

## Progetto del software

### Prova scritta – 29 gennaio 2025 – 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)** L'architettura CLEAN
    - a) Consente di massimizzare la sicurezza cibernetica di un sistema
    - b) Prevede che le dipendenze vadano “verso l'esterno”, ovvero che eventuali strati software dipendano da tecnologie specifiche, come ad esempio MongoDB
    - c) Richiede una profonda conoscenza della programmazione ad oggetti
    - d) **Prevede l'utilizzo di *adapter* e *dependency inversion* per interfacciare tecnologie specifiche, come ad esempio MongoDB**
  
  2. **(3, -.5)** I *design pattern*
    - a) **Rappresentano soluzioni note a problemi noti**
    - b) Consentono di aumentare la scalabilità del sistema grazie all'utilizzo delle interfacce Java
    - c) Non sono di pratico utilizzo nei sistemi molto grandi
    - d) Nessuna delle precedenti
  
  3. **(3, -.5)** Il principio *Open/Close* della programmazione SOLID
    - a) Non si applica né alla programmazione ad oggetti, né a quella funzionale
    - b) Si può applicare sia alla programmazione ad oggetti, sia a quella funzionale
    - c) **Si applica solo alla programmazione ad oggetti**
    - d) Si applica solo alla programmazione funzionale

## 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
4. I diagrammi UML di sequenza
- a) Non possono venire utilizzati nello sviluppo del SW, ma solo per la documentazione da condividere con il cliente
  - b) Modellano una linea temporale di uno o più specifici caso d'uso
  - c) Modellano sia chiamate sincrone che asincrone
  - d) Modellano classi ed oggetti
5. Il design pattern *Singleton*
- a) Esiste per interfacciare due moduli software con contratti/interfacce/protocolli differenti
  - b) E' utile anche nella programmazione non ad oggetti
  - c) Può esistere solo nella programmazione ad oggetti
  - d) Nessuna delle precedenti
6. Il rapporto generalizzazione/specializzazione di un diagramma UML
- a) Può essere utilizzato per modellare la relazione padre-figlio in un diagramma delle classi
  - b) Può essere utilizzata per modellare la relazione cliente-fornitore in un diagramma delle classi
  - c) Consente di specificare interfacce e classi, sia astratte che concrete
  - d) Consente di specificare l'overriding di metodi di una classe
7. L'utilizzo di tool per il versioning come Git
- a) E' raramente richiesto dai customer, trattandosi di una metodologia "interna"
  - b) E' in conflitto con le architetture basate su sistemi distribuiti
  - c) Non impattano sul processo di sviluppo
  - d) Hanno un impatto iniziale sul tempo di sviluppo, ma i benefici superano questo svantaggio
8. Un approccio modulare *top-down*
- a) Richiede l'utilizzo di tool come Git
  - b) Parte da un'analisi di alto livello fatta sulla base delle specifiche
  - c) E' incompatibile con l'uso di librerie pre-esistenti
  - d) Talvolta può richiedere di partire da zero

### **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**

9. (7 pt) Si descriva cos'è il TRL in 3 righe

Nome e Cognome \_\_\_\_\_ Matricola: \_\_\_\_\_

**10. (4 pt)** Che cos'è un *Data Flow Diagram*?

Nome e Cognome \_\_\_\_\_ Matricola: \_\_\_\_\_

## Progetto del software

### Prova scritta – 29 gennaio 2025 – 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 (4 pt):**