

	Nome: _____ Matricola _____
--	--

Note alla esecuzione di programmi con oggetti:

Quando si ha a che fare con tipi riferimento (creati con new), oltre alle variabili locali ed ai parametri si hanno anche gli oggetti, che risiedono in un'area di memoria comune a tutto il programma che si chiama heap.

Esempio: nello statement `int [] a = new int[3];`

vengono creati:

una variabile riferimento di nome a nell'area locale

un oggetto array di tre elementi nello heap, comune a tutto il programma.

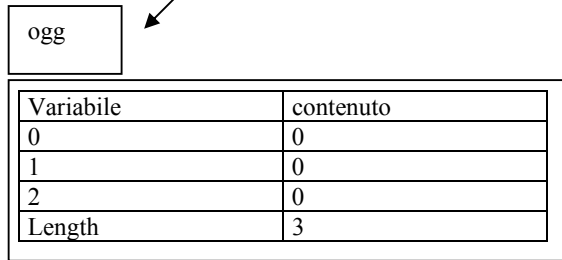
La variabile riferimento a contiene un riferimento (punta) ad un oggetto nello heap.

Area locale

Contenuto

A
punta all'oggetto ogg nello heap

Heap



Per risolvere l'esercizio, si può procedere ad occhio oppure con il metodo presentato nell'esercizio precedente (evoluzione dello stato statement per statement), arricchito con l'evoluzione dell'area globale.

Dunque,, può essere utile mantenere traccia non solo dell'evoluzione dell'area locale (variabili locali, parametri, passaggio dei parametri e valore di ritorno), ma anche:

1. di come vengono modificati gli oggetti nello heap.
2. di come vengono modificati (a cosa puntano) i riferimenti nell'area locale.

Adottando questa seconda strada, conviene aggiungere, oltre all'area locale dei metodi invocati nel programma e all'output, anche l'area globale (lo heap) e tenere traccia, in corrispondenza di ogni statement, non solo dell'evoluzione delle aree locali, ma anche dell'area globale per arrivare allo heap.

Esempio: effetto della esecuzione dei seguenti statement:

```

int [] a = new int[3];    // statement1
a[0]=1;                  // statement2
int [] b;                 // statement3
b=a;                     // statement4
b[1]=2;                  // statement5
a = new int [3];         // statement6
a[1] = 3;                 // statement7
system.out.println(b[0] + " " + a[0] + " " + b[1] + " " + a[1]);

```

Esecuzione statement e loro effetto

// statement 1: crea un array di tre elementi nell'area globale, e crea un riferimento di nome a che riferisce tale // array. L'array viene inizializzato a zero per ogni componente, e viene inizializzata la variabile length

//statement2: setta a[0] a 1. La modifica e' solo nell'area globale, nell'area locale non ho alcuna modifica

//statement3: crea un riferimento b ad array non inizializzato

// statement 4: b punta ad a

// statement 5: viene modificato b[1] (che e' lo stesso oggetto puntato da a)

// statement 6: viene creato un nuovo oggetto nell'area globale (ogg2) e il riferimento viene assegnato ad a

// statement 7: viene modificata la componente 1 dell'array riferito da a

// statement 8: vengono stampate le componenti indicate dei relativi oggetti

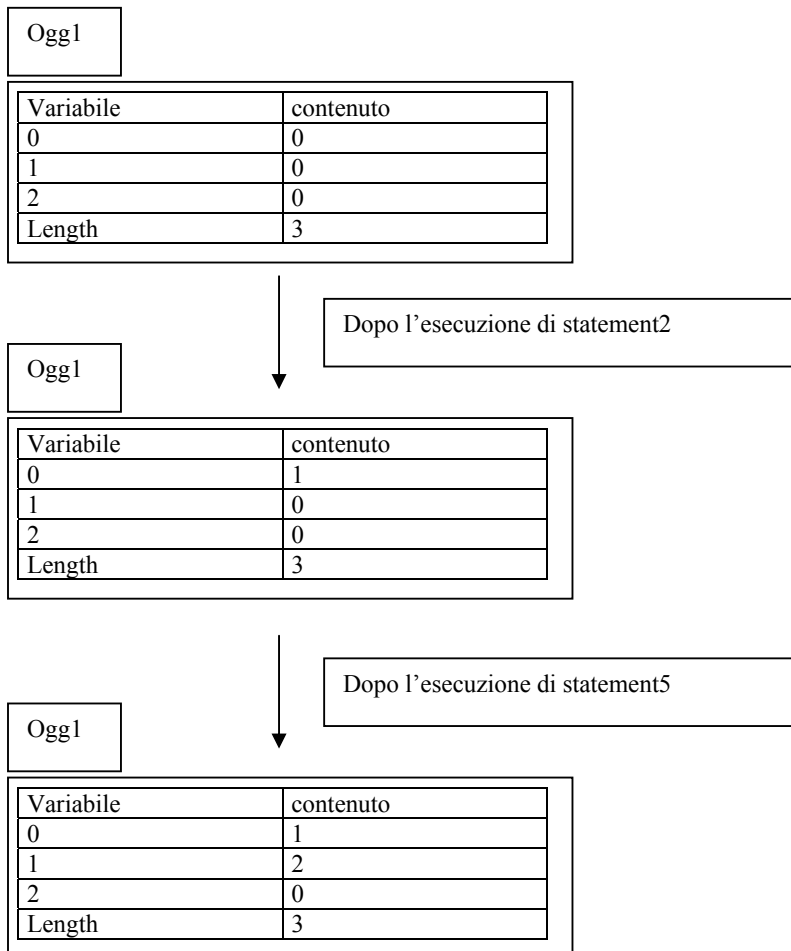
AREA LOCALE:

di seguito viene riportata l'evoluzione dei valori di a e b per l'area locale in corrispondenza di ogni statement

	A	B
Statement 1	Punta ogg1	
Statement 2	Punta ad ogg1	
Statement3	Punta ad ogg1	Creata e Non iniz.
Statement4	Punta ad ogg1	Punta ad ogg1
Statement5		
Statement6	Punta ad ogg2	Punta ad ogg1
Statement7	Punta ad ogg2	Punta ad ogg1

AREA GLOBALE

di seguito viene riportata l'evoluzione dell'area globale in corrispondenza degli statement significativi





Dopo l'esecuzione di statement6

Ogg1

Variabile	contenuto
0	1
1	2
2	0
Length	3

Ogg2

Variabile	contenuto
0	0
1	3
2	0
Length	3

OUTPUT
1 0 2 3

Risoluzione dell'esercizio

1) etichettatura degli statement

```
public class test {
    public static double f(int a[]) // int_f
    {
        int i=3;           //f1
        a[0]=0;           //f2
        a[1]=1;           //f3
        a[2]=2;           //f4
        System.out.println(i); //f5
    }

    public static void main(String [] args) //int_main
    {
        int a = 5;           //main1
        int [] b = new int[3]; //main2
        int [] c = new int [3]; //main3
        f(c);                //main4
        System.out.println(a + " " + b[0] + " " + c[1]); //main5
    }
}
```

Esecuzione statement main

//da main1 a main3 e' col solito significato
// main4: invoca f ed il parametro e' il rif. Ad ogg2. l'esecuzione salta ad f
// main5: stampa i valori di a, b[0] e c[1].

Area Locale main

statement	Parametro – variabile locale		
	a	b	c
Main1	5	Non creato	Non creato
Main2	5	Punta ogg1	Non creato
Main3	5	Punta ogg1	Punta ogg2
Main4	5	Punta ogg1	Punta ogg2
Main5	5	Punta ogg1	Punta ogg2

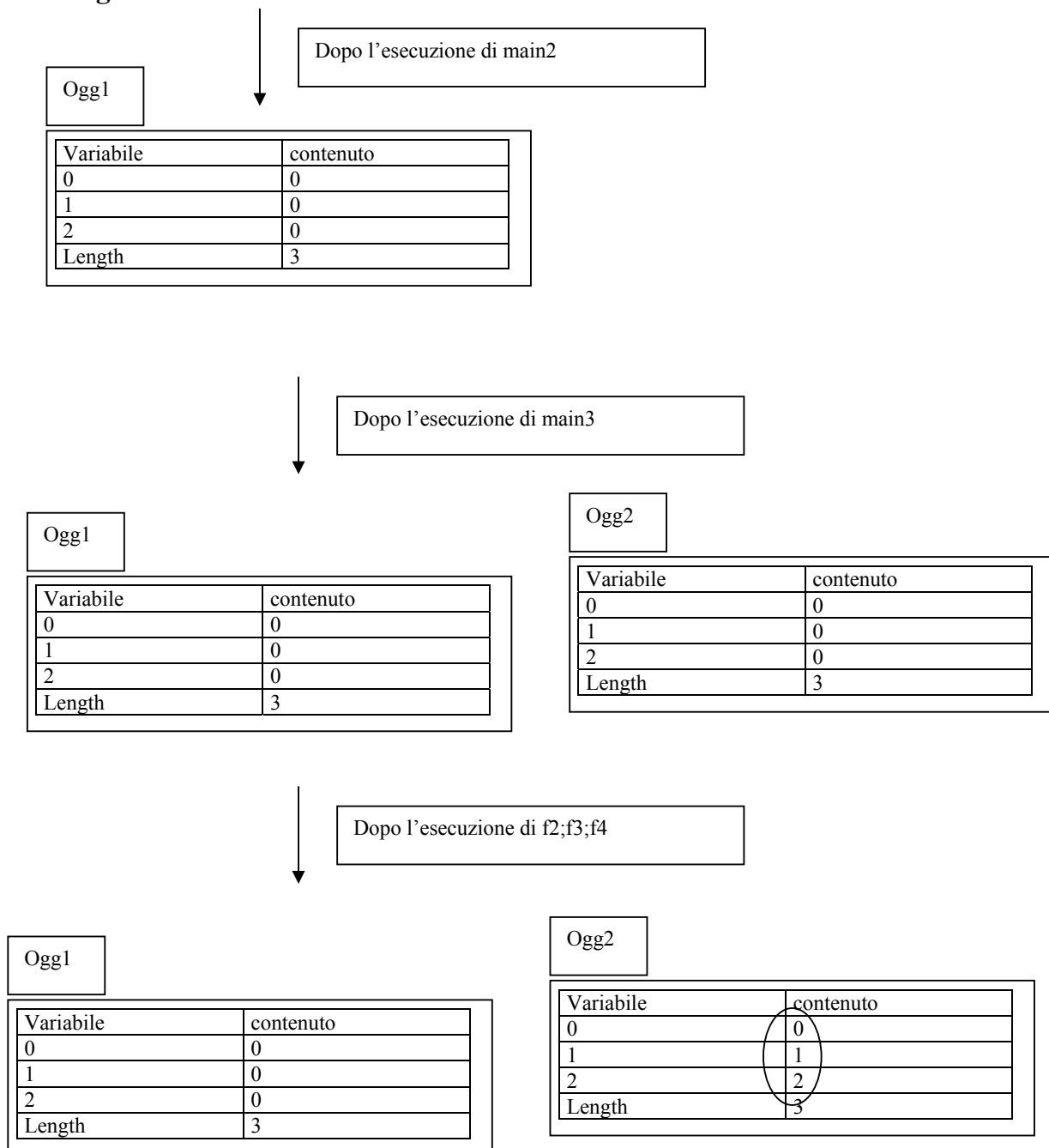
Esecuzione statement f

//int_f: a viene inizializzato con un ref a ogg2
//f1: crea i e gli assegna 3
//f2, f3, f4, modificano le componenti dell'oggetto riferito da a come indicato
// f5 stampa il valore di i
// l'esecuzione torna al main

Area locale f

statement	Parametro – variabile locale	
	a	i
int f	Punta a ogg2	Non creato
f1	Punta a ogg2	3
f2	Punta a ogg2	3
f3	Punta a ogg2	3
f4	Punta a ogg2	3
f5	Punta a ogg2	3

Area globale



OUTPUT

3

5 0 1