

## *Introducere in Java*

### **Aparitia programului Java**

În 1990, unei echipe de programatori condusă de James Gosling i-a fost încredințat un proiect destinat controlului aparaturii electronice. Gosling și echipa sa lucrând la compania Sun Microsystems au început proiectarea sistemului utilizând C++, unul dintre limbajele cele mai răspândite, datorită naturii sale de limbaj orientat pe obiecte. Din păcate Gosling a realizat destul de repede că acest limbaj nu era potrivit pentru aplicația propusă. Principalele probleme de care aceștia s-au lovit au fost aspectele complicate ale limbajului C++ în lucrul cu clase și modul de gestionare a memoriei. Decizia lui Gosling s-a materializat prin crearea unui nou limbaj, simplificat care să evite problemele întâlnite în cadrul libajului C++.

Pe de alta parte Gosling a dorit simplificarea limbajului C++, păstrând elementele sintactice de bază, dar și particularitățile de limbaj orientat pe obiecte. Astfel, atunci când acesta a conturat noul limbaj a folosit C++ ca model, modificând toate elementele limbajului C++ care nu au putut conduce la bune rezultate în cadrul proiectului său destinat controlului aparaturii electronice. Numele limbajului creat de Gosling în cadrul proiectului său a fost Oak - Stejar. (Ca o anecdotă se spune că numele Oak i-a venit lui Gosling atunci când, gândindu-se la proiect, privea în gol un stejar care se vedea de la fereastra biroului său.)

Oak a fost folosit prima dată în așa-numitul proiect Green, proiect în care se încerca controlul tuturor sistemelor electronice utilizate într-o casă. Acest sistem de control permitea utilizatorului să comande o listă de dispozitive, cum ar fi televizoarele, lumina, video recorder-ul, telefonul, toate prin intermediul unui minicomputer numit \*7 (Star Seven). Sistemul \*7 folosea un ecran touch-sensitive care oferea utilizatorului posibilitatea de a selecta și controla dispozitivele incluse în cadrul aplicației.

#### **Observație**

Ecranul proiectului \*7 includea o serie de figuri animate, printre care și Duke (acum considerat ca mascota programului Java ).Pe site-ul Sun Microsystems se întâlnesc o multime de Duke,(Figura 1.1), acesta fiind și personajul principal în multe applete Java, de pe server-ul companiei Sun.

[Figura 1.1 : Duke a devenit mascota programului Java.](#)

Următorul pas al limbajului Oak a fost un proiect Video-On-Demand (VOD), în care

limbajul a fost folosit ca bază pentru soft-ul care controla un sistem de televiziune interactiv. Chiar dacă nici \*7 nici proiectul VOD nu au condus la forma actuală a limbajului Java, ele au oferit limbajului Oak o șansă să se dezvolte și să se maturizeze.

Șansa limbajului Java s-ar putea datora tocmai faptului că acesta nu a fost dezvoltat în laboratoarele unei universități sau companii orientată exclusiv pe programare, ci într-un mediu în care inginerii electroniști, automatiști erau dominanți. Pentru a explica această idee trebuie să remarcăm faptul că dacă un programator descoperă o problemă de programare acesta se va orienta spre o altă metodă sau artificiu care să soluționeze problema. Un inginer nu poate să își permită ca în cazul unui bug al programului să lanseze o altă variantă, cu indicativ superior, variantă care să explice de ce toaster-ul controlat de prima variantă s-a ars!!!! În aceste condiții soluția acestuia este să elimine acele elemente generatoare de erori, prin simplificare și robustețe a programelor.

Pe de altă parte, numeroasele variante constructive de aparatură electronică, trebuiau controlate utilizând un pachet soft cu o portabilitate deosebită, care să poată rula în aceleași condiții pe toată diversitatea de microchip-uri implicate în aparatura electronică casnică.

Între timp Sun a descoperit că numele "Oak" a fost deja folosit de un alt produs soft, ceea ce a condus la schimbarea numelui în Java. (Tot ca o anecdotă se pare că numele Java nu a implicat un efort intelectual prea mare din partea celor care l-au ales, acest nume fiind primul care inclus în baza de căutare nu era rezervat)

Cel mai important aspect al limbajului, Java este că acesta este o platforma neutră, ceea ce înseamnă că programele dezvoltate sub acest limbaj pot rula pe orice calculator, fara nici o schimbare. Aceasta independență de platformă s-a datorat formatului special pe care îl au programele Java compilate. Acest format numit byte-code, poate fi citit și executat pe orice computer care dispune de interpretorul Java. Desigur că interpretorul Java trebuie scris special pentru sistemul sub care rulează.

În 1993, după ce World Wide Web-ul a transformat Internet-ul bazat pe fișiere text, într-un mediu bogat în grafică, echipa Java a realizat că limbajul lor este perfect pentru programarea în Web. Ei au pus la punct conceptul de Web applets, care sunt de fapt mici programe ce pot fi incluse în paginile Web, mergând chiar mai departe la crearea unui browser Web în anul 1994 numit inițial WebRunner, redenumit apoi HotJava (câtă agitație în lumea numelor produselor soft!!!!).

În a doua jumătate a anului 1995, Sun Microsystems a lansat în mod oficial limbajul Java. Acest "nou" limbaj a fost rapid îmbrățișat ca o puternică unealtă în dezvoltarea aplicațiilor Internet. Netscape Communications, cel care a lansat popularul Netscape Navigator Web browser, a adăugat suportul Java începând cu varianta Netscape Navigator 2.0. Alți developeri de soft Internet au urmat cu repeziciune, printre aceștia numărându-se fără îndoială și Microsoft, care a oferit suportul java începând cu varianta Internet Explorer 3.

Dupa mai mult de 5 ani de căutari , Java și-a găsit în sfârșit locul.

## **Scurtă introducere în lumea WEB**

Chiar dacă marea majoritate a cititorilor este familiarizată cu acest concept o scurtă trecere în revistă a principalelor elemente definitorii, este considerată binevenită. World Wide Web-ul este o colecție imensă de documete de tip hypertext interconectate prin Internet. Un document hypertext este un document care conține legături - *links* sau *hot links* - către alte documente. Aceste link-uri sunt reprezentate în cadrul documentului sub forma unor cuvinte, litere cu aspect diferit, concretizat de utilizarea unei culori diferite de culoarea contextului sau de efecte speciale de tip intermitent, text care se deplasează sau font diferit. Deseori aceste link-uri pot fi realizate și prin utilizarea unor imagini sau animații pe care utilizatorul accesându-le cu ajutorul mouse-ului, activează legătura către documentele indicate de legătură.

Deoarece atât utilizatorii de astfel de documete sunt extrem de numeroși, cât și creatorii colecția de documete hypertext se îmbogățește cu elemente mai importante (ultimele evenimente din lumea politicului, artei, științei) sau mai puțin importante (ultimele "urări" pentru ocazii nu tocmai ortodoxe) în fiecare fracțiune de secundă.

Din fericire motoarele de căutare își aduc o contribuție deosebită în identificarea informațiilor utile unui anumit utilizator. Acestea folosind o serie de metode de căutare identifică adresele care conțin anumite cuvinte cheie care corespund cerințelor clientului. Cele mai utilizate motoare de căutare sunt [www.yahoo.com](http://www.yahoo.com) , [www.google.com](http://www.google.com) , [www.Infoseek.com](http://www.Infoseek.com) , [www.Excite.com](http://www.Excite.com) . Chiar dacă în această lume poate accede oricine este capabil să scrie un document hypertext, ceea ce conduce la un haos desăvârșit în ceea ce privește regulile de organizare a documentelor hypertext, pentru asigurarea comunicației sunt impuse două standarde:

- Hypertext Transfer Protocol - http : - acesta este un protocol de comunicație care indică modul prin care documentele sunt vehiculate în interiorul Internet-ului.

Prin <b>protocol</b> se înțelege un set de reguli care trebuie respectate de calculatoare referitor la comenzile generate de client și la modul în care serverul răspunde. Cele mai populare protocoale de comunicație existente în Internet sunt <b>HTTP</b> , File Transfer Protocol - <b>FTP</b> , Simple Mail Transfer protocol - <b>SMTP</b> .
---

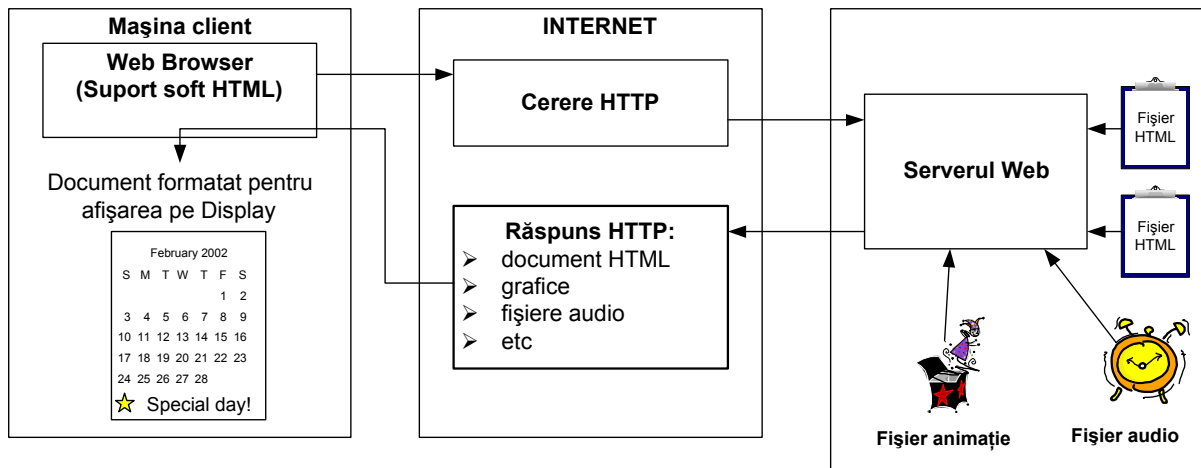
- Hypertext Markup Language - HTML - acesta este un limbaj care specifică modul de redactare și sintaxa unui document hypertext.

Adresele, resursele, din Internet sunt definite prin intermediul unui identificator unic numit Uniform Resource Locator - URL. Acesta specifică protocolul pentru a prelua un document, cât și localizarea acestuia. Un exemplu de URL cuprinde următoarele elemente:

http: //robotics.ucv.ro :80 /students/courses/ java.html

Protocolul HTTP	Adresa serverului	Numărul portului de comunicare	Directorul în care se află documentul	Numele documentului HTML
-----------------	-------------------	--------------------------------	---------------------------------------	--------------------------

Protocolul HTTP este implementat în cadrul softului de pe server, dar și pe mașina utilizatorului sub numele Web server sau HTTP server pentru primul, respectiv Web browser - pentru client. Schimbul de informații este reprezentat schematic în [Figura 1.4](#).



Unele servere generează pagini HTML utilizând programe CGI - Common Gateway Interface. Acest ansamblu CGI+HTML permite utilizatorilor să completeze anumite chestionare cu date referitoare la e-commerce sau sondaje de opinie, generând informațiile cerute sau furnizate în mod dinamic. O aplicație extrem de interesantă este cea oferită de serviciul DHL, [www.dhl.com](http://www.dhl.com), prin care poate fi urmărit un pachet din momentul în care acesta a fost înmănat curierului sau oficiului DHL, până în momentul în care acesta a fost livrat.

Chiar dacă răspunsul oferit de aplicațiile CGI pare a fi dinamic, în realitate documentele oferite sunt statice, informațiile fiind numai selectabile dinamic în funcție de cerințele clientului. Un alt dezavantaj este legat de faptul că atunci când este creat un document HTML de către programul CGI este lansat un program care executat de către mașina gazdă, încarcă server-ul încetinindu-i viteza de răspuns la alte solicitări.

Un alt aspect, care trebuie subliniat în lumea actuală a Internet-ului, este abundența de efecte, de animație, de imagini și sunete care întregesc aspectul unei pagini HTML cât mai interesante. Acest lucru însă este consumator de timp și spațiu, fapt care conduce la reducerea vitezei de acces și implicit la micșorarea numărului de clienți care pot accesa un server.

Spre deosebire de situațiile prezentate până acum, Java oferă o soluție mult mai avantajoasă. Protocolul de comunicare adresează de fapt un applet Java - care este un program integrat în cadrul documentului HTML. Spre exemplu, dacă se dorește urmărirea

în timp real a unor evenimente - dinamica tranzacțiilor bursiere , Java oferă soluția optimă, prin faptul că poate implementa un protocol de interogare în timp real a unor informații, prin specificarea unei adrese și a comenzii GET , fără a mai fi nevoie de specificarea unor noi elemente în cadrul protocolului standard în care rulează appletul - HTTP.

## **Caracteristicile limbajului Java**

Oricine își pune întrebarea "De ce Java este considerat un instrument atât de puternic pentru dezvoltarea aplicațiilor Internet?". Așa cum s-a arătat anterior Java este o versiune simplificată a limbajului C++. Oricine a încercat să învețe C++ s-a lovit de faptul că așa zisa simplificare a limbajului C a condus la apariția unor dificultăți chiar și pentru programatorii profesioniști ai limbajului C.

Potrivit lui Sun Microsystems, Java este "simplu, limbaj object-oriented, de tip static, compilator, neutru arhitectural, cu execuție pseudo-paralelă, curat din punct de vedere al fișierelor utilizate, robust, sigur, extensibil." Chiar dacă această descriere se vrea atotcuprinzătoare, nu ajută prea mult în clarificarea locului acestui limbaj. În continuare toți termenii enumerați, vor fi explicați pe larg, pentru a lămurii avantajele pe care le oferă Java:

- Simplu. Proiectanții limbajului Java au eliminat în mod deliberat multe din facilitățile ce nu au fost necesare în cadrul altor limbaje de programare de nivel înalt. Spre exemplu, Java, nu permite utilizarea pointerilor matematici, structuri implicite sau reuniuni, operatori de overloading, templates, fișiere de tip header sau succesiuni multiple.
- Limbaj orientat pe obiecte. Ca și C++, Java utilizează clase pentru a organiza codul în module logice. În timpul rulării, un program creează obiecte din clasele declarate anterior. Clasele Java pot moșteni structuri din cadrul altor clase. Nu este însă permisă moștenirea multiplă, alături de metode multiple, respectiv câmpuri din mai mult de o singură clasa.
- Tip static. Toate obiectele utilizate într-un program trebuie declarate anterior utilizării. Aceasta permit compiler-ului Java să localizeze și să raporteze conflictele.
- Compilator. Înainte de a rula un program scris în limbajul Java, programul trebuie compilat utilizând un compilator Java. După compilare rezultă un fișier "bytecode", similar fișierelor scrise în cod mașină, putând fi executat sub orice sistem de operare care dispune de un interpretor Java. Acest interpretor citește fișierul bytecode și transformă comenzile bytecode în comenzi mașina care pot fi executate direct de mașinile care rulează programul Java. Se poate spune ca Java este atât un compilator cât și un interpretor.
- Execuție multi-intrări. Programele Java pot conține mai multe intrări de execuție ,

care permit programelor să execute anumite task-uri în mod concurent. Spre exemplu, un astfel de program poate afișa o imagine pe ecran, în timp ce așteaptă sau acceptă de la tastatură caractere. Toate aplicațiile au cel puțin o intrare, reprezentată de calea principală de execuție a programului.

- **Curat.** Programele Java colectează singure datele sau fișierele adiționale utilizate, astfel încât acestea nu necesită ștergerea altor obiecte care au utilizat memoria. Acest lucru oferă o degrevare totală a programatorului în ceea ce privește problemele de management a memoriei.
- **Robust.** Deoarece interpreter-ul Java verifică toate operațiile sistem realizate de un program Java, acestea nu pot bloca sau cauza căderea unui sistem. Atunci când o eroare serioasă este descoperită, programul Java creează o excepție, avertizare. Această excepție poate fi capturată și apoi modificat programul astfel încât acesta să ruleze fără a risca ca sistemul să fie blocat.
- **Sigur.** Sistemul Java nu numai că verifică toate accesele la memorie, dar asigură și faptul că nici un virus nu este inclus într-un applet care rulează. Deoarece pointerii nu sunt utilizați, în cadrul limbajului Java, programele nu pot avea acces la alte zone ale sistemului la care acestea nu sunt autorizate.
- **Extensibile.** Programele Java suportă metodele native, care sunt funcții scrise în alte limbaje, usual în C++. Acest suport permite programatorilor să scrie funcții care vor fi executate mai rapid decât funcțiile echivalente scrise în Java. Metodele native sunt legate dinamic de programele Java. Acest lucru implică faptul că acestea sunt asociate unor programe cu rulare în timp real. Cum limbajul Java suferă permanent transformări în ceea ce privește viteza, în viitor, se preconizează că aceste metode nu vor mai fi necesare.
- **Ușor de înțeles.** Limbajul Java este bazat pe o tehnologie dezvoltată de-a lungul mai multor ani. Din acest motiv, Java poate fi rapid și ușor înțeles de oricine care are un minim de experiență cu limbajele de programare moderne cum ar fi C++.

Așa cum se desprinde din enumerarea caracteristicilor limbajului Java, acesta este un mediu relativ simplu, oferind totodată facilitățile superioare caracteristice limbajelor avansate cum ar fi C++. Datorită caracteristicii sale de a colecta toate obiectele utilizate în cadrul execuției, programatorii se pot concentra asupra dezvoltării programelor și nu vor mai pierde timp prețios în a vâna greșelile de tipul Hard-To-Find Memory-Allocation. Caracteristicile mediului Java, cum ar fi natura de program orientat pe obiect, ca și abilitatea de a asigura mai multe intrări în execuție, asigură limbajului o deosebită putere și mai ales actualitate.

## **Programe Java**

Java poate crea doua tipuri de programe: applet-uri și programe independente. Un applet Java este o parte a unei pagini Web, la fel ca o imagine sau ca o linie de text. Așa cum un browser gestionează afișarea unei imagini la care se face referire într-un document HTML, un Java-browser localizează și rulează un applet. Odată cu localizarea appletului Web browser-ul îl încarcă și execută.

Utilizând applet-uri, se pot adăuga grafice animate unei pagini Web, se pot crea jocuri complete și alte utilitare ce pot fi executate în Internet. Se pot enumera o serie de applet-uri mai cunoscute cum ar fi Bar Chart, care permite includerea într-un document HTML a unui bar chart reconfigurabil; Crossword Puzzle, permite rezolvarea unor cuvinte încrucișate în cadrul Web-ului; LED Sign, care prezintă defilarea continuă a unor mesaje computerizate pe ecranele celor care apelează paginile Web în care este respectivul applet inclus. În Figura 1.5 este prezentat un applet de tip calcul tabelar rulant sub Netscape Navigator 2.0.

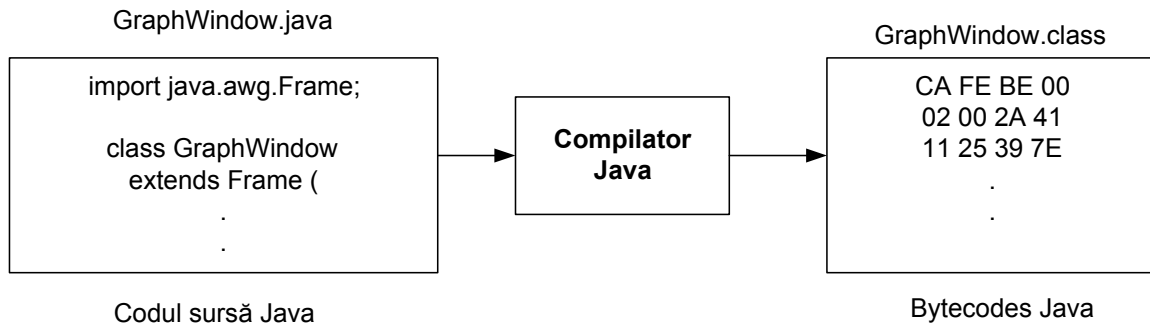
Chiar dacă mare majoritate a aplicațiilor Java sunt materializate prin applet-uri Java, pot fi realizate și programe Java independente, care nu sunt integrate în documente HTML. Cea mai cunoscută aplicație este browser-ul HotJava, prezentat în Figura 1.6. Browser-ul este complet scris în Java, demonstrând astfel că acest mediu de programare este folosit atât în aplicații de programare normale, cum ar fi bucle și evaluarea unor expresii matematice, dar și pentru programarea activităților de telecomunicații complexe.

## **CUM LUCREAZĂ JAVA**

Ca multe alte limbaje de programare, Java utilizează un compilator care convertește codul sursă de nivel înalt, scris de programator, în program executabil. În general codul mașină oferit de compilator poate fi executat sub o platformă Windows 95 sau de către procesoarele din familia Intel, sau din familia Motorola. Compilatorul Java generează un cod independent de arhitectură - *bytecodes*. Acest cod va fi rulat și executat de către o mașină virtuală - VM- implementată în marea majoritate soft.

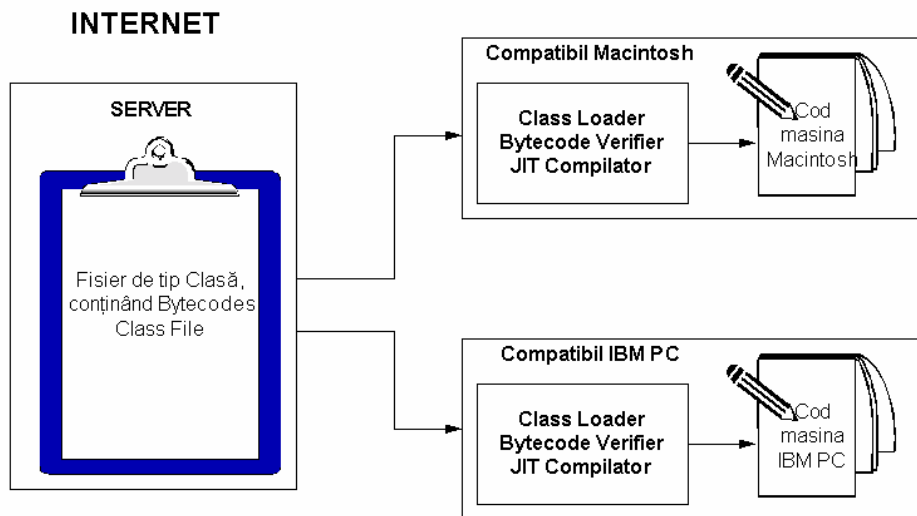
În ultima perioadă numeroase companii au anunțat că lucrează la implementarea unor procesoare hard Java, procesoare care vor îmbunătăți în mod simțitor performanțele programelor.

Fișierele bytecodes sunt denumite fișiere de tip clasă - *class files* - deoarece conțin o singură clasă Java. Pentru moment se poate considera că o clasă conține un grup de rutine și date. Majoritatea programelor Java apelează una sau mai multe clase - [Figura 1.7](#).



Pentru a executa bytecod-ul Java încărcat din rețea sau de pe disc, mașina virtuală utilizează un **class loader**. Fiecare clasă este verificată de un **bytecode verifier** care examinează dacă formatul este corect și nu accesează zone de memorie interzise. Chiar dacă pentru această verificare se consumă timp, în realitate acest lucru reduce timpul total de execuție, deoarece o dată validat programul, acesta nu va mai fi verificat la fiecare rulare. Elementul de execuție al mașinii virtuale rulează apoi bytecod-ul programului. Acest interpretor poartă numele de interpretor Just in Time. Această tehnică convertește bytecodul în cod mașină client imediat după execuție.

Modul de execuție al programelor/appletelor Java este ilustrat în [Figura 1.8](#).



## KIT-UL DE INSTALARE JAVA

Java este mai mult decât un limbaj de programare; este un mediu de programare incluzând un set complet de unelte de programare. Aceste unelte sunt: un compilator, un interpretor, un debugger, un desassembler, un profiler, și chiar mai multe alte aplicații. Pentru a rula un program Java, mai întâi este utilizat un editor de text pentru a crea un fișier cod-sursă. Acest fișier este scris, desigur, în limbajul Java. După editarea acestui fișier, salvarea acestuia se va realiza cu extensia *.java*, urmând compilarea acestuia într-un program cu format bytecode, fișier care va avea extensia *.class*. În continuare acest fișier, cu extensia



`.class`, este cel care va fi interpretat și executat. Deoarece fișierele bytecode sunt complet portabile între sistemele de operare, ele pot fi executate de către orice sistem ce dispune de un interpretor Java.

#### **NOTA**

Anumite unelte Java necesita nume lungi, în special pentru tipurile `.java` și `.class`. Atat Windows 95, 98, 2000 cât și Windows NT permit lucrul cu aceste nume, chiar și în cadrul sesiunilor DOS.

Dupa compilare și rularea unui program Java, este posibil ca și codul sursă să fie nevoie a fi modificat. Debuggerul Java permite găsirea erorilor, în timp ce Java profiler oferă scurte informații în legătură cu programul. Dacă se dorește codul sursă al unui program Java compilat Java disassembler va realiza aceasta translație. Java include programe care permit creerea unor fișiere ce includ metode native (funcții scrise în alte limbaje, cum ar fi C++). Sunt programe care creează documente HTML din fișiere cod-sursă Java. De altfel, toate uneltele de dezvoltare sunt aplicații DOS, care chiar dacă nu rulează sub Windows, oferă un mediu complet pentru creerea și managementul proiectelor Java.

## **Unde se găsește Java?**

Pentru a avea o copie personală a mediului Java Development Kit (JDK) se recomandă accesarea site-ului Sun Microsystems, la adresa <http://www.sun.com>.

Dacă doriți a avea o copie a celui mai nou browser Netscape se recomandă accesarea site-ului <http://www.netscape.com>.

Dacă doriți a avea o copie a celui mai nou browser Internet Explorer se recomandă accesarea site-ului <http://www.microsoft.com>.

## **Întrebări**

1. Enumerați 3 motive pentru care limbajul Java este considerat potrivit pentru aplicații Web.
2. Care este diferența între un applet Java și o aplicație Java independentă?
3. De ce applet-urile Java sunt considerate platforme neutre?
4. Cum lucrează un applet Java într-un document HTML?
5. Cum lucrează un program Java?
6. Ce înțelegeți prin protocol?

7. Care sunt cele mai răspândite protocoale utilizate în Internet?
8. Care este diferența dintre mediul Java și mediul C++?

## **Exerciții**

1. Explorați cât mai multe resurse disponibile programatorilor Java, prin căutarea acestora utilizând motoare de cautare.
2. Explorați pagina Web Netscape <http://www.netscape.com>.
3. Explorați pagina Web Internet Explorer
4. Utilizând un browser compatibil Java realizați o căutare Webcrawler ([www.webcrawler.com](http://www.webcrawler.com)) utilizând cuvântul "java" sau "java applet." Localizați cât mai multe applet-uri scrise, urmăriți legăturile și încărcați-le în browser-ul dumneavoastră.