

## PASSAGGIO PARAMETRI:

Non Solo Funzioni Vuote che chiamo per non duplicare codice ma voglio ASTRARRE DI PIÙ per COMUNICARE tra chiamato e chiamante

**IN-MODE** ⇒ Tra **chiamante e Chiamato**, in ENTRATA alla Procedura Chiamata **[PASSAGGIO PER VALORE]**

**OUT-MODE** ⇒ Dal **Chiamato al Chiamante**, MENO USATO visto che si usa il RETURN di solito **[PASSAGGIO PER RISULTATO]**

**INOUT-MODE** ⇒ **Bidirezionale** **[PASSAGGIO PER RIFERIMENTO O ANCHE VALORE-RISULTATO]**

Valore di Ritorno è una Comunicazione dal Chiamato al Chiamante

## PASSAGGIO PER VALORE:

**Parametri Attuali Sono Valutati** (possono essere EXPRESSION) ed il Valore OTTENUTO è Copiato nei Parametri formali.

Siamo Nella Metodologia IN-MODE.

**Si CREA delle Vere e proprie COPIE** visto che CREO Nuove Locazioni di MEMORIA (che inizialmente Coincidono)

Modifica al Parametro formale NON Sono Visibili alle proced. chiamante. [MODIFICHE SONO LOCALI, DOPO INIZIALIZZAZIONE NESSUN LEGAME]

Se i Valori passati ha Grandi DIMENSIONI NON è per nulla Efficiente

```
void foo (int x) { x := x + 1 }
```

...

```
y = 1
```

```
foo (y + 1)
```

→ Param. formale Allocato Nello STACK

→ Valutato  $y + 1 = 2$  e 2 usato per Inizializzare  $x$  di foo

NON c'è Nessun legame Tra  $y$  del Chiamante e  $x$  del chiamato

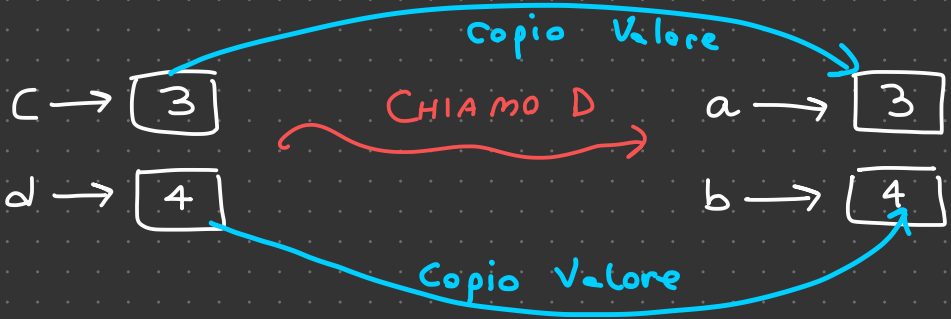
$x$  Viene Eliminata all'uscita da foo

## COSA SUCCEDE IN MEMORIA:

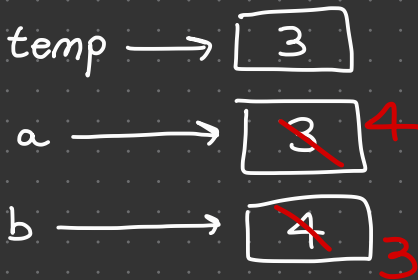
```
void Swap(Value int a, Value int b) {  
    int Temp = a;  
    int a = b;  
    b = Temp;  
}
```

Swap(c, d)  $\rightarrow$   $f = [c \mapsto 3, d \mapsto 4]$

main:



Eseguo Swap:



Esco da  
Swap

Elimino le  
Memorie Allocate  
 $a, b, Temp$

ALL' USCITA di SWAP (c, d) NON Sono Cambiati

## PASSAGGIO PER RIFERIMENTO:

Viene Passato dal Chiamante al Chiamato l'indirizzo del parametro ATTUALE

Ora NON posso Mettere un ESPRESSIONE (l-Value) ma un r-Value ovvero una LOCAZIONE

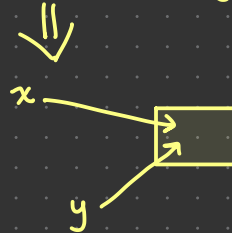
È una Metodologia InOut... Sto Modificando con un Altro Nome la Stessa Cella.

NON devo fare NESSUNA COPIA però devo USARE i Riferimento, Quindi Sarò più LENTO.

```
void foo(reference int x) { x = x + 1; }  
...  
y = 1
```

modifica del Parametro Attuale va a Modificare la cella alla Quale si Riferisce

foo(y) → Prendo l'indirizzo di y e lo Associo ad x e Quindi ho che x è ALIAS di y



All' USCITA elimino il nome x, ma NON il suo CONTENUTO perché è legato ad y.

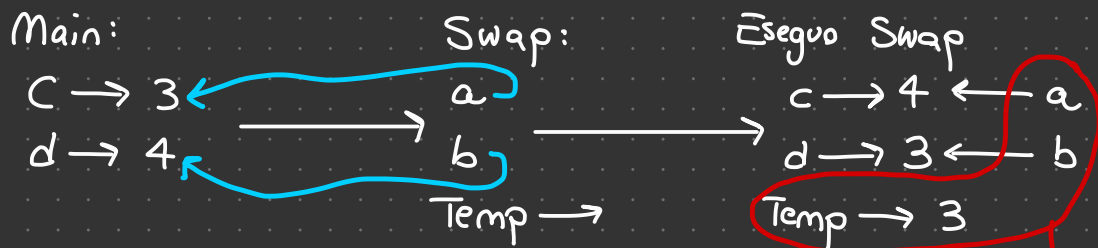
Tutte le MODIFICHE al Parametro Attuale Sono MODIFICHE  
Permanenti al Valore del PARAMETRO ATTUALE

(anche Quando Esco Vedo le MODIFICHE su y)

```
void Swap(Reference int a, Reference int b) {  
    int Temp = a;  
    int a = b;  
    b = Temp;  
}
```

Swap(c, d)  $\rightarrow$   $\mathcal{S} = [c \mapsto 3, d \mapsto 4]$

La Memoria finale Sarà  $\Rightarrow \mathcal{S} = [c \mapsto 4, d \mapsto 3]$



Puntatori Alle Celle del MAIN

ALL' USCITA di SWAP  
LI ELIMINO

```

void fie (ref int x, ref int y) {
    y = 2;
    x = 5;
    if (x == y) Then y = y + 2;
}

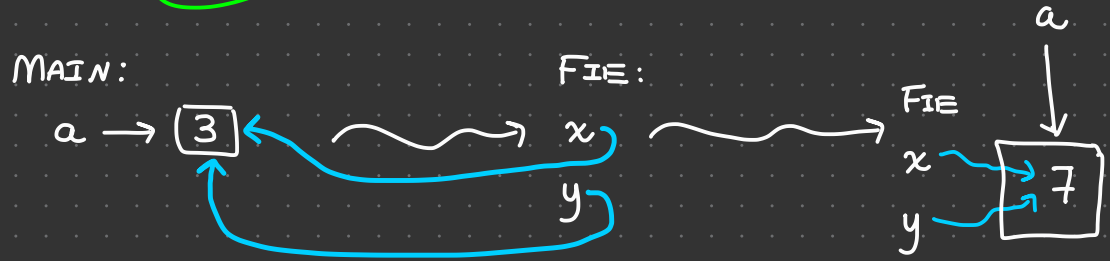
```

```

int a = 3;

```

fie (a, a)
 Parametro Attuale è lo Stesso Sia per x che y



Passare come Parametri una Var. Globale è INUTILE, Ne Avrai Accesso Comunque

# FUNZIONI DI ORDINE SUPERIORE:

Procedere Come Parametri di Altre Procedure

ATTUALI-FORMALI

RESTITUZIONE DI UNA  
FUNZIONE

`int x = 4;` Ambiente dove definisco `f`  $\Rightarrow$  SCOPING STATICO

`int z = 0;`

`int f(int y) { return x * y; }` Funzione con AMBIENTE NON LOCALE  $\Rightarrow$  regole SCOPING Necessarie

`int g(int h(int n)) {` SCOPING NON Necessario

`int x = 7;` Ambiente dove chiamo `f` (MEDIANTE `h`)  $\Rightarrow$  SHALLOW

`int z = h(3) + x;` Chiama `f` in Realtà

}

...

{ `int x = 5;` Ambiente dove CREO legame Tra attuale `f` e Formale `h`.

`g(f);` Come PARAMETRO

}

$\Downarrow$   
DEEP

SERVONO anche Regole di BINDING e NON Solo SCOPING

SHALLOW BINDING:

DEEP BINDING:

Solo in Caso di SCOPING DINAMICO

Cosa SIGNIFICA DEFINIRE, IMPLEMENTARE?



SHALLOW BINDING:

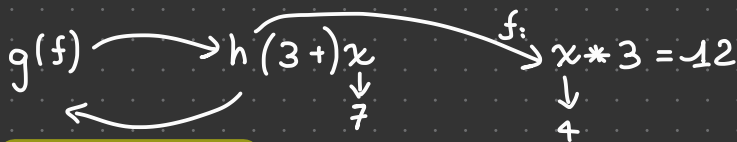
Sceglie l'Ambiente DINAMICO al Momento Effettivo della chiamata attraverso il PARAMETRO ATTUALE

DEEP BINDING:

Sceglie l'Ambiente DINAMICO al Momento della Creazione del legame tra ATTUALE e FORMALE (della fz. Passata come Parametro)

Nell'Esempio di Prima

SCOPING STATICO: ambiente Globale



$$z = 12 + 7 = 19$$

$$f \rightsquigarrow x=4$$

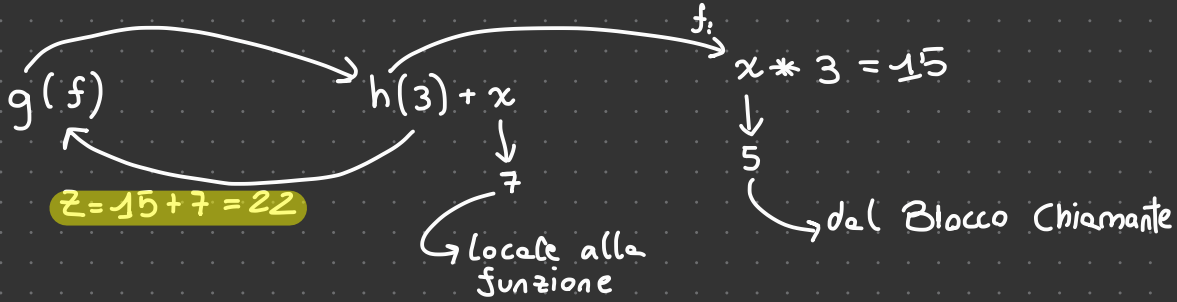
$$g \rightsquigarrow z=0$$



## SHALLOW BINDING:

$$f \rightsquigarrow x = 5$$

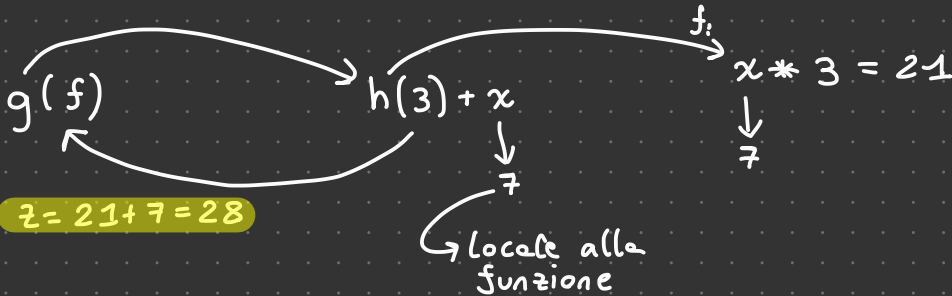
$$g \rightsquigarrow z = 0 \text{ (globale)}$$



## DEEP BINDING:

$$f \rightsquigarrow x = 7$$

$$g \rightsquigarrow z = 0 \text{ (Globale)}$$



① Individuare Variabili non Locali

② Serve BINDING?

③ Stabilire Ambiente di Riferimento