

Esame di LP1

Prof Piero Bonatti

12 Settembre 2013

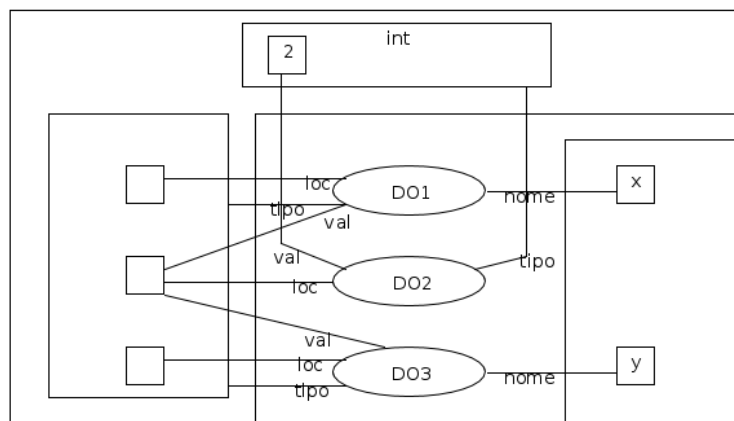
Domande generali – Max 6 punti

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

- a) In un linguaggio fortemente tipato, a tempo di compilazione viene associato un tipo specifico ad ogni espressione del programma.
- b) SQL è un linguaggio *general purpose*.
- c) In un linguaggio dinamicamente tipato una stessa variabile può cambiare il suo tipo a tempo di esecuzione.
- d) Per la rappresentazione di strutture dati (ad es. liste, code ed alberi) di tipo *omogeneo* il polimorfismo per inclusione è più adatto del polimorfismo parametrico.

Esercizio 2: Scrivere una espressione C che denoti $mem(env(v)+1)$. [2 punti]

Esercizio 3: Scrivere delle istruzioni C che portino a costruire i seguenti data object: [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 per riferimento
3. Scoping statico, [MODE] = IN per copia
4. Scoping statico, [MODE] = IN OUT 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 q; int r; int s; int t;
  procedure p2([IN OUT x rif] int q)
    int t;
      procedure p3([MODE] int r,[IN OUT x rif] int t)
        int q;
        BEGIN
          q=3;
          r=s+2;
          t=1;
          s=t;
          p4(r, q);
          write(q, r, s, t);
        END

      procedure p4([IN x copia] int s,[IN OUT x copia] int q)
        int t;
        BEGIN
          t=s-4;
          s=t;
          q=4;
          r=1;
          write(q, r, s, t);
        END

    BEGIN
      t=r;
      q=q;
      if s>1 then s=t*1 else s=t;
      r=1;
      p3(s, r);
      write(q, r, s, t);
    END

  BEGIN
    q=2;
    r=3;
    s=0;
    t=4;
    p2(t);
    write(q, r, s, t);
  END
```

UML – Max 15 punti

Si vuole progettare una applicazione per la gestione di ristoranti che possa trattare piu' di un ristorante (il che è utile per le catene di ristorazione). Ogni ristorante ha un certo numero di tavoli (almeno uno) e gestisce delle prenotazioni. Ogni tavolo possiede un suo numero e una capienza (cioè il numero di posti). Le prenotazioni contengono giorno, ora, uno o più tavoli, e il cliente che ha prenotato. I clienti hanno nome e numero di telefono, e i ristoranti possono reperire velocemente i clienti usando il loro nome come chiave. Infine ogni ristorante ha una sua lista di attesa (che rispetta l'ordine di arrivo) i cui elementi contengono un cliente e il numero di posti richiesto; un cliente può comparire al massimo in una sola lista di attesa. Si possono inserire clienti in coda alle liste di attesa (specificando anche il numero di posti richiesti); in aggiunta, specificando il numero di posti che si sono liberati, si può cercare nella lista il primo cliente (nell'ordine di arrivo) che ha richiesto quel numero di posti.

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. Modellare la ricerca nella lista in stile ricorsivo.

L'operatore chiede al ristorante R1 di cercare nella propria lista d'attesa il primo cliente che ha chiesto un tavolo da 2 posti. Il primo elemento nella lista richiede 4 posti, quindi si passa la richiesta al secondo elemento della lista, che contiene una richiesta di due posti. Il cliente corrispondente (C2) viene restituito all'operatore.

[max 7 punti]