

Esame di LP1

Prof Piero Bonatti

10 Settembre 2015

Domande generali – Max 6 punti

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

1. Il primo Fortran supporta la ricorsione []
2. Il LISP supporta la ricorsione []
3. Java ha una implementazione: compilata [] interpretata [] mista []
4. Esistono linguaggi a oggetti senza gerarchia di tipi []

Esercizio 2: [2 punti] Nell'assegnamento $x = *(y + 2)$ dei linguaggi C e C++, l'espressione a destra denota:

- a) $env(y) + 2$ []
- b) $mem(env(y)) + 2$ []
- c) $mem(mem(env(y)) + 2)$ []
- d) $mem(mem(env(y))) + 2$ []

Esercizio 3: [2 punti] Disegnare i data object relativi al seguente codice C:

```
int x = 1, y = 2;  
int * z = &x;  
*z = y;
```

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 OUT 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 x; int y; int z; int t;
  procedure p2([MODE] int z)
    int y; int t;
    procedure p3([IN OUT x copia] int y,[IN x copia] int t)
      int x;
      procedure p4([IN OUT x copia] int z)
        int y; int t;
        BEGIN
          y=z-2;
          t=x;
          z=x*2;
          x=2;
          write(x,y,z,t);
        END

      BEGIN
        x=z+2;
        y=2;
        if y=7 then t=4 else t=t*1;
        z=2;
        p4(t);
        write(x,y,z,t);
      END

    BEGIN
      y=1;
      t=z;
      x=z*2;
      p3(t, y);
      write(x,y,z,t);
    END

  BEGIN
    x=4;
    y=4;
    z=2;
    t=4;
    p2(x);
    write(x,y,z,t);
  END
```

UML – Max 15 punti

Si vuole progettare un sistema di avvisi per i monitor di un aeroporto, Comprende dei voli (caratterizzati da un numero di volo, una destinazione, un orario di partenza e uno di imbarco), degli sportelli di check-in (caratterizzati da un numero identificativo e dal numero di terminale in cui si trovano) e dei *gate* (caratterizzati da un numero identificativo e dal terminale in cui si trovano). Inoltre, ogni volo è associato al massimo ad un gate (potrebbe non essere ancora noto) e ad un insieme di sportelli in cui si può fare il check-in. Le schermate informative sono di due soli tipi: *check-in* e *imbarco*. Ogni schermata informativa ha un suo messaggio di intestazione e una lista ordinata di voli, da cui si può eliminare il primo elemento e aggiungerne uno in coda. L'operazione di *refresh*, che aggiorna le schermate, mostra informazioni diverse nelle due tipologie di schermata. Il refresh utilizza un componente esterno – di cui si sa solo che supporta un metodo *get_ads()* – per ricevere le pubblicità da mostrare vicino alle informazioni sui voli.

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. **Se un oggetto ha bisogno di un attributo privato di un altro oggetto dovete mostrare come se lo procura.**

Un operatore chiede a una schermata di aggiungere il volo AX123 alla lista, che al momento contiene 2 voli. Poi l'operatore fa il refresh della schermata (operazione che causa lo scaricamento delle pubblicità).

[max 7 punti]