

Ingegneria del Software (matr. pari-dispari): Esempi di domande a risposta multipla

1. Cosa è un metodo ? 1
 - Un sistema per fare qualcosa in modo migliore
 - Un procedimento generale per risolvere classi di problemi
 - Un particolare approccio o filosofia per fare qualcosa
2. Quale di queste affermazioni non è appropriata per il modello di processo a cascata ? 2
 - Le fasi del processo sono in progressione sequenziale
 - I semilavorati all'uscita di una fase sono congelati e non possono essere più modificati
 - Le funzionalità del sistema sono sviluppate in maniera incrementale
3. Un prototipo viene sviluppato per: 3
 - Ridurre i costi di sviluppo
 - Rilasciare rapidamente al cliente una prima versione del sistema
 - Interagire con il committente per convalidare i requisiti
4. Quale tipo di prototipazione parte con i requisiti meglio compresi ? 4
 - Prototipazione esplorativa
 - Prototipazione throw-away
 - Prototipazione mock-ups
5. Quale tipo di prototipazione tende a realizzare l'interfaccia utente? 5
 - Prototipazione esplorativa
 - Prototipazione breadboards
 - Prototipazione mock-ups
6. Quale di queste affermazioni è appropriata per il modello di processo incrementale ? 6
 - Ogni versione produce funzionalità/sottosistemi più affidabili
 - Ogni versione aggiunge nuove funzionalità/sottosistemi al sistema
 - Ogni versione raffina le funzionalità/sottosistemi che sono presenti fin dall'inizio
7. Quale di queste affermazioni è appropriata per il modello di processo iterativo ? 7
 - Ogni versione raffina le funzionalità/sottosistemi che sono presenti fin dall'inizio
 - Ogni versione aggiunge nuove funzionalità/sottosistemi al sistema
 - I requisiti a più alta priorità vengono rilasciati per primi

8. Cosa è un team ? (8)
- Un insieme di persone a cui è assegnato un task comune, ma che lavorano individualmente e senza necessità di interazione
 - Un insieme di persone che rivedono un work product e che propongono azioni
 - Un piccolo insieme di persone che lavorano in stretta interazione sulla stessa attività o task
9. Il tempo durante il quale un certo lavoro in un progetto deve essere fatto, viene denominato: (9)
- Work-product
 - Schedule
 - Task
10. Il lavoro che deve essere eseguito da un partecipante al progetto, viene denominato: (10)
- Work-product
 - Schedule
 - Task
11. Quale di queste affermazioni non è valida per una struttura organizzativa di reporting gerarchica ? (11)
- Lo stato è riportato dall'alto verso il basso
 - Le decisioni vengono comunicate dall'alto verso il basso
 - Le informazioni relative a stato e decisioni sono unidirezionali
12. Cosa è un liason all'interno di un team ? (12)
- Il leader del team
 - Il responsabile della comunicazione con un altro team
 - Il consulente amministrativo del team
13. Che tipo di ruolo è un configuration manager ? (13)
- Uno sviluppatore
 - Un manager
 - Un liason
14. Che cosa è un GANNT ? (14)
- Un grafico a barre che descrive lo schedule e la durata dei vari task
 - Un grafo che descrive le dipendenze tra i task del progetto
 - Un grafo che descrive le dipendenze tra i work-product del progetto
15. Che cosa è un PERT? (15)
- Un grafico a barre che descrive lo schedule e la durata dei vari task
 - Un grafo che descrive le dipendenze tra i task del progetto
 - Un grafo che descrive le dipendenze tra i work-product del progetto

16. Quale dei seguenti non è una comunicazione pianificata? (16)
- Definizione del problema
 - Review di progetto
 - Richiesta di modifica

17. Quale dei seguenti è una comunicazione non pianificata? (11)
- Risoluzione di un problema
 - Ispezione
 - Rilascio

18. Quale dei seguenti meccanismi di comunicazione è asincrono? (18)
- Intervista strutturata
 - Riunione
 - Posta elettronica

19. Cosa è un modello ? (13)
- Un'astrazione che descrive il sistema o un sottoinsieme di un sistema
 - Un insieme di regole grafiche o testuali per rappresentare viste
 - Una vista di aspetti del sistema

20. Cosa è una vista ? (20)
- Un'astrazione che descrive il sistema o un sottoinsieme di un sistema
 - Una visualizzazione di particolari aspetti di un modello
 - Un insieme di regole grafiche o testuali

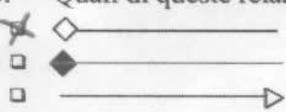
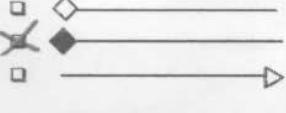
21. Cosa è una notazione ? (21)
- Un'astrazione che descrive il sistema o un sottoinsieme di un sistema
 - Una visualizzazione di particolari aspetti di un modello
 - Un insieme di regole grafiche o testuali per rappresentare viste

22. Cosa è UML ? (22)
- Una notazione grafica per progettare sistemi software
 - Un insieme di linguaggi per modellare software
 - Un modello astratto per descrivere sistemi software

23. Quale di questi diagrammi non è usato per descrivere il comportamento dinamico di un sistema software ? (23)
- Sequence diagram
 - Activity diagram
 - Use case diagram

24. I diagrammi dei casi d'uso: (24)
- Descrivono il comportamento funzionale del sistema così come visto dagli utenti
 - Descrivono il comportamento dinamico tra gli attori e il sistema e tra gli oggetti del sistema
 - Descrivono il comportamento dinamico di un sistema, in particolare il workflow
25. I diagrammi delle attività: (25)
- Descrivono il comportamento funzionale del sistema così come visto dagli utenti
 - Descrivono il comportamento dinamico tra gli attori e il sistema e tra gli oggetti del sistema
 - Descrivono il comportamento dinamico di un sistema, in particolare il workflow
26. I diagrammi di sequenza: (26)
- Descrivono il comportamento funzionale del sistema così come visto dagli utenti
 - Descrivono il comportamento dinamico tra gli attori e il sistema e tra gli oggetti del sistema
 - Descrivono il comportamento dinamico di un sistema, in particolare il workflow
27. I diagrammi di stato: (27)
- Descrivono la struttura statica del sistema: oggetti, attributi e relazioni
 - Descrivono il comportamento dinamico tra gli attori e il sistema e tra gli oggetti del sistema
 - Descrivono il comportamento dinamico di un singolo oggetto come una macchina a stati finiti
28. I diagrammi delle classi: (28)
- Descrivono la struttura statica del sistema: oggetti, attributi e relazioni
 - Descrivono il comportamento dinamico tra gli attori e il sistema e tra gli oggetti del sistema
 - Descrivono il comportamento dinamico di un singolo oggetto come una macchina a stati finiti
29. In UML le classi e gli oggetti (istanze) sono rappresentate con: (29)
- Rettangoli
 - Ovali
 - Rettangoli con angoli arrotondati
30. In UML i casi d'uso sono rappresentati con: (30)
- Rettangoli
 - Ovali
 - Rettangoli con angoli arrotondati
31. In UML gli stati sono rappresentati con: (31)
- Rettangoli
 - Ovali
 - Rettangoli con angoli arrotondati

32. Quale di questi elementi non è parte di un diagramma dei casi d'uso: (32)
- Attore
 - Caso d'uso
 - Classe
33. Quale di questi elementi non è parte di un diagramma di sequenza: (33)
- Attore
 - Caso d'uso
 - Oggetti
34. Che tipo di relazione può esistere tra un attore e un caso d'uso: (34)
- Associazione
 - Dipendenza
 - Generalizzazione
35. Che tipo di relazione può esistere tra due attori: (35)
- Associazione
 - Dipendenza
 - Generalizzazione R
36. Quale di queste relazioni non esiste tra due casi d'uso: (36)
- Associazione
 - Dipendenza
 - Generalizzazione
37. Quali di queste relazioni rappresenta una dipendenza: (37)
- -
 -
38. Quali di queste relazioni rappresenta una generalizzazione: (38)
- -
 -
39. Quali di queste relazioni rappresenta una associazione: (39)
- -
 -

40. Quali di queste relazioni rappresenta una aggregazione: (40)
- 
- 
- 
41. Quali di queste relazioni rappresenta una composizione: (41)
- 
- 
- 
42. Quali di questi è un requisito funzionale: (42)
- Il sistema deve visualizzare l'ora in base alla sua locazione
- Il tempo di risposta deve essere meno di un secondo
- Il linguaggio di implementazione deve essere Java
43. Quali di questi è un requisito non funzionale: (43)
- Il sistema deve visualizzare l'ora in base alla sua locazione
- Il tempo di risposta deve essere meno di un secondo
- Il linguaggio di implementazione deve essere Java
44. Cosa si intende per correttezza dei requisiti ? (44)
- I requisiti rappresentano la vista dell'utente
- Sono descritti tutti i possibili scenari del sistema
- Non ci sono requisiti funzionali e non funzionali che si contraddicono
45. Cosa si intende per completezza dei requisiti ? (45)
- I requisiti rappresentano la vista dell'utente
- Sono descritti tutti i possibili scenari del sistema
- Non ci sono requisiti funzionali e non funzionali che si contraddicono
46. Cosa si intende per consistenza dei requisiti ? (46)
- I requisiti rappresentano la vista dell'utente
- Sono descritti tutti i possibili scenari del sistema
- Non ci sono requisiti funzionali e non funzionali che si contraddicono
47. Cosa si intende per chiarezza dei requisiti ? (47)
- Non ci sono ambiguità nei requisiti
- I requisiti possono essere implementati e rilasciati
- Non ci sono requisiti funzionali e non funzionali che si contraddicono

48. Cosa si intende per realismo dei requisiti ? (48)
- Non ci sono ambiguità nei requisiti
 - I requisiti possono essere implementati e rilasciati
 - Per ogni funzione del sistema è possibile individuare un insieme di requisiti funzionali
49. Cosa si intende per tracciabilità dei requisiti ? (49)
- Non ci sono ambiguità nei requisiti
 - I requisiti possono essere implementati e rilasciati
 - Per ogni funzione del sistema è possibile individuare un insieme di requisiti funzionali
50. Quale di questi requisiti vanno affrontati durante il design ma non durante l'implementazione ? (50)
- Requisiti ad alta priorità
 - Requisiti a media priorità
 - Requisiti a bassa priorità
51. Cosa si intende per interface engineering ? (51)
- Lo sviluppo parte da zero, non esiste un sistema precedente
 - Riprogettazione o reimplementazione di un sistema esistente con nuove tecnologie
 - Fornire i servizi di un sistema esistente in un nuovo ambiente operativo
52. Che tipo di scenari sono usati per descrivere un sistema futuro? (52)
- as-is scenarios
 - visionary scenarios
 - training scenarios
53. Che tipo di scenari sono usati per descrivere un sistema esistente ? (53)
- as-is scenarios
 - visionary scenarios
 - training scenarios
54. Che tipo di scenari sono usati per guidare un utente nuovo nell'utilizzo del sistema ? (54)
- as-is scenarios
 - visionary scenarios
 - training scenarios
55. Perché si usano gli scenari ? (55)
- Per individuare i requisiti del sistema
 - Per analizzare le associazioni tra gli oggetti del sistema
 - Per analizzare il flusso di eventi di un caso d'uso

56.	Come si descrive un caso d'uso ?	(3)
	<input type="checkbox"/> Mediante uno scenario <input checked="" type="checkbox"/> Mediante un flusso di eventi <input type="checkbox"/> Mediante un insieme di requisiti funzionali	
57.	Cosa esprime una relazione di inclusione tra casi d'uso ?	(51)
	<input type="checkbox"/> Una variante del normale flusso di eventi di un caso d'uso <input type="checkbox"/> Una specializzazione di un caso d'uso <input checked="" type="checkbox"/> Una decomposizione funzionale di un caso d'uso	
58.	Cosa esprime una relazione di estensione tra casi d'uso ?	(58)
	<input checked="" type="checkbox"/> Una variante del normale flusso di eventi di un caso d'uso <input type="checkbox"/> Una specializzazione di un caso d'uso <input type="checkbox"/> Una decomposizione funzionale di un caso d'uso	
59.	Cosa esprime una relazione di generalizzazione tra casi d'uso ?	(59)
	<input type="checkbox"/> Una variante del normale flusso di eventi di un caso d'uso <input checked="" type="checkbox"/> Una specializzazione di un caso d'uso <input type="checkbox"/> Una decomposizione funzionale di un caso d'uso	
60.	Un oggetto che interagisce con un attore in un sequence diagram è:	(60)
	<input type="checkbox"/> Un entity object <input checked="" type="checkbox"/> Un boundary object <input type="checkbox"/> Un control object	
61.	Quale di queste categorie di requisiti non funzionali indica la facilità di cambiamenti al sistema dopo il rilascio ?	(61)
	<input type="checkbox"/> Reliability <input type="checkbox"/> Performance <input checked="" type="checkbox"/> Supportability	<i>Cop. 4.</i>
62.	Quale di queste categorie di preudo-requirements indica vincoli legati alle modalità di rilascio del sistema?	(62)
	<input type="checkbox"/> Interface requirements <input checked="" type="checkbox"/> Packaging requirements <input type="checkbox"/> Operations requirements	<i>Cop. 4.</i>
63.	Una trasformazione applicata agli oggetti di una classe e definita in fase di analisi è chiamata:	(63)
	<input checked="" type="checkbox"/> Operazione <input type="checkbox"/> Signature <input type="checkbox"/> Metodo	

64. Una connessione tra due istanze di oggetti è chiamata: 64
- Associazione
 - Link
 - Relazione

65. Una associazione è: 65
- Una connessione tra due istanze di oggetti
 - Una dipendenza tra classi
 - Un mapping bidirezionale tra classi

66. Quale di questi tipi di oggetti cambia meno frequentemente in un sistema software ? 66
- Boundary object
 - Control object
 - Entity object

67. Cosa indica un ruolo ? 67
- Una estremità dell'associazione
 - La navigabilità dell'associazione
 - La direzione del nome dell'associazione

68. Cosa indica la direzione di una associazione ? 68
- Una estremità dell'associazione
 - La navigabilità dell'associazione
 - La direzione del nome dell'associazione

69. Nel diagramma seguente, "filename"  69
- E' un attributo della classe directory
 - E' un attributo della classe file
 - E' il nome della associazione

70. Cosa indicano le frecce in un diagramma di sequenza ? 70
- Eventi inviati da un oggetto ad un altro
 - Relazioni tra oggetti
 - Dipendenze tra oggetti

71. Cosa indica la linea tratteggiata verticale sotto un oggetto in un diagramma di sequenza ? 71
- L'invio di un messaggio all'oggetto
 - Il periodo di attivazione di una operazione dell'oggetto
 - Il periodo durante il quale è possibile inviare un messaggio all'oggetto

72. Cosa indica un rettangolo verticale sotto un oggetto in un diagramma di sequenza ? 2
- L'invio di un messaggio all'oggetto
 - Il periodo di attivazione di una operazione dell'oggetto
 - Il periodo durante il quale è possibile inviare un messaggio all'oggetto
- b,e
73. Cosa dovrebbe contenere la prima colonna di un diagramma di sequenza ? 23
- L'attore che inizia il caso d'uso
 - Un boundary object
 - Il control object che gestisce il caso d'uso
74. Cosa dovrebbe contenere la seconda colonna di un diagramma di sequenza ? 24
- L'attore che inizia il caso d'uso
 - Un boundary object
 - Il control object che gestisce il caso d'uso
75. Cosa dovrebbe contenere la terza colonna di un diagramma di sequenza ? 25
- L'attore che inizia il caso d'uso
 - Un boundary object
 - Il control object che gestisce il caso d'uso
76. Quali di queste affermazioni non è corretta ? 26
- Un boundary object accede ad un entity object *vive*
 - Un entity object accede ad un control object *vive*
 - Un control object crea un boundary object
77. In un diagramma di stato, quando viene valutata la condizione di guardia su una transizione ? 27
- Quando il sistema si trova nello stato sorgente della transizione
 - Quando si verifica l'azione indicata sulla transizione *vedere*
 - Quando si verifica l'evento indicato sulla transizione
78. Quale di queste affermazioni non si riferisce ad un'attività in un diagramma di stato ? 28
- Operazione che prende tempo per essere completata
 - E' associata ad un evento
 - E' associata ad uno stato *vedere*
79. Quale di queste affermazioni non si riferisce ad un'azione in un diagramma di stato ? 29
- Operazione che prende tempo per essere completata
 - E' associata ad uno stato
 - E' associata ad un evento *vedere*

80. Quale di queste affermazioni si riferisce ad un diagramma di stato ? 80
- Serve ad individuare le relazioni temporali tra oggetti nel tempo
 - Serve ad individuare la sequenza di operazioni come risposta ad uno o più eventi
 - Serve ad individuare i cambiamenti degli oggetti nel tempo *vedere*
-
81. Quale di queste affermazioni non si riferisce ad un diagramma di sequenza ? 81
- Serve ad individuare i cambiamenti degli oggetti nel tempo
 - Serve ad individuare le relazioni temporali tra oggetti nel tempo *vedere*
 - Serve ad individuare la sequenza di operazioni come risposta ad uno o più eventi
-
82. Quale di questi prodotti dell'analisi dei requisiti vanno usati durante la fase di definizione degli obiettivi di design ? 82
- Modello dei casi d'uso
 - Modello a oggetti
 - Requisiti non funzionali
-
83. Quale di questi prodotti dell'analisi dei requisiti vanno usati durante la fase di decomposizione del sistema ? 83
- Modello dei casi d'uso *SCIDE può modello funzionale*
 - Modello a oggetti
 - Requisiti non funzionali
-
84. Quale di questi prodotti dell'analisi dei requisiti vanno usati durante la fase di design relativa a mapping hardware/software e a gestione dei dati persistenti ? 84
- Modello dei casi d'uso
 - Modello a oggetti
 - Requisiti non funzionali
-
85. Come viene rappresentato un sottosistema in UML ? 85
- Con una classe
 - Con un caso d'uso
 - Con un package
-
86. Cosa è un servizio di un sottosistema ? 86
- Un insieme di operazioni con signature completamente specificata
 - Un gruppo di operazioni che condividono uno scopo comune
 - Un insieme di associazioni, eventi e vincoli legati tra di loro
-
87. Cosa è l'interfaccia di un sottosistema ? 87
- Un insieme di operazioni con signature completamente specificata
 - Un gruppo di operazioni che condividono uno scopo comune
 - Un insieme di associazioni, eventi e vincoli legati tra di loro

88. Quali di queste affermazioni è vera ? 88
- L'accoppiamento misura le dipendenze tra le classi di un sottosistema
 - In un sistema con elevato accoppiamento le modifiche ad un sottosistema hanno forte impatto sugli altri sottosistemi
 - L'obiettivo del system design è massimizzare l'accoppiamento
89. Quali di queste affermazioni è vera ? 89
- La coesione misura le dipendenze tra i sottosistemi di un sistema
 - In sottosistemi con elevata coesione le modifiche ad un sottosistema hanno forte impatto sugli altri sottosistemi
 - Le classi di un sottosistema con elevata coesione eseguono task simili
90. A cosa servono le partizioni di un sistema ? 90
- A dividere verticalmente un sistema in sottosistemi debolmente accoppiati
 - A dividere orizzontalmente un sistema in sottosistemi a diversi livelli di astrazione
 - A dividere un sistema sia orizzontalmente che verticalmente in sottosistemi indipendenti
91. A cosa servono i layer di un sistema ? 91
- A dividere verticalmente un sistema in sottosistemi debolmente accoppiati
 - A dividere orizzontalmente un sistema in sottosistemi a diversi livelli di astrazione
 - A dividere un sistema sia orizzontalmente che verticalmente in sottosistemi indipendenti
92. A chi fornisce servizi un layer di un sistema ? 92
- Ai layer di livello più basso
 - Ai layer dello stesso livello
 - Ai layer di livello più alto
93. In un'architettura software aperta: 93
- I sottosistemi di un livello possono accedere solo ai sottosistemi del livello immediatamente inferiore
 - I sottosistemi di un livello possono accedere ai sottosistemi di qualunque livello inferiore
 - I sottosistemi di un livello possono accedere sia ai sottosistemi dei livelli inferiori che a quelli dei livelli superiori
94. In un'architettura software chiusa: 94
- I sottosistemi di un livello possono accedere solo ai sottosistemi del livello immediatamente inferiore
 - I sottosistemi di un livello possono accedere solo ai sottosistemi dello stesso livello
 - I sottosistemi di un livello possono accedere ai sottosistemi di qualunque livello inferiore

101.	Le relazioni tra i componenti in un component diagram sono	101
	<input type="checkbox"/> associazioni <input checked="" type="checkbox"/> dipendenze <input type="checkbox"/> generalizzazioni	
102.	Quale di queste affermazioni relative ad un component diagram è falsa ?	
	<input checked="" type="checkbox"/> Mostra il mapping hardware/software <input type="checkbox"/> Mostra la struttura del sistema a compilation time <input type="checkbox"/> Mostra la struttura a design time	
103.	Quale di queste affermazioni relative ad un deployment diagram è falsa ?	
	<input type="checkbox"/> Mostra la struttura del sistema a run-time <input type="checkbox"/> Mostra il mapping hardware/software <input checked="" type="checkbox"/> Mostra le dipendenze tra componenti e interfacce dei sottosistemi	
104.	Le relazioni tra i nodi di un deployment diagram sono:	104
	<input checked="" type="checkbox"/> associazioni <input type="checkbox"/> dipendenze <input type="checkbox"/> generalizzazioni	
105.	In un thin client model	
	<input checked="" type="checkbox"/> Il sistema è two thier e la logica applicativa è eseguita sul server <input type="checkbox"/> Il sistema è two thier e la logica applicativa è eseguita sul client <input type="checkbox"/> Il sistema è three thier e la logica applicativa è eseguita sul client	
106.	Se il mio obiettivo è fornire implementazioni diverse per uno stesso sottosistema	106
	<input checked="" type="checkbox"/> Uso un bridge pattern <input type="checkbox"/> Uso un adapter pattern <input type="checkbox"/> Uso un façade pattern	
107.	Se il mio obiettivo è quello di realizzare una architettura chiusa	107
	<input type="checkbox"/> Uso un adapter pattern <input type="checkbox"/> Uso un bridge pattern <input checked="" type="checkbox"/> Uso un façade pattern	
108.	Un invariant è	108
	<input type="checkbox"/> Un predicato che deve essere vero prima dell'invocazione di un metodo di una classe <input type="checkbox"/> Un predicato che è vero dopo l'invocazione di un metodo di una classe <input checked="" type="checkbox"/> Un predicato che è vero prima e dopo l'invocazione di un metodo di una classe	

109. Quale di queste affermazioni relative a JavaDoc è corretta ?

- Il commento che precede un metodo consente di specificare la precondizione del metodo ✓
- Il commento che precede un metodo consente di specificare la postcondizione del metodo ✓
- Il commento che precede un metodo consente di specificare l'invariante della classe F

110. Se devo realizzare una associazione qualificata uso come struttura dati

- Una tabella
- Un insieme
- Una lista

111. Se devo estendere una classe mi interessa conoscere

- Tutti i membri privati, protetti e pubblici della classe
- Solo i membri protetti e pubblici della classe
- Solo i membri pubblici della classe

112. La realizzazione delle associazioni del modello a oggetti

- E' una trasformazione del modello a oggetti R
- E' una trasformazione di tipo forward engineering
- E' una trasformazione di refactoring

113. Se devo memorizzare dati voluminosi e necessari per un periodo breve di tempo

- uso una struttura dati in memoria
- uso un file
- uso un database

113

114. Se devo effettuare una trasformazione prima sul codice sorgente e poi sul modello a oggetti faccio

- Prima operazioni di forward engineering e poi di refactoring
- Prima operazioni di object model transformation e poi di forward engineering
- Prima operazioni di refactoring e poi di reverse engineering

115. Se eredito un contratto in UML quale di queste affermazioni non è corretta ?

- La precondizione del metodo nella sottoclassa consente al metodo di gestire meno casi del corrispondente metodo nella superclasse
- Il metodo nella sottoclassa deve assicurare la stessa postcondizione del corrispondente metodo nella superclasse
- L'invariante della sottoclassa può essere più restrittivo dell'invariante della superclasse

115

116. Quali di questi operazioni non è opportuno implementare durante la realizzazione di contratti ?

- Controllare che la precondizione all'inizio di un metodo sia soddisfatta
- Controllare che la postcondizione all'uscita di un metodo sia soddisfatta
- Incapsulare il codice di controllo in metodi separati ai fini del riuso R

117. L'evento percepito dall'utente come differenza tra comportamento atteso di un sistema software e comportamento esibito dal sistema software viene denominato
- Failure
 Fault
 Error
118. Nel testing di integrazione bottom-up ho bisogno di realizzare
- Test Driver
 Test Stub
 Sia test driver che test stub
119. Quale di queste affermazioni è falsa ?
- Nel testing white box la derivazione dei casi di test mira a coprire la maggior parte di una classe di elementi della struttura del codice
 Nel testing white box il numero di casi di test da eseguire dipende dal criterio di copertura adottato
 Nel testing white-box non si controlla se l'output prodotto in corrispondenza di un input è uguale all'output atteso
120. Quando si effettua il testing di regressione ?
- Dopo aver effettuato il test di integrazione e prima di effettuare il test di sistema
 Dopo aver effettuato il testing di sistema e prima di effettuare il testing di accettazione
 Dopo aver effettuato delle modifiche al codice per correggere un fault

IS - OBIETTIVI 2^a PARTE

80. Quale di queste affermazioni si riferisce ad un diagramma di stato ?
- Serve ad individuare le relazioni temporali tra oggetti nel tempo
 - Serve ad individuare la sequenza di operazioni come risposta ad uno o più eventi
 - Serve ad individuare i cambiamenti degli oggetti nel tempo
81. Quale di queste affermazioni non si riferisce ad un diagramma di sequenza ?
- Serve ad individuare i cambiamenti degli oggetti nel tempo
 - Serve ad individuare le relazioni temporali tra oggetti nel tempo
 - Serve ad individuare la sequenza di operazioni come risposta ad uno o più eventi
- CAPITOLO 6-7
82. Quale di questi prodotti dell'analisi dei requisiti vanno usati durante la fase di definizione degli obiettivi di design ?
- Modello dei casi d'uso
 - Modello a oggetti
 - Requisiti non funzionali
83. Quale di questi prodotti dell'analisi dei requisiti vanno usati durante la fase di decomposizione del sistema ?
- Modello dei casi d'uso
 - Modello a oggetti
 - Requisiti non funzionali
84. Quale di questi prodotti dell'analisi dei requisiti vanno usati durante la fase di design relativa a mapping hardware/software e a gestione dei dati persistenti ?
- Modello dei casi d'uso
 - Modello a oggetti
 - Requisiti non funzionali
85. Come viene rappresentato un sottosistema in UML ?
- Con una classe
 - Con un caso d'uso
 - Con un package
86. Cosa è un servizio di un sottosistema ?
- Un insieme di operazioni con signature completamente specificata
 - Un gruppo di operazioni che condividono uno scopo comune
 - Un insieme di associazioni, eventi e vincoli legati tra di loro
87. Cosa è l'interfaccia di un sottosistema ?
- Un insieme di operazioni con signature completamente specificata
 - Un gruppo di operazioni che condividono uno scopo comune
 - Un insieme di associazioni, eventi e vincoli legati tra di loro

Ponte Imzule

① Che cos'è un metodo?

un sistema x fare qualcosa in modo migliore

* un procedimento generale a risolvere classi di problemi.

un particolare approccio o filosofia x fare qualcosa.

② Quelli di queste affermazioni non è appropriato x il modello di processo e crescere?

Le funzionalità del sistema sono sviluppate in maniera incrementale

③ Un prototipo viene sviluppato x

interagire con il committente x considerare i requisiti.

④ Quale tipo di prototipazione pone con i requisiti meglio compresi?

prototipazione esplorativa

⑤ Quale tipo di prototipazione tende a realizzare l'interfaccia utente?

prototipazione mock-up.

⑥ Quale di queste affermazioni è appropriata x il modello di processo matematico?

Ogni versione aggiunge nuove funzionalità/basi sistemi al sistema

⑦ Quale di queste affermazioni è appropriata?

88. Quali di queste affermazioni è vera ?
- L'accoppiamento misura le dipendenze tra le classi di un sottosistema
 - In un sistema con elevato accoppiamento le modifiche ad un sottosistema hanno forte impatto sugli altri sottosistemi
 - L'obiettivo del system design è massimizzare l'accoppiamento
89. Quali di queste affermazioni è vera ? *accoppiamento*
- La coesione misura le dipendenze tra i sottosistemi di un sistema
 - In sottosistemi con elevata coesione le modifiche ad un sottosistema hanno forte impatto sugli altri sottosistemi
 - Le classi di un sottosistema con elevata coesione eseguono task simili
90. A cosa servono le partizioni di un sistema ?
- A dividere verticalmente un sistema in sottosistemi debolmente accoppiati
 - A dividere orizzontalmente un sistema in sottosistemi a diversi livelli di astrazione
 - A dividere un sistema sia orizzontalmente che verticalmente in sottosistemi indipendenti
91. A cosa servono i layer di un sistema ?
- A dividere verticalmente un sistema in sottosistemi debolmente accoppiati
 - A dividere orizzontalmente un sistema in sottosistemi a diversi livelli di astrazione
 - A dividere un sistema sia orizzontalmente che verticalmente in sottosistemi indipendenti
92. A chi fornisce servizi un layer di un sistema ?
- Ai layer di livello più basso
 - Ai layer dello stesso livello
 - Ai layer di livello più alto
93. In un'architettura software aperta:
- I sottosistemi di un livello possono accedere solo ai sottosistemi del livello immediatamente inferiore
 - I sottosistemi di un livello possono accedere ai sottosistemi di qualunque livello inferiore
 - I sottosistemi di un livello possono accedere sia ai sottosistemi dei livelli inferiori che a quelli dei livelli superiori
94. In un'architettura software chiusa:
- I sottosistemi di un livello possono accedere solo ai sottosistemi del livello immediatamente inferiore
 - I sottosistemi di un livello possono accedere solo ai sottosistemi dello stesso livello
 - I sottosistemi di un livello possono accedere ai sottosistemi di qualunque livello inferiore

1^o fase (Individuazione - CVS)

- ① Che cos'è un metodo?
 - Un procedimento generale a soluzioni classi di problemi
- ② Quale di queste affermazioni non è appropriata e il modello di processo è corretto?
 - le funzionalità del sistema sono sviluppate in mettere incrementale
- ③ Un prototipo viene sviluppato a
 - Inizialmente con il committente e con validare i requisiti.
- ④ Quale tipo di prototipazione porta con i requisiti meglio compresi?
 - Prototipazione esplorativa
- ⑤ Quale tipo di prototipazione si deve realizzare prima delle fasi di analisi?
 - Prototipazione mock-up
- ⑥ Quale di queste affermazioni è appropriata e il modello di processo incrementale?
 - Ogni versione aggiunge nuove funzionalità/sottosistemi al sistema
- ⑦ Quale di queste affermazioni è appropriata e il modello di processo iterativo?
 - Ogni versione raffina le funzionalità/sottosistemi che sono presenti fin dall'inizio

Cap 3

- ⑧ Che cos'è un Team?
 - Un piccolo insieme di persone che lavorano in stretta interazione sulle stesse attività o Task
- ⑨ Il tempo durante il quale in certo lavoro in un progetto deve essere fatto, viene denominato:
 - Schedule
- ⑩ Il lavoro che deve essere eseguito da un partecipante al progetto, viene denominato:
 - Task
- ⑪ Quale di queste affermazioni non è valida a me simili organizzative di reporting gerarchiche?
 - Lo stesso è rispetto dell'elenco verso il basso
- ⑫ Così in liaison all'interno di un Team?
 - il responsabile delle comunicazioni con un elenco Team
- ⑬ Che tipo di ruolo è la configurazione manager? (VEDERE)

- ⑭ Che cos'è un GANT?
 - Un grafico a barre che descrive le schedule e le durate dei vari Task
- ⑮ Che cos'è un PERT?
 - Un grafo che descrive le dipendenze fra i Task del progetto
- ⑯ Quale dei seguenti non è una comunicazione più efficace?
 - Richiesta di modifica
- ⑰ Quale dei seguenti è una comunicazione non più efficace?
 - Risoluzione di un problema
- ⑱ Quale dei seguenti meccanismi di comunicazione è esclusivo posti elettronici

guardare design goals

95. Quali sono gli obiettivi di design di un'architettura software aperta?

- Manutenibilità
- Efficienza (a man-tene)
- Affidabilità

96. Quali sono gli obiettivi di design di un'architettura software chiusa?

- Usabilità
- Efficienza
- Manutenibilità (+ portabile)

97. Quale di queste affermazioni è falsa?

- In un'architettura client-server, il server fornisce i servizi al client
- In un'architettura client-server, il server conosce l'interfaccia del client
- Una repository architecture è un caso particolare di architettura client-server *e' il contrario...*

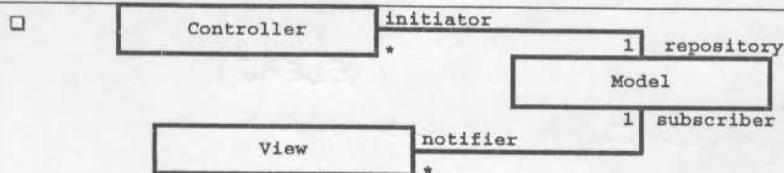
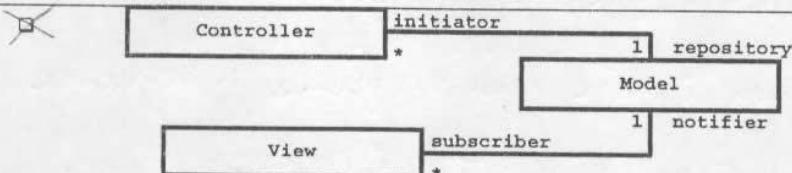
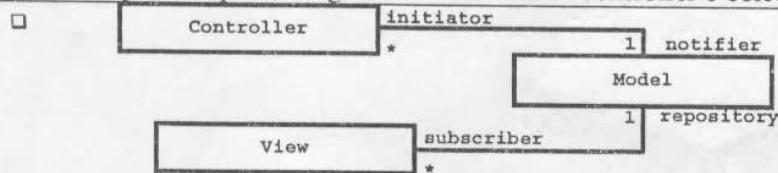
98. Quali di queste affermazioni è vera?

- In un'architettura peer-to-peer non c'è differenza tra client e server
- Un'architettura peer-to-peer favorisce una gestione centralizzata dei dati
- Un'architettura peer-to-peer consente di evitare problemi di deadlock *falso che li introduce*

99. Un'architettura model-view-controller, il sottosistema "model":

- E' responsabile della conoscenza del dominio applicativo
- E' responsabile della visualizzazione degli oggetti del dominio applicativo
- E' responsabile della sequenza dei interazioni con l'utente

100. Dire quali di questi diagrammi model-view-controller è corretto:



Cop 2

- ⑯ Che cos'è un modello?
 - Un'estensione che descrive il sistema o un sottosistema di un sistema
- ⑰ Che cos'è una vista?
 - Una visualizzazione di particolari aspetti di un modello
- ⑱ Che cos'è una notazione?
 - Un insieme di regole grafiche o simboli per rappresentare viste
- ⑲ Che cos'è UML?
 - Un insieme di linguaggi e modelli di software
- ⑳ Quale di questi diagrammi non è usato a descrivere il comportamento dinamico di un sistema software?
 - Use case diagram
- ㉑ I diagrammi dei casi d'uso
 - descrivono il comportamento funzionale del sistema così come è visto dagli utenti
- ㉒ I diagrammi delle attività
 - descrivono il comportamento dinamico di un sistema, in particolare il workflow
- ㉓ I diagrammi di sequenze
 - descrivono il comportamento dinamico fra gli attori, il sistema e fra gli oggetti del sistema
- ㉔ I diagrammi di stati
 - descrivono il comportamento dinamico di un singolo oggetto come ricezione e stati finiti
- ㉕ I diagrammi delle classi
 - descrivono la struttura statica del sistema: oggetti, attributi e relazioni
- ㉖ In UML le classi e gli oggetti sono rappresentate con rettangoli
- ㉗ In UML i casi d'uso sono rappresentati con ovali
- ㉘ In UML gli stati sono rappresentati con rettangoli con angoli arrotondati
- ㉙ Quale di questi elementi non è parte di un diagramma di casi d'uso? Classe
 -
- ㉚ Quale di questi elementi non è parte di un diagramma di sequenze (caso d'uso)
 -
- ㉛ Che tipo di associazione può esistere fra un attore e un caso d'uso?
 - Associazione
- ㉜ Che tipo di associazione può esistere fra due attori
 - Generalizzazione (VEDERE)
- ㉝ Quale di queste relazioni non esiste fra due casi d'uso?
 - Associazione
- ㉞ Quale di queste relazioni rappresenta una dipendenza
 - ---->
- ㉟ Quale di queste relazioni rappresenta una generalizzazione
 - ----->
- ㉟ Quale di queste relazioni rappresenta un associazione
 - ----->

CAPITOLO 7

101. Le relazioni tra i componenti in un component diagram sono
- associazioni
 - dipendenze
 - generalizzazioni

102. Quale di queste affermazioni relative ad un component diagram è falsa ?
- Mostra il mapping hardware/software
 - Mostra la struttura del sistema a compilation time
 - Mostra la struttura a design time

103. Quale di queste affermazioni relative ad un deployment diagram è falsa ?
- Mostra la struttura del sistema a run-time
 - Mostra il mapping hardware/software
 - Mostra le dipendenze tra componenti e interfacce dei sottosistemi

104. Le relazioni tra i nodi di un deployment diagram sono:
- associazioni
 - dipendenze
 - generalizzazioni

105. In un thin client model
- Il sistema è two thier e la logica applicativa è eseguita sul server
 - Il sistema è two thier e la logica applicativa è eseguita sul client
 - Il sistema è three thier e la logica applicativa è eseguita sul client

CAPITOLO 8

106. Se il mio obiettivo è fornire implementazioni diverse per uno stesso sottosistema
- Uso un bridge pattern
 - Uso un adapter pattern
 - Uso un façade pattern

107. Se il mio obiettivo è quello di realizzare una architettura chiusa
- Uso un adapter pattern
 - Uso un bridge pattern
 - Uso un façade pattern

108. Un invariant è
- Un predicato che deve essere vero prima dell'invocazione di un metodo di una classe
 - Un predicato che è vero dopo l'invocazione di un metodo di una classe
 - Un predicato che è vero prima e dopo l'invocazione di un metodo di una classe

- (40) Quelli di questi criteri rappresentano la raggruppazione
 • 
- (41) Quelli di questi criteri rappresentano la composizione
 • 

Cap 4

- (42) Quelli di questi, è in rapporto funzionale?
 • Il sistema deve visualizzare cose in base alle sue conoscenze
- (43) Quelli di questi, è in rapporto non funzionale?
 • Il tempo di risposta deve essere meno di 1 secondo
- (44) Cose si intende x correttezza dei requisiti?
 • I requisiti rappresentano il punto di vista dell'utente
- (45) Cose si intende x completezza dei requisiti?
 • Sono descritti tutti i possibili scenari del sistema
- (46) Cose si intende x consistenza dei requisiti?
 • Non ci sono requisiti funzionali e non funzionali che si contraddicono
- (47) Che cose si intende x chiarezza dei requisiti?
 • Non ci sono ambiguità nei requisiti
- (48) Cose si intende x realismo dei requisiti?
 • I requisiti possono essere implementati e rilesinati
- (49) Cose si intende x ricercabilità dei requisiti?
 • Per ogni funzione del sistema è possibile individuare un insieme di requisiti funzionali
- (50) Quelli di questi, requisiti vengono effettuati durante il sistema ma non durante l'implementazione!

VEDERE

- (51) Cose si intende x interface engineering?
 • Funzi., servizi di un sistema esistente in un nuovo ambiente operativo
- (52) Che tipo di scenari sono usati x descrivere un sistema futuro?
 • Visionary - scenario
- (53) Che tipo di scenario sono usati x descrivere un sistema esistente?
 • QS-IS scenario
- (54) Che tipo di scenari sono usati x guidare un utente nuovo nell'utilizzo del sistema?
 • Training scenarios
- (55) Xki si usano gli scenario?
 • X individuare i requisiti del sistema

VEDERE

- (56) Come si definisce in caso d'uso?
 • Mediante un flusso di eventi
- (57) Cose esprime le relazioni di inclusione fra casi d'uso?
 • La decomposizione funzionale del caso d'uso
- (58) Cose esprimono le relazioni di estensione fra i casi d'uso?
 • Le variazioni del normale flusso di eventi di un caso d'uso

VEDERE

- (59) Cose esprimono le relazioni di generalizzazione fra i casi d'uso?
 • La specializzazione di un caso d'uso

VEDERE

109. Quale di queste affermazioni relative a JavaDoc è corretta ?
- Il commento che precede un metodo consente di specificare la precondizione del metodo
 Il commento che precede un metodo consente di specificare la postcondizione del metodo
 Il commento che precede un metodo consente di specificare l'invariante della classe
110. Se devo realizzare una associazione qualificata uso come struttura dati
- Una tabella
 Un insieme
 Una lista
111. Se devo estendere una classe mi interessa conoscere
- Tutti i membri privati, protetti e pubblici della classe
 Solo i membri protetti e pubblici della classe
 Solo i membri pubblici della classe
112. La realizzazione delle associazioni del modello a oggetti
- E' una trasformazione del modello a oggetti
 E' una trasformazione di tipo forward engineering
 E' una trasformazione di refactoring
113. Se devo memorizzare dati voluminosi e necessari per un periodo breve di tempo
- uso una struttura dati in memoria
 uso un file
 uso un database
114. Se devo effettuare una trasformazione prima sul codice sorgente e poi sul modello a oggetti faccio
- Prima operazioni di forward engineering e poi di refactoring
 Prima operazioni di object model transformation e poi di forward engineering
 Prima operazioni di refactoring e poi di reverse engineering
115. Se eredito un contratto in UML quale di queste affermazioni non è corretta ?
- La precondizione del metodo nella sottoclassse consente al metodo di gestire meno casi del corrispondente metodo nella superclasse
 Il metodo nella sottoclassse deve assicurare la stessa postcondizione del corrispondente metodo nella superclasse
 L'invariante della sottoclassse può essere più restrittivo dell'invariante della superclasse
116. Quali di questi operazioni non è opportuno implementare durante la realizzazione di contratti ?
- Controllare che la precondizione all'inizio di un metodo sia soddisfatta
 Controllare che la postcondizione all'uscita di un metodo sia soddisfatta
 Incapsulare il codice di controllo in metodi separati ai fini del riuso

Cap 5

- 60 Un oggetto Ke mi regissee con un etice in un sequenze di spese i boundary object
- 61 Quelli di queste categorie di oggetti non funzionali indice le feature di cambiamenti el tempi dopo il release?
- **Sufficiency** 61 Cap. 4
- 62 Quelli di queste categorie pseudo-requirements indice vincoli legati alle modellazioni di riferimento del sistema
- **Packaging requirements** 62 Cap. 4
- 63 Una trasformazione applicata agli oggetti di una classe i definiti in faz di analisi, i
• **Operative** VEDERE
- 64 Una connessione fra due istanze di oggetti, e Kremote
- **Associative**
- 65 Un associazione i
- Una connessione fra due istanze di oggetti.
- 66 Quelli di questi tipi di oggetti cambia meno frequentemente in sistemi software?
- **Entity object**
- 67 Cose indice in ruolo?
- 68 Cose indice le direttive di un associazione? VEDERE Cap. 2
- **Univocità dell'associazione** 62 Cap. 2 UNI
 - **de leggibilità dell'associazione**
- 69 Nel diagramma seguente "Prenome" ~~Diretta~~ ~~Indiretta~~ ~~Reflexiva~~ CAP2 Qualificazione è un attributo della classe file
- 70 Cose indice le palle in un diagramma di sequenze?
- Eventi, inviti, de un oggetto ed un altro
- 71 Cose indice che trattengono varie sotto oggetto in un diagramma di sequenze?
- Il periodo durante il quale i possibile muore un messaggio all'oggetto
- 72 Cose indice in settorelo vittuale sotto un oggetto di un diagramma di sequenze?
- Il periodo di attivazione di un operazione dell'oggetto.
- 73 Cose dovrebbe contenere la 1^a colonna di un diagramma di sequenze?
- L'etichetta Ke mire il cesso d'uso
- 74 Cose dovrebbe contenere la 2^a colonna di un diagramma di sequenze?
- Un boundary object
- 75 Cose dovrebbe contenere la 3^a colonna di un diagramma di sequenze?
- Il causal object Ke gestisce il cesso d'uso
- 76 Quelli di queste affermazioni non i corrette?
- **entity object** accade ad un boundary
- 77 In un diagramma di sette, quanto viene detto le cardinalità di qualche sottosistema?
- 78 Quando si verifica l'etichetta michele sulla transizione
- Quale di queste ~~etichette~~ affermazioni non si riferisce ad un attività in un diaj.

CAPITOLO 11

117. L'evento percepito dall'utente come differenza tra comportamento atteso di un sistema software e comportamento esibito dal sistema software viene denominato
- Failure
 - Fault
 - Error
118. Nel testing di integrazione bottom-up ho bisogno di realizzare
- Test Driver
 - Test Stub
 - Sia test driver che test stub
119. Quale di queste affermazioni è falsa ?
- Nel testing white box la derivazione dei casi di test mira a coprire la maggior parte di una classe di elementi della struttura del codice
 - Nel testing white box il numero di casi di test da eseguire dipende dal criterio di copertura adottato
 - Nel testing white-box non si controlla se l'output prodotto in corrispondenza di un input è uguale all'output atteso
120. Quando si effettua il testing di regressione ?
- Dopo aver effettuato il test di integrazione e prima di effettuare il test di sistema
 - Dopo aver effettuato il testing di sistema e prima di effettuare il testing di accettazione
 - Dopo aver effettuato delle modifiche al codice per correggere un fault