Java Applets

Dave Price
Computer Science Department
University of Wales, Aberystwyth

Java Applets

- An **applet** is a special type of Java program designed to run inside a web browser or an applet viewer.
- Applets are one way to create “live” web pages.
- Applets are constrained in what they are allowed to do. Remember you may be downloading and executing “untrusted” code on your computer.

Java Applets

- Applets are created by extending the provided class called Applet from the package Java.applet.
- The class Applet itself extends the AWT class called Panel.
- An applet does not contain a “main” method

Applet inheritence hierarchy

java.lang.Object
|
+--**java.awt.Component**
|
+--**java.awt.Container**
|
+--**java.awt.Panel**
|
+--**java.applet.Applet**

The methods of the Applet class

- Applet provides directly or via inheritance from Panel etc a set of methods which have a key role.
- The methods are invoked when the browser or appletviewer does certain things which affect the environment of the Applet.
- The environment changes involve things like starting the applet, returning to the applet's web page, moving away from the applets web page and so on.

Applet Methods

- It's normal for your applet to override many of the provided methods of Applet.
- The provided methods are often "dummy" and do not really do anything.
- You need to understand exactly which methods will get called when.
- Your methods need to return fairly quickly, if you have a "continuous" algorithm to run start a new thread to do the work.

Important Methods

- **void init()**
  - Called by the browser or applet viewer to inform this applet that it has been loaded into the system.
- **void destroy()**
  - Called by the browser or applet viewer to inform this applet that it is being reclaimed and that it should destroy any resources that it has allocated.

Important Methods (2)

- **start()**
  - Called by the browser or applet viewer to inform this applet that it should start its execution. **NOTE this may be called many times !, e.g. every time a window is resized, when a page is re-visited and so on.**
- **stop()**
  - Called by the browser or applet viewer to inform this applet that it should stop its execution. **NOTE this may be called many times !, e.g. every time a window is resized, when a page is left.**

Important Methods (3)

- **URL getCodeBase()**
  - Gets the base URL from which the applet class file was loaded.
- **URL getDocumentBase()**
  - Gets the document URL of the HTML file that refers to the applet.
- note: URL is java.net.URL

Important Methods (4)

- **String getParameter(String name)**
  - Returns the value of the named parameter in the HTML tag.
- **void showStatus(String msg)**
  - Requests that the argument string be displayed in the "status window".

Fun Methods

- **void play(URL url, String name)**
  - Plays the audio clip given the URL and a specifier that is relative to it.

A suitable Web Page

```
<html><head><title>Dave Price's Applet test page
</title></head><body>
<h1>Dave Price's Applet test page.</h1>
<HR>
<applet code="NewApplet.class" width=180
  height=180>
</applet>
<HR>
</body></html>
```

Remember

- My applet extends Applet
- Applet extends Panel
- Panel has a flowLayout manager by default
- I can use the add() method to add things to a panel
- Label instances are simple areas with text in

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.net.URL;

public class NewApplet extends
 Applet {

    public void init (){
        add(new Label("hello"));
    }
```

```
  public void start(){
    add(new Label("Start called"));
    showStatus("Dave's applet's
                    starting");
    URL docBase = getDocumentBase();
    play(docBase, "test.au");
  }
  public void stop(){
    add(new Label("Stop called"));
  }
}
```

A demonstration

- **Go and select**
- **http://www.aber.ac.uk/~dap/Java/newtestit.html**

Another Example

- If you need to load multiple classes for your applet, it's more efficient to plave them all in a "jar" file and let them load together.
- this uses another parameter in the applet line of your html file
- **`archive="MyApplet.jar"`**
- together with a code= parameter to identify the starting class

```
<html><head><title>Dave Price's
Applet test page
</title></head><body>
<h1>Dave Price's Applet test
page.</h1>
<HR>
<applet archive="MyApplet.jar"
code="MyApplet.class" width=180
height=180>
</applet>
<HR>
</body></html>
```

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

public class MyApplet extends Applet {

 public void init (){
  Button myButton [][] = new Button[3][3];
   for(int i=0;i<=2;i++)
    for(int j=0;j<=2;j++)
      myButton[i][j] = new Button("empty");
  setLayout(new GridLayout(3,3));
  MyHandler mh = new MyHandler();
```

```
  for(int i=0;i<=2;i++)
   for(int j=0;j<=2;j++){
     add(myButton[i][j]);
     myButton[i][j].setLabel("");
     myButton[i][j].addActionListener(mh);
     }
  setVisible(true);
 }
}
```

```
class MyHandler implements ActionListener {

  String next_symbol = "X";

  public void actionPerformed(ActionEvent e){

    Button button = ((Button) e.getSource());

    if ( button.getLabel().equals("") ) {
      button.setLabel(next_symbol);
      if (next_symbol.equals("X") )
            next_symbol = "Y";
      else next_symbol = "X";
    } else button.setLabel("Already Set");
    }

}
```

A demonstration

- **Go and select**
- **http://www.aber.ac.uk/~dap/Java/testit_jar.html**