

**Domanda 1**Parzialmente  
correttaPunteggio  
ottenuto 1,60 su  
2,00Contrassegna  
domanda

Dire quali delle seguenti affermazioni sono vere e quali false.

Un caso d'uso rappresenta una maniera di utilizzare il sistema da parte di un utente per raggiungere un suo obiettivo.

Vero



I requisiti funzionali non catturati dai casi d'uso vengono descritti nelle Specifiche Supplementari.

Falso



Un caso d'uso rappresenta l'insieme di funzionalità di un sistema.

Vero



I requisiti non funzionali sono descritti completamente dai casi d'uso.

Falso

In UP, la disciplina dei requisiti ha l'obiettivo di produrre una lista dei requisiti, capire il contesto del sistema, catturare i requisiti funzionali e i requisiti non-funzionali.

Vero

**Domanda 2**

Risposta corretta

Punteggio  
ottenuto 2,00 su  
2,00Contrassegna  
domanda

Dire quali delle seguenti affermazioni sono vere e quali false.

Il modello di dominio include la definizione di oggetti, associazioni e attributi di classi software.

Falso



L'analisi linguistica dei casi d'uso nel formato dettagliato è una fonte di ispirazione per la costruzione del modello di dominio.

Vero



Il modello di dominio è un dizionario visuale delle classi concettuali.

Vero



Il modello di dominio riporta i concetti significativi relativi ai casi d'uso.

Vero



IL modello di dominio è una rappresentazione testuale delle classi concettuali, oggetti reali del dominio.

Falso

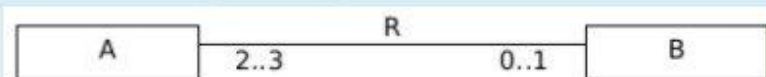


**Domanda 3**

Risposta corretta

Punteggio  
ottenuto 2,50 su  
2,50Contrassegna  
domanda

Si consideri il diagramma seguente:

Si supponga  $A = \{a1, a2, a3, a4\}$  e  $B = \{b1, b2\}$ .

Dire quali delle seguenti affermazioni sono vere e quali false.

R può essere  $\{(a1, b1), (a2, b2), (a3, b1), (a4, b2)\}$ . Vero ✓R può essere  $\{(a1, b1), (a2, b1), (a3, b1), (a4, b2)\}$ . Falso ✓R può essere  $\{(a1, b1), (a2, b2)\}$ . Falso ✓R può essere  $\{(a1, b1), (a2, b2), (a2, b2), (a3, b2)\}$ . Falso ✓**Domanda 4**

Risposta corretta

Punteggio  
ottenuto 2,00 su  
2,00Contrassegna  
domanda

Si considerino i seguenti problemi di progettazione:

1. Come creare famiglie di classi correlate che implementano un'interfaccia comune?
2. Come trattare un gruppo o una struttura composta di oggetti (polimorficamente) dello stesso tipo nello stesso modo di un oggetto non composto (atomico)?
3. Il comportamento di un oggetto dipende da suo stato e i suoi metodi contengono logica condizionale per casi che riflette le azioni condizionali che dipendono dallo stato. C'è un'alternativa alla logica condizionale?

Associare la corrispondente soluzione progettuale.

Crea delle classi stato per ciascuno stato, che implementano un'interfaccia comune. Delega le operazioni che dipendono dallo stato dall'oggetto contesto all'oggetto stato corrente corrispondente. Assicura che l'oggetto contesto referenzi sempre un oggetto stato che riflette il suo stato corrente.

Definisci ciascun algoritmo in una classe separata, con un'interfaccia comune.

Definire un'interfaccia factory (la factory astratta). Definire una classe factory concreta per ciascuna famiglia di elementi da creare. Opzionalmente, definire una vera classe astratta che implementa l'interfaccia factory e fornisce servizi comuni alle factory concrete che la estendono.

Definisci le classi per gli oggetti composti e atomici in modo che implementino la stessa interfaccia

Problema (3) ▼



Non è presente il problema corrispondente ▼



Problema (1) ▼



Problema (2) ▼





**Domanda 5**Parzialmente  
correttaPunteggio  
ottenuto 2,40 su  
3,00Contrassegna  
domanda

Dire quali delle seguenti affermazioni sono vere e quali false.

I pattern GoF incentivano i meccanismi di riuso del software attraverso la definizione i gerarchie di classi.

Falso



La composizione di oggetti è definita staticamente attraverso la specifica delle classi e delle associazioni.

Falso



Il meccanismo di specializzazione per il riuso del codice è detto white-box.

Vero



La modifica dell'interfaccia di una classe ha in generale un basso impatto nelle sottoclassi.

Vero



I pattern GoF prediligono l'utilizzo del meccanismo di ereditarietà per ottenere la specializzazione.

Falso

**Domanda 6**Parzialmente  
correttaPunteggio  
ottenuto 2,00 su  
2,50Contrassegna  
domanda

Dire quali delle seguenti affermazioni sono vere e quali false.

Un metodo di una classe software rappresenta una responsabilità "a fare" per le istanze di tale classe.

Vero



I pattern GRASP sono espressi in termini di responsabilità, ruoli e collaborazioni.

Vero



La decisione sull'assegnazione delle responsabilità precede sempre la codifica.

Vero



L'utilizzo dei pattern GRASP è svolto all'interno della disciplina dei requisiti di UP.

Falso



Le responsabilità sono assegnate alle classi durante la OOD.

Vero



**Domanda 1**

Risposta corretta

Punteggio  
ottenuto 2,00 su  
2,00



Contrassegna  
domanda

Dire quali delle seguenti affermazioni sono vere e quali false.

L'analisi linguistica dei casi d'uso nel formato dettagliato è una fonte di ispirazione per la costruzione del modello di dominio.

Vero ☐ ✓

Il modello di dominio riporta i concetti significativi relativi ai casi d'uso.

Vero ☐ ✓

IL modello di dominio è una rappresentazione testuale delle classi concettuali, oggetti reali del dominio.

Falso ☐ ✓

Il modello di dominio è un dizionario visuale delle classi concettuali.

Vero ☐ ✓

Il modello di dominio include la definizioe di oggetti, associazioni e attributi di classi software.

Falso ☐ ✓

**Domanda 2**

Risposta corretta

Punteggio  
ottenuto 3,00 su  
3,00



Contrassegna  
domanda

Dire quali delle seguenti affermazioni sono vere e quali false.

Il meccanismo di delega è preferibile al meccanismo di specializzazione per il riuso del codice.

Vero ☐ ✓

I pattern GoF prediligono l'utilizzo del meccanismo di ereditarietà per ottenere il polimorfismo.

Vero ☐ ✓

L'ereditarietà rispetta l'incapsulamento.

Falso ☐ ✓

I pattern GoF incentivano l'uso dell'ereditarietà come meccanismo di riuso del codice, in particolare attraverso il pattern composite.

Falso ☐ ✓

La composizione di oggetti è un meccanismo di riuso del codice detto black-box.

Vero ☐ ✓

**Domanda 3**

Risposta corretta

Punteggio  
ottenuto 2,50 su  
2,50



Contrassegna  
domanda

Dire quali delle seguenti affermazioni sono vere e quali false.

Le responsabilità sono assegnate alle classi durante la OOD.

Vero ☐ ✓

I pattern GRASP sono espressi in termini di responsabilità, ruoli e collaborazioni.

Vero ☐ ✓

Un metodo di una classe software rappresenta una responsabilità "a fare" per le istanze di tale classe.

Vero ☐ ✓

La decisione sull'assegnazione delle responsabilità precede sempre la codifica.

Falso ☐ ✓

L'utilizzo dei pattern GRASP è svolto all'interno della disciplina dei requisiti di UP.

Falso ☐ ✓



Si considerino i seguenti problemi di progettazione:

1. Come creare famiglie di classi correlate che implementano un'interfaccia comune?
2. Come trattare un gruppo o una struttura composta di oggetti (polimorficamente) dello stesso tipo nello stesso modo di un oggetto non composto (atomico)?
3. Il comportamento di un oggetto dipende da suo stato e i suoi metodi contengono logica condizionale per casi che riflette le azioni condizionali che dipendono dallo stato. C'è un'alternativa alla logica condizionale?

Associare la corrispondente soluzione progettuale.

Definisci le classi per gli oggetti composti e atomici in modo che implementino la stessa interfaccia

Definisci ciascun algoritmo in una classe separata, con un'interfaccia comune.

Definire un'interfaccia factory (la factory astratta). Definire una classe factory concreta per ciascuna famiglia di elementi da creare. Opzionalmente, definire una vera classe astratta che implementa l'interfaccia factory e fornisce servizi comuni alle factory concrete che la estendono.

Crea delle classi stato per ciascuno stato, che implementano un'interfaccia comune. Delega le operazioni che dipendono dallo stato dall'oggetto contesto all'oggetto stato corrente corrispondente. Assicura che l'oggetto contesto referenzi sempre un oggetto stato che riflette il suo stato corrente.

Problema (2) ▾



Non è presente il problema corrispondente ▾



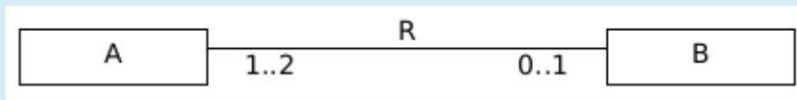
Problema (1) ▾



Problema (3) ▾



Si consideri il diagramma seguente:



Si supponga  $A = \{a1, a2, a3, a4\}$  e  $B = \{b1, b2\}$ .

Dire quali delle seguenti affermazioni sono vere e quali false.

R può essere  $\{(a1, b1), (a2, b1)\}$ .

Falso ▾



R può essere  $\{(a1, b1), (a2, b2), (a3, b1), (a4, b2)\}$ .

Vero ▾



R può essere  $\{(a2, b1), (a2, b2), (a3, b2)\}$ .

Falso ▾



R può essere  $\{(a1, b1), (a2, b2)\}$ .

Vero ▾



Dire quali delle seguenti affermazioni sono vere e quali false.

In UP i casi d'uso vengono utilizzati per catturare i requisiti funzionali nella disciplina dei requisiti.

Vero



I casi d'uso sononcaratteristiche del sistema.

Falso



I casi d'uso sono utilizzati per pianificare le iterazioni.

Vero



In UP i casi d'uso sono descritti mediante il linguaggio UML.

Falso

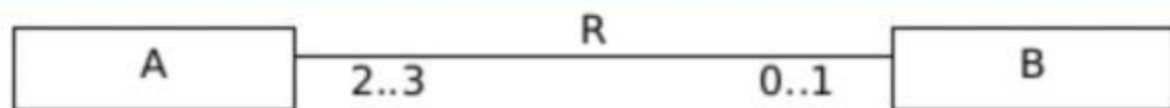


In modello di dominio non è parte della disciplina dei requisiti.

Vero



Si consideri il diagramma seguente:



Si supponga  $A = \{a1, a2, a3, a4\}$  e  $B = \{b1, b2\}$ .

Dire quali delle seguenti affermazioni sono vere e quali false.

R può essere  $\{(a1, b1), (a2, b2), (a3, b1), (a4, b2)\}$ .

Falso



R può essere  $\{(a1, b1), (a2, b1), (a3, b1), (a4, b2)\}$ .

Falso



R può essere  $\{(a1, b1), (a2, b2), (a2, b2), (a3, b2)\}$ .

Vero



R può essere  $\{(a1, b1), (a2, b2)\}$ .

Vero



#### Domanda 4

Risposta salvata

Punteggio max.:  
2,50



Contrassegna  
domanda

Dire quali delle seguenti affermazioni sono vere e quali false.

La decisione sull'assegnazione delle responsabilità è svolta attraverso i soli diagrammi di classe.

Falso



Un attributo di una classe software rappresenta una responsabilità "di conoscenza" per le istanze di tale classe.

Falso



I pattern GRASP sono alla base dello sviluppo guidato dalle responsabilità.

Vero



Un progetto orientato agli oggetti vede un sistema come una comunità di oggetti con responsabilità che collaborano.

Vero



L'utilizzo di pattern GRASP è finalizzato alla scoperta dei requisiti del sistema.

Vero






**Domanda 3**Parzialmente  
correttaPunteggio  
ottenuto  
1,60 su  
2,00Contrassegna  
domanda

Dire se le seguenti affermazioni sulle associazioni utilizzate nel modello di dominio sono vere o false.

La direzione di lettura va sempre specificata.

Vero  

E' per sua natura unidirezionale

Falso  

Rappresenta una relazione significativa tra classi

Vero  

Rappresenta un valore logico degli oggetti di una classe

Falso  

Rappresenta un'insieme di n-tuple di oggetti delle classi

Vero  

La risposta corretta è: La direzione di lettura va sempre specificata. – Falso, E' per sua natura unidirezionale – Falso, Rappresenta una relazione significativa tra classi – Vero, Rappresenta un valore logico degli oggetti di una classe – Falso, Rappresenta un'insieme di n-tuple di oggetti delle classi – Vero

**Domanda 1**

Risposta  
corretta

Punteggio  
ottenuto  
2,00 su  
2,00



Contrassegna  
domanda

Durante la fase di elaborazione

Non vengono effettuati test al codice sviluppato

Falso ▼



Non si sviluppa codice

Falso ▼



Si scrivono circa il 10% dei casi d'uso tra i più critici in formato dettagliato utilizzando template appositi (caso d'uso dettagliato e strutturato)

Falso ▼



Viene scoperta e stabilizzata la maggior parte dei requisiti

Vero ▼



Si realizza uno studio economico per stabilire l'ordine di grandezza del progetto e dei costi

Falso ▼



Viene programmato il nucleo, rischioso, dell'architettura

Vero ▼



La risposta corretta è: Non vengono effettuati test al codice sviluppato – Falso, Non si sviluppa codice – Falso, Si scrivono circa il 10% dei casi d'uso tra i più critici in formato dettagliato utilizzando template appositi (caso d'uso dettagliato e strutturato) – Falso, Viene scoperta e stabilizzata la maggior parte dei requisiti – Vero, Si realizza uno studio economico per stabilire l'ordine di grandezza del progetto e dei costi – Falso, Viene programmato il nucleo, rischioso, dell'architettura – Vero

**Domanda 2**Parzialmente  
correttaPunteggio  
ottenuto  
1,50 su  
2,00Contrassegna  
domanda

Dire quali delle seguenti affermazioni sono vere e quali false.

In UML per responsabilità si intende la specifica di un metodo associato ad una classe Java

Vero ▾



In UML per responsabilità si intende la specifica di una variabile di istanza di una classe Java

Falso ▾

In UML per responsabilità si intende un contratto o un obbligo di un classificatore

Vero ▾

In UML per responsabilità si intende la specifica di una variabile di istanza o di un metodo associato ad una classe Java

Falso ▾




La risposta corretta è: In UML per responsabilità si intende la specifica di un metodo associato ad una classe Java – Falso, In UML per responsabilità si intende la specifica di una variabile di istanza di una classe Java – Falso, In UML per responsabilità si intende un contratto o un obbligo di un classificatore – Vero, In UML per responsabilità si intende la specifica di una variabile di istanza o di un metodo associato ad una classe Java – Falso




2.00

  
Contrassegna  
domanda


Il refactoring è una pratica promossa dal metodo iterativo e agile XP

  
Vero ▾

Il refactoring prevede lo sviluppo guidato dai test, ovvero uno sviluppo preceduto dai test

  
Vero ▾

Il refactoring è un metodo strutturato e disciplinato per scrivere o ristrutturare del codice esistente

  
Falso ▾