

Ingegneria del Software B

Allievi della Laurea Specialistica in Ingegneria Informatica
Tema d'esame – 26 Novembre 2009 – ore 9.00-10.30

NOME: **COGNOME:**
MATRICOLA: **FIRMA:**

Le risposte chiuse valgono 1/30 ciascuna. Il valore degli esercizi è riportato nel prospetto a lato.

| Esercizio | 1 | 2a | 2b | 2c | 3 | 4 | 5a | 5b | 5c |
|-------------|---|----|----|----|---|---|-----|-----|----|
| Valore | 2 | 1 | 2 | 2 | 2 | 3 | 1,5 | 1,5 | 3 |
| Valutazione | | | | | | | | | |

Risposte chiuse

| Affermazione | Vera o falsa? |
|--|---------------|
| La relazione di dipendenza fra package è transitiva | |
| Il linguaggio Java supporta l'ereditarietà multipla di una classe da più <i>interface</i> | |
| Nell'esecuzione di un programma orientato agli oggetti che rispetta il principio di sostituzione di Liskov non si possono verificare up-call | |
| Push e Pull sono due modalità di attuazione del pattern Composite | |
| Il design pattern State è classificato come Object Behavioral (cioè comportamentale e basato su oggetti) | |
| FEP(t_i), il Fault Exposing Potential di un singolo caso di test t_i relativamente a un programma, è una misura compresa fra 0 e 1 | |
| L'impalcatura per il testing (stub, driver e oracle) non è necessaria per il testing in grande | |
| Le ispezioni del codice sono effettuate conclusa l'attività di testing | |
| I grafi di tracciabilità sono sfruttati per valutare l'impatto di un cambiamento | |
| La checklist per l'ispezione del codice può comprendere anche controlli di aderenza agli standard aziendali | |
| Il modello di Dromey assume che i valori delle qualità esterne di prodotto dipendano dai valori delle qualità interne | |
| ISO 9126 è un modello di qualità di processo | |

Esercizi

- 1) Chiarire il concetto di sistema ereditato (*legacy system*).
- 2) Sia dato il design pattern Observer.
 - a) Classificare tale pattern e descriverne l'intento.
 - b) Descrivere la sua struttura.
 - c) Derivare un diagramma degli oggetti, contenente almeno due istanze di ogni classe, compatibile con la struttura di cui al punto precedente.
- 3) Introdurre il concetto di testing di regressione.

4) Creare un insieme di test funzionali di unità per un modulo la cui specifica è la seguente:

acquisiti in ingresso dei valori numerici, ordina gli stessi in ordine non decrescente.

Si assuma che valgano le seguenti precondizioni:

- I valori numerici di ingresso sono scritti, ciascuno su una linea distinta, in un file di testo privo di linee vuote contenente al più 1000 linee.
- Ogni valore numerico è descritto da una sequenza finita di cifre che può essere preceduta da un segno e/o spezzata da un punto radice (cioè tale punto non è mai l'ultimo carattere della rappresentazione di un valore).
- Ogni valore numerico appartiene al dominio dei valori reali rappresentabili sulla piattaforma di destinazione.

5) Sia dato il seguente programma.

```
main(){
1   scanf("%d", &a);
2   scanf("%d", &b);
3   scanf("%d", &c);
4   if (b == a) and (b > c)
5       while (c < 0)
6           c++;
7   else a = c;
8   if (b == c) {
9       a = b - a;
10      c = b*4;
11      if (a == 4)
12          a--;
13  }
14  else b--;
15  printf("%d\n", a);
16  printf("%d\n", b);
17  printf("%d\n", c);
18  }
```

- Tracciarne il CFG (Control Flow Graph).
- Determinarne l'espressione dei cammini.
- Stabilire, mediante esecuzione simbolica, se il cammino $\langle 1,2,3,4,5,6,5,8,9,10,11,14,15,16 \rangle$ sia percorribile (*feasible*).