

## Observatii

1. Josza: "A ramas nu cu limbajele invatate, ci cu principiile generale insusite"
2. Metodologia programarii e mult mai dificil de asimilat decat sintaxa unui limbaj.
3. Program = Algoritm in L.P.

## SCOP

1. Deprinderi de programare perfecta la nivel de subalgoritmi si programe simple
2. Deprinderi de proiectare a algoritmilor prin descompunere topdown
3. Testarea completa a programelor scrise
4. Deprinderi de documentare a programelor (si de redactare în scris - rol important revine laboratorului)
5. Elaborarea corecta a programelor
6. Disciplina în programare: - variabilele au semnificatii; - claritatea e necesara... - proiectarea trebuie sa preceada codificarea ! - documentarea e importanta. Rolul comentariilor - rolul l.p.f. - programarea e o activitate intelectuala dificila. A gandi acasa.

## CONTINUT

C1: Limbaje de descriere a algoritmilor (Scheme logice, Pseudocod, Pascal)

C2: Subalgoritmi (Pseudocod)

C3: Programe Pascal simple

C4: Faze in viata unui program (specificare, proiectare, codificare, testare, documentare, intretinere). Consecinte. Testarea programelor

C5: Corectitudinea algoritmilor (Floyd). Dezvoltarea corecta a algoritmilor din specificatii

C6: Metode generale de elaborare a algoritmilor: topdown, rafinare în pasi succesivi, programare modulara, programare structurata. Reguli importante în programare. Stil

C7+8: Tipuri Abstracte de date

C9: Complexitatea algoritmilor

C10: Recursivitate. Tehnici de programare: Backtracking.

C11+12: Metoda greedy + Programare dinamica+Divide&Impera+Branch&Bound+ Metode euristice

C13: Algoritmi de cautare si complexitatea lor

C14: Algoritmi de sortare si complexitatea lor