

Domande a risposta multipla

Cicli di vita del Software

1. Si completi la seguente frase: "Un prototipo [...]"
 - a. [...] porta sempre e comunque alla produzione completa di uno o più sottosistemi del sistema software.
 - b. [...] porta alla realizzazione di una implementazione più o meno incompleta che serve a comprendere i requisiti del sistema. (X)
 - c. [...] porta sempre e comunque alla produzione completa dell'interfaccia utente.
 - d. [...] porta alla realizzazione di una implementazione più o meno incompleta che serve agli sviluppatori per poter dimostrare lo stato di avanzamento del prodotto ai loro project manager.

2. **Si completi la seguente frase: "Il modello a cascata [...]"**
 - o [...] è un modello lineare, senza ricicli che aiuta a tenere sotto controllo tempi e costi. (X)
 - o [...] consente di gestire i rischi di cattiva comprensione dei requisiti.
 - o [...] è stato il primo modello di ciclo di vita che ha consentito di prendere in considerazione la gestione dei rischi.
 - o [...] è apprezzato dai manager e dagli sviluppatori perchè è facile da comprendere e da applicare e non gode di buona trasparenza.

3. Si completi la seguente frase: "Un processo software è un [...]"
 - a. [...] insieme organizzato di attività che sovrintendono alla costruzione del prodotto da parte del team di sviluppo. (X)
 - b. [...] catalogo di standard, regolamenti e procedure per la realizzazione di sottosistemi software.
 - c. [...] metodo di sviluppo software basato sull'idea di prototipazione rapida per consentire una più efficace gestione dei requisiti funzionali.
 - d. [...] catalogo di pratiche adottato dai project manager per interagire con gli stakeholder interessati alla realizzazione di un prodotto software.

4. Si completi la seguente frase: "Uno strumento Upper-CASE [...]"
 - a. [...] supporta l'attività di interazione con gli stakeholder, offrendo strumenti in grado di agevolare la comunicazione.
 - b. [...] supporta l'attività della fase di analisi e specifica dei requisiti, oltre che la progettazione di un processo software. (X)
 - c. [...] supporta l'attività di progettazione di basso livello, inclusa la selezione e l'implementazione di strumenti di riuso.
 - d. [...] supporta l'attività di verifica e convalida del software, con particolare riferimento al testing di accettazione.

5. Si completi la seguente frase: "Uno strumento Lower-CASE [...]"

- a. [...] supporta l'attività di interazione con gli stakeholder, offrendo strumenti in grado di agevolare la comunicazione.
 - b. [...] supporta l'attività della fase di analisi e specifica dei requisiti, oltre che la progettazione di un processo software. (X)
 - c. [...] supporta l'attività di progettazione di basso livello, inclusa la selezione e l'implementazione di strumenti di riuso.
 - d. [...] supporta l'attività di verifica e convalida del software, con particolare riferimento al testing di accettazione.
6. Quale delle seguenti NON è una caratteristica/vantaggio dello *sviluppo incrementale*?
- a. Il rischio di fallimento è ridotto grazie ai continui rilasci.
 - b. I requisiti con più alta priorità sono quelli che verranno maggiormente testati.
 - c. I requisiti con una priorità più alta vengono rilasciati per prima.
 - d. Tutte le funzionalità sono implementate subito, così da poter essere successivamente raffinate. (X)
7. Quale delle seguenti meglio riflette la definizione di *sprint* in SCRUM?
- a. Una sprint rappresenta lo sforzo effettuato dal team di sviluppo per rilasciare un sistema software nei tempi prestabiliti con il cliente.
 - b. Una sprint consiste nella specifica dei requisiti funzionali e non funzionali che il team di sviluppo dovrà sviluppare per soddisfare gli stakeholder.
 - c. Una sprint consiste nell'insieme di attività condotte dal team di sviluppo per prioritizzare i requisiti non funzionali.
 - d. Una sprint rappresenta l'insieme di attività di progettazione, codifica e verifica di uno o più requisiti effettuati in un periodo di 2/4 settimane. (X)
8. Quale delle seguenti meglio riflette la definizione di *greenfield engineering*?
- a. Il processo di greenfield engineering avviene quando lo sviluppo di un'applicazione parte da zero, senza alcun sistema preesistente. (X)
 - b. Il processo di greenfield engineering avviene quando un sistema esistente deve essere migrato in un diverso ambiente.
 - c. Il processo di greenfield engineering avviene quando l'architettura di un sistema esistente deve essere re-ingegnerizzata nuovamente, preservando comunque le funzionalità esistenti.
 - d. Il processo di greenfield engineering avviene quando un sistema esistente deve essere adattato ad una nuova tecnologia.
9. Quale delle seguenti meglio riflette la definizione di *manutenzione del software*?
- a. Il processo di manutenzione avviene quando lo sviluppo di un'applicazione parte da zero, senza alcun sistema preesistente.
 - b. Il processo di manutenzione avviene quando un sistema esistente deve essere adattato ad una nuova tecnologia.

- c. Il processo di manutenzione avviene quando un sistema esistente deve essere migrato in un diverso ambiente.
 - d. Il processo di manutenzione avviene quando uno stakeholder richiede delle modifiche a un sistema già esistente. (X)
10. Quale delle seguenti meglio riflette la definizione di re-engineering?
- a. Il processo di re-engineering avviene quando lo sviluppo di un'applicazione parte da zero, senza alcun sistema preesistente.
 - b. Il processo di re-engineering avviene quando un sistema esistente deve essere adattato ad una nuova tecnologia. (X)
 - c. Il processo di re-engineering avviene quando un sistema esistente deve essere migrato in un diverso ambiente.
 - d. Il processo di re-engineering avviene quando uno stakeholder richiede delle modifiche a un sistema già esistente.
11. Quale delle seguenti affermazioni sullo sviluppo a cascata è VERA?
- a. I requisiti con più bassa priorità sono quelli che verranno maggiormente testati.
 - b. Il modello di sviluppo a cascata è facilmente comprensibile e applicabile. (X)
 - c. Il modello a cascata prevede la realizzazione di un prototipo per validare i requisiti.
 - d. Il rischio di fallimento è ridotto grazie ai continui rilasci.

Requirements Elicitation & Analysis

12. Si completi la seguente frase. "Un insieme di requisiti funzionali deve essere [...]"
- a. "[...] consistente, completo, misurabile, limitato."
 - b. "[...] consistente, incompleto, sostenibile, limitato."
 - c. "[...] completo, consistente, sostenibile, limitato." (X)
 - d. "[...] completo, limitato, coeso, sostenibile."
13. Quale dei seguenti punti NON è descritto in un *problem statement*?
- a. La situazione corrente del problema che si intende risolvere.
 - b. I deliverable attesi dagli stakeholder.
 - c. Le date di rilascio dei deliverable.
 - d. La specifica dei requisiti non funzionali. (X)
14. Si completi la seguente frase. "Un vincolo [...]"
- a. [...] è un aspetto del sistema visibile all'utente ma non strettamente connesso al suo comportamento funzionale.
 - b. [...] descrive le interazioni tra utente e sistema, specificando quindi il modo in cui il sistema software sarà utilizzato dal cliente.
 - c. [...] è una condizione imposta dal cliente o dall'ambiente in cui il sistema software dovrà operare. (X)
 - d. [...] rappresenta il limite massimo di risorse che il team di sviluppo avrà a disposizione per realizzare i requisiti funzionali del sistema software.

15. Quale delle seguenti linee guida sulla scrittura dei casi d'uso è FALSA?
- a. Il nome di un caso d'uso dovrebbe includere dei verbi.
 - b. Un caso d'uso dovrebbe descrivere un'interfaccia del sistema. (X)
 - c. Un caso d'uso dovrebbe descrivere una transazione utente completa.
 - d. Le relazioni causali tra i passi successivi dovrebbero essere chiare
16. Quale delle seguenti affermazioni relative ai sequence diagram è FALSA?
- a. I sequence diagram descrivono le interazioni tra oggetti che collaborano per svolgere un compito.
 - b. La sequenza dei messaggi scambiati tra i vari oggetti è mostrata su linea verticale. (X)
 - c. I sequence diagram possono corrispondere a uno scenario specifico o a un intero caso d'uso.
 - d. I sequence diagram evidenziano la sequenza temporale delle azioni.
17. Quale dei seguenti tipi di scenario fa riferimento alla descrizione dettagliata dei passi che un utente dovrà compiere per utilizzare il sistema software?
- a. Training scenario. (X)
 - b. As-is scenario.
 - c. Visionary scenario.
 - d. Evaluation scenario.
- 18. Quale delle seguenti non è una attività dell'analisi dei requisiti?**
- o Identificare gli Oggetti Control
 - o Identificare gli Attori (X)
 - o Identificare gli Attributi
 - o Modellare le Relazioni di Ereditarietà
19. Quale dei seguenti tipi di scenario fa riferimento alla descrizione dettagliata dei passi che un utente compie attualmente per la risoluzione del problema che dovrà essere risolto con l'introduzione di un sistema software?
- a. Training scenario.
 - b. As-is scenario. (X)
 - c. Visionary scenario.
 - d. Evaluation scenario.
20. Si completi la seguente frase: "Un caso d'uso è [...]"
- a. [...] una descrizione narrativa delle azioni e delle esperienze fatte dalle persone quando utilizzano un sistema software.
 - b. [...] esclusivamente utilizzato per descrivere un sistema software che verrà sviluppato in futuro.
 - c. [...] una modellazione delle funzionalità del sistema software dal punto di vista dell'utente. (X)
 - d. [...] una descrizione narrativa delle azioni che gli sviluppatori dovranno compiere per realizzare l'implementazione del sistema software.
21. Quale delle seguenti affermazioni sugli StateChart diagram è FALSA?

- a. Uno stato rappresenta una situazione in cui un oggetto ha un insieme di proprietà stabili.
- b. Una transizione modella un cambiamento di stato.
- c. In uno StateChart diagram l'attore compare per modificare lo stato iniziale. (X)
- d. Uno StateChart diagram specifica il ciclo di vita di un oggetto.

22. Quale delle seguenti affermazioni sugli StateChart diagram NON è *corretta*?

- a. Uno stato rappresenta una situazione in cui un oggetto ha un insieme di proprietà stabili.
- b. Una transizione modella un cambiamento di stato.
- c. In uno StateChart diagram l'attore compare per modificare lo stato iniziale. (X)
- d. Uno StateChart diagram specifica il ciclo di vita di un oggetto.

23. Si completi la seguente frase: "Gli oggetti entity [...]"

- a. [...] rappresentano le interazioni tra gli attori e il sistema (ad esempio, possono essere oggetti relativi all'interfaccia utente).
- b. [...] rappresentano l'informazione persistente (ad esempio, oggetti del dominio di applicazione). (X)
- c. [...] rappresentano le entità che andranno a utilizzare il sistema.
- d. [...] si occupano di realizzare gli use case, contengono la logica e determinano l'ordine dell'interazione degli oggetti.

24. Si completi la seguente frase: "Uno scenario è [...]"

- a. [...] una descrizione narrativa delle azioni e delle esperienze fatte dalle persone quando utilizzano un sistema software. (X)
- b. [...] esclusivamente utilizzato per descrivere un sistema software che verrà sviluppato in futuro.
- c. [...] una modellazione sistematica del sistema software dal punto di vista dell'utente.
- d. [...] una descrizione narrativa delle azioni che gli sviluppatori dovranno compiere per realizzare l'implementazione del sistema software.

25. Quali delle seguenti meglio riflette la definizione di *requisito non funzionale*?

- a. Un requisito non funzionale rappresenta una proprietà misurabile e percepibile di un sistema software che non è direttamente connessa alle azioni che il sistema fornirà agli utenti. (X)
- b. Un requisito non funzionale descrive le interazioni tra gli utenti del sistema software che verrà sviluppato e gli sviluppatori chiamati ad implementare il sistema software.
- c. Un requisito non funzionale è un modello, generalmente sviluppato tramite modellazione UML, che descrive la struttura statica del sistema software che si intende sviluppare.
- d. Un requisito non funzionale descrive le interazioni tra il sistema software e il suo ambiente, indipendentemente dalla sua implementazione.

26. Quali delle seguenti meglio riflette la definizione di *requisito funzionale*?

- a. Un requisito funzionale rappresenta una proprietà misurabile e percepibile di un sistema software che non è direttamente connessa alle azioni che il sistema fornirà agli utenti.
 - b. Un requisito funzionale descrive le interazioni tra gli utenti del sistema software che verrà sviluppato e gli sviluppatori chiamati ad implementare il sistema software. (X)
 - c. Un requisito funzionale è un modello, generalmente sviluppato tramite modellazione UML, che descrive la struttura statica del sistema software che si intende sviluppare.
 - d. Un requisito funzionale descrive le interazioni tra il sistema software e il suo ambiente, indipendentemente dalla sua implementazione.
27. Quali delle seguenti caratteristiche, riferite all'utilizzo dei requisiti funzionali come tecnica di specifica, è *falsa*?
- a. Un requisito funzionale dovrebbe essere *tracciabile*.
 - b. Un requisito funzionale dovrebbe essere *realizzabile*.
 - c. Nessuna delle altre. (X)
 - d. Un requisito funzionale dovrebbe essere *specifico*.
28. Quale dei seguenti è un attributo di qualità del software indicato dall'acronimo FURPS?
- a. Comprensibilità
 - b. Usabilità (X)
 - c. Portabilità
 - d. Leggibilità
29. Si completi la seguente frase: "Gli oggetti *control* [...]"
- a. [...] rappresentano le tabelle della base di dati che verranno memorizzate all'interno del sistema software.
 - b. [...] rappresentano le interazioni tra gli attori e il sistema (ad esempio, possono essere oggetti relativi all'interfaccia utente).
 - c. [...] si occupano di realizzare gli use case, rappresentano il controllo dei task eseguiti dal sistema, contengono la logica e determinano l'ordine dell'interazione degli oggetti. (X)
 - d. [...] rappresentano l'informazione persistente (ad esempio, oggetti del dominio di applicazione).
30. Quale delle seguenti affermazioni, riferite ai *sequence diagram*, è *corretta*?
- a. Un sequence diagram è più intuitivo degli use case dalla prospettiva degli utenti.
 - b. Un sequence diagram è adatto alla comunicazione con il cliente.
 - c. Un sequence diagram consente di modellare il comportamento statico del sistema software.
 - d. Un sequence diagram consente di identificare oggetti mancanti e/o aree non chiare nelle specifiche. (X)
31. Quale delle seguenti affermazioni, riferite ai *sequence diagram*, è *falsa*?

- a. Un sequence diagram si focalizza sull'interazione tra utente e sistema.
 - b. Un sequence diagram è adatto alla comunicazione con il cliente. (X)
 - c. Un sequence diagram complementa il class diagram.
 - d. Un sequence diagram consente di identificare oggetti mancanti e/o aree non chiare nelle specifiche
32. Si completi la seguente frase: "Nella modellazione UML, una *generalizzazione* [...]"
- a. [...] consente di identificare concetti più specifici a partire da quelli di più alto livello.
 - b. [...] consente di modellare la delega del lavoro di una classe ad un'altra.
 - c. [...] consente di organizzare concetti in gerarchie. (X)
 - d. [...] consente di identificare concetti astratti a partire da quello di più basso livello. (X)
33. Quale delle seguenti meglio riflette la definizione di *attore*?
- a. Un attore è il committente del progetto software.
 - b. Un attore è una specifica istanza di un'entità del problema che dovrà essere modellata all'interno del sistema.
 - c. Un attore è un'astrazione che rappresenta un'entità del problema che dovrà essere modellata all'interno del sistema.
 - d. Un attore è un'entità esterna al sistema, che interagisce con esso, che il progettista software dovrà modellare. (X)
34. Si completi la seguente frase: "Una user story [...]"
- a. [...] definisce esclusivamente a chi è rivolto un requisito funzionale da implementare nel contesto dell'applicazione da realizzare.
 - b. [...] rappresenta le esigenze degli utenti esprimendo in maniera generale, non dettagliata, caratteristiche, funzioni e requisiti per il prodotto da realizzare. (X)
 - c. [...] porta sempre e comunque alla produzione di un prototipo throw-away.
 - d. [...] è uno strumento utilizzato dai project manager per stimare costi, rischi e strategie di mitigazione dei rischi connessi alla realizzazione di requisiti non funzionali.
35. Quale delle seguenti euristiche è utilizzata per identificare le associazioni tra gli oggetti del dominio del problema?
- a. Esaminare il soggetto delle frasi.
 - b. Esaminare i complementi oggetto nelle frasi.
 - c. Esaminare gli avverbi nelle frasi.
 - d. Esaminare i verbi nelle frasi. (X)

System Design

36. Si completi la seguente frase: "La metrica di coesione [...]"
- a. [...] indica il grado di connessione tra due generiche classi o sottosistemi.
 - b. [...] indica il grado di connessione tra gli elementi (metodi o classi) all'interno di una classe o un sottosistema. (X)
 - c. [...] indica il grado di connessione tra due classi o sottosistemi.

- d. [...] indica il grado di connessione tra requisiti non funzionali e design goal.
37. Si completi la seguente frase: "La metrica di accoppiamento [...]"
- a. [...] indica il grado di connessione tra due generiche classi o sottosistemi. (X)
 - b. [...] indica il grado di connessione tra gli elementi (metodi o classi) all'interno di una classe o un sottosistema.
 - c. [...] indica il grado di connessione tra una classe implementata nel sistema software e la corrispettiva user story.
 - d. [...] indica il grado di connessione tra requisiti non funzionali e design goal.
38. Si completi la seguente frase: "In un'architettura software chiusa [...]"
- a. [...] i sottosistemi del livello centrale possono accedere a tutti gli altri sottosistemi.
 - b. [...] i sottosistemi di un livello possono accedere ai sottosistemi di qualunque livello inferiore.
 - c. [...] i sottosistemi di un livello possono accedere solo ai sottosistemi del livello immediatamente inferiore. (X)
 - d. [...] i sottosistemi di un livello possono accedere solo ai sottosistemi dello stesso livello.
39. Si completi la seguente frase: "In un'architettura software aperta [...]"
- a. [...] i sottosistemi del livello centrale possono accedere a tutti gli altri sottosistemi.
 - b. [...] i sottosistemi di un livello possono accedere ai sottosistemi di qualunque livello inferiore. (X)
 - c. [...] i sottosistemi di un livello possono accedere solo ai sottosistemi del livello immediatamente inferiore.
 - d. [...] i sottosistemi di un livello possono accedere solo ai sottosistemi dello stesso livello.
40. Quale delle seguenti attività è collegata al *mapping hardware/software*?
- a. Identificazione degli oggetti persistenti che dovranno essere memorizzati.
 - b. Identificazione e gestione dei processi concorrenti.
 - c. Gestione di come gli utenti del sistema dovranno accedere alle funzionalità del sistema software.
 - d. Gestione e comunicazione dei nodi dell'applicazione. (X)
41. Quale delle seguenti affermazioni, riferite al *deployment diagram*, è falsa?
- a. Un deployment diagram mostra il mapping hardware/software.
 - b. Un deployment diagram mostra la struttura del sistema a run-time.
 - c. Un deployment diagram mostra le dipendenze tra componenti e interfacce dei sottosistemi. (X)
 - d. Un deployment diagram è realizzato tramite box 3-D che rappresentano i nodi dell'applicazione.
42. Quale delle seguenti proprietà del software *non* è associata alla maintainability di un sistema software?
- a. Estensibilità.

- b. Robustezza. (X)
- c. Adattabilità.
- d. Leggibilità.

43. Quale delle seguenti affermazioni, riferite all'architettura *Model-View-Controller*, è vera?

- a. Nessuna delle altre opzioni.
- b. In questo tipo di architettura, il sottosistema View può richiedere e fornire servizi al sottosistema Model.
- c. Questo tipo di architettura è particolarmente adatta per gestire la distribuzione tra alcuni processori geograficamente separati.
- d. In questo tipo di architettura, il sottosistema Controller accede e modifica una singola struttura dati implementata nel Model. (X)

44. Quale delle seguenti affermazioni, riferite all'architettura *Model-View-Controller*, è falsa?

- a. L'architettura MVC è un caso particolare dell'architettura Repository.
- b. L'aggiornamento delle View avviene attraverso messaggi dai Controller. (X)
- c. I Model mantengono la conoscenza del dominio dell'applicazione.
- d. L'architettura MVC è appropriata per i sistemi interattivi.

45. Si completi la seguente frase: "I Design Goal (Obiettivi di design) [...]"

- a. [...] descrivono le qualità del sistema che uno sviluppatore dovrebbe ottimizzare. (X)
- b. [...] sono i requisiti non funzionali del sistema.
- c. [...] indicano in che modo le componenti software devono essere mappati su dei nodi hardware.
- d. [...] sono template di soluzioni utilizzabili per risolvere un insieme di problemi ricorrenti.

46. Si completi la seguente frase: "Le interfacce dei sottosistemi [...]"

- a. [...] descrivono i servizi offerti da un sottosistema agli altri sottosistemi. (X)
- b. [...] consentono al progettista di mappare i requisiti non funzionali sui design goal.
- c. [...] descrivono gli elementi hardware necessari all'utilizzo di un sottosistema.
- d. [...] forniscono template di soluzioni utilizzabili per risolvere un insieme di problemi ricorrenti.

47. Quale delle seguenti affermazioni, riferite al *layering*, è falsa?

- a. Il layering consente ai sottosistemi di un livello superiore di comunicare con i servizi offerti dai sottosistemi di livello inferiore.
- b. Il layering organizza i sottosistemi come peer che mutualmente forniscono servizi differenti agli altri sottosistemi. (X)
- c. Nessuna delle opzioni.
- d. Il layering consente ad un sistema di essere organizzato come una gerarchia di sottosistemi.

48. Si completi la seguente frase: "In una architettura client/server [...]"

- a. [...] i Client non conoscono i servizi offerti del Server.
- b. [...] i Server conoscono le interfacce dei Client.
- c. [...] gli utenti interagiscono solo con il Client. (X)
- d. [...] il flusso di controllo nei client e nei server è vincolato.

49. Quale delle seguenti attività è associata al System Design?

- a. Scomposizione in Sottosistemi. (X)
- b. Raccolta dei requisiti non funzionali.
- c. Identificazione dei Design Pattern.
- d. Identificazione degli Oggetti Control.

50. Si completi la seguente frase: “Un servizio è [...]”:

- o [...] un metodo implementato in una classe Java.
- o [...] un insieme di operazioni correlate fornite dal sottosistema per uno specifico scopo. (X)
- o [...] un insieme di operazioni correlate fornite da un nodo hardware.
- o [...] descritto in termini di prototipi.

51. Si completi la seguente frase: “Nella gestione della memorizzazione, i file [...]”:

- o [...] consentono un recupero elastico delle informazioni.
- o [...] non supportano l'esecuzione di operazioni di basso livello.
- o [...] necessitano dell'aggiunta di codice per fornire un opportuno livello di astrazione. (X)
- o Nessuna delle precedenti.

Object Design

52. Quale delle seguenti meglio riflette la definizione di *refactoring*?

- a. Il refactoring è una trasformazione che consente al codice sorgente di essere rappresentato tramite un modello ad oggetti.
- b. Il refactoring è una trasformazione che ha l'obiettivo di migliorare la struttura e/o leggibilità del codice sorgente senza però modificarne il suo comportamento esterno (ovvero, le funzionalità fornite). (X)
- c. Il refactoring è una trasformazione che consente agli elementi del modello ad oggetti di essere mappati sul codice sorgente.
- d. Il refactoring è una trasformazione che ha l'obiettivo di semplificare e/o ottimizzare il modello a oggetti definito in fase di object design.

53. Quale delle seguenti meglio riflette la definizione di *reverse engineering*?

- a. Il reverse engineering è una trasformazione che consente al codice sorgente di essere rappresentato tramite un modello ad oggetti. (X)
- b. Il reverse engineering è una trasformazione che ha l'obiettivo di migliorare la struttura e/o leggibilità del codice sorgente senza però modificarne il suo comportamento esterno (ovvero, le funzionalità fornite).
- c. Il reverse engineering è una trasformazione che consente agli elementi del modello ad oggetti di essere mappati sul codice sorgente.
- d. Il reverse engineering è una trasformazione che ha l'obiettivo di semplificare e/o ottimizzare il modello a oggetti definito in fase di object design.

54. Quale dei seguenti design pattern consente di *realizzare un'unica interfaccia per accedere ad un insieme di oggetti che compongono un sottosistema*?
- a. Composite.
 - b. Facade. (X)
 - c. Adapter.
 - d. Bridge.
55. Quale dei seguenti design pattern consente di *organizzare oggetti in strutture ad albero, in cui alcuni nodi sono oggetti complessi e le foglie degli oggetti semplici*?
- a. Composite. (X)
 - b. Facade.
 - c. Adapter.
 - d. Bridge.
56. Quale dei seguenti design pattern consente di separare l'interfaccia di una classe dalla sua implementazione?
- a. Composite.
 - b. Facade.
 - c. Adapter.
 - d. Bridge. (X)
57. Quale delle seguenti affermazioni, riferite al flusso di controllo procedure-driven, è vera?
- a. In questo flusso di controllo, il sistema può creare un numero arbitrario di threads, ognuno dei quali risponde ad un differente evento.
 - b. In questo flusso di controllo, il controllo risiede in un dispatcher che chiama le funzioni di sottosistema.
 - c. In questo flusso di controllo, il ciclo principale attende un evento esterno.
 - d. In questo flusso di controllo, i controlli risiedono nel codice del programma. (X)
58. Si completi la seguente frase: "Una invariante [...]"
- a. [...] è un predicato che è sempre vero per tutte le istanze di una classe. (X)
 - b. [...] è un predicato associato ad una specifica operazione e deve essere vero prima che l'operazione sia invocata.
 - c. [...] è un predicato associato ad una specifica operazione e deve essere vero dopo che l'operazione sia invocata.
 - d. [...] è un predicato associato a tutti i casi d'uso del sistema sotto progettazione.
59. Si completi la seguente frase: "Una preconditione [...]"
- a. [...] è un predicato che è sempre vero per tutte le istanze di una classe.
 - b. [...] è un predicato associato ad una specifica operazione e deve essere vero prima che l'operazione sia invocata. (X)
 - c. [...] è un predicato associato ad una specifica operazione e deve essere vero dopo che l'operazione sia invocata.
 - d. [...] è un predicato associato a tutti i casi d'uso del sistema sotto progettazione.

60. Si completi la seguente frase: "Una postcondizione [...]"
- a. [...] è un predicato che è sempre vero per tutte le istanze di una classe.
 - b. [...] è un predicato associato ad una specifica operazione e deve essere vero prima che l'operazione sia invocata.
 - c. [...] è un predicato associato ad una specifica operazione e deve essere vero dopo che l'operazione sia invocata. (X)
 - d. [...] è un predicato associato a tutti i casi d'uso del sistema sotto progettazione.
61. Quale delle seguenti attività NON è associata all'Object Design?
- a. Definizione della visibilità di attributi e operazioni di una classe.
 - b. Definizione di tipi e signature di attributi e operazioni di una classe.
 - c. Specifica di vincoli e contratti.
 - d. Convalida dell'adeguatezza delle operazioni rispetto ai requisiti. (X)
62. Quale tra i seguenti strumenti non abilita l'attività di riuso durante l'object design?
- a. Ereditarietà.
 - b. Composizione.
 - c. Modellazione logica. (X)
 - d. Design Pattern.
63. Quale delle seguenti affermazioni, riferite alla trasformazione nota come *Forward Engineering*, è corretta?
- a. Nel forward engineering, un insieme di istruzioni di codice sorgente è trasformato in un insieme di elementi del modello.
 - b. L'attività di forward engineering è realizzata con l'obiettivo di migliorare la manutenibilità e la leggibilità del codice sorgente.
 - c. Il forward engineering implica l'implementazione dei contratti.
 - d. L'attività di forward engineering è realizzata con l'obiettivo di mantenere una forte corrispondenza fra il modello di design ad oggetti ed il codice. (X)
64. Quali sono gli strumenti che si possono utilizzare per effettuare le attività di *riuso* durante l'object design?
- a. Ereditarietà, Composizione, Design Pattern. (X)
 - b. JavaDoc e documentazione.
 - c. Reverse Engineering ed euristiche.
 - d. Definizione di un modello logico e mapping hardware/software

Testing

65. Quale delle seguenti meglio riflette la definizione della tecnica di testing di integrazione nota come *sandwich*?
- a. Con questa tecnica, il sistema è visto come se avesse tre layer; di volta in volta il livello centrale è quello che sarà testato. (X)

- b. Con questa tecnica, i sottosistemi nella parte bassa della gerarchia sono i primi ad essere integrati, per poi man mano proseguire l'integrazione con le componenti a più alto livello.
- c. Con questa tecnica, i sottosistemi nella parte alta della gerarchia sono i primi ad essere integrati, per poi man mano proseguire l'integrazione con le componenti a più basso livello.
- d. Con questa tecnica, tutti i sottosistemi vengono integrati allo stesso momento.

66. Quali delle seguenti affermazioni, riferite al testing, è FALSA?

- a. Il testing dovrebbe sempre essere complementato con altre tecniche di verifica del codice sorgente, come ad esempio le ispezioni software.
- b. Il testing può solo essere utilizzato per verificare la presenza di difetti, ma non per verificarne l'assenza.
- c. Il testing fornisce prova della correttezza che un programma si comporti correttamente per tutti i possibili dati di ingresso. (X)
- d. Il testing è una tecnica dinamica che mira ad eseguire un programma con dati di input selezionati, verificando poi che il risultato atteso sia lo stesso di quello osservato.

67. Si completi la seguente frase: "Un *fault* [...]"

- a. [...] è il risultato di un errore nella documentazione o nel codice sorgente. (X)
- b. [...] è l'errore visualizzato da un utente in caso di fallimento del sistema.
- c. [...] è una implementazione parziale di componenti da cui la componente testata dipende.
- d. [...] è un insieme di input e di risultati attesi che esercitano una componente con lo scopo di causare fallimenti.

68. Si completi la seguente frase: "Una *failure* [...]"

- a. [...] è il risultato di un errore nella documentazione o nel codice sorgente.
- b. [...] è l'errore visualizzato da un utente in caso di fallimento del sistema. (X)
- c. [...] è una implementazione parziale di componenti da cui la componente testata dipende.
- d. [...] è un insieme di input e di risultati attesi che esercitano una componente con lo scopo di causare fallimenti.

69. Quale delle seguenti meglio riflette la definizione della tecnica di testing di integrazione nota come top-down?

- a. Con questa tecnica, i sottosistemi nella parte alta della gerarchia sono i primi ad essere integrati, per poi man mano proseguire l'integrazione con le componenti a più basso livello.
- b. Con questa tecnica, i sottosistemi nella parte bassa della gerarchia sono i primi ad essere integrati, per poi man mano proseguire l'integrazione con le componenti a più alto livello.
- c. Con questa tecnica, il sistema è visto come se avesse tre layer; di volta in volta il livello centrale è quello che sarà testato.

- d. Con questa tecnica, tutti i sottosistemi vengono integrati allo stesso momento.
70. Quale delle seguenti affermazioni, riferite al *testing di unità*, è vera?
- a. Il testing di unità è effettuato dagli sviluppatori per verificare la corretta integrazione dei sottosistemi sviluppati.
 - b. Il testing di unità è effettuato dagli utenti per verificare la comprensibilità del sistema software.
 - c. Il testing di unità è effettuato dagli sviluppatori utilizzando esclusivamente un approccio di tipo white-box.
 - d. Nessuna delle altre opzioni. (X)
71. Quale tra le seguenti è il principale vantaggio apportato dall'utilizzo della tecnica del *category partition*?
- a. Il category partition consente di esplicitare le relazioni semantiche tra diverse variabili, così da evitare la generazione di casi di test concettualmente irrilevanti. (X)
 - b. Il category partition consente di testare il codice sorgente in maniera esaustiva.
 - c. Il category partition consente di ottimizzare il numero di casi di stub e driver da sviluppare durante il testing di integrazione.
 - d. Nessuna delle altre opzioni.
72. Quale delle seguenti meglio riflette la definizione di *white-box testing*?
- a. Il testing white-box è una metodologia di selezione di casi di test che si focalizza sulle relazioni input-output degli input derivanti dall'analisi dei casi d'uso.
 - b. Il testing white-box è una metodologia di selezione di casi di test che si focalizza sulle relazioni input-output degli input derivanti dall'analisi del codice sorgente. (X)
 - c. Nessuna delle altre opzioni.
 - d. Il testing white-box è una metodologia di selezione di casi di test che si focalizza sulle specifiche del sistema.
73. Quale delle seguenti meglio riflette la definizione di *black-box testing*?
- a. Il testing black-box è una metodologia di selezione di casi di test che si focalizza sulle relazioni input-output degli input derivanti dall'analisi del codice sorgente.
 - b. Il testing black-box è una metodologia di selezione di casi di test che si focalizza sulle relazioni input-output degli input derivanti dall'analisi dei casi d'uso. (X)
 - c. Il testing black-box è una metodologia di selezione di casi di test che si focalizza sul modello dinamico dell'applicazione da sviluppare.
 - d. Nessuna delle altre opzioni.
74. Quale delle seguenti affermazioni, riferite al *testing di accettazione*, è vera?
- a. Il testing di accettazione è effettuato dagli sviluppatori per verificare la corretta integrazione dei sottosistemi sviluppati.

- b. Il testing di accettazione è effettuato dai clienti per verificare l'adeguatezza del sistema software. (X)
- c. Il testing di accettazione è effettuato dagli sviluppatori utilizzando esclusivamente un approccio di tipo white-box.
- d. Il testing di accettazione è effettuato dagli stakeholder per verificare l'usabilità del sistema software rilasciato.

75. Si completi la seguente frase: "Un *test driver* [...]"

- a. [...] è un'implementazione parziale dei requisiti funzionali da rilasciare.
- b. [...] è un'implementazione parziale di componenti da cui la componente testata dipende.
- c. [...] è un'implementazione parziale di una componente dipendente dalla componente testata. (X)
- d. [...] è un'implementazione parziale di un intero sottosistema.

76. Si completi la seguente frase: "Un *test stub* [...]"

- a. [...] è un'implementazione parziale dei requisiti funzionali da rilasciare.
- b. [...] è un'implementazione parziale di componenti da cui la componente testata dipende. (X)
- c. [...] è un'implementazione parziale di una componente dipendente dalla componente testata.
- d. [...] è un'implementazione parziale di un intero sottosistema.

77. Quali dei seguenti è un tipo di white-box testing?

- ☐ Sequence Testing.
- ☐ Boundary Testing.
- ☒ Path Testing. (X)
- ☐ Method Testing.

76. Quale delle seguenti affermazioni, relativa all'impatto dei requisiti sulle attività di testing, è falsa?

- ☐ La specifica degli use case influenza il Functional Testing.
- ☐ La decomposizione dei sottosistemi influenza lo Structural Testing.
- ☐ I requisiti non funzionali e i vincoli influenzano il Performance Testing.
- ☒ Il mapping hardware/software influenza lo Unit Testing. (X)

78. Quale tra le seguenti è il principale svantaggio della tecnica dell'*equivalence class testing*?

- a. La tecnica non consente di esplorare le combinazioni dei dati di input al test. (X)
- b. La tecnica non consente di verificare l'adeguatezza dei casi di test rispetto ai requisiti non funzionali del sistema da sviluppare.
- c. La tecnica è eccessivamente costosa da implementare.
- d. Nessuna delle altre opzioni.

78. Si completi la seguente frase: "La metrica di line coverage [...]"

- a. misura l'efficienza di un algoritmo implementato nel codice sorgente.
- b. indica la percentuale di linee di codice eseguite durante i test. (X)

- c. valuta la qualità dell'architettura definita in fase di system design.
- d. qualifica il numero di failure del sistema proposto.

Domande a risposta aperta

- 79. Descrivere brevemente tutte le attività del system design, focalizzandosi poi sull'attività di mapping hardware/software, indicando le principali scelte progettuali da effettuare e come rappresentarle tramite l'utilizzo di UML.
- 80. Descrivere il design pattern *Adapter*, fornendo anche un problema di progettazione in cui potrebbe essere adottato.
- 81. Descrivere il design pattern *Observer*, fornendo anche un problema di progettazione in cui potrebbe essere adottato.
- 82. Descrivere le principali differenze tra testing black-box e testing white-box.
- 83. Descrivere brevemente l'attività di decomposizione in sottosistemi, con particolare riferimento alla definizione all'utilizzo delle metriche di coesione e accoppiamento.
- 84. Descrivere il design pattern *Facade*, fornendo anche un problema di progettazione in cui potrebbe essere adottato.
- 85. Descrivere il design pattern *Proxy*, fornendo anche un problema di progettazione in cui potrebbe essere adottato.
- 86. Descrivere brevemente la metodologia di testing di integrazione nota come *sandwich*, evidenziandone punti di forza e limitazioni.
- 87. **Fornire una definizione di testing di unità, nonché descrivere gli approcci di weak and strong equivalence class testing, riportando le principali differenze tra questi.**
- 88. Descrivere brevemente la tecnica di testing nota come *Category Partition*, evidenziandone punti di forza e campi di utilizzo.
- 89. Fornire una definizione dei concetti di coesione e accoppiamento, nonché descrivere il modo in cui tali metriche possono essere utilizzate nel contesto della suddivisione in sottosistemi.
- 90. Descrivere le modalità di utilizzo dell'euristica di Abbott nel contesto dell'analisi e specifica dei requisiti.