


*Core Java Course
Applets (Part 1)
(Chapter 10)*

Dr. Jeff Blessing
MSOE


1



Agenda

- ❖ Homework
- ❖ Why Applets?
- ❖ Applet Security
- ❖ Applets vs. Applications
- ❖ Applet Lifecycle
- ❖ Homework Assignment


2



Why Applets?

- ❖ An **Applet** is a special kind of Java program that can be downloaded and run within a Java-enabled browser
- ❖ Using Applets
 - is fueled mostly by Web development
 - takes distinct advantage of multi-platform portability
 - provides adequate security
 - gives much better UI capabilities than HTML
 - allows for local data processing, saving network traffic
 - gives access to advanced features like audio and video


3



The Java Plug-in

- ❖ Browser vendors have been behind in supporting the latest features of the JDK
 - ❖ Netscape (AOL) is closely aligned with Sun (and Mozilla.org remains)
 - ❖ Microsoft has dropped J++, and has never implemented the parts of the JDK that directly compete with Microsoft's platform
 - ❖ So Sun has developed the *Java Plug-in*
 - ❖ which is required to run the examples in Chapter 10 in a browser like Navigator or Internet Explorer
 - ❖ use the HTML converter to convert <applet> tags

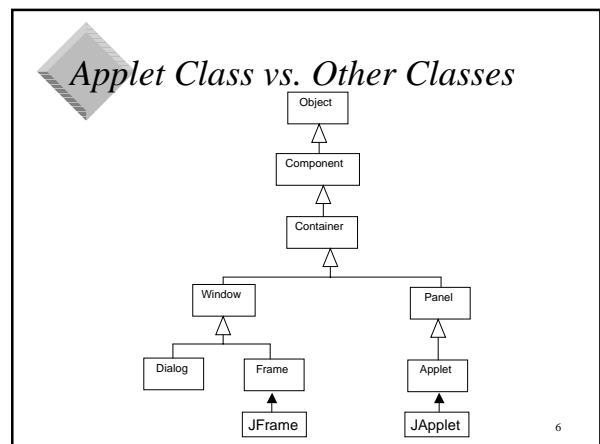
4



What Applets can Do?

- ❖ Show (animated) images
- ❖ Play sounds
- ❖ Receive user input events (mouse clicks/moves, keystrokes)
- ❖ Send user data to the host the applet was downloaded from (originating host)
- ❖ Receive/display new information from the host
- ❖ Pop up new windows

5



What Applets in a Browser cannot do (“Sandbox” approach)

- ❖ Start an executable on the local machine, call `exit()` method, load native code libraries
- ❖ Connect/listen to any network server other than originating host (ports < 1024)
- ❖ Access any local filesystem resources
- ❖ Access any information about local computer, its operating system (except for the Java version, ...)
- ❖ Start a print job or access the system clipboard
- ❖ An attempt to violate restrictions while executing applet code causes `SecurityException`

7

Default Java Access Levels

Table 10-1 (page 542)

- ❖ Application
 - ❖ No restrictions
- ❖ Applet in an applet viewer
 - ❖ Cannot delete files, otherwise the same as application
- ❖ Applet in a browser* loaded as a local file
 - ❖ Networking (to any host), access to Java library, access to `user.name` attribute, unrestricted new windows
- ❖ Applet in a browser loaded from any URL
 - ❖ Connecting to the origination host only, new windows carry a warning message

8

Outside of the “sandbox”

- ❖ All the restrictions above imply that the local system has a default Security Manager enabled
- ❖ Using `SecurityManager` class, some security restrictions can be relaxed
- ❖ Applet signing technique allows identification of trusted applets/applet sources that require fewer security restrictions
- ❖ Applets loaded from within an Intranet may be granted additional privileges

9

HTML File for an Applet

- ❖ Applets run within an HTML page


```
<HTML>
<TITLE>Applet Name</TITLE>
<BODY>
<P>Any text can be added here</P>
<APPLET CODE="AppletFile.class" WIDTH=100
  HIGHT=100>Text to display for non-Java browsers
</APPLET>
<P>Any text can be added here</P>
</BODY>
</HTML>
```

10

HTML File for an Applet (cont.)

- ❖ Other useful/optional `<APPLET>` tag attributes
 - ❖ `CODEBASE` - an URL (directory) for the applet
 - ❖ `ARCHIVE` - a Java archive file containing other classes
 - ❖ `OBJECT` - a file containing serialized applet object
 - ❖ `ALT` - text to be displayed if browser doesn't support Java
 - ❖ `ALIGN` - possible values `left`, `right`, `top`, etc...
 - ❖ `VSPACE` - the margin in pixels above and below the applet
 - ❖ `HSPACE` - the margin in pixels on either side of the applet

11

HTML File for an Applet (cont.)

- ❖ The truth about HTML tags for applets:
 - ❖ The `EMBED` tag is understood by Netscape Navigator
 - ❖ The `OBJECT` tag is understood by Internet Explorer
 - ❖ The `APPLET` tag is understood by Sun's JDK and the HotJava browser
 - ❖ The WWW specification for HTML 4 has replaced (deprecated) the `APPLET` tag with the `OBJECT` tag
 - ❖ Sun is updating the JDK and HotJava to support this
 - ❖ Until then, use the `APPLET` tag and run your HTML file through Sun's Java Plug-in HTML Converter

12

Applet vs. Application (similarities)

- ❖ Applet class is derived from Container and Component classes
- ❖ The same Graphics context object to draw on
- ❖ The same `paint()`/`repaint()` methods to handle imaging
- ❖ The same user interface building approach
 - ❖ Using same AWT components
 - ❖ Using Layout Managers
- ❖ The same event model
 - ❖ Using event sources and listeners

13

Applet vs. Application (cont.)

vs. JApplet

- ❖ Import `java.applet.*` package
- ❖ Derive from `Applet` (`JApplet`) rather than `Frame` (`JFrame`) class
- ❖ No `main()` method
 - ❖ Browser takes care of creating and displaying an applet
- ❖ No `setSize()` and `setLocation()` methods
 - ❖ Size is set by `HEIGHT` and `WIDTH` fields in HTML file
 - ❖ Location is subject to browser's own layout control
- ❖ Do not override constructor, use `init()` instead
 - ❖ The `init()` method is called by the browser

14

Applet vs. Application (cont.)

- ❖ Do not rely on the implicit `BorderLayout`, set it explicitly in the `init()`
- ❖ No `setTitle()` method
 - ❖ Use HTML `<TITLE>` tag instead
- ❖ No menus
 - ❖ Use buttons or other components instead
- ❖ Calculator Applet Conversion Code Example

15

Living in a Browser: Applet Lifecycle

- ❖ A browser manages applets by calling the following public methods: `init()`, `start()`, `stop()`, and `destroy()`
- ❖ Method `init()`
 - ❖ automatically called after an applet is created
 - ❖ used for initialization and setting up UI components
 - ❖ used for processing `PARAM` values (an alternative to the command line parameters not available to applets)
 - ❖ used for code that needs to be executed *only once* in applet's lifetime

16

Applet Lifecycle (cont.)


- ❖ Method `start()`
 - ❖ first time called right after the `init()` method
 - ❖ also called every time the applet becomes visible
 - ❖ used for the functionality that needs to be restarted when the applet becomes visible (playing sound, animation)
- ❖ Method `stop()`
 - ❖ called when applet becomes *temporarily* invisible
 - ❖ called to let the applet know that it should stop some resource consuming activities useless when the applet is invisible (e.g. animation)

17

Applet Lifecycle (cont.)

- ❖ Method `destroy()`
 - ❖ called when the applet is about to be unloaded
 - ❖ used for releasing any resources other than memory allocated by the applet
- ❖ There is also `paint()` method that is invoked by the browser when the applet is displayed and then whenever the applet moves on the screen or its exposed (visible) area changes
- ❖ More Code Examples


18



Homework Assignment

- ❖ Convert your favorite stand-alone Java Application into an Applet

19



20