

## Progetto del software

### Prova scritta – 12 settembre 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)** I requisiti funzionali di un software:
    - a) Sono quelli che danno valore al software stesso, e pertanto non sono concordati col cliente
    - b) Sono quelli che danno valore al software stesso, e pertanto sono direttamente concordati col cliente
    - c) Sono quelli relativi al processo di produzione del software, e generalmente non sono concordati col cliente
    - d) Nessuna delle precedenti
  
  2. **(3, -.5)** Il modello di sviluppo denominato *Waterfall*
    - a) Prevede una fase di specifica dei requisiti, ed una di test/assessment di tali requisiti
    - b) Prevede una fase di specifica dei requisiti, ma non una di test/assessment di tali requisiti
    - c) Non prevede la fase di specifica dei requisiti
    - d) Non prevede una fase di specifica dei requisiti, né una di test/assessment di tali requisiti
  
  3. **(3, -.5)** Il principio *Interface Segregation* della programmazione SOLID:
    - a) Si applica solo alla programmazione funzionale
    - b) Si può applicare anche alla programmazione funzionale
    - c) Prevede che una classe non possa estendere le caratteristiche di una classe padre
    - d) Prevede che una classe possa estendere una classe padre, a patto che possano venir sempre sostituite l'una con l'altra

## 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. 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
5. I design pattern:
- a) Forniscono soluzioni note a problemi noti, e specificano anche eventuali *side effect*/criticità nell'adottarle
  - b) Richiedono una profonda conoscenza del problema che si affronta, per evitare di cadere nella trappola degli *antipattern*
  - c) Sono contrapposti agli *antipattern*, ossia una serie di regole generali, chiare e note a tutti per evitare di programmare male
  - d) Non sono applicabili ai sistemi embedded, ma solo ai sistemi cloud
6. Un sistema asincrono:
- a) Può essere basato su interrupt
  - b) E' sempre basato su interrupt
  - c) Può essere basato su *callback*, a meno che non si adotti il paradigma *fire-and-forget*
  - d) Non è mai basato su *callback*
7. I sistemi di *versioning* come Git:
- a) Non si adattano ai progetti di piccole dimensioni
  - b) Non si adattano ai progetti di grandi dimensioni
  - c) Hanno tipicamente un costo economico elevato
  - d) Richiedono uno sforzo da parte degli utilizzatori per adottare una metodologia di lavoro strutturata e completa
8. I diagrammi UML di classe:
- a) Sono incompatibili con l'uso di librerie preesistenti
  - b) Sono accompagnati da disegni e rappresentazioni tabellari, chiare, univoche e complete, che specificano la struttura delle classi e (se serve) anche le loro relazioni
  - c) Consentono di raggruppare le classi in *package/namespaces*
  - d) Sono una rappresentazione degli studenti della classe, così come li vedo io dalla cattedra

### **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 in non più di 5 righe l'architettura CLEAN, a grandi linee

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

10. **(6 pt)** Si descriva la differenza fra *testing in the small* e *testing in the large*.

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

## Progetto del software

### Prova scritta – 12 settembre 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 (6 pt):**