MODULE 5

Java Applet

Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.

Advantage of Applet

There are many advantages of applet. They are as follows:

- It works at client side so less response time.
- Secured
- It can be executed by browsers running under many plateforms, including Linux, Windows, Mac Os etc.

Drawback of Applet

- Plugin is required at client browser to execute applet.
- Hierarchy of Applet

Lifecycle of Java Applet

Applet is initialized.

Applet is started.

Applet is painted.

Applet is stopped.

Applet is destroyed.

Lifecycle methods for Applet:

The java.applet.Applet class 4 life cycle methods and java.awt.Component class provides 1 life cycle methods for an applet.

java.applet.Applet class

For creating any applet java.applet.Applet class must be inherited. It provides 4 life cycle methods of applet.

public void init(): is used to initialized the Applet. It is invoked only once.

public void start(): is invoked after the init() method or browser is maximized. It is used to start the Applet.

public void stop(): is used to stop the Applet. It is invoked when Applet is stop or browser is minimized.

public void destroy(): is used to destroy the Applet. It is invoked only once.

java.awt.Component class

The Component class provides 1 life cycle method of applet.

public void paint(Graphics g): is used to paint the Applet. It provides Graphics class object that can be used for drawing oval, rectangle, arc etc.

How to run an Applet?

There are two ways to run an applet

- **By html file.**
- **By appletViewer tool (for testing purpose).**

Simple example of Applet by html file:

To execute the applet by html file, create an applet and compile it. After that create an html file and place the applet code in html file. Now click the html file.

//First.java

```
import java.applet.Applet;

import java.awt.Graphics;

public class First extends Applet{

public void paint(Graphics g){

g.drawString("welcome",150,150);

}
```

```
 }
```

Note: class must be public because its object is created by Java Plugin software that resides on the browser.

myapplet.html

```
<html>
<body>
<applet code="First.class" width="300" height="300">
</applet>
</body>
</html>
```

Simple example of Applet by appletviewer tool:

To execute the applet by appletviewer tool, create an applet that contains applet tag in comment and compile it. After that run it by: appletviewer First.java. Now Html file is not required but it is for testing purpose only.


```
//First.java
import java.applet.Applet;
import java.awt.Graphics;
public class First extends Applet{
  public void paint(Graphics g){
g.drawString("welcome to applet",150,150);
}
  }
/*
<applet code="First.class" width="300" height="300">
</applet>
```

*/

To execute the applet by appletviewer tool, write in command prompt:

c:\>javac First.java

c:\>appletviewer First.java

Displaying Graphics in Applet

java.awt.Graphics class provides many methods for graphics programming.

Commonly used methods of Graphics class:

public abstract void drawString(String str, int x, int y): is used to draw the specified string.

public void drawRect(int x, int y, int width, int height): draws a rectangle with the specified width and height.

public abstract void fillRect(int x, int y, int width, int height): is used to fill rectangle with the default color and specified width and height.

public abstract void drawOval(int x, int y, int width, int height): is used to draw oval with the specified width and height.

public abstract void fillOval(int x, int y, int width, int height): is used to fill oval with the default color and specified width and height.

public abstract void drawLine(int x1, int y1, int x2, int y2): is used to draw line between the points(x1, y1) and (x2, y2).

public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer): is used draw the specified image.

public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle): is used draw a circular or elliptical arc.

public abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle): is used to fill a circular or elliptical arc.

public abstract void setColor(Color c): is used to set the graphics current color to the specified color.

public abstract void setFont(Font font): is used to set the graphics current font to the specified font.

Example of Graphics in applet:

```
import java.applet.Applet;
import java.awt.*;
  public class GraphicsDemo extends Applet{
public void paint(Graphics g){
g.setColor(Color.red);
g.drawString("Welcome",50, 50);
g.drawLine(20,30,20,300);
g.drawRect(70,100,30,30);
g.fillRect(170,100,30,30);
g.drawOval(70,200,30,30);
  g.setColor(Color.pink);
g.fillOval(170,200,30,30);
g.drawArc(90,150,30,30,30,270);
g.fillArc(270,150,30,30,0,180);
 }
}
```

myapplet.html

```
<html>
<body>
<applet code="GraphicsDemo.class" width="300" height="300">
</applet>
</body>
</html>
```

Displaying Image in Applet

Applet is mostly used in games and animation. For this purpose image is required to be displayed. The java.awt.Graphics class provide a method drawImage() to display the image.

Syntax of drawImage() method:

1. **public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):** is used draw the specified image.

How to get the object of Image:

The java.applet.Applet class provides getImage() method that returns the object of Image. Syntax:

```
1. public Image getImage(URL u, String image){}
```

Other required methods of Applet class to display image:

1. **public URL getDocumentBase():** is used to return the URL of the document in which applet is embedded.
2. **public URL getCodeBase():** is used to return the base URL.

Example of displaying image in applet:

```
1. import java.awt.*;
2. import java.applet.*;
3.     public class DisplayImage extends Applet {
4.    Image picture;
5.     public void init() {
6.     picture = getImage(getDocumentBase(),"sonoo.jpg");
7.   }
8.
9.   public void paint(Graphics g) {
```

```
10.    g.drawImage(picture, 30,30, this);
11.  }
12.      }
```

In the above example, drawImage() method of Graphics class is used to display the image. The 4th argument of drawImage() method of is ImageObserver object. The Component class implements ImageObserver interface. So current class object would also be treated as ImageObserver because Applet class indirectly extends the Component class.

myapplet.html

```
1.  <html>
2.  <body>
3.  <applet code="DisplayImage.class" width="300" height="300">
4.  </applet>
5.  </body>
6.  </html>
```

Animation in Applet

Applet is mostly used in games and animation. For this purpose image is required to be moved.

Example of animation in applet:

```
1.  import java.awt.*;
2.  import java.applet.*;
3.  public class AnimationExample extends Applet {
4.     Image picture;
5.     public void init() {
6.     picture =getImage(getDocumentBase(),"bike_1.gif");
7.    }
8.       public void paint(Graphics g) {
9.     for(int i=0;i<500;i++){
10.      g.drawImage(picture, i,30, this);
11.        try{Thread.sleep(100);}catch(Exception e){}
```

```
12.    }
13.  }
14. }
```

In the above example, drawImage() method of Graphics class is used to display the image. The 4th argument of drawImage() method of is ImageObserver object. The Component class implements ImageObserver interface. So current class object would also be treated as ImageObserver because Applet class indirectly extends the Component class.

myapplet.html

```
1. <html>
2. <body>
3. <applet code="DisplayImage.class" width="300" height="300">
4. </applet>
5. </body>
6. </html>
```

EventHandling in Applet

As we perform event handling in AWT or Swing, we can perform it in applet also. Let's see the simple example of event handling in applet that prints a message by click on the button.

Example of EventHandling in applet:

```
1. import java.applet.*;
2. import java.awt.*;
3. import java.awt.event.*;
4. public class EventApplet extends Applet implements ActionListener{
5. Button b;
6. TextField tf;
7.   public void init(){
8. tf=new TextField();
9. tf.setBounds(30,40,150,20);
10.  b=new Button("Click");
```

```
11. b.setBounds(80,150,60,50);
12.   add(b);add(tf);
13. b.addActionListener(this);
14.  setLayout(null);
15. }
16.    public void actionPerformed(ActionEvent e){
17.   tf.setText("Welcome");
18.  }
19. }
```

In the above example, we have created all the controls in init() method because it is invoked only once.

myapplet.html

```
1.  <html>
2.  <body>
3.  <applet code="EventApplet.class" width="300" height="300">
4.  </applet>
5.  </body>
6.  </html>
```