

Ingegneria del Software a.a. 2011-12

Prova Scritta del 4 febbraio 2013

Esercizio di sbarramento

COGNOME

NOME

MATRICOLA

Rispondere alle seguenti domande. Per ogni domanda, solo una soluzione è corretta. L'esercizio si ritiene superato se si risponde correttamente ad **almeno 6 domande**, la valutazione è di 1 punto per ogni risposta corretta oltre le 6.

Domanda 1

Quali tra questi fattori è considerato più difficile da controllare nello sviluppo di un sistema software di "grosse" dimensioni (cioè gruppi di lavoro composti da molte persone, tempi di sviluppo > 6 mesi e software costituito da milioni di LOCs)?

- a) Complessità intrinseca del problema che si deve affrontare
- b) Rapida evoluzione e difficoltà nella scelta di linguaggi di programmazione, frameworks da utilizzare e tools per lo sviluppo (es. IDE)
- c) Gestione del personale (es. creare gruppi di lavoro e sostituzione del personale) e allocazione delle risorse
- d) Gestione dei costi e del bilancio

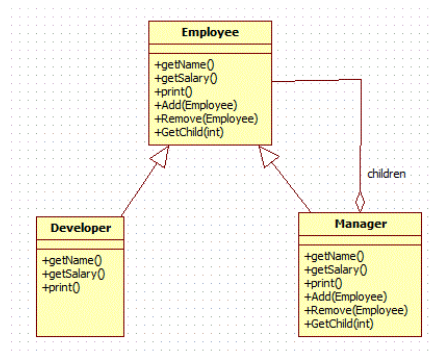
Domanda 2

Per quale motivo i requisiti vengono prioritizzati?

- a) Per decidere cosa includere in un incremento; se stiamo utilizzando metodi di sviluppo software incrementali
- b) Per stabilirne la fattibilità e la correttezza (es. per identificare i "missing requirements")
- c) Per decidere come suddividere il lavoro di sviluppo tra i vari componenti del gruppo; se stiamo utilizzando una metodologia agile
- d) Per decidere quali funzionalità della nostra applicazione testare per prime

Domanda 3

La seguente Figura rappresenta un'istanza di un design pattern. Quale?



- a) Adapter
- b) Visitor
- c) Facade
- d) Composite

Domanda 4

Supponendo di dover testare nel dettaglio una componente software (es. classe) molto complessa (cioè con molti cicli e molte istruzioni condizionali) di un applicazione per smartphone che è stata sviluppata in un linguaggio che conosci molto bene (es. Java) quale tipo di approccio di testing sceglieresti?

- a) Black box testing usando equivalence partitioning
- b) White box testing con criterio branch coverage
- c) Testing di regressione
- d) White box testing con criterio path coverage completo (cioè senza trattare i loop in modo binario)

Domanda 5

Quale tra i vari modelli di sviluppo è stato il primo ad introdurre il concetto di rischio nello sviluppo software?

- a) Modello a cascata
- b) Unified process (UP)
- c) V-model
- d) Modello a spirale

Domanda 6

Qual è lo scopo degli oggetti DAO o layer di persistenza?

- a) Sostituire l'utilizzo di connettori a basi di dati quali ad esempio JDBC
- b) Incapsulare in un livello software la comunicazione tra l'applicazione e il DBMS
- c) Disaccoppiare la "presentation logic" dalla "business logic"
- d) Generare automaticamente il codice di accesso ai dati a partire dal relativo diagramma ER

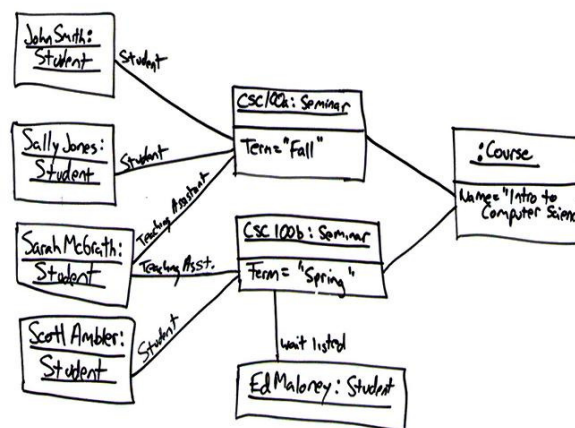
Domanda 7

Nel contesto del design architetturale di un sistema software quale delle seguenti frasi è falsa?

- a) Le componenti "large-grain" migliorano la performance ma riducono la maintainability
- b) Dati ridondanti migliorano la availability ma peggiorano la safety
- c) Usare un'architettura a livelli aumenta le comunicazioni tra componenti e così degrada le prestazioni
- d) Usare un'architettura a livelli con le risorse critiche nei livelli più interni peggiora la safety

Domanda 8

Che tipo di diagramma UML è rappresentato nella Figura?



- a) State diagram
- b) Class diagram
- c) Object diagram
- d) Non è un diagramma UML

Domanda 9

Che cosa è un code smell?

- a) Un "indicatore" che qualcosa nel codice non va bene. Lo sviluppatore deciderà a posteriori se intervenire o no
- b) Una "certezza" che qualcosa nel codice non va bene. Lo sviluppatore dovrà intervenire per sistemare il problema
- c) Un sinonimo di "code clone"
- d) Un errore che necessariamente si manifesta durante l'esecuzione del software

Domanda 10

Come può essere catalogata un'operazione di refactoring di un sistema software che ha l'obiettivo di rendere più coesi (e quindi più mantenibili) alcuni moduli?

- a) Manutenzione correttiva
- b) Manutenzione preventiva
- c) Manutenzione migliorativa
- d) Manutenzione adattiva