

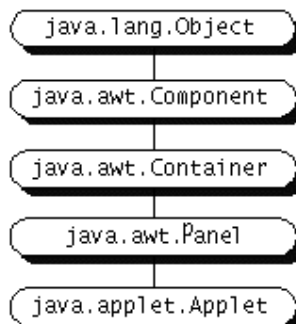
Applets

- **Un applet es:**
 - desde el punto de vista de código Java, **un programa diseñado para ser ejecutado en el contexto de una página web.**
 - desde el punto de vista de una página html, **una pequeña ventana gráfica dentro de la misma.**
- Diferencia con una aplicación Java:
 - Para cargar y ejecutar una aplicación Java utilizamos directamente el intérprete Java.
 - Un *applet* se carga y se ejecuta desde cualquier explorador que soporte Java.

1

Applets

- Los *applets* se implementan creando una subclase de la clase *Applet*



2

Applets

- Applet “Hola Mundo”

```
import java.applet.Applet;
import java.awt.Graphics;

public class HolaMundo extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hola mundo!", 50, 25);
    }
}
```

3

Applets

- Cómo ejecutar un applet

```
<HTML>
<HEAD>
<TITLE> Primer programa </TITLE>
</HEAD>
<BODY>
```

Salida del programa (Applet):

```
<APPLET CODE="HolaMundo.class" WIDTH=150 HEIGHT=25>
</APPLET>
</BODY>
</HTML>
```

4

Ciclo de vida de un applet

1. Carga del applet
 - Se crea una instancia de la clase del applet
 - El applet se *inicializa* (*init*)
 - El applet *comienza* su ejecución (*start*)
2. Abandonar y volver a la página del applet
 - Cuando el usuario abandona la página, el applet puede *pararse* a sí mismo (*stop*)
 - Cuando el usuario vuelve a la página, el applet *comenzará* de nuevo (*start*)

5

Applets: métodos básicos

- Métodos para todo lo anterior:

```
public class Simple extends Applet {  
    ...  
    public void init() { ... }  
    public void start() { ... }  
    public void stop() { ... }  
    public void destroy() { ... }  
    ...  
}
```

6

Applets: métodos básicos

- Para procesar los eventos anteriores, la mayoría de applets redefinen estos 4 métodos:
 - *init*
 - Para inicializar el applet cada vez que se carga o se recarga
 - Suele usarse para operaciones cortas de inicialización (por ejemplo, cargar imágenes)
 - *start*
 - Para comenzar la ejecución del applet: cuando el applet se carga o cuando se revisita la página web
 - *stop*
 - Para detener la ejecución del applet: cuando se abandona la página web o se cierra el explorador
 - *destroy*
 - Para realizar operaciones de limpieza (liberar recursos) antes de descargar el applet

7

Applets: métodos básicos

- Para visualizar objetos, los applets pueden redefinir 2 métodos heredados de AWT:
 - *paint*
 - Es el método básico de visualización. Se usa para dibujar la representación del applet dentro de la página web.
 - *update*
 - Se usa para redibujar el applet

8

Applets: métodos básicos

- Para visualizar objetos, los applets pueden redefinir 2 métodos heredados de AWT:
 - *paint*
 - Es el método básico de visualización. Se usa para dibujar la representación del applet dentro de la página web.
 - *update*
 - Se usa para redibujar el applet

9

Applets: métodos básicos

- Para reaccionar ante eventos, los applets pueden redefinirse:
 - Métodos específicos de manejo del evento (ejemplo: *mouseDown*)
 - El método *handleEvent*.

10

Applets: métodos básicos

- Otros componentes visuales heredados de AWT:
 - Botones (java.awt.Button)
 - Checkbox (java.awt.Checkbox)
 - Campos de texto de una línea (java.awt.TextField)
 - Áreas de edición más grandes (java.awt.TextArea)
 - Etiquetas (java.awt.Label)
 - Listas (java.awt.List)
 - Listas desplegadas (java.awt.Choice)
 - Sliders y barras de desplazamiento (java.awt.Scrollbar)
 - Áreas de dibujo (java.awt.Canvas)
 - Menús (java.awt.Menu, java.awt.MenuItem, java.awt.CheckboxMenuItem)
 - Contenedores (java.awt.Panel, java.awt.Window and its subclasses)

11

La etiqueta <APPLET>

```
<APPLET CODE=AppletSubclass.class
        WIDTH=entero
        HEIGHT=entero>
</APPLET>
```

12

La etiqueta <APPLET>

- Especificación de parámetros

```
<APPLET CODE=AppletSubclass.class  
        WIDTH=entero HEIGHT=entero>  
    <PARAM NAME=nombreP1 VALUE=Valor>  
    <PARAM NAME=nombreP2 VALUE=OtroValor>  
</APPLET>
```

13

La etiqueta <APPLET>

- Por defecto, el applet se busca en la misma carpeta donde esté el fichero html, pero puede especificarse otra carpeta:

```
<APPLET CODE=AppletSubclass.class  
        CODEBASE=carpeta WIDTH=entero HEIGHT=entero>  
</APPLET>
```

14

La etiqueta <APPLET>

```
< APPLET
  [CODEBASE = codebaseURL]
  CODE = appletFile
  [ALT = alternateText]
  [NAME = appletInstanceName]
  WIDTH = pixels HEIGHT = pixels
  [ALIGN = alignment]
  [VSPACE = pixels] [HSPACE = pixels] >

  [< PARAM NAME = appletParameter1 VALUE = value >]
  [< PARAM NAME = appletParameter2 VALUE = value >]
  ...
  [alternateHTML]

</APPLET>
```

15

La etiqueta <APPLET>

- Para obtener el valor de los parámetros puede usarse el método *getParameter*

```
public String getParameter(String name)
```

Devuelve el valor del parámetro cuyo nombre es *name* en la etiqueta <APPLET> de html

16

Comunicación entre applets

```
public AppletContext getAppletContext()
```

- Devuelve el contexto del applet
- Este contexto es un objeto que le permite al applet obtener información y modificar el entorno en el que se está ejecutando.
- Este entorno representa el documento que contiene al applet

17

Comunicación entre applets

```
public AppletContext getAppletContext()
```

- Devuelve el contexto del applet
- Este contexto es un objeto que le permite al applet obtener información y modificar el entorno en el que se está ejecutando.
- Este entorno representa el documento que contiene al applet
- El contexto se representa con el interfaz *AppletContext*

18

Applets que utilizan componentes Swing

- Debe crearse una subclase de *JApplet*
- La principal diferencia entre *Applet* y *JApplet* es que *JApplet* es un contenedor de nivel superior que proporciona un panel raíz para colocar otros componentes
- El resto de conceptos y métodos coinciden entre *Applet* y *JApplet*