



# Pledoarie pentru JAVA

Mihai Scorțaru

**Nu știu câți dintre voi își aduc aminte de un articol cu titlul "Pledoarie pentru C", scris de un membru al Lotului Național de Informatică, în urmă cu aproape zece ani. În articolul respectiv, se spunea că la concursurile de programare, majoritatea concurenților folosesc limbajul Pascal, ignorând facilitățile oferite de C. Eu nu mă voi referi la concursuri, ci la cursuri... Mai exact la limbajele de programare predate în școli. Aici avem Pascal sau C, dar poate ar fi mai bine să fie Java...**

În România la liceu se predă Pascal și C++, folosindu-se, de obicei, mediile Borland. În facultate, studentul învață și alte limbaje de care are nevoie pentru a deveni programator sau cercetător.

După ce termină facultatea, studentul român cunoaște o mulțime de limbaje, care mai de care mai ciudate și mai nefolositoare, dar știe și câteva pe care le poate folosi dacă va deveni programator.

Din păcate, această afirmație este adevărată doar din punct de vedere teoretic. Practic, studentul român nu învață nimic la facultate. Aproape toți programatorii susțin că au învățat singuri...

Să ne gândim puțin la un elev de clasa a IX-a al cărui vis este să devină programator. În liceu va învăța din greu să implementeze algoritmi care mai de care mai complicați, folosind limbajele mai sus amintite și va învăța să lucreze cu baze de date.

După ce termină liceul își dă seama că nu știe nimic din ceea ce are nevoie. Știe rezolva tot felul de probleme dar nu știe să creeze o aplicație simplă, cu o interfață prietenoasă. Așa că... merge la facultate ca să învețe acolo.

Așadar, un elev de la clasele de matematică-informatică (intensiv sau nu) este obligat să urmeze și o facultate. Liceul nu îl pregătește suficient pentru meseria pe care și-o dorește deși, în opinia mea, așa ar fi normal.

Nu mi se pare a fi hazardată afirmația potrivit căreia un absolvent de liceu poate deveni programator (cu acte în regulă) fără să fie și "om cu facultate". De ce? Pur și simplu pentru că poate! Știm că o mulțime de studenți lucrează pe post de programatori deși nu sunt "programatori cu acte în regulă". Cu siguranță, au învățat singuri... dar au învățat, deci se poate!

Și de ce să nu fie ajutați toți? Așa ar trebui, veți spune... De ce un elev obișnuit, care termină liceul, să nu poată deveni programator? Nu știu! Dar, deocamdată nu poate!

## O problemă de mentalitate

*Era demult,  
Era frumos,  
Era ca-ntr-o poveste,  
Era atunci,  
Era și-a fost,  
Era și nu mai este!*

Era demult o perioadă când la liceu în clasa a IX-a se învăța QBasic. Aceasta se întâmpla în prima jumătate a anilor '90. Apoi a venit cineva și a zis că nu se mai poate! Și așa, elevii de clasa a IX-a au început direct cu Pascalul Wow!!!. Nu mai e așa... Basic-ul e prea vechi. Pascalul e mult mai bun, C-ul e și mai bun, C++-ul e mai ceva, iar Java e... undeva pe aici...

S-a schimbat... Deci.. se poate schimba! A fost greu? Da! De ce? Fiindcă le va fi mult mai greu elevilor să învețe Pascal decât Basic!

Acesta era răspunsul oficial! Din nefericire, cel real era: Fiindcă le va fi mult mai greu profesorilor să predea Pascal decât Basic!

Și totuși... s-a schimbat... Basic-ul a fost uitat și au apărut Pascal și C/C++ (pentru clasele mai mari).

Apoi s-a pus problema ca în clasa a IX-a să se înceapă direct cu C-ul. Uitatul "război" Basic vs. Pascal s-a transformat în Pascal vs. C. De ce? Fiindcă le va fi mult mai greu elevilor să învețe C decât Pascal!

Țin minte remarcă unui profesor de informatică atunci când am susținut că limbajele de programare predate în licee trebuie schimbate mult mai des: "Profesorul de mate' iese din facultate și știe tot ce are nevoie pentru a preda matematică la liceu cincizeci de ani de acum încolo. Are salariu cât al meu. El nu mai trebuie să învețe nimic, eu de ce să trebuiască să învăț?". Poate fiindcă sunteți profesor de informatică și nu de matematică, m-am gândit eu... Dar



cuvintele nu au fost rostite din... motive de politețe. De ce apar aici? Așa... fiindcă trebuie să apară!

Și totuși, și atunci s-a făcut o jumătate de pas. Programă nu mai precizează limbajul de programare. Profesorul poate alege... Poate preda *Pascal* sau *C*. Unii susțin chiar faptul că poate preda orice: *Algol*, *Fortran*, *Java*, *C#* sau orice altceva îi trece prin minte. Nu prea îmi vine să cred... Din câte știu eu, la *Bacalaureat*, există o variantă *Pascal* și o variantă *C*. Doar atât!

Acești pași (și jumătăți de pași) au fost utili... Dar nu ajunge! Programă actuală duce la crearea unor vârfuri (care obțin premii la olimpiade) și a unei mulțimi de neștiutori care nu sunt în stare să deosebească un algoritm de determinare a permutărilor de un algoritm de determinare a elementului minim dintr-un șir. Și ce dacă nu știu? Cineva îi va învăța să deseneze interfețe în *Delphi* și vor deveni programatori.

Mi se pare inutil acest efort de a îi învăța pe elevi algoritmică... E necesară, într-adevăr, dar nu atât de necesară precum cunoașterea unui limbaj de programare modern.

Problema este, cu siguranță, una de mentalitate... Se aplică principiul: "*Rezultate de calitate care se nimeresc, dar cu eforturi minime*". Cu o astfel de mentalitate, elevul va trece prin liceu fără să cunoască nimic util. Cu o astfel de mentalitate, profesorii se înscriu la doctorat propunându-și ca temă a lucrării metodele de sortare! Credeți-mă! Am fost de față. Aceasta a fost tema lucrării pe care a propus-o un profesor care predă informatică la liceu!

Și acum... să vedem de ce îi este mai greu unui elev să învețe *Java* decât să învețe *C* sau *Pascal*. *For*-ul e tot *for*, *while*-ul e tot *while*, *if*-ul tot *if*. Deci, structurile de control care trebuie explicate sunt aceleași. Subprogramele sunt tot subprograme, deci nu avem nici aici probleme. Sintaxa e mai greoaie în *Java*! Posibil, deși practic identică celei din *C*! Numai că în *Java* avem obiecte și trebuie să fie explicate noțiuni complicate cum ar fi moștenire, polimorfism, etc. Chiar trebuie? Bineînțeles! Dar nu la început!

La început nu trebuie să se predea nimic mai complicat decât ceea ce se predă acum. Pe parcurs, se pot introduce paradigmele programării orientate obiect. Ceea ce nu se are în vedere este faptul că elevul nu știe nimic. De ce să credem că îi va fi mai ușor să înțeleagă și să utilizeze principiile programării structurate decât pe cele ale programării orientate obiect? Numai fiindcă nouă ni se pare că programarea structurată e mai ușor de învățat? Cu siguranță cei care stăpânesc ambele tehnici nu au probleme în a le utiliza. Nu li se pare că una este mai ușoară decât alta. Ambele tehnici conțin elemente de bază care trebuie înțelese.

Permiteți-mi să afirm că principiile programării orientate obiect nu sunt mai greu de înțeles decât principiile programării structurate. Și sunt mult mai utile! De ce nu sunt mai greu de înțeles? Fiindcă elevul nu are alte cunoștințe! Nu e obișnuit să lucreze pe baza unor principii, la fel ca profesorii. Orice cunoștințe noi sunt la fel de ușor de acumulat; totul depinde de talentul și seriozitatea celui

care transmite cunoștințele respective. Așadar, totul depinde de profesori. Cei care au salarii mizere și abia așteaptă să scape de învățământ. Dar dacă sunt plătiți, atunci să își facă treaba. Dacă nu să plece!

Veți spune că vor pleca toți... Poate așa și trebuie... Poate că dacă nimeni nu va mai preda, salariile vor crește... Poate... Oricum, faptul că ești prost plătit nu reprezintă o scuză pentru a nu-ți face meseria. Dacă meseria ta implică învățarea permanentă de lucruri noi, atunci trebuie să le înveți... Dacă scopul meseriei tale este să îi ajuți pe ceilalți, atunci ajută-i, oferă-le ceea ce au nevoie!

Mă opresc aici cu generalitățile... În continuare voi încerca să arăt de ce consider că limbajul *Java* este, într-adevăr, o variantă demnă de luat în seamă.

## Mediul de programare

Pentru *Pascal* se folosește acum *Borland Pascal 7.0*. Pentru *C/C++* avem *Borland C++ 3.1*. Poate sunt alte versiuni, poate în unele cazuri sunt chiar alte medii. Totuși, în majoritatea cazurilor, acestea sunt cele folosite.

Puțini își pun problema, dar aceste pachete *NU* sunt gratuite. Chiar dacă *Borland* oferă gratuit în scopuri educaționale anumite aplicații, în majoritatea școlilor românești nu există licențe.

Pentru *Java* avem o mulțime de medii de programare gratuite. Chiar avem de unde alege!

În plus, pentru mediile *Borland* avem nevoie de *Windows* (sau *DOS*). Și sistemul de operare costă. Chiar dacă acum, împreună cu ultimele dotări, în școli au ajuns și licențe *Windows*, acestea nu există peste tot!

Dacă utilizăm *Java*, putem să folosim *Linux*, ceea ce rezolvă problema pirateriei.

În plus, mediile de programare *Java* oferă o serie de facilități la care elevii de acum nici măcar nu visează. Pe lângă *code-completion* (facilitate care ar fi suficientă pentru ca să renunțăm la un mediu de programare datorită lipsei ei), avem tot felul de avantaje cum ar fi semnalarea erorilor de compilare pe măsură ce edităm codul, includerea automată a librărilor pe care le utilizăm etc.

## Avantaje

Cel mai important avantaj este faptul că *Java* este un limbaj care s-a impus și nu o simplă modă. Cu siguranță va supraviețui câțiva ani buni. Așadar, cu cât este adoptat mai repede, cu atât mai bine... Ar fi chiar caraghios să apară în programele școlare atunci când începe să nu mai fie folosit.

Un alt avantaj este faptul că există o sumedenie de tehnologii *Java*, create pe baza nucleului limbajului. Chiar dacă nu se va putea preda în liceu *RMI*, *JSP* etc., unui elev îi va fi mult mai ușor să înțeleagă aceste tehnologii dacă deține cunoștințele de bază.

*Java* este un limbaj obiectual, la fel ca marea majoritate a celorlalte limbaje folosite acum pentru crearea de produse *software*. Dacă este cunoscut un astfel de limbaj, celelalte sunt mult mai ușor de învățat atunci când e nevoie de ele!

## Dezavantaje

Nu voi enumera nici un dezavantaj deoarece nu consider nici unul a fi demn de luat în seamă. Aștept replicile tuturor celor care nu sunt de acord cu ceea ce am afirmat anterior. Așadar, dragi profesori, aici *GInfo* vă oferă dreptul la replică.

## Câteva exemple

În cadrul acestei secțiuni voi prezenta câteva scurte exemple prin care voi încerca să arăt "cât de bună e cafeaua".

### Operații cu numere mari

Aceste operații sunt un chin pentru toți cei care participă regulat la concursurile de programare. Foarte des e nevoie să implementezi așa ceva. Iar dacă mai apare și o împărțire, atunci concursul devine un coșmar.

Datorită faptului că *Java* este un limbaj obiectual și unul dintre avantajele programării orientate obiect este reutilizabilitatea codului, în *Java* nu avem această problemă. Cineva a implementat deja aceste operații și noi nu trebuie decât să le utilizăm.

Iată cum aceste operații complicate sunt implementate în *Java* folosind doar câteva linii de cod:

```
import java.math.BigInteger;

public class NumereMari {
    public static void main(String[] args) {
        BigInteger a = new BigInteger("2458");
        BigInteger b = new BigInteger("13");
        System.out.println(a + " + " + b + " = "
                           + a.add(b));
        System.out.println(a + " - " + b + " = "
                           + a.subtract(b));
        System.out.println(a + " * " + b + " = "
                           + a.multiply(b));
        System.out.println(a + " / " + b + " = "
                           + a.divide(b));
        System.out.println(a + " % " + b + " = "
                           + a.remainder(b));
    }
}
```

Cam atât e nevoie să scriem... Dar nu se oprește aici! Avem și operații cu numere reale... Iată!

```
import java.math.BigDecimal;

public class NumereRealeMari {
    public static void main(String[] args) {
        BigDecimal a = new BigDecimal("2458");
        BigDecimal b = new BigDecimal("13");
        System.out.println(a + " + " + b + " = "
                           + a.add(b));
        System.out.println(a + " - " + b + " = "
                           + a.subtract(b));
    }
}
```

```
System.out.println(a + " * " + b + " = "
                   + a.multiply(b));
System.out.println(a + " / " + b + " = "
                   + a.divide(b, 5, BigDecimal.ROUND_DOWN));
}
```

### Arhivare

Puteți spune că, totuși, după câteva ore, ați fi reușit oricum să implementați operațiile cu numere mari, chiar și pentru numere reale... Probabil că aveți dreptate dar, oricum, ar fi câteva ore pierdute.

Totuși, *Java* vă ajută să economisiți zile sau chiar luni. Cât credeți că v-ar lua să transformați un fișier într-o arhivă *ZIP*? Probabil că veți avea nevoie de câteva zile doar pentru a studia formatul fișierului. În *Java*, veți reuși să arhivați fișiere în două-trei minute. Următoarea secvență este, cred eu, sugestivă:

```
import java.util.zip.ZipOutputStream;
import java.util.zip.ZipEntry;
import java.io.FileOutputStream;
import java.io.FileInputStream;

public class ArhivareZIP {
    public static void main(String args[])
        throws Exception {
        ZipOutputStream zo = new ZipOutputStream(
            new FileOutputStream("fisier.zip"));
        ZipEntry e = new ZipEntry("fisier.txt");
        zo.putNextEntry(e);
        FileInputStream in = new
            FileInputStream("fisier.txt");
        byte b[] = new byte[16384];
        int i;
        while ((i = in.read(b)) > 0)
            zo.write(b, 0, i);
        zo.flush();
        zo.closeEntry();
        zo.close();
    }
}
```

### Încheiere

Ar trebui să trag câteva concluzii acum, dar nu o voi face... Prefer să aștept reacțiile dumneavoastră. Așadar, vă rog să vă exprimați opinia în legătură cu această "porție de cafea".

Evident, nu am vrut să încep aici o prezentare a limbajului *Java*. Secvențele de cod au fost incluse doar pentru a ilustra cât de ușoară este realizarea unor operații care necesită multe ore sau zile de muncă dacă se folosesc limbajele predate acum în licee.

Este important să vă faceți cunoscut punctul de vedere. Nu trebuie să fiți de acord cu ce este scris aici... dar nu puteți fi de acord nici cu statul pe loc în învățământul românesc de informatică. **Așteptăm scrisorile voastre!**

