

LABORATORUL 1

1. Crearea și rularea programelor Java din linie de comandă

Pașii ce trebuie urmați pentru crearea unui program Java sunt prezentați schematic în figura de mai jos:

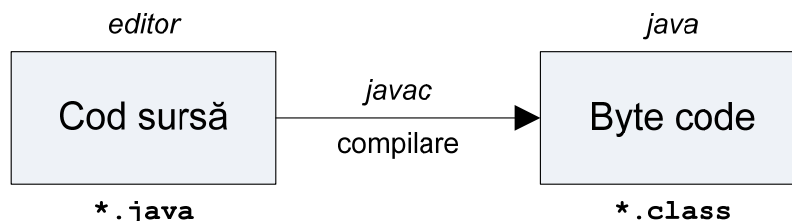


Fig. 1. Etapele necesare creării unui program Java

1.1. Crearea codului sursă

Codul sursă este scris în limbajul Java și rezidă într-unul sau mai multe fișiere text având extensia “.java”. Pentru scrierea programului se poate utiliza orice editor de texte. Dacă se lucrează sub sistemul de operare *Microsoft Windows*, se poate utiliza, de exemplu, aplicația *Notepad*. Dacă se lucrează sub Unix/Linux programul *vi* poate fi folosit pentru scrierea codului sursă.

Figura 2 prezintă codul sursă al unui program Java care afișează pe ecran textul „Hello World din Java!”, așa cum arată el în aplicația *Notepad*.

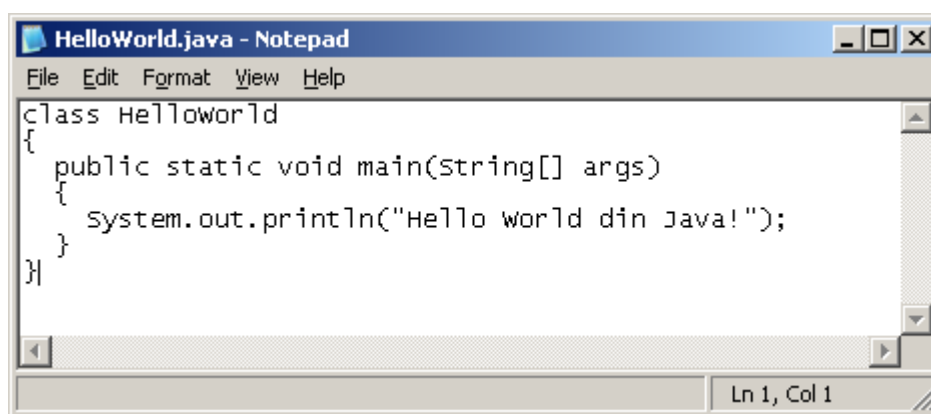


Fig. 2. Editarea programelor Java cu ajutorul aplicației *Notepad* din Windows



Numele fișierului care conține codul sursă al programului trebuie să aibă numele identic cu numele clasei ce conține metoda `main()`.

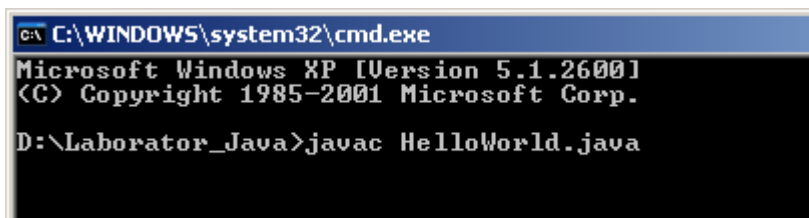
1.2. Compilarea programului

Transformarea codului sursă în codul de octeți (*byte code*) înțeles de JVM (*Java Virtual Machine*) se realizează prin compilarea programului. Pe sistemele Windows acest lucru este realizat de executabilul `javac.exe`, ce poate fi apelat dintr-o fereastră sistem.

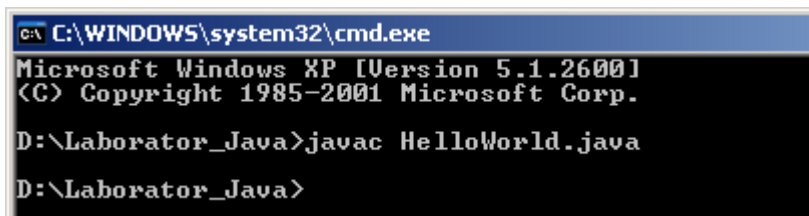
Compilerul Java face parte din pachetul JDK (*Java Development Kit*) care trebuie să fie instalat pe calculatorul pe care se dezvoltă programe Java. Acest pachet poate fi descărcat, gratuit, de pe site-ul companiei *Sun microsystems*. Programele prezentate în continuare au fost scrise și testate folosindu-se versiunea 6 a pachetului JDK. Acest pachet poate fi descărcat de la adresa: <http://java.sun.com/javase/downloads/index.jsp>.

Pentru a compila programul „HelloWorld”, prezentat în figura 2, se deschide o fereastră sistem (*Command Prompt*), în care se scrie următoarea comandă, urmată de tasta *CR* (*Enter*):

```
javac HelloWorld.java
```



Dacă programul a fost compilat cu succes, pe ecran apare din nou „*command prompt*”-ul:



În directorul de lucru apare un nou fișier, numit `HelloWorld.class`, ce conține codul de octeți al programului (așa cum arată figura 3).

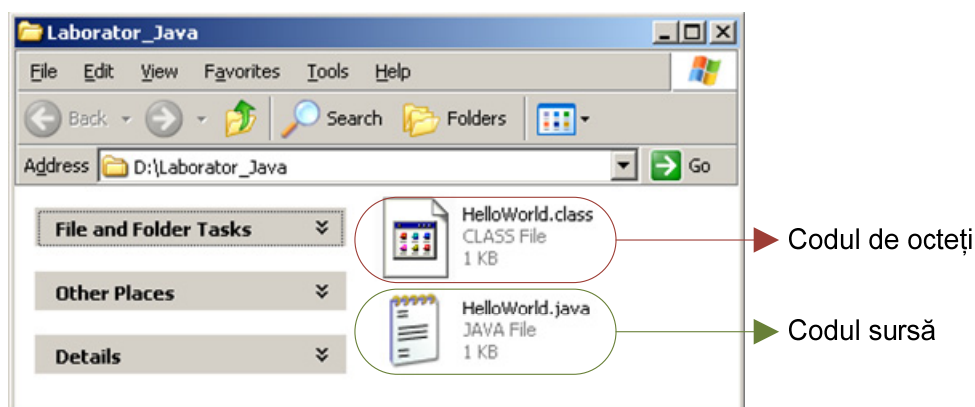


Fig. 3. Codul sursă și fișierul obținut în urma compilării

Dacă programul nu a fost compilat cu succes, în fereastra sistem pot apare diferite mesaje de eroare, precum:

- **'javac' is not recognized as an internal or external command, operable program or batch file.**

Cauze posibile:

1. nu este instalat pachetul JDK.
 - Soluție: instalarea pachetului JDK;
2. este instalat pachetul JDK, dar calea directorului **bin** nu a fost inclus în variabila de sistem **PATH**.
 - Soluție: adăugarea la variabila de sistem **PATH** a directorului **bin**. Acest lucru se poate realiza, în Windows, din fereastra *System properties*, accesibilă prin clic dreapta pe iconița *My Computer* și selecția opțiunii *Properties*, sau din Control Panel (accesibil din *Start Menu* → *Settings*) În secțiunea *Advanced* se efectuează clic pe butonul *Environment Variables*. În fereastra care se deschide, din lista *System variables*, se selectează variabila **PATH**, după care se apasă butonul *Edit*. La sfârșitul valorii variabilei se adaugă caracterul **' ; '**, dacă nu există, după care se trece calea completă către directorul **bin** al pachetului JDK.

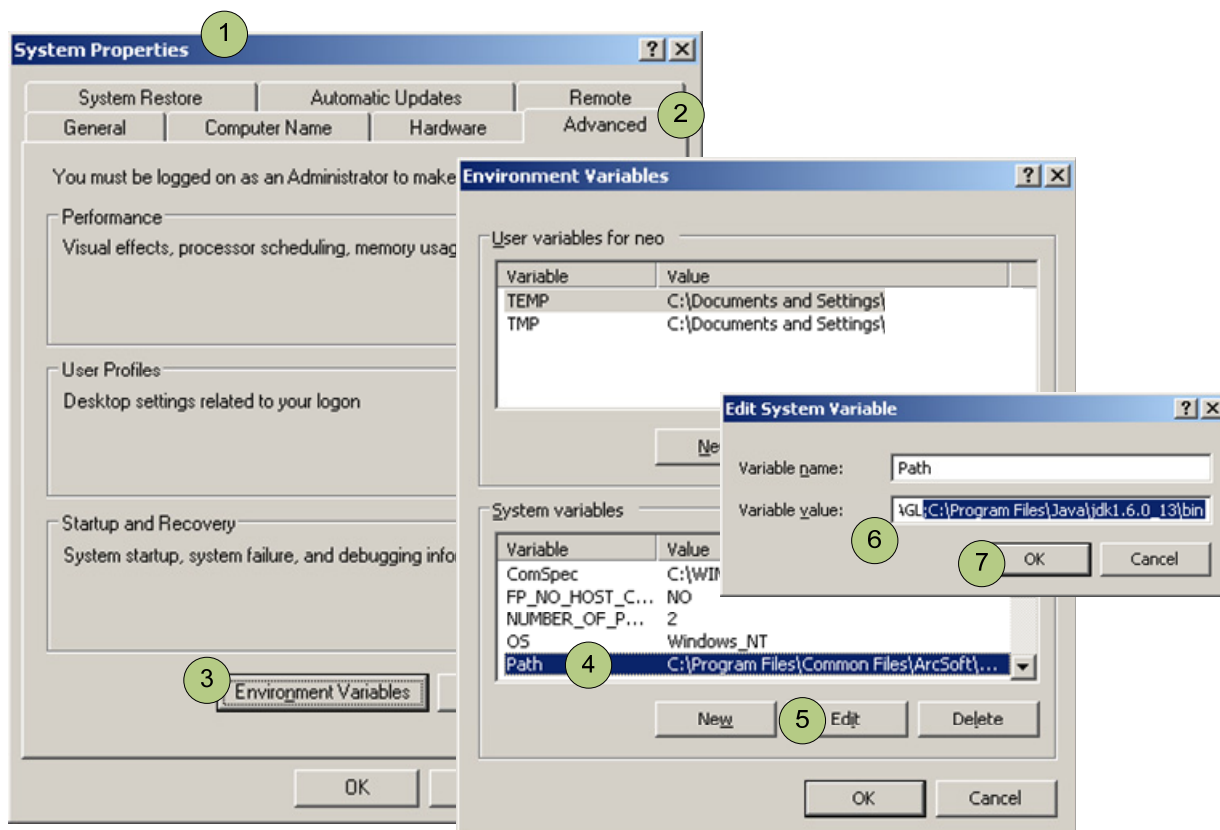


Fig. 4. Includerea directorului **bin** al pachetului JDK în variabila de sistem **PATH**

2. "...should be declared in a file named...."

Cauze posibile:

1. numele fișierului nu este salvat cu același nume cu cel al clasei conținute.
 - Soluție: redenumirea fișierului sau a clasei, astfel încât cele două nume să coincidă.
- **javac:invalid flag:** dacă apare acest mesaj, urmat de o listă lungă de cuvinte care încep cu '-' cauzele posibile sunt:
 1. compilatorul nu poate găsi fișierul sursă Java. Se va afișa directorul curent.
 2. scrierea greșită a numelui fișierului la compilare.
 3. omiterea extensiei fișierului (.java).
- **Hello java:6:';expected**

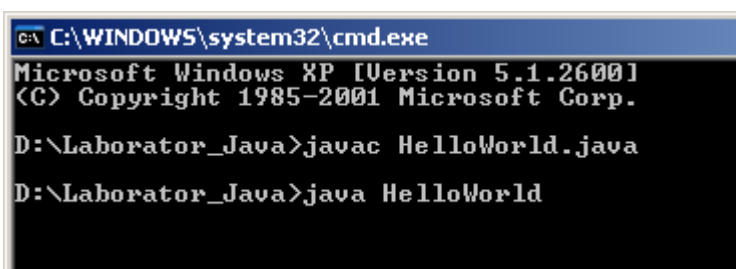
Când o eroare conține cuvântul *expected* urmat de un fragment de program înseamnă că fragmentul respectiv conține o eroare de sintaxă. În acest exemplu, linia 6 este cea care a generat eroarea.

1.3. Lansarea în execuție a programului

Pentru a putea lansa în execuție un program Java, calculatorul gazdă trebuie să aibă instalată mașina virtuală Java (*JVM*). Aceasta este reprezentată de pachetul JRE (*Java Runtime Environment*), care este inclus și în pachetul JDK.

Lansarea în execuție a programului „HelloWorld” se realizează prin comanda:

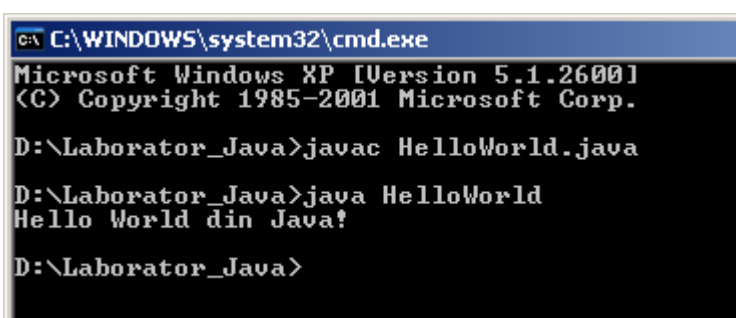
```
java HelloWorld
```



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Laborator_Java>javac HelloWorld.java
D:\Laborator_Java>java HelloWorld
```

În fereastra *Command Prompt* apare mesajul „Hello World din Java!”:



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Laborator_Java>javac HelloWorld.java
D:\Laborator_Java>java HelloWorld
Hello World din Java!

D:\Laborator_Java>
```



La lansarea în execuție a unui program Java din linie de comandă este utilizat fișierul cu extensia „.class”, dar această extensie nu se menționează, ci numai numele fișierului (care trebuie să coincidă exact cu numele clasei ce conține metoda `main()`).

2. Crearea și rularea programelor Java cu ajutorul platformei Eclipse

Eclipse este o platformă multi-scop pentru dezvoltarea de software, scrisă, în mare parte, în limbajul Java, astfel putând rula pe orice sistem de operare actual. Oferă un mediu integrat de dezvoltare (*IDE – Integrated Development Environment*) pentru diverse limbaje de programare (*Java, C/C++, PHP, Python, Perl, Cobol*). Baza codului sursă provine din platforma *VisualAge* dezvoltată de *IBM*, astfel se explică suportul primit din partea acestei companii.

Platforma *Eclipse* se încadrează în categoria programelor gratuite și *open source*. Cea mai recentă versiune, *Eclipse Ganymede*, poate fi descărcată de pe site-ul oficial www.eclipse.org, de la adresa: <http://www.eclipse.org/downloads/>, link-ul *Eclipse IDE for Java Developers*.

Printre facilitățile platformei *Eclipse* merită a fi menționate:

- crearea și gestiunea de proiecte;
- *debuging*;
- completarea automată a codului (*code completion*);
- automatizări pentru operații des utilizate (redenumire, creare de set-ere și get-ere, completarea automată a secțiunii de import).



Toate programele Java prezentate în continuare sunt dezvoltate folosind platforma *Eclipse Ganymede* și *JDK 6* și rulează pe *JRE 6*.

Versiunile platformei *Eclipse*, începând cu versiunea 3.2, sunt denumite după sateliții naturali și artificiali ai planetei *Jupiter* (*Callisto, Europa, Ganymede*). Următoarea versiune, programată a fi lansată oficial în 26 iunie 2009, se va numi *Eclipse Galileo*.

2.1. Interfața de lucru a platformei Eclipse

La pornirea mediului *Eclipse* este cerută calea unui director care va fi spațiul de lucru al platformei. Aici se vor salva proiectele și vor fi stocate datele necesare rulării proiectelor în lucru.

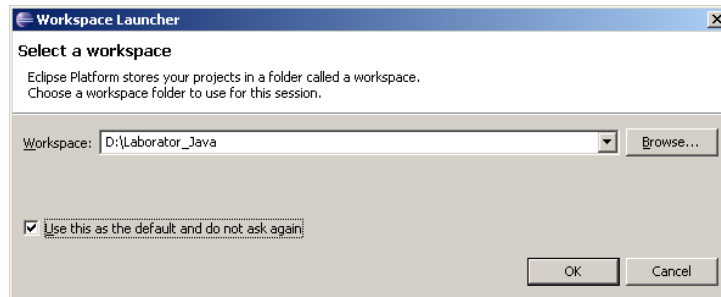


Fig. 5. Selecția spațiului de lucru

Dacă se utilizează același spațiu de lucru pentru toate proiectele, se poate bifa căsuța de validare *Use this as the default and do not ask again*, pentru a elimina această întrebare la fiecare pornire a platformei.

Ferestrele vizibile din interfața *Eclipse* sunt grupate logic în *perspective*.

O perspectivă indică numărul de ferestre vizibile, poziția lor și opțiunile vizibile în aceste ferestre, în funcție de limbajul utilizat în proiectul deschis. În capturile de ecran prezentate în acest îndrumar se va folosi perspectiva Java. Perspectiva curentă se poate schimba de la butoanele dedicate situate în partea dreaptă-sus a *workbench*-ului *Eclipse*.

Figura 6 prezintă ferestrele vizibile din perspectiva Java și semnificația lor.

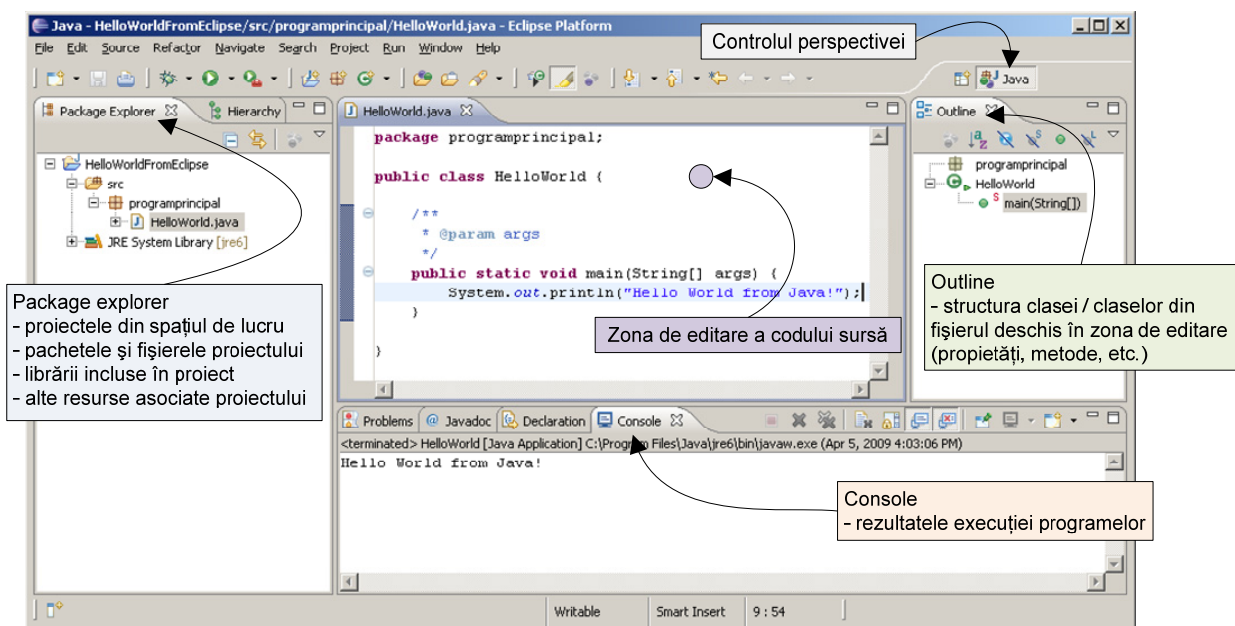


Fig. 6. Componenta perspectivei Java

2.2. Crearea unui proiect Java

Din meniul *File*→*New* se alege opțiunea *Java Project*.

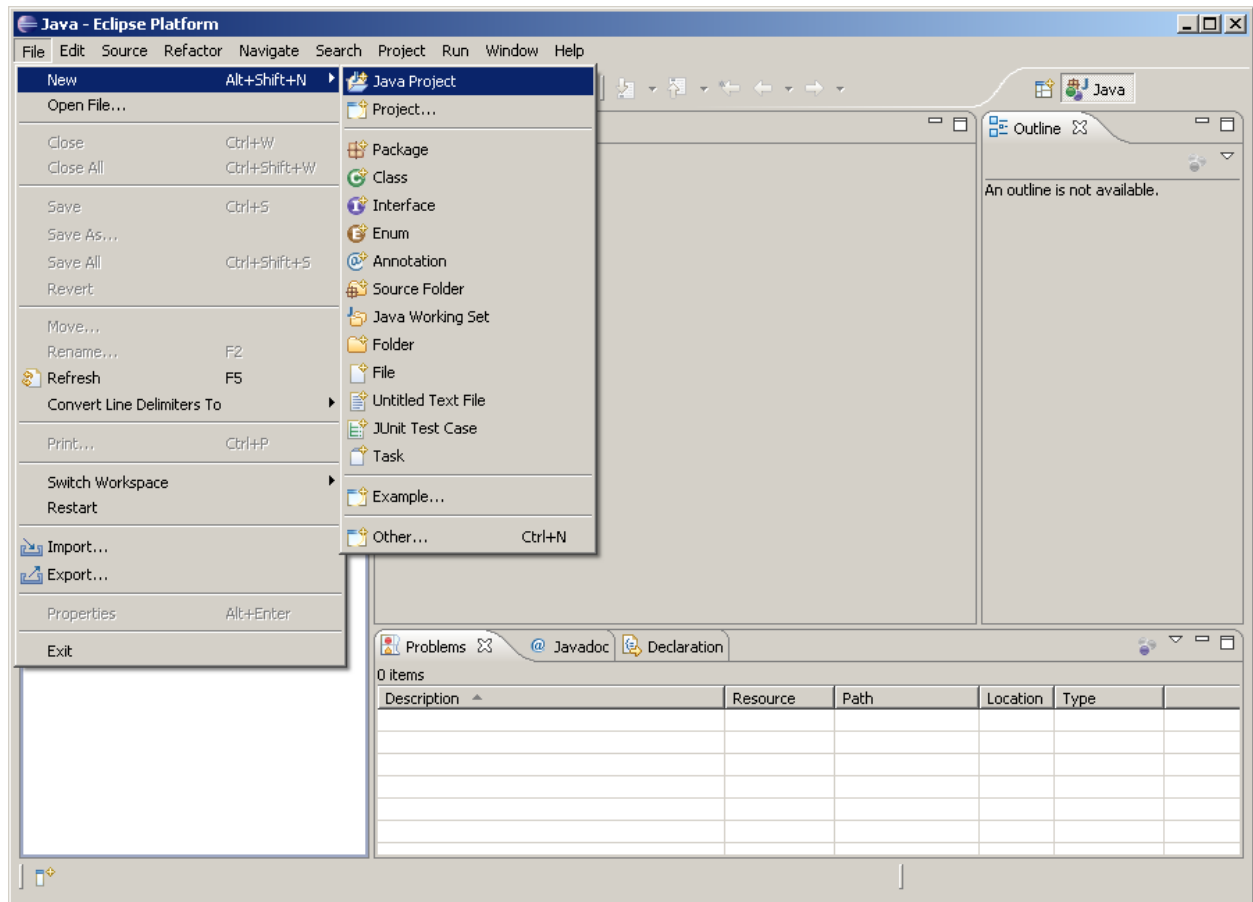


Fig. 7.a. Crearea unui proiect Java în Eclipse

În fereastra care se deschide trebuie specificat numele proiectului. Este indicat să fie selectată opțiunea *Create separate folders for sources and class files* din secțiunea *Project layout*, pentru a nu amesteca fișierele ce conțin codul sursă cu cele ce conțin codul de octeți, executabil. Pentru un proiect complet nou este bine să se aleagă opțiunea *Create new project in workspace* a secțiunii *Contents*. Apăsarea butonului *Next* va determina accederea la mai multe opțiuni, cum ar fi importul unor librării suplimentare, modificarea directoarelor unde vor fi salvate sursele și codul de octeți al proiectului, etc.

Crearea și deschiderea în *workbench* a noului proiect se realizează prin apăsarea butonului *Finish*.

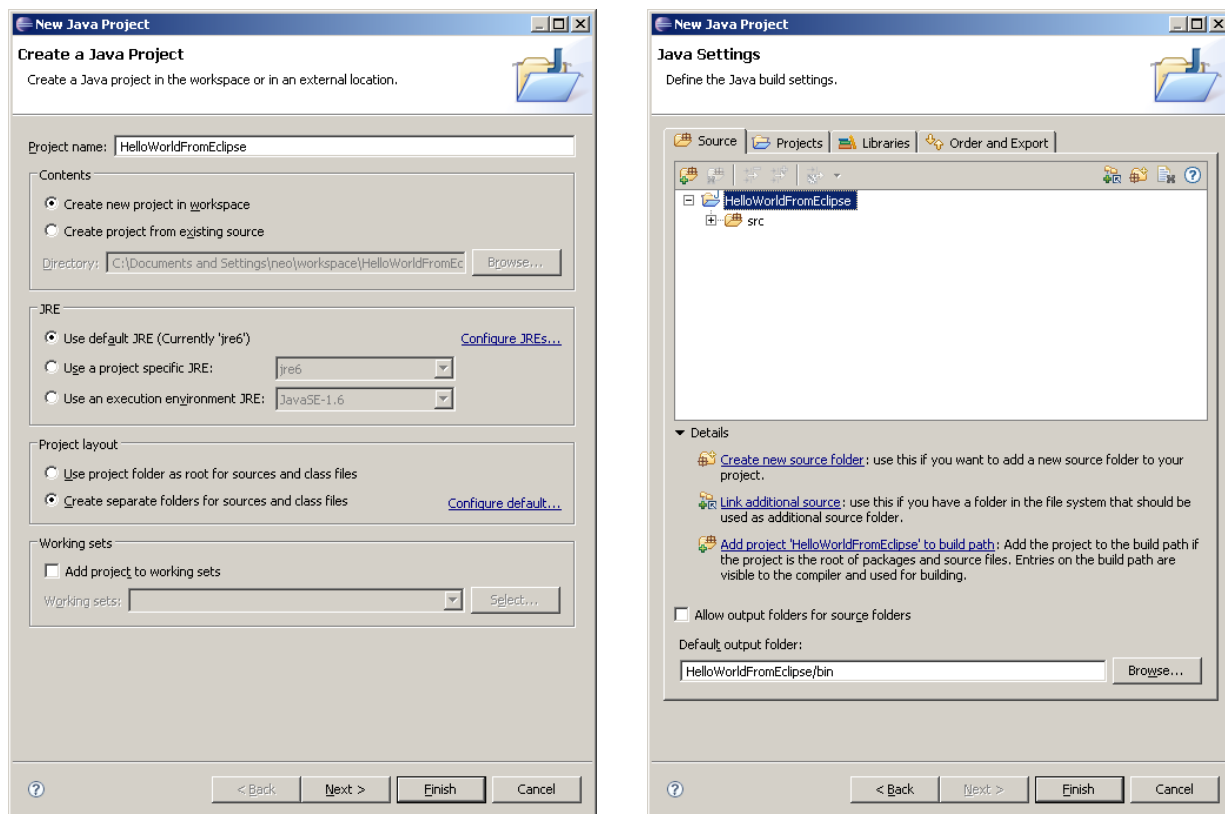


Fig. 7.b. Crearea unui proiect Java în Eclipse

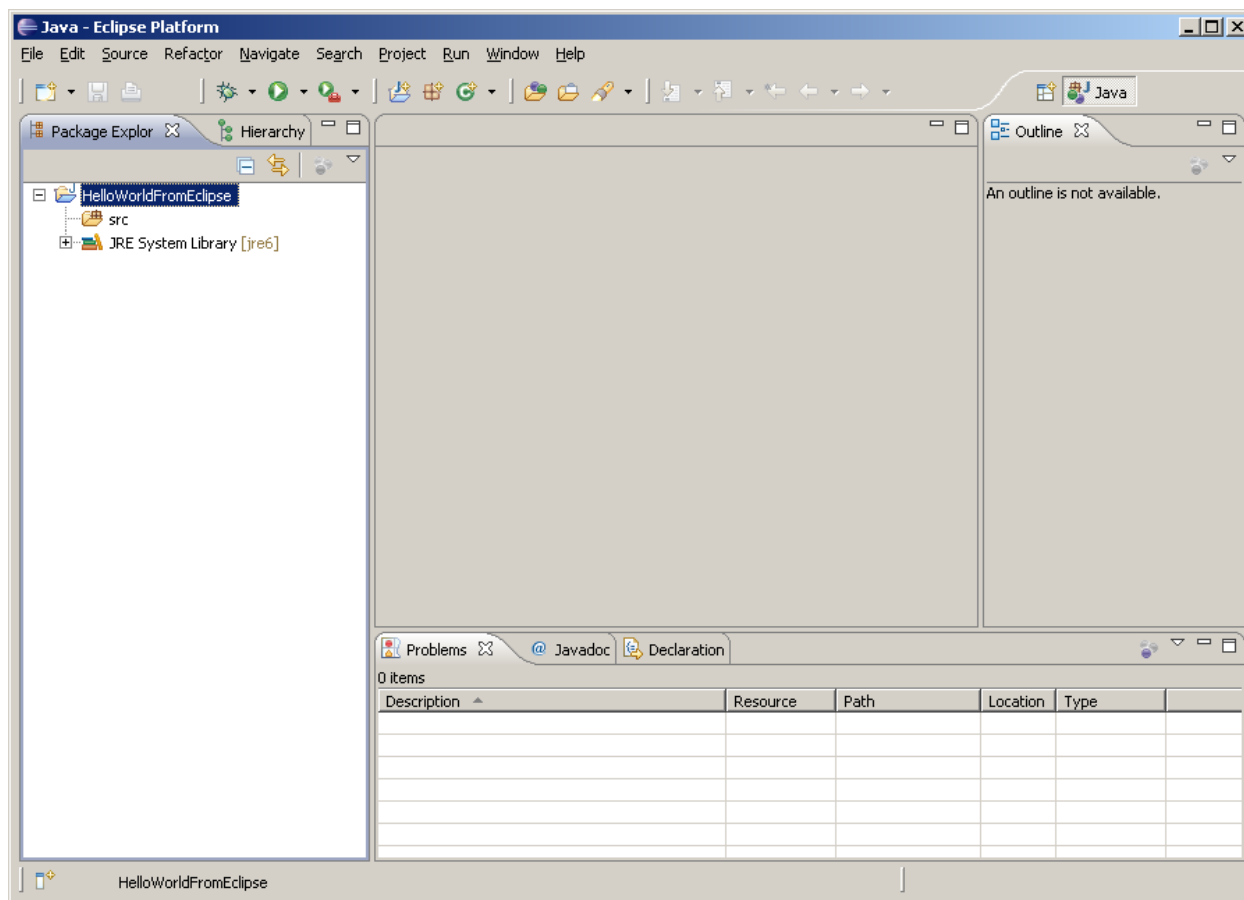


Fig. 8. Proiect nou deschis în workbench-ul platformei Eclipse

2.3. Crearea pachetelor

Clasele unui proiect Java sunt grupate, de regulă, în *pachete*. Criteriile de grupare țin de funcțiile pe care le îndeplinesc acele clase în proiect (lucrul cu fișiere, accesul la baze de date, comunicația prin rețea, etc.). De asemenea, pachetele asigură și controlul numelor și al vizibilității. Fizic, un pachet este un director al proiectului.

Accesul la clasele dintr-un pachet se face prin utilizarea instrucțiunii `import` la începutul codului sursă. Instrucțiunea `import` trebuie să conțină numele pachetelor și ale claselor care vor fi folosite în codul sursă.

Crearea unui pachet în *Eclipse* se realizează prin clic dreapta pe numele proiectului, iar din meniul care apare se alege *New→Package*. În fereastra ce se deschide se specifică numele noului pachetului, după care se apasă butonul *Finish*.

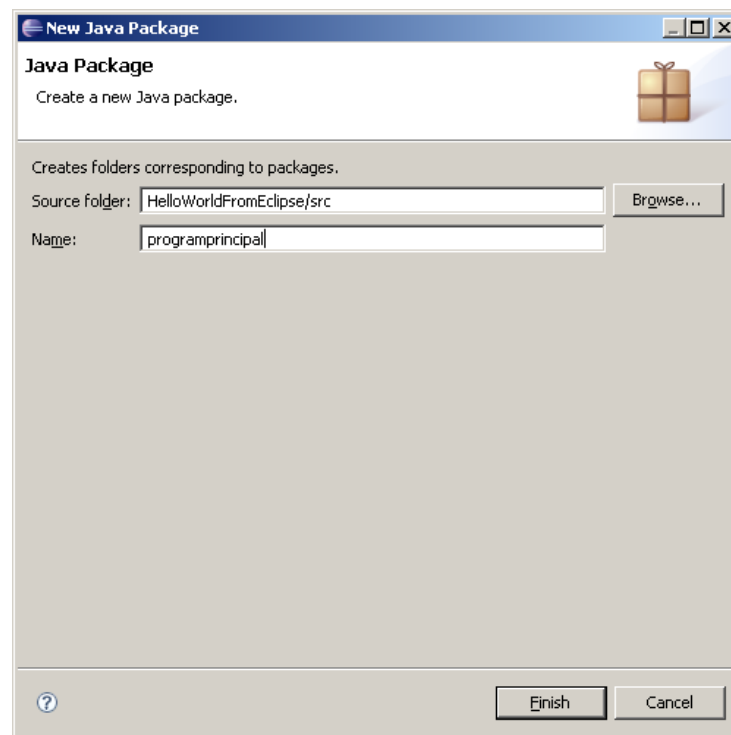


Fig. 9. Creare pachetului *programprincipal*

Un pachet poate conține alte pachete, care, la rândul lor, pot conține alte pachete. În cazul acesta se efectuează clic pe numele pachetului în care se face adăugarea, după care se parcurge aceleași etape prezentate mai sus.

2.4. Crearea claselor

Clasele aparțin pachetelor, prin urmare adăugarea se va face prin clic dreapta pe numele pachetului; se selectează *New→Class*. În fereastra care se deschide se completează numele clasei și se selectează opțiunile pentru aceasta.

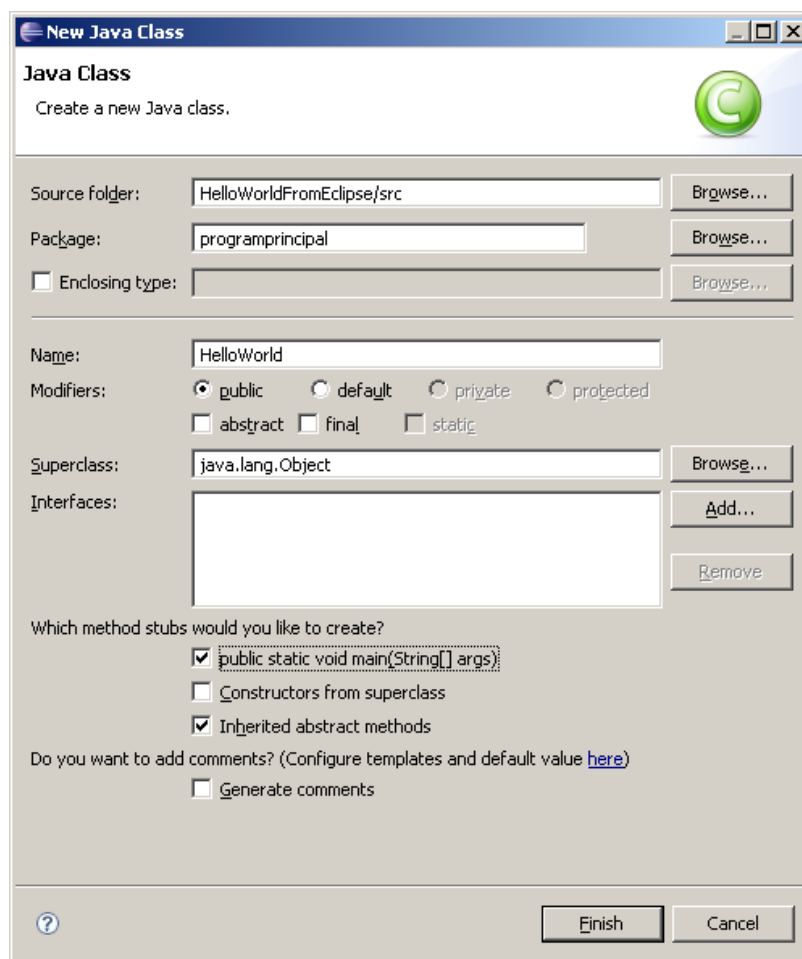


Fig. 10. Crearea unei clase

Dacă se dorește ca clasa creată să conțină metoda `main()`, atunci se bifează opțiunea *public static void main(String[] args)* din secțiunea *Which method stubs would you like to create?*. Se recomandă ca modificatorul de acces al claselor să fie cel `public` (secțiunea *Modifiers*).



Se recomandă respectarea următoarei convenții de notare: numele claselor încep întotdeauna cu literă mare. Dacă numele conține mai mulți atomi lexicali (cuvinte), fiecare dintre ei încep cu literă mare. Mai multe detalii despre convenția de notare Java a identificatorilor și beneficiile aduse de respectarea acestora se găsesc la adresa: [http://en.wikipedia.org/wiki/Naming_conventions_\(programming\)#Java_language](http://en.wikipedia.org/wiki/Naming_conventions_(programming)#Java_language)

După apăsarea butonului *Finish*, este deschis automat fișierul ce conține declarația noii clase, așa cum se vede în figura 11.

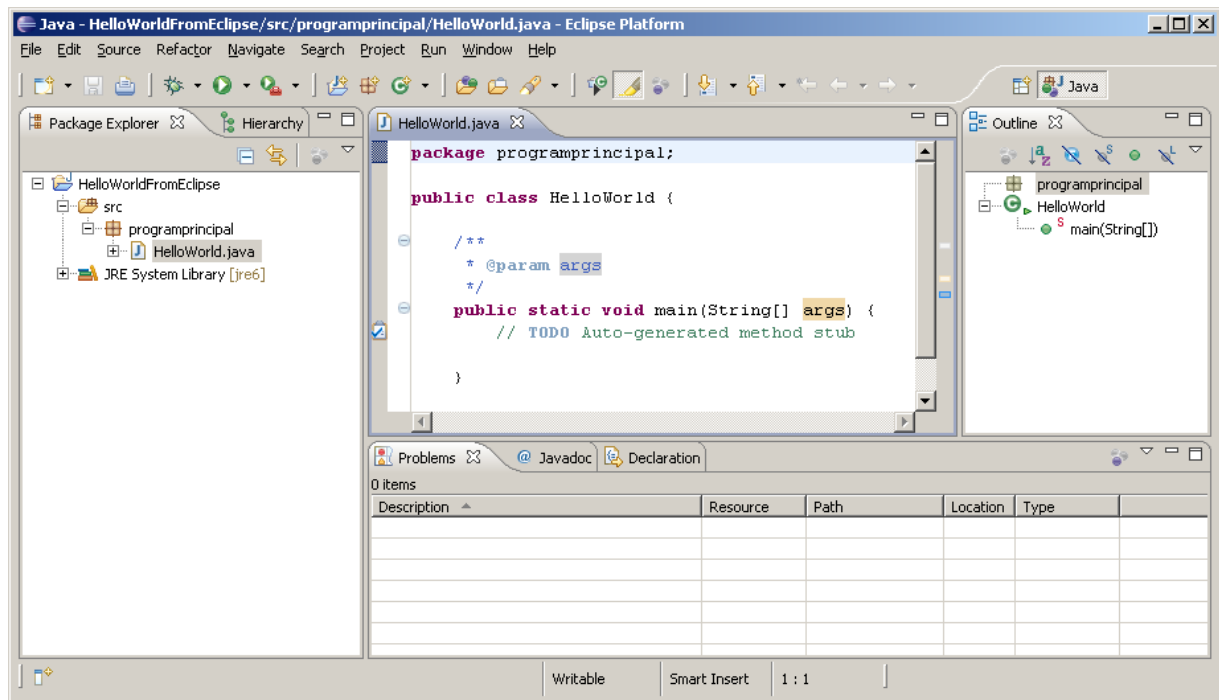


Fig. 11. Clasa generată automat de *Eclipse*

2.5. Aplicația Hello World

Pentru afișarea mesajului “*Hello World din Java!*” se folosește instrucțiunea:

```
System.out.println("Hello World din Java!");
```

Această instrucțiune trebuie plasată în interiorul metodei `main()` (care este echivalentul programului principal: execuția programului începe cu această metodă).

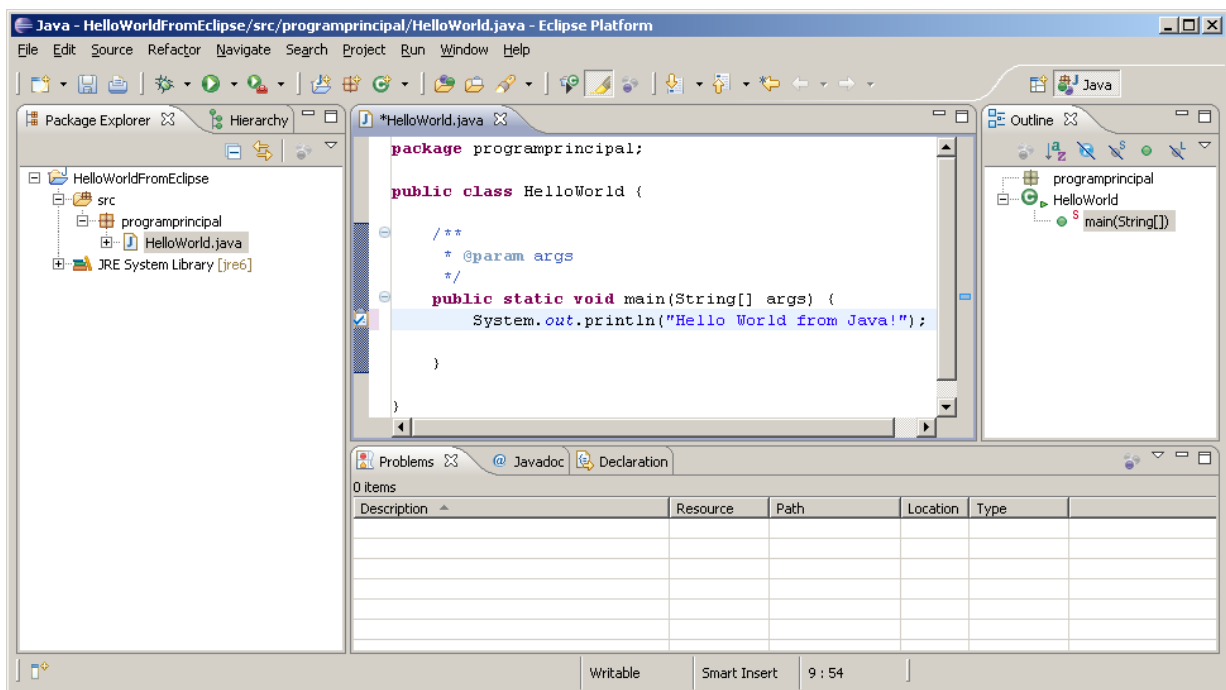



Fig. 12. Aplicația Hello World



Deoarece instrucțiunea `System.out.println()` este foarte uzitată, editorul platformei Eclipse oferă o scurtătură pentru scrierea ei. Tastarea textului `syso`, urmat de combinația de taste `CTRL+Space` determină introducerea instrucțiunii `System.out.println()`.

Lansarea în execuție a programului se poate face din meniul *Run*→*Run*, prin combinația de taste `CTRL+F11` sau prin apăsarea butonului  din bara de instrumente.

Înainte de lansarea în execuție a unui program este recomandabil să salvați fișierele pe care le-ați modificat. Oricum, mediul *Eclipse* vă va cere permisiunea de a salva aceste fișiere înainte de lansarea în execuție.

Rezultatul rulării programului este afișat în fereastra *Console*, aflată în parte de jos a workbench-ului.

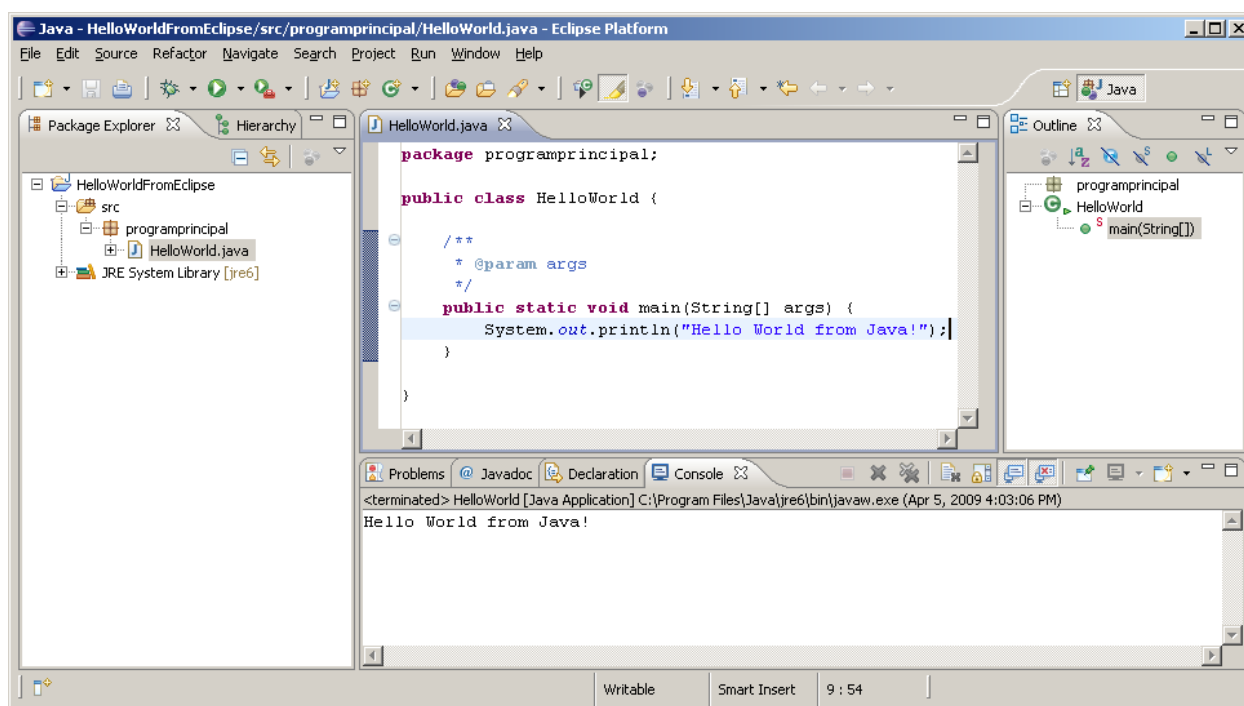
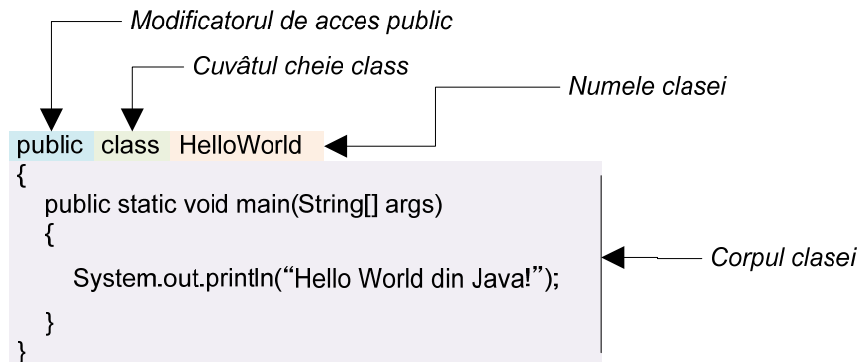


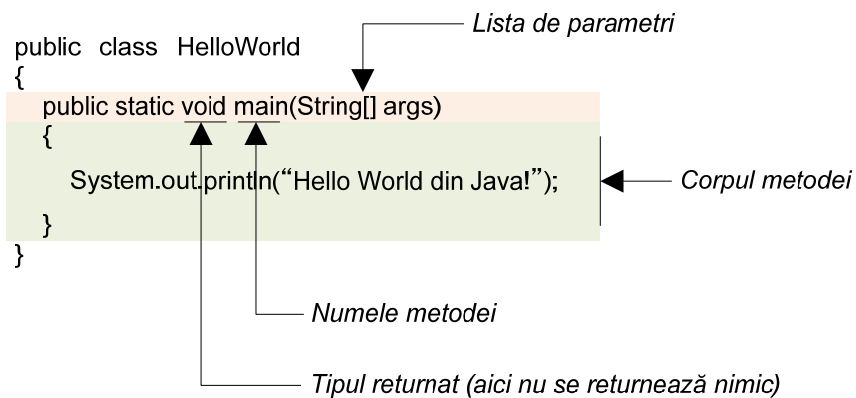
Fig. 13. Rezultatele rulării sunt afișate în fereastra *Console*

3. Structura unui program Java

3.1. Definiția clasei



3.2. Definiția metodelor



3.3. Instrucțiuni

