectedItem() :-- String getSelectedItem()
3. getSelectedIndex() :-- int getSelectedIndex()
4. getItemCount() :-- int getItemCount()
5. getItem() :-- String getItem(int index)
6. select() :-- void select(int index)
7. getSelectedItems() :-- String[ ] getSelectedItems()
8. getSelectedIndexes() :-- int[ ] getSelectedIndexes()

**Event class:--**

ItemEvent / ActionEvent

When user clicks on list item, ItemEvent is generated and when user double cliks on list item,ActionEvent is generated

**Listener:**--

ItemListener/ ActionListener

**Eg:--**

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*

<Applet code="ListDemo" width="300" height="300">

</Applet>

\*/

public class ListDemo extends Applet implements ActionListener

{

Label L;

List lst;

public void init()

{

L=new Label("Choose Players:--");

lst=new List(3, true);

lst.add("Sachin");

lst.add("Saurav");

lst.add("Rahul");

lst.add("Anil");

add(L);

add(lst);

//lst.addItemListener(this);

lst.addActionListener(this);

}

/\* public void itemStateChanged(ItemEvent ie)

{

repaint();

} \*/

public void actionPerformed(ActionEvent ae)

{

repaint();

}

public void paint(Graphics g)

{

String arr[ ] =lst.getSelectedItems();

String str="";

for(int i=0; i<arr.length; i++)

str=str + arr[i] + " ";

g.drawString(str,20,100);

}

}

**Creating scrollbar :--**

**AWT Class** :-- scrollbar

**Constructors:--**

Scrollbar()

Scrollbar(int style)

|

|---🡪 Scrollbar.HORIZONTAL

Scrollbar.VERTICAL

Scrollbar(int style, int initvalue, int thumbsize, int min, int max)

**Methods of ‘Scrollbar’ class:--**

i) getMinimum():-- int getMinimum()

ii) getMaximum():-- int getMaximum()

iii)getValue() :-- int getValue()

**Event class:--**

AdjustmentEvent

**Listener:**--

AdjustmentListener

**Method of ‘AdjustmentListener’ interface :--**

void adjustmentValueChanged(AdjustmentEvent e)

**Eg:--**

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*

<Applet code="ScrollDemo" width="300" height="300">

</Applet>

\*/

public class ScrollDemo extends Applet implements AdjustmentListener

{

Label L1,L2,L3;

Scrollbar scr1,scr2,scr3;

public void init()

{

L1=new Label("Red");

L2=new Label("Green");

L3=new Label("Blue");

scr1=new Scrollbar(Scrollbar.HORIZONTAL, 1, 1, 0, 255);

scr2=new Scrollbar(Scrollbar.HORIZONTAL, 1, 1, 0, 255);

scr3=new Scrollbar(Scrollbar.HORIZONTAL, 1, 1, 0, 255);

add(L1);

add(scr1);

add(L2);

add(scr2);

add(L3);

add(scr3);

scr1.addAdjustmentListener(this);

scr2.addAdjustmentListener(this);

scr3.addAdjustmentListener(this);

}

public void adjustmentValueChanged(AdjustmentEvent ae)

{

repaint();

}

public void paint(Graphics g)

{

Color c=new Color(scr1.getValue(),scr2.getValue(),scr3.getValue());

setBackground(c);

}

}

**Creating TextField :--**

**AWT Class** :-- TextField

**Constructors:--**

TextField(int numchars)

TextField(String str, int numchars)

**Methods of ‘TextField’ class:--**

1. setText():-- void setText(String str)
2. getText():-- String getText()
3. getSelectedText():-- String getSelectedText()

**Event class:--**

ActionEvent

This event is generated when user presses Enter button in TextField.

**Listener:**--

ActionListener

**Eg:--**

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*

<Applet code="TextDemo" width="300" height="300">

</Applet>

\*/

public class TextDemo extends Applet implements ActionListener

{

TextField t1;

public void init()

{

t1=new TextField(10);

add(t1);

t1.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

repaint();

}

public void paint(Graphics g)

{

g.drawString(t1.getText(), 20, 100);

}

}

**Creating TextArea :--**

**AWT Class** :-- TextArea

**Constructors:--**

TextArea(int numrows, int numchars)

TextArea(String str, int numrows, int numchars)

**Methods of ‘TextArea’ class:--**

1. setText():-- void setText(String str)
2. getText():-- String getText()
3. getSelectedText():-- String getSelectedText()

**Eg:--**

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*

<Applet code="TextAreaDemo" width="300" height="300">

</Applet>

\*/

public class TextAreaDemo extends Applet

{

TextArea t;

public void init()

{

String str="India is\nmy country";

t=new TextArea(str,5,20);

add(t);

}

}

**Using setBounds() method :--**

Eg:--

import java.awt.\*;

import java.applet.\*;

/\*

<Applet code="MyApplet" width="300" height="300">

</Applet>

\*/

public class MyApplet extends Applet

{

Button b1, b2;

public void init()

{

setLayout(null);

b1=new Button("One");

b2=new Button("Two");

add(b1);

add(b2);

b1.setBounds(20,20,100,50);

b2.setBounds(20,90,100,50);

}

}

Layout Managers:--

1. FlowLayout :-

import java.awt.\*;

import java.applet.\*;

/\*

<Applet code="MyApplet" width="700" height="300">

</Applet>

\*/

public class MyApplet extends Applet

{

Button b1,b2,b3,b4,b5;

public void init()

{

FlowLayout FL=new FlowLayout(FlowLayout.RIGHT);

setLayout(FL);

b1=new Button("One");

b2=new Button("Two");

b3=new Button("Three");

b4=new Button("Four");

b5=new Button("Five");

add(b1);

add(b2);

add(b3);

add(b4);

add(b5);

}

}

1. Border Layout:--

import java.awt.\*;

import java.applet.\*;

/\*

<Applet code="MyApplet" width="700" height="300">

</Applet>

\*/

public class MyApplet extends Applet

{

Button b1,b2,b3,b4,b5;

public void init()

{

BorderLayout BL=new BorderLayout();

setLayout(BL);

b1=new Button("One");

b2=new Button("Two");

b3=new Button("Three");

b4=new Button("Four");

b5=new Button("Five");

add(b1, BorderLayout.EAST);

add(b2, BorderLayout.WEST);

add(b3, BorderLayout.SOUTH);

add(b4, BorderLayout.NORTH);

add(b5, BorderLayout.CENTER);

}

}

Iii) Grid Layout:--

import java.awt.\*;

import java.applet.\*;

/\*

<Applet code="MyApplet" width="700" height="300">

</Applet>

\*/

public class MyApplet extends Applet

{

Button b1,b2,b3,b4,b5;

public void init()

{

GridLayout GL=new GridLayout(3, 2);

setLayout(GL);

b1=new Button("One");

b2=new Button("Two");

b3=new Button("Three");

b4=new Button("Four");

b5=new Button("Five");

add(b1);

add(b2);

add(b3);

add(b4);

add(b5);

}

}

1. Card Layout:--

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*

<Applet code="MyApplet" width="700" height="300">

</Applet>

\*/

public class MyApplet extends Applet implements ActionListener

{

Button b1,b2;

Checkbox ch1,ch2,ch3,ch4;

Panel p1=new Panel();

Panel p2=new Panel();

Panel mainpanel=new Panel();

CardLayout CL=new CardLayout();

public void init()

{

mainpanel.setLayout(CL);

b1=new Button("Sun");

b2=new Button("Microsoft");

add(b1);

add(b2);

b1.addActionListener(this);

b2.addActionListener(this);

ch1=new Checkbox("Core Java");

ch2=new Checkbox("Adv Java");

p1.add(ch1);

p1.add(ch2);

p1.setBackground(Color.cyan);

ch3=new Checkbox("VB.Net");

ch4=new Checkbox("ASP.Net");

p2.add(ch3);

p2.add(ch4);

p2.setBackground(Color.yellow);

mainpanel.add(p1, "sun");

mainpanel.add(p2, "micro");

add(mainpanel);

}

public void actionPerformed(ActionEvent ae)

{

String str=ae.getActionCommand();

if(str.equals("Sun"))

CL.show(mainpanel, "sun");

else

CL.show(mainpanel, "micro");

}

}

Creating Frames :---

Difference between applets & frames :--

1. Frames are light weight
2. Frames have some extra components like menus & dialog boxes
3. Frames are less secured

Resources for frames:--

Component

Java.awt Container

Panel Window java.awt

Java.applet-------- Applet Frame

MyApplet MyFrame

Eg of Frame:--

import java.awt.\*;

import java.awt.event.\*;

class MyFrame extends Frame implements

WindowListener, ActionListener

{

Button b1,b2,b3;

MyFrame()

{

setLayout(new FlowLayout());

addWindowListener(this);

b1=new Button("Red");

b2=new Button("Green");

b3=new Button("Blue");

add(b1);

add(b2);

add(b3);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

}

public void windowOpened(WindowEvent we)

{}

public void windowClosed(WindowEvent we)

{}

public void windowClosing(WindowEvent we)

{

setVisible(false);

dispose();

}

public void windowActivated(WindowEvent we)

{

setBackground(Color.cyan);

}

public void windowDeactivated(WindowEvent we)

{

setBackground(Color.yellow);

}

public void windowIconified(WindowEvent we)

{ setTitle("Minimized"); }

public void windowDeiconified(WindowEvent we)

{setTitle("Maximized");}

public void actionPerformed(ActionEvent ae)

{

Button b=(Button)ae.getSource();

if(b==b1)

setBackground(Color.red);

else if(b==b2)

setBackground(Color.green);

else if(b==b3)

setBackground(Color.blue);

}

}

class Demo

{

public static void main(String args[ ])

{

MyFrame m=new MyFrame();

m.setVisible(true);

m.setSize(300,300);

}

}

**Menus :--**

import java.awt.\*;

import java.awt.event.\*;

class MyFrame extends Frame implements

WindowListener, ActionListener

{

MenuItem m1,m2,m3;

MyFrame()

{

setLayout(new FlowLayout());

addWindowListener(this);

MenuBar mbar=new MenuBar();

setMenuBar(mbar);

Menu m=new Menu("Color");

mbar.add(m);

m1=new MenuItem("Red");

m2=new MenuItem("Green");

m3=new MenuItem("Blue");

m.add(m1);

m.add(m2);

m.add(m3);

m1.addActionListener(this);

m2.addActionListener(this);

m3.addActionListener(this);

}

public void windowOpened(WindowEvent we)

{}

public void windowClosed(WindowEvent we)

{}

public void windowClosing(WindowEvent we)

{

setVisible(false);

dispose();

}

public void windowActivated(WindowEvent we)

{

setBackground(Color.cyan);

}

public void windowDeactivated(WindowEvent we)

{

setBackground(Color.yellow);

}

public void windowIconified(WindowEvent we)

{ setTitle("Minimized"); }

public void windowDeiconified(WindowEvent we)

{setTitle("Maximized");}

public void actionPerformed(ActionEvent ae)

{

MenuItem m=(MenuItem)ae.getSource();

if(m==m1)

setBackground(Color.red);

else if(m==m2)

setBackground(Color.green);

else if(m==m3)

setBackground(Color.blue);

}

}

class Demo

{

public static void main(String args[ ])

{

MyFrame m=new MyFrame();

m.setVisible(true);

m.setSize(300,300);

}

}