

Lucrarea de laborator nr. 1

Introducere in Java

Java este un limbaj de programare orientat pe obiect, puternic tipizat ce a fost dezvoltat de Sun Microsystems la începutul anilor 90.

Limbajul Java poate fi utilizat cu succes pentru a proiecta aplicații ce rulează pe un singur calculator sau aplicații ce sunt distribuite prin intermediul serverelor și clienților într-o rețea. Deasemenea, Java poate fi utilizat pentru a dezvolta module sau appleturi pentru aplicații Web.

Java este un limbaj complet orientat pe obiecte, în care secțiunile de cod reprezintă obiecte ce interacționează unele cu altele.

Java este un limbaj facil de utilizat chiar și de programatorii neprofesioniști, prin eliminarea mostenirii multiple, a supraincării operatorilor sau a pointerilor.

O altă caracteristică este portabilitatea, Java fiind independent de mașina pe care lucrează. Abilitatea de a compila o singură dată și a rula pe diverse platforme se realizează prin intermediul compilatorului Java și a mașinii virtuale Java. Natura limbajului Java îl face ideal pentru cei care dezvoltă aplicații pe platforme diferite.

Rularea unei aplicații Java presupune apelarea interpretorului Java (mașina virtuală Java – JVM) pentru fișierul *.class* corespunzător clasei principale a aplicației.

În cazul în care vorbim despre un applet Java vom întâlni modificări numai în privința rulării acestuia, compilarea realizându-se similar cu aplicațiile Java. Applet-urile reprezintă aplicații Java de mici dimensiuni ce pot fi rulate în browsere web. Pentru lucrul cu applet-uri, limbajul Java pune la dispoziția utilizatorilor o altă unealtă numită *appletviewer*, utilizată în general în testare.

Platforma NetBeans IDE (Integrated Development Environment)

Pentru o introducere în fereastra de lucru NetBeans IDE , vom crea o simplă aplicație “Hello World” cu ajutorul consolei Java. Pentru a rula această aplicație, este nevoie de următoarele resurse software:

resurse	Versiune ceruta
NetBeans IDE	version 6.9
Java Development Kit (JDK)	version 6 or version 5

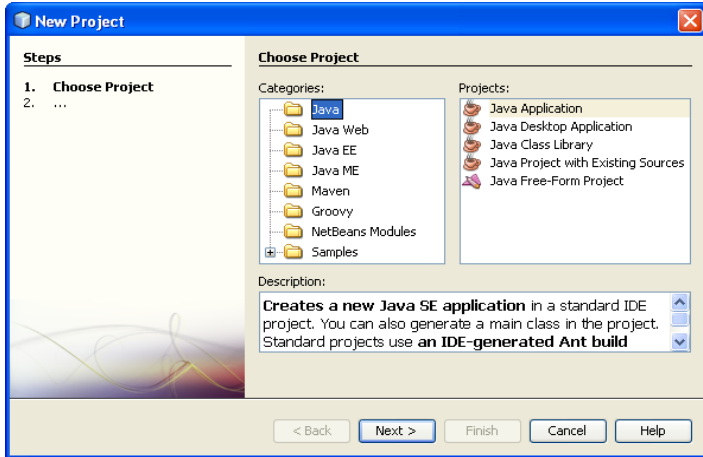
Configurarea proiectului

Pentru a crea un proiect IDE:

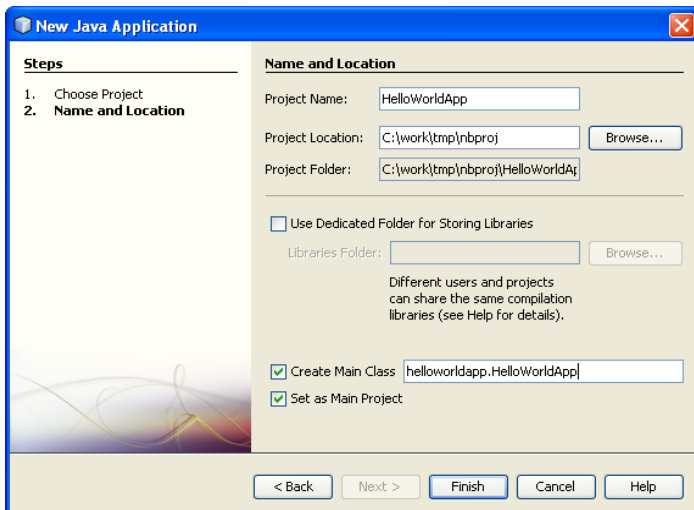
1. Se pornește NetBeans IDE.
2. Se alege din meniul File > New Project (CTRL+SHIFT+N) dupa cum se observă în figura următoare:



3. În fereastra care se afișează se extinde categoria Java și se alege Java Application , apoi se apasă butonul Next.



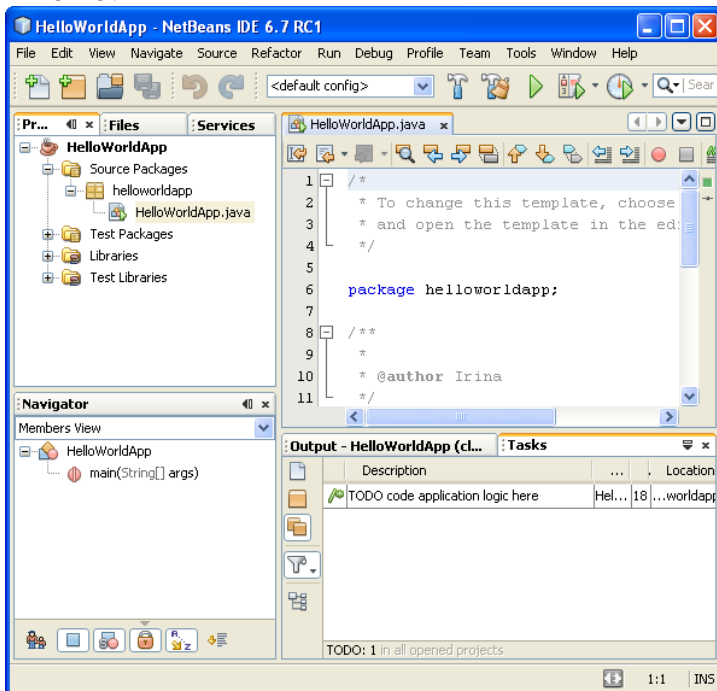
4. În câmpul cu numele și locația paginii, după cum arată figura următoare se vor executa pașii:
- În câmpul Project name scrieți numele aplicației **HelloWorldApp**.
 - În câmpul Create Main Class, tipăriți **helloworldapp.HelloWorldApp**.
 - Lăsați selectată opțiunea Set as Main Project.



5. Se apasă butonul Finish.

Proiectul este creat și deschis în IDE. Ar trebui să se observe următoarele componente:

- Fereastra proiectului, care conține o vedere de arbore a componentelor proiectului, incluzând fișierele sursă, bibliotecile de care depinde codul, ș.a.m.d.
- Editorul sursă- fereastră cu un fișier numit **HelloWorldApp** deschis.
- Fereastra Navigator, care poate fi folosită pentru a trece rapid între elemente în clasa selectată.
- Fereastra activităților (task-urilor), cu erorile de compilare și alte sarcini care sunt marcate prin cuvinte cheie ca XXX și TODO.



Adăugarea codului la fișierul sursă generat

Deoarece în stânga avem căsuța de activare Create Main Clase, a fost selectată opțiunea New project, pentru dvs. A fost creată o clasă schelet. Se poate adăuga mesajul “Hello World!”, mesaj la secvența programării de bază înlocuind linia:

```
public class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

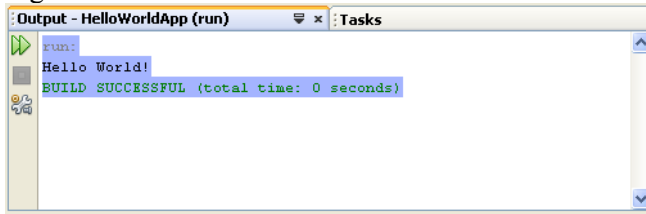
Compilarea și rularea programului

Deoarece IDE-urile compilează pe caracteristica Save, nu este nevoie să se compileze proiectul pentru a rula în IDE. Când s-a salvat un fișier sursa Java, IDE îl compilează automat.

Pentru a rula programul:

- Se alege Run>Run Main Project (F6)

Figura următoare arată ce ar trebui să se vadă:



În acest moment programul funcționează.

Dacă există erori de compilare, ele vor fi marcate cu semne roșii în marginea stânga sau dreaptă a Editorului Sursă. Semnele plasate în marginea stânga indică erori corespunzătoare liniei, iar cele din dreapta indică zona fișierului în care există erori, incluzând erorile în linii care nu sunt vizibile.

Dacă se poziționează mouse-ul peste un semn al erorii se poate afla descrierea acesteia. Se poate apăsa butonul stâng al mouse-ului pentru semnele din dreapta pentru a sări la linia cu eroare.

Construirea și desfășurarea aplicației

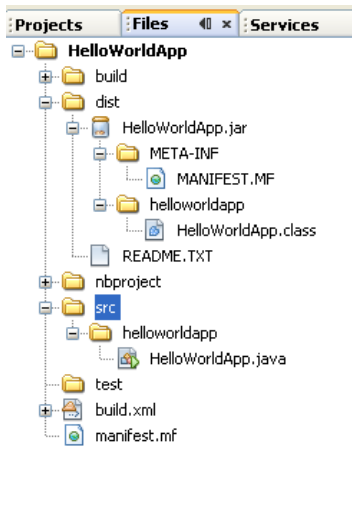
Odată ce avem scrisă și rulată o aplicație, se poate folosi comanda Clean and Build pentru a construi și desfășura aplicația. Când se folosește comanda Clean and Build, IDE rulează pentru a construi o secvență de comenzi care să execute următoarele sarcini:

- Șterge oricare fișier compilat anterior și construiește ieșirile.
- Recompilază aplicația și construiește un fișier JAR, care conține fișierele compilate


Pentru a construi aplicația:

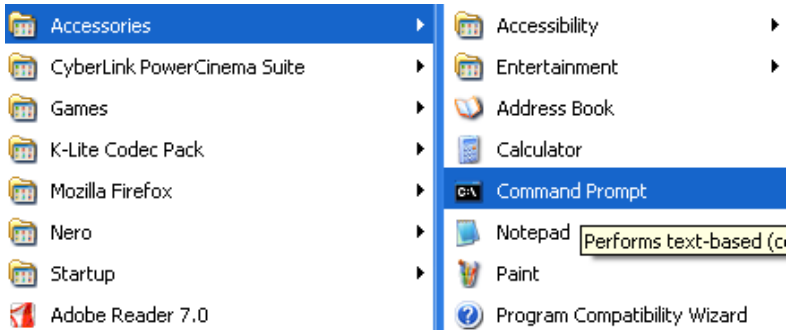
Se alege comanda Run>Clean and Build Main Project (Shift-F11).

Ar trebui să se vadă ieșirile construite deschizând fereastra Files și extinzând nodul HelloWorldApp. Fișierul compilat “bytecod” **HelloWorldApp.class** este pentru a construi subnoduri **build/classes/helloworldapp**. O desfășurare a fișierului JAR care conține clasa HelloWorldApp este în interiorul nodului **dist**.



Rularea unei aplicații/applet Java în consola DOS

Pentru lansarea în execuție a compilatorului Java – javac.exe, respectiv a appletviewer-ului se lansează aplicația Dos – Command Prompt -  Command Prompt .



Prin comenzi specifice Dos, de schimbare a directorului – comenzi de tip change directory – cd nume_director/subdirector/sub-sub-director/.../etc este localizat directorul **jdk1.6.0/bin** –(de exemplu: C:\Program Files\Java\jdk1.6.0_11\bin).

Cu ajutorul unui editor de text, spre exemplu WordPad, NotePad sau pentru eliminarea tuturor problemelor comanda edit de sub MsDos, se editează următorul applet Java:

```
//*****  
// hello_java.java  
//*****  
import java.applet.*;  
//*****  
  
public class hello_java extends Applet  
{
```

```

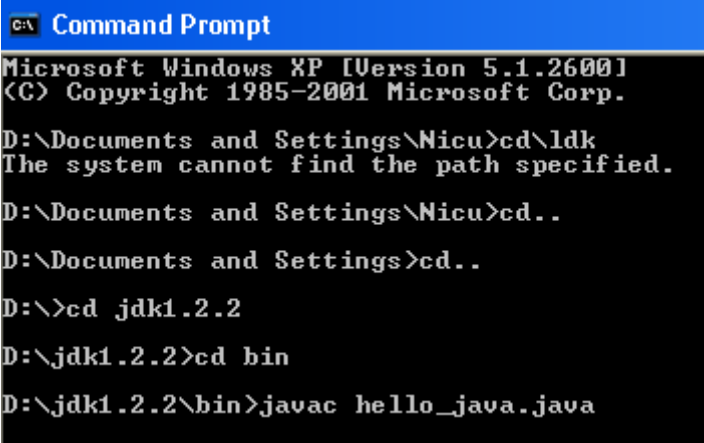
public void init()
{
    System.out.println("Ce mai faci, Java?");
}
}

```

Acest fișier este salvat cu numele `hello_java.java`.

Observație. Datorită problemelor întâlnite la salvarea documentelor, se recomandă folosirea, pentru editarea fișierelor java, respectiv html, editorul de sub DOS. Accesarea acestuia se realizează prin activarea prompter-ului MsDos și introducerea comenzii `edit`. Pentru ușurința abordării se recomandă salvarea fișierelor în directorul `bin`. Orice alt director va implica specificarea, în timpul compilării, a căii la care se află fișierul sursă.

În continuare, apelând la prompter-ul Dos este compilat fișierul java și transformat în fișier *bytecodes*, class. Compilarea se realizează apelând compilatorul de Java : **javac**



```

C:\ Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Documents and Settings\Nicu>cd\jdk
The system cannot find the path specified.

D:\Documents and Settings\Nicu>cd..

D:\Documents and Settings>cd..

D:\>cd jdk1.2.2

D:\jdk1.2.2>cd bin

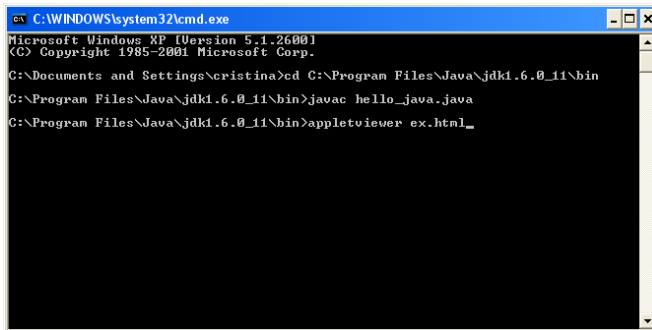
D:\jdk1.2.2\bin>javac hello_java.java

```

La apăsarea tastei Enter și revenirea cursorului, în cadrul directorului `bin` a luat naștere fișierul `hello_java.class`. Pentru activarea acestei clase se creează un fișier html, **ex.html** cu următorul conținut:

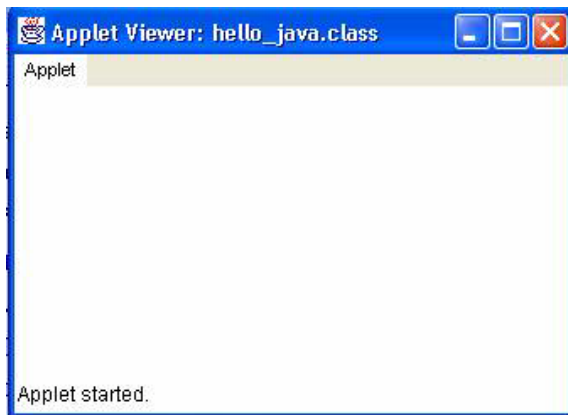
```
<applet code=hello_java.class width=330 height=170> </applet>
```


Apoi este activat appletviewer-ul pentru vizualizarea clasei hello_java, activată în cadrul fișierului html ex.html. Activarea se realizează prin comanda **appletviewer ex.html**, având următorul rezultat:



```
ex C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600.1
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Cristina>cd C:\Program Files\Java\jdk1.6.0_11\bin
C:\Program Files\Java\jdk1.6.0_11\bin>javac hello_java.java
C:\Program Files\Java\jdk1.6.0_11\bin>appletviewer ex.html_
```

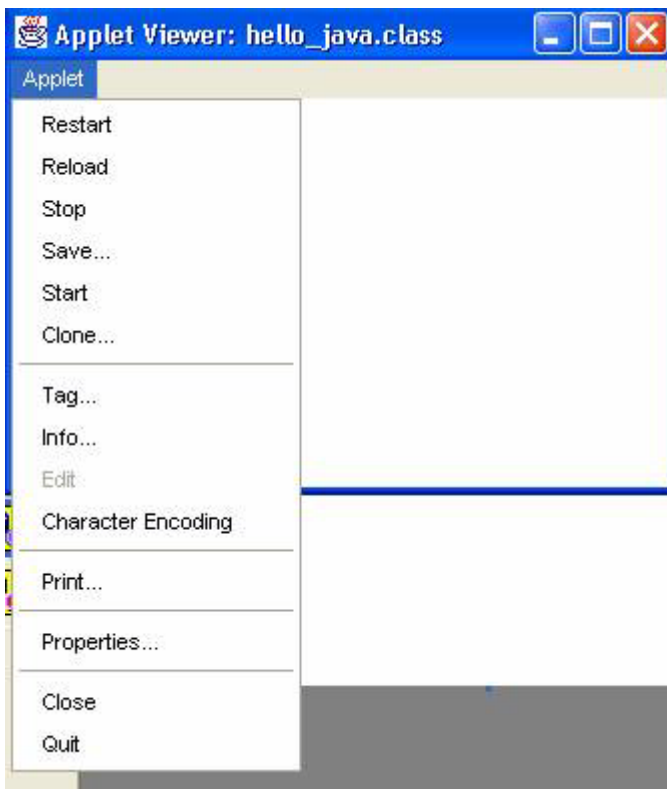
Se observă că la prima rulare este activat Copyright Notice, care va dispărea imediat ce sunt acceptați termenii respectivei note. Este activat un Applet Viewer,



iar rezultatul este trimis imediat la consola Java:

```
D:\jdk1.2.2>cd bin
D:\jdk1.2.2\bin>javac hello_java.java
D:\jdk1.2.2\bin>appletviewer ex.html
Ce mai faci, Java?
```

Oprirea acestui applet se realizează accesând facilitățile oferite de Applet Viewer, respectiv opțiunea Quit care întrerupe rulajul applet-ului:



Observații

- Comentariile în cadrul unui program Java încep cu două backslash-uri //.
- Un program/applet Java începe cu o declarație a claselor care vor fi importate în construcția Applet-ului curent. În acest caz este folosită o singură clasă – import **java.applet.***; Această declarație de import este asemănătoare cu declarația **#include** din C/C++. Asterixul care urmează după declarația import și numele clasei ce va fi încărcate, indică faptul că toate obiectele din clasa respectivă vor fi puse la dispoziția noii clase ce va fi formată în urma compilării – clasa **hello_java**.
- Instrucțiunea **public class hello_java extends Applet** definește numele clasei ce va fi creată în cazul programului Java. Cuvântul **extends** precizează compilatorului că acea clasă creată în cadrul programului extinde o clasă anterior creată.
- Applet-ul este apelat de către un browser Web, începând cu funcția **init()**, care indică locul de intrare în applet.
- Instrucțiunea **public void init()** precizează, prin cuvântul cheie **public** că funcția **init** va fi apelată din exteriorul funcției Applet. Cuvântul cheie **void** indică faptul că funcția **init** nu întoarce nici o valoare.
- Applet-ul are în componența sa o singură *declarație* **System.out.println("Ce mai faci, Java?");**. Afișarea mesajului se realizează prin utilizarea funcției **println**, a obiectului **out**, din clasa **System**.

Chestiuni de studiat

1. Editați programul **hello_java**, **ex.html**. Rulați programele conform indicațiilor din cadrul platformei de laborator. Ce observații aveți? Înlocuiți cuvântul **extends**, din linia a 6-a a

programului cu **extinde**. Ce observați? Corecți și rulați din nou programul.

2. Înlocuiți zona principală a programului, liniile dintre acoladele interioare, cu următoarele instrucțiuni:

```
int i = 7;
int a = 6;
int y = a+i;
System.out.println("Ce mai faci, Java?");
System.out.println("Intregul este " + y);
```

3. Rulați noul program. Modificați mesajele ce intră în componența instrucțiunilor de tip *println*.
4. Introduceți următoarea variantă a applet-ului Hello_Java:

```
/**
 * hello_java.java
 */
import java.applet.*;
import java.awt.Graphics;

public class hello_java extends Applet
{
    public void paint(Graphics g)
    {
        something s;
        s = new something(105);
        g.drawString("Hello Java ! Value "+s, 20,20);
    }
}

class something
{
    private int value;
}

```

```

public something(int value)
    {
        this.value = value;
    }
//_____
public String toString()
    {
        String s;
        s("<" + value + ">");
        return s;
    }
}

```

Rulați aplicația utilizând javac, respectiv appletviewer ex.html. Care sunt modificările care apar comparative cu prima variantă? Folosiți comanda javadoc și studiați rezultatul acestui program.

5. Introduceți următorul applet:

```

//*****
// hello_marquee.java
//*****
import java.applet.*;
import java.awt.Graphics;
//*****

public class marquee extends Applet
implements Runnable
    {
        int x = 0;
        int y = 0;
        int width = 0;
        Thread my_thread = null;

//_____
public void init()
    {
        x = size().width;
        y = size().height/2;
    }
}

```

```

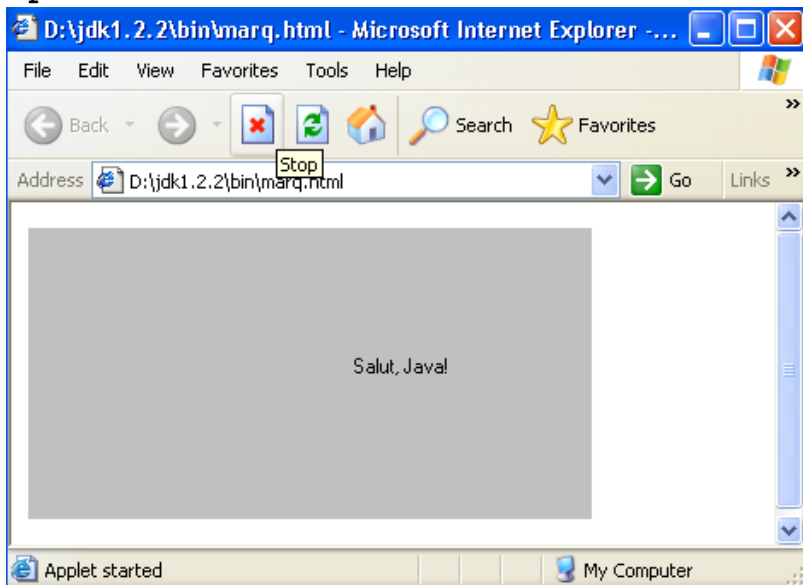
width = x;
}
//_____
public void start()
{
my_thread = new Thread(this);
my_thread.start();
}
//_____

public void run()
{
while(true)
{
    repaint();
    x -=10;
    if(x < 0)
    x = width;
    try
        {
            Thread.sleep(100);
        }
    catch(InterruptedException e)
        {
        }
}
}
}
//_____
public void paint(Graphics g)
{
g.drawString("Salut, Java!",x,y);
}
}

```

Construiți fișierul **exemplu.html** cu următoarea structură:
<applet code=marquee.class width=330 height=170>
</applet>

Compilați appletul și lansați appletviewer-ul pentru **exemplu.html**.



6. Modificați caracteristicile de viteză ale appletului. Modificați cursa textului.
7. **Important.** Salvați toate aplicațiile dezvoltate, fișierele html realizate pe dischetă. Etichetați soft și "hard" discheta cu numele și grupa dumneavoastră. Aceasta va constitui referatul dumneavoastră.