

Progetto del software

Prova scritta – 13 settembre 2024 – 2h

PARTE 1 – RISPOSTA SINGOLA - Ogni domanda ha una sola risposta VERA.

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
 - Una risposta errata fa perdere il punteggio negativo riportato a fianco della domanda
 - Una risposta lasciata in bianco viene valutata 0
1. **(3, -.5)** Il principio di *dependency inversion*:
 - a) **E' sfruttato per disaccoppiare le varie parti di un sistema, applicando l'architettura CLEAN**
 - b) Non può essere usato per implementare i test funzionali
 - c) Non può essere usato per implementare i test di integrazione
 - d) Nessuna delle precedenti
 2. **(3, -.5)** Le *Software Requirement Specification* (SRS):
 - a) Sono usate per l'implementazione, ma non per la raccolta dei requisiti
 - b) Sono utili solamente in fase di contrattazione con il cliente, e poi possono venire ignorate
 - c) **Sono utili in fase di contrattazione col cliente, e poi servono da linea guida durante la fase di sviluppo**
 - d) Una volta scritte, non sono mai modificabili, neppure a fronte di un cambiamento di specifiche da parte del cliente
 3. **(3, -.5)** L'architettura dotNet:
 - a) Prevede l'utilizzo del solo C# come linguaggio *managed*
 - b) **Prevede l'utilizzo di una serie di linguaggi *managed*, ovvero gestiti dal Framework, e di linguaggi *unmanaged*, ossia non gestiti dal Framework**
 - c) Non è mai compatibile con l'utilizzo di linguaggi *unmanaged*, neppure "esterni" al framework
 - d) Nessuna delle precedenti

PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE -

Ogni domanda può avere da zero a quattro risposte CORRETTE.

- Ogni risposta esatta viene calcolata: +1
 - Ogni risposta errata viene calcolata: -0.5
 - Una risposta lasciata in bianco viene calcolata: 0
5. I diagrammi UML:
- a) Consentono di modellare gli aspetti di un sistema software, a vari livelli di astrazione
 - b) Non consentono di modellare gli scenari dei casi d'uso
 - c) Non consentono di modellare i repository git
 - d) Nessuna delle precedenti
4. I design pattern:
- a) Danno una soluzione precisa ad ogni problema, per ogni tipo di progetto
 - b) Danno una base concettuale e motivazionale per certe scelte progettuali, come risoluzione ai problemi che si incontra comunemente durante il processo di sviluppo
 - c) Non coprono mai la parte architetturale di un sistema
 - d) Possono coprire la parte architetturale di un sistema
5. Un'architettura client-server:
- a) Prevede una comunicazione asimmetrica
 - b) Prevede una comunicazione simmetrica
 - c) Prevede una comunicazione basata esclusivamente su pattern *Singleton*
 - d) Non prevede di utilizzare mai il pattern *Singleton* nel codice
6. Il tool Git:
- a) Non consente di tornare indietro ad una modifica precedente del codice
 - b) Non è compatibile con i flussi automatici di integrazione e pubblicazione degli artefatti software, la cosiddetta CI/CD (*continuous integration/continuous delivery*)
 - c) Può essere usato solo nei progetti di piccole dimensioni
 - d) Può essere usato solo nei progetti di grandi dimensioni
7. Il pattern MVC (Model-View-Control):
- a) Consente di suddividere un sistema nei tre componenti fondamentali al suo funzionamento
 - b) Non ha senso, nei sistemi moderni
 - c) E' ancora valido, nei sistemi moderni, ma ne sono state promosse delle varianti, quali ad esempio MV-VM
 - d) Nessuna delle precedenti

PARTE 3 – DOMANDE APERTE

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
- Una risposta errata può eventualmente causare una penalità che dipende dalla gravità dell'errore
- Una risposta lasciata in bianco viene calcolata: 0
- L'eventuale sfioramento del limite di righe o parole (laddove imposto), porterà a una decurtazione di un punto per ogni riga. Eventuali schematici e listati di codice non verranno presi in considerazione nel calcolo delle righe
- **SI RICORDA CHE L'UNICO FOGLIO DA CONSEGNARE E' IN CALCE AL COMPITO. QUESTO FOGLIO, PUO' SERVIRE ESCLUSIVAMENTE COME "BRUTTA COPIA". EVENTUALI RISPOSTE SCRITTE IN QUESTO FOGLIO NON VERRANNO PRESE IN CONSIDERAZIONE**

8. (7 pt) Si descriva il design pattern *Factory*. Non ci sono limiti di parole.

Nome e Cognome _____ Matricola: _____

9. (7 pt) Si descriva la differenza fra progettazione top-down e bottom-up, in massimo 5 righe.

Nome e Cognome _____ Matricola: _____

Progetto del software

Prova scritta – 13 settembre 2024 – 2h

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Questa è l'unica pagina che dovete consegnare.

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					3	-0.5
2					3	-0.5
3					3	-0.5
4						
5						
6						
7						
8						

Risposta alla domanda 9 (7 pt):

Nome e Cognome _____ Matricola: _____

Risposta alla domanda 10 (7 pt):