

Linguaggi di Programmazione I – Lezione 4

Prof. Marco Faella

<mailto://m.faella@unina.it>

<http://wpage.unina.it/mfaella>

Materiale didattico elaborato con i Proff. Sette e Bonatti

6 maggio 2025



Panoramica della lezione

Parametrizzazione di procedure

Modalità di passaggio parametri

Bibliografia



Parametrizzazione di procedure

Parametri

Tipi dei p.

Associazione dei p.

Esempio

Associazione di default

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Modalità di passaggio
parametri

Bibliografia

Parametrizzazione di procedure



Parametri

[Parametrizzazione di procedure](#)

[Parametri](#)

[Tipi dei p.](#)

[Associazione dei p.](#)

[Esempio](#)

[Associazione di default](#)

[Ordine di valutazione dei param. attuali](#)

[Ordine di valutazione dei param. attuali](#)

[Ordine di valutazione dei param. attuali](#)

[Modalità di passaggio parametri](#)

[Bibliografia](#)

- Costituiscono il mezzo attraverso il quale le informazioni transitano esplicitamente tra l'unità chiamante e quella chiamata.



Parametri

[Parametrizzazione di procedure](#)

[Parametri](#)

[Tipi dei p.](#)

[Associazione dei p.](#)

[Esempio](#)

[Associazione di default](#)

[Ordine di valutazione dei param. attuali](#)

[Ordine di valutazione dei param. attuali](#)

[Ordine di valutazione dei param. attuali](#)

[Modalità di passaggio parametri](#)

[Bibliografia](#)

- Costituiscono il mezzo attraverso il quale le informazioni transitano esplicitamente tra l'unità chiamante e quella chiamata.
- Alcuni linguaggi distinguono tre categorie:
 - ◆ parametri di ingresso (IN): sono passati dalla unità chiamante alla unità chiamata al momento dell'invocazione;



Parametri

[Parametrizzazione di procedure](#)

[Parametri](#)

[Tipi dei p.](#)

[Associazione dei p.](#)

[Esempio](#)

[Associazione di default](#)

[Ordine di valutazione dei param. attuali](#)

[Ordine di valutazione dei param. attuali](#)

[Ordine di valutazione dei param. attuali](#)

[Modalità di passaggio parametri](#)

[Bibliografia](#)

- Costituiscono il mezzo attraverso il quale le informazioni transitano esplicitamente tra l'unità chiamante e quella chiamata.
- Alcuni linguaggi distinguono tre categorie:
 - ◆ parametri di ingresso (IN): sono passati dalla unità chiamante alla unità chiamata al momento dell'invocazione;
 - ◆ parametri di uscita (OUT): sono passati dall'unità chiamata alla unità chiamante al momento della terminazione della prima;



Parametri

Parametrizzazione di
procedure

Parametri

Tipi dei p.

Associazione dei p.

Esempio

Associazione di default

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Modalità di passaggio
parametri

Bibliografia

- Costituiscono il mezzo attraverso il quale le informazioni transitano esplicitamente tra l'unità chiamante e quella chiamata.
- Alcuni linguaggi distinguono tre categorie:
 - ◆ parametri di ingresso (IN): sono passati dalla unità chiamante alla unità chiamata al momento dell'invocazione;
 - ◆ parametri di uscita (OUT): sono passati dall'unità chiamata alla unità chiamante al momento della terminazione della prima;
 - ◆ parametri sia di ingresso che di uscita (IN-OUT): servono a far transitare le informazioni in entrambe le direzioni.



Parametri

Parametrizzazione di
procedure

Parametri

Tipi dei p.

Associazione dei p.

Esempio

Associazione di default

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Modalità di passaggio
parametri

Bibliografia

- Costituiscono il mezzo attraverso il quale le informazioni transitano esplicitamente tra l'unità chiamante e quella chiamata.
- Alcuni linguaggi distinguono tre categorie:
 - ◆ parametri di ingresso (IN): sono passati dalla unità chiamante alla unità chiamata al momento dell'invocazione;
 - ◆ parametri di uscita (OUT): sono passati dall'unità chiamata alla unità chiamante al momento della terminazione della prima;
 - ◆ parametri sia di ingresso che di uscita (IN-OUT): servono a far transitare le informazioni in entrambe le direzioni.
- Devono essere specificati in due punti:



Parametri

Parametrizzazione di
procedure

Parametri

Tipi dei p.

Associazione dei p.

Esempio

Associazione di default

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Modalità di passaggio
parametri

Bibliografia

- Costituiscono il mezzo attraverso il quale le informazioni transitano esplicitamente tra l'unità chiamante e quella chiamata.
- Alcuni linguaggi distinguono tre categorie:
 - ◆ parametri di ingresso (IN): sono passati dalla unità chiamante alla unità chiamata al momento dell'invocazione;
 - ◆ parametri di uscita (OUT): sono passati dall'unità chiamata alla unità chiamante al momento della terminazione della prima;
 - ◆ parametri sia di ingresso che di uscita (IN-OUT): servono a far transitare le informazioni in entrambe le direzioni.
- Devono essere specificati in due punti:
 - ◆ nella definizione della procedura: *parametri formali*;



Parametri

Parametrizzazione di
procedure

Parametri

Tipi dei p.

Associazione dei p.

Esempio

Associazione di default

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Modalità di passaggio
parametri

Bibliografia

- Costituiscono il mezzo attraverso il quale le informazioni transitano esplicitamente tra l'unità chiamante e quella chiamata.
- Alcuni linguaggi distinguono tre categorie:
 - ◆ parametri di ingresso (IN): sono passati dalla unità chiamante alla unità chiamata al momento dell'invocazione;
 - ◆ parametri di uscita (OUT): sono passati dall'unità chiamata alla unità chiamante al momento della terminazione della prima;
 - ◆ parametri sia di ingresso che di uscita (IN-OUT): servono a far transitare le informazioni in entrambe le direzioni.
- Devono essere specificati in due punti:
 - ◆ nella definizione della procedura: *parametri formali*;
 - ◆ nelle invocazioni della procedura: *parametri attuali* o *argomenti*.



Tipi dei parametri

[Parametrizzazione di
procedure](#)

[Parametri](#)

[Tipi dei p.](#)

[Associazione dei p.](#)

[Esempio](#)

[Associazione di default](#)

[Ordine di valutazione dei
param. attuali](#)

[Ordine di valutazione dei
param. attuali](#)

[Ordine di valutazione dei
param. attuali](#)

[Modalità di passaggio
parametri](#)

[Bibliografia](#)

- **Linguaggi staticamente tipati:** nella definizione deve essere specificato il tipo dei parametri formali; nell'invocazione è richiesta la compatibilità di tipo tra parametri formali e attuali (es.: Java, C, C++, etc.)



Tipi dei parametri

[Parametrizzazione di
procedure](#)

[Parametri](#)

[Tipi dei p.](#)

[Associazione dei p.](#)

[Esempio](#)

[Associazione di default
Ordine di valutazione dei
param. attuali](#)

[Ordine di valutazione dei
param. attuali](#)

[Ordine di valutazione dei
param. attuali](#)

[Modalità di passaggio
parametri](#)

[Bibliografia](#)

- **Linguaggi staticamente tipati:** nella definizione deve essere specificato il tipo dei parametri formali; nell'invocazione è richiesta la compatibilità di tipo tra parametri formali e attuali (es.: Java, C, C++, etc.)
- **Linguaggi dinamicamente tipati:** i parametri formali non hanno alcun vincolo di tipo; il legame di tipo si instaura durante l'esecuzione (run time) allo stesso tipo dei parametri attuali (impossibile il type checking in compilazione) (es.: Python)



Tipi dei parametri

[Parametrizzazione di
procedure](#)

[Parametri](#)

[Tipi dei p.](#)

[Associazione dei p.](#)

[Esempio](#)

[Associazione di default](#)

[Ordine di valutazione dei
param. attuali](#)

[Ordine di valutazione dei
param. attuali](#)

[Ordine di valutazione dei
param. attuali](#)

[Modalità di passaggio
parametri](#)

[Bibliografia](#)

- **Linguaggi staticamente tipati:** nella definizione deve essere specificato il tipo dei parametri formali; nell'invocazione è richiesta la compatibilità di tipo tra parametri formali e attuali (es.: Java, C, C++, etc.)
- **Linguaggi dinamicamente tipati:** i parametri formali non hanno alcun vincolo di tipo; il legame di tipo si instaura durante l'esecuzione (run time) allo stesso tipo dei parametri attuali (impossibile il type checking in compilazione) (es.: Python)



Associazione dei parametri

[Parametrizzazione di procedure](#)

[Parametri](#)

[Tipi dei p.](#)

[Associazione dei p.](#)

[Esempio](#)

[Associazione di default](#)

[Ordine di valutazione dei param. attuali](#)

[Ordine di valutazione dei param. attuali](#)

[Ordine di valutazione dei param. attuali](#)

[Modalità di passaggio parametri](#)

[Bibliografia](#)

Come si stabilisce la corrispondenza tra parametri attuali e formali

Due **metodi di associazione**:

- *per posizione*: a seconda della posizione relativa nella sequenza dei parametri;
- *per nome*: il nome del parametro formale è aggiunto come prefisso al parametro attuale;



Esempio

Parametrizzazione di
procedure

Parametri

Tipi dei p.

Associazione dei p.

Esempio

Associazione di default

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Modalità di passaggio
parametri

Bibliografia

Data l'intestazione della seguente procedura (linguaggio ADA):

```
procedure TEST (A: in Atype; b: in out Btype; C: out Ctype)
```



Esempio

[Parametrizzazione di procedure](#)

[Parametri](#)

[Tipi dei p.](#)

[Associazione dei p.](#)

Esempio

[Associazione di default](#)

[Ordine di valutazione dei param. attuali](#)

[Ordine di valutazione dei param. attuali](#)

[Ordine di valutazione dei param. attuali](#)

[Modalità di passaggio parametri](#)

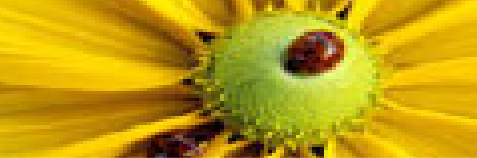
[Bibliografia](#)

Data l'intestazione della seguente procedura (linguaggio ADA):

```
procedure TEST (A: in Atype; b: in out Btype; C: out Ctype)
```

allora una invocazione che usi associazione per posizione è:

```
TEST (X, Y, Z);
```

Esempio

Parametrizzazione di procedure

Parametri

Tipi dei p.

Associazione dei p.

Esempio

Associazione di default

Ordine di valutazione dei param. attuali

Ordine di valutazione dei param. attuali

Ordine di valutazione dei param. attuali

Modalità di passaggio parametri

Bibliografia

Data l'intestazione della seguente procedura (linguaggio ADA):

```
procedure TEST (A: in Atype; b: in out Btype; C: out Ctype)
```

allora una invocazione che usi associazione per posizione è:

```
TEST (X, Y, Z);
```

mentre una che usi associazione per nome può essere:

```
TEST (A=>X, C=>Z, b=>Y);
```



Associazione di default

Parametrizzazione di
procedure

Parametri

Tipi dei p.

Associazione dei p.

Esempio

Associazione di default

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Modalità di passaggio
parametri

Bibliografia

Un'ulteriore tecnica è la cosiddetta *associazione di default*

Essa permette di specificare valori di default per quei parametri formali che non sono stati legati a valori da parametri attuali.

Esempio in C++:

```
void print_error_msg(int line,  
                    string message = "Generic_error")  
{ ... }
```

Questa funzione si può invocare con due argomenti o con un argomento solo (line)



Ordine di valutazione dei param. attuali

Parametrizzazione di
procedure

Parametri

Tipi dei p.

Associazione dei p.

Esempio

Associazione di default

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Modalità di passaggio
parametri

Bibliografia

- I parametri attuali possono essere espressioni arbitrarie
- Tali espressioni possono contenere *effetti collaterali*
 - ◆ Per effetto collaterale si intende qualsiasi effetto osservabile che esuli dal semplice calcolo del valore dell'espressione
- In tal caso, può essere rilevante l'ordine in cui i parametri attuali vengono valutati a runtime
- Il linguaggio può specificare o meno un ordine di valutazione fisso



Ordine di valutazione dei param. attuali

Parametrizzazione di
procedure

Parametri

Tipi dei p.

Associazione dei p.

Esempio

Associazione di default

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Modalità di passaggio
parametri

Bibliografia

Qual è l'output del seguente programma in linguaggio C?

```
void foo(char a, char b) {  
    printf("%c%c\n", a, b);  
}  
  
int main(void) {  
    char msg[] = "ciao";  
    int i = 0;  
    foo(msg[i], msg[++i]);  
    return 0;  
}
```

- Il parametro attuale `msg[++i]` contiene provoca un effetto collaterale (l'incremento di `i`)
- L'output dipende dall'ordine di valutazione dei parametri di `foo`!



Ordine di valutazione dei param. attuali

Parametrizzazione di
procedure

Parametri

Tipi dei p.

Associazione dei p.

Esempio

Associazione di default

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Ordine di valutazione dei
param. attuali

Modalità di passaggio
parametri

Bibliografia

Qual è l'output del seguente programma in linguaggio C?

```
void foo(char a, char b) {  
    printf("%c%c\n", a, b);  
}  
  
int main(void) {  
    char msg[] = "ciao";  
    int i = 0;  
    foo(msg[i], msg[++i]);  
    return 0;  
}
```

- Il parametro attuale `msg[++i]` contiene provoca un effetto collaterale (l'incremento di `i`)
- L'output dipende dall'ordine di valutazione dei parametri di `foo`!
- L'output può essere "ci" oppure "ii", a seconda del compilatore usato (provare su Compiler Explorer)
- Il linguaggio C non stabilisce un ordine fisso!



Ordine di valutazione in diversi linguaggi

Parametrizzazione di procedure

Parametri

Tipi dei p.

Associazione dei p.

Esempio

Associazione di default

Ordine di valutazione dei param. attuali

Ordine di valutazione dei param. attuali

Ordine di valutazione dei param. attuali

Modalità di passaggio parametri

Bibliografia

| Linguaggio | Ordine di valutazione |
|------------|-----------------------|
| C/C++ | Non specificato |
| Java | Da sinistra a destra |
| Python | Da sinistra a destra |



Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

Modalità di passaggio parametri



Parametri IN

[Parametrizzazione di
procedure](#)

[Modalità di passaggio
parametri](#)

[Parametri IN](#)

[Parametri OUT](#)

[Esempio OUT](#)

[Parametri IN-OUT](#)

[Esempio IN-OUT](#)

[Esempi](#)

[Aliasing](#)

[Procedure come . . .](#)

[Macro](#)

[Esercizio](#)

[Funzioni](#)

[Bibliografia](#)

Possono essere realizzati in due modi:

1. Per riferimento

- la locazione del parametro attuale diventa la locazione del parametro formale
- si deve impedire (a tempo di compilazione, se possibile) la modifica all'interno della procedura



Parametri IN

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

Possono essere realizzati in due modi:

1. Per riferimento

- la locazione del parametro attuale diventa la locazione del parametro formale
- si deve impedire (a tempo di compilazione, se possibile) la modifica all'interno della procedura

2. Per copia

- il valore del parametro attuale viene copiato in una nuova locazione, quella del parametro formale



Parametri IN

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedures come . . .

Macro

Esercizio

Funzioni

Bibliografia

Possono essere realizzati in due modi:

1. Per riferimento

- la locazione del parametro attuale diventa la locazione del parametro formale
- si deve impedire (a tempo di compilazione, se possibile) la modifica all'interno della procedura

2. Per copia

- il valore del parametro attuale viene copiato in una nuova locazione, quella del parametro formale
- parametro formale visto come variabile locale



Parametri IN

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

Possono essere realizzati in due modi:

1. Per riferimento

- la locazione del parametro attuale diventa la locazione del parametro formale
- si deve impedire (a tempo di compilazione, se possibile) la modifica all'interno della procedura

2. Per copia

- il valore del parametro attuale viene copiato in una nuova locazione, quella del parametro formale
- parametro formale visto come variabile locale
- modifica permessa, perché valida solo nell'ambiente di esecuzione della procedura.



Parametri IN

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

Possono essere realizzati in due modi:

1. Per riferimento

- la locazione del parametro attuale diventa la locazione del parametro formale
- si deve impedire (a tempo di compilazione, se possibile) la modifica all'interno della procedura

2. Per copia

- il valore del parametro attuale viene copiato in una nuova locazione, quella del parametro formale
- parametro formale visto come variabile locale
- modifica permessa, perché valida solo nell'ambiente di esecuzione della procedura.

Il secondo modo è meno efficiente del primo, sia rispetto allo spazio sia al tempo, ma garantisce automaticamente che il parametro attuale non sia modificato



Parametri OUT

[Parametrizzazione di
procedure](#)

[Modalità di passaggio
parametri](#)

[Parametri IN](#)

[Parametri OUT](#)

[Esempio OUT](#)

[Parametri IN-OUT](#)

[Esempio IN-OUT](#)

[Esempi](#)

[Aliasing](#)

[Procedure come . . .](#)

[Macro](#)

[Esercizio](#)

[Funzioni](#)

[Bibliografia](#)

Possono essere realizzati:

1. Per riferimento
2. Per copia (cioè per *risultato*)
 - La copia avviene all'uscita



Parametri OUT

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

Possono essere realizzati:

1. Per riferimento
2. Per copia (cioè per *risultato*)
 - La copia avviene all'uscita

Rappresentano risultati \Rightarrow bisogna proibirne la “lettura”, ad es.

- uso a destra di un assegnamento
- passaggio a un parametro IN o IN-OUT di un'altra procedura

Non esistono regole generali, nemmeno tra diverse versioni di uno stesso linguaggio

- Ada 83 proibisce di “leggere” i parametri OUT
- Le versioni successive invece lo permettono

Nei nostri esercizi: errore di compilazione se un parametro OUT viene letto senza essere stato inizializzato



Esempio OUT

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come ...

Macro

Esercizio

Funzioni

Bibliografia

Determinare l'output (linguaggio didattico pseudo-Pascal):

```
program main (input, output);  
  var a,b,c: integer;  
  
  procedure p1 (OUT a,b: integer);  
  begin  
    a := a*b;  
    if (c/b)=a then a:=0 else a:=100  
  end;  
  
begin  
  a := 5; b := 6; c := 7;  
  p1(b, c);  
  writeln(a, b, c);  
end.
```

L'output dipende dal tipo di scoping? (statico/dinamico)

L'output dipende dall'implementazione di OUT? (rif/valore)



Esempio OUT

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

Determinare l'output (linguaggio didattico pseudo-Pascal):

```
program main (input, output);  
  var a,b,c: integer;  
  
  procedure p1 (OUT a,b: integer);  
  begin  
    a := a*b;  
    if (c/b)=a then a:=0 else a:=100  
  end;  
  
begin  
  a := 5; b := 6; c := 7;  
  p1(b, c);  
  writeln(a, b, c);  
end.
```

L'output dipende dal tipo di scoping? (statico/dinamico)

L'output dipende dall'implementazione di OUT? (rif/valore)

Soluzione:

Errore di compilazione

Non dipende dal tipo di scoping, né dall'implementazione di OUT



Parametri IN-OUT

Sono la combinazione dei due precedenti.

[Parametrizzazione di
procedure](#)

[Modalità di passaggio
parametri](#)

[Parametri IN](#)

[Parametri OUT](#)

[Esempio OUT](#)

[Parametri IN-OUT](#)

[Esempio IN-OUT](#)

[Esempi](#)

[Aliasing](#)

[Procedure come . . .](#)

[Macro](#)

[Esercizio](#)

[Funzioni](#)

[Bibliografia](#)



Parametri IN-OUT

[Parametrizzazione di
procedure](#)

[Modalità di passaggio
parametri](#)

[Parametri IN](#)

[Parametri OUT](#)

[Esempio OUT](#)

[Parametri IN-OUT](#)

[Esempio IN-OUT](#)

[Esempi](#)

[Aliasing](#)

[Procedure come . . .](#)

[Macro](#)

[Esercizio](#)

[Funzioni](#)

[Bibliografia](#)

Sono la combinazione dei due precedenti. Anch'essi possono essere realizzati:

1. Per riferimento

- non ci sono limitazioni all'uso all'interno della procedura



Parametri IN-OUT

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

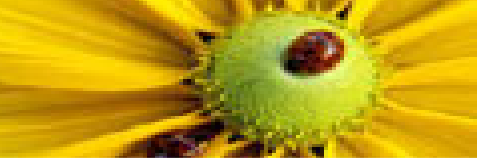
Esercizio

Funzioni

Bibliografia

Sono la combinazione dei due precedenti. Anch'essi possono essere realizzati:

1. Per riferimento
 - non ci sono limitazioni all'uso all'interno della procedura
2. Per copia (per *valore-risultato*)
 - avvengono *due processi di copia*: uno durante l'attivazione ed uno al termine della procedura



Esempio IN-OUT

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

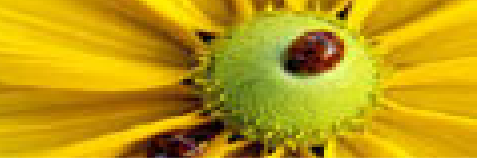
Bibliografia

Determinare l'output (linguaggio didattico pseudo-Pascal):

```
program main (input, output);  
  var a,b,c: integer;  
  
  procedure p1 (IN-OUT a,b: integer);  
  begin  
    a := a*b;  
    if (c/b)=a then a:=0 else a:=100  
  end;  
  
begin  
  a := 5; b := 6; c := 7;  
  p1(b, c);  
  writeln(a, b, c);  
end.
```

L'output dipende dal tipo di scoping? (statico/dinamico)

L'output dipende dall'implementazione di IN-OUT? (rif/valore)



Esempio IN-OUT

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

Determinare l'output (linguaggio didattico pseudo-Pascal):

```
program main (input, output);  
  var a,b,c: integer;  
  
  procedure p1 (IN-OUT a,b: integer);  
  begin  
    a := a*b;  
    if (c/b)=a then a:=0 else a:=100  
  end;  
  
begin  
  a := 5; b := 6; c := 7;  
  p1(b, c);  
  writeln(a, b, c);  
end.
```

L'output dipende dal tipo di scoping? (statico/dinamico)

L'output dipende dall'implementazione di IN-OUT? (rif/valore)

Soluzione:

5 100 7

Non dipende dal tipo di scoping, né dall'implementazione di IN-OUT



Esempi

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

| Linguaggio | IN | OUT | IN-OUT |
|---------------------|----|-----|----------------------------------|
| Ada | Sì | Sì | Sì |
| Pascal | Sì | No | Sì (keyword var) |
| C | Sì | No | No. Emulato con puntatori |
| C++ | Sì | No | Sì (con riferimenti) |
| Java ¹ | Sì | No | No. Emulato con oggetti mutabili |
| Python ² | Sì | No | No. Emulato con oggetti mutabili |

Nota 1: in Java, tutti i parametri, primitivi e non, vengono passati in modalità “IN per valore”, ma le variabili di tipo non primitivo sono riferimenti.

Nota 2: in Python, tutte le variabili sono riferimenti, e vengono passati in modalità “IN per valore”.



Aliasing

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

È la possibilità di riferirsi alla stessa locazione con nomi diversi.



Aliasing

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

È la possibilità di riferirsi alla stessa locazione con nomi diversi.
Nel passaggio dei parametri può causare notevoli problemi di interpretazione.



Aliasing

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come ...

Macro

Esercizio

Funzioni

Bibliografia

È la possibilità di riferirsi alla stessa locazione con nomi diversi.

Nel passaggio dei parametri può causare notevoli problemi di interpretazione. Per esempio:

```
program MAIN;  
  var  
    A: integer;  
  procedure TEST (X, Y: IN-OUT integer);  
  begin  
    X := A + Y;  
    writeln(A, X, Y)  
  end;  
begin  
  A := 1;  
  TEST(A, A);  
  writeln(A)  
end.
```

Esercizio: determinare l'output del programma nel caso in cui i parametri siano realizzati *per riferimento* e nel caso in cui siano realizzati *per copia*.



Procedure come parametri di procedura

Alcuni linguaggi permettono l'uso di procedure come argomento di altre procedure.

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

Procedure come parametri di procedura

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come ...

Macro

Esercizio

Funzioni

Bibliografia

Alcuni linguaggi permettono l'uso di procedure come argomento di altre procedure.

Esempio:

```
program MAIN;
  VAR a: real;
  procedure TESTPOS (X: real; procedure ERROR (MSG: string));
    begin
      if X <= 0 then ERROR ('Negative X in TESTPOS ')
    end;
  procedure E1 (M: string);
    begin
      writeln('E1 error: ', M)
    end;
  procedure E2 (M: string);
    begin
      writeln('E2 error: ', M)
    end;
begin
  readln (A);
  TESTPOS(A, E1);
  TESTPOS(A, E2)
end.
```



Macro

[Parametrizzazione di
procedure](#)

[Modalità di passaggio
parametri](#)

[Parametri IN](#)

[Parametri OUT](#)

[Esempio OUT](#)

[Parametri IN-OUT](#)

[Esempio IN-OUT](#)

[Esempi](#)

[Aliasing](#)

[Procedure come . . .](#)

Macro

[Esercizio](#)

[Funzioni](#)

[Bibliografia](#)

Generazione di un nuovo brano di codice sorgente (espansione della macro) in cui i nomi dei parametri attuali sostituiscono i nomi dei parametri formali.



Macro

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

Generazione di un nuovo brano di codice sorgente (espansione della macro) in cui i nomi dei parametri attuali sostituiscono i nomi dei parametri formali.

Esempio: data la procedura

```
procedure swap (a, b: integer);  
  var temp: integer;  
  begin  
    temp:= a;  
    a:= b;  
    b:= temp  
  end;
```

allora la chiamata `swap(x, y)` viene sostituita dal seguente brano di codice:

```
temp:= x;  
x:= y;  
y:= temp;
```



Esercizio

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedures come . . .

Macro

Esercizio

Funzioni

Bibliografia

Determinare i problemi che nascono dai due programmi, se `swap` è una macro:

```
program main;
var
  i: integer;
  m: array[1..100] of integer;
  ...
begin
  ...
  swap(i, m[i]);
  ...
end.
```



Esercizio

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedures come . . .

Macro

Esercizio

Funzioni

Bibliografia

Determinare i problemi che nascono dai due programmi, se `swap` è una macro:

```
program main;
var
  i: integer;
  m: array[1..100] of integer;
  ...
begin
  ...
  swap(i, m[i]);
  ...
end.
```

```
program main;
var
  i, temp: integer;
  ...
begin
  ...
  swap(i, temp);
  ...
end.
```



Funzioni

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

Sono procedure che **restituiscono un valore** alla procedura chiamante.



Funzioni

[Parametrizzazione di procedure](#)

[Modalità di passaggio parametri](#)

[Parametri IN](#)

[Parametri OUT](#)

[Esempio OUT](#)

[Parametri IN-OUT](#)

[Esempio IN-OUT](#)

[Esempi](#)

[Aliasing](#)

[Procedure come . . .](#)

[Macro](#)

[Esercizio](#)

[Funzioni](#)

[Bibliografia](#)

Sono procedure che **restituiscono un valore** alla procedura chiamante.

Sono realizzate:

1. utilizzando un'istruzione “return” per restituire esplicitamente il controllo al chiamante, inviando allo stesso tempo il valore di un'espressione.

Sono procedure che **restituiscono un valore** alla procedura chiamante.

Sono realizzate:

1. utilizzando un'istruzione “return” per restituire esplicitamente il controllo al chiamante, inviando allo stesso tempo il valore di un'espressione.
2. oppure creando una *pseudovariabile* nell'ambiente locale della procedura chiamata. Tale variabile può essere solo modificata; non è possibile l'accesso in lettura.

Esempio in MATLAB:

```
function [x1, x2, x3] = foo()  
    x1 = rand(1);           % a scalar  
    x2 = rand(1, 10);       % a 1x10 vector  
    x3 = rand(10, 10);      % a 10x10 matrix  
end
```



Funzioni

Parametrizzazione di
procedure

Modalità di passaggio
parametri

Parametri IN

Parametri OUT

Esempio OUT

Parametri IN-OUT

Esempio IN-OUT

Esempi

Aliasing

Procedure come . . .

Macro

Esercizio

Funzioni

Bibliografia

Sono procedure che **restituiscono un valore** alla procedura chiamante.

Sono realizzate:

1. utilizzando un'istruzione "return" per restituire esplicitamente il controllo al chiamante, inviando allo stesso tempo il valore di un'espressione.
2. oppure creando una *pseudovariabile* nell'ambiente locale della procedura chiamata. Tale variabile può essere solo modificata; non è possibile l'accesso in lettura.

Esempio in MATLAB:

```
function [x1, x2, x3] = foo()  
    x1 = rand(1);           % a scalar  
    x2 = rand(1, 10);       % a 1x10 vector  
    x3 = rand(10, 10);      % a 10x10 matrix  
end
```

3. oppure l'intero corpo della funzione è un'espressione (tipico dei linguaggi funzionali).

Esempio in Java:

```
Function<Double,Double> f = x -> 2*x;
```



Bibliografia

[Parametrizzazione di
procedure](#)

[Modalità di passaggio
parametri](#)

[Bibliografia](#)

Bibliografia

Capitolo 9 (“Astrarre sul controllo”) di *Linguaggi di Programmazione, principi e paradigmi*, di Gabbrielli e Martini (2a edizione)