Pre-Test per la teoria di Sviluppo Applicazioni SW

Appello del 25/6/2017 - Data pre-test: 25/6/2017

NOME: MATRICOLA:

Il punteggio del pre-test arriva fino a 32. Nella media finale un voto superiore a 30 viene conteggiato come un 30 e lode.

Nota per gli studenti degli A.A. precedenti: chi aveva in precedenza sostenuto la parte di teoria con la prof. Bono deve rispondere solo alle domande delle sezioni B1 e B2. Chi aveva in precedenza sostenuto la parte di teoria con i proff. Picardi e Torta deve rispondere solo alle domande della sezione A.

Sezione A: UML, UP e Design pattern.

Domande a risposta chiusa

A-1: Quando si fa il System Sequence Diagram (SSD)? **[2 punti]**	
☐	Quando si comincia a scrivere il codice.
☐	All'inizio dell'analisi dei requisiti.
☐	Dopo i casi d'uso dettagliati.
☐	Non esiste questo diagramma in UP.

A-2: Cos'è il modello di dominio? **[2 punti]**	
☐	È una rappresentazione implementativa del sistema.
☐	È un altro nome per indicare il modello di progetto.
☐	È una rappresentazione concettuale del sistema.
☐	Serve a rappresentare i casi d'uso in modo grafico.

A-3: Quante iterazioni ha la fase di ideazione? **[2 punti]**	
☐	Quante ne servono.
☐	Normalmente una.
☐	Dipende dalla percentuale dei casi d'uso che si intende implementare.
☐	Non ha iterazioni.

A-4: Dire quale NON è un pattern GoF. **[2 punti]**

- ☐ Abstract Factory
- ☐ Proxy
- ☐ Factory
- ☐ Composite

Domande a risposta aperta

A-A: Disegnare il pattern **Decorator** e scrivere il codice delle sue classi caratterizzanti, inclusi i costruttori. (**NOTA**: basta scrivere del codice schematico, che contenga però gli elementi essenziali.) **[6 punti]**

NOME:

MATRICOLA:

Sezione B2: Testing

Domande a risposta chiusa

B2-1: Qual è il principale svantaggio della **strategia top-down per l'integration testing**, che la strategia bottom-up invece non presenta? **[2 punti]**

- ☐ Poiché combina subito insieme tutte le unità, rende difficile capire la causa dei problemi rilevati.
- ☐ Non può essere applicata in caso di progettazione orientata agli oggetti.
- ☐ Richiede di scrivere molti stub
- ☐ Testa per ultime le componenti più complesse e quindi più soggette ad errori, ritardandone la scoperta.

B2-2: Quando si parla di **regression testing o test di regressione**? **[2 punti]**

- ☐ Quando si effettuano dei test frequenti e non esaustivi relativi alle funzionalità principali, per decidere se il prodotto è sufficientemente stabile per essere testato da un team specializzato.
- ☐ Quando si testano i componenti o le unità in ordine inverso rispetto a quando sono stati realizzati.
- ☐ Quando si scrivono i test prima di avere implementato il codice.
- ☐ Quando, a seguito della correzione di un bug o dell'integrazione di un nuovo componente, si vanno a ripetere i test già effettuati per verificare che le modifiche apportate non abbiano introdotto dei problemi nuovi o non rilevati.

B2-3: Quale delle seguenti <u>non</u> è una tecnica di **black-box testing**? **[2 punti]**

- ☐ Testing casuale
- ☐ Calcolo dei cammini indipendenti
- ☐ Boundary value analysis o analisi dei valori limite
- ☐ Array ortogonali

B2-4: Quale delle seguenti affermazioni sul **validation testing** è **falsa**? **[2 punti]**	
☐	Il validation testing verifica che il software realizzato corrisponda alle specifiche iniziali/requisiti
☐	Alcune fasi del validation testing vengono eseguite direttamente dall'utente finale
☐	Documenti quali i Casi d'Uso dettagliati o i Contratti delle Operazioni costituiscono un input rilevante per il validation testing
☐	Il validation testing può essere effettuato solo quando il software è stato integralmente sviluppato

Domanda a risposta aperta

B2-A: Esporre sinteticamente come si calcola la **complessità ciclomatica** di una porzione di codice, cosa esprime, e in che modo questo è rilevante per il testing. **[4 punti]**