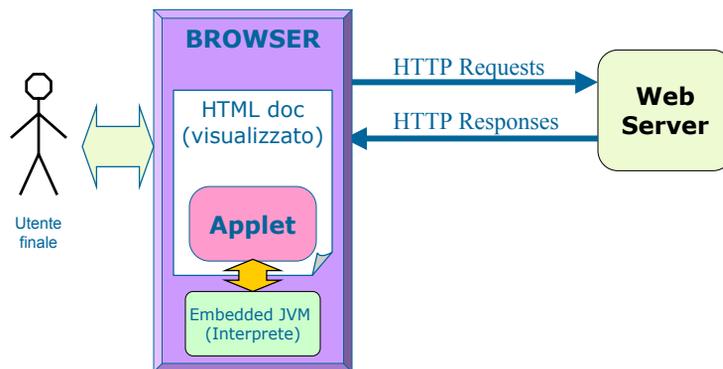


# Esecuzioni client-side: le Applet

A. Bechini 2004

## Introduzione

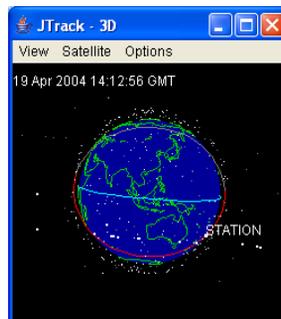
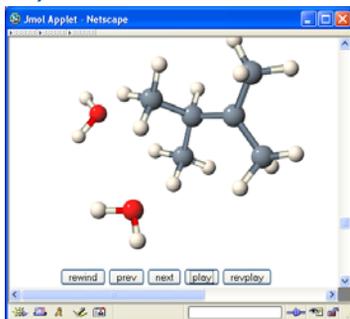
- Le Applet sono programmi Java che vengono eseguiti all'interno dei browser



A. Bechini 2004

## Esempi di Applet

- Visualizzazione di molecole (<http://jmol.sourceforge.net/applet/>)
- Tracker di satelliti artificiali, NASA (<http://science.nasa.gov/RealTime/jtrack/3d/JTrack3D.html>)



3

## Un esempio di codice

- Un'Applet è una classe Java che estende `java.applet.Applet`
- In alternativa, si può far uso della libreria grafica `Swing`, e in questo caso l'Applet deve estendere `javax.swing.JApplet`

```
import javax.swing.*; import java.awt.*;
public class PrimaJApplet extends JApplet {
    public void init() {
        Container contentPane = getContentPane();
        JLabel label = new JLabel("Questa e'
            la mia prima Applet",
            SwingConstants.CENTER);
        contentPane.add(label);
    }
}
```

4

## Una pagina HTML per l'Applet

- L'Applet deve essere integrata all'interno di una pagina HTML che la ospita
- A questo scopo viene utilizzato un apposito tag HTML
- Esempio:

```
<html>
  <body>
    <H2> Esempio di pagina con Applet </H2>
    <applet
      code="PrimaJApplet.class"
      width=300 height=300>
    </applet>
  </body>
</html>
```

*La dimensione è specificata qui, non dentro l'Applet!*

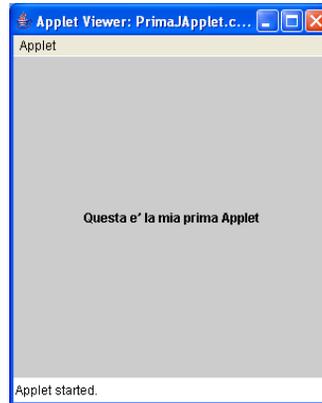
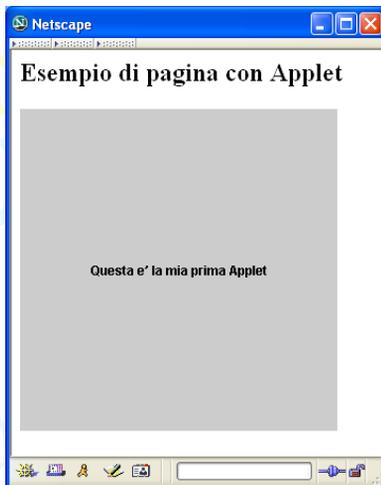
5

## Compilazione ed esecuzione

- **Compilazione:**  
`javac PrimaJApplet.java`
- **Esecuzione:**
  - Si fa riferimento al file HTML che contiene la Applet, e la si visualizza con il browser
  - In alternativa, si può utilizzare una utility standard di Java: `appletviewer`  
Questa utility non visualizza l'intero contenuto HTML, ma solo la Applet

6

# Risultato:



7

# Il codice: commento

```
import javax.swing.*;  
import java.awt.*;
```

Package utilizzati

```
public class PrimaJApplet extends JApplet {
```

Classe Swing

Si ottiene il Container su cui operare

```
public void init() {
```

```
    Container contentPane = getContentPane();
```

```
    JLabel label = new JLabel("Questa e'  
    la mia prima Applet",  
    SwingConstants.CENTER);
```

Si definisce una JLabel

```
    contentPane.add(label);
```

Si posiziona sul Container

```
    }  
}
```

8

## Osservazioni

- Le Applet sono create, eseguite e distrutte dal browser
- La dimensione di una Applet è stabilita nel relativo file HTML
- Le Applet non hanno un titolo, come le pagine HTML (ma possono avere menù)
- Il codice per la costruzione dell'Applet viene collocato nel metodo `init()`.
- Non c'è la necessità di fornire un `main()`
- Una Applet non può essere chiusa: la terminazione è determinata dal browser.
- Non occorre invocare il metodo `show()`, come nelle applicazioni grafiche Java: una Applet viene visualizzata automaticamente.

9

## Applet con più file .class

- Un'Applet può essere sviluppata con più di una classe
- In questo caso, la sua esecuzione coinvolge tutti i relativi file .class, ma dentro l'HTML ci si riferisce soltanto alla classe "applet"
- Il caricamento delle classi viene svolto dal modulo "class loader" che provvede a scaricarsi dal server i .class necessari, al momento del loro primo utilizzo.



10

## Trasporto di Applet con file JAR

- Per praticità, si possono raggruppare i file richiesti dalla Applet, e trasferirli *con un'unica connessione di rete*
- A questo scopo si usano i file **.jar** (Java Archive).
- Si tratta di file .zip che contengono un file "*manifest*", usato per descrivere il contenuto dell'archivio.
- Il Class Loader cerca nel JAR I file necessari, prima di provare a ottenerli tramite le rete.

11

## Riferimento a JAR da HTML

- Se si utilizzano file JAR, occorre specificare qualche informazione in più nel documento HTML che contiene la Applet, p.es.:

```
<APPLET  
  CODE="MiaApplet.class"  
  ARCHIVE="ClassiEImmagini.jar,AltroMateriale.jar"  
  WIDTH = 65  
  HEIGHT = 20 >  
</APPLET>
```

Classe "applet",  
entro i JAR

JAR da scaricare

12

# Creare JAR

- Un file .jar può essere creato con l'apposito comando, p.es.

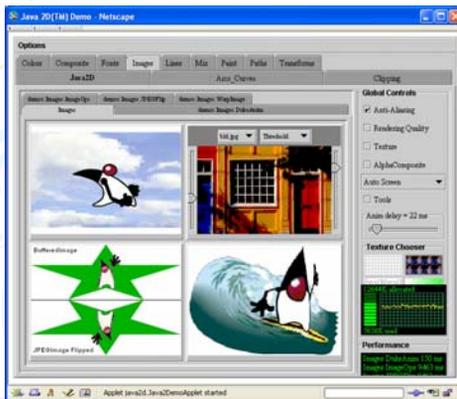
```
jar cf ClassiEImmagine.jar *class *.gif
```

- -> Inserisce nel jar tutti i file .class e .gif presenti nella directory, creando anche il manifest

A. Bechini 2004

13

# Esempio: Applet complessa



Contenuto (parziale)  
del JAR relativo

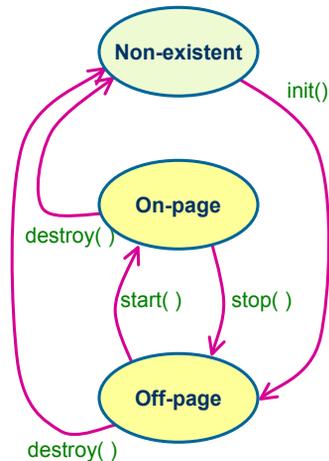
Name	Size	Packed	Type	Modified	CRC32
java_logo.png	5,076	5,081	PNG Image	16-Sep-03 6:36...	D629EAB
java_logo.gif	915	902	GIF Image	16-Sep-03 6:36...	770E988
gabe.gif	4,057	4,052	GIF Image	16-Sep-03 6:36...	F784916
light.gif	3,737	3,495	GIF Image	16-Sep-03 6:36...	8149571
duke_running.gif	2,537	2,057	GIF Image	16-Sep-03 6:36...	2669964
duke.png	9,946	5,207	GIF Image	16-Sep-03 6:36...	7300E99
duke.gif	2,212	2,217	PNG Image	16-Sep-03 6:36...	0701DE2
duke.gif	1,929	1,857	GIF Image	16-Sep-03 6:36...	59E0E83
cupcake.gif	5,856	5,241	GIF Image	16-Sep-03 6:36...	D89E506
clouds.jpg	5,116	5,001	JPEG Image	16-Sep-03 6:36...	A987663
duke.gif	885	130	GIF Image	16-Sep-03 6:36...	D908E22
bonwave.gif	4,690	4,515	GIF Image	16-Sep-03 6:36...	09E3880
bor.gif	4,241	4,076	GIF Image	16-Sep-03 6:36...	3D79010

A. Bechini 2004

14

## Ciclo di vita di una Applet

- A differenza di una normale applicazione, l'esecuzione di una Applet è controllata dal browser attraverso alcuni metodi:
  - **init()**
    - Invocato al caricamento
  - **start()**
    - Invocato dopo `init` e quando il browser ritorna sulla pagina
  - **stop()**
    - Invocato quando esce dalla pagina
  - **destroy()**
    - Invocato alla terminazione del browser



15

## Ciclo di vita: metodi (I)

- Per controllare il comportamento della Applet, il programmatore deve *ridefinire* (overriding) i metodi che ne regolano il ciclo di vita:
- **public void init()**
  - Eseguito una sola volta al caricamento dell'Applet: deve contenere le azioni di inzializzazione
- **public void start()**
  - Chiamato ogni volta che il browser torna alla pagina con la Applet, dopo essersi spostato su altre pagine.
  - Usato per riprendere attività interrotte (p.es. animazioni).

16



## Ciclo di vita: metodi (II)

- `public void stop()`
  - Chiamato ogni volta che il browser lascia la pagina con la Applet.
  - Usato per fermare temporaneamente attività che impegnano la CPU (p.es. animazioni)
- `public void destroy()`
  - Chiamato allo shut-down del browser
  - Usato per rilasciare risorse acquisite
  - Tipicamente poco utilizzato

17

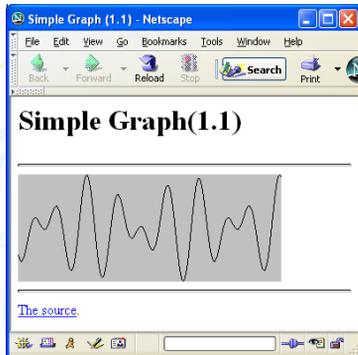


## Applet come programma standalone

- Per poter eseguire una Applet come un programma a se stante, occorre:
- Fornirla di un metodo main che:
  - istanzi la Applet all'interno di un `Frame` (o `JFrame`)
  - ne specifichi le dimensioni
  - richiami su tale istanza i metodi `init()` e `start()`

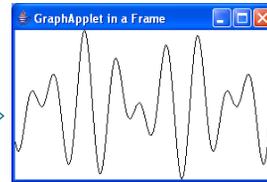
18

## Esempio di Applet "standalone"



Browser

Standalone



```
...
public static void main( String args[] ) {
    GraphApplet applet = new GraphApplet();
    Frame fr = new Frame("GraphApplet in a Frame");
    fr.add( applet );
    fr.setSize( 300, 200 );
    applet.init(); applet.start();
    fr.setVisible( true );
}
...
```

19

## Applet e sicurezza: la "sandbox"

- Possiamo scaricare Applet dalla rete, ed eseguirle sul nostro computer: **ci fidiamo?**
- Non abbiamo la possibilità di controllare direttamente l'esecuzione delle Applet: **esistono problemi di sicurezza**
- Risposta al problema: si impongono restrizioni a ciò che un'Applet può fare (sandbox)
- Si impiega un "Applet security manager", che lancia una **SecurityException** ogni volta che un'Applet tenta un accesso non consentito

20



## Restrizioni di default (I)

- Di default, l'esecuzione di una Applet è forzata dentro la "sandbox",
- ovvero con restrizioni molto strette
- Le cosiddette "signed Applets" possono avere maggiori privilegi.

A. Bechini 2004

21

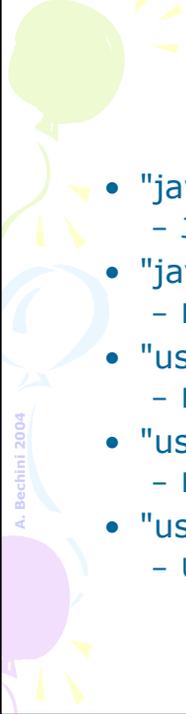


## Restrizioni di default (II)

- Ciò che un'Applet normalmente **non può fare**:
  - Leggere, scrivere, cancellare e avere informazioni su qualsiasi file locale
  - Eseguire un altro programma
  - Leggere alcune proprietà di sistema
  - Connettersi a una porta che non sia sull'host da cui proviene
  - Caricare una libreria Java
  - Invocare "exit"

A. Bechini 2004

22

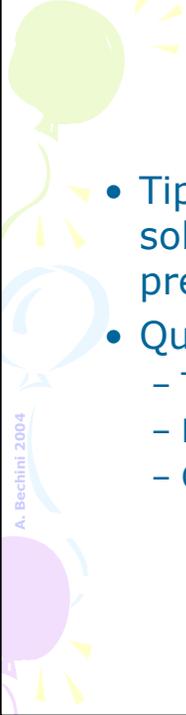


## Proprietà di sistema inaccessibili alle Applet

- "java.class.path"
  - Java classpath
- "java.home"
  - Directory di installazione di Java
- "user.dir"
  - Directory corrente del client
- "user.home"
  - Home directory del client
- "user.name"
  - User account corrente

A. Bechini 2004

23



## Comunicare con l'Home Server

- Tipicamente le Applet utilizzano la rete solo per accedere a risorse presenti sul server da cui provengono
- Quale tipo di risorse?
  - Testi
  - Multimedia
  - Qualsiasi tipo, tramite connessioni TCP-like

A. Bechini 2004

24

## Dov'è l' Home Server?

- Una risorsa web si localizza tramite un URL; in Java, una classe apposita gestisce oggetti di questo tipo: `java.net.URL`
- Come recuperare l'URL del server di provenienza?
- Da una sottoclasse di Applet:
  - `getCodeBase()` restituisce l'URL della Applet
  - `getDocumentBase()` restituisce l'URL della pagina HTML che contiene la Applet
- Una volta ottenuto l'URL, si possono aprire stream in input da esso

25

## Esempio: accedere a un file di testo

- Un file di testo può essere letto utilizzando un `InputStream`
- L'`InputStream` può essere ottenuto direttamente da un oggetto `URL`

```
import java.net.*;  
...  
InputStream in = url.openStream();  
...
```

26