

# Programmazione imperativa

mercoledì 19 settembre 2018 11:14

(basate sul paradigma iterativo)

Stato (astratto)

AMBIENTE

insieme di  
associazioni (coppie)  
<nome, locazione>

x	l <sub>0</sub>
y	l <sub>1</sub>
z	l <sub>2</sub>

MEMORIA

insieme di  
associazioni  
(locazione, valore)

l <sub>0</sub>	10
l <sub>1</sub>	5
l <sub>2</sub>	15

Il valore di x sta nelle locazione di  
memoria associata a x nell'ambiente (l<sub>0</sub>)

I nomi de qui in poi saranno  
chiamati "VARIABILI"

- il loro valore in memoria può variare

Come si crea lo stato?

mercoledì 19 settembre 2018 11:23

Nei linguaggi di programmazione esistono dei costanti che si chiamano "DICHIARAZIONI" che hanno il compito di creare lo stato.

tipo

int x = 10;

dichiarazione delle variabili x che associano in memoria valori interi e inizialmente ha valore 10

EFFETTO: (significato o semantica)

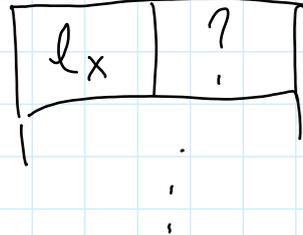
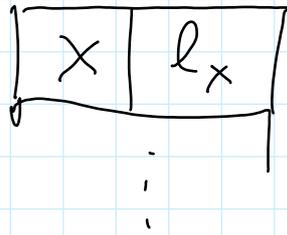
Creare una nuova associazione nell'ambiente e nella memoria

x	$l_x$
	⋮

$l_x$	10
	⋮

*int x;*

di dichiarazione SENZA  
inizializzazione



↓  
*int x = 10;*  
*int z = 25;*

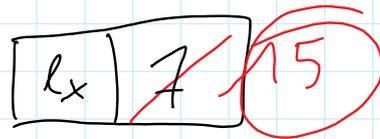
Come si modifica lo stato?

Attraverso altri costrutti che si chiamano "COMANDI"

### ASSEGNAMENTO

$\rightarrow$  nome  $\leftarrow$  espressione  
 si valuta l'espressione e il valore risultante viene assegnato, in memoria, alla variabile

$x = 10 + 5;$  assegnamento in C  
 valore dell'espressione  $\rightarrow 15$



Nei linguaggi imperativi le espressioni possono contenere variabili, es:  $10 + y$

valore di y nello stato corrente

int x = 10;

int y = 5;

x = 10 + y;

x = x + 1;

10 + 5

15 + 1

:=

y	l <sub>1</sub>
x	l <sub>0</sub>

l <sub>1</sub>	5
l <sub>0</sub>	10

~~15~~ 16



Altri comandi sono compositivi  
(anche completi) di anagrammi

### Comando condizionale

if (condizione) comando<sub>1</sub> else comando<sub>2</sub>

Se la condizione è vera viene eseguito comando<sub>1</sub> altrimenti viene eseguito comando<sub>2</sub>

```

int x = 10;
int y = 5;
if (x > 0) y = y + 2; else x = x + 1;

```

*(Note: In the original image, '10' is circled in blue and crossed out with a blue 'X'. 'y = y + 2;' is circled in red and crossed out with a blue 'X'. 'x = x + 1;' is circled in blue.)*

y	5
x	10

5	6
10	11

if (condizione) comando

se la condizione è vera si esegue

Comando altrimenti non si fa  
niente

```
int x = 10;  
int y = 5;  
if (x > 0)  
    if (y > 0) x = x + 1;  
    else y = y + 1;  
else y = y + 2;
```

int x = 10;

int y = 5;

if (x > 0)

if (y > 0) x = x + 1;

else y = y + 1;

if (x > 0)  
{ if (y > 0) x = x + 1; }

else y = y + 1;

## Sequenze di comandi

{.....}

Unico Comando

mercoledì 19 settembre 2018

11:55

```
int x = 10;
```

```
int y = 5;
```

```
if (x > 0)
```

```
{ x = x + 1;
```

```
  y = y + 1; }
```

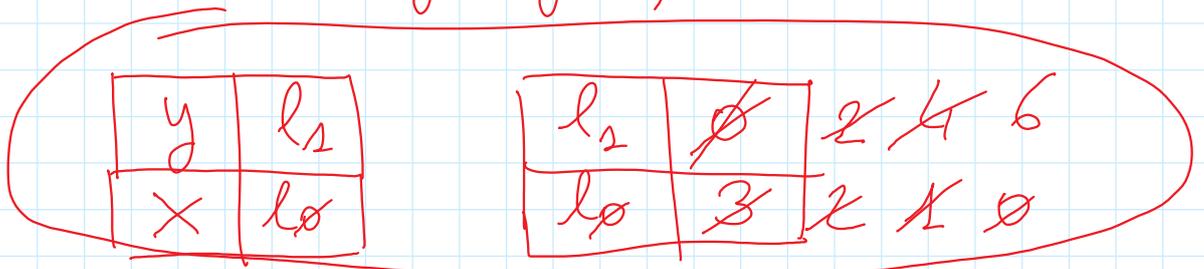
## Comando ITERATIVO (WHILE)

while (condizione) Comando

Fintanto che la condizione rimane  
Vera si esegue il comando

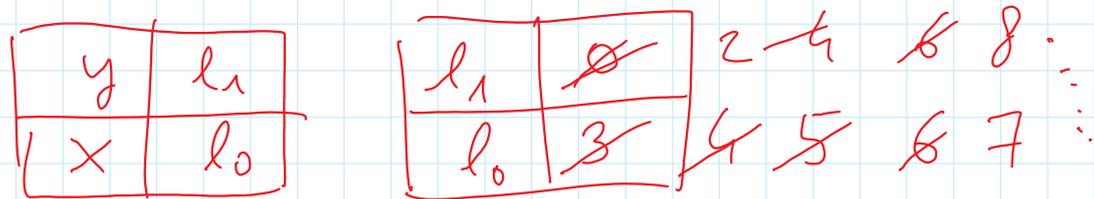
```

int x = 3;
int y = 0;
while (x > 0)
{
    x = x - 1;
    y = y + 2;
}
    
```



```

int x = 3;
int y = 0;
while (x > 0)
{
    x = x + 1;
    y = y + 2;
}
    
```



Il comando while permette di avere  
 COMPUTAZIONI INFINITE

# Comando iterativo FOR

FOR era met come comando di iterazione DETERMINATA (quando veniva scritto si sapeva quanti volte avrebbe iterato il comando interno)

while comando di iterazione INDETERMINATA

for ( $i=0$ ;  $i < 10$ ;  $i=i+1$ ) comando

↑  
inizializzazione delle variabili di controllo

↑  
Condizione di iterazione

↑  
incremento delle variabili di controllo da eseguire a ogni iterazione

~~$i=i-1$~~

Perché non si definiva un linguaggio in cui solamente l'iterazione determinata?

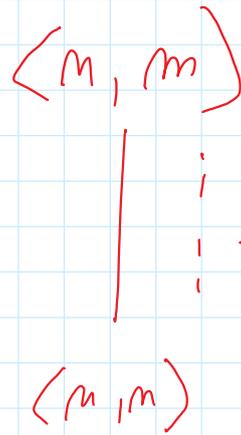
# Calcolo del Massimo Comune Divisore (MCD)

tra due valori naturali positivi:

$$\begin{aligned} \text{MCD}(m, m) &= m \\ \text{MCD}(m, m) &= \text{MCD}(m-m, m) \text{ se } m > m \\ \text{MCD}(m, m) &= \text{MCD}(m, m-m) \text{ se } m > m \end{aligned} \left. \begin{array}{l} m \in \mathbb{N}^+ \\ m \in \mathbb{N}^+ \end{array} \right\}$$

$$\begin{aligned} &\text{MCD}(21, 9) \\ = &\text{MCD}(12, 9) \\ = &\text{MCD}(3, 9) \\ = &\text{MCD}(3, 6) \\ = &\text{MCD}(3, 3) \\ = &3 \end{aligned}$$

i due valori appartengono a  $\mathbb{N}^+$



è un numero finito

```
int x = 21;  
int y = 9;  
while (x != y)  
    if (x > y) x = x - y;  
    else y = y - x;
```

iterazione  
indeterminata  
che  
termina

risultato

# Funzione fattoriale

$$0! = 1$$

$$n! = \underline{n \cdot (n-1) \cdot \dots \cdot 1}$$

per  $n > 0$

```
int x = 6;
```

```
int fact = 1;
```

```
int i;
```

```
for (i = x;  $i > 0$ ; i = i - 1)
```

```
    fact = fact * i;
```

$i > 1$

~~int x = 21;~~

~~int y = 3;~~

~~for (y = y; y != x; x = x)~~

~~if (x > y) x = x - y;~~

~~else y = y - x;~~

```
int x = 6;  
int fact = 1;  
int i = x;  
while (i > 1)  
{ fact = fact * i;  
  i = i - 1; }
```