

Cuprins

1. SUBPROGRAME	6
2. RECURSIVITATEA.....	37
3. METODA BACKTRACKING	50
4. METODA GREEDY	107
5. METODA DIVIDE ET IMPERA	120
6. METODA PROGRAMĂRII DINAMICE	128
BIBLIOGRAFIE	143

1. SUBPROGRAME

Cum se rezolvă problemele dintr-o firmă într-o lume tehnologică?

Cum poate rezolva directorul (managerul) toate problemele?

Singur ...

... sau poate trasa diferite sarcini subalternilor.

Aceștia le pot rezolva singuri ... sau pot apela la subalternii lor ... pentru a rezolva probleme mai mici.

În concluzie, atunci când un director are un program încărcat este recomandabil să delege sarcini subordonaților și să-i cheme la raport pentru a vedea stadiul îndeplinirii sarcinilor.

Când un program devine foarte complex, este recomandat ca acesta să fie divizat în mai multe părți. În acest mod programul devine mult mai ușor de gestionat.

Pentru a putea diviza un program, se vor folosi subprogramele.

Definiție: Un subprogram este o colecție de instrucțiuni care execută o anumită sarcină.

În C/C++ subprogramele se mai numesc și **funcții**.

Există două tipuri de subprograme/funcții:

- **modulul/subprogramul apelant** - Modulul/subprogramul care apelează la alte subprograme pentru rezolvarea unei probleme.

- **modulul/subprogramul apelat** - Modulul/subprogramul care este apelat de un alt modul, pentru a-i rezolva acestuia o problemă.

Subprogramul apelant poate fi un subprogram sau subprogramul principal. Subprogramul principal sau subprogramul rădăcină este definit de funcția *main()*.

Subprogramul apelat poate fi doar un subprogram.

Când în programul principalese întâlnește un apel, execuția continuă cu prima instrucțiune din programul apelat. Când se termină executarea instrucțiunilor din subprogram, se revine la instrucțiunea imediat următoare apelului din programul principal. Identic se procedează și când subprogramul apelant este un subprogram.

Exemplu din viața reală:

Gândiți-vă la o firmă.

O firmă este formată dintr-un șef care are în subordine mai mulți angajați.

Șeful întotdeauna îi va trage la răspundere pe angajații acestuia. Angajații niciodată nu vor putea să îl tragă la răspundere pe șef. În acest caz, îl vom putea numi pe șef “subprogramul principal” care poate doar să apeleze.

Șeful poate să angajeze și un subșef. Acest subșef poate la rândul lui să fie apelat de către șef, dar și să apeleze angajații pe care acesta îi are în subordine. În acest caz îl putem numi pe subșef “subprogram” deoarece poate atât să apeleze cât și să fie apelat.

Avantajele utilizării subprogramelor:

- Modularizarea problemei (descompunerea în subprobleme)
- Reutilizarea codului - subprogramul odată scris poate fi utilizat și în alte programe

- Reducerea numărului de erori care apar în scrierea programelor
- Depistarea și corectarea cu ușurință a erorilor
- Realizarea unor programe ușor de urmărit

Variabile locale și variabile globale

Variabile globale

Variabilele globale sunt declarate în afara oricărei funcții și sunt vizibile (pot fi utilizate) în tot programul (în programul principal și în subprograme) din momentul declarării lor.

Variabile locale

La fel cum se declară variabilele în cadrul funcției *main*, la fel se pot declara variabile în cadrul celorlalte funcții. Aceste variabile se numesc locale și sunt accesibile doar de funcția care le-a declarat.

Parametri formali și parametri actuali

Parametrii se utilizează pentru a permite transferuri de date la o funcție în momentul apelului ei.

Acest mecanism de transfer al datelor prin intermediul parametrilor ne permite construirea de funcții făcând abstracție de valorile concrete care vor fi prezentate abia la execuția programului.

- **Parametri formali** apar în antetul subprogramului și sunt utilizați de subprogram pentru descrierea abstractă a unui proces de calcul .
- **Parametri actuali** apar în instrucțiunea de apelare a unui subprogram și sunt folosiți la execuția unui proces de calcul pentru valori concrete.