Esame di LP1

Prof Piero Bonatti

8 Aprile 2015

Domande generali – Max 6 punti

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

- [ ] Il primo Fortran supportava la ricorsione.
- [ ] Se il linguaggio è dinamicamente tipato, allora il tipo di una variabile può cambiare durante l'esecuzione del programma.
- [ ] C adotta la structural equivalence per le `struct` e la name equivalence per gli altri tipi.
- [ ] Si può accedere alle variabili non locali di una procedura in tempo costante, indipendentemente da quanti record di attivazione si devono attraversare.

Esercizio 2: **[2 punti]** Considerate l'implementazione efficiente dell'ambiente non locale con scoping statico. Dato il codice Pascal-like qui sotto, indicare negli appositi spazi (sulla destra) con quali coppie livello-offset vengono rappresentate le variabili `x` e `y` nella procedura `r`.

```
program p
int a; int x;
     procedure q
      int x; int y;
           procedure r
           BEGIN ... END
      BEGIN ... END
BEGIN ... END
```

x → (___ , ___)

y → (___ , ___)

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

```
float f, *g, **h;
h = &g;
g = &f;
**h = 5;
```

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 OUT per riferimento
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 a; int b; int c; int d;
     procedure p2([IN OUT x copia] int b,[MODE] int a)
     int d;
     BEGIN
     d=1;
     b=d;
     a=2;
     c=1;
     p4(a);
     write(a,b,c,d);
     END

     procedure p3([IN OUT x copia] int d)
     int a;
     BEGIN
     a=3;
     d=a-2;
     b=2;
     c=4;
     write(a,b,c,d);
     END

     procedure p4([IN x copia] int d)
     int a; int c;
     BEGIN
     a=3;
     c=3;
     if b<9 then d=b+3 else d=a+3;
     b=b-2;
     p3(d);
     write(a,b,c,d);
     END

BEGIN
a=1;
b=2;
c=3;
d=0;
p2(d, c);
write(a,b,c,d);
END
```

UML – Max 15 punti

Progettare un sistema di prenotazione viaggi con le seguenti caratteristiche. Un viaggio è una sequenza ordinata di tratte. Ogni tratta possiede un contratto, rappresentato con una stringa, e un biglietto. I biglietti possono essere aerei o ferroviari. Ogni biglietto ha un luogo e una data di partenza e luogo e data di arrivo. I biglietti aerei hanno anche un'ora di imbarco, mentre i biglietti ferroviari hanno un codice di sconto rappresentato con una stringa. Ai viaggi si possono aggiungere tratte in coda.

Esercizio 1: Disegnare un diagramma delle classi per queste specifiche.
[max 7 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 utente crea una nuova tratta via aerea (che a sua volta crea il proprio biglietto), con contratto standard CS, partenza da Napoli il 15/4/2015 e arrivo a Venezia il giorno stesso. Poi appende la nuova tratta a un viaggio che già contiene una tratta T0.

[max 8 punti]