

## Progetto del software

### Prova scritta – 7 giugno 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)** Un'architettura *client-server*
  - a) Non si applica ai sistemi distribuiti
  - b) Si applica ai sistemi distribuiti, ed è asimmetrica
  - c) Si applica ai sistemi distribuiti, ed è simmetrica
  - d) Prevede sempre l'utilizzo di un bus asincrono (es: MQTT) per lo scambio dei messaggi
  
2. **(3, -.5)** Un diagramma delle classi UML
  - a) Può essere utilizzato per modellare la struttura di una classe, ma non le relazioni fra più classi
  - b) Può essere utilizzato per modellare le relazioni fra più classi, ma non la struttura di una classe
  - c) Può essere utilizzato per modellare la struttura di una classe, ed anche le relazioni fra più classi
  - d) Non si applica alla programmazione ad oggetti
  
3. **(3, -.5)** Il principio *Single responsibility* della programmazione SOLID
  - a) Si applica solo alla programmazione ad oggetti
  - b) Si può applicare sia alla programmazione ad oggetti, sia a quella funzionale
  - c) Si applica solo alla programmazione funzionale
  - d) La domanda è mal posta: la programmazione funzionale non è mai stata insegnata, nel corso di laurea

## 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) Modellano sia chiamate sincrone che asincrone
    - b) Non possono venire utilizzati nello sviluppo del SW, ma solo per la documentazione
    - c) Modellano una linea temporale di uno o più specifici caso d'uso
    - d) Modellano classi ed oggetti
  
  5. Il design pattern *Adapter*
    - 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 utilizzata 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. Le proprietà funzionali del software
    - a) Sono tipicamente esplicitate dal *customer*
    - b) Sono raramente esplicitate dal *customer*
    - c) Sono, ad esempio, la manutenibilità, scalabilità, interoperabilità
    - d) Non impattano sul processo di sviluppo
  
  8. Un approccio modulare *bottom-up*
    - a) Esclude a priori l'utilizzo di componenti e librerie pre-esistenti
    - b) Parte da una visione di basso livello, dei componenti e librerie pre-esistenti
    - c) Parte da una visione di alto livello
    - 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 esaustivamente il modello architetturale sincrono, ad esempio, per un'architettura server

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

**10. (4 pt)** Si descrivano le principali fasi del modello di sviluppo più generico, il Waterfall

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

## Progetto del software

### Prova scritta – 7 giugno 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 (4 pt):**