

FORTRAN

Ferran Conde Codorniu

A large, dark blue, diagonal shape that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

FORmula TRANslator

Introducció

- Desenvolupat per IBM els anys 1950
- Orientat a anàlisi numèrica i computació científica
- Primer llenguatge d'alt nivell usat àmpliament

Principals aplicacions

Predicció meteorològica

Mecànica de fluids computacional

Cristal·lografia

Anàlisi matemàtica general

Versions

Paradigmes

- FORTRAN 77:
programació estructurada
- Fortran 90:
programació modular i genèrica
- Fortran 2003:
programació orientada a objectes
- Fortran 2008:
programació concurrent

Característiques

- Compilat
- Multi-paradigma
(imperatiu)
- Alt nivell

Sistema de tipus

Fortament tipat
Tipatge estàtic
Tipatge explícit

- Enter
- Real (precisió simple)
- Double
- Caràcter
- Complex
- Lògic

Forma fixa

L'era pre-disquets

S'escrivia FORTRAN77
en targetes perforades

- 1 a 5: Label field
(definir etiquetes per IF, ...)
- 6: Continuation field
(és la continuació d'una
targeta anterior?)
- 7 a 72: Statement field
- 73 a 80: Ignorades
(IBM 704 només 72)

Fortran 90

Introdueix diverses millores

- Forma lliure
- Identificadors:
de 6 a 31 caràcters
- Recursivitat
- Reserva dinàmica de memòria
- Sobrecàrrega d'operadors

Estructura d'un programa

```
program nom
```

```
(implicit none)
```

```
...
```

```
end program nom
```

implicit none

Fortran assumia:

variables **[i-n]\w+** són enters; reals altrament.

Anul·lar-ho evita confusions de tipus.

```

*      euclid.f (FORTRAN 77)
*      Find greatest common divisor using the Euclidean algorithm

PROGRAM EUCLID
  PRINT *, 'A?'
  READ *, NA
  IF (NA.LE.0) THEN
    PRINT *, 'A must be a positive integer.'
    STOP
  END IF
  PRINT *, 'B?'
  READ *, NB
  IF (NB.LE.0) THEN
    PRINT *, 'B must be a positive integer.'
    STOP
  END IF
  PRINT *, 'The GCD of', NA, ' and', NB, ' is', NGCD(NA, NB), '.'
  STOP
END

FUNCTION NGCD(NA, NB)
  IA = NA
  IB = NB
1  IF (IB.NE.0) THEN
    ITEMP = IA
    IA = IB
    IB = MOD(ITEMP, IB)
    GOTO 1
  END IF
  NGCD = IA
  RETURN
END

```

```
! sum.f90
! Performs summations using in a loop using EXIT statement
! Saves input information and the summation in a data file

program summation
implicit none
integer :: sum, a

print*, "This program performs summations. Enter 0 to stop."
open(unit=10, file="SumData.DAT")

sum = 0

do
  print*, "Add:"
  read*, a
  if (a == 0) then
    exit
  else
    sum = sum + a
  end if
  write(10,*) a
end do

print*, "Summation =", sum
write(10,*) "Summation =", sum
close(10)

end
```