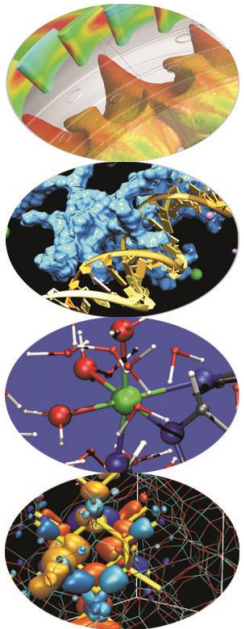


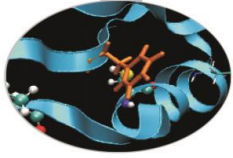
# Overloading di operatori

## *Introduction to modern Fortran*

Paride Dagna, *CINECA*

*Maggio 2016*





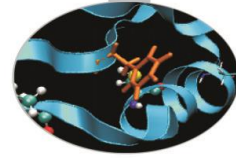
# Overloading di operatori

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



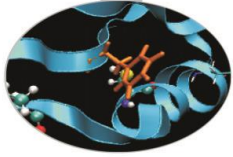
# Assegnazione

I 3 passi per **estendere l'operatore di assegnazione**:

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

```
MODULE Anagrafe
  IMPLICIT NONE

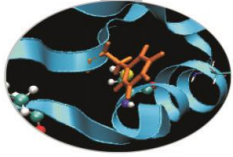
  TYPE data
    INTEGER :: g, m, a
  END TYPE data
  TYPE nascita
    CHARACTER(132) :: luogo
    TYPE(data) :: giorno
  END TYPE nascita
  TYPE(nascita) :: n
```



# Assegnazione

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

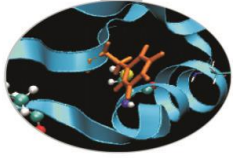
```
SUBROUTINE copia(n,d)
  IMPLICIT NONE
  TYPE(nascita), INTENT(INOUT) :: n
  TYPE(data), INTENT(IN) :: d
```



# Assegnazione

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

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



# Assegnazione

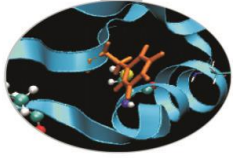
```
INTERFACE ASSIGNMENT (=)
  MODULE PROCEDURE copia
END INTERFACE
```

CONTAINS

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

  n%giorno = d

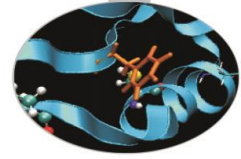
  RETURN
END SUBROUTINE copia
```



# Altri operatori

I 3 passi per **estendere l'operatore di somma**:

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



# Altri operatori

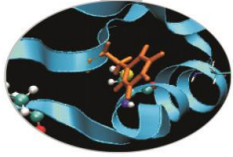
2. definire una FUNCTION con due soli argomenti di intent IN

```
FUNCTION concatena(a, b)
```

```
  IMPLICIT NONE
```

```
  CHARACTER (LEN=*) , INTENT (IN) :: a, b
```

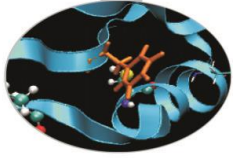




# Altri operatori

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

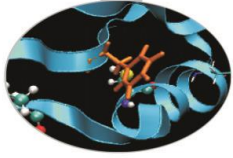
```
INTERFACE OPERATOR (+)  
    MODULE PROCEDURE concatena  
END INTERFACE
```



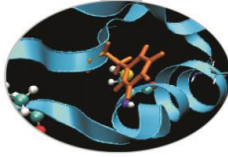
# Altri operatori

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

# Esempio



Nell'esempio *StampaMatrici.f90* vengono definite funzioni specifiche per la stampa di matrici di tipo diverso, associate all'operatore di assegnazione.



# Esercizi

1. Si realizzi l'overloading dell'operazione somma per concatenare 2 stringhe con il simbolo + (ricordiamo che la concatenazione di stringhe è realizzata dal simbolo //).
2. 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*
3. Riprendere l'esercizio del capitolo precedente sostituendo l'operatore .distanza. Con l'overloading dell'operatore -