



Avem nevoie de ALGORITMI?

Mulți dintre cei care doresc să se angajeze la o firmă de software consideră că este suficient să cunoască cele mai noi limbaje de programare și cele mai noi tehnologii software apărute. De obicei, se ignoră faptul că în scrierea programelor este nevoie de cunoașterea numeroșilor algoritmi care trebuie folosiți în diferite situații.

Elevii și studenții afirmă frecvent că ceea ce învață la școală sau facultate nu le va folosi în viitor. Tinerii informaticieni în devenire susțin că limbajul *Pascal*, pe care îl studiază, nu este folosit în producție și însușirea cunoștințelor necesare pentru utilizarea sa este o pierdere de vreme. Nu cred că este cazul să polemizăm cu ei în favoarea acestui limbaj, care a fost creat în scopuri didactice și, practic, nu a fost folosit în producție. Tinerii noștri prieteni observă că, pentru a crea o interfață grafică, sunt suficiente doar câteva *click-uri* dacă se folosește un mediu vizual și nici nu vor să se gândească la efortul pe care l-ar fi depus dacă ar fi dorit să creeze aceeași interfață în limbajul *Pascal*. Nu pot decât să le dau dreptate; crearea unei astfel de interfețe grafice necesită un efort considerabil dacă nu se folosesc medii vizuale.

Se pune, totuși, problema motivului pentru care programa de liceu include studierea acestui limbaj. Explicația este foarte simplă: în liceu se urmărește însușirea unor cunoștințe de algoritmică, și nu a unor cunoștințe referitoare la ultimele tehnologii *software*.

Chiar dacă mă alătur celor care cred că aceste cunoștințe de algoritmică pot fi dobândite și în condițiile în care implementările se realizează în *C/C++*, cred că algoritmiile sunt **mult mai ușor de prezentat** folosind un limbaj cu o structură cât mai simplă.

Să vedem acum ce se întâmplă atunci când elevul, studentul sau absolventul începe să lucreze la primul proiect. De obicei, creează interfața, adaugă câteva linii de cod și aplicația este gata. Nu este nevoie de nici un algoritm, ci doar de înțelegerea modului în care funcționează mediul vizual. Am întrebat un prieten care lucrează la astfel de proiecte ce se întâmplă dacă, totuși, are nevoie de un algoritm mai complicat pe care nu știe să îl implementeze, deoarece nu are cunoștințele de algoritmică necesare. Răspunsul a venit foarte repede: "*Caut pe Internet și găsesc o mulțime de implementări ale algoritmului. Le descarc și apoi le inserez în codul sursă; nu mă interesează cum au fost realizate.*"

M-am mirat puțin și apoi l-am întrebat dacă găsește chiar orice. Mi-a răspuns: "*Aproape! Am găsit tot ce am avut nevoie până acum.*"

Totuși, acel "aproape" mi se pare destul de îngrijorător. Este foarte probabil ca, la un moment dat, să nu mai găsească ce are nevoie. Probabil că nu se va întâmpla așa ceva prea curând, dar cred că o astfel de situație este inevitabilă atunci când se lucrează la proiecte de anvergură.

Mai apare o problemă; datorită lipsei cunoștințelor de algoritmică este foarte probabil ca, la un moment dat, programatorul să nu își dea seama care este algoritmul de care are nevoie. Am asistat la o discuție în care un ast-

fel de programator nu putea să își închipuie că soluția problemei sale este reprezentată de o parcurgere în lățime a unui graf. Știa că poate găsi algoritmul pe *Internet*, dar nu și-a dat seama că acesta este algoritmul de care are nevoie.

În plus, uneori acești programatori nu își dau seama că anumiți algoritmi pe care îi utilizează nu sunt eficienți. Există cazuri în care se folosesc algoritmi cu ordine de complexitate de câteva ori mai mari decât cei eficienți. De exemplu, la un moment dat, un programator se mira de ce aplicația sa nu este suficient de rapidă. Unul dintre algoritmiile folosiți era cel de determinare a poziției primei apariții a unui subșir într-un alt șir. El a scris repede un algoritm care rezolva problema, dar care rula în timp pătratic. I-am spus că există *KMP* și, după ce a descărcat de pe *Internet* implementarea acestui algoritm și a inclus-o în programul său, a observat o îmbunătățire simțitoare a vitezei.

Pentru ca un programator să își desfășoare activitatea în condiții bune, sunt necesare cunoștințe de algoritmică. De ce să depinzi de *Internet*? De ce să depinzi de fructul gândirii altora? De ce să renunți la bucuria cunoașterii algoritmilor și, mai ales, la bucuria pe care o simți după o optimizare reușită?

Mihai Scorțaru