Aplicatii JAVA

3

JAVA
Applet-uri în Java

Adrian Runceanu
www.runceanu.ro/adrian

Curs 3

Applet-uri în Java

Applet-uri în Java

1. Ce este un applet?
2. Crearea unui applet simplu
3. Ciclul de viata al unui applet
4. Interfata grafica cu utilizatorul
5. Definirea si folosirea parametrilor
6. Tag-ul <APPLET>
7. Alte metode oferite de clasa Applet
8. Probleme de securitate

1. Ce este un applet?

Definitie

Un **applet** reprezinta *o suprafata de afisare (container) ce poate fi inclusa într-o pagina Web si gestionata printr-un program Java*.

Un astfel de program se mai numeste *miniaplicatie*.

1. Ce este un applet?

- Codul unui applet poate fi format din una sau mai multe clase.
- Una dintre acestea este principala si extinde clasa **Applet**, fiind clasa ce trebuie specificata în documentul HTML ce descrie pagina de Web în care dorim sa includem applet-ul.
- Diferenta fundamentala dintre un applet si o aplicatie consta în faptul ca, *un applet nu poate fi executat independent, ci va fi executat de browser-ul în care este încarcata pagina Web ce contine applet-ul respectiv*.

1. Ce este un applet?

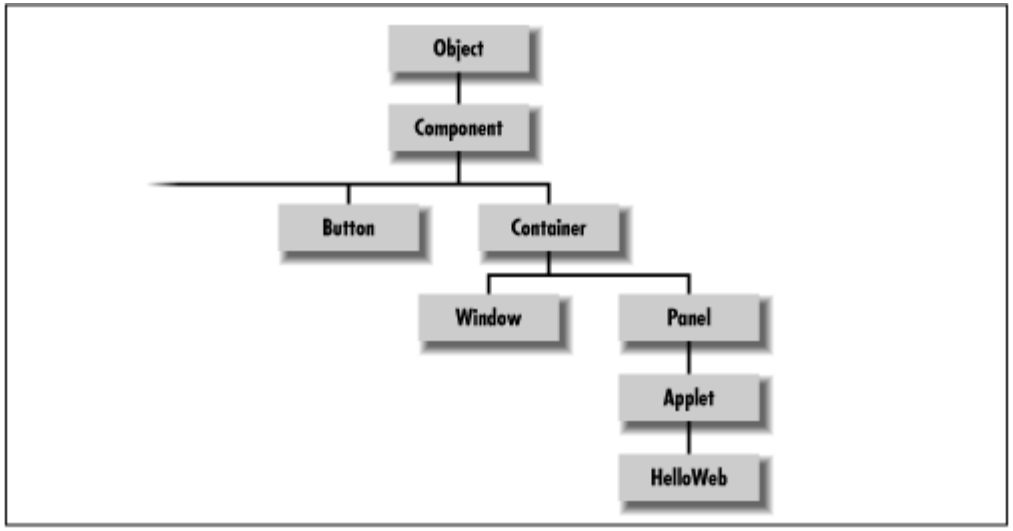
- O aplicatie independenta este executata prin apelul interpretorului **Java**, având ca parametru numele clasei principale a aplicatiei, clasa principala fiind cea care contine *metoda main*.
- Ciclul de viata al unui applet este complet diferit, fiind dictat de evenimentele generate de catre browser la vizualizarea documentului HTML ce contine applet-ul.
- Pachetul care ofera suport pentru creearea de applet-uri este **java.applet**.

Applet-uri în Java

1. Ce este un applet?
2. Crearea unui applet simplu
3. Ciclul de viata al unui applet
4. Interfata grafica cu utilizatorul
5. Definirea si folosirea parametrilor
6. Tag-ul <APPLET>
7. Alte metode oferite de clasa Applet
8. Probleme de securitate

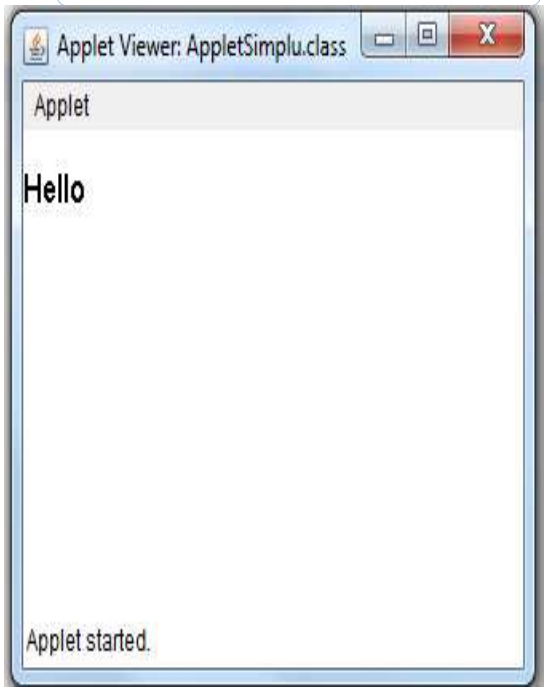
2. Crearea unui applet

- Orice applet este implementat prin crearea unei subclase a clasei **Applet**.
- Ierarhia claselor din care deriva **Applet** este prezentata în figura de mai jos:



- Fiind derivata din clasa Container, clasa Applet descrie de fapt suprafete de afisare, asemenea claselor Frame sau Panel.

2. Crearea unui applet



Un exemplu de applet simplu:

```
import java.applet.Applet;
import java.awt.*;
public class AppletSimplu extends Applet {
        public void paint(Graphics g) {
        g.setFont(new Font("Arial", Font.BOLD, 16));
        g.drawString("Hello", 0, 30);
        }
}
```

Se va salva clasa de mai sus într-un fisier AppletSimplu.java.

2. Crearea unui applet

Compilarea

- Compilarea se face la fel ca si la aplicatiile independente, apelând compilatorul javac pentru clasa principal a applet-ului (cea care extinde Applet).

  javac AppletSimplu.java

- In cazul în care compilarea a reusit va fi generat fisierul AppletSimplu.class.

2. Crearea unui applet

Executia (vizualizarea)

Pentru a vizualiza acest applet trebuie sa cream un document HTML, de exemplu demo.html, în care sa specificam cel putin urmatoarele informatii:

- clasa ce contine codul applet-ului
- latimea si înaltimea suprafetei alocate pe pagina Web

2. Crearea unui applet

```
// demo.html
<HTML>
        <HEAD>
                <TITLE> Un applet simplu </TITLE>
        </HEAD>
        <APPLET CODE="AppletSimplu.class"
WIDTH=100 HEIGHT=50></APPLET>
</HTML>
```

Vizualizarea acestui document se poate face cu orice browser (Internet Explorer, Chrome, Firefox, Safari, Opera etc), sau cu utilitarul appletviewer ce vine în pachetul **JDK**.

Applet-uri în Java

1. Ce este un applet?
2. Crearea unui applet simplu
3. Ciclul de viata al unui applet
4. Interfata grafica cu utilizatorul
5. Definirea si folosirea parametrilor
6. Tag-ul <APPLET>
7. Alte metode oferite de clasa Applet
8. Probleme de securitate

3. Ciclul de viata al unui applet

- Executia unui applet începe în momentul în care un browser afiseaza o pagina Web în care este inclus applet-ul respectiv si poate trece prin mai multe etape.

- Fiecare etapa este strâns legata de un eveniment generat de catre browser si determina apelarea unei metode specifice din clasa ce implementeaza applet-ul.

3. Ciclul de viata al unui applet

1. Incarcarea in memorie
   - Este creata o instanta a clasei principale a applet-ului si încarcata în memorie.

2. Initializarea
   - Este apelata metoda init ce permite initializarea diverselor variabile, citirea unor parametri de intrare, etc.

3. Pornirea
   - Este apelata metoda start

3. Ciclul de viata al unui applet

4. Executia propriu-zisa

- Consta în interactiunea dintre utilizator si componentele afisate pe suprafata applet-ului sau în executarea unui anumit cod într-un fir de executie.

- In unele situatii întreaga executie a applet-ului se consuma la etapele de initializare si pornire.

3. Ciclul de viata al unui applet

5. Oprirea temporara

- In cazul în care utilizatorul paraseste pagina Web în care ruleaza applet-ul este apelata metoda stop a acestuia, dându-i astfel posibilitatea sa se opreasca temporar cât timp nu este vizibil, pentru a nu consuma inutil din timpul procesorului.
- Acelasi lucru se întâmpla daca fereastra browserului este minimizata.
- In momentul când pagina Web ce contine applet-ul devine din nou activa, va fi reapelata metoda start.

3. Ciclul de viata al unui applet

6. Oprirea definitiva

- La închiderea tuturor instantelor browserului folosit pentru vizualizare, applet-ul va fi eliminat din memorie si va fi apelata metoda destroy a acestuia, pentru a-i permite sa elibereze resursele detinute.

- Apelul metodei destroy este întotdeauna precedat de apelul metodei stop.

3. Ciclul de viata al unui applet

Metodele specifice applet-urilor

Asadar, exista metode specifice applet-ului ce sunt apelate automat la diverse evenimente generate de catre browser:

Metoda	Situatia în care este apelata
init	La initializarea appletului; teoretic, aceasta metoda ar trebui sa se apeleze o singura data, la prima afisare a appletului în pagina, însa, la unele browsere, este posibil ca ea sa se apeleze de mai multe ori.
start	Imediat dupa initializare si de fiecare data când appletul redevine activ, dupa o oprire temporara.
stop	De fiecare data când appletul nu mai este vizibil (pagina Web nu mai este vizibila, fereastra browserului este minimizata, etc) si înainte de `destroy`.
destroy	La închiderea ultimei instante a browserului care a încarcat în memorie clasa principala a appletului.
Aceste metode sunt apelate automat de browser si nu trebuie apelate explicit din program !	

3. Ciclul de viata al unui applet

Structura generala a unui applet:

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class StructuraApplet extends Applet {
        public void init() { }
        public void start() { }
        public void stop() { }
        public void destroy() { }
        public void paint(Graphics g) { }
}
```

Applet-uri în Java

1. Ce este un applet?
2. Crearea unui applet simplu
3. Ciclul de viata al unui applet
4. Interfata grafica cu utilizatorul
5. Definirea si folosirea parametrilor
6. Tag-ul <APPLET>
7. Alte metode oferite de clasa Applet
8. Probleme de securitate

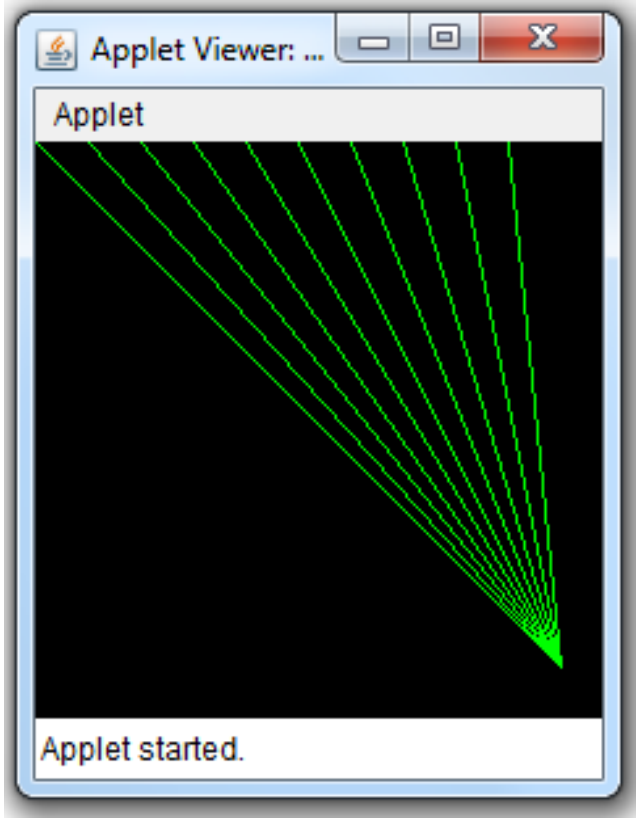
4. Interfata grafica cu utilizatorul

- Dupa cum am vazut, clasa **Applet** este o extensie a superclasei Container, ceea ce înseamna ca applet-urile sunt, înainte de toate, suprafete de afisare.
- Plasarea componentelor, gestionarea pozitionarii lor si tratarea evenimentelor generate se realizeaza la fel ca si în cazul aplicatiilor.
- Uzual, adaugarea componentelor pe suprafata appletului precum si stabilirea obiectelor responsabile cu tratarea evenimentelor generate sunt operatiuni ce vor fi realizate în metoda init.
- Gestionarul de pozitionare implicit este FlowLayout, însa acesta poate fi schimbat prin metoda setLayout.

4. Interfata grafica cu utilizatorul

Desenarea pe suprafata unui applet

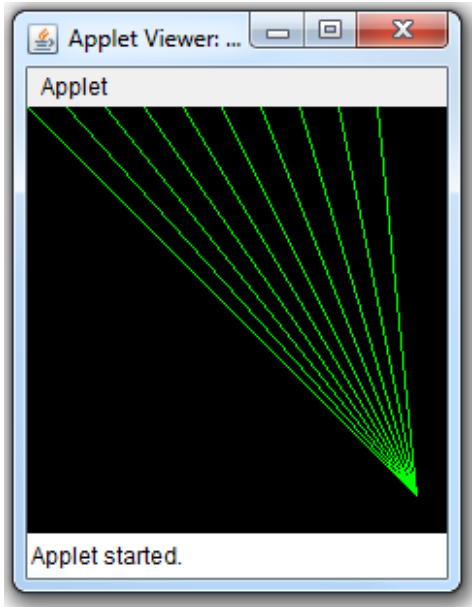
- Exista o categorie întreaga de applet-uri ce nu comunica cu utilizatorul prin intermediul componentelor ci, executia lor se rezuma la diverse operatiuni de desenare executate în metoda paint.



Exemplu metoda paint

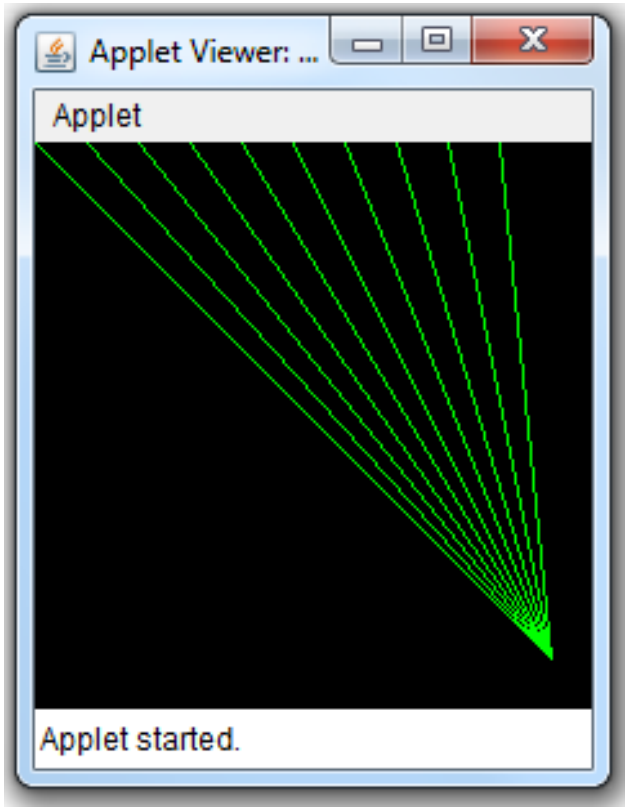
```
import java.applet.*;
import java.awt.*;
public class DrawingLines extends Applet {
  // Declare two variables of type "int" (integer).
   int width, height;
  // This gets executed when the applet starts.
   public void init() {
    // Store the height and width of the applet for future reference.
      width = getSize().width;
      height = getSize().height;

      // Make the default background color black.
      setBackground( Color.black );
   }
```



Exemplu metoda paint(continuare)

```
// This gets executed whenever the applet is asked to redraw itself.
public void paint( Graphics g ) {
  // Set the current drawing color to green.
    g.setColor( Color.green );
  // Draw ten lines using a loop.
  // We declare a temporary variable, i, of type "int".
  // Note that "++i" is simply shorthand for "i=i+1"
    for ( int i = 0; i < 10; ++i ) {
// The "drawLine" routine requires 4 numbers:
      // the x and y coordinates of the starting point,
      // and the x and y coordinates of the ending point,
      // in that order.  Note that the cartesian plane,
      // in this case, is upside down (as it often is
      // in 2D graphics programming): the origin is at the
      // upper left corner, the x-axis increases to the right,
      // and the y-axis increases downward.
        g.drawLine( width, height, i * width / 10, 0 );  } }}
```



Applet-uri în Java

1. Ce este un applet?
2. Crearea unui applet simplu
3. Ciclul de viata al unui applet
4. Interfata grafica cu utilizatorul
5. Definirea si folosirea parametrilor
6. Tag-ul <APPLET>
7. Alte metode oferite de clasa Applet
8. Probleme de securitate

5. Definirea si folosirea parametrilor

- In cazul în care este aleasa aceasta solutie, evenimentele tratate uzual vor fi cele generate de mouse sau tastatura.

- Parametrii sunt pentru applet-uri ceea ce argumentele de la linia de comanda sunt pentru aplicatiile independente.

- Ei permit utilizatorului sa personalizeze aspectul sau comportarea unui applet fara a-i schimba codul si recompila clasele.

5. Definirea si folosirea parametrilor

- Definirea parametrilor se face în cadrul tagului APPLET din documentul HTML ce contine applet-ul si sunt identificati prin atributul PARAM.
- Fiecare parametru are un nume, specificat prin NAME si o valoare, specificata prin VALUE, ca în exemplul de mai jos:

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

- Ca si în cazul argumentelor trimise aplicatiilor de la linia de comanda, *tipul parametrilor este sir de caractere*, indiferent daca valoarea este între ghilimele sau nu.

5. Definirea si folosirea parametrilor

- Fiecare applet are si un set de parametri prestabiliti ale caror nume nu vor putea fi folosite pentru definirea de noi parametri folosind metoda de mai sus.

- Acestia apar direct în corpul tagului APPLET si definesc informatii generale despre applet.

- Exemple de astfel de parametri sun CODE, WIDTH sau HEIGHT.

5. Definirea si folosirea parametrilor

- Folosirea parametrilor primiti de catre un applet se face prin intermediul metodei getParameter care primeste ca argument numele unui parametru si returneaza valoarea acestuia.
- In cazul în care nu exista nici un parametru cu numele specificat, metoda întoarce null, caz în care programul trebuie sa atribuie o valoare implicita variabilei în care se dorea citirea respectivului parametru.
- Sa rescriem applet-ul considerat initial (AppletSimplu) astfel încât acesta sa afiseze textul primit ca parametru, folosind un font cu numele si dimeniunea specificate de asemenea ca parametri.

5. Definirea si folosirea parametrilor

```
import java.applet.Applet;
import java.awt.*;
public class AppletSimplu 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);
}
}
```

5. Definirea si folosirea parametrilor

- Orice applet poate pune la dispozitie o documentatie" referitoare la parametrii pe care îi suporta, pentru a veni în ajutorul utilizatorilor care doresc sa includa applet-ul într-o pagina Web.
- Aceasta se realizeaza prin supradefinirea metodei getParameterInfo, care returneaza un vector format din triplete de siruri.
- Fiecare element al vectorului este de fapt un vector cu trei elemente de tip String, cele trei siruri reprezentând *numele parametrului*, *tipul* sau si *o descriere a sa*.

5. Definirea si folosirea parametrilor

```
public String[][] getParameterInfo() {
String[][] info = {
        // Nume Tip Descriere
        {"textAfisat", "String", "Sirul ce va fi afisat"},
        {"numeFont", "String", "Numele fontului"},
        {"dimFont", "int", "Dimensiunea fontului"}
};
return info;
}
```

Informatiile furnizate de un applet pot fi citite din browserul folosit pentru vizualizare prin metode specifice acestuia.

Applet-uri în Java

1. Ce este un applet?
2. Crearea unui applet simplu
3. Ciclul de viata al unui applet
4. Interfata grafica cu utilizatorul
5. Definirea si folosirea parametrilor
6. Tag-ul <APPLET>
7. Alte metode oferite de clasa Applet
8. Probleme de securitate

6. Tag-ul <APPLET>

< APPLET
[CODEBASE = directorApplet]
CODE = clasaApplet
[ALT = textAlternativ]
[NAME = numeInstantaApplet]
WIDTH = latimeInPixeli
HEIGHT = înaltimeInPixeli
[ALIGN = aliniere]
[VSPACE = spatiuVertical]
[HSPACE = spatiuOrizontal]
>
[< PARAM NAME = numeParametru1 VALUE = valoare1 >]
[< PARAM NAME = numeParametru2 VALUE = valoare2 >]
. . .
[text HTML alternativ]
</APPLET>

6. Tag-ul <APPLET>

- Atributele puse între paranteze patrate sunt optionale.

CODEBASE = directorApplet

- Specifica URL-ul în care se gaseste clasa applet-ului. Uzual se exprima relativ la directorul documentului HTML.
- In cazul în care lipseste, se considera implicit URL-ul documentului.

CODE = clasaApplet

- Numele fisierului ce contine clasa principala a applet-ului.
- Acesta va fi cautat în directorul specificat de CODEBASE. Nu poate fi absolut.

6. Tag-ul <APPLET>

ALT = textAlternativ

- Specifica textul ce trebuie afisat daca browserul înțelege tagul APPLET dar nu poate rula applet-uri Java.

NAME = numeInstantaApplet

- Ofera posibilitatea de a da un nume respectivei instante a applet-ului, astfel încât mai multe applet-uri aflate pe aceeasi pagina sa comunice între ele folosindu-se de numele lor.

6. Tag-ul <APPLET>

WIDTH = latimeInPixeli
HEIGHT = înaltimeInPixeli

- Specifica latimea si înaltimea suprafetei în care va fi afisat applet-ul.

ALIGN = aliniere

- Semnifica modalitatea de aliniere a applet-ului în pagina Web.
- Acest atribut poate primi una din urmatoarele valori: left, right, top, texttop, middle, absmiddle, baseline, bottom, absbottom, semnificatiile lor fiind aceleasi ca si la tagul **IMG**.

6. Tag-ul <APPLET>

VSPACE = spatiuVertical
HSPACE = spatiuOrizontal

- Specifica numarul de pixeli dintre applet si marginile suprafetei de afisare.

< PARAM NAME = numeParametru1 VALUE = valoare1 >

- Tag-urile <PARAM> sunt folosite pentru specificarea parametrilor unui applet.

Applet-uri în Java

1. Ce este un applet?
2. Crearea unui applet simplu
3. Ciclul de viata al unui applet
4. Interfata grafica cu utilizatorul
5. Definirea si folosirea parametrilor
6. Tag-ul <APPLET>
7. Alte metode oferite de clasa Applet
8. Probleme de securitate

7. Alte metode oferite de clasa Applet

Clasa **Applet** ofera metode specifice applet-urilor pentru:

- Punerea la dispozitie a unor informatii despre applet
- Similara cu metoda getParameterInfo ce oferea o "documentatie" despre parametrii pe care îi suporta un applet, exista metoda getAppletInfo ce permite specificarea unor informatii legate de applet cum ar fi numele, autorul, versiunea, etc.
- Metoda returneaza un sir de caractere continând informatii despre applet.

```
public String getAppletInfo() {
return "Cel mai simplu applet, autor necunoscut, ver 1.0";
}
```

7. Alte metode oferite de clasa Applet

Aflarea unor adrese **URL** referitoare la applet

Se realizeaza cu metodele:

- getCodeBase - ce returneaza **URL**-ul directorului ce contine clasa appletului
- getDocumentBase - returneaza **URL**-ul directorului ce contine documentul HTML în care este inclus applet-ul respectiv.

Sunt foarte utile deoarece permit specificarea relativa a fisierelor folosite de un applet.

7. Alte metode oferite de clasa Applet

Afisarea imaginilor

- Afisarea imaginilor într-un applet se face fie prin intermediul unei componente ce permite acest lucru, cum ar fi o suprafata de desenare de tip Canvas, fie direct în metoda paint a applet-ului, folosind metoda drawImage a clasei Graphics.

- In ambele cazuri, încarcarea imaginii în memorie se va face cu ajutorul metodei getImage din clasa **Applet**.

7. Alte metode oferite de clasa Applet

Aceasta poate primi ca argument fie adresa URL absoluta a fisierului ce contine imaginea, fie calea sa relativa la o anumita adresa URL, cum ar fi cea a directorului în care se gaseste documentul HTML ce contine applet-ul (getDocumentBase) sau a directorului în care se gaseste clasa applet-ului (getCodeBase).

7. Alte metode oferite de clasa Applet

```
import java.applet.Applet;
import java.awt.*;
public class AppletImagine extends Applet {
        Image img = null;
        public void init() {
                img = getImage(getCodeBase(), "imag.gif");
        }
        public void paint(Graphics g) {
                g.drawImage(img, 0, 0, this);
        }
}
```

7. Alte metode oferite de clasa Applet

Afisarea unor mesaje în bara de stare a browserului

Acest lucru se realizeaza cu metoda showStatus

```
public void init() {
        showStatus("Initializare applet...");
}
```

7. Alte metode oferite de clasa Applet

Aflarea contextului de executie

- Contextul de executie al unui applet se refera la pagina în care acesta ruleaza si este descris de interfata AppletContext.
- Crearea unui obiect ce implementeaza aceasta interfata se realizeaza de catre browser, la apelul metodei getAppletContext a clasei **Applet**.
- Prin intermediul acestei interfete un applet poate "vedea“ în jurul sau, putând comunica cu alte applet-uri aflate pe aceeasi pagina sau cere browser-ului sa deschida diverse documente.

```
AppletContext env = getAppletContext();
```

7. Alte metode oferite de clasa Applet

Afisarea unor documente în browser

Se face cu metoda showDocument ce primeste adresa URL a fisierului ce contine documentul dorit (text, html, imagine, etc).

Aceasta metoda se gaseste în interfata AppletContext.

```
try {
        URL doc = new
URL("http://www.scoaladeinformatica.ro");
        getAppletContext().showDocument(doc);
} catch(MalformedURLException e) {}
```

7. Alte metode oferite de clasa Applet

Comunicarea cu alte applet-uri aflate pe aceeasi pagina

- Aceasta comunicare implica de fapt identificarea unui applet aflat pe aceeasi pagina si apelarea unei metode sau setare unei variabile publice a acestuia.
- Identificarea se face prin intermediu numelui pe care orice instanta a unui applet îl poate specifica prin atributul NAME.
- Obtinerea unei instante a unui applet al carui nume îl cunoastem sau obtinerea unei enumerari a tuturor applet-urilor din pagina se fac cu metodele definite de interfata AppletContext **getApplet** si **getApplets**.

Applet-uri în Java

1. Ce este un applet?
2. Crearea unui applet simplu
3. Ciclul de viata al unui applet
4. Interfata grafica cu utilizatorul
5. Definirea si folosirea parametrilor
6. Tag-ul <APPLET>
7. Alte metode oferite de clasa Applet
8. Probleme de securitate

8. Probleme de securitate

Un applet nu poate sa:

- Citeasca sau scrie fisiere pe calculatorul pe care a fost încarcat (client)
- Deschida conexiuni cu alte masini în afara de cea de pe care provine (host)
- Porneasca programe pe masina client
- Citeasca diverse proprietati ale sistemului de operare al clientului

Ferestrele folosite de un applet, altele decât cea a browserului, vor arata altfel decât într-o aplicatie obisnuita.

Arhivarea appleturilor

- Am observant ca un applet aflat pe o pagina Web pentru a putea fi executat trebuie sa fie transferat de pe serverul care gazduieste pagina Web solicitata pe masina clientului.
- Deoarece transferul datelor prin retea este un proces lent, cu cat dimensiunea fisierelor care formeaza applet-ul este mai redusa, cu atat incarcarea acestuia se va face mai repede.
- Mai mult, daca applet-ul contine si alte clase in afara de cea principala sau diverse resurse (imagini, sunete, etc), acestea vor fi transferate prin retea abia in momentul in care va fi nevoie de ele, oprind temporar activitatea applet-ului pana la incarcarea lor.

Arhivarea appleturilor

- Din aceste motive, cea mai eficienta modalitate de a distribui un applet este sa arhivam toate fisierele necesare acestuia.
- Arhivarea fisierelor unui applet se face cu utilitarul jar, oferit in distributia J2SDK.

- Includerea unui applet arhivat intr-o pagina Web se realizeaza specificand pe langa numele clasei principale si numele arhivei care o contine:

<applet archive=arhiva.jar code=ClasaPrincipala width=400 height=200 />

Referinte

- Curs practic de Java, Cristian Frasinaru – capitolul Applet-uri
- http://docs.oracle.com/javase/8/docs/technotes/guides/jweb/applet/using_tags.html
- https://docs.oracle.com/javase/tutorial/deployment/applet/
- http://www.tutorialspoint.com/java/java_applet_basics.htm
- http://archive.oreilly.com/oreillyschool/courses/java3/java314.html
- http://www.javakode.com/applets/

Întrebări?