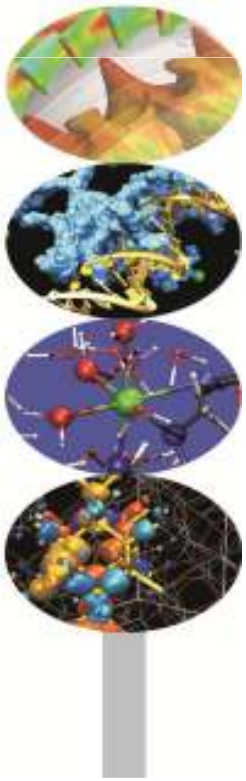
CINECA SCAI
SuperComputing Applications and Innovation

Overloading di operatori



Introduction to Fortran 90

Nicola Spallanzani, *CINECA*

Paolo Ramieri, *CINECA*

Ottobre 2014

CINECA

Overloading di operatori

La possibilità di

- estendere gli operatori predefiniti

- definire nuovi operatori

rappresenta uno dei più potenti strumenti sintattici introdotti dal Fortran 90 .

Overloading = la possibilità di estendere un operatore

Assegnazione

I 3 passi per

estendere l'operatore di assegnazione:

1. generare un **modulo** che contiene le strutture dati da manipolare

Assegnazione

2. definire una SUBROUTINE con due soli argomenti: il primo di intent OUT o INOUT, il secondo di intent IN

```
SUBROUTINE copia(n,d)
  IMPLICIT NONE
  TYPE(nascita), INTENT(INOUT) :: n
  TYPE(data), INTENT(IN) :: d
```

Assegnazione

3. definire un costrutto INTERFACE appropriato, che faccia riferimento a questa subroutine

```
INTERFACE ASSIGNMENT (=)
    MODULE PROCEDURE copia
END INTERFACE
```

Assegnazione

```
INTERFACE ASSIGNMENT (=)
   MODULE PROCEDURE copia
END INTERFACE


SUBROUTINE copia(n,d)
   IMPLICIT NONE
   TYPE(nascita), INTENT(INOUT) :: n
   TYPE(data), INTENT(IN) :: d

   n%giorno = d

   RETURN
END SUBROUTINE copia
```

Altri operatori

I 3 passi per

estendere l'operatore di somma:

1. generare un **modulo** che contiene le strutture dati da manipolare

Altri operatori

2. definire una FUNCTION con due soli argomenti di intent IN

```
FUNCTION concatena(a, b)
 IMPLICIT NONE
 CHARACTER(LEN=*), INTENT(IN) :: a, b
```

Altri operatori

3. definire un costrutto **INTERFACE** appropriato, che faccia riferimento a questa function

```
INTERFACE OPERATOR (+)
    MODULE PROCEDURE concatena
END INTERFACE
```

Altri operatori

Non è possibile ridefinire gli operatori per i tipi per cui sono già definiti.

Esercizi

1. Dato i tipi *nascita* e *data*, come dagli esempi precedenti, si definisca il tipo "*generalita*" costituito da Nome, Cognome e Evento, dove Evento è di tipo *nascita*. Si realizzi l'overloading dell'assegnazione tra la componente Evento di *generalita* e un tipo *nascita*
2. Si realizzi l'overloading dell'operazione somma per concatenare 2 stringhe con il simbolo + (ricordiamo che la concatenazione di stringhe è realizzata dal simbolo //). Si utilizzi nell'esercizio precedente.

Proposte di soluzioni

Esercizio 1:

```
MODULE anagrafe
    IMPLICIT NONE
    TYPE data
      INTEGER :: g, m, a
    END TYPE data
    TYPE nascita
      CHARACTER(132) :: luogo
      TYPE(data) :: giorno
    END TYPE nascita
    TYPE generalita
      CHARACTER(132) :: nome, cognome
      TYPE(nascita) :: evento
    END TYPE generalita
```

Proposte di soluzioni

```
INTERFACE ASSIGNMENT (=)
    MODULE PROCEDURE copia_anagrafe
END INTERFACE
CONTAINS
    SUBROUTINE copia_anagrafe(g,n)
      IMPLICIT NONE
      TYPE(nascita), INTENT(IN) :: n
      TYPE(generalita), INTENT(INOUT) :: g
      g%evento = n
      RETURN
    END SUBROUTINE copia_anagrafe
END MODULE anagrafe
```

Proposte di soluzioni

```
PROGRAM stampa_anagrafe
 USE anagrafe
 IMPLICIT NONE
 TYPE(nascita) :: n
 TYPE(generalita) :: dati

 dati%nome = "John"
 dati%cognome = "Smith"
 n=nascita("Citta' del Capo, Sudafrica",data(29,2,2004))

 PRINT *,"Mi chiamo ",dati PRINT *,"Sono nato a ",n
 WRITE (*,'(3a,2(i2,"/"),i4)')
    & "Sono nato a ",TRIM(n%luogo),", il giorno ",
    & n%giorno%g, n%giorno%m, n%giorno%a
    STOP
END PROGRAM stampa_anagrafe
```

Proposte di soluzioni

Esercizio 2:

```
MODULE parole
    IMPLICIT NONE

    INTERFACE OPERATOR (+)
      MODULE PROCEDURE concatena
    END INTERFACE
```

Proposte di soluzioni

```
CONTAINS

   FUNCTION concatena(a,b)
     IMPLICIT NONE
     CHARACTER (LEN=*), INTENT(IN) :: a, b
     CHARACTER (LEN=(LEN_TRIM(a) + LEN_TRIM(b) + 1)) :: concatena

     concatena = TRIM(a)//" "//TRIM(b)
     RETURN
   END FUNCTION concatena

END MODULE parole
```

Proposte di soluzioni

```
MODULE anagrafe
    IMPLICIT NONE
    TYPE data
      INTEGER :: g, m, a
    END TYPE data
    TYPE nascita
      CHARACTER(132) :: luogo
      TYPE(data) :: giorno
    END TYPE nascita
    TYPE generalita
      CHARACTER(132) :: nome, cognome
      TYPE(nascita) :: evento
    END TYPE generalita
```

Proposte di soluzioni

```
INTERFACE ASSIGNMENT (=)
    MODULE PROCEDURE copia_anagrafe
END INTERFACE
CONTAINS
    SUBROUTINE copia_anagrafe(g,n)
      IMPLICIT NONE
      TYPE(nascita), INTENT(IN) :: n
      TYPE(generalita), INTENT(INOUT) :: g
      g%evento = n
      RETURN
    END SUBROUTINE copia_anagrafe
END MODULE anagrafe
```

Proposte di soluzioni

```
PROGRAM stampa_anagrafe
    USE parole
    USE anagrafe
    IMPLICIT NONE
    CHARACTER (LEN=80) :: frase
    TYPE(nascita) :: n
    TYPE(generalita) :: dati

    dati%nome = "John"
    dati%cognome = "Smith"
    n=nascita("Citta'          del          Capo,
    Sudafrica",data(29,2,2004))
```

Proposte di soluzioni

```
frase="Il mio nome e' " + dati%nome + dati%cognome

PRINT*, frase

frase="Sono nato a " + TRIM(n%luogo) + " il giorno"

PRINT*, frase, n%giorno%g, n%giorno%m, n%giorno%a

STOP
END PROGRAM stampa_anagrafe
```