

Quiz Teoria

Il SEMAT associa 4 gradi progressivi di avanzamento (*milestone*) all'attività di codifica, denominati "*architecture selected*"; "*demonstrable*"; "*usable*", "*ready*". Nel nostro progetto didattico abbiamo usato il concetto di *baseline*, istanziandolo in TB e PB. Collega tali nozioni, proponendo una corrispondenza tra la TB e la PB del progetto didattico, singolarmente, e qualcuna delle *milestone* citate.

Scegli una o più alternative:

- ☒ a. La TB, essendo associata a un PoC, corrisponde bene alla *milestone* "*demonstrable*"; la PB invece sta tra "*usable*" e "*ready*". **Bene:** questa è la corrispondenza più vicina all'interpretazione che abbiamo dato a TB e PB nel progetto didattico.
- ☐ b. Nessuna associazione è possibile, perché non vi è alcuna corrispondenza tra *milestone* e *baseline*, che sono concetti tra loro indipendenti.
- ☐ c. Più corrispondenze sono possibili, in ragione dell'interpretazione che vogliamo dare di TB e PB.

Risposta corretta.

La risposta corretta è: La TB, essendo associata a un PoC, corrisponde bene alla *milestone* "*demonstrable*"; la PB invece sta tra "*usable*" e "*ready*".

Pensando alla funzione del *repository*, per contenuto e destinatari, all'interno di un progetto *software*, scegli una tra le seguenti affermazioni:

Scegli una o più alternative:

- ☒ a. L'uso di un *repo* permette di dare ordine e stabilità al lavoro collaborativo. La natura dei prodotti di progetto ne determina l'organizzazione interna. **Affermativo.** Di conseguenza, però, occorre ragionare bene su quale organizzazione interna dare al *repo* di progetto, considerata la possibile eterogeneità dei suoi contenuti, tra codice, documenti, e altri artefatti.
- ☐ b. Il *repo* contiene esclusivamente artefatti *software* in sviluppo o in manutenzione, va centralizzato (per progetto), e serve **solo** ai programmatori. Nessun altro deve avervi accesso.
- ☐ c. Il *repo* è risorsa personale e privata di ogni singolo sviluppatore, che ne determina anche i contenuti. La funzione primaria del *repo* è proteggere lo stato del lavoro di quella persona da intrusioni e malfunzionamenti.

Risposta corretta.

La risposta corretta è: L'uso di un *repo* permette di dare ordine e stabilità al lavoro collaborativo. La natura dei prodotti di progetto ne determina l'organizzazione interna.

Iniziato	venerdì, 27 agosto 2021, 14:30
Stato	Completato
Terminato	venerdì, 27 agosto 2021, 14:34
Tempo impiegato	4 min. 28 secondi
Valutazione	6,00 su un massimo di 6,00 (100%)

Domanda 1
Risposta corretta
Punteggio ottenuto 3,00 su 3,00
V
Contrassegna domanda

Uno dei compiti ricorrenti del responsabile di progetto (*project manager*) è la compilazione del preventivo a finire (PaF), al termine di ogni periodo di rendicontazione. Vogliamo sapere se tale atto implichi o meno la ripianificazione delle attività rimanenti nel tempo residuo.

☐ 1. Il piano di progetto è da modificare solo in conseguenza dell'emergere di gravi criticità: doverlo modificare, pertanto, è "cattiva notizia". Quanto più possibile, dunque, conviene preservare il piano iniziale, cosicché la compilazione del PaF avvenga naturalmente per sottrazione del consuntivo dei consumi rilevati nel periodo di rendicontazione.

☐ 2. Certamente no. La compilazione del PaF è un atto di natura strettamente contabile: tutto ciò che serve in esso è tenere conto dei consumi consuntivati nel periodo di rendicontazione, eventualmente modificando l'impiego dei ruoli, per assicurare che il costo totale del progetto non ecceda quanto pattuito contrattualmente.

☒ 3. La ripartizione del tempo di progetto in brevi periodi, con obiettivi e risorse contingentati, ognuno dei quali culminante in una retrospettiva, serve ad attenuare i rischi e a favorire il miglioramento continuo. Tale retrospettiva riguarda il rapporto tra l'avanzamento conseguito e il consumo riscontrato. Dalla sua analisi scaturiscono indicazioni utili alla rivisitazione del piano delle attività rimanenti, e da esso la determinazione del PaF corrispondente.

Affermativo. Questa è la visione corretta della questione.

Risposta corretta.

La risposta corretta è:
La ripartizione del tempo di progetto in brevi periodi, con obiettivi e risorse contingentati, ognuno dei quali culminante in una retrospettiva, serve ad attenuare i rischi e a favorire il miglioramento continuo. Tale retrospettiva riguarda il rapporto tra l'avanzamento conseguito e il consumo riscontrato. Dalla sua analisi scaturiscono indicazioni utili alla rivisitazione del piano delle attività rimanenti, e da esso la determinazione del PaF corrispondente.

Domanda 2
Risposta corretta
Punteggio ottenuto 3,00 su 3,00
V
Contrassegna domanda

Vostri colleghi di lavoro discutono se la copertura dei requisiti, come indicatore quantitativo, sia determinabile in un qualunque momento del progetto, oppure lo sia solo al momento del *test* di sistema. Vedendovi freschi di studi, vi chiedono di selezionare la più plausibile tra le seguenti affermazioni al riguardo.

☐ 1. Per copertura dei requisiti intendiamo la percentuale di *test* funzionali superati con successo. In tale accezione, tale valore può essere determinato solo dopo tali *test*.

☐ 2. Per copertura dei requisiti intendiamo la proporzione di requisiti utente, espressi nel capitolato, presi in carico formalmente dal fornitore. In tale accezione, l'indicatore viene determinato in sede di revisione dei requisiti, alla stipula del contratto.

☒ 3. Per copertura dei requisiti intendiamo la percentuale di requisiti software associati a specifici artefatti/componenti/parti del prodotto come determinati dalla progettazione (*design*); la successiva campagna di *test* serve a confermare l'effettivo grado di copertura conseguito a valle della codifica.

Affermativo. Questa è l'accezione canonica.

Risposta corretta.

La risposta corretta è:
Per copertura dei requisiti intendiamo la percentuale di requisiti software associati a specifici artefatti/componenti/parti del prodotto come determinati dalla progettazione (*design*); la successiva campagna di *test* serve a confermare l'effettivo grado di copertura conseguito a valle della codifica.

Domanda 1

Risposta corretta
V
Contrassegna domanda

In una discussione con un vostro collega senior (cioè con più lunga vita professionale), il tema vira sull'utilità dei *test* di unità. Il vostro collega è scettico al riguardo, soprattutto perché – dice – non vi è una definizione chiara e univoca di cosa sia una "unità" in un programma *software*. Il vostro interlocutore vi chiede di esprimervi sul tema, scegliendo una tra le seguenti risposte:

Scegli una o più alternative:

- ☐ a. Vero. La nozione di "unità" *software* non è ben definita, e questo è fonte di confusione indesiderabile. Meglio concentrarsi sui *test* di sistema, dove non vi sono dubbi sulla natura dell'oggetto di *test*.
- ☐ b. Dipende. Per certi linguaggi di programmazione e certi tipi di programmi, il *test* di unità o non serve o è troppo oneroso. Per altri invece, vi è buona sinergia tra la tecnologia e la verifica e allora le cose "vanno da sole".
- ☒ c. Falso. Cosa sia l'elemento unitario di un programma *software* è una nozione architettuale sufficientemente chiara: una parte del tutto, piccola abbastanza da essere verificata individualmente (priva di catene di dipendenze significative), ma grande abbastanza di meritarsi verifica (cioè con correttezza non accertabile "a vista"). **Posizione condivisibile:** questa è la posizione canonica. Naturalmente, la sua attuazione concreta richiede buon senso architettuale.

Risposta corretta.

La risposta corretta è: Falso. Cosa sia l'elemento unitario di un programma *software* è una nozione architettuale sufficientemente chiara: una parte del tutto, piccola abbastanza da essere verificata individualmente (priva di catene di dipendenze significative), ma grande abbastanza di meritarsi verifica (cioè con correttezza non accertabile "a vista").

Domanda 2

Risposta corretta
V
Contrassegna domanda

I professionisti dello sviluppo *software* concordano, in generale, che i metodi agili siano da preferire. In un colloquio di lavoro, il vostro intervistatore vi chiede di provare a mettere in relazione lo sviluppo incrementale con quello agile, scegliendo una tra le seguenti risposte:

Scegli un'alternativa:

- ☒ a. Entrambi guardano nella stessa direzione, ma lo sviluppo agile contempla il rischio di iterazioni distruttive, che invece lo sviluppo incrementale cerca in ogni modo di evitare. **Visione ben fondata:** quello indicato nella risposta è il rapporto effettivo tra i due modelli di sviluppo in questione.
- ☐ b. Si tratta di visioni radicalmente diverse e tra loro incompatibili. Lo sviluppo agile è flessibile, e adattivo. Quello incrementale è pieno di vincoli.
- ☐ c. Perfetta identità. Ogni metodo di sviluppo agile è intrinsecamente incrementale.

Risposta corretta.

La risposta corretta è: Entrambi guardano nella stessa direzione, ma lo sviluppo agile contempla il rischio di iterazioni distruttive, che invece lo sviluppo incrementale cerca in ogni modo di evitare.

Domanda 1

Risposta corretta

▼ Contrassegna domanda

Il SEMAT associa 4 gradi progressivi di avanzamento (*milestone*) all'attività di codifica, denominati "**architecture selected**"; "**demonstrable**"; "**usable**"; "**ready**".

Assumendo che a una *milestone* corrisponda una *baseline*, proponi una associazione tra la **TB** (*Technology Baseline*) e la **PB** (*Product Baseline*) del progetto didattico e qualcuna di tali *milestone*.

Scegli un'alternativa:

- ☐ a. Nessuna associazione è possibile, perché non vi è corrispondenza tra *milestone* e *baseline*.
- ☒ b. La TB, essendo associata a un PoC, corrisponde bene alla *milestone* "**demonstrable**"; la PB invece sta tra "**usable**" e "**ready**". ✓ **Bene:** questa è la corrispondenza più vicina all'interpretazione che abbiamo dato a TB e PB nel progetto didattico.
- ☐ c. Più di una corrispondenza è possibile, in funzione dell'interpretazione che diamo a TB e PB.

Risposta corretta.

La risposta corretta è: La TB, essendo associata a un PoC, corrisponde bene alla *milestone* "**demonstrable**"; la PB invece sta tra "**usable**" e "**ready**".

Domanda 2

Risposta corretta

▼ Contrassegna domanda

La metrica "**efficacia**" misura il grado di conformità del risultato di una attività con le attese, mentre la metrica "**efficienza**" è funzione inversa della quantità di risorse impiegate per raggiungere l'obiettivo prefissato, oltre il minimo necessario.

Ritieni vi sia un ordine di precedenza, in pianificazione, nel fissare gli obiettivi quantitativi di tali metriche?

Scegli un'alternativa:

- ☒ a. Pianificando, prima decido fino a che livello voglio soddisfare le attese, e poi calcolo il costo corrispondente, iterando fino a quanto i costi preventivati siano entro gli eventuali vincoli contrattuali. ✓ **Ragionevole:** non a caso, ogni pianificazione si basa su studio di fattibilità e analisi dei requisiti.
- ☐ b. Prima di tutto decido quanto posso e voglio spendere, e poi, con quello, vedo fin dove posso andare.
- ☐ c. Nessun ordine di precedenza: si tratta di metriche completamente indipendenti tra loro.

Risposta corretta.

La risposta corretta è: Pianificando, prima decido fino a che livello voglio soddisfare le attese, e poi calcolo il costo corrispondente, iterando fino a quanto i costi preventivati siano entro gli eventuali vincoli contrattuali.

A lezione abbiamo visto che, secondo il SEMAT, il gruppo incaricato (**team**) di svolgere un lavoro (**work**) applica un **way of working** per organizzare e condurre le attività necessarie. Vogliamo sapere cosa venga fissato prima tra *team*, *work* e *way of working*.

- ☒ a. Il **way of working**: prima di ogni altra cosa, si ✓ **Sì.** Questa è la modalità che abbiamo provato a praticare nel progetto didattico, ed è quella più vicina al caso reale, dove il *way of working* è fondamento dell'organizzazione.
- ☐ b. Il lavoro (**work**) da svolgere: prima occorre determinare quale sia il lavoro da svolgere; il resto segue.
- ☐ c. Il **team**: prima di tutto, si decide chi dovrà svolgere il lavoro; il resto segue.

Risposta corretta.

La risposta corretta è:

Il **way of working**: prima di ogni altra cosa, si istituisce una prassi (le norme che guidano il lavoro); il resto segue.

La metrica "efficacia" misura il grado di conformità del risultato di una attività con le corrispondenti attese.

La metrica "efficienza" è funzione inversa della quantità di risorse impiegate per raggiungere l'obiettivo prefissato, oltre il minimo necessario.

Vogliamo sapere se vi sia un ordine di precedenza, nella pianificazione di progetto, nel fissare gli obiettivi quantitativi di tali metriche.

- ☐ a. Prima di tutto decido quanto posso e voglio spendere, e poi, con quello, vedo fin dove posso andare.
- ☒ b. Pianificando, prima decido fino a che livello voglio soddisfare le attese, e poi calcolo il costo corrispondente, iterando fino a quanto i costi preventivati siano entro gli eventuali vincoli contrattuali. ✔ **Ragionevole:** non a caso, ogni pianificazione si basa su studio di fattibilità e analisi dei requisiti.
- ☐ c. Nessun ordine di precedenza: si tratta di metriche completamente indipendenti tra loro.

Risposta corretta.

La risposta corretta è:

Pianificando, prima decido fino a che livello voglio soddisfare le attese, e poi calcolo il costo corrispondente, iterando fino a quanto i costi preventivati siano entro gli eventuali vincoli contrattuali.

Quiz Pratica

Si selezionino tra le seguenti le affermazioni corrette relative alle architetture a **monolite** e a **microservizi**.

Scegli una o più alternative:

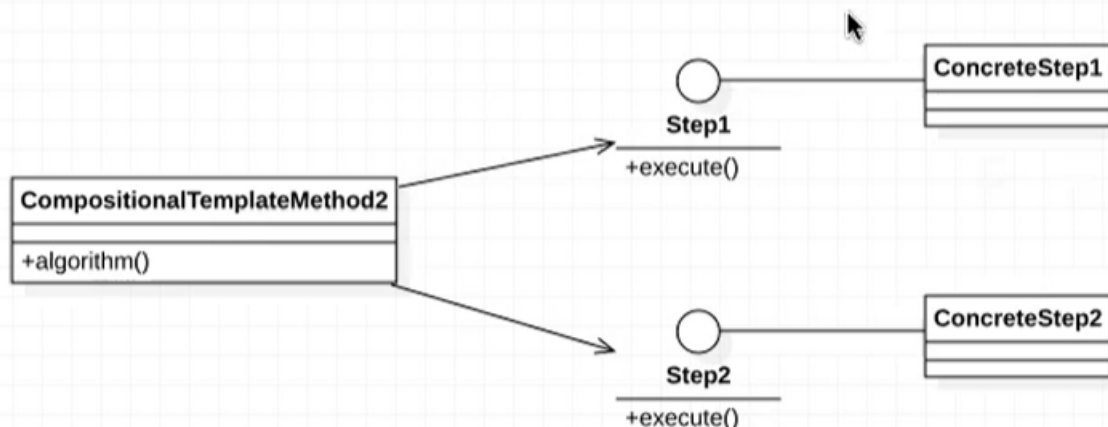
- ☒ a. Formando un sistema fortemente distribuito, in un'architettura a microservizi si dovrebbe preferire uno stile di comunicazione asincrona ad uno stile sincrono. ✓
- ☒ b. Secondo la Legge di Conway, un'architettura a microservizi dovrebbe essere utilizzata soprattutto all'interno di organizzazioni che hanno gruppi di sviluppo per lo più indipendenti. ✓
- ☐ c. Un monolite è uno stile architetturale sorpassato, completamente rimpiazzato dei microservizi in ogni tipo di applicazione.
- ☒ d. Un monolite di per se, non induce forzatamente all'accoppiamento tra le componenti che lo compongono. ✓
- ☒ e. E' possibile aggiornare solamente una piccola componente di un'architettura a monolite, senza dover effettuare il deploy dell'intera applicazione. ✗
- ☐ f. I microservizi possono condividere il medesimo database, leggendo e scrivendo dalle medesime tabelle, collezioni, ecc... In questo modo, è possibile condividere le informazioni fra i microservizi, mantenendo però la completa segregazione del codice sorgente.

Risposta parzialmente esatta.

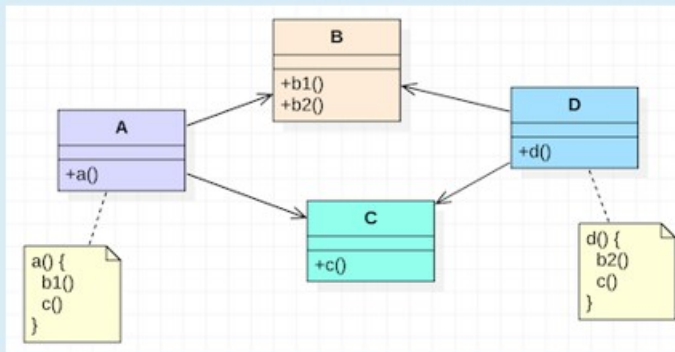
Hai selezionato troppe opzioni.

Le risposte corrette sono: Secondo la Legge di Conway, un'architettura a microservizi dovrebbe essere utilizzata soprattutto all'interno di organizzazioni che hanno gruppi di sviluppo per lo più indipendenti., Un monolite di per se, non induce forzatamente all'accoppiamento tra le componenti che lo compongono., Formando un sistema fortemente distribuito, in un'architettura a microservizi si dovrebbe preferire uno stile di comunicazione asincrona ad uno stile sincrono.

Si fornisca il diagramma delle classi del *Template Method Pattern*. Si richiede, però, di non utilizzare l'ereditarietà, ma esclusivamente la composizione.



Sia dato il seguente diagramma delle classi.



Si prenda in considerazione un sistema costituito unicamente dai tipi riportati nel diagramma. E' possibile affermare che il sistema abbia il massimo grado di coesione possibile?

Scegli una risposta:

☐ Vero

☒ Falso ✓

Si selezionino le affermazione corrette fra le seguenti relative ai *SOLID principles*.

Scegli una o più alternative:

- ☐ a. E' possibile che un gruppo di classi sia *Open-Close* anche se si utilizzano delle variabili globali, nei modi e termini permessi dal linguaggio di programmazione di riferimento.
- ☒ b. Usando il *Liskov Substitution Principle* è associabile al concetto di *Design by Contract* ✓
- ☐ c. Un insieme di classi può essere definito *Open-Close* rispetto a qualsiasi tipo di modifica richiesta.
- ☒ d. Il *Single Responsibility Principle* è anche noto con il nome di coesione ✓

Risposta corretta.

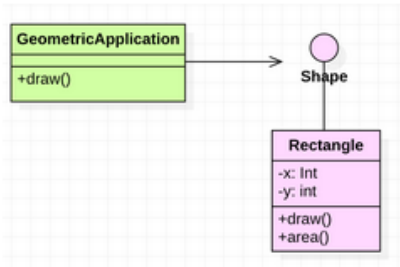
Le risposte corrette sono: Il *Single Responsibility Principle* è anche noto con il nome di coesione., Usando il *Liskov Substitution Principle* è associabile al concetto di *Design by Contract*

Domanda 1

Risposta errata

Contrassegna domanda

Sia data la seguente struttura, utilizzata da un'applicazione che visualizza figure geometriche su uno schermo.



La classe **Rectangle** soddisfa il *Single Responsibility Principle*.

Scegli una risposta:

☐

Vero

☒

Falso ✗

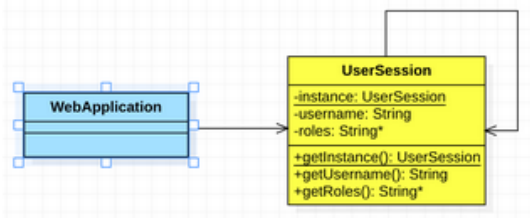
La risposta corretta è "Vero".

Domanda 2

Parzialmente corretta

Contrassegna domanda

Dato il seguente diagramma delle classi, si selezioni quali delle affermazioni sono corrette.



Scegli una o più alternative:

☐a. Le implementazioni del Singleton sono sempre *thread-safe*☒

b. In Java, l'implementazione del Singleton accettata dalla comunità prevede l'uso di una enumerazione ✓

☒c. L'utilizzo del pattern architettura *Dependency Injection* permette di utilizzare una versione più evoluta del pattern Singleton, che ha meno contointerazioni ✓☐

d. L'utilizzo del pattern Singleton della GoF può interagire con la corretta stesura degli Unit Test

☒e. Il diagramma delle classi per la classe **UserSession** implementa il design pattern Singleton classico ✗☐

f. E' corretto usare un Singleton per modellare una session utente di un'applicazione multiutente

Risposta parzialmente esatta.

Hai selezionato correttamente 2.

Le risposte corrette sono: L'utilizzo del pattern Singleton della GoF può interagire con la corretta stesura degli Unit Test, In Java, l'implementazione del Singleton accettata dalla comunità prevede l'uso di una enumerazione, L'utilizzo del pattern architettura *Dependency Injection* permette di utilizzare una versione più evoluta del pattern Singleton, che ha meno contointerazioni

Domanda 3

Completo

Contrassegna domanda

Si elenchino in un apposito diagramma, tutte le possibili **azioni iniziali** previste nei diagramma di attività. Il diagramma non deve essere valido, ma deve essere utilizzato unicamente come un raccoglitore di tali azioni.

Vedi immagine sotto "uml.png"



uml.png

Punti 3/30

Data la classe Singleton che partecipa al design pattern *singleton*, si fornisca una sua implementazione in linguaggio Java, che sia *thread safe* rispetto alla costruzione della singola istanza *instance* della classe.

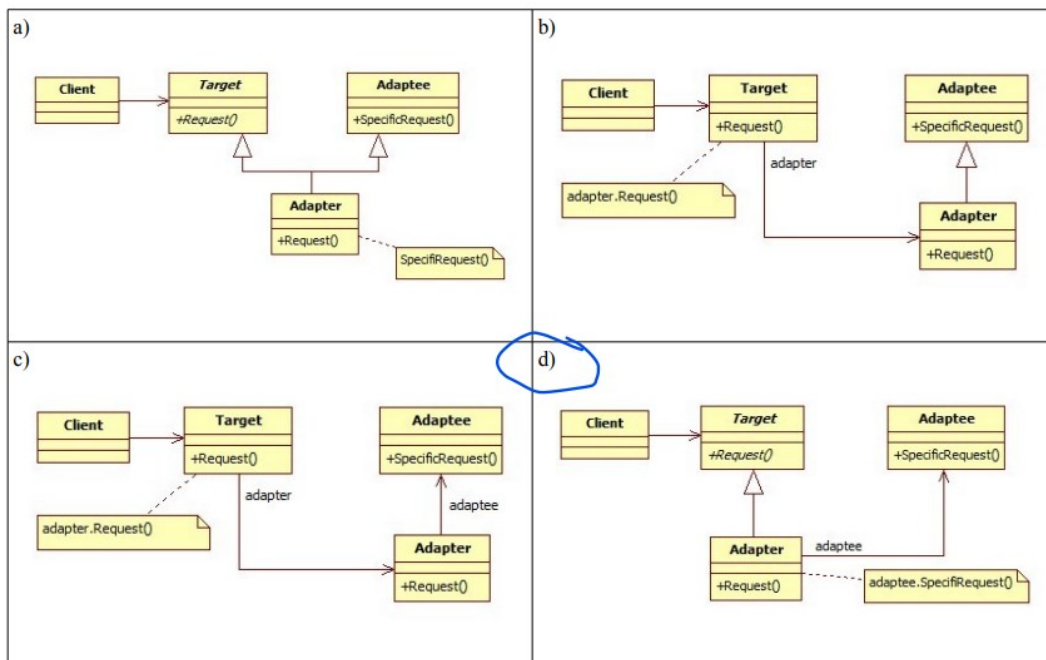
```
public class Singleton {
    private static Singleton instance = null;
    private Singleton() {} // costrutto privato, impedisco l'acce-
                        // solo attraverso getInstance() metodo
                        // sincronizzato
}
```

~~private static~~
~~private~~

```
public static synchronized Singleton getInstance() {
    if (instance == null)
        instance = new Singleton();
    return instance;
}
```

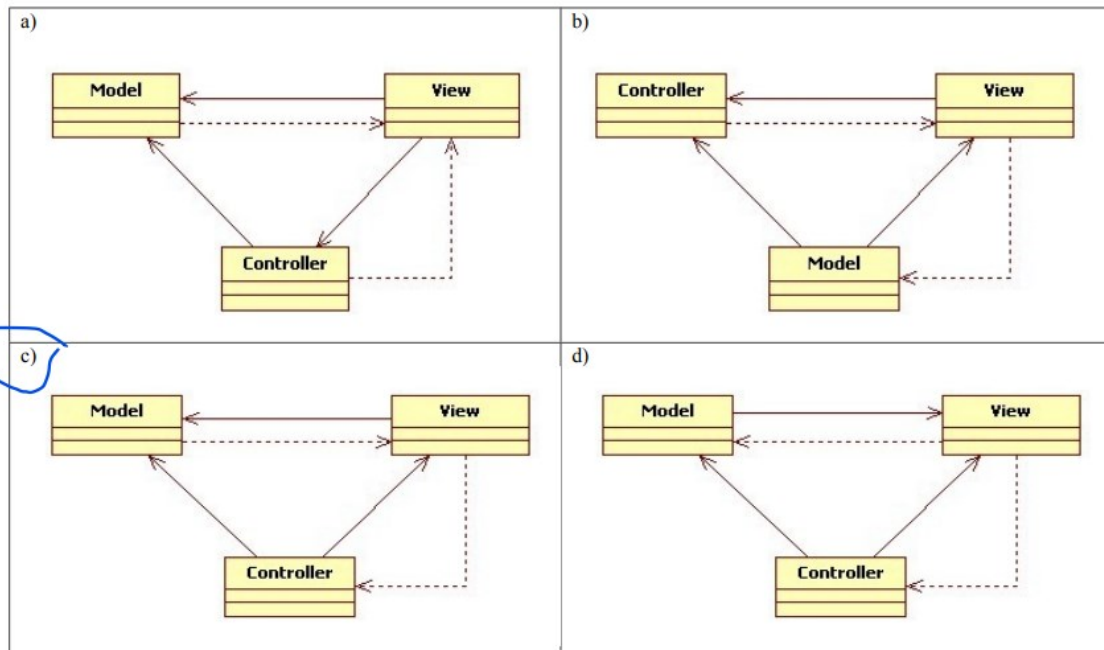
Punti 2/30

Si indichi quale dei diagrammi delle classi sotto riportati modelli correttamente il *design pattern* Adapter nella sua variante *object adapter*.



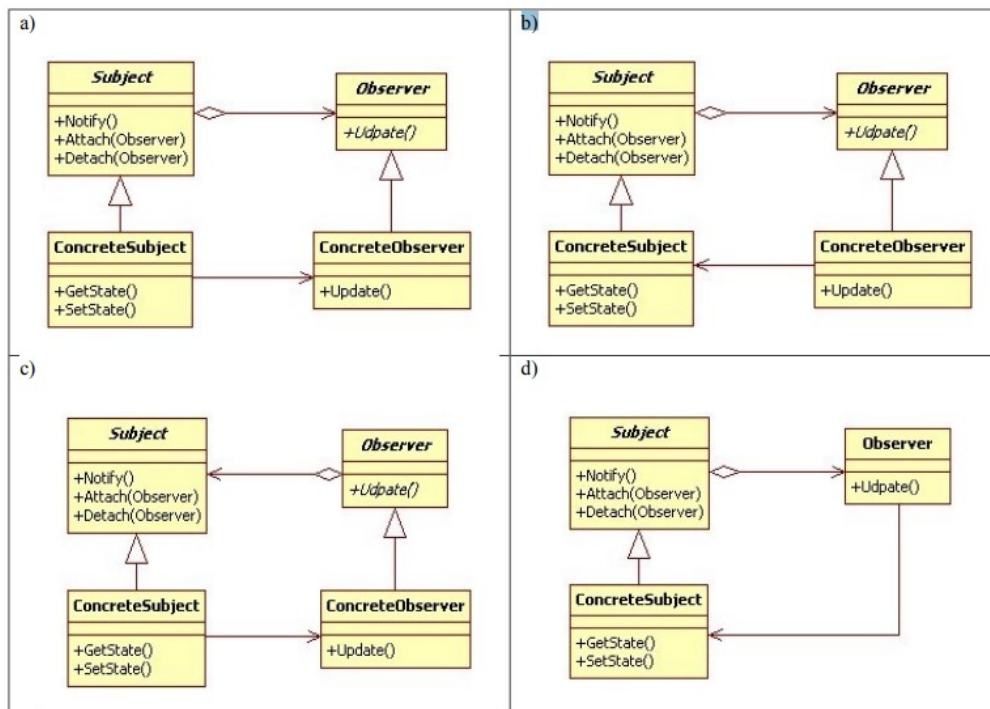
Punti 6/30

Indicare con una X sulla corrispondente lettera identificativa il diagramma delle classi che, fra quelli sotto riportati, rappresenta correttamente il *design pattern* architetturale Model View Controller (MVC).



Punti 2/30

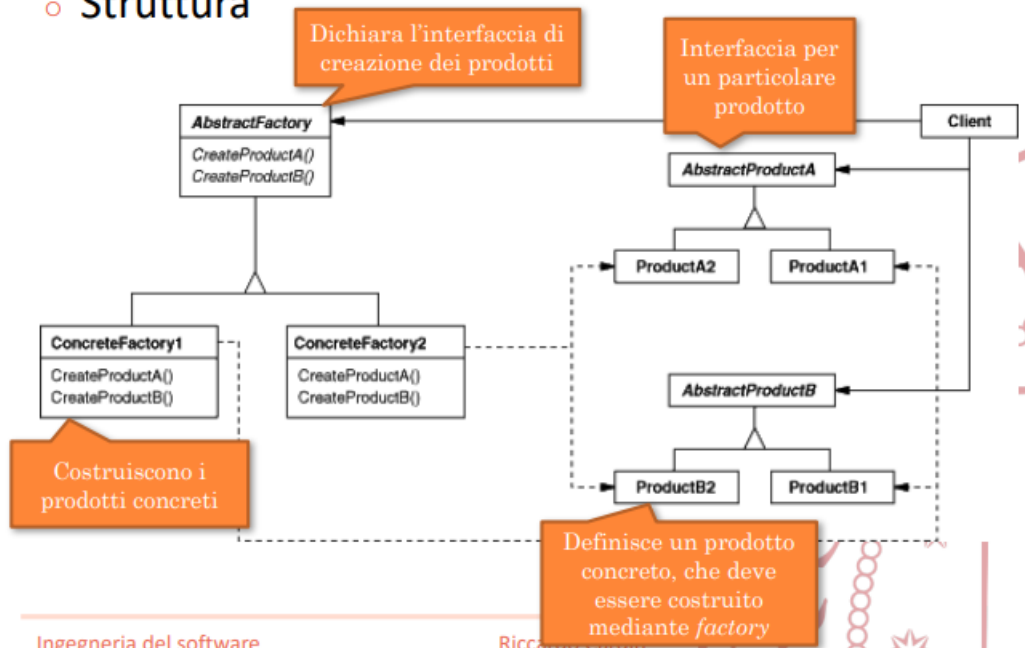
Barrare con una X la lettera del diagramma delle classi che fra i seguenti rappresenta in modo corretto il *design pattern* comportamentale Observer.



Punti 2/30

Si disegni il diagramma delle classi che modella le componenti e le relazioni che intercorrono tra di esse all'interno del design pattern *Abstract Factory* (GoF). Si ponga particolare attenzione alle operazioni definite all'interno di ogni classe.

○ Struttura



Ingegneria del software

Riccardo Garim

Si selezionino le affermazioni **vere** tra le seguenti.

- ☐ a. Un'architettura a microservizi è sempre consigliata per lo sviluppo di **qualsiasi tipo** di applicazione.
- ☒ b. Un microservizio può avere un'architettura logica di tipo *esagonale*. ✓
- ☐ c. È buona norma condividere le informazioni fra microservizi utilizzando delle **tabelle database comuni**.
- ☒ d. La comunicazione fra microservizi dovrebbe essere il più possibile **sincrona**. ✓
- ☒ e. Un microservizio può avere un'architettura logica di tipo *multi-layered*. ✓
- ☐ f. Le architetture a microservizi, se ben progettate, consentono uno *scaling verticale* delle componenti dell'applicazione.

Risposta corretta.

Le risposte corrette sono:

Un microservizio può avere un'architettura logica di tipo *multi-layered*.,

La comunicazione fra microservizi dovrebbe essere il più possibile **sincrona**.,

Un microservizio può avere un'architettura logica di tipo *esagonale*.

Si specifichi se la seguente affermazione è **vera** o **falsa**:

Utilizzando il *design pattern Abstract Factory* risulta immediato aggiungere un **nuovo tipo di prodotto**, mentre è complessa l'aggiunta di una **nuova famiglia di prodotti**.

Scegli una risposta:

- ☐ Vero
- ☒ Falso ✓

La risposta corretta è 'Falso'.

Si selezionino fra le seguenti le affermazioni **vere**.

- ☒ a. Nell'architettura *multi-tier* la *business logic* è isolata dall'*application* e dalla *persistence logic*.
- ☐ b. Un microservizio può essere anche un monolite.
- ☐ c. Un microservizio non può essere implementato utilizzando un'architettura di tipo *multi-tier*.
- ☒ d. Un microservizio rappresenta una unità di *deploy*.
- ☐ e. Un monolite può essere implementato utilizzando un'architettura di tipo esagonale.
- ☒ f. Esistono varie tipologie di architetture, che si differenziano sulla base di quali requisiti non funzionali intendono soddisfare (ad esempio, *deploy*, dipendenze fra le componenti, ecc...).

Una classe si dice **coesa** se ognuno dei suoi utilizzatori utilizza metodi differenti rispetto a quelli utilizzati dagli altri.

Scegli una risposta:

- ☐ Vero
- ☒ Falso

Domanda aperta: Disegnare il grafico del abstract object pattern

Domanda 1

Risposta corretta

Tra le attività in capo al(la) responsabile di progetto sta la redazione del "**consuntivo di periodo**". Tra le attività essenziali dello *sprint*, come inteso nei metodi di sviluppo agile, sta la "**retrospettiva**". La relazione tra tali due concetti risulta spesso oscura agli apprendisti Informatici. L'elenco qui sotto riporta alcune tra le ipotesi più ricorrenti. Indicate quale tra esse sia per voi la più condivisibile.

- ☒ a. Quando lo sviluppo è suddiviso in periodi successivi e la gestione di progetto segue prassi allo stato dell'arte, i due concetti si sovrappongono perfettamente.
- ☐ b. La retrospettiva può essere prassi utile in situazioni critiche, a valle del verificarsi di qualche problema, oppure come innesco a un ciclo di auto-miglioramento. Il consuntivo di periodo, invece, è richiesto a ogni periodo. Ne segue che i due concetti sono disgiunti e non sovrapponibili.
- ☐ c. Tra "consuntivo di periodo" e "retrospettiva" non vi è alcuna relazione. Il primo è un ragionamento esclusivamente contabile; il secondo è un modo per alimentare lo "spirito di gruppo".



Affermativo: questo è quanto stipula la teoria cui facciamo riferimento e anche la pratica che abbiamo messo in atto.

Risposta corretta.

La risposta corretta è:

Quando lo sviluppo è suddiviso in periodi successivi e la gestione di progetto segue prassi allo stato dell'arte, i due concetti si sovrappongono perfettamente.

Domanda 2

Risposta corretta

Tra aspiranti Informatici sorge una discussione accesa sul rapporto tra **ora di orologio** e **ora produttiva**. L'elenco qui sotto riporta alcune tra le ipotesi più ricorrenti in tale discussione. Indicate quale tra esse sia per voi la più condivisibile.

- ☐ a. Esiste solo l'ora di orologio, che è misura oggettiva e indipendente dall'individuo. L'ora produttiva è un concetto arbitrario, soggettivo, e non misurabile.
- ☐ b. I due concetti esistono, ma, dove il primo è oggettivo e indipendente dall'individuo, il secondo è fortemente variabile e di difficile uso. Pertanto, affidarsi al secondo è esercizio pieno di rischi, mentre usare come riferimento il primo fornisce certezze contabili.
- ☒ c. La "taglia" di un progetto, per lunghezza temporale e costi economici, è determinata in sede di preventivo dal numero di ore produttive stimate necessarie per la realizzazione di quanto richiesto. Lo svolgimento di ogni singola attività di progetto consuma simultaneamente ore di orologio (nella misura del tempo personale) e ore produttive (in funzione del tasso di raggiungimento degli obiettivi di tale attività). La direzione del rapporto tra tali due quantità rispetto al punto di perfetto equilibrio dice la pressione sulla persona (se maggiore di 1) o il margine utile del fornitore (se minore di 1).

✔ **Affermativo:**
questo è quanto
stipula la teoria cui
facciamo
riferimento e
anche la pratica
che abbiamo
messo in atto.

Risposta corretta.

La risposta corretta è:

La "taglia" di un progetto, per lunghezza temporale e costi economici, è determinata in sede di preventivo dal numero di ore produttive stimate necessarie per la realizzazione di quanto richiesto. Lo svolgimento di ogni singola attività di progetto consuma simultaneamente ore di orologio (nella misura del tempo personale) e ore produttive (in funzione del tasso di raggiungimento degli obiettivi di tale attività). La direzione del rapporto tra tali due quantità rispetto al punto di perfetto equilibrio dice la pressione sulla persona (se maggiore di 1) o il margine utile del fornitore (se minore di 1).

Domanda 1

Risposta
corretta



Contrassegna
domanda

Un'altra fonte di frequente equivoco tra apprendisti informatici è l'interpretazione della relazione intercorrente tra le nozioni di **progetto** e di **processi di ciclo di vita del software**. L'elenco qui sotto riporta alcune tra le ipotesi più ricorrenti. Indicate quale tra esse sia per voi la più condivisibile.

- ☒ a. Un progetto si compone di attività provenienti da vari processi, la cui specifica costituisce il *way of working* del fornitore. ✓

Corretto: un progetto è un insieme di attività pianificate per il raggiungimento di un insieme di svariati obiettivi – caratteristiche di prodotto, documentazione associata – da raggiungere entro determinati vincoli di costi e di tempi, secondo richieste fissate da un committente; un processo invece raccoglie in se attività con un singolo, specifico obiettivo e dà loro ordine, regole, e struttura.

- ☐ b. Progetto e processi non hanno alcuna relazione tra loro: il primo (progetto) non ha bisogno dei secondi; similmente, i secondi (processi), se mai esistano, lo fanno al di fuori dei confini di un progetto.
- ☐ c. L'unica relazione possibile è intorno al concetto di "sviluppo". Se esiste un processo "sviluppo", allora quel singolo processo ingloba un intero progetto, che – appunto – sviluppa un dato prodotto.

Risposta corretta.

La risposta corretta è:

Un progetto si compone di attività provenienti da vari processi, la cui specifica costituisce il *way of working* del fornitore.

Domanda 2Risposta
correttaContrassegna
domanda

Parliamo di temi e ambiti di SWE; ciò fissa il contesto della domanda e delle risposte proposte.

Tra apprendisti informatici, nascono sovente dubbi interpretativi sulla relazione intercorrente tra i concetti di *milestone* e *baseline* nel dominio IS. L'elenco qui sotto riporta alcune tra le ipotesi più ricorrenti. Indicate quale tra esse sia per voi la più condivisibile.

- ☐ a. I due concetti sono del tutto disgiunti e indipendenti: la *milestone* attiene alla pianificazione di progetto, e sostanzialmente corrisponde a una data di calendario, alla quale corrispondono certe attese. La *baseline* descrive il contenuto di un *repository* soggetto a controllo di versione e di configurazione, a un certo istante temporale. In certe situazioni, il primo può riferire al secondo, ma non sempre e non necessariamente
- ☒ b. I due concetti sono distinti, ma strettamente correlati e complementari nel significato. La *milestone* fissa una particolare punto (data) nel calendario di progetto, al quale associa specifiche attese di avanzamento (progresso atteso). La *baseline* costituisce l'evidenza tecnica di uno specifico punto di avanzamento, costituita dalla parti che la compongono secondo determinate regole di composizione. ✔
- ☐ c. I due concetti sono sinonimi uno dell'altro. Entrambi denotano un particolare punto nel tempo di progetto, fissato dal contratto con il committente.

Affermativo:

questo è quanto stipula la teoria cui facciamo riferimento e anche la pratica che abbiamo messo in atto.

Risposta corretta.

La risposta corretta è:

I due concetti sono distinti, ma strettamente correlati e complementari nel significato. La *milestone* fissa una particolare punto (data) nel calendario di progetto, al quale associa

Domanda 1Risposta
salvataContrassegna
domanda

Si selezionino fra le seguenti le affermazioni corrette relative al principio *Open-Close*.

- ☒ a. Una componente che utilizza variabili con *scope* globale può essere chiusa al cambiamento.
- ☐ b. La *Run Time Type Identification* (RTTI) è alla base del principio *Open-Close*.
- ☒ c. Solitamente, la chiusura al cambiamento viene ottenuta attraverso l'utilizzo di astrazioni e polimorfismo.
- ☐ d. Solitamente, una componente chiusa al cambiamento non è affetta da "cambiamenti a cascata".
- ☐ e. Un'insieme di classi può essere chiuso al 100% dei cambiamenti.
- ☒ f. L'*encapsulation* abilita una componente ad essere chiusa al cambiamento.

[Pagina successiva](#)

Domanda 2: "Si descrivano testualmente le caratteristiche peculiari della relazione di composizione fra due classi."

Domanda **3**

Risposta non ancora data



Contrassegna domanda

Una delle *best practice* nell'applicazione di un'architettura a microservizi è quella di utilizzare un database con delle **tabelle condivise** per lo scambio di informazioni fra diversi microservizi.

Scegli una risposta:

☐ Vero

☒ Falso