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.

There are some important differences between an applet and a standalone Java application, including the following:

- An applet is a Java class that extends the Java.applet.Applet class.

- A main() method is not invoked on an applet, and an applet class will not define main().

- Applets are designed to be embedded within an HTML page.

- When a user views an HTML page that contains an applet, the code for the applet is downloaded to the user's machine.

- A JVM is required to view an applet. The JVM can be either a plug-in of the Web browser or a separate runtime environment.

Unlike stand-alone applications, applet cannot run independently. They are run from inside a web page using a special feature known as html tag.

Although both applets and standalone applications are Java programs but applets are not full featured application programs. Applets are small restricted Java programs that are primarily used in Internet computing.

Other Advantages of Applet

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

- It works at client side so less response time.

- Secured

- It can be executed by browsers running under many platforms, including Linux, Windows, Mac Os etc.

Drawback of Applet

- Plug-in is required at client browser to execute applet

Hierarchy of Applet

Lifecycle of Java Applet methods:

1. Applet is initialized.

2. Applet is started.

3. Applet is painted.

4. Applet is stopped.

5. Applet is destroyed.

Java.applet.Applet class

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

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

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

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

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

Following is the schematic representation of the methods.

3. paint(): This method takes a Java.awt.Graphics object as parameter. This class includes many methods of drawing necessary to draw on the applet window. This is the place where the programmer can write his code of what he expects from applet like animation etc. This is equivalent to runnable state of thread. This paint method belongs to AWT component class.

Note:

Java.awt.Component class

The Component class provides 1 life cycle method of applet.

1. 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

1. By html file

2. By appletViewer tool (for testing purpose).

Java repaint() – Call paint()

The paint() method is called by the JVM implicitly in two circumstances – one when the first time frame is created and displayed and the other when the frame is resized (by dragging the frame border with mouse) by the user. If the programmer would like to call the paint() method, to draw some graphics on the frame or applet window in the middle of the coding, he is permitted to call repaint() method and not paint() method directly

What is the designing secret in not allowing paint() to call directly?

Yes, calling paint() method raises compilation error. Before the window is to be drawn with new data, the old data must be erased, else, both overwrite (not override) the other and finally the data is not readable. This automatic erasing is done by the repaint() method. How?

The repaint() method calls automatically update() method and in turn update() method calls paint() method

repaint() –> update() –> paint()

The update() method job is to erase the earlier drawings on the frame. On the cleared surface, the paint() method draws afresh with the latest graphics. If the programmer is allowed to call paint() method directly, there is every possibility that he may forget to call the update() method explicitly. To avoid this and to make Java as a robust language, the designers do not allow to call paint() directly.

Note: i) Other difference is calling paint() method in itself may cause infinite looping.

GUI in applet

GUI in applet is same as that of GUI in window application except

i) When paint() method is called object of class containing paint() method is created automatically. So, no any main method having object of same class is needed

ii) Instead of constructor use init() method.

iii) paint() method provides canvas like Frame or JFrame class in window programs but with limitations of drawings. So use Frame or JFrame class to utilize full features of AWT and SWING.i.e. no any need to use Frame or Jframe if using AWT graphics in applet(see example of adding two numbers with SWING and AWT

iv) Other methods like actionPerformed(), mouseMotion() etc and interfaces or adaptor classes are used same way.

v) Most of the task in applet is of repeating nature i.e. repainting. So, may usemultithreading (if possible) feature.