

**Elementos de Programación OO
con el Lenguaje Java**

**Introducción a
los Java Applets**

Con base a los conceptos del curso de Sun Microsystems
“Lenguaje de Programación Java”
(Java Programming Language. SL275).

Luis Ernesto Rubio.

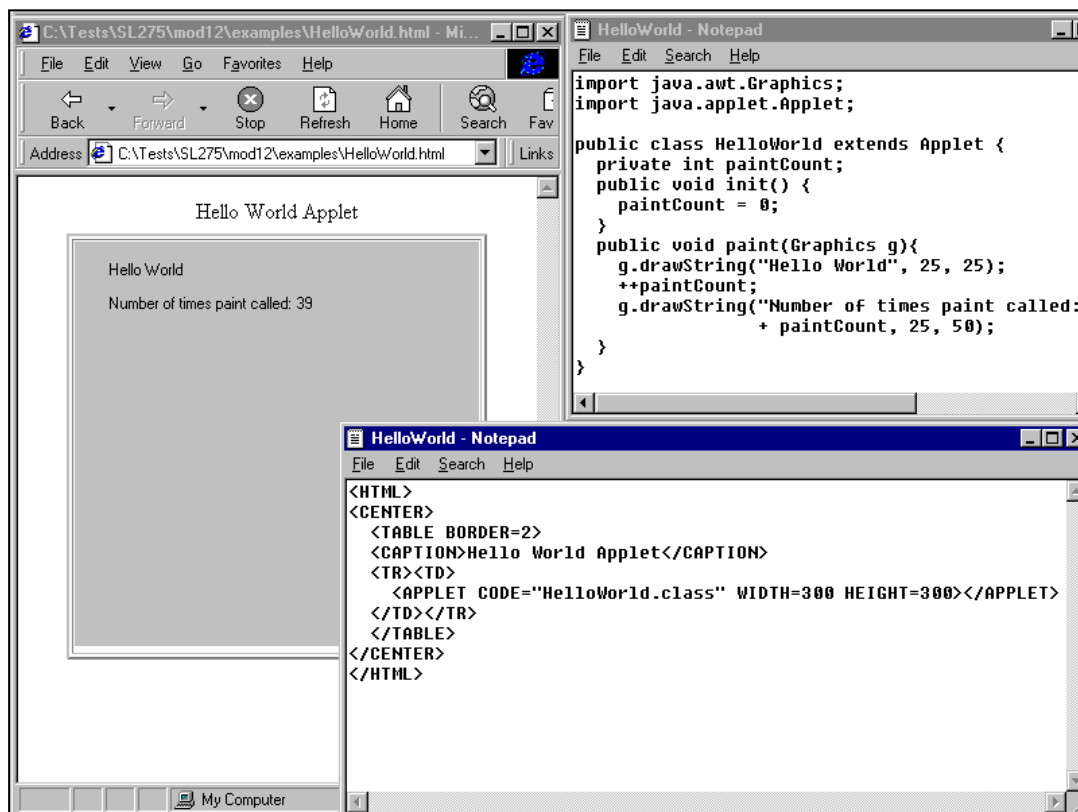
Los Applets de Java.

Los applets son pequeños programas escritos en el lenguaje de programación Java, que se incrustan en páginas Web.

Cuando los usuarios entran a dichas páginas, los applets son cargados en sus computadoras y ejecutados. En lugar de que los procesos sucedan en el servidor como en el caso de la programación CGI, en Java, suceden en el cliente en un Web browser compatible con Java.

Esto significa que los applets no están restringidos por el ancho de banda de la red o la velocidad de acceso de módems cuando se ejecutan. Los usuarios pueden ver y escuchar efectos de multimedia en páginas Web de manera más eficiente y efectiva.

Dado que Java es un lenguaje eficiente, los applets que ofrecen interacción con el usuario generalmente operan a la misma velocidad que las otras aplicaciones del escritorio del usuario, tales como los programas procesadores de textos.



La API Applet.

Las clases en la API Applet están definidas en el paquete `java.*`. Los applets contruidos usando estas clases podrán ejecutarse en cualquier ambiente que soporte Java.

Ciclo de Vida.

El ciclo de vida de un applet comprende una serie de fases, desde el momento en que es cargado, hasta cuando ya no es accesible y reclamado por el sistema.

Cada fase en el ciclo de vida de un applet esta determinada por la llamada a un método del applet, de modo que el applet pueda tomar las acciones adecuadas. No se requiere invocar estos directamente, debido a que el sistema a tiempo de ejecución los invocará cuando sea necesario.

Por ejemplo, cuando un applet se activa dentro de un página Web mediante un Web browser, y se abandona esa página para ir a otra, el browser invocará el método `stop` en el applet.

A continuación se describen los principales métodos en un applet y su propósito:

`void init()`

Se llama una vez que un applet ha sido cargado en el sistema. Es un buen sitio para efectuar cualquier inicialización.

`void start()`

Se invoca para informar al applet que debe efectuar alguna acción para comenzar su ejecución. Por ejemplo applets que utilicen threads para animación podrán inicializar aquí los threads. Este método se llama después de `init` y siempre que el applet es revisitado en una página web.

`void stop()`

Se invoca para informar al applet que debe terminar su ejecución. Para un applet que utiliza threads, este método puede utilizarse para detener los threads. Típicamente es invocado cuando la página Web que contiene el applet es reemplazada por otra página.

void destroy()

Invocado para indicar al applet que está siendo reclamado por el sistema y que debe destruir cualquier recurso del que haya dispuesto. Un applet que use threads puede destruirlos en este método. Nótese que stop siempre será invocado antes que destroy.

No se requiere que todo applet redefina todos estos métodos, sino únicamente los necesarios. Si un applet utiliza threads ó almacena recursos que deban ser liberados, es necesario sobredefinir los métodos apropiados del ciclo de vida a fin de obtener un applet de comportamiento adecuado.

Métodos de etiqueta HTML.

Los siguientes métodos se utilizan para obtener información acerca del applet, tal como el URL del documento en el cual está incrustado ó los parámetros que le son enviados al applet a través de la etiqueta HTML.

URL getDocumentBase()

Devuelve el URL del documento que contiene el applet. La clase URL devuelta está en el paquete java.net, y tiene una serie de métodos que permiten manipular y leer el contenido del URL.

URL getCode Base()

Devuelve el URL del applet mismo, que puede diferir del URL en donde está el documento.

String getParameter (String name)

Devuelve el valor de un parámetro nombrado en la etiqueta HTML. Por ejemplo una etiqueta applet se especifica como:

```
<applet code = "Fun.class" width = 50 height = 50>  
<param name= Color value ="red">  
</applet>
```

Llamando a getParameter("Color") para el anterior applet, se devolverá el valor "red". Si el parámetro no se especifica en la etiqueta HTML, entonces este método devuelve el valor null.

La clase URL en java.net permite especificar un URL y además obtener el contenido de donde el URL apunta. Son muchos los métodos que componen a la clase URL , por lo cual sólo se describirán algunas de las formas más comunes para el almacenamiento de objetos URL.

La forma más simple para especificar un URL es inicializarlo mediante una cadena:

```
URL someURL = new URL ("http://www.javasoft.com/");
```

Esto crea un URL absoluto, debido a que está completamente especificado en la cadena. Para muchos de los métodos applet, es deseable especificar un URL relativo al URL del applet.

Para hacer esto, se utiliza simplemente un constructor diferente que toma un URL como argumento y una especificación URL que es relativa al URL dado. por ejemplo, se desea conformar un URL que apunte a images/Smiley.gif se escribe lo siguiente:

```
URL docuementURL = new URL ("http://www.javasoft.com/");  
URL imageURL = new URL (docuementURL, " images/Smiley.gif ");
```

Soporte de Medios.

La clase applet incluye métodos para obtener imágenes y sonidos desde URLs, que facilitan el uso de este tipo de medios en los applets. En el caso de las imágenes, la conversión de formatos desde GIF o JPEG se hace automáticamente.

Image getImage(URL url)

Devuelve un objeto imagen que puede ser dibujado en la pantalla. El URL que se da como argumento, debe ser un URL absoluto. Este método retornará inmediatamente exista ó no la imagen. Cuando se hace un intento de dibujar la imagen, los datos serán cargados. Las primitivas gráficas que dibujan la imagen pintarán paso a paso una imagen en la pantalla.

Image getImage (URL url, String name)

Toma una imagen de un URL dado y de un especificador URL que es relativo a éste.

AudioClip getAudioClip (URL url)

Toma los datos de audio de donde apunta el URL dado y devuelve un objeto de la clase AudioClip, que permite tocar el clip de sonido.

AudioClip getAudioClip(URL url, String name)

Toma una audio clip de un URL y un especificador URL que es relativo a éste.

void play(URL url)

Toca un audio clip directamente de su URL indicado.

void play(URL url, String name)

Toca un audio clip directamente de su URL indicado y de un especificador URL que es relativo a éste.

La clase AudioClip es una abstracción muy simple que permite tocar clips de sonido. Pueden tocarse al mismo tiempo múltiples objetos AudioClip, y los sonidos se combinarán para producir una composición. Esta clase tiene únicamente tres métodos:

void play() Comienza a tocar el clip de sonido.

void loop() Toca el clip de sonido de forma cíclica.

void stop() Detiene la ejecución de un clip de sonido.

Manipulación del Ambiente del Applet.

Algunos applets deberán afectar el ambiente sobre el cual se están ejecutando. Por ejemplo, si un applet corre sobre un Web browser, puede requerir reemplazar la página Web que el browser está visualizando.

La API applet presenta un soporte limitado para esta característica, debido a que no puede asumirse que el applet se está ejecutando en un Web browser.

Aún si el applet se está ejecutando en un browser compatible con Java, el browser puede tener características específicas que lo diferencian de otros, por lo que una interface unificada puede no ser factible.

Es necesario usar los métodos de la clase AppletContext para aplicar esta necesaria limitante de la API applet.

AppletContext getAppletContext()

Devuelve una instancia de la clase AppletContext, que permite al applet afectar el ambiente de forma limitada.

Applet getApplet(String name)

Devuelve un applet si se da el nombre de éste. El nombre puede indicarse en la etiqueta HTML mediante el atributo name. Si un applet con este nombre no existe en la página actual, entonces este método devuelve null.

Enumeration getApplets()

Devuelve todos los applets de la página activa (sujeto a consideraciones de seguridad).

void showDocument(URL url)

Reemplaza la página Web activa con el URL dado. Este método puede ser ignorado si la aplicación que está accediendo al applet no es un Web browser.

Información del Applet.

Algunos de los métodos applet se han diseñado para dar información acerca de un applet ó mostrar mensajes en la pantalla relativos a éste.

void showStatus (String status)

Despliega una cadena de status en la pantalla.

String AppletInfo()

Cuando se sobredifine, devuelve una cadena informativa acerca del applet, tal como su autor, versión ó copyright.

String[][] getParameterInfo()

Cuando se sobredifine devuelve un arreglo que describe los parámetros que el applet en cuestión sabe cómo procesar.

Este tipo de información puede ser útil para determinar que parámetros son útiles para un applet, ó en un futuro para permitir la configuración de un applet gráficamente. Cada elemento en el arreglo es un arreglo de tres cadenas que tiene la siguiente forma:

{name, type, comment}

Enseguida se ilustra n ejemplo que define este método para un applet que permite un parámetro Color, el cual será el color que el applet utilice para dibujar.

```
public String[][] getParameterInfo() {  
    Stringinfo[][] = {{"Color", "String", "foreground color"}};  
    return info;  
}
```

Consideraciones de Seguridad.

Los applets se han diseñado para ejecutarse dentro de una aplicación, tal como un Web browser. Este contenido ejecutable puede tomarse desde la red, lo que significa que código posiblemente inseguro corra dentro de la aplicación.

Java y la API applet presentan mecanismos de salvaguarda que minimizan el riesgo de correr applets, pero estas salvaguardas implican que los programadores de applets se verán limitados en lo que puedan hacer.

El modelo de seguridad para los applets Java considera a los applets como código inseguro ejecutándose dentro de un ambiente confiable.

Por ejemplo, cuando se instala una copia de un Web browser en una máquina, se confía en que éste será un código apropiado para correr en el ambiente de esta máquina.

Típicamente los usuarios son (o deberían de) ser cuidadosos con los programas que instalan en sus máquinas, especialmente cuando los programas provienen de algún lugar de la red. Un applet, por otro lado, es cargado desde la red, sin asumir nada acerca de su confiabilidad.

El lenguaje Java y el applet runtime están escritos para proteger al cliente en contra de applets inseguros. Estas salvaguardas se implementan verificando que el bytecode de las clases de los applets no rompan las reglas básicas del lenguaje, ni las restricciones de acceso dentro del runtime.

Sólo cuando estos requisitos han sido satisfechos se permite al código del applet ser ejecutado. Cuando se ejecuta, el applet se etiqueta como un applet dentro del intérprete Java. Esta etiqueta permite a las clases a tiempo de ejecución determinar cuando a una pieza de código puede permitírsele invocar un cierto método.

Por ejemplo, un applet está restringido en el conjunto de hosts para los cuales puede abrir una conexión de red o en el grupo de URLs a los que puede tener acceso. Juntas este conjunto de restricciones constituyen una póliza de seguro.

La actual póliza de seguro afecta las características que un applet puede usar:

Los applets tendrán restringido el acceso a archivos. De forma particular, escribir archivos y posiblemente incluso leer archivos no será una característica estándar que pueda ser soportada en todos los browsers compatibles con Java.

Las conexiones de red se restringirán a sólo la conexión con el host del cual el applet proviene. Para URLs, sólo aquellos que apunten al host del applet pueden ser abiertos.

Un applet no será capaz de usar ningún método que pueda resultar en la ejecución arbitraria de código de comportamiento no verificado. Esto incluye métodos que corran programas arbitrarios, como aquellos métodos que carguen librerías dinámicas.