

Esame di LP1

Prof Piero Bonatti

5 Marzo 2014

Domande generali – Max 6 punti

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

- a) La classica implementazione del predicato *member* nel paradigma logico può essere utilizzata sia per verificare se un dato elemento appartiene a una lista che come “generatore”, cioè per enumerare tutti i membri della lista uno a uno.
- b) Il LISP è un linguaggio imperativo.
- c) In una implementazione interpretativa pura l'interprete produce codice oggetto a partire dal sorgente.
- d) Se implementiamo le liste mediante template, i membri di una lista hanno tutti lo stesso tipo (o nel caso object oriented, appartengono a qualche sottoclasse di una singola classe).

Esercizio 2: Dato il codice C qui riportato:

```
int x, *y;  
y = &x;
```

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

- mem(y)*
- env(y)*
- mem(env(y))*

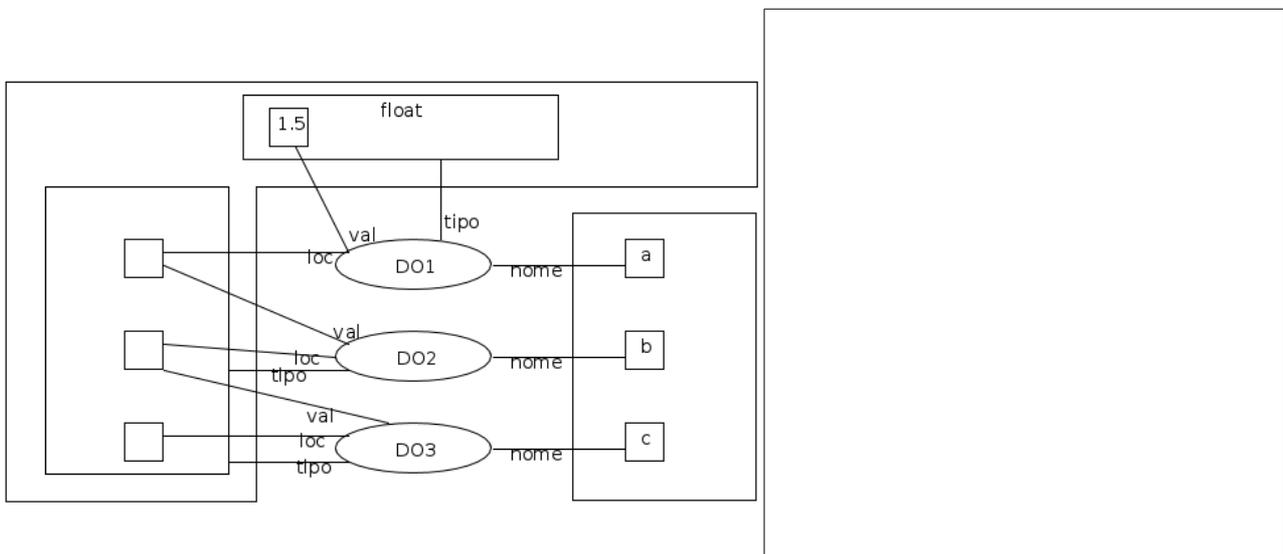
e l'espressione che rappresenta la parte *destra* dell'assegnamento:

- mem(x)*
- env(x)*
- mem(env(x))*

[2 punti]

Esercizio 3: Scrivere nel riquadro istruzioni C che generino i seguenti data objects:

[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 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 b)
    int d; int c;
    BEGIN
    d=2;
    c=1;
    b=d;
    a=c;
    p3(d, c);
    write(a,b,c,d);
    END

  procedure p3([MODE] int b,[IN x copia] int a)
    int d;
    BEGIN
    d=3;
    b=a;
    a=d;
    c=a+1;
    p4(d, b);
    write(a,b,c,d);
    END

  procedure p4([IN OUT x rif] int b,[IN OUT x rif] int d)
    int a;
    BEGIN
    if b>1 then a=c else a=4;
    b=c;
    d=a;
    c=a-2;
    write(a,b,c,d);
    END

BEGIN
a=3;
b=3;
c=3;
d=4;
p2(b);
write(a,b,c,d);
END
```

UML – Max 15 punti

Si vuole progettare un gioco che è costituito da personaggi e livelli.

I personaggi hanno un nome e una immagine che li rappresenta (le immagini sono entità complesse che possono essere visualizzate). Vi sono due tipi di personaggi: protagonisti e nemici. I personaggi possono essere visualizzati e possono agire; *i protagonisti agiscono in modo diverso dai nemici*.

I livelli sono 10. Ogni livello è raffigurato da una immagine e possiede un insieme di personaggi attivi. I livelli permettono di eseguire un turno, cioè di far agire tutti i propri personaggi attivi.

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.

Il gioco chiede a un livello di eseguire un turno. Il livello chiede all'insieme di personaggi attivi un primo elemento, e riceve il protagonista. Il livello fa agire il protagonista. Poi il livello chiede all'insieme di personaggi il prossimo elemento. Riceve null, per cui il turno termina.

[max 7 punti]