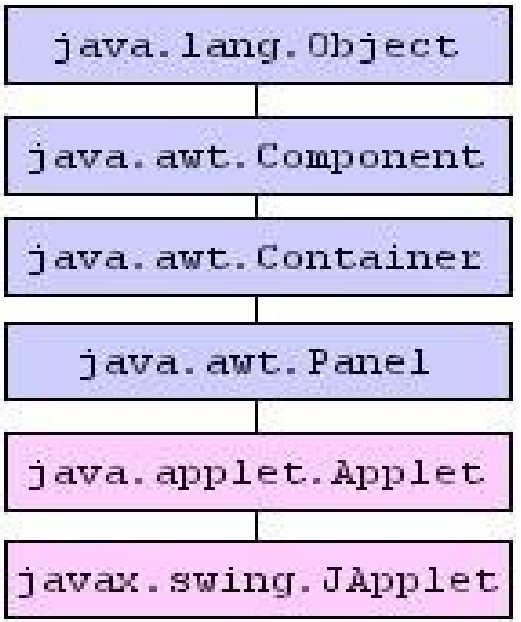
Programare avansată

Applet-uri

Ce este un applet?

Un *applet* reprezintă un program Java ce gestionează o suprafață de afișare (container) ce urmează a fi inclusă într-o pagină Web.

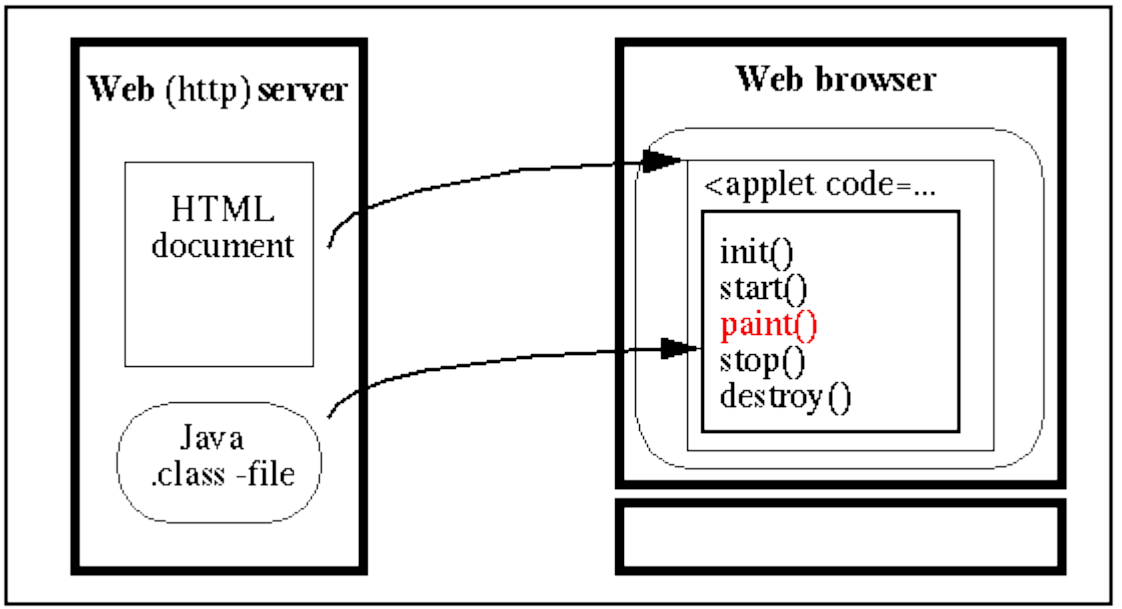


Exemplu

```
import java.awt.* ;
import java.applet.* ;
public class FirstApplet extends Applet {
   Image img;
   public void init() {
     img = getImage(getCodeBase(), "taz.gif");
   }
   public void paint (Graphics g) {
     g.drawImage(img, 0, 0, this);
     g.drawOval(100,0,150,50);
     g.drawString(
       "Hello! My name is Taz!",
        110, 25);
   }
}
```

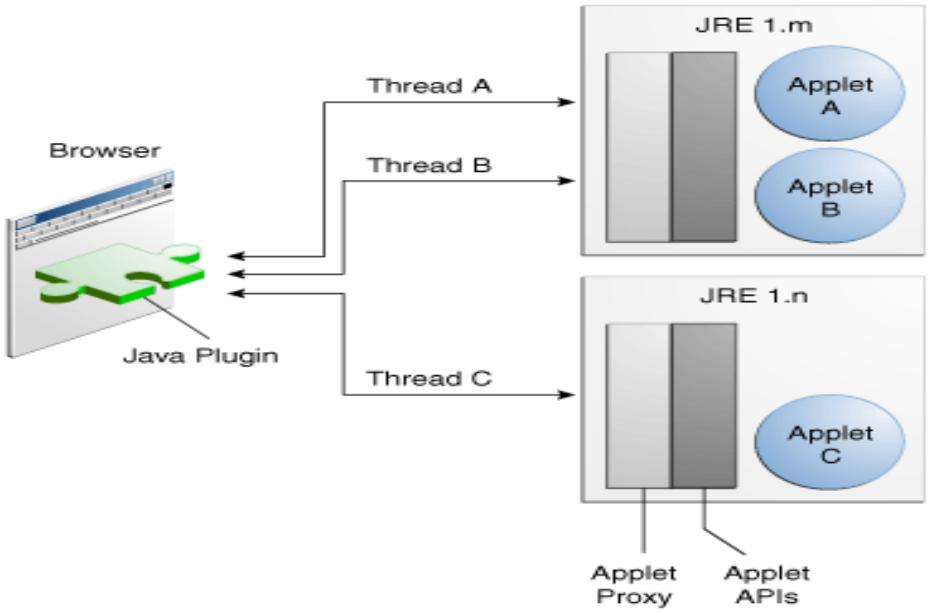
Execuția unui applet

```
<html>
<head> <title>Primul applet Java</title> </head>
<body>
  <applet code=FirstApplet.class
          width=400 height=400>
  </applet>
</body>
</html>
```



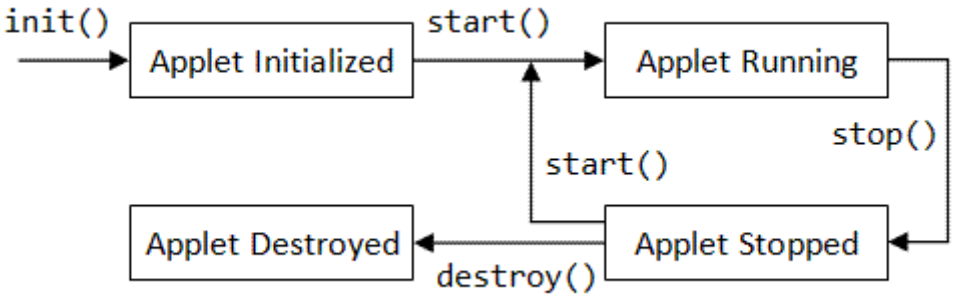
Java Plug-in

- Pentru a executa applet-uri un browser-ul va utiliza runtime-ul Java (JRE).
- “In mod normal” toate appleturile vor rula în cadrul aceleiași mașini virtuale (instanța JRE).
- Fiecărui applet îi va fi creat un *thread* propriu.



Ciclul de viață al unui applet

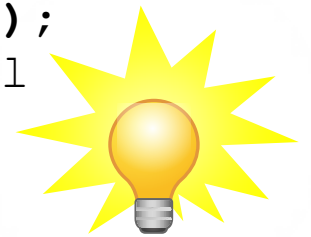
```
public class MyApplet extends java.applet.Applet {
    public void init() { ... }
    public void start() { ... }
    public void stop() { ... }
    public void destroy() { ... }
}
```



Arhitectura unui applet

```
public class MySwingApplet extends JApplet {
  public void init() {
    try {
      SwingUtilities.invokeAndWait(new Runnable() {
                public void run() {
                    createGUI();
                }
            });
    } catch (Exception e) {
        System.err.println(e);
    }
  }

  private void createGUI() {
      MyContentPane myContentPane = new MyContentPane();
      //MyContentPane va fi uzual o clasa de tip JPanel
      //ce contine reprezentarea grafica a appletului
      this.setContentPane(newContentPane);
  }
}
```



Tag-ul APPLET

```
<APPLET
     CODE = clasaApplet
     WIDTH = latimeInPixeli
     HEIGHT = inaltimeInPixeli
     [ARCHIVE = arhiva.jar]
     [CODEBASE = URLApplet] ->
     [ALT = textAlternativ]
     [NAME = numeInstanta]
     [ALIGN = aliniere]
     [VSPACE = spatiuVertical]
     [HSPACE = spatiuOrizontal] >
     [< PARAM NAME = parametru1 VALUE = valoare1 >]
     [< PARAM NAME = parametru2 VALUE = valoare2 >]
     ...
     [text HTML alternativ]
</APPLET>
```

getCodeBase()
URL-ul de unde provine clasa appletului
getDocumentBase()
URL-ul de unde provine documentul html

Folosirea parametrilor

Parametrii permit personalizarea aspectului

sau comportării unui applet fără a-i schimba codul.

Definirea:

```
<APPLET CODE="TestParametri.class" WIDTH=100 HEIGHT=50
  <PARAM NAME="textAfisat" VALUE="Salut">
  <PARAM NAME="numeFont" VALUE="Times New Roman">
  <PARAM NAME="dimFont" VALUE=20>
</APPLET>
```

Folosirea - *getParameter(name)*

”Documentarea”: *getParameterInfo()*

returnează un tablou de triplete: (nume, tip, descriere).

Exemplu de utilizare a parametrilor

```
import java.applet.Applet ;
import java.awt.*;
public class TestParametri extends Applet {
  String text, numeFont ;
  int dimFont ;
  public void init () {
    text = getParameter ("textAfisat");
    if ( text == null ) text = "Hello"; // valoare implicita
    numeFont = getParameter ("numeFont");
    if ( numeFont == null ) numeFont = "Arial";
    try {
      dimFont = Integer.parseInt(getParameter ("dimFont"));
    } catch ( NumberFormatException e) {
      dimFont = 16;
    }
  }
  public void paint (Graphics g) {
    g.setFont (new Font ( numeFont , Font.BOLD , dimFont ));
    g.drawString (text , 20, 20);
  }
  public String [][] getParameterInfo () {
    String [][] info = { // Nume Tip Descriere
      {"textAfisat", "String", "Sirul ce va fi afisat"},
      {"numeFont",   "String", "Numele fontului"},
      {"dimFont",     "int",   "Dimensiunea fontului"}
   };
}
```

Folosirea firelor de execuție

- Fiecărui applet îi este creat automat un fir de execuție responsabil cu apelarea metodelor acestuia. Acestea vor rula concurent.
- Din perspectiva GUI, fiecare applet are acces la un același fir de execuțe (*event dispatching thread)* responsabil cu desenarea și cu transmiterea evenimentelor generate de către componente.
- Operațile consumatoare de timp trebuie executate în fire de execuție proprii.

Incorect – blocarea EDT

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

public class BadApplet extends Applet {

  public void paint ( Graphics g) {
    while ( true ) {
      int x = (int )( Math.random () * getWidth ());
      int y = (int )( Math.random () * getHeight ());
      g.drawString ("Hello", x, y);
      try {
        Thread . sleep (1000) ;
      } catch ( InterruptedException e) {}
    }
  }
}
```

Folosirea unui *Thread* propriu

```
public class GoodApplet extends Applet implements Runnable {
  int x, y;
  Thread fir = null ;
  public void init () {
    if (fir == null ) {
      fir = new Thread ( this );
      fir.start ();
    }
  }
  public void run () {
    while ( true ) {
      x = (int )( Math . random () * getWidth ());
      y = (int )( Math . random () * getHeight ());
      repaint ();
      try {
        Thread . sleep (1000) ;
      } catch ( InterruptedException e) {}
    }
  }
  public void paint ( Graphics g) {
    g.drawString ("Hello", x, y);
  }
}
```

Folosirea metodelor *start-stop*

```
public class VeryGoodApplet extends Applet implements Runnable {
  int x, y;
  boolean activ = false;

  public void start () {
    if (!activ) {
      activ = true ;
      new Thread (this).start();
    }
  }
  public void stop () {
    activ = false ;
  }
  public void run () {
    while ( activ ) {
        ...
    }
  }
  public void paint ( Graphics g) {
    g.drawString ("Hello", x, y);
  }
}
```

Contextul de execuție

Contextul încare rulează appletul (pagina care îl conține).

AppletContext context = getAppletContext();

- Afișarea unor documente în browser

```
context.showDocument(new URL("http://www.infoiasi.ro"));
```

- Comunicare între appleturi de pe aceeași pagină
  - ✔ Identificarea unui applet: *getApplet, getApplets*
- Invocarea de funcții JavaScript din aceeași pagină

```
JSObject window = JSObject.getWindow(this);
Number value = (Number) window.eval("someFunction()");
```

- Accesarea arborelui DOM al paginii

```
HTMLDocument
```

Restricții de securitate

Un applet nu poate să:

- Citească sau să scrie fișiere pe calculatorul pe care a fost încărcat (client).
- Deschidă conexiuni cu alte mașini în afară de cea de pe care provine (host).
- Pornească programe pe mașina client.
- Citească diverse proprietăți ale sistemului de operare al clientului.
- ...

Securitatea aplicațiilor desktop

Protecția informațiilor de la nivelul *clientului* împotriva acțiunilor neautorizate ale unei aplicații (secvență de cod) provenite din rețea.

- *De unde provine codul?*

  Codebase

- *Cine a creat acel cod?*

  Semnături digitale (*jarsigner*)

- *Ce fel de operații dorește să execute?*

  Permisiuni *(File, Socket, Net, etc.)*

SecurityManager

- **checkRead(String file) throws SecurityException,…**
- **checkWrite(String file) throws SecurityException,…**
- **…**

Exemplu

```
public class java.io.File {
...
  public boolean canRead() {
    SecurityManager security = System.getSecurityManager();
    if (security != null) {
      security.checkRead(path);
    }
    FileSystem fs = FileSystem.getFileSystem();
    return fs.checkAccess(this, FileSystem.ACCESS_READ);
  }
}
```

Permisiuni

Crearea unui fisier de permisiuni (policy file)

`policytool`

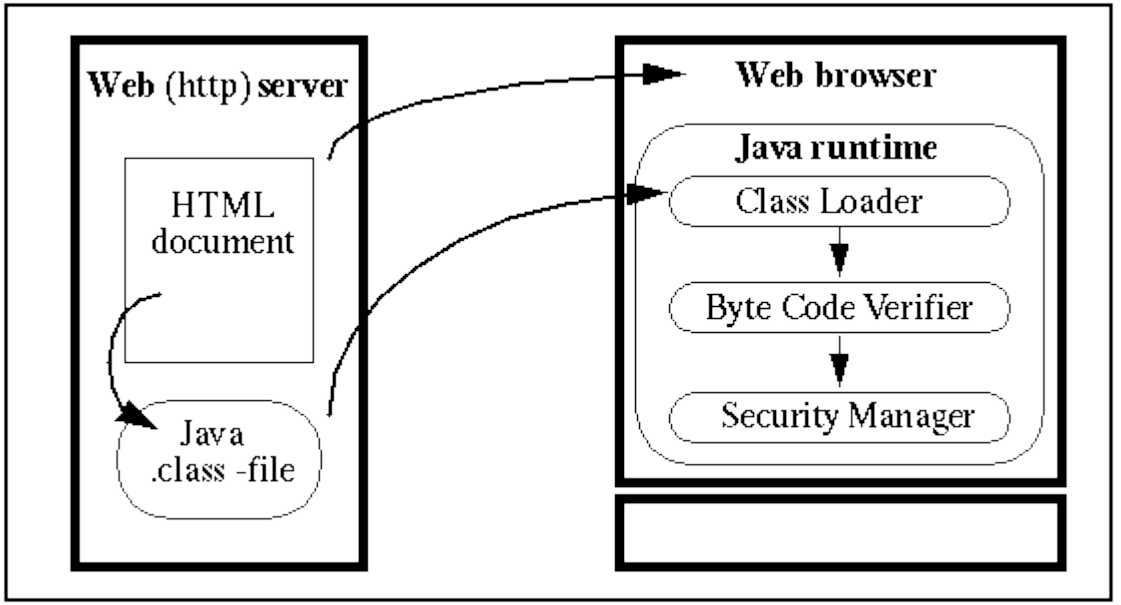
CodeBase=URL ("de unde")

SignedBy ("de la cine")

```
grant signedBy "Hacker" codeBase "file://d:/java/application/" {
  permission java.io.FilePermission "/test/*" , "read, write";
};
```

```
java -Djava.security.manager
     -Djava.security.policy=test.policy TestApp
```

Securitatea appleturilor



java.home/lib/security/java.security

```
# The default is to have a single system-wide policy file,
# and a policy file in the user's home directory.
policy.url.1=file:${java.home}/lib/security/java.policy
policy.url.2=file:${user.home}/.java.policy
```

Java Tutorial

Lesson: Java Applets

http://docs.oracle.com/javase/tutorial/deployment/applet/index.html