

TEHNICI AVANSATE DE PROGRAMARE

LABORATORUL 6

1. Fire de execuție:

Limbajul Java suportă *multithreading* prin clase disponibile în pachetul `java.lang` care este pachetul fundamental al limbajului Java.

În pachetul `java.lang` există două clase și o interfață cu care se pot dezvolta programe *multithread*: clasele **Thread** și **ThreadGroup** și interfața **Runnable**.

Există două metode pentru crearea unui *thread*:

- creăm o clasă derivată din clasa `Thread`
- creăm o clasă care implementează interfața `Runnable`

EXP 1: Pentru a crea un fir de execuție folosind clasa `Thread`, etapele sunt următoarele:

- se crează o clasă derivată din clasa `Thread`
- se suprascrie funcția `public void run()` moștenită din clasa `Thread`.
- se instanțiază clasa nou creată
- `thread`-ul instanțiat este pornit prin apelul metodei `start()` moștenită din clasa `Thread`.

```
public class TestThread extends Thread {
    private int nrAfisari = 5;
    private static int nrThread = 0;
    public TestThread() {
        super("firul de executie numarul " + ++nrThread);
        start();
    }
    public String toString() {
        return "#" + getName() + ": " + nrAfisari;
    }
    public void run() {
        while(true) {
            System.out.println(this);
            if(--nrAfisari == 0) return;
        }
    }
    public static void main(String[] args) {
        for(int i = 0; i < 5; i++)
            new TestThread();
    }
}
```

EXP 2: Pentru a crea un fir de execuție folosind interfața `Runnable` se parcurg următoarele etape:

- Se crează o clasă care implementează interfața `Runnable`
- Se implementează metoda `run()` din interfața `Runnable`
- Se crează un obiect din clasa `Thread` folosind un constructor care are ca parametru un obiect de tip `Runnable`
- Se porneste firul de execuție.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class TestControlBar
extends JFrame implements ActionListener, Runnable{
```

```

Thread t;
JProgressBar pb;
JButton A;
public TestControlBar(String titlu){
    super(titlu);
    setSize(200,200);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    Container x = getContentPane();
    //se stabileste gestionarul de pozitionare
    x.setLayout(new FlowLayout());
    A = new JButton("start");
    //se stabileste ascultatorul de evenimente pt A
    A.addActionListener(this);
    pb = new JProgressBar();
    pb.setMinimum(0);
    pb.setMaximum(1000);
    pb.setStringPainted(true);
    x.add(pb);
    x.add(A);
    setVisible(true);
}
public void run(){
    for(int i = 0; i <= 1000; i++){
        pb.setValue(i);
        pb.setString(""+ i/10 + "%");
        try{
            Thread.sleep(10);
            //Thread-ul curent isi intrerupe executia
            //pentru 10 milisecunde
        }
        catch(Exception e){
        }
    }
}
public void actionPerformed(ActionEvent e)
{
    if (t==null){
        t = new Thread(this);
        t.start();
    }
}
public static void main(String args[]){
    TestControlBar fer = new TestControlBar("test");
}
}

```



EXE : Creați pentru exemplul de mai sus clasa Fir care să extindă Thread. Clasa Fir va avea un constructor care va primi ca argument un obiect de tipul Jprogressbar.

2. Applet-uri

Un applet este un program care se încarca de pe un calculator aflat la distanță și care este apoi executat local (în navigatorul Web). Pentru a crea un Applet se va extinde clasa `java.applet.Applet`

Metode ale clasei **Applet**

- `void init()` prima metodă apelată, se execută o singură dată
- `void start()` se apelează atunci când applet-ul devine active
- `void stop()` se apelează atunci când se opreste applet-ul sau când a fost părăsită pagina web care-l conține
- `void paint(Graphics g)` este apelată după `init()` și `start()`.

Alte metode utile ale clasei **Applet**

- `public Image getImage(URL url, Image)`
- `public Image getImage(URL director, String nume)`
- `public URL getDocumentBase()` Returnează URL-ul documentului HTML în care este înglobat appletul
- `public void resize(int lungime, int inaltime)` redimensionează suprafața appletului
- `public AppletContext getAppletContext` returnează “contextul appletului” care permite interacțiunea cu documentul în care conține appletul

Metode ale clasei **AppletContext**

- `public void showDocument(URL url)`
- `public void showDocument(URL url, String target)`
- `public void showStatus(String mesaj)` Afisează mesajul specificat în bara de stare a navigatorului.

EXP 3: Consultați exemplele `AppletReceptor`, `AppletEmitator`



EXE: Creați un applet care permite inserarea unei adrese a unei pagini Web. La acționarea unui buton se va accesa pagina indicată. Se va da posibilitatea utilizatorului, prin butoane radio, să aleagă dacă pagina va fi afișată în aceeași fereastră a navigatorului Web sau într-una nouă.