

Introducere

Realizarea unui program pe calculator are sens doar în situația în care volumul de calcule ar depăși posibilitățile manuale. Asemenea situații se pot ivi în cazul necesității rezolvării multor probleme similare, a unor calcule complexe și laborioase, sau în cazul unor mulțimi mari de date care trebuie prelucrate. Pentru crearea unui program de calcul sunt necesare câteva cunoștințe specifice. Eficiența și performanțele programului vor depinde de cunoștințele și de experiența celor care colaborează pentru realizarea lui. În cele ce urmează ne vom referi realizarea unor aplicații simple, utilizând un limbaj de programare de nivel înalt (apropiat de limbajul vorbit).

Etapele de programare sunt în general sintetizate prin 3 faze: concepția (nivelul logic de rezolvare a problemei cu elaborarea sau alegerea algoritmului corespunzător), codificarea (transcrierea algoritmului într-un limbaj de programare, accesibil calculatorului), testarea și implementarea (verificarea corectitudinii cu date de test și punerea la punct a programului). Aceste faze pot fi realizate atât într-o formă empirică, cât și în manieră structurată, mai eficientă decât prima variantă (permițând dezvoltarea și testarea modulară). Preferând abordarea structurată, în locul celor 3 faze generice amintite prezentăm următoarea succesiune de etape, considerate necesare realizării unui program:

- recunoașterea și definirea problemei;
- alegerea și descrierea metodei de rezolvare;
- transpunerea descrierii metodei într-un limbaj de programare (crearea fișierului sursă);
- realizarea programului prin compilare (traducerea în cod mașină a fișierului sursă, generând imaginea obiect) și editarea legăturilor (completarea imaginii obiect cu elemente din biblioteca limbajului, generând fișierul executabil);
- rularea și testarea programului creat.

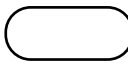

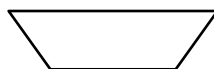
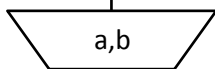

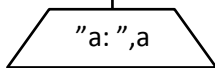
Având în vedere că aceste etape au fost discutate în semestrul anterior (la cursul de "Informatică aplicată"), ne vom orienta pe prezentarea schemelor logice structurate și pe utilizarea limbajului Fortran pentru crearea aplicațiilor consolă.

Scheme logice structurate

Schema logică este un instrument grafic prin care se pot reprezenta pașii dintr-un algoritm, sub formă de blocuri (simboluri) legate prin linii. Pentru a folosi acest instrument într-o manieră structurată, trebuie cunoscute câteva principiile fundamentale, cum ar fi:

- Schemele se alcătuiesc și se citesc de sus în jos (excepțiile se marchează cu săgeți).
- Blocurile pot avea un singur punct de intrare (exceptând cel de pornire, care are doar o ieșire), iar numărul ieșirilor depinde de tipul blocului: modulele și cele care reprezintă operații de intrare / ieșire sau de atribuire doar o ieșire, cele decizionale în funcție de tipul expresiei (cele logice 2, pentru adevărat sau fals; cele aritmetice 3, pentru negativ, nul sau pozitiv), iar blocul final nu va avea punct de ieșire.
- În alcătuirea schemelor se folosesc pe cât posibil primitive structurate (care au un singur punct de intrare și un singur punct de ieșire), care se înlănțuiesc secvențial.

Un modul poate conține orice, cu condiția ca să aibă doar o intrare și o ieșire. Conținutul unui modul trebuie detaliat separat, atunci când se impune. Simbolurile folosite la alcătuirea schemelor logice sunt prezentate în tabelul următor:

Utilizare	Simbol	Variante (exemple)
Bloc de pornire (se marchează cu START) sau de final (se notează cu STOP)		
Bloc de intrare (se marchează cu elementele care se citesc)		
Bloc de ieșire (se marchează cu elementele care se scriu)		

Alternativă la bloc de intrare / ieșire (se notează dacă este intrare sau ieșire)		
Bloc de atribuire (conține o singură expresie, a cărei valoare se atribuie variabilei din stânga)		
Blocuri pentru decizie (la ieșire se ramifică în funcție de tipul expresiei evaluate, se marchează și condițiile)		
Bloc de procedură sau modul (se marchează cu numele modulului)		
Conector intern (pentru întrerupere sau continuare în cadrul aceleiași pagini, cu marcaj corespunzător)		
Conector extern (pentru întrerupere sau continuare între pagini diferite, cu marcaj corespunzător)		

Pentru crearea unui program, aveți nevoie de un editor de text (ASCII) și de un pachet cu compilator și editor de legături. Puteți opta pentru **Code::Blocks** (<https://www.codeblocks.org/>) pentru sistemele de operare Windows, Linux și MacOS, sau pentru **JDOODLE** (<https://www.jdoodle.com/execute-fortran-online/>) dacă preferați să lucrați on-line (printr-un program de navigare). În cazul sistemelor de operare Windows, vă recomandăm pachetul **Force 2** (<https://force.lepsch.com/p/download.html>) cu compilatorul **G95 Fortran**, care se va utiliza și la orele de lucrări. Pentru a putea utiliza forma liberă în Force 2, după instalare va trebui să completați opțiunile pentru compilare cu `-ffree-form` (după cum este ilustrat) și să experimentați modalitatea convenabilă de rulare (fișier de comandă, direct, sau consolă):

