

Lucrarea nr. 10 – Imagini și sunete în Java

Breviar teoretic

Java este un mediu dedicat în special mediului multimedia, implementării interfețelor om-calculator, om-masina, asigurând astfel adaptabilitatea programelor dezvoltate sub această platformă pe orice masina de calcul pe care este instalat interpretorul Java. În continuare vor fi prezentate elementele introductorii necesare lucrului cu elemente specifice interfețelor om-calculator, astfel încât aplicațiile dezvoltate să conducă la aplicații optimale din punct de vedere al funcționabilității interfeței.

În continuare vor fi prezentate și exemplificate următoarele elemente de baza destinate implementării interfețelor om-masina:

- lucrul cu imagini
- utilizarea sunetelor
- mouse-ul
- tastatura
- obiecte de tip cursor (bară de defilare)

Pe parcursul lucrării, datorită faptului că utilizatorul interacționează cu entitățile grafice, este explicată modalitatea de lucru cu evenimente, specifică utilizării mouse-ului, tastaturii și barelor de defilare

Lucrul cu imagini

Declararea unui obiect de tip imagine se realizează prin utilizarea tipului `Image`. Încărcarea unei imagini implică în primul rând localizarea sa. Pentru acest lucru se va folosi un obiect de tip `URL` ce memorează locația imaginii. Desigur că, în cazul în care imaginea se află în același director cu aplicația, nu mai este nevoie de obiectul `URL`. Dar la mutarea imaginii în alt director, va trebui ca appletul să fie recompilat. Pentru a preveni aceste situații se va recurge la metoda `getDocumentBase()` sau `getCodeBase()`. Obiectul `URL` va returna astfel directorul în care este memorată imaginea ce va fi utilizată.

Structura unui apel al metodei `getDocumentBase()`, metodă care întoarce directorul din care a fost încărcat documentul HTML. Un apel al acestei metode are forma:

```
URL url = getDocumentBase();
```

Această metodă oferă posibilitatea mutării structurii paginii HTML, la o altă locație, fără a mai fi nevoie de modificarea tuturor locațiilor în care se află celelalte elemente ale paginii. Metoda `getCodeBase()` lucrează similar cu metoda `getDocumentBase()`, cu observația că oferă locația din care a fost încărcat appletul:

```
URL url = getCodeBase();
```

Utilizarea acestor metode se poate realiza numai dacă în prealabil a fost încărcat pachetul - `java.net.*`.

Încărcarea unui imagini

Odată ce imaginea a fost localizată corespunzător, încărcarea acesteia se realizează prin metoda `getImage()` astfel:

```
URL codeBase = getCodeBase();
Image imagineaMea=getImage(codeBase, imagini/ imagineaMea.gif");
```

Se observă din exemplu că imaginea este localizată într-un subdirector - imagini, al directorului curent.

Afișarea unei imagini

Afișarea unei imagini se realizează simplu prin apelarea metodei **drawImage()** specifice obiectului **Graphics**:

```
g.drawImage(myImage, x, y, width, height, this);
```

Argumentele acestei metode sunt obiectul imagine ce va fi afișat, coordonatele X și Y de la care va începe afișarea imaginii, lungimea- **width** și înălțimea- **height** imaginii și referința **this** a appletului curent.

Există și o variantă rapidă a metodei drawImage la care se specifică numai variabila imagine, și punctul de început al afișării imaginii:

```
drawImage(image, x, y, this).
```

Imaginea va fi afișată utilizând dimensiunile implicite ale ei, fără ca mediul de programare să fie obligat să afișeze imaginea conform dimensiunilor specificate de prima variantă a metodei de desenare. Deoarece aproape întodeauna nu se cunosc lungimea și înălțimea imaginii sunt disponibile două metode ce pot determina acești parametri **getWidth()** și **getHeight()**, metode ce întorc valorile numerice ale celor doi parametri ai imaginii. Sintaxa utilizării acestor metode este:

```
int width = image.getWidth(this);
int height = image.getHeight(this);
g.drawImage(image, x, y, width, height, this);
```

Argumentul metodelor este referința la appletul curent **this**.

Aplicatie privind modul in care poate fi inserata o imagine in cadrul unui applet Java

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

public class ImageApplet extends Applet
{
    Image imagine;
    public void init()
    {
        URL director = getCodeBase();

        imagine = getImage(director, "shic.gif");
        resize(250, 250);
    }
}
```

```
public void paint(Graphics g)
{

int lungime = imagine.getWidth(this);
int inaltime = imagine.getHeight(this);
g.drawRect(52, 52, lungime+10, inaltime+10);
g.drawImage(imagine,57,57,lungime, inaltime, this);
}

}
```

Probleme propuse

1. Scrieți un applet care afișează un fișier imagine mărit de 3 ori față de dimensiunea sa inițială.
2. Scrieți un applet care dispune de 3 butoane care realizează afișarea unei imagini la scară diferită: 50%, 100%, și 200%. Implicite este afișarea de 25% din dimensiunea sa nominală.

Utilizarea sunetelor

Chiar dacă sunt mai multe tipuri de fișiere audio, Java utilizează numai fișiere de tip AU. Aceste fișiere sunt populare pentru mașini UNIX . Cum localizarea fișierelor se realizează prin utilizarea celor două metode prezentate anterior **getDocumentBase()** sau **getCodeBase()** , descrierea acestora nu va mai fi reluată. Rularea unui fișiere audio se realizează utilizând metoda **play()**:

```
play(baseUrl, relLocation);
```

Un exemplu de utilizare a acestei metode este următorul :

```
URL codeBase = getCodeBase();
play(codeBase, "audio/sound.au");
```

În cele ce urmează va fi prezentat un exemplu care rulează un fișier audio AU, la acționarea unui buton Play.

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

public class SoundApplet extends Applet
{
    Button button;

    public void init()
    {
        BorderLayout layout = new BorderLayout(); setLayout(layout);

        Font font = new Font("TimesRoman", Font.BOLD, 32); setFont(font);

        button = new Button("Play Sound");
        add("Center", button);

        resize(250, 250);
    }

    public boolean action(Event evt, Object arg)
    {
        if (evt.target instanceof Button)
        {
            URL codeBase = getCodeBase();
            play(codeBase, "spacemusic.au");
        }

        return true;
    }
}
```

Controlul sunetelor

Un fișier audio poate fi rulat, oprit sau rulat de un număr de ori. Acest lucru se realizează prin utilizarea obiectului **AudioClip** și a metodelor pe care acesta le pune la dispoziția programatorului **play**, **stop** și **loop**.

Crearea unui astfel de obiect se realizează astfel:

```
AudioClip audioClip=getAudioClip(baseUrl, relLocation);
```

Lansarea metodelor anterior amintite va necesita apelul instanței **AudioClip** și a metodei dorite. Astfel rularea unui obiect AudioClip are forma: **sunetClip.play()**, respectiv oprirea: **sunetClip.stop()** și în fine reluarea rulării până la accesarea unei metode de tip stop se face prin: **sunetClip.loop()**

Aplicatie de utilizare a sunetelor intr-un applet Java

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

public class Sunet2 extends Applet
{
    AudioClip sunetClip;

    public void init()
    {
        GridLayout layout = new GridLayout(1, 3, 10, 10); setLayout(layout);

        Font font = new Font("TimesRoman", Font.BOLD, 24); setFont(font);

        Button button = new Button("Play");
        add(button);

        button = new Button("Stop");
        add(button);

        button = new Button("Loop");
        add(button);
        URL codeBase = getCodeBase();

        sunetClip = getAudioClip(codeBase, "spacemusic.au"); resize(250, 250);
    }

    public boolean action(Event evt, Object arg)
    {
        if (arg == "Play") sunetClip.play();
        else if (arg == "Stop") sunetClip.stop();
    }
}
```

```
else if (arg == "Loop") sunetClip.loop();  
return true;  
}  
}
```

Probleme propuse

1. Scrieți un applet care pornește fișierul SPACEMUSIC.AU de îndată ce appletul este încărcat. Oprirea acestui fișier se realizează numai la activarea unui buton de STOP disponibil utilizatorului.