BCA – 401: Java Programming

Rahul Kumar Singh

In today's Class we have discussed on Applet Programming in Java.

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.

Example:-

// 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.

// First.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.

Example:-

//First1.java

```
import java.applet.Applet;

import java.awt.Graphics;

public class First1 extends Applet
```

```
{
public void paint(Graphics g)
{
g.drawString("welcome to applet",150,150);
}
}

/*
<applet code="First1.class" width="300" height="300">
</applet>
*/
```

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

```
c:\>javac First1.java
c:\>appletviewer First1.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:-

```
// GraphicsDemo.java

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);
}
}
```

// GraphicsDemo.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:

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:

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

Other required methods of Applet class to display image:

public URL getDocumentBase(): is used to return the URL of the document in which applet is embedded.

public URL getCodeBase(): is used to return the base URL.

Example of displaying image in applet:

// DisplayImage.java

```
import java.awt.*;

import java.applet.*;

public class DisplayImage extends Applet

{

  Image picture;

  public void init()

{

    picture = getImage(getDocumentBase(),"sonoo.jpg");
```

```
 }
 public void paint(Graphics g)
{
   g.drawImage(picture, 30,30, this);
 }
 }
```

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.

//DisplayImage.htm

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

// AnimationExample.java

```
import java.awt.*;

import java.applet.*;

public class AnimationExample extends Applet

{

 Image picture;

 public void init()

 {

  picture =getImage(getDocumentBase(),"bike_1.gif");

 }

 public void paint(Graphics g)

 {

  for(int i=0;i<500;i++)

  {

   g.drawImage(picture, i,30, this);

   try

   {
      Thread.sleep(100);
       }
       catch(Exception e){}
   }
 }
}
```

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.

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

// EventApplet.java

```
import java.applet.*;

import java.awt.*;

import java.awt.event.*;

public class EventApplet extends Applet implements ActionListener

{

Button b;

TextField tf;

public void init()

{

tf=new TextField();

tf.setBounds(30,40,150,20);

b=new Button("Click");

b.setBounds(80,150,60,50);

add(b);
add(tf);
b.addActionListener(this);
setLayout(null);
}
 public void actionPerformed(ActionEvent e)
{
  tf.setText("Welcome");
 }
}
```

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

// EventApplet.html

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

Painting in Applet:-

We can perform painting operation in applet by the mouseDragged() method of MouseMotionListener.

Example of Painting in Applet:

// MouseDrag.java

```
import java.awt.*;

import java.awt.event.*;

import java.applet.*;

public class MouseDrag extends Applet implements MouseMotionListener

{

public void init()

{

addMouseMotionListener(this);

setBackground(Color.red);

}

public void mouseDragged(MouseEvent me)

{

Graphics g=getGraphics();

g.setColor(Color.white);

g.fillOval(me.getX(),me.getY(),5,5);
}
public void mouseMoved(MouseEvent me){}
}
```

In the above example, getX() and getY() method of MouseEvent is used to get the current x-axis and y-axis. The getGraphics() method of Component class returns the object of Graphics.

// MouseDrag.html

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

Parameter in Applet:-

We can get any information from the HTML file as a parameter. For this purpose, Applet class provides a method named getParameter().

Syntax:

```
public String getParameter(String parameterName)
```

Example of using parameter in Applet:

// UseParam.java

```
import java.applet.Applet;
import java.awt.Graphics;
public class UseParam extends Applet
{
public void paint(Graphics g)
{
String str=getParameter("msg");
g.drawString(str,50, 50);
}
}
```

// UseParam.html

```
<html>
<body>
<applet code="UseParam.class" width="300" height="300">
<param name="msg" value="Welcome to applet">
</applet>
</body>
</html>
```

Digital clock in Applet:-

Digital clock can be created by using the Calendar and SimpleDateFormat class. Let's see the simple example:

Example of Digital clock in Applet:

// DigitalClock.java

```
import java.applet.*;
import java.awt.*;
import java.util.*;
import java.text.*;
public class DigitalClock extends Applet implements Runnable
{
  Thread t = null;
  int hours=0, minutes=0, seconds=0;
  String timeString = "";
  public void init()
  {
    setBackground( Color.green);
  }
  public void start()
  {
      t = new Thread( this );
      t.start();
   }
   public void run()
  {
     try
      {
       while (true)
       {
         Calendar cal = Calendar.getInstance();
         hours = cal.get( Calendar.HOUR_OF_DAY );
         if ( hours > 12 ) hours -= 12;
         minutes = cal.get( Calendar.MINUTE );
         seconds = cal.get( Calendar.SECOND );
         SimpleDateFormat formatter = new
SimpleDateFormat("hh:mm:ss");
         Date date = cal.getTime();
         timeString = formatter.format( date );
         repaint();
         t.sleep( 1000 );  // interval given in milliseconds
       }
      }
      catch (Exception e) { }
   }
  public void paint( Graphics g )
{
      g.setColor( Color.blue );
      g.drawString( timeString, 50, 50 );
   }
}
```

In the above example, getX() and getY() method of MouseEvent is used to get the current x-axis and y-axis. The getGraphics() method of Component class returns the object of Graphics.

// DigitalClock.html

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

Analog clock in Applet:-

Analog clock can be created by using the Math class.

Example of Analog clock in Applet:

// MyClock.java

```
import java.applet.*;

import java.awt.*;

import java.util.*;

import java.text.*;

public class MyClock extends Applet implements Runnable
{

  int width, height;

  Thread t = null;

  boolean threadSuspended;

  int hours=0, minutes=0, seconds=0;

  String timeString = "";

  public void init()

  {

    width = getSize().width;

    height = getSize().height;

    setBackground( Color.black );

  }
```

```
public void start()
{
   if ( t == null )
  {
      t = new Thread( this );
      t.setPriority( Thread.MIN_PRIORITY );
      threadSuspended = false;
      t.start();
   }
   else
  {
      if ( threadSuspended )
      {
         threadSuspended = false;
         synchronized( this )
         {
            notify();
         }
      }
   }
  }
  public void stop()
  {
    threadSuspended = true;
  }
  public void run()
  {
    try
     {
      while (true)
      {
        Calendar cal = Calendar.getInstance();
        hours = cal.get( Calendar.HOUR_OF_DAY );
        if ( hours > 12 ) hours -= 12;
        minutes = cal.get( Calendar.MINUTE );
        seconds = cal.get( Calendar.SECOND );
        SimpleDateFormat formatter
          =       new       SimpleDateFormat(       "hh:mm:ss",
Locale.getDefault() );
        Date date = cal.getTime();
        timeString = formatter.format( date );
        // Now the thread checks to see if it should suspend
itself
        if ( threadSuspended )
        {
          synchronized( this )
          {
            while ( threadSuspended )
            {
              wait();
            }
          }
        }
        repaint();
        t.sleep( 1000 );  // interval specified in milliseconds
      }
    }
    catch (Exception e) { }
  }
  void drawHand( double angle, int radius, Graphics g )
  {
    angle -= 0.5 * Math.PI;
    int x = (int)( radius*Math.cos(angle) );
    int y = (int)( radius*Math.sin(angle) );
    g.drawLine( width/2, height/2, width/2 + x, height/2 + y );
  }
  void drawWedge( double angle, int radius, Graphics g )
  {
    angle -= 0.5 * Math.PI;
    int x = (int)( radius*Math.cos(angle) );
    int y = (int)( radius*Math.sin(angle) );
    angle += 2*Math.PI/3;
    int x2 = (int)( 5*Math.cos(angle) );
    int y2 = (int)( 5*Math.sin(angle) );
    angle += 2*Math.PI/3;
    int x3 = (int)( 5*Math.cos(angle) );
    int y3 = (int)( 5*Math.sin(angle) );
    g.drawLine( width/2+x2, height/2+y2, width/2 + x,
height/2 + y );
    g.drawLine( width/2+x3, height/2+y3, width/2 + x,
height/2 + y );
    g.drawLine( width/2+x2, height/2+y2, width/2 + x3,
height/2 + y3 );
```

```
  }
  public void paint( Graphics g )
  {
    g.setColor( Color.gray );
    drawWedge( 2*Math.PI * hours / 12, width/5, g );
    drawWedge( 2*Math.PI * minutes / 60, width/3, g );
    drawHand( 2*Math.PI * seconds / 60, width/2, g );
    g.setColor( Color.white );
    g.drawString( timeString, 10, height-10 );
  }
}
```

// MyClock.html

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

Write an Applet program to display "I LOVE JAVA" on the screen.

//First.java

```
import java.applet.Applet;
import java.awt.Graphics;
public class First extends Applet
{
public void paint(Graphics g)
{
g.drawString("I LOVE JAVA",150,150);
}
}
```

//First.html

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

Write an Applet program to accept two numbers from the user and display their sum.

// appletinput.java

```
import java.awt.*;

import java.awt.event.*

import java.applet.";

public class appletinput extends Applet implements ActionListener

TextField t1 = new TextField(10);

TextField t2= new TextField(10);

TextField t3 = new TextField(10);

Label l1 = new Label("FIRST NO:");

Label l2= new Label("SECOND NO:");

Label l3 = new Label("SUM:");

Button b = new Button("ADD");

public void init()

add(l1);

add(t1):

add(l2):

add(t2):

add(l3):
```

```
add(t3);
add(b);
b.addActionListener(this);
}
public void actionPerformed(ActionEvent e)
{
if (e.getSource() == b)
int n1 = Integer.parseInt(t1.getText());
int n2 = Integer.parseInt(t2.getText());
t3.setText("" + (n1 + n2));
}
}
}
```

// appletinput.html

```
<HTML>
<HEAD>
<TITLE>WELCOME TO JAVA APPLET</TITLE>
</HEAD>
<BODY>
```

```
<CENTER>
<H1>WELCOME TO THE APPLET</H1> </CENTER>
<BR>
<APPLET CODE=appletinput.class WIDTH=400
HEIGHT=400></APPLET>
</BODY>
</HTML>
```

Write an Applet program to find the greatest among 3 numbers.

// largenumber.java

```
importjava.applet.*;
importjava.awt.*;
importjava.awt.event.*;
public class largenumber extends Applet implements
ActionListener
{
TextField t1,t2,t3,t4;
  Button b1;
public void init()
  {
```

```
setLayout(null);
    t1 = new TextField(15);
t1.setBounds(100,25,50,20);
    t2 = new TextField(15);
t2.setBounds(100,50,50,20);
    t3 = new TextField(15);
t3.setBounds(100,75,50,20);
    t4 = new TextField("Ans");
t4.setBounds(175,40,50,20);
    b1 = new Button("Find");
b1.setBounds(175,65,50,30);
add(t1);
add(t2);
add(t3);
add(t4);
add(b1);
b1.addActionListener(this);
    }
public void actionPerformed(ActionEvent e)
    {
```

```
inti,j,k;
        i = Integer.parseInt(t1.getText());
        j=Integer.parseInt(t2.getText());
        k=Integer.parseInt(t3.getText());
if(i<j)
        {
if(j<k)
t4.setText(""+k);
else
t4.setText(""+j);
        }
else
t4.setText(""+i);
   }
}
```

// largenumber.html

```
<HTML>
<HEAD>
<TITLE>WELCOME TO JAVA APPLET</TITLE>
```

```
</HEAD>

<BODY>

<CENTER>

<H1>WELCOME TO THE APPLET</H1> </CENTER>

<BR>

<APPLET CODE=largenumber.class WIDTH=400
HEIGHT=400></APPLET>

</BODY>

</HTML>
```