Appleturi

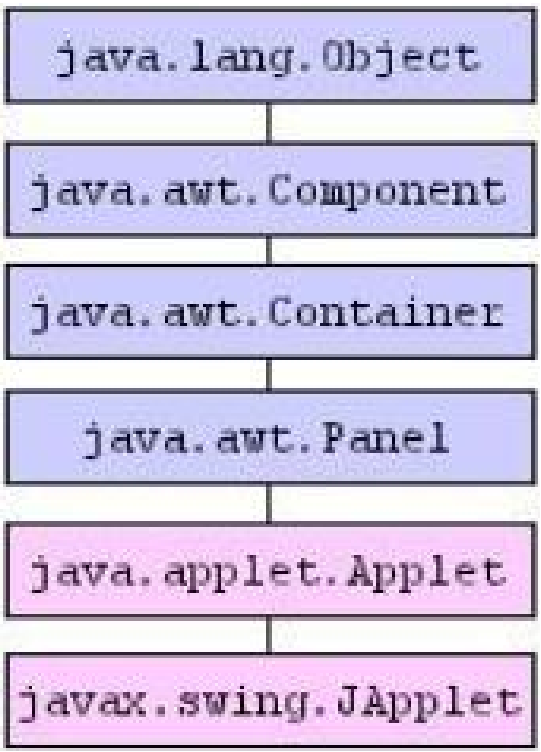
Programare Orientată pe Obiecte

Introducere

- program Java de dimensiuni reduse ce gestionează o suprafață de afişare (container) care poate fi inclusă într-o pagină Web.
- miniaplicatie.
- poate fi format din una sau mai multe clase
- o clasă principală ce extinde clasa Applet (JApplet) - clasa ce trebuie specificată în documentul HTML ce descrie pagina Web în care dorim să includem appletul.
- un applet nu poate fi executat independent, va fi executat de browserul în care este încărcată pagina Web ce conține appletul respectiv.
- ciclul de viată al unui applet: dictat de evenimentele generate de către browser la vizualizarea documentului HTML ce conține appletul.

Ierarhia claselor din care derivă appleturile

- Pachetul care oferă suport pentru crearea de appleturi este java.applet. In pachetul javax.swing există şi clasa JApplet, care extinde Applet, oferind suport pentru crearea de appleturi pe arhitectura de componente JFC/Swing.



Crearea unui applet simplu

1. Scrierea codului sursa

```
import java.awt.* ;
import javax.swing.* ;
public class FirstApplet extends JApplet {
    Image img;
    public void init() {
        img = getImage(getCodeBase(), “image.jpg");
    }
    public void paint (Graphics g) {
        g.drawImage(img, 100, 0, this);
        g.drawOval(0,0,120,50);
        g.drawString(“Ce ziceti de asta?", 0, 25);
    }
}
```

- Pentru a putea fi executată de browser, clasa principală a appletului trebuie să fie publică.

Crearea unui applet simplu (2)

2. Salvarea fişierelor sursă

   - clasa principală a appletului va fi salvată într-un fişier cu acelaşi nume şi extensia “java”

   - FirstApplet.java.

3. Compilarea

   - javac FirstApplet.java

   - rezulta FirstApplet.class

4. Rularea appletului

   Applet-urile nu ruleaza independent. Ele pot fi rulate doar prin intermediul unui browser: Internet Explorer, Netscape, Mozilla, Opera, etc. sau printr-un program special cum ar fi appletviewer din kitul de dezvoltare J2SDK.

Crearea unui applet simplu (3)

Pentru a executa un applet trebuie să facem două operații:

- Crearea unui fişier HTML în care vom include applet-ul.

  fişierul simplu.html:

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

- Vizualizarea appletului: se deschide fisierul simplu.html folosind unul din browser-ele amintite sau efectuând apelul: appletviewer simplu.html.

Ciclul de viață al unui applet

- Fiecare etapă este strâns legată de un eveniment generat de către browser şi determină apelarea unei metode specifice din clasa ce implementează appletul.

• Incărcarea în memorie

  Este creată o instanța a clasei principale a appletului şi încărcată în memorie.

• Inițializarea

  Este apelată metoda **init** ce permite inițializarea diverselor variabile,citirea unor parametri de intrare, etc.

• Pornirea

  Este apelată metoda **start**

• Execuția propriu-zisă

  - Interacțiunea dintre utilizator şi componentele afişate pe suprafața appletului
  - Executarea unui anumit cod într-un fir de execuție.
  - In unele situații întreaga execuție a appletului se consumă la etapele de inițializare şi pornire.

Ciclul de viață al unui applet

- Oprirea temporară

  - este apelată metoda **stop** ce opreşte temporar execuția pe perioada în care nu este vizibil, pentru a nu consuma inutil din timpul procesorului.
    - în cazul în care utilizatorul părăseşte pagina Web în care rulează appletul
    - dacă fereastra browserului este minimizată. În momentul în care pagina Web ce conține appletul devine din nou activă, va fi reapelată metoda start.

- Oprirea definitivă

  - La închiderea tuturor instanțelor browserului folosit pentru vizualizare, appletul va fi eliminat din memorie şi va fi apelată metoda destroy a acestuia, pentru a-i permite să elibereze resursele deținute.

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

Metodele specifice appleturilor

- Aceste metode sunt apelate automat la diverse evenimente generate de către browser şi nu trebuie apelate explicit din program
- sunt definite în clasa Applet

Metoda	Situația în care este apelată
init	La inițializarea appletului. Teoretic, această metodă ar trebui să se apeleze o singură dată, la prima afişare a appletului în pagină, însă, la unele browsere, este posibil ca ea să se apeleze de mai multe ori.
start	Imediat după inițializare şi de fiecare dată când appletul redevine activ, după o oprire temporară.
stop	De fiecare dată când appletul nu mai este vizibil (pagina Web nu mai este vizibilă, fereastra browserului este minimizată, etc) şi înainte de destroy.
destroy	La închiderea ultimei instanțe a browserului care a încărcat în memorie clasa principală a appletului.

Structura generală a unui applet

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class StructuraApplet extends JApplet {
    public void init() {
    }
    public void start() {
    }
    public void stop() {
    }
    public void destroy() {
    }
}
```

Interfața grafică cu utilizatorul

- Clasa Applet este o extensie a superclasei Container, ceea ce înseamnă că appleturile sunt, înainte de toate, suprafețe de afişare.
- Plasarea componentelor, gestionarea poziționării lor şi tratarea evenimentelor generate se realizează la fel ca şi în cazul aplicațiilor.
- Uzual, adăugarea componentelor pe suprafața appletului precum şi stabilirea obiectelor responsabile cu tratarea evenimentelor generate sunt operațiuni ce vor fi realizate în metoda init.
- Gestionarul de poziționare implicit este FlowLayout, însă acesta poate fi schimbat prin metoda setLayout.

Definirea şi folosirea parametrilor

- Parametrii sunt pentru appleturi ceea ce argumentele de la linia de comandă sunt pentru aplicațiile independente.
- Permit utilizatorului să personalizeze aspectul sau comportarea unui applet fără a-i schimba codul şi recompila clasele.
- Definirea: în cadrul tagului APPLET din documentul HTML ce conține appletul prin atributul PARAM.
- Fiecare parametru are un nume, specificat prin NAME şi o valoare, specificată prin VALUE:

<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>

- tipul parametrilor este întotdeauna şir de caractere, indiferent dacă valoarea este între ghilimele sau nu.
- Fiecare applet are şi un set de parametri prestabiliți ale căror nume nu vor putea fi folosite pentru definirea de noi parametri

Definirea şi folosirea parametrilor

- getParameter: primeşte ca argument numele unui parametru şi returnează valoarea acestuia. In cazul în care nu există nici un parametru cu numele specificat, metoda întoarce null, caz în care programul trebuie să atribuie o valoare implicită variabilei în care se dorea citirea respectivului parametru.
- Orice applet poate pune la dispoziție o ”documentație” referitoare la parametrii pe care îi suportă
- Aceasta se realizează prin supradefinirea metodei getParameterInfo, care returnează un vector format din triplete de şiruri (numele parametrului, tipul său şi o descriere a sa).
- Informațiile furnizate de un applet pot fi citite din browserul folosit pentru vizualizare prin metode specifice acestuia.
- De exemplu, în appletviewer informațiile despre parametri pot fi vizualizate la rubrica Info din meniul Applet, în Netscape se foloseşte opțiunea Page info din meniul View, etc.

Folosirea parametrilor

- un applet care să afişează un text primit ca parametru, folosind un font cu numele şi dimensiunea specificate de asemenea ca parametri.

```
import javax . swing.* ;
import java . awt .*;
public class TestParametri extends JApplet {
    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;
    }
}
```

Folosirea parametrilor

```
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 "}
    };
    return info ;
  }
}
```

Tag-ul APPLET

- Sintaxa completă a tagului APPLET, cu ajutorul căruia pot fi incluse appleturi în cadrul paginilor Web este:

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

Atributele puse între paranteze pătrate sunt opționale.

Tag-ul APPLET

- CODE = clasaApplet

  Numele fişierului ce conține clasa principală a appletului. Acesta va fi căutat în directorul specificat de CODEBASE. Extensia ”.class” poate sau nu să apară.
- WIDTH =latimeInPixeli, HEIGHT =inaltimeInPixeli

  Specifică lățimea şi înălțimea suprafeței în care va fi afişat appletul. Sunt obligatorii.
- ARCHIVE = arhiva.jar

  Specifică arhiva în care se găsesc clasele appletului.
- CODEBASE = directorApplet

  Specifică URL-ul la care se găseşte clasa appletului. Uzual se exprimă relativ la directorul documentului HTML. In cazul în care lipseşte, se consideră implicit URL-ul documentului.
- ALT = textAlternativ

  Specifică textul ce trebuie afişat dacă browserul înțelege tagul APPLET dar nu poate rula appleturi Java.

Tag-ul APPLET

- NAME =numeInstantaApplet

  Oferă posibilitatea de a da un nume respectivei instanțe a appletului, astfel încât mai multe appleturi aflate pe aceeaşi pagină să poată comunica între ele folosindu-se de numele lor.

- ALIGN =aliniere

  Semnifică modalitatea de aliniere a appletului în pagina Web. Acest atribut poate primi una din următoarele valori: left, right, top, texttop, middle, absmiddle, baseline, bottom, absbottom

- VSPACE =spatiuVertical, HSPACE = spatiuOrizontal

  Specifică numarul de pixeli dintre applet şi marginile suprafetei de afişare.

- PARAM

  Tag-urile PARAM sunt folosite pentru specificarea parametrilor unui applet

- text HTML alternativ

  Este textul ce va fi afişat în cazul în care browserul nu întelege tagul APPLET. Browserele Java-enabled vor ignora acest text.

Alte metode oferite de clasa Applet

- Punerea la dispozitie a unor informații despre applet

- getAppletInfo: permite specificarea unor informații legate de applet cum ar fi numele, autorul, versiunea, etc. Metoda returnează un sir de caractere continând informațiile respective.

```
public String getAppletInfo() {
    return "Applet simplist, autor necunoscut, ver 1.0";
}
```

- Aflarea adreselor URL referitoare la applet

• **getCodeBase** - returnează URL-ul directorului ce conține clasa appletului;

• **getDocumentBase** - returnează URL-ul directorului ce conține documentul HTML în care este inclus appletul respectiv.

Aceste metode sunt foarte utile deoarece permit specificarea relativă a unor fişiere folosite de un applet, cum ar fi imagini sau sunete.

Alte metode oferite de clasa Applet

- Afişarea unor mesaje în bara de stare a browserului

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

- Afişarea imaginilor
- prin intermediul unei componente ce permite acest lucru (Canvas), fie direct în metoda paint a applet-ului, folosind metoda drawImage a clasei Graphics.
- obținerea unei referințe la imaginea respectivă se va face cu ajutorul metodei getImage din clasa Applet.
- argument:
  - adresa URL absolută a fişierului ce reprezintă imaginea
  - calea relativă la o anumită adresă URL
    - cea a directorului în care se găseşte documentul HTML ce conține appletul (getDocumentBase)
    - cea a directorului în care se găseşte clasa appletului (getCodeBase).

Afişarea imaginilor

```
import javax . swing . *;
import java . awt .*;
public class Imagini extends JApplet {
    Image img = null ;
    public void init () {
        img = getImage ( getCodeBase () ,
    "taz.gif");
    }
    public void paint ( Graphics g) {
        g. drawImage (img , 0, 0, this );
    }
}
```

Alte metode oferite de clasa Applet

- Aflarea contextului de execuție:
  - pagina în care acesta rulează, eventual împreună cu alte appleturi
  - este descris de interfața AppletContext.
  - Crearea unui obiect ce implementează această interfață se realizează de către browser, la apelul metodei getAppletContext a clasei Applet.
  - Prin intermediul acestei interfețe un applet poate ”vedea” în jurul sau, putând comunica cu alte applet-uri aflate pe aceeasi pagină sau cere browser-ului să deschidă diverse documente
  - AppletContext contex = getAppletContext();
- Afişarea unor documente în browser

  showDocument: primeşte adresa URL a fişierului ce conține documentul pe care dorim sa-l deschidem (text, html, imagine, etc). Această metodă este accesată prin intermediul contextului de execuție al appletului.

```
try {
    URL doc = new URL("http://www.infoiasi.ro");
    getAppletContext().showDocument(doc);
}
catch(MalformedURLException e) {
            System.err.println("URL invalid! \n" + e);}
```

Arhivarea appleturilor

- pentru ca un applet aflat pe o pagină Web să poată fi executat codul său va fi transferat de pe serverul care găzduieşte pagina Web solicitată pe maşina clientului.
- dimensiunea fişierelor care formează appletul trebuie să fie cât mai redusă
- dacă appletul conține şi alte clase în afară de cea principală sau diverse resurse (imagini, sunete, etc), acestea vor fi transferate prin rețea abia în momentul în care va fi nevoie de ele, oprind astfel temporar activitatea appletului până la încărcarea lor.
- cea mai eficientă modalitate de a distribui un applet este să arhivăm toate fişierele necesare acestuia.
- Arhivarea fişierelor unui applet se face cu utilitarul jar, oferit în distribuția J2SDK.

```
// Exemplu
jar cvf arhiva.jar ClasaPrincipala.class AltaClasa.class
    imagine.jpg sunet.au
// sau
jar cvf arhiva.jar *.class *.jpg *.au
```

- Includerea unui applet arhivat într-o pagină Web se realizează specificând pe lângă numele clasei principale şi numele arhivei care o conține:

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

Restricții de securitate

- un applet se execută pe maşina utilizatorului care a solicitat pagina Web ce conține appletul respectiv
- este foarte important să existe anumite restricții de securitate care să controleze activitatea acestuia, pentru a preveni acțiuni rău intenționate, cum ar fi ştergeri de fişiere, etc., care să aducă prejudicii utilizatorului.
- procesul care rulează appleturi instalează un manager de securitate, un obiect de tip SecurityManager care va ”superviza” activitatea metodelor appletului, aruncând excepții de tip SecurityException în cazul în care una din acestea încearcă să efectueze o operație nepermisă.

Un applet nu poate să:

- Citească sau să scrie fişiere de pe calculatorul pe care a fost încarcat (client).
- Deschidă conexiuni cu alte maşini în afară de cea de pe care provine (host).
- Pornească programe pe maşina client.
- Citească diverse proprietăți ale sistemului de operare al clientului.
- Ferestrele folosite de un applet, altele decât cea a browserului, vor arăta altfel decât într-o aplicație obişnuită, indicând faptul că au fost create de un applet.

Appleturi care sunt şi aplicații

- Deoarece clasa Applet este derivată din Container, deci şi din Component, ea descrie o suprafață de afişare care poate fi inclusă ca orice altă componentă într-un alt container, cum ar fi o fereastră.
- Un applet poate funcționa şi ca o aplicație independentă astfel:

• Adăugăm metoda main clasei care descrie appletul, în care vom face operațiunile următoare.
• Creăm o instanță a appletului şi o adăugăm pe suprafața unei ferestre.
• Apelăm metodele init şi start, care ar fi fost apelate automat de către browser.
• Facem fereastra vizibilă.

Applet şi aplicație

```
import java . applet . Applet ;
import java . awt .*;
public class AppletAplicatie extends Applet {
    public void init () {
        add (new Label (" Applet si aplicatie "));
    }
    public static void main ( String args []) {
        AppletAplicatie applet = new AppletAplicatie ();
        Frame f = new Frame (" Applet si aplicatie ");
        f. setSize (200 , 200) ;
        f.add(applet , BorderLayout . CENTER );
        applet . init ();
        applet . start ();
        f. show ();
    }
}
```