

Esame di LP1

Prof Piero Bonatti

17 Luglio 2014

Domande generali – Max 6 punti

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

- a) Nei linguaggi funzionali l'ambiente associa direttamente gli identificatori al loro valore.
- b) Nei linguaggi imperativi l'ambiente associa direttamente gli identificatori al loro valore.
- c) HTML è un linguaggio *general purpose*.
- d) Il codice oggetto deve essere eseguito da un interprete diverso dalla macchina hardware.

Esercizio 2: Data le dichiarazioni C: `typedef int T; int x,y; T z;` dire quali dei seguenti assegnamenti rispettano la name equivalence e quali la structural equivalence, barrando gli spazi appositi (notare che le due opzioni non sono mutuamente esclusive):

- a) `x=y;` name equivalence structural equivalence
- b) `z=x;` name equivalence structural equivalence
- c) `z=z+1;` name equivalence structural equivalence

[2 punti]

Esercizio 3: Disegnare i data object generati dal seguente codice C:

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

[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 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 q; int r; int s; int t;
  procedure p2([MODE] int r)
    int t;
    BEGIN
    t=1;
    if q>4 then r=r+2 else r=t*1;
    q=r;
    if t>6 then s=1 else s=3;
    write(q,r,s,t);
    END

  procedure p3([IN OUT x copia] int t,[IN OUT x rif] int q)
    int r;
    procedure p4([IN OUT x rif] int s,[IN OUT x rif] int t)
      int r;
      BEGIN
      r=2;
      s=3;
      t=q;
      if t<5 then q=t*1 else q=2;
      p2(r);
      write(q,r,s,t);
      END

    BEGIN
    r=s-4;
    t=s-1;
    q=4;
    s=r;
    p4(r, s);
    write(q,r,s,t);
    END

BEGIN
q=3;
r=0;
s=1;
t=2;
p3(t, q);
write(q,r,s,t);
END
```

UML – Max 15 punti

Progettare una applicazione per la gestione di utenze telefoniche con le seguenti caratteristiche. Ogni utente ha un nome e un credito. Inoltre è associato a due liste concatenate di telefonate, quelle in ingresso e quelle in uscita. Tutte le telefonate hanno una durata. In aggiunta, quelle in ingresso hanno il numero chiamante, mentre quelle in uscita hanno il numero chiamato e un costo. Non si possono creare telefonate generiche (ma solo in ingresso o in uscita). Ogni utente è associato a un contratto che specifica le date di inizio e fine del contratto stesso, e il costo per minuto delle chiamate.

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 aggiunge in coda alla lista di telefonate in uscita una nuova chiamata al numero 333-456789 della durata di 50 secondi. Prima dell'aggiunta la lista contiene già due telefonate. L'aggiunta deve essere eseguita in modo *ricorsivo*.

[max 7 punti]