

Corso di laurea in Informatica Applicata

Fondamenti di Programmazione

Seconda verifica intermedia

19/12/2002

ESERCIZIO 1

Si consideri lo stack di frames: $\sigma = \sigma_1.\sigma_2.\Omega$ dove σ_1 e σ_2 sono i seguenti frames: $\sigma_1 = \sigma_1[0/x] \sigma_1[1/y]$ $\sigma_2 = \sigma_2[4/y] \sigma_2[10/x, 0/z]$. Indicare i valori $\sigma_1'(x)$, $\sigma_1'(y)$ e $\sigma_1'(z)$ nei seguenti casi:

- a) $\sigma_1' = (\sigma_1[8/z])\sigma_1[100/z]$
- b) $\sigma_1' = \sigma_1[10/z].\sigma_2.\Omega$
- c) $\sigma_1' = \sigma_1[2/y,]. \sigma_2[5/x, 1/z].\Omega$

Soluzione

- a) $\sigma_1'(x)=0, \sigma_1'(y)=1 \sigma_1'(z)=100$
- b) $\sigma_1'(x)=0, \sigma_1'(y)=1 \sigma_1'(z)=10$
- c) $\sigma_1'(x)=0, \sigma_1'(y)=2 \sigma_1'(z)=1$

ESERCIZIO 2

Si supponga di estendere la sintassi dei comandi con il seguente comando:

Com ::= **if** (E) C

Dove l'espressione E è di tipo boolean e il significato informale del comando è che viene valutata l'espressione E, se risulta vera si esegue il comando C, altrimenti si lascia lo stato inalterato. Dare la semantica operativa del nuovo comando, con riferimento al modello in cui lo stato è costituito solo da stack di frames.

Soluzione

$$\begin{array}{l}
 \text{ifthen-tt} \frac{\langle E, \sigma \rangle \text{ exp tt} \quad \langle C, \sigma \rangle \text{ cmd } \sigma'}{\langle \text{if}(E) C, \sigma \rangle \text{ com } \sigma'} \\
 \text{ifthen-ff} \frac{\langle E, \sigma \rangle \text{ exp ff}}{\langle \text{if}(E) C, \sigma \rangle \text{ com } \sigma}
 \end{array}$$

ESERCIZIO 3

Utilizzando le regole definite nell'esercizio 2, si dimostri l'equivalenza dei seguenti comandi C1 e C2:

C1: **if** (x==0) x=x+1;
C2: **while**(x==0) x=x+1;

a partire da un generico stato σ in cui $\sigma(x) \neq 0$ e $\sigma(y) \neq 0$.

Soluzione

Per casi $\sigma(x) = 0$

C1: if-tt $\sigma[[x+1;]] \sigma=1 \langle x=x+1, \sigma \rangle \sigma = \sigma[1/x]$

C2: : while-tt $\langle x=x+1; \text{while}(x==0) x=x+1; , \sigma \rangle \sigma = \text{while}(x==0) x=x+1; \sigma[1/x]$
 $\sigma \text{ while-ff} \sigma[1/x]$

Per casi $\sigma(x) \neq 0$

C1: if-ff σ

C2: while-ff σ

ESERCIZIO 4

Si vuole aggiungere alla classe Arrays vista a lezione, un nuovo metodo statico SommaPos. L'intestazione di tale metodo è:

public static int SommaPos (int [] a)

/** Restituisce la somma degli elementi di a che sono maggiori di 0.

param a: un array di interi

return: la somma degli elementi positivi.

*/

Si definisca il corpo del metodo, in modo che calcoli la somma degli elementi positivi dell'array a argomento del metodo. Ad esempio, se a è l'array rappresentato dalla seguente tabella:

15	0	-5	-96	14	7	-4	-22
----	---	----	-----	----	---	----	-----

la chiamata Arrays.SommaPos(a) restituisce 36.

Soluzione

```
public static int SommaPos (int [ ] a){  
    int s=0;  
    for (int i=0; i<a.length; i++)  
        if (a[i]>0) s=s+a[i];  
    return s;  
}
```

ESERCIZIO 5

Dato il seguente programma:

```
prog {class Primo{
```

```

public int x;
}
class Secondo{
public int y;
public boolean ok;
public void UpdO(Primo o) {
    if (this.ok) o.x=this.y      (5)
}
}
}
{
Primo og1= new Primo();
Secondo og2= new Secondo();    (2)
og1.x=30;
og2.y=10;
og2.ok=true;                   (3)
og2.UpdO(og1)                   (4)
}
}
}

```

rappresentare graficamente:

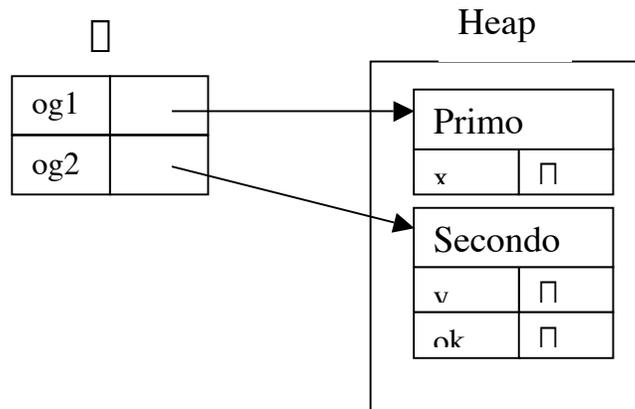
- I. l'ambiente delle classi al punto (1);
- II. lo stack di frames e lo heap dopo l'esecuzione dei comandi (2), (3) e (4),
- III. lo stack di frame e lo heap prima dell'esecuzione del comando (5) (esecuzione del metodo Upd invocato in (4)).
- IV. lo stack di frame e lo heap dopo l'esecuzione del comando (5) (esecuzione del metodo Upd invocato in (4)).

Soluzione

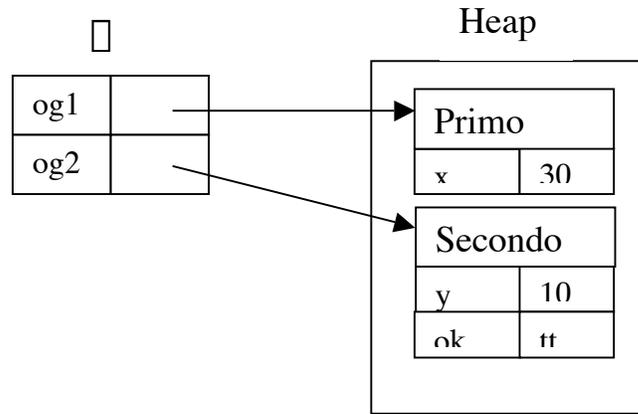
I.

Primo	{(x, □)}	□
Secondo	{(y, □), (ok, □)}	Upd o if (this.ok) {o.x=this.y;}

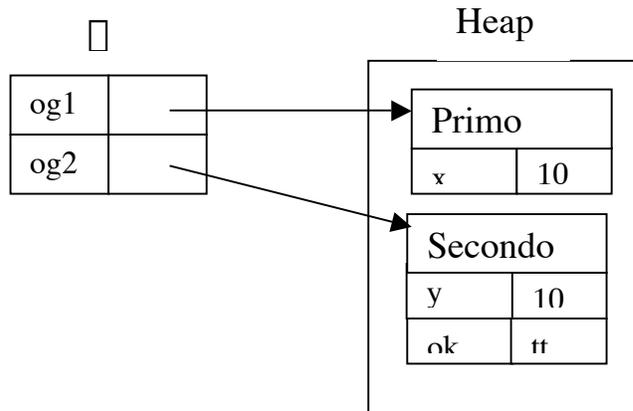
II (2)



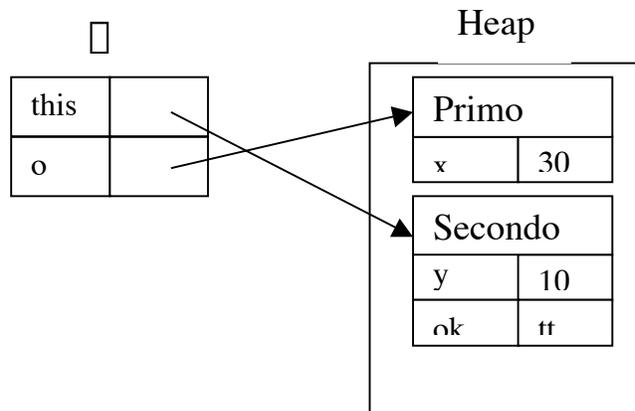
II (3)



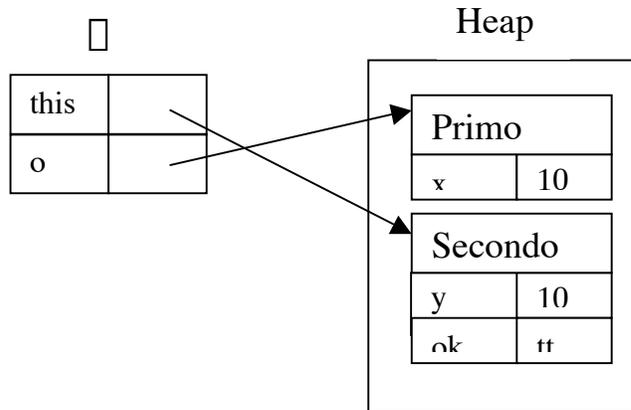
II(4)



III



IV



ESERCIZIO 6

Si dica cosa calcola il seguente metodo, definito per la classe Lista visto a lezione:

```

public int Che? () {
    if (resto==null) return 0;
    else if (val>0) return (val+resto.Che?());
    else return rest.Che?();
}

```

Soluzione

L'invocazione del metodo su L di tipo Lista cioè L.Che.() calcola la somma degli elementi positivi presenti nella lista L.