

Esame di LP1

Prof Piero Bonatti

23 Gennaio 2014

Domande generali – Max 6 punti

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

- a) Una lista di oggetti i cui tipi non sono confrontabili tra loro si può definire con il polimorfismo parametrico (*templates*).
- b) HTML è un linguaggio *general purpose*.
- c) Nelle soluzioni puramente interpretate, l'interprete di un linguaggio L riceve in input sia un programma P scritto in L che gli input di P.
- d) Java è dinamicamente tipato.
- e) UML può essere usato per descrivere un software preesistente.

Esercizio 2: Dato il codice C qui riportato:

```
int * x;  
*(x+1) = 5;
```

barrare l'espressione che rappresenta la parte sinistra dell'assegnamento:

- mem(x + 1)*
- mem(env(x)) + 1*
- mem(mem(env(x)) + 1)*
- env(mem(env(x)) + 1)*

[2 punti]

Esercizio 3: Disegnare i data object generati dalle seguenti istruzioni C:

```
int **x;  
int *y;  
y = malloc(sizeof(int));  
x = &y;
```

[2 punti]

vedere diagramma nel file .png

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 a; int b; int c; int d;
  procedure p2([IN OUT x rif] int c,[IN OUT x rif] int b)
    int d;
    BEGIN
    if c<4 then d=2 else d=c+2;
    c=1;
    b=4;
    a=2;
    p3(c);
    write(a,b,c,d);
    END

  procedure p3([IN OUT x copia] int a)
    int d; int c;
    procedure p4([MODE] int b)
      int c; int a;
      BEGIN
      c=b;
      a=1;
      b=a;
      d=3;
      write(a,b,c,d);
      END

    BEGIN
    d=a+2;
    c=a;
    a=2;
    b=4;
    p4(b);
    write(a,b,c,d);
    END

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

vedere correttori nell'apposita directory

UML – Max 15 punti

Si vuole progettare un sito per organizzare viaggi, prenotando sia il trasporto che il pernottamento. Il sito gestisce clienti (caratterizzati da nome ed indirizzo mail), ciascuno dei quali possiede un proprio “carrello della spesa”, il cui contenuto *deve essere mantenuto ordinato rispetto all'ordine di inserimento nel carrello*. Il carrello contiene ordini, che possono essere biglietti di viaggio (con data e ora di partenza) oppure prenotazioni alberghiere (con il nome dell'hotel, la data di arrivo e numero di notti). I biglietti di viaggio a loro volta possono essere biglietti aerei (con il nome della compagnia aerea) o ferroviari (con i nomi delle stazioni di partenza e arrivo). Per prenotare un albergo, il sito usa *un componente che verrà sviluppato da altri* di cui si sa *solo* che supporta l'operazione *prenota*(NomeHotel,DataArrivo,NumNotti), che restituisce un booleano che è *true* se e solo se la prenotazione ha successo. *Non si possono creare ordini o biglietti generici*: bisogna per forza specificarne il tipo (cioè bisogna specificare se si tratta di biglietti aerei, ferroviari o prenotazioni alberghiere, nel caso degli ordini, e se si tratta di biglietti aerei o ferroviari nel caso dei biglietti).

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 oggetto cliente C1 crea un nuovo biglietto aereo B1 e lo inserisce in coda al proprio carrello con una procedura ricorsiva. Il carrello prima dell'inserzione contiene già due ordini.

[max 7 punti]

vedere diagrammi nell'apposita directory