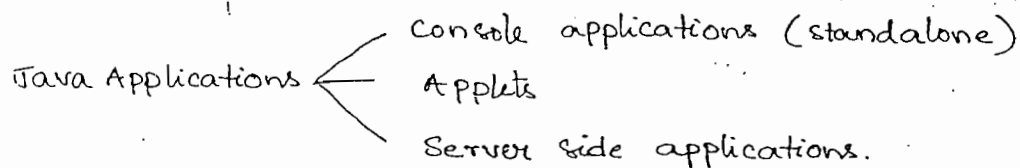


21-09-2006

Applets

- Applet: An applet is a java program which can be executed on a java enabled browser.

- Java applications can be classified as shown below



- Life cycle: It is the procedure followed by JVM to execute the programs.

NOTE: Life cycle differs from one category to another category of applications.

For example:

In console applications, the life cycle starts in the `main()` method and also ends in the `main()` method.

In applet, the life cycle starts in the `init()` method `start()` and ends in the `destroy` method.

In this way, it differs from one kind of application to another.

- We know that to execute an applet, a java enabled browser is needed. Also we know that only JVM can deal with java programs (i.e. .class files). Then

Then how can a browser execute an applet which is in turn a java program?

contd.. (P.T.O)

- This would be possible because, every java enabled browser has an in-built JVM in it which would deal with the applets (.class files).
- Browser is a window and it understands only html. How can browser understand the java programs (applets)?
Since browser understands only HTML, using HTML we inform the browser about the applet.
- There is a tag by name `<applet>`
The `<applet>` tag has three attributes, as given below
 - code - Mentions the name of the .java file which is to be executed as an applet.
 - height, width - These two attributes describe the dimensions of the applet.
- Whenever the browser encounters the `<applet>` tag, then it handsover the file name present in the 'code' attribute to the in-built JVM. Now, this JVM starts executing the program.
- Using applets we can display any GUI and the same GUI can be displayed even by using HTML in a more simpler way.

Question: why should we use applets, when we have an easy alternative (HTML)?

Ans: whenever it is required to hide the source code,

then we prefer applets to display the GUI's rather than using html to display the same GUI.

In other words, the disadvantage with HTML is that, in the browser, if we click the 'view source' option then the entire html code will be visible to the user which would be sometimes very dangerous.

If we use applets, then this possibility would not be there.

Life cycle of an applet

- As soon as browser encounters the `<applet>` tag in the html file, the browser hands over the .class file to its JVM mentioned in the 'code' attribute.
- JVM creates an object of that class and this object would be assigned to a reference of the Applet class.
- Now, JVM calls init() method and after that it calls start() method on the reference of the Applet class.
- If control has to come out of the applet program, then JVM calls stop() method and then the control is relinquished.
- Suppose if control comes back to the html page on which the applet program was running, then once again the html code will be executed and in this process, `<applet>` tag will be encountered and once again the .class file would be handed over to the JVM.

- Now, JVM will not create another object, since the object has already been created earlier.
- Since the object is not created (for the second time) `init()` method will not be executed. Instead, JVM executes the `start()` method.

NOTE: `init()` method would be called only once in the lifecycle of an applet

- If the browser is closed, then before deleting the object of the .class file, JVM calls the `destroy()` method on the reference of the Applet class
- The four functions (`init()`, `start()`, `stop()`, `destroy()`) through which JVM executes the applet program are known as Life Cycle Methods
- These life cycle methods are defined in Applet class of `java.applet` package.
- `java.applet` package is one of the smallest package in the entire JDK. It has only one class (Applet) and some interfaces.
- In the present version, `destroy()` is a deprecated method

NOTE: `paint()` is not a life cycle method.

Moreover `paint()` method is present in the Component class

Question: Develop a simple applet program which displays a button "Hello" and a textfield. when you click the button, your name should be displayed in the textfield.

```
import java.awt.*;  
import java.awt.event.*; HelloApplet.java  
import java.applet.*;  
  
public class HelloApplet extends Applet implements ActionListener  
{  
    Button b1;  
    TextField t1;  
    public void init ()  
    {  
        b1 = new Button ("Hello");  
        t1 = new TextField ();  
    }  
    public void start ()  
    {  
        add (b1);  
        add (t1);  
        b1.addActionListener (this);  
    }  
    public void actionPerformed (ActionEvent e)  
    {  
        t1.setText ("SANDEEP DEV");  
    }  
}
```

contd ... (P.T.O)

<html>

hello.html

<body>

<H3> This is my first applet program </H3>

<applet code = "HelloApplet1" width = "437" height = "440" >

</applet>

</body>

</html>

Question: can we define a constructor in an applet class?

Ans: Yes, we can define a constructor in an applet class provided, the constructor should be a no-argument constructor.

Image class:

- The Image class is used to represent the images.
- we cannot directly create object of the Image class because it is an abstract class.

∴ getImage() function (a non-static function of Applet class) creates ^{object} of the Image class, representing the contents of the gif file that we pass as an argument to the getImage() method.

NOTE: getImage() method takes two arguments,

(i) name of the gif file - a string class object

(ii) the path to the gif file in the form of a URL class object.

22.09.2006

To display the images on the browser,

- (i) create Image class object which represents the contents of the gif file
- (ii) display the Image object on the applet.

Hint: (i) get CodeBase () function which belongs to Applet class returns object of the URL class, which represents the path/url of the current applet class file.

(ii) get DocumentBase () function returns object of the URI class which represents the path/url of the current html file.

Ex: Image img = getImage (getCodeBase (), "tiger.gif");

- To display the image on the applet, we use a function by name drawImage (). It is a non-static function present in Graphics class.

- drawImage () method is defined to accept four arguments.

(i) Image class object representing the image (gif file)

(ii) X - co-ordinate value

(iii) Y - co-ordinate value

(iv) the object of the container on which the image has to be displayed.

- drawImage () method is a non-static method of Graphics class. So, to call drawImage () method we need object of Graphics class, but Graphics class is an abstract class.

Question: How can we get the object of Graphics class?

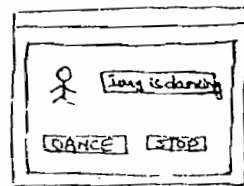
(P.T.O)

- paint() is a method present in the Component class and it is defined to accept object of Graphics class.

i.e. paint (new Graphics () { ... });

- An anonymous inner class extends the Graphics class and this returns the object of the Graphics class.
- Now, we can call drawImage() method on the object of the Graphics class.

Question: create an applet GUI as shown



- Requirement: Just display the image, buttons and textfield, on the applet.
Do not handle any events.

```
import java.awt.*;
```

```
import java.applet.*;
```

ImageApplet1.java

```
public class ImageApplet1 extends Applet
```

```
{
```

```
    Button b1, b2;
```

```
    TextField tf1;
```

```
    Image img;
```

```
    public void init ()
```

```
{
```

```
    b1 = new Button (" DANCE ");
```

```
    b2 = new Button (" STOP ");
```

```
    tf1 = new TextField ();
```

```
    img = getImage (getCodeBase (), " tiger.gif ");
```

```
}
```



```

public void start ( )
{
    add (b1);
    add (b2);
    add (t1);
}
public void paint (Graphics g)
{
    g.drawImage (img, 22, 40, this);
}
}

```

```

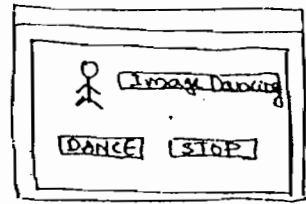
<!-- ImageDemo.html --> ImageDemo.html
<html>
<body>
  <H3> My First Image Application </H3> <BR>
  <applet code = "ImageApplet1" width = "437" height = "440">
  </applet>
</body>
</html>

```

In the above program, we used an image by name tiger.gif. we can use any of the image files with .gif extension, to be displayed on the applet.

NOTE: Before executing the above example, we should see that, ImageApplet1.java, ImageDemo.html, tiger.gif - all lie in the same path.

Question: Develop a GUI as shown.



Requirement: when we click the button

"DANCE", then the image should start dancing and in the text field it should display

"Image is dancing" on a green background.

when we click the button "STOP", then the image should stop dancing and in the text field it should display

"Image stopped dancing" on a red background.

Hints: - To meet the above requirement, first we need to have different postures of an image i.e. different gif files of the same image with slight differences between them.

- we have to make use of a non-static function `repaint()` of the Component class.
- The functionality of the `repaint()` method is to call the `paint()` method of the current object on which we call the `repaint()` method.

NOTE: To execute the `repaint()` method, the `paint()` method has to be called at least once.