

# Cap 8. Programe Fortran secvențe de instrucțiuni simple

- *Structura programelor Fortran*
- *Forma codului sursă*
- *Înregistrări formatate*
- *Input-Output cu formatare condusă de listă*
- *Compilarea, editarea legăturilor și execuția*
- *Output formatat*

# *Structura programelor Fortran*

Orice *program Fortran* este alcătuit din una sau mai multe *unități de program*. Unitate de program poate fi: *programul principal* (main), un *subprogram extern* (subrutină sau funcție), un *modul* sau un *bloc de date* (block data). Dacă programul este format dintr-o singură unitate de program, acea unitate, în mod obligatoriu, este programul principal.

*Subprogramul extern* se poate utiliza pentru a efectua prelucrări asupra unor entități care sunt disponibile subprogramului extern. Pentru a apela o subrutină externă se folosește instrucțiunea CALL. Atunci când este nevoie de valoarea unei funcții externe se va apela un subprogram funcție.

*Modulul* conține definiții de date, definiții de tip, definiții de procedură, numite *subprograme modul* ce pot fi făcute accesibile altor unități de program.

# *Structura programelor Fortran*

Subprogramele modul pot fi subprograme subrutină sau funcție. Un subprogram modul poate fi apelat de orice alt subprogram modul sau de orice altă unitate de program ce are acces la modul.

Programul principal, subprogramele externe și subprogramele modul pot conține *subprograme interne* care pot fi subrutine sau funcții. Subprogramele interne se introduc cu instrucțiunea **CONTAINS**.

Ex:

Program principal

...

CONTAINS

Subprogram

END

# Forma codului sursă

*Fișierul sursă în Fortran* conține textul unui program ce se creează de obicei cu un editor de texte. Textul sursă este divizat în unități fizice numite *linii* (lines) sau înregistrări ale fișierului sursă și conține instrucțiuni, comentarii și linii INCLUDE.

Există două forme de scriere a codului sursă, *forma fixă* acceptată de standardele Fortran 77 și Fortran 90 și *forma liberă* acceptată de standardul Fortran 90.

În *forma fixă* înregistrările fișierului sursă au o lungime de 80 de coloane. În coloanele 1-5 se pot scrie doar etichetele instrucțiunilor urmând ca în coloanele 7-72 să fie scrise instrucțiunile.

Dacă poziția 6 conține orice caracter Fortran cu excepția caracterului blank sau 0, atunci acea linie este o *continuare* a liniei precedente.

# Forma codului sursă

De obicei, pe o linie se regăsește o singură instrucțiune însă aceasta se poate continua pe mai multe linii. Limita maximă de linii pe care se poate întinde o instrucțiune este de 20 de linii.

Coloanele 73-80 sunt ignorate de către compilator iar liniile mai scurte de 72 de caractere se completează cu blaturi.

Într-o instrucțiune blaturile nu sunt semnificative cu excepția prezenței lor într-o constantă caracter.

## **Exemplu:**

Read\*, a, b, c este echivalent cu Read\*,a,b,c

Caracterul c sau \* în coloana 1 indică faptul că întreaga linie este un *comentariu*. Semnul exclamării dacă nu face parte dintr-o constantă caracter, marchează începutul unui comentariu. Comentariul se termină cu sfârșitul liniei.

# Forma codului sursă

## Exemplu:

```
C      Aceasta este o linie comentariu  
*      și aceasta este o linie comentariu  
      REAL alfa, beta, gama, delta, epsilon ! Începe un  
comentariu  
!      si aceasta este o linie comentariu
```

Comentariile sunt ignorate de către compilator. Ele se folosesc de către programator pentru a face programul mai inteligibil. De exemplu comentariile se folosesc pentru a arăta scopul fiecărei secțiuni din program.

În timpul procesării, într-un fișier sursă poate fi importat și inclus un text sursă din alt fișier. Pentru aceasta se scrie o linie ce are sintaxa:

```
INCLUDE const_lit_char
```

# Forma codului sursă

unde `const_lit_char` este o constantă literară caracter care reprezintă de obicei numele fișierului ce conține textul ce trebuie inclus. Linia `INCLUDE` se scrie în coloanele 7-72.

*Exemplul:*

O unitate de program ce conține o linie `INCLUDE`:

```
PROGRAM ANALIZA  
IMPLICIT NONE  
REAL A, B, C  
  
...  
INCLUDE 'ECUATIE.FOR'  
  
...  
END
```

# *Forma codului sursă*

În *forma liberă*, instrucțiunile se scriu pe linii ce pot conține de la 0 la 132 caractere. În forma liberă dispăre interdicția formei fixe de a scrie etichetele și instrucțiunile în anumite coloane.

O instrucțiune poate continua pe linia următoare dacă ultimul caracter al liniei, ce trebuie să fie continuată, este &. O instrucțiune poate continua pe cel mult 39 linii. În absența semnului de continuare &, sfârșitul liniei marchează sfârșitul instrucțiunii.

Apariția semnului exclamării în afara unei constante caracter marchează începutul unui comentariu.

Pe o linie se pot scrie două sau mai multe instrucțiuni cu condiția ca respectivele instrucțiuni să fie separate prin punct și virgulă.

Blancurile se pot folosi pentru a îmbunătăți aspectul textului.



# Înregistrări formatare

În computer informațiile sunt stocate în diferite tipuri de fișiere. Fișierul este format dintr-o succesiune de înregistrări iar valorile dintr-o înregistrare se pot reprezenta în două moduri, *formatare* sau *neformatate*.

Dacă înregistrarea este scrisă cu caractere pe care le putem citi atunci se spune că înregistrarea este *formatată*.

*Înregistrările neformatate* constau din valori așa cum sunt ele stocate în memoria computerului, adică în formă binară.

De exemplu, la execuția unei instrucțiuni de citire, atunci când o valoare este citită dintr-o înregistrare formatată, ea trebuie convertită de la forma externă, o înșiruire de caractere, la forma de reprezentare internă, adică cea binară.

# *Input-Output cu formatare condusă de listă*

Instrucțiunea de citire (input) cu formatare condusă de listă are sintaxa

```
READ* [ , var [, var]...]
```

unde *var* este o variabilă. Asteriscul înseamnă formatare implicită, adică este lăsat la alegerea celui ce pregătește fișierul de intrare.

Instrucțiunea de scriere (output) cu formatare condusă de listă are sintaxa

```
PRINT* [ , ent [, ent ]...]
```

unde *ent* este o entitate, poate fi o variabilă sau o expresie. Asteriscul în instrucțiunea PRINT înseamnă formatare implicită adică formatul datelor de ieșire este lăsat pe seama computerului.

# *Input-Output cu formatare condusă de listă*

Separatorii de valori într-o înregistrare input sunt:

- o virgulă – este opțional precedată de blancuri unul după altul
- un slash – este opțional precedat blancuri unul după altul
- un blanc - ce se află între două caractere non blanc.

# Compilarea, editarea legăturilor și execuția

Înainte de a fi compilat, un program Fortran trebuie mai întâi convertit la o formă binară, proces ce are loc în două faze:

1) *compilarea*: compilatorul FTN90 traduce textul Fortran în cod binar care este conținut într-un fișier binar relocabil.

2) *încărcarea*. Folosind editorul de legături (linker) fișierul binar este încărcat în memorie împreună cu alte fișiere ce conțin cod binar relocabil sau fișiere de bibliotecă care au fost produse prin compilări anterioare.

În mod normal după compilare se vor produce:

- un fișier *binar relocabil* care are același nume cu fișierul sursă Fortran și care are extensia "OBJ".
- un fișier *executabil* care are același nume cu fișierul sursă Fortran și care are extensia „EXE”.

# Output formatat

Formatarea implicită este foarte comodă și simplă, însă uneori este rigidă. De exemplu, dacă la ieșire dorim să formăm un tabel cu date reale, nu întotdeauna reușim să aliniem datele pe coloane.

În Fortran programatorul poate introduce stilul său personal în forma outputului prin instrucțiuni de scriere ce utilizează informația dintr-o *specificație de format*.

Sintaxa unei instrucțiuni de afișare cu output formatat este:

```
PRINT fmt [ , entitate [ , entitate]... ]
```

unde *fmt* este o constantă caracter a cărei valoare este specificația de format. *Specificația de format* este o listă de *descriptori de editare* separați prin virgule și închiși în paranteze.

# Output formatat

Pentru a putea formata output-ul se poate utiliza următoarea instrucțiune:

```
PRINT "(A, I3, 2X, A, F6.2, 2X, A, E15.6)", 'n =', n, 'b = ', b, 'c = ', c
```

Dacă  $n = 100$ ,  $b = 12.3$  și  $c = 15 \cdot 10^8$  ea va produce outputul

```
n =100 b = -12.3 c = 0.150000E10
```

Exemplul arată 5 dintre cei mai des folosiți descriptori: A pentru date alfanumerice, I pentru date întregi, F și E pentru date reale, X pentru spațiere.

# Output formatat

Descriptorul A rezervă spațiu pentru outputul datelor caracter. Lungimea reală a constantei caracter de tipărit sau lungimea declarată a variabilei determină câte coloane sunt folosite pentru ea la ieșire. *Forma generală* a descriptorului A este

$$A [w]$$

unde  $w$  este lungimea câmpului măsurată în caractere.

Dacă  $w > \text{len}$  (numărul de caractere ce vor fi printate) atunci vor fi introduse blancuri pentru a compensa diferența (aliniată la dreapta).

Dacă  $w < \text{len}$  atunci vor fi afișate un număr de caractere, de la stânga la dreapta, ce va fi egal cu  $w$ .

Dacă  $w$  este omis se folosește lungimea existentă a valorii la ieșire.

# *Output formatat*

Exemple

**PRINT \*, "Simple text" ⇒ Simple text**

**PRINT "(A)", "Simple text" ⇒ imple text**

**CHARACTER(LEN=5) :: FMT = "(A15)"**

**PRINT FMT, "Simple text" ⇒ Simple text**

**FMT = "(A10)"**

**PRINT FMT, "Simple text" ⇒ imple tex**

**CHARACTER(LEN=20) :: VAR = "Simple text"**

**PRINT '(A)', VAR ⇒ imple text + 9 spații**

**PRINT '(A)', VAR(8:11) ⇒ text**



# Output formatat

*Descriptorul I* se folosește la formatarea întregilor. Forma generală a acestui descriptor este

***Iw***

unde *w* este lungimea câmpului. Trebuie incluse toate cifrele inclusiv semnul, dacă este negativ.

ex.                    *I4*    -123  
Print "(I4)", 12345    - \*\*\*\*

# *Output formatat*

*Descriptorul F6.2* înseamnă că un total de 6 coloane sunt rezervate pentru a tipări valoarea variabilei reale *b* în forma pozițională, valoarea fiind rotunjită la două zecimale. Punctul zecimal ocupă o coloană, iar dacă semnul este negativ, semnul mai ocupă o coloană, prin urmare, în formatul *F6.2*, numărul cel mai mare tipăribil este 999.99, iar -99.99, cel mai mic.

Forma generală a acestui descriptor este

$Fw.d$

unde *w* este lungimea în caractere a câmpului, iar *d* este numărul de zecimale care se rotunjesc la tipărire.

# Output formatat

*Descriptorul* E15.6 înseamnă că un total de 15 coloane sunt rezervate pentru a tipări valoarea variabilei reale  $c$ . Forma generală a acestui descriptor este

$$Ew.d$$

unde  $w$  este lungimea câmpului dată în coloane și  $d$  este numărul de zecimale.

*Descriptorul* 2X este un editor de editare a poziției ce lasă două spații. Forma generală este

$$nX$$

unde  $n$  înseamnă număr de caractere blanc.

# Bibliografie

- *Octavian PETRUȘ, Fortran 90/95, Limbaj și Tehnici de programare, Editura Universității Tehnice “Gheorghe Asachi” din Iași, 2001*
- Romeo CHELARIU, Sisteme de operare și limbaje de programare (Îndrumar de laborator), <http://www.sim.tuiasi.ro/wp-content/uploads/Chelariu-indrumar-solp.pdf>, 2004
- <https://ro.wikipedia.org>