

# Esame di LP1

Prof Piero Bonatti

19 Giugno 2014

## Domande generali – Max 6 punti

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

- a)  Nel paradigma logico non vi è una distinzione netta tra parametri di input e di output.
- b)  SQL è un linguaggio *general purpose*.
- c)  In una implementazione compilativa pura, compilatore e codice oggetto devono essere eseguiti dalla stessa macchina.
- d)  Il polimorfismo per inclusione è il più indicato per la definizione di collezioni di oggetti omogenei.

**Esercizio 2:** In C e C++, l'assegnazione le cui parti sinistra e destra denotano  $env(x)+1$  e  $mem(env(y))+1$ , rispettivamente, è:

- a)  $x[1] = *(y+1);$
- b)  $*(x+1) = y+1;$
- c)  $x[1] = y+1;$
- d)  $x[1] = y[1];$

[2 punti]

**Esercizio 3:** Disegnare nel riquadro lo schema di implementazione misto, parzialmente compilato e parzialmente interpretato, come quello adottato da Java.

[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 statico, [MODE] = IN per copia
3. Scoping statico, [MODE] = IN OUT per riferimento
4. Scoping statico, [MODE] = 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 a; int b; int c; int d;
  procedure p2([IN OUT x copia] int a)
    int c; int d;
    BEGIN
    if b>3 then c=b else c=a+2;
    d=4;
    a=c-4;
    if a>10 then b=c else b=d-3;
    write(a,b,c,d);
    END

  procedure p3([MODE] int b,[IN x copia] int c)
    int d;
    procedure p4([IN x copia] int d)
      int a; int c;
      BEGIN
      a=b*4;
      c=3;
      d=b-4;
      b=d;
      p2(b);
      write(a,b,c,d);
      END

    BEGIN
    d=a;
    b=b+2;
    c=a*3;
    a=1;
    p4(b);
    write(a,b,c,d);
    END

BEGIN
a=0;
b=3;
c=0;
d=2;
p3(d, c);
write(a,b,c,d);
END
```

## UML – Max 15 punti

Si vuole progettare una applicazione gestionale per una azienda, che possiede degli impiegati caratterizzati da nome e stipendio. Gli impiegati possono essere tecnici, amministrativi, o manager. Non si possono creare impiegati che non appartengano a una di queste tre categorie. I manager hanno un bonus; il loro stipendio totale è calcolato sommando l'attributo stipendio con il bonus. Per gli altri tipi di impiegato lo stipendio coincide con l'attributo omonimo. Un *rapporto impiegati* è una lista ordinata di impiegati. Un rapporto impiegati è in grado di calcolare la somma degli stipendi totali di tutti i dipendenti che contiene.

**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 chiede di calcolare il totale degli stipendi dei dipendenti elencati in un rapporto impiegati che contiene due elementi: un tecnico con stipendio 1500, e un manager con stipendio 2000 e bonus 500. Il calcolo avviene in modo *ricorsivo* sulla lista di impiegati.

**[max 7 punti]**