Esame di LP1

Prof Piero Bonatti

20 Febbraio 2014

Domande generali – Max 6 punti

Esercizio 1: Barrare tutte le frasi vere. **[2 punti]**

a) [ ] Un linguaggio senza alcuna forma di cicli (for, while, do-while) non può essere computazionalmente completo.

b) [X] Nei linguaggi funzionali il valore dei parametri delle funzioni non può essere modificato.

c) [ ] Nel primo Fortran già esistevano costrutti per allocare memoria a run-time.

d) [ ] In una implementazione compilativa pura il compilatore riceve anche l'input del programma sorgente.

Esercizio 2: Dato il codice C qui riportato:

```
int x[10];
x[5] = x[6];
```

barrare l'espressione che rappresenta la parte *sinistra* dell'assegnamento:

[ ] *mem*(x + 5)
[X] *env*(x) + 5
[ ] *mem*(*env*(x)) + 5

e l'espressione che rappresenta la parte *destra* dell'assegnamento:

[ ] *mem*(x + 6)
[ ] *mem*(*env*(x)) + 6
[X] *mem*(*env*(x) + 6) (i vettori denotano l'indirizzo iniziale in cui vengono memorizzati)

[2 punti]

Esercizio 3: Disegnare i data object generati dalle seguenti istruzioni C:

```
int *x, *y;
x = (int*) malloc(sizeof(int));
y = x; *y=3;
```

[2 punti]

Esercizio sul passaggio di parametri – Max 11 punti

Dire qual è l'output del seguente programma nei casi elencati qui sotto:

1. Scoping dinamico, [MODE] = IN OUT per copia
2. Scoping dinamico, [MODE] = IN OUT per riferimento
3. Scoping statico, [MODE] = IN per copia
4. Scoping statico, [MODE] = IN per riferimento

Mostrare gli stack di attivazione (pena la perdita di punti), tranne nei casi di errore, nei quali bisogna invece indicare l'istruzione che causa l'errore.

```
program p1
int a; int b; int c; int d;
     procedure p2([IN OUT x rif] int c,[MODE] int a)
     int b;
     BEGIN
     b=a;
     c=c+2;
     a=1;
     d=a-4;
     p4(d, c);
     write(a,b,c,d);
     END

     procedure p3([IN OUT x rif] int d)
     int a; int b;
     BEGIN
     if d<20 then a=d else a=d+3;
     b=4;
     d=c;
     c=d*3;
     write(a,b,c,d);
     END

     procedure p4([IN x copia] int d,[IN OUT x copia] int a)
     int b;
     BEGIN
     b=a+4;
     d=a;
     if b<0 then a=4 else a=1;
     c=4;
     p3(c);
     write(a,b,c,d);
     END

BEGIN
a=1;
b=3;
c=1;
d=3;
p2(c, d);
write(a,b,c,d);
END
```

UML – Max 15 punti

Si vuole progettare un programma di grafica a oggetti che definisce figure piane. Le figure piane hanno un colore e possono essere ellissi, rettangoli, o linee spezzate. In particolare, i rettangoli sono descritti dalle coordinate dell'angolo in basso a sinistra e dalla lunghezza dei due lati. Le figure piane possono essere visualizzate; per ogni tipo specifico di figura si adotta un algoritmo ad hoc. Non si possono creare figure piane generiche, ma solo ellissi, rettangoli e linee spezzate. Le figure piane utilizzano un componente "display" di cui si sa solamente che supporta due metodi: *set_pixel*(x:int,y:int,color:int):void e *clear*():void. Le figure costruite dall'utente vengono mantenute in una lista concatenata. La lista di figure piane permette di visualizzare tutti i suoi membri.

Esercizio 1: Disegnare un diagramma delle classi per queste specifiche.
[max 8 punti]

Esercizio 2: Disegnare un diagramma delle sequenze per lo scenario seguente, mostrando tutte le interazioni tra gli oggetti coinvolti.

Un utente invoca il metodo di visualizzazione della lista di figure, che visualizza i due membri che in quel momento sono presenti nella lista nell'ordine in cui vi compaiono (così in caso di sovrapposizione le prime sono visualizzate sotto le successive). La visualizzazione avviene in modo ricorsivo.

[max 7 punti]