

Applet and Graphics Programming

Applet Programming

- Applets are small Java programs that are primarily used in Internet computing.
- They can be transported over the Internet from one computer to another and run using the Applet Viewer or any Web browser that supports Java.
- An applet, like any application program, can do many things for us. It can perform arithmetic operations, display graphics, play sounds, accept user input, create animation, and play interactive games.

Local and Remote Applets

- An applet developed locally and stored in a local system is known as a local applet.
- A remote applet is that which is developed by someone else and stored on a remote computer connected to the Internet.
- In order to locate and load a remote applet, we must know the applet's address on the Web. This address is known as Uniform Resource Locator(URL).

How Applets differ from Applications

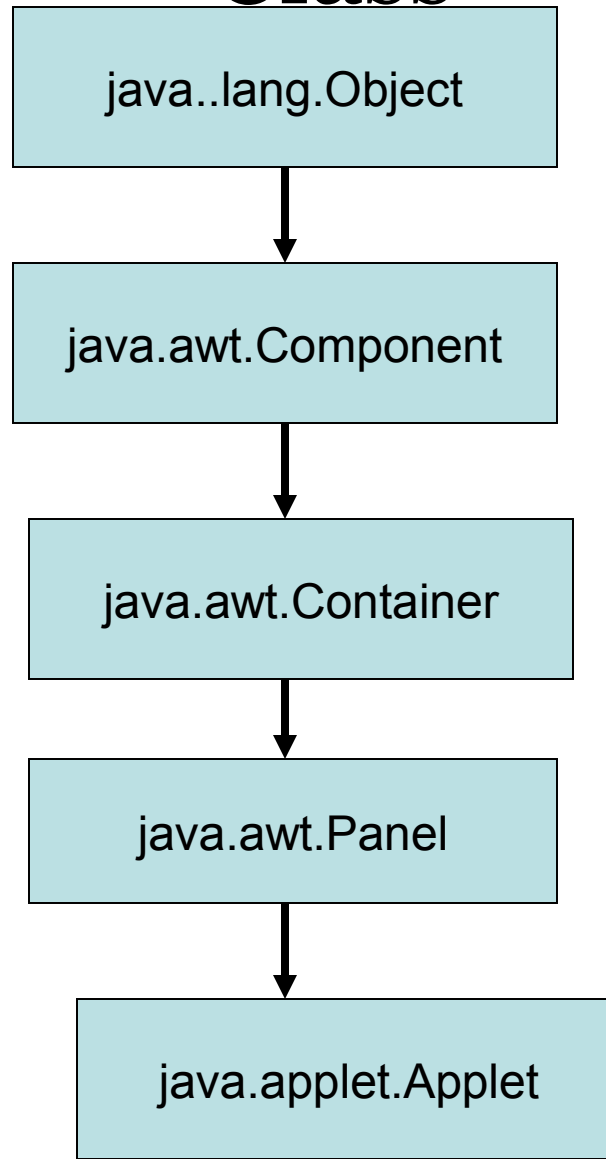
- Applets do not use the `main()` methods for initiating the execution of the code.
- Unlike stand-alone applications, applets cannot be run independently. They are run from inside a Web page using a special feature known as HTML tag.
- Applets cannot read from or write to the files in the local computer.
- Applets cannot communicate with other servers on the network.
- Applets cannot run any program the local computer.
- Applets are restricted from using libraries from other languages such as C or C++.

To Write Applets

- Building an applet code (.java file)
- Creating an executable applet (*.class file)
- Designing a Web page using HTML tags
- Preparing <APPLET> tag
- Incorporating <APPLET> tag into the Web page
- Creating HTML file
- Testing the applet code

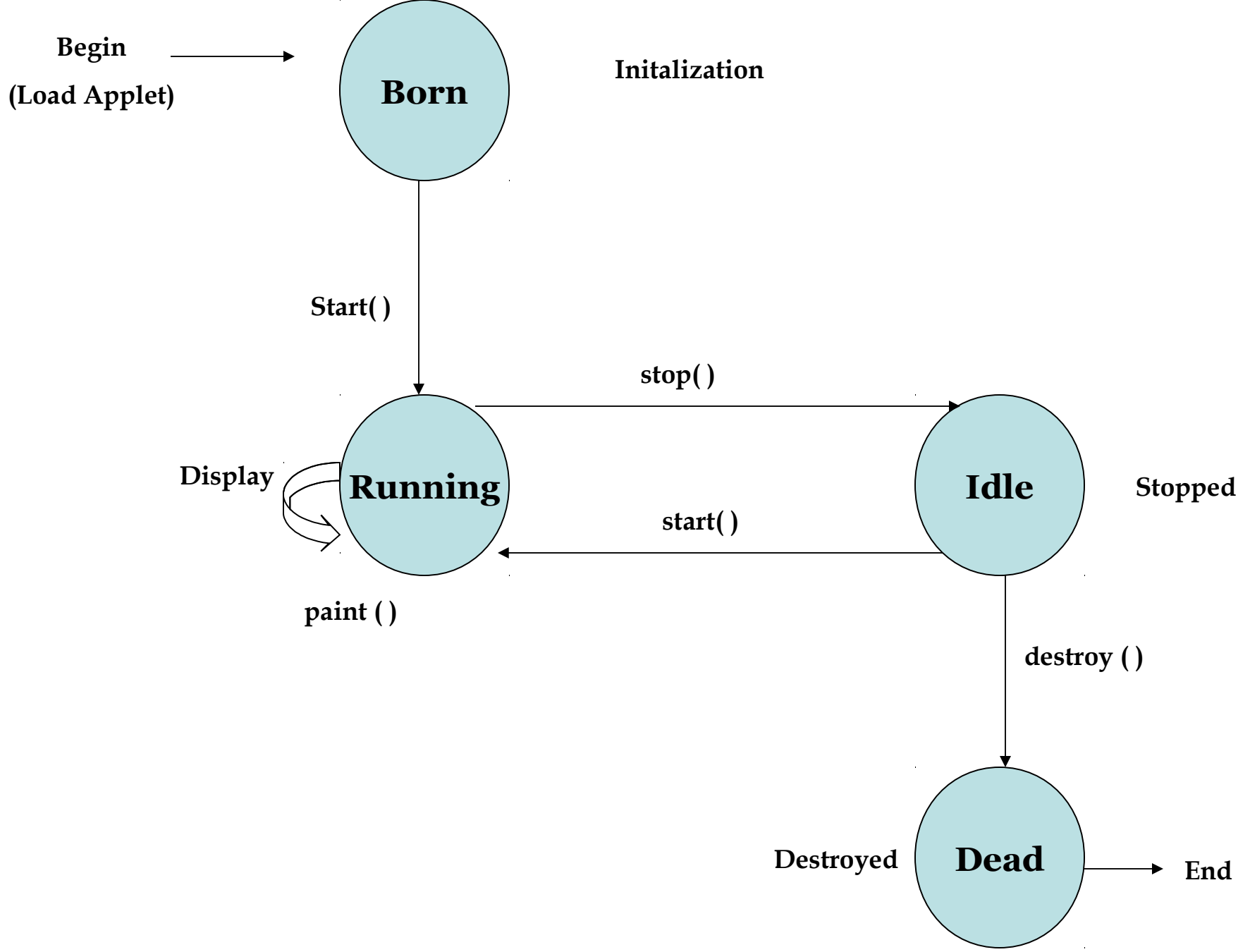
Chain of classes inherited by Applet

Class



Applet Life Cycle

- The applet states include:
 - Born or initialization state
 - Running state
 - Idle state
 - Dead or destroyed state



An applet's state transition diagram

Applet Life Cycle

- Initialization State
 - Applet enters the initialization state when it is first loaded. This is achieved by calling the `init()` method of Applet Class.
 - The applet is born.
 - The initialization occurs only once in the applets' life cycle.
 - To provide any of the behaviours mentioned above, we must override the `init()` method:

```
public void init()  
{  
    //statements  
}
```

Applet Life Cycle

- Running State

- Applet enters the running state when the system calls the start () method of Applet Class.
- This occurs automatically after the applet is initialized.
- Starting can also occur if the applet is already in “stopped” (Idle) state.
- Unlike init() method, the start () method may be called more than once.

```
public void start()  
{  
//      statements  
}
```

Applet Life Cycle

- Idle or Stopped State
 - An applet becomes idle when it is stopped from running. Stopping occurs automatically when we leave the page containing the currently running applet.

```
public void stop( )  
{  
    //statements  
}
```

Applet Life Cycle

- Dead State
 - An applet is said to be dead when it is removed from memory. This occurs automatically by invoking the destroy () method when we quit the browser.
 - Like initialization, destroying stage occurs only once in the applet's life cycle.

```
public void destroy( )  
{  
    //statements  
}
```

Applet Life Cycle

- Display state
 - Applet moves to the display state whenever it has to perform some output operations on the screen.
 - This happens immediately after the applet enters into the running state.
 - The paint() method is called to accomplish this task.
 - Almost every applet will have a paint() method.
 - Paint() method is inherited from the Component class , a super class of Applet.

```
public void paint (Graphics g)
{
    //statements
}
```

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

```
public class HelloJava extends Applet  
{  
    public void paint(Graphics g)  
    {  
        g.drawString("Hello Java",10,100);  
    }  
}
```

```
<html>  
<body>  
<applet code = "HelloJava.class"  
height="100" width = "200">  
</applet>  
</body></html>
```

Applet Tag

- The <Applet. .>tag supplies the name of the applet to be loaded and tells the browser how much space the applet requires.
- <APPLET> tag specifies three things:
 - Same of the applet
 - Width of the applet (in pixels)
 - Height of the applet (in pixels)

Passing Parameters to Applets

- We can supply user-defined parameters to an applet using `<PARAM...>` tags.
- Each `<PARAM...>` tag has a name attribute such as color and a value attribute such as red.
- Inside the applet code, the applet can refer to that parameter by name to find its value.
- To set up and handle parameters, we need to do two things:
 - Include appropriate `<PARAM...>` tags in the HTML document
 - Provide Code in the applet to parse these parameters.
- Parameters are passed to an applet when it is loaded. We can define the `init()` method in the applet to get hold of the parameters defined in the `<PARAM>` tags.
- This is done using the `getParameter()` method, which takes one string argument representing the name of the parameter and returns a string containing the value of the parameter.


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

public class JavaParam extends Applet
{
    String s;
    public void init()
    {
        s =
getParameter("uname");
        if(s==null)
            s = "ABC";
        s = "Hello" + s;
    }
    public void paint(Graphics g)
    {
        g.drawString(s,10,100);
    }
}
```

```
<html>
<body>
    <APPLE CODE =
JavaParam.class
width=400 height = 200>

    <PARAM NAME = "uname"
VALUE="ruchita">
</APPLE>
</body>
</html>
```