



Etapele rezolvării unei probleme cu ajutorul calculatorului

1. Analiza problemei: presupune:

- *citirea și analiza* cu atenție a enunțului problemei date pentru identificare datelor de intrare (datele inițiale ale problemei). (Ce se cunoaste ?)
- *Înțelegerea și identificarea rezultatului* așteptat (stabilirea datelor de iesire). (Ce se cere?)
- *Alegerea strategiei de rezolvare* a problemei (identifierea algoritmului optim de rezolvare a problemei) (Cum rezolv?) Pentru rezolvarea unei probleme date pot exista mai multe metode. În acest moment trebuie identificat care este cel mai optim mod de rezolvare a problemei astfel încât în cel mai scurt timp și cu cele mai puține resurse să obținem rezultatul dorit. Analiza timpului de lucru a unui program se numeste complexitate^[1], dar pentru acest subiect vom discuta mai pe larg într-un capitol separat.

2. Descrierea în limbaj natural a modului de rezolvare propus, stabilirea tipurilor de informații care trebuie prelucrate precum și a tipurilor de rezultate așteptate. Limbajul natural descrie în termeni largi etapele propuse pentru rezolvarea problemei date.



3.Descrierea algoritmului în limbaj pseudocod: Limbajul pseudocod este un limbaj intermediar între limbajul natural și limbajul de programare. În aceasta etapă se stabilesc pașii de algoritm, tipurile datelor de intrare, de manevră și de ieșire, se identifică condițiile care impun și se scriu în expresii matematice care sunt ușor de interpretat de calculator (calculatorul nu face presupuneri și nici nu ghicește ce doriți dumneavoastră ci doar execută rapid comenzile pe care le-a primit). Un algoritm scris în limbaj pseudocod nu este un program care poate fi rulat direct pe calculator, dar poate și transcris ușor într-un limbaj de programare ținând cont de vocabularul, semantica și sintaxa limbajului de programare. În această carte vom folosi un limbaj pseudocod scris în limba română (pentru a fi mai ușor de înțeles logica algoritmicii). Limbajul pseudocod se va baza implementarea ideii de rezolvare într-un format apropiat de limbajul C++, dar care nu pune accent pe sintaxa riguroasă care trebuie respectată atunci când scrii într-un limbaj de programare.

4.Scrierea codului sursă: Aceasta este etapa în care algoritmul scris în limbaj pseudocod este transcris în limbaj de programare, ținând cont de regulile impuse de acesta și respectând semnificația construcțiilor sintactice corecte impuse de limbajul de programare.



5. Corectarea erorilor sintactice: Corectarea unui cod sursă din punct de vedere sintactic se numeste **compilare**[2]. În aceasta etapă programul este verificat, iar eventualele greșeli sau omisiuni ale respectării regulilor sintactice sau semantice sunt semnalate de compilator print-o listă de erori. Cel care a scris programul are obligația să corecteze aceste erori folosindu-se de mesajele de eroare afișate precum și de cunoștințele sale. Mesajul de eroare afișat, care semnalează o eroare de compilare, nu indică întotdeauna corect modalitatea de corectare (el este mai mult informativ). Erorile de eroare vor fi căutate începând de la poziția semnalat de compilator sau de la poziția semnalată spre începutul programului. O serie de greșeli frecvent întâlnite la începători sunt generate de definirea eronată a tipurilor de date care vor conține rezultatele. Dacă rezultatul așteptat depășește ca valoare domeniul de definiție stabilit atunci aceasta greșeala nu va fi identificată de către compilator, deci nu va fi corectată în această etapă. La execuția programului vom avea însă surpriza unor rezultate neașteptate, chiar dacă soluția de rezolvare găsită este corectă și programul scris nu are greșeli de compilare.

6. Testarea programului: În această etapă se verifică corectitudinea rezolvării problemei prin analiza rezultatului obținut pentru diferite valori ale datelor de intrare. Tot aici se identifică și eventualele *erori de gândire*, care apar prin alegerea unui algoritm de rezolvare greșit. În această etapă trebuie identificate seturi de date de intrare numite și *cazuri limită* pentru care programul s-ar putea bloca sau ar genera rezultate greșite datorită unei neglijențe în etapa de analiză și implementare a programului.