

### Domanda 1

Dire se le seguenti affermazioni sono vere o false.

Risposta non ancora  
data

Punteggio max.: 2,00

Contrassegna  
domanda

Le pre-condizioni sono ipotesi significative dello stato del sistema o degli oggetti del modello di progetto prima dell'esecuzione dell'operazione a cui è associata

F

Vera

La post-condizione descrive i cambiamenti di stato degli oggetti nel Modello di Dominio dopo il completamento dell'operazione

V

Vera

Le post-condizioni descrivono i cambiamenti nello stato degli oggetti del modello di progetto. I cambiamenti dello stato del modello di progetto comprendono gli oggetti creati, i collegamenti formati o rotti, e gli attributi modificati

F

Falsa

### Domanda 2

Dire se le seguenti affermazioni sono vere o false.

Risposta non ancora  
data

Punteggio max.: 2,00

Contrassegna  
domanda

La narrativa di un caso d'uso viene espressa a livello delle azioni concrete dell'utente e delle responsabilità del sistema

Falso

Un caso d'uso è sempre completato in una sola iterazione

Falso

La narrativa di un caso d'uso viene espressa a livello delle intenzioni dell'utente e delle responsabilità del sistema

Vero

V

### Domanda 3

Dire se le seguenti affermazioni sono vere o false.

Risposta non ancora  
data

Punteggio max.: 2,00

Contrassegna  
domanda

Un modello di dominio è la rappresentazione visuale degli oggetti software

F

Falso

La disciplina dei requisiti è parte della fasi di Ideazione

V

Vero

La programmazione di qualità-produzione per un sottoinsieme dei requisiti si inizia dopo che l'analisi di tutti i requisiti sia stata completata

F

Falso

**Domanda 4**

Risposta non ancora

data

Punteggio max: 2,00

 Contrassegna

domanda

Dire se le seguenti affermazioni sono vere e false.

Una sottoclasse è fortemente accoppiata alla sua superclasse

Vero

Ricevere un parametro di tipo A per un metodo di una classe B rappresenta una relazione di dipendenza da B a A mentre estendere una classe A per una classe B non rappresenta una relazione di dipendenza da B a A.

Falso

E' possibile che due classi siano collegate da più di una associazione

Vero

**Domanda 1**

Parzialmente corretta  
Punteggio ottenuto 0,40 su 2,00



Quali delle seguenti affermazioni relative agli SSD sono vere e quali sono false?

Sono espressi attraverso i diagrammi di comunicazione di UML.

|       |                                     |
|-------|-------------------------------------|
| Vero  | <input checked="" type="checkbox"/> |
| Falso | <input checked="" type="checkbox"/> |

Mostrano l'ordine degli eventi generati dagli attori esterni al sistema.

I contratti costituiscono un input per gli SSD delle operazioni e per la progettazione degli oggetti.

Un evento di sistema è un evento interno al sistema.

Mostrano gli eventi generati dagli attori esterni al sistema.

La risposta corretta è: Sono espressi attraverso i diagrammi di comunicazione di UML. – Falso, Mostrano l'ordine degli eventi generati dagli attori esterni al sistema. – Vero, I contratti costituiscono un input per gli SSD delle operazioni e per la progettazione degli oggetti. – Falso, Un evento di sistema è un evento interno al sistema. – Falso, Mostrano gli eventi generati dagli attori esterni al sistema. – Vero

**Domanda 2**

Parzialmente corretta  
Punteggio ottenuto 0,67 su 2,00



Associare le corrette percentuali.

In percentuale sul totale di casi d'uso complessivi di un progetto, quanti sono quelli che dovrebbero essere realizzati entro la conclusione della fase di ideazione?

|                     |                                     |
|---------------------|-------------------------------------|
| 5%                  | <input checked="" type="checkbox"/> |
| Quasi il 100%       | <input checked="" type="checkbox"/> |
| Tra il 50% e il 70% | <input checked="" type="checkbox"/> |

In percentuale sul totale dei requisiti funzionali di un progetto, quanti sono quelli che dovrebbero essere identificati entro la conclusione della fase di ideazione?

In percentuale sul totale dei requisiti funzionali di un progetto, quanti sono quelli che dovrebbero essere identificati entro la conclusione della fase di elaborazione?

La risposta corretta è: In percentuale sul totale di casi d'uso complessivi di un progetto, quanti sono quelli che dovrebbero essere realizzati entro la conclusione della fase di ideazione? – 5%, In percentuale sul totale dei requisiti funzionali di un progetto, quanti sono quelli che dovrebbero essere identificati entro la conclusione della fase di ideazione? – Tra il 50% e il 70%, In percentuale sul totale dei requisiti funzionali di un progetto, quanti sono quelli che dovrebbero essere identificati entro la conclusione della fase di elaborazione? – Quasi il 100%

**Domanda 3**

Risposta corretta  
Punteggio ottenuto 2,00 su 2,00



Quali delle seguenti affermazioni sono vere e quali sono false?

I casi d'uso mettono in risalto gli obiettivi degli utenti.

|      |                                     |
|------|-------------------------------------|
| Vero | <input checked="" type="checkbox"/> |
|------|-------------------------------------|

I casi d'uso sono utilizzati per la scoperta e la definizione dei requisiti non funzionali.

|       |                                     |
|-------|-------------------------------------|
| Falso | <input checked="" type="checkbox"/> |
|-------|-------------------------------------|

I casi d'uso sono utilizzati per la scoperta e la definizione dei requisiti funzionali.

|      |                                     |
|------|-------------------------------------|
| Vero | <input checked="" type="checkbox"/> |
|------|-------------------------------------|

I casi d'uso mettono in risalto gli obiettivi del sistema.

|       |                                     |
|-------|-------------------------------------|
| Falso | <input checked="" type="checkbox"/> |
|-------|-------------------------------------|

I casi d'uso sono utilizzati solo nelle prime iterazioni di sviluppo.

|       |                                     |
|-------|-------------------------------------|
| Falso | <input checked="" type="checkbox"/> |
|-------|-------------------------------------|

La risposta corretta è: I casi d'uso mettono in risalto gli obiettivi degli utenti. – Vero, I casi d'uso sono utilizzati per la scoperta e la definizione dei requisiti non funzionali. – Falso, I casi d'uso sono utilizzati per la scoperta e la definizione dei requisiti funzionali. – Vero, I casi d'uso mettono in risalto gli obiettivi del sistema. – Falso, I casi d'uso sono utilizzati solo nelle prime iterazioni di sviluppo. – Falso

**Domanda 4**

Parzialmente corretta  
Punteggio ottenuto 1,67 su 2,00



Quale delle seguenti affermazioni sul modello di dominio sono vere e quali false?

Insieme di diagrammi di classi UML che includono associazioni tra classi software.

|       |                                     |
|-------|-------------------------------------|
| Falso | <input checked="" type="checkbox"/> |
|-------|-------------------------------------|

Rappresentazione grafica degli oggetti software.

|       |                                     |
|-------|-------------------------------------|
| Falso | <input checked="" type="checkbox"/> |
|-------|-------------------------------------|

Rappresentazione visuale delle classi concettuali.

|      |                                     |
|------|-------------------------------------|
| Vero | <input checked="" type="checkbox"/> |
|------|-------------------------------------|

Insieme di diagrammi di classi UML che includono associazioni tra classi concettuali.

|      |                                     |
|------|-------------------------------------|
| Vero | <input checked="" type="checkbox"/> |
|------|-------------------------------------|

Rappresentazione visuale delle classi Java.

|      |                                     |
|------|-------------------------------------|
| Vero | <input checked="" type="checkbox"/> |
|------|-------------------------------------|

Insieme di diagrammi di classi UML che includono le responsabilità di fare.

|       |                                     |
|-------|-------------------------------------|
| Falso | <input checked="" type="checkbox"/> |
|-------|-------------------------------------|

La risposta corretta è: Insieme di diagrammi di classi UML che includono associazioni tra classi software. – Falso, Rappresentazione grafica degli oggetti software. – Falso, Rappresentazione visuale delle classi concettuali. – Vero, Insieme di diagrammi di classi UML che includono associazioni tra classi concettuali. – Vero, Rappresentazione visuale delle classi Java. – Falso, Insieme di diagrammi di classi UML che includono le responsabilità di fare. – Falso

a 3

Durante la fase di elaborazione.

F

I requisiti e le iterazioni sono organizzate in base alle richieste dell'utente finale.

Si effettua attività di programmazione di qualità-produzione e test.

I requisiti e le interazioni sono organizzate in base al rischio, coperture e criticità.

Vengono realizzati prototipi "usa e getta" per attenuare i rischi maggiori. F

a 4

Quale pattern GRASP è corretto utilizzare per i seguenti problemi?

creator

Chi crea un oggetto A? Ovvero, chi deve essere responsabile della creazione di una nuova istanza di una classe?

controller

Qual è il primo oggetto oltre lo strato UI che riceve e coordina un'operazione di sistema?

Come mantenere gli oggetti focalizzati, comprensibili e gestibili e, come effetto collaterale, sostenere Low Coupling? High Cohesions

Expert

Qual è un principio di base, generale, per l'assegnazione di responsabilità agli oggetti?

Come ridurre l'impatto dei cambiamenti? Come sostenere una dipendenza bassa, un impatto dei cambiamenti basso e una maggiore opportunità di riuso? low coupling

Domanda 1

Risposta non  
ancora data

Punteggio max.:

3,00



Contrassegna  
domanda

"Un catalogo di prodotti è costituito da più descrizioni di prodotto. Ogni descrizione di prodotto è costituita da un identificatore, una descrizione e un prezzo. Ogni catalogo di prodotti è utilizzato da alcuni negozi, identificati da un nome e un indirizzo." Disegnare il modello di dominio (utilizzando UML).



Domanda 2

Risposta non ancora data

Punteggio max.:

3,00



Contrassegna domanda

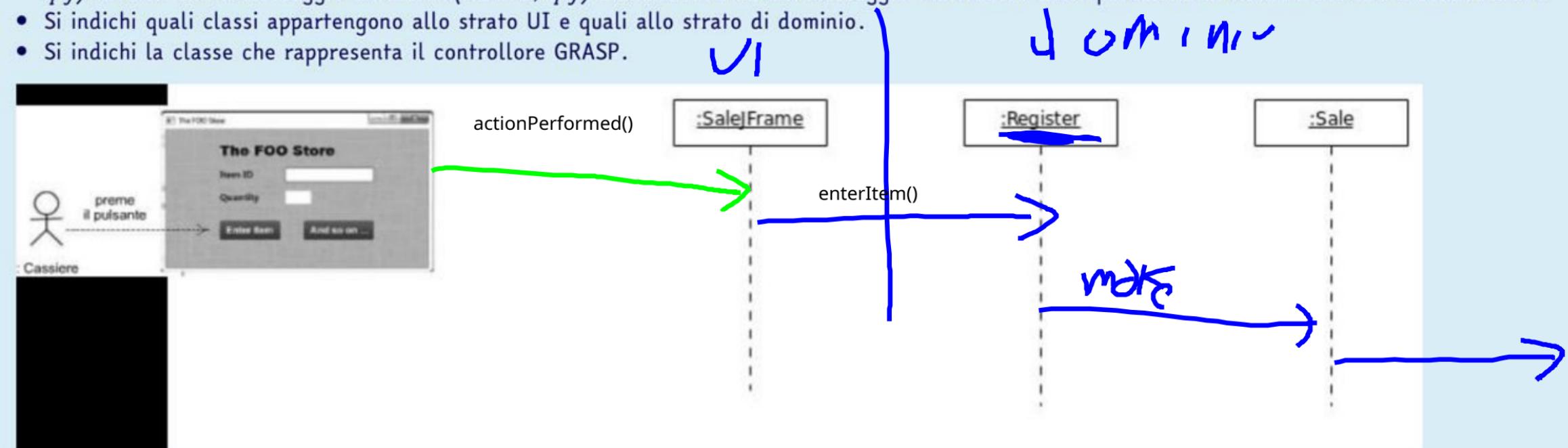
V Dire a quale pattern GoF la seguente implementazione della struttura dati albero binario si conforma. Disegnare la struttura del pattern e il diagramma delle classi della soluzione applicata alla struttura dati albero binario.

```
public abstract class Tree {  
    public abstract boolean empty();  
    public abstract int getRootElement();  
    public abstract void stampapostvisita();  
}  
  
public class Leaf extends Tree {  
    public Leaf() {}  
    public boolean empty() {  
        return true;  
    }  
    public int getRootElement() {  
        assert false; return 0;  
    }  
    public void stampapostvisita() {}  
}  
  
public class Branch extends Tree {  
    private int elem;  
    private Tree left;  
    private Tree right;  
    public Branch(int elem, Tree left, Tree right) {  
        this.elem = elem;  
        this.left = left;  
        this.right = right;  
    }  
    public boolean empty() {  
        return false;  
    }  
    public int getRootElement() {  
        return elem;  
    }  
    public void stampapostvisita() {  
        right.stampapostvisita();  
        left.stampapostvisita();  
        System.out.print(this.getRootElement() + " ");  
    }  
}  
  
public class TestTree {  
    public static void main(String[] args) {  
        ...  
        System.out.println("Stampo albero postvisita");  
        t.stampapostvisita();  
        System.out.println("");  
    }  
}
```



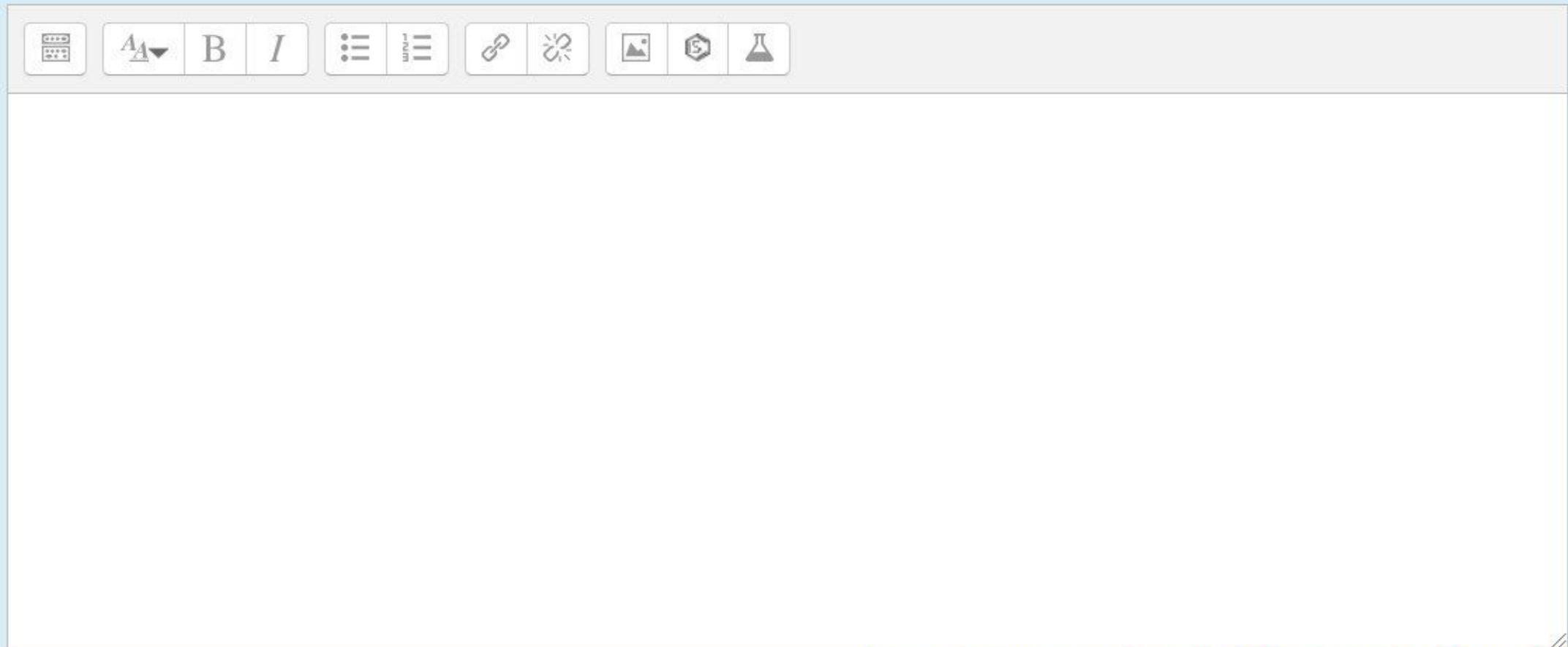
Si consideri l'applicazione *NextGen* vista a lezione e presentata sul libro di testo. Si supponga che abbia una finestra (realizzata mediante la classe *SaleJFrame*) che visualizza le informazioni sulla vendita e che cattura le operazioni del cassiere, la classe *SaleJFrame* è inoltre ascoltatore degli eventi generati dalla pressione dei bottoni della finestra (cioè implementa il metodo *actionListener*). Sono inoltre presenti le classi *Register*, un'astrazione dell'unità fisica, e la classe *Sale* che rappresenta la vendita in corso.

- Si completi il diagramma di sequenza sottostante (si assuma la figura della finestra come rappresentativa dell'IU senza preoccuparsi dei suoi dettagli ma come fosse un'entità unica) inserendo i messaggi *actionPerformed(actionEvent)*, *enterItem(itemID, qty)* e *makeLineItem(itemID, qty)*. Si noti che il messaggio *enterItem(itemID, qty)* è anche il nome del messaggio individuato dall'operazione di sistema che si sta realizzando.
- Si indichi quali classi appartengono allo strato UI e quali allo strato di dominio.
- Si indichi la classe che rappresenta il controllore GRASP.



Rappresentare mediante un diagramma delle classi (modello di dominio) le seguenti affermazioni:

- una vendita è saldata da un pagamento;
- un pagamento è necessariamente associato ad una vendita;
- una vendita è necessariamente saldata;
- un pagamento per contanti, con assegno o con carta di credito, è un pagamento;
- un pagamento è caratterizzato da un ammontare di denaro.



Dimensione massima per i file nuovi: 128MB, numero massimo di allegati: 1



Si consideri il seguente scenario di base di Elabora Vendita:



1. Il Cliente arriva alla cassa POS con gli articoli e/o i servizi da acquistare,
2. Il Cassiere inizia una nuova vendita.
3. Il Cassiere inserisce il codice identificativo di un articolo.
4. Il Sistema registra la riga di vendita per l'articolo e mostra una descrizione dell'articolo, il suo prezzo e il totale parziale.
5. Il Cassiere ripete i passi 3-4 fino a che non indica che ha terminato.
6. Il sistema mostra il totale.
7. Il Cassiere riferisce il totale al Cliente, e richiede il pagamento.
8. Il Cliente paga (in contanti) e il sistema gestisce il pagamento.
9. Il Sistema registra la vendita completata.
10. Il Sistema genera la ricevuta.
11. Il Cliente va via con la ricevuta e gli articoli acquistati.

Si disegni lo SSD corrispondente.



Chess.com Moodle Amazon Netflix Corso: Basi d.../8/19, Corso A - Leda OGame

esami.i-learn.unito.it

Seconda parte: domande aperte (36 minuti)

Plattforma Esami HelpDesk Italiano (it) +

Giulio Brazzo

NAVIGAZIONE QUIZ

1 2

Termina il tentativo...

Tempo rimasto 0:35:51

Domanda 1

Risposta non ancora data

Punteggio max.: 3,00

Contrassegna domanda

Si consideri il seguente frammento di codice in Java:

```
import java.io.*;
public class ProvaFile {
    public static void main(String arg[]) throws Exception {
        InputStream is = new FileInputStream("DatiNumerici.txt");
        FilterInputStream bis = new BufferedInputStream(is);
        FilterInputStream dis = new DataInputStream(bis);

        int i;
        char c;
        try {
            while ((i = dis.read()) != -1) {
                c = (char) i;
                System.out.print(c);
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (dis != null) dis.close();
        }
    }
}
```

Il file DatiNumerici.txt è il seguente:

9  
13  
7

l'esecuzione del main di ProvaFile restituisce il seguente risultato a console:

9  
13  
7

Presentare un modello di progetto (diagramma UML delle classi) che utilizzando il pattern decorator realizzi il comportamento descritto.

Dire quali delle seguenti affermazioni sono vere e quali false.

La responsabilità è un'astrazione di ciò che fa o rappresenta un oggetto o un componente software

Scegli... ▾

Le responsabilità sono assegnate ai concetti definiti nel modello di dominio

Scegli... ▾

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

Scegli... ▾

Nel Responsibility-Drive Development gli oggetti software sono considerati come dotati di responsabilità

Scegli... ▾

← → C esami.i-learn.unito.it/mod/quiz/attempt.php?attempt=138794

HelpDesk Italiano (it) ▾

Lorenzo Battaglia

Plataforma Esami

Domanda 3

Risposta non ancora data

Punteggio max.: 2,00

Domanda 4

Risposta non ancora data

Punteggio max.: 2,00

Contrassegna domanda

Dire se le seguenti affermazioni sono vere o false.

La disciplina dei requisiti è completata durante la fase dell'elaborazione ✓

La disciplina dei requisiti è completata durante la fase dell'ideazione

La disciplina dei requisiti è iniziata durante l'ideazione ✓

La disciplina dei requisiti è il processo per scoprire cosa deve essere costruito ✓

La disciplina dei requisiti deve orientare lo sviluppo verso il sistema corretto ✓

Dire quali delle seguenti affermazioni sono vere e quali false.

Le responsabilità sono assegnate ai concetti definiti nel modello di dominio ✓

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

Nel Responsibility-Driven Development gli oggetti software sono considerati come dotati di responsabilità ✓

La responsabilità è un'astrazione di ciò che fa o rappresenta un oggetto o un componente software ✓

Termina il tentativo...

Si consideri la seguente codice:

```
public class TestFileSystem {  
    public static void main(String[ ] args) {  
  
        FileSystem root = new Directory("/");  
        FileSystem devices = new Directory("/dev");  
        FileSystem std_out = new Device("/dev/console");  
        FileSystem home = new Directory("/home");  
        FileSystem baldoni = new Directory("/home/baldoni");  
        FileSystem compitoA = new File("/home/baldoni/compitoA.pdf");  
        FileSystem compitoB = new File("/home/baldoni/compitoB.pdf");  
  
        root.add(devices);  
        root.add(home);  
  
        devices.add(std_out);  
        home.add(baldoni);  
  
        baldoni.add(compitoA);  
        baldoni.add(compitoB);  
        baldoni.delete(compitoA);  
  
        System.out.println("Path: " + compitoB.getName() + ":" + compitoB.getPath());  
  
        System.out.println(std_out.getName() + " is a " + std_out.isFile());  
        System.out.println(std_out.getName() + " is a " + std_out.isDevice());  
        System.out.println(std_out.getName() + " is a " + std_out.isDirectory());  
        System.out.println(compitoB.getName() + " is a " + compitoB.isFile());  
        System.out.println(compitoB.getName() + " is a " + compitoB.isDevice());  
        System.out.println(baldoni.getName() + " is a " + baldoni.isDirectory());
```

```
public class TestPaint {  
    public static void main(String[] args) {  
        Figure[ ] figura = new Figure[5];  
  
        figura[0] = new Cerchio();  
        figura[1] = new Rettangolo();  
  
        FiguraColorata cerchioBordoRosso = new ColoreBordo(new Cerchio(), "rosso");  
        FiguraColorata cerchioRosso = new ColoreSfondo(cerchioBordoRosso, "rosso");  
  
        figura[2] = cerchioRosso;  
  
        FiguraColorata cerchioSfondoBlu = new ColoreBordo(new Cerchio(), "blu");  
        FiguraColorata cerchioBlu = new ColoreSfondo(cerchioSfondoBlu, "blu");  
  
        figura[3] = cerchioBlu;  
  
        FiguraColorata rettangoloSfondoBlu = new ColoreSfondo(new Rettangolo(), "blu");  
  
        figura[4] = rettangoloSfondoBlu;  
        for (int i = 0; i < 5; i++) {  
            figura[i].disegna();  
        }  
    }  
}
```



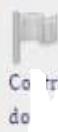
Questo utilizza un insieme di classi che realizzano un programma di disegno utilizzando un noto pattern GoF. Si dica di quale pattern si tratta e disegnare il diagramma UML delle classi coinvolte.

**Domanda 1**

Quali delle seguenti affermazioni sui contratti sono vere e quali false.

Risposta non ancora data

Punteggio max.:  
2,00



Il principale input per la redazione dei contratti sono le operazioni di sistema e il modello di dominio. V

Scegli... ▾

Usano pre-condizione e post-condizione per descrivere nel dettaglio i cambiamenti agli oggetti in un modello di progetto (software). F

Scegli... ▾

I contratti servono come input per il modello di dominio. F

Scegli... ▾

Sono considerati parte del Modello dei Casi d'Uso, poiché forniscono maggiori dettagli dell'analisi sull'effetto delle operazioni di sistema implicate dai casi d'uso. X F

Scegli... ▾

Le pre-condizioni descrivono i cambiamenti di stato degli oggetti nel modello di dominio dopo il completamento dell'operazione. F

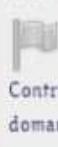
Scegli... ▾

**Domanda 2**

Dire quali delle seguenti affermazioni sui casi d'uso sono vere e quali false.

Risposta non ancora data

Punteggio max.:  
2,00



I casi d'uso servono a catturare i requisiti funzionali. V

Scegli... ▾

Nei processi a cascata, l'analisi e la progettazione si basano sulla realizzazione di casi d'uso. X F

Scegli... ▾

Sono una collezione di soli scenari di successo che descrivono un attore che usa il sistema per raggiungere un obiettivo specifico. V

Scegli... ▾

I casi d'uso sono descrizioni testuali di scenari di uso interessanti del sistema software che si deve realizzare. V

Scegli... ▾

All'inizio del progetto vengono definiti in dettaglio tutti i requisiti derivanti dai casi d'uso. F

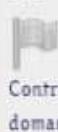
Scegli... ▾

**Domanda 3**

Dire se le seguenti affermazioni su UP sono vere o false.

Risposta non ancora data

Punteggio max.:  
2,00



La fase di Ideazione viene prima della fase di Elaborazione. V

Scegli... ▾

La fase di Costruzione viene dopo la fase di Costruzione. F

Scegli... ▾

La fase di ideazione è suddivisa in iterazioni. F

Scegli... ▾

La fase di Ideazione è una visione raffinata, implementazione iterativa del nucleo dell'architettura, risoluzione dei rischi maggiori, identificazione della maggior parte dei requisiti e della portata. F

Scegli... ▾

**Domanda 2**

Risposta non ancora data

Punteggio max.:  
5,00



Si consideri la seguente codice:

```
public class TestAmmazzon {  
    public static void main(String[] args) {  
  
        Carrello cart1 = new Carrello();  
  
        Item crema = new Item("Crema da barba", 6);  
        Item rasoio = new Item("Rasoio elettrico", 75);  
        Item dopobarba = new Item("Dopobarba", 12);  
  
        cart1.addItem(crema);  
        cart1.addItem(rasoio);  
        cart1.addItem(dopobarba);  
  
        cart1.removeItem(crema);  
  
        Pagamento pagapaypal = new PayPal("matteo@baldoni.it", "passwd");  
        cart1.tipopagamento(pagapaypal);  
  
        cart1.paga();  
  
        Carrello cart2 = new Carrello();  
  
        Item altracrema = new Item("Crema da barba", 6);  
        Item altrorasoio = new Item("Rasoio a lama", 9);  
        Item altrodopobarba = new Item("Crema lenitiva", 10);  
  
        cart2.addItem(altracrema);  
        cart2.addItem(altrorasoio);  
        cart2.addItem(altrodopobarba);  
  
        Pagamento cc = new CrediCard("Matteo Baldoni", "1234123412341234", "789", "12/21");  
        cart2.tipopagamento(cc);  
  
        cart2.paga();  
  
    }  
}
```

Questo utilizza un insieme di classi che realizzano un carrello per un sito di e-commerce utilizzando un noto pattern GoF. Si dica di quale pattern si tratta e disegnare il diagramma UML delle classi coinvolte.

Un aeroporto è costituito da uno o più terminali. Un volo ha origine da un solo aeroporto e ha destinazione in un solo aeroporto. Lo stesso aeroporto può essere origine di più voli e destinazione di più voli." Disegnare il modello di dominio (utilizzando UML) ✓

Definire in modo preciso la pre-condizione e la post-condizione di un'operazione con le sue pre- e post-condizioni.

Si consideri il seguente codice:

```
public interface CompressionAlgorithm {  
    public void compressFiles(ArrayList<File> files);  
}  
  
public class ZipCompressionAlgorithm implements CompressionAlgorithm {  
    public void compressFiles(ArrayList<File> files) {  
        //using ZIP algorithm  
    }  
}  
  
public class RarCompressionAlgorithm implements CompressionAlgorithm {  
    public void compressFiles(ArrayList<File> files) {  
        //using RAR algorithm  
    }  
}  
  
public class CompressionContext {  
    private CompressionAlgorithm algorithm;  
    //this can be set at runtime by the application preferences  
    public void setCompressionAlgorithm(CompressionAlgorithm algorithm) {  
        this.algorithm = algorithm;  
    }  
    public void createArchive(ArrayList<File> files) {  
        algorithm.compressFiles(files);  
    }  
}  
  
public class Client {  
    public static void main(String[] args) {  
        CompressionContext ctx = new CompressionContext();  
        //we could assume context is already set by preferences  
        ctx.setCompressionAlgorithm(new ZipCompressionAlgorithm());  
        //get a list of files...  
        ctx.createArchive(fileList);  
    }  
}
```

Decorator  
↓ Start

Questo utilizza un insieme di classi che adottano un noto pattern GoF. Si dica di quale pattern si tratta.

Scegli un'alternativa:

Dire quali delle seguenti affermazioni sono vere e quali false.

Il processo sviluppo incrementale è migliore rispetto al processo a cascata sotto ogni punto di vista e contesto di applicazione attuale.

Vero

Il processo sviluppo incrementale è basata su uno svolgimento sequenziale delle diverse attività dello sviluppo software.

Falso

Il processo a cascata è migliore rispetto allo sviluppo incrementale sotto ogni punto di vista e contesto di applicazione attuale.

Falso

Lo sviluppo a cascata si basa sull'idea di sviluppare un'implementazione iniziale, esporla agli utenti e perfezionarla attraverso molte versioni.

Vero

Dire quali delle seguenti affermazioni sono vere e quali false.

In generale il Controller MVC delega le richieste di lavoro dall'utente al Controller GRASP del dominio.

V

V

Vero

Il controller GRASP fa parte della UI e gestisce l'interazione con l'utente; la sua implementazione dipende in larga misura dalla tecnologia UI e dalla piattaforma che viene utilizzata.

F

Falso

Falso

E sia il Controller GRASP che il Controller MVC si occupano di gestire le richieste provenienti dall'utente.

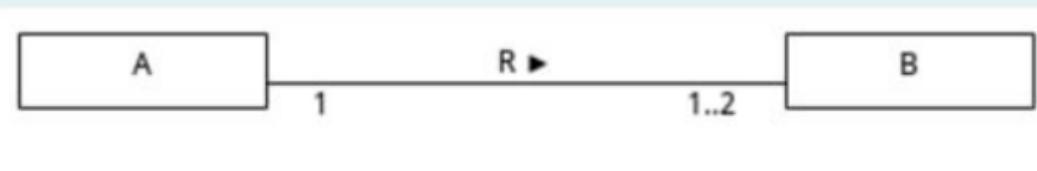
F

F

Vero



Si consideri la seguente associazione R tra A e B:



dire quali delle seguenti affermazioni sono vere e quali false.

Se  $A = \{a_1, a_2\}$  e  $B = \{b_1, b_2, b_3\}$ , R può essere  $\{(a_1, b_1), (a_1, b_2), (a_2, b_3)\}$ . ✓

Se  $A = \{a_1, a_2\}$  e  $B = \{b_1, b_2, b_3\}$ , R può essere  $\{(a_1, b_1), (a_2, b_1), (a_2, b_3)\}$ . ✗

Se  $A = \{a_1, a_2\}$  e  $B = \{b_1, b_2, b_3\}$ , R può essere  $\{(a_1, b_1), (a_2, b_2), (a_2, b_3)\}$ . ✗

Se  $A = \{a_1, a_2\}$  e  $B = \{b_1, b_2, b_3\}$ , R può essere  $\{(a_1, b_1), (a_2, b_3)\}$ . ✗

Le istanze di B sono visibili ad A ma non viceversa. ✗

Falso  ✓

Falso  ✗

Falso  ✓

Falso  ✗

Vero  ✗

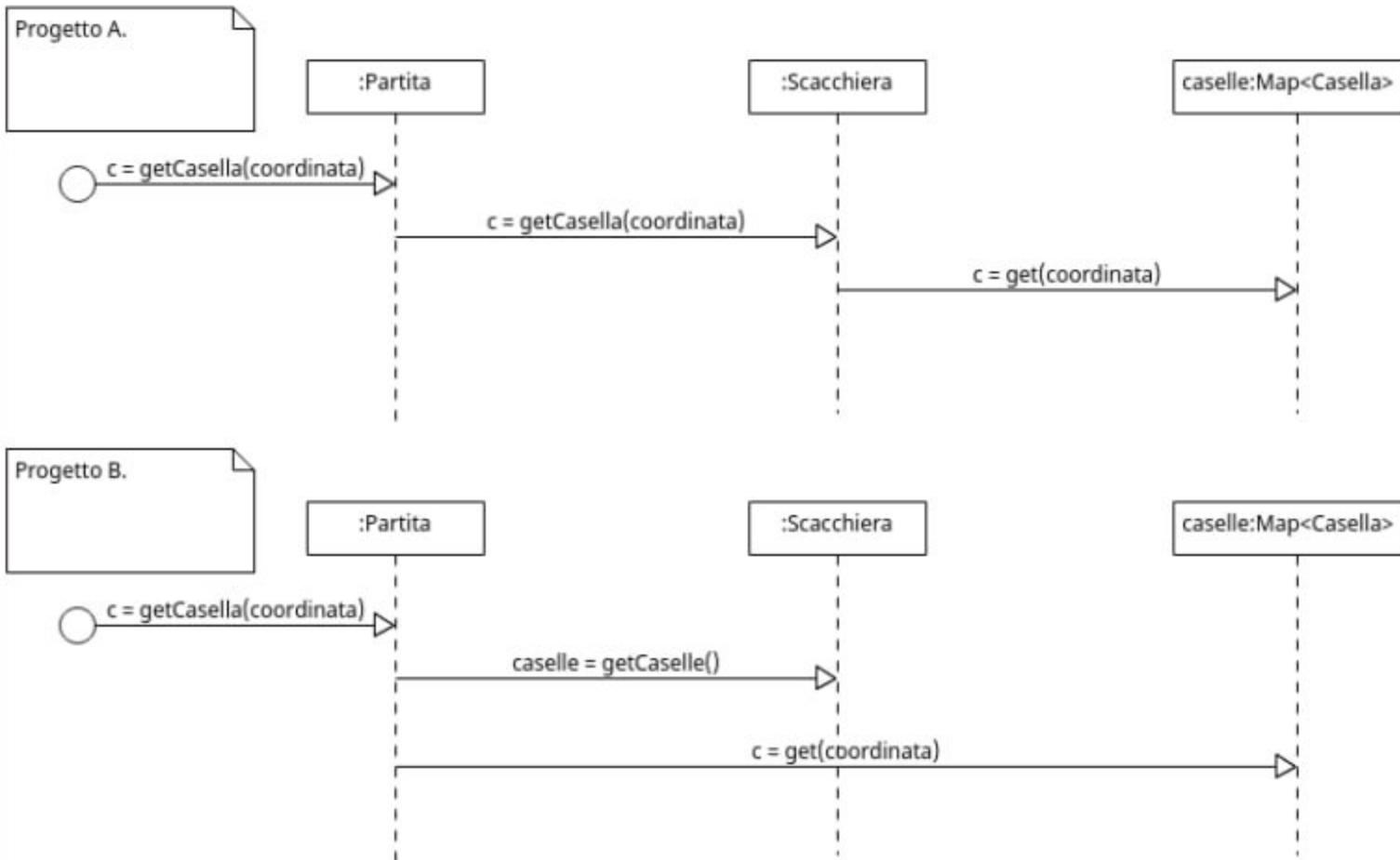
Domanda 5

Risposta non ancora data

Punteggio max.: 2,50

Contrassegna domanda

Si consideri il modello di dominio (in alto) e i due progetti di interazione, denominati progetto A e progetto B. Si dica quali delle seguenti sono vere e quali false.



Il progetto B presenta 2 sole dipendenze/accoppiamenti: Partita -> Scacchiera e Partita -> Casella.

 Falso

Il progetto A prevede la delega di getCasella da Partita a Scacchiera.

 Vero

Il progetto A presenta 2 sole dipendenze/accoppiamenti: Partita -> Scacchiera e Scacchiera -> Casella.

 Falso

Il progetto migliore secondo il principio di low coupling è il progetto A.

 Vero

Si consideri il seguente codice:

```
public abstract class MenuElement {  
    private String name;  
    private String url;  
  
    public void add(MenuElement component) {  
        throw new UnsupportedOperationException();  
    }  
    public abstract void displayMenu();  
}  
  
public class MenuItem extends MenuElement {  
    public MenuItem(String name, String url) {  
        super(name, url);  
    }  
    public void displayMenu() {  
        System.out.println(getName() + " : " + getUrl());  
    }  
}  
  
public class Menu extends MenuElement {  
    List<MenuElement> subMenus = new ArrayList<>();  
    public Menu(String name, String url) {  
        super(name, url);  
    }  
    public void add(MenuElement MenuElement) {  
        this.subMenus.add(MenuElement);  
    }  
    public void displayMenu() {  
        System.out.println(getName() + " : " + getUrl() + "\n");  
        this.subMenus.forEach(MenuElement::displayMenu);  
    }  
}
```



## Composite

Questo utilizza un insieme di classi che adottano un noto pattern GoF. Si dica di quale pattern si tratta, tenendo conto che un esempio di classe client potrebbe essere:

```
MenuElement allTutorials = new Menu("Tutorials", "/tutorials");  
MenuElement spring = new Menu("Spring", "/spring");  
MenuElement versioning  
= new Menu("Version Management", "/versioning");  
MenuItem java = new MenuItem("Java", "/java");  
  
allTutorials.add(spring);  
allTutorials.add(versioning);  
allTutorials.add(java);  
spring.add(new MenuItem("Spring Core", "/core"));  
spring.add(new MenuItem("Spring Boot", "/boot"));
```

Domanda 7

Risposte non

ancora date

Punteggio

max.: 2,25

V

Contrassegna

domanda

Si consideri il seguente codice:

```

public abstract class Beverage {
    protected String description = "Unknown Beverage";

    public String getDescription() {
        return description;
    }
    public abstract double cost();
}

public class Espresso extends Beverage {

    public Espresso() {
        this.description = "Espresso";
    }
    public double cost() {
        return 1;
    }
}

public class HouseBlend extends Beverage {

    public HouseBlend() {
        this.description = "House Blend";
    }
    public double cost() {
        return 1.5;
    }
}

public abstract class Condiment extends Beverage {

    public abstract String getDescription();
}

public class Mocha extends Condiment {

    private final Beverage beverage;

    public Mocha(Beverage beverage) {
        this.beverage = beverage;
    }
    public double cost() {
        return 0.20 + beverage.cost();
    }
    public String getDescription() {
        return beverage.getDescription() + ", Mocha";
    }
}

public class Whip extends Condiment {

    private final Beverage beverage;

    public Whip(Beverage beverage) {
        this.beverage = beverage;
    }
    public double cost() {
        return 0.15 + beverage.cost();
    }
    public String getDescription() {
        return beverage.getDescription() + ", Whip";
    }
}

```

Questo utilizza un insieme di classi che adottano un noto pattern GoF. Si dica di quale pattern si tratta, tenendo conto che un esempio di classe client potrebbe essere:

```

public static void main(String[] args) {
    Beverage b = new Mocha(new Mocha(new Whip(new HouseBlend())));
    System.out.println(b.getDescription());
    System.out.println(b.cost());
}

```

La cui esecuzione restituisce:

House Blend, Whip, Mocha, Mocha  
2.05



Dire quali delle seguenti affermazioni sono vere e quali false.

Durante l'elaborazione i documenti vengono abbozzati e definiti in maniera "leggera".

F

La pianificazione delle iterazioni è basata sul rischio, inizialmente vengono scelti i casi d'uso meno rischiosi.

F

L'ideazione è la prima iterazione dell'elaborazione.

F

La maggior parte dei requisiti è individuata durante la fase di ideazione.

✓ Uc non  
requisiti

Il modello dei casi d'uso viene iniziato durante l'elaborazione.

F

Falso F

Vero F

Falso F

Falso V

Vero F

Domanda 1

Completo

Punteggio  
ottenuto 1,50  
su 1,50



Contrassegna  
domanda

Dire quali delle seguenti affermazioni sono vere e quali false.

L'ideazione è la prima iterazione dell'elaborazione.

Falso

La maggior parte dei requisiti è individuata durante la fase di ideazione.

Falso

Durante l'elaborazione i documenti vengono abbozzati e definiti in maniera "leggera".

Falso

La pianificazione delle iterazioni è basata sul rischio, inizialmente vengono scelti i casi d'uso meno rischiosi.

Falso

Il modello dei casi d'uso viene iniziato durante l'elaborazione.

Falso

Domanda **2**

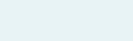
Completo

Punteggio  
ottenuto 1,75  
su 1,75

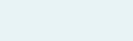
 Contrassegna  
domanda

Dire quali delle seguenti affermazioni sono vere e quali false.

In generale il Controller GRASP delega le richieste di lavoro dall'utente al Controller MVC della UI.

Falso 

Sia il Controller GRASP che il Controller MVC si occupano di gestire le richieste provenienti dall'utente.

Vero 

Il controller MVC fa parte della UI e gestisce l'interazione con l'utente; la sua implementazione dipende in larga misura dalla tecnologia UI e dalla piattaforma che viene utilizzata.

Vero 

Domanda 3  
Completo  
Punteggio  
ottenuto 0,00  
su 2,25  
 Contrassegna  
domanda

Si consideri il seguente codice:

```
public class ScannerPerCarrello implements Scanner {  
    @Override  
    public Double scan(MerceVendutaAPeso merce) {  
        return merce.getPeso() * merce.getPrezzoAlKg();  
    }  
  
    @Override  
    public Double scan(MerceVendutaInPezzi merce) {  
        return merce.getNumeroDiPezzi() * merce.getPrezzonitario();  
    }  
}  
  
public class Main {  
  
    public static void main(String[] args) {  
        List<Merce> carrello = new ArrayList<>();  
  
        MerceVendutaInPezzi p1 = new MerceVendutaInPezzi("CO1","Cereali",2.30D,2);  
        MerceVendutaInPezzi p2 = new MerceVendutaInPezzi("CO2","Quaderno",1.10D,1);  
        MerceVendutaAPeso p3 = new MerceVendutaAPeso("CO3","Mele",2.50D,2.0D);  
  
        carrello.add(p1);  
        carrello.add(p2);  
        carrello.add(p3);  
  
        Double totaleSpesa = calcolaTotale(carrello);  
        System.out.println("Costo totale " + totalCost + " euro");  
    }  
  
    private static Double calcolaTotale(List<Merce> carrello) {  
        Double totale = 0.0D;  
        Scanner scanner = new ScannerPerCarrello();  
  
        for(Merce merce : carrello){  
            totale = totale + merce.accept(scanner);  
        }  
        return totale;  
    }  
}
```

Visitor

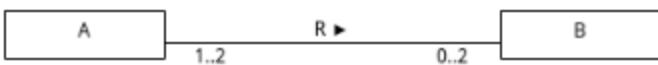
Questo utilizza un insieme di classi che realizzano uno scanner per carrello della spesa utilizzando un noto pattern GoF. Si dica di quale pattern si tratta.

**Domanda 4**

Completo

Punteggio  
ottenuto 2,00  
su 2,00 Contrassegna  
domanda

Si consideri la seguente associazione R tra A e B:



dire quali delle seguenti affermazioni sono vere e quali false.

Se  $A = \{a_1, a_2, a_3\}$  e  $B = \{b_1, b_2\}$ , R può essere  $\{(a_1, b_1), (a_2, b_1), (a_3, b_1)\}$ . FalsoSe  $A = \{a_1, a_2, a_3\}$  e  $B = \{b_1, b_2\}$ , R può essere  $\{(a_1, b_1), (a_3, b_2)\}$ . VeroSe  $A = \{a_1, a_2, a_3\}$  e  $B = \{b_1, b_2\}$ , R può essere  $\{(a_1, b_1), (a_2, b_1), (a_3, b_2)\}$ . Vero

Le istanze di B sono visibili ad A ma non viceversa.

 FalsoSe  $A = \{a_1, a_2, a_3\}$  e  $B = \{b_1, b_2\}$ , R può essere  $\{(a_2, b_2), (a_3, b_2)\}$ . Falso**Domanda 5**

Completo

Punteggio  
ottenuto 1,75  
su 1,75 Contrassegna  
domanda

Dire quali delle seguenti affermazioni sono vere e quali false.

Il processo a cascata è migliore rispetto allo sviluppo incrementale sotto ogni punto di vista e contesto di applicazione.

 Falso

Il processo a cascata è basata su uno svolgimento sequeziale delle diverse attività dello sviluppo software.

 Vero

Il processo sviluppo incrementale è migliore rispetto al processo a cascata sotto ogni punto di vista e contesto di applicazione.

 Falso

Lo sviluppo incrementale si basa sull'idea di sviluppare un'implementazione iniziale, esporla agli utenti e perfezionarla attraverso molte versioni.

 Vero

Domanda 6  
Completo  
Punteggio ottenuto 0,00 su 2,25  
 Contrassegna domanda

Si consideri il seguente codice:

```
public interface Behavior {  
    void onEnter();  
    void observe();  
}  
  
public class PeacefulBehavior implements Behavior {  
  
    private final Mammoth mammoth;  
  
    public PeacefulBehavior(Mammoth mammoth) {  
        this.mammoth = mammoth;  
    }  
    public void observe() {  
        LOGGER.info("{} is calm and peaceful.", mammoth);  
    }  
    public void onEnter() {  
        LOGGER.info("{} calms down.", mammoth);  
    }  
}  
  
public class AngryBehavior implements Behavior {  
  
    private final Mammoth mammoth;  
  
    public AngryBehavior(Mammoth mammoth) {  
        this.mammoth = mammoth;  
    }  
    public void observe() {  
        LOGGER.info("{} is furious!", mammoth);  
    }  
    public void onEnter() {  
        LOGGER.info("{} gets angry!", mammoth);  
    }  
}  
  
public class Mammoth {  
  
    private Behavior behavior;  
  
    public Mammoth() {  
        behavior = new PeacefulBehavior(this);  
    }  
    public void timePasses() {  
        if (behavior.getClass().equals(PeacefulBehavior.class)) {  
            changeBehaviorTo(new AngryBehavior(this));  
        } else {  
            changeBehaviorTo(new PeacefulBehavior(this));  
        }  
    }  
    private void changeBehaviorTo(Behavior newBehavior) {  
        this.behavior = newBehavior;  
        this.behavior.onEnter();  
    }  
    public String toString() {  
        return "The mammoth";  
    }  
    public void observe() {  
        this.behavior.observe();  
    }  
}
```

V state

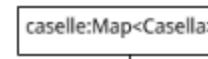
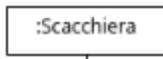
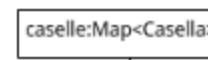
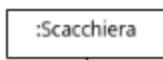
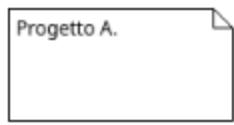
Questo utilizza un insieme di classi che adottano un noto pattern GoF. Si dica di quale pattern si tratta.

Domanda 7

Completo

Punteggio  
ottenuto 1,88  
su 2,50 Contrassegna  
domanda

Si consideri il modello di dominio (in alto) e i due progetti di interazione, denominati progetto A e progetto B. Si dica quali delle seguenti affermazioni sono vere e quali false.



Il progetto A prevede la delega di getCasella da Partita a Scacchiera. ✓

Il progetto migliore secondo il principio di low coupling è il progetto A. ✓ ✓

Il progetto A presenta 2 sole dipendenze/accoppiamenti: Partita -> Scacchiera e Scacchiera -> Casella.

Il progetto B presenta 2 sole dipendenze/accoppiamenti: Partita -> Scacchiera e Partita -> Casella. ↗

 Vero

 Vero

 Vero

 Vero

→ ha 3!

2,00



Contrassegna  
domanda

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



Vero



Vero



Falso

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

## Sviluppo di Applicazioni Software, a.a. 2018/2019

1. (2 punti) Quali delle seguenti affermazioni sono vere?

- I requisiti non funzionali sono le proprietà del sistema nel suo complesso
- I requisiti non funzionali sono i requisiti comportamentali
- I requisiti funzionali sono le proprietà del sistema nel suo complesso
- I requisiti funzionali sono i requisiti comportamentali

2. (2 punti) Quali delle seguenti affermazioni sono vere?

- I casi d'uso sono una collezione di scenari correlati
- Gli scenari sono una collezione di casi d'uso correlati
- Un caso d'uso è una collezione di storie nell'uso del sistema
- Un caso d'uso è una collezione di SSD
- I casi d'uso si rappresentano mediante gli SSD
- Un caso d'uso è una descrizione testuale di scenari
- Uno scenario esprime un obiettivo specifico raggiungibile da un attore attraverso sequenza di azioni ed interazioni con un sistema

3. (2 punti) Quali delle seguenti affermazioni sono vere?

- La disciplina dei requisiti è il processo per scoprire cosa deve essere costruito.
- La disciplina dei requisiti deve orientare lo sviluppo verso il sistema corretto.

- La disciplina dei requisiti è completata durante la fase dell'elaborazione
- La disciplina dei requisiti è completata durante la fase dell'ideazione
- La disciplina dei requisiti è iniziata durante l'elaborazione

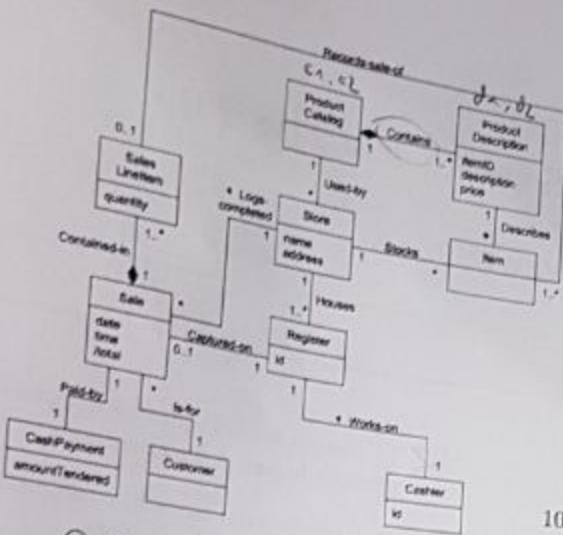
4. (1 punto) La programmazione di qualità-produzione per un sottoinsieme dei requisiti si inizia dopo che l'analisi di tutti i requisiti sia stata completata.

- Vero
- Falso

5. (1 punto) Quali di questi elaborati sono iniziati durante l'ideazione?

- Scenario di sviluppo
- SSD
- Modello di progetto
- Modello dei casi d'uso
- DCD
- Glossario
- Visione e studio economico

6. (2 punti) Si consideri il seguente *Modello di Domio*, quali delle seguenti affermazioni è vera? Si supponga  $Product\ Catalog = \{c_1, c_2\}$  e  $Product\ Description = \{d_1, d_2\}$ .



- Contains può essere  $\{(c_1, d_1), (c_1, d_2), (c_2, d_2)\}$
- Contains può essere  $\{(c_1, d_1), (c_2, d_1), (c_1, d_2)\}$
- Contains può essere  $\{(c_1, d_1), (c_2, d_2)\}$
- Contains può essere  $\{(c_1, d_1), (c_1, d_2)\}$

7. (1 punto) Le post-condizioni descrivono i cambiamenti nello stato degli oggetti del modello di dominio. I cambiamenti dello stato del modello di progetto comprendono gli oggetti creati, i collegamenti formati o rotti, e gli attributi modificati.

- Vero
- Falso

8. (2 punti) Quali delle seguenti affermazioni sulle associazioni utilizzate nel modello di dominio sono vere?

- È per sua natura unidirezionale
- La direzione di lettura è una specifica di visibilità
- Rappresenta un valore logico degli oggetti di una classe
- Rappresenta una relazione significativa tra classi
- Rappresenta un'insieme di n-tuple di oggetti delle classi

9. (2 punti) Durante la fase di elaborazione:

- Viene scoperta e stabilizzata la maggior parte dei requisiti
- Non vengono effettuati test al codice sviluppato

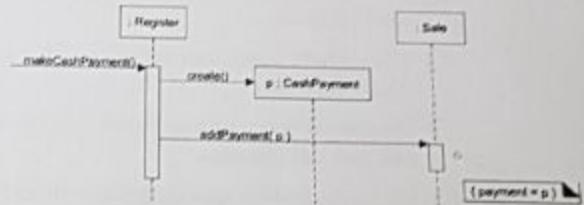
I rischi maggiori sono attenuati o rientrano

- Non si sviluppa codice
- Si realizza uno studio economico per stabilire l'ordine di grandezza del progetto e dei costi
- Viene programmato il nucleo, rischioso, dell'architettura
- Si analizzano circa il 10% dei casi d'uso in dettaglio

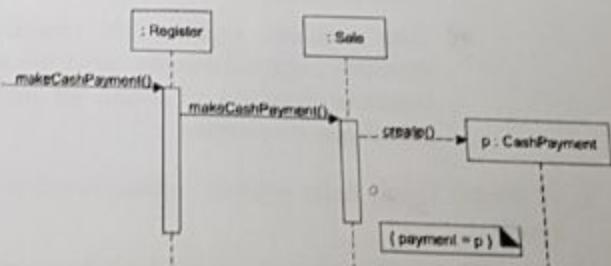
10. (1 punto) Ricevere un parametro di tipo A per un metodo di una classe B rappresenta una relazione di dipendenza da B a A mentre estendere una classe A per una classe B non rappresenta una relazione di dipendenza da B a A.

- Vero
- Falso

11. (3 punti) Quanti e quali "accoppiamenti" mostra il diagramma di interazione (A) e il diagramma di interazione di (B)? Spiegare perché.



(A)



(B)

Risposta:

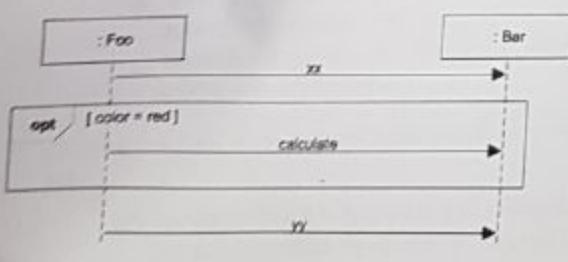
4) To Register = CahnBgment

Foo obj2.  
Bar obs  
public void method() {  
 obj2.xx();  
 if (obj2.color == "red")  
 obj2.calculate();  
 obj2.yy();

12. (2 punti) In UML per responsabilità si intende:

- la specifica di un metodo associato ad una classe Java
- la specifica di una variabile di istanza o di un metodo associato ad una classe Java
- la specifica di una variabile di istanza di una classe Java
- un contratto o un obbligo di un classificatore

13. (3 punti) Scrivere il codice Java corrispondente al seguente diagramma di sequenza:



Risposta:

14. (1 punto) Quale pattern tratta il problema "Come ridurre l'impatto dei cambiamenti? Come sostenere una dipendenza bassa, un impatto dei cambiamenti basso e una maggiore opportunità di riuso?"

- Abstract Factory
- Information Expert
- Low Coupling
- Controller
- Singleton

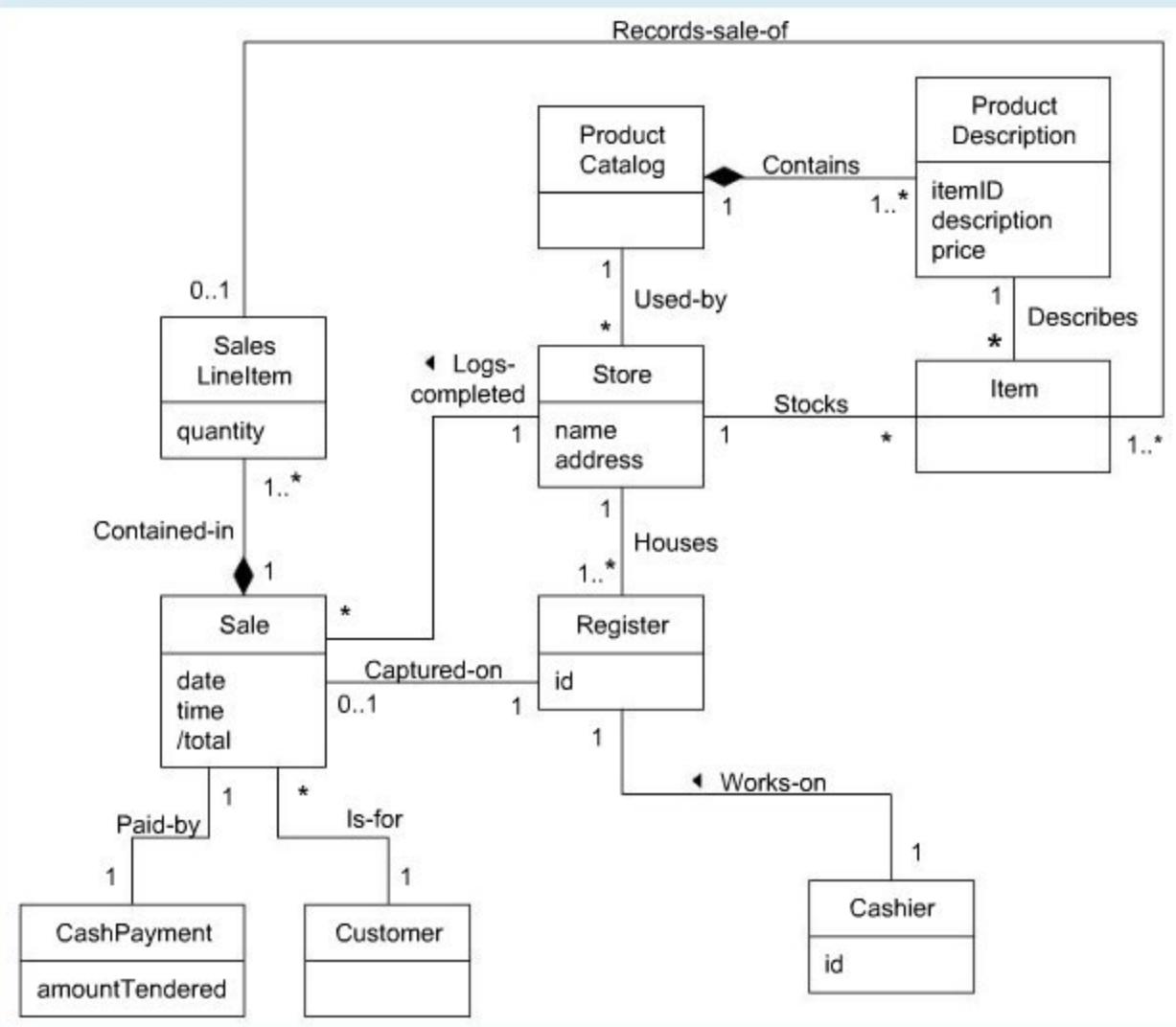
15. (5 punti) Dire a quale pattern GoF la seguente implementazione della struttura dati albero binario si conforma. Disegnare la struttura del pattern e il diagramma delle classi della soluzione applicata alla struttura dati albero binario.

```
1 public abstract class Tree {  
2     public abstract boolean empty();  
3     public abstract int getRootElement();  
4     public abstract void stampaPostVisita();  
5 }  
6  
7 public class Leaf extends Tree {  
8     public Leaf() {}  
9     public boolean empty() {  
10         return true;  
11     }  
12     public int getRootElement() {
```

Si consideri il seguente Modello di Domnio, dire se le seguenti affermazioni sono vere o false. Si supponga:

SalesLineItem = {s1, s2}

Item = {i1, i2, i3, i4}



Records-sale-of può essere:  $\{(s1, i1), (s1, i2), (s2, i1)\}$

Falso



Records-sale-of può essere:  $\{(s1, i1), (s2, i1)\}$

Falso



Records-sale-of può essere:  $\{(s1, i1), (s2, i2)\}$

Vero



Records-sale-of può essere:  $\{(s1, i1), (s1, i2)\}$

Vero



Records-sale-of può essere:  $\{(s1, i1), (s1, i2), (s2, i3), (s2, i4)\}$

Vero



Domanda 2  
Risposta non ancora data  
Punteggio max.: 3,00  
Contrassegna domanda

Quanti e quali "accoppiamenti" mostra il diagramma di interazione (A) e il diagramma di interazione di (B)? Quale dei due diagrammi è preferibile e perché?

(A)

```
sequenceDiagram
    participant Register
    participant Sale
    Register->>Sale: makeCashPayment()
    activate Register
    Register->>p: create()
    deactivate Register
    activate p
    p-->>Sale: addPayment(p)
    deactivate p
    deactivate Sale
```

(B)

```
sequenceDiagram
    participant Register
    participant Sale
    Register->>Sale: makeCashPayment()
    activate Register
    Register->>p: makeCashPayment()
    deactivate Register
    activate p
    p-->>Sale: create()
    deactivate p
    deactivate Sale
```

Domanda 2



Risposta non  
ancora data

Punteggio max.:  
3,00



Contrassegna  
domanda

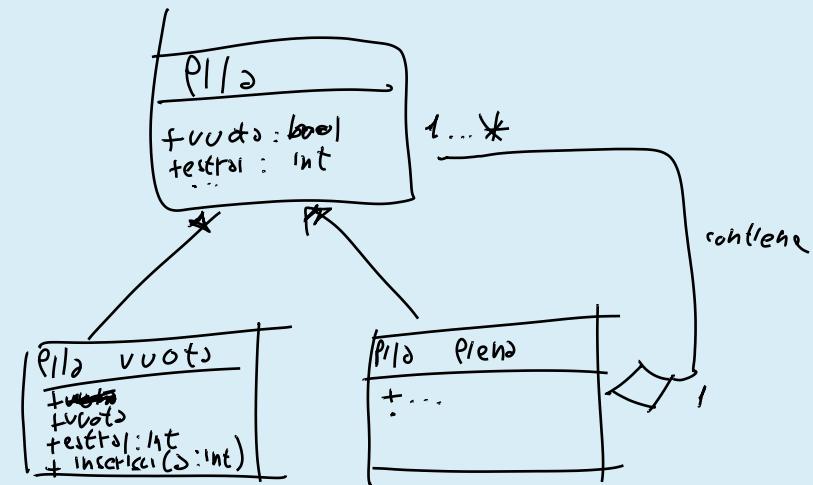
Si consideri il seguente frammento di codice in Java:

```
public class TestPila {  
    public static void main(String[] args) {  
        Pila p = new PilaVuota();  
        System.out.println(p.vuota());  
        p = p.inserisci(2);  
        System.out.println(p.vuota());  
        System.out.println(p.leggiCima());  
        p = p.inserisci(3);  
        System.out.println(p.leggiCima());  
        p = p.estrai();  
        System.out.println(p.leggiCima());  
        p = p.estrai();  
        System.out.println(p.leggiCima());  
    }  
}
```

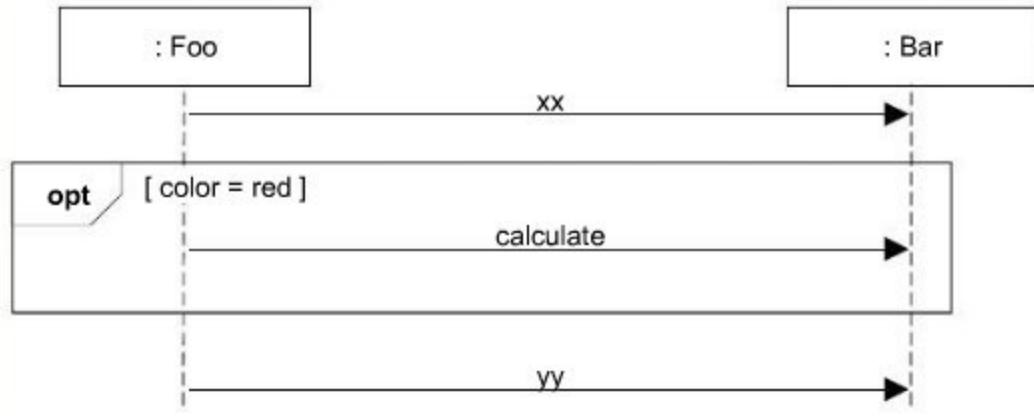
la cui esecuzione restituisce il seguente risultato a console:

```
true  
false  
2  
3  
2  
Exceptioninthread "main" java.lang.AssertionError  
at PilaVuota.leggiCima(PilaVuota.java: 18)  
at TestPila.main(TestPila.java: 15)
```

Presentare un modello di progetto (diagramma UML delle classi, si può usare UMLetino) che utilizzando il pattern composite realizza il comportamento descritto.



Scrivere il codice Java corrispondente al seguente diagramma di sequenza:



```
public class Foo {  
    public String color = "";  
  
    public void test(Bar bar) {  
        bar.xx();  
        if(this.color == "red")  
            bar.calculate();  
        bar.yy();  
    }  
}
```

```
public class Bar {  
    public void xx() {...}  
    public void yy() {...}  
    public void calculate() {...}  
}
```

**Domanda 1**

Risposta corretta

Punteggio ottenuto  
2,00 su  
2,00

Contrassegna domanda

Durante la fase di elaborazione

F

Non vengono effettuati test al codice sviluppato

F

Non si sviluppa codice

F

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)

✓

Viene scoperta e stabilizzata la maggior parte dei requisiti

F

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

V

Viene programmato il nucleo, rischioso, dell'architettura

- Falso ✓
- Falso ✓
- Falso ✓
- Vero ✓
- Falso ✓
- 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 corretta

Punteggio ottenuto

1,50 su  
2,00

Contrassegna 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

F

Vero

X

Falso

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

F

X

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

V

Vero

X

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

F

Falso

X

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

**Domanda 3**

Parzialmente corretta

Punteggio ottenuto

1,60 su  
2,00

Contrassegna 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  X

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

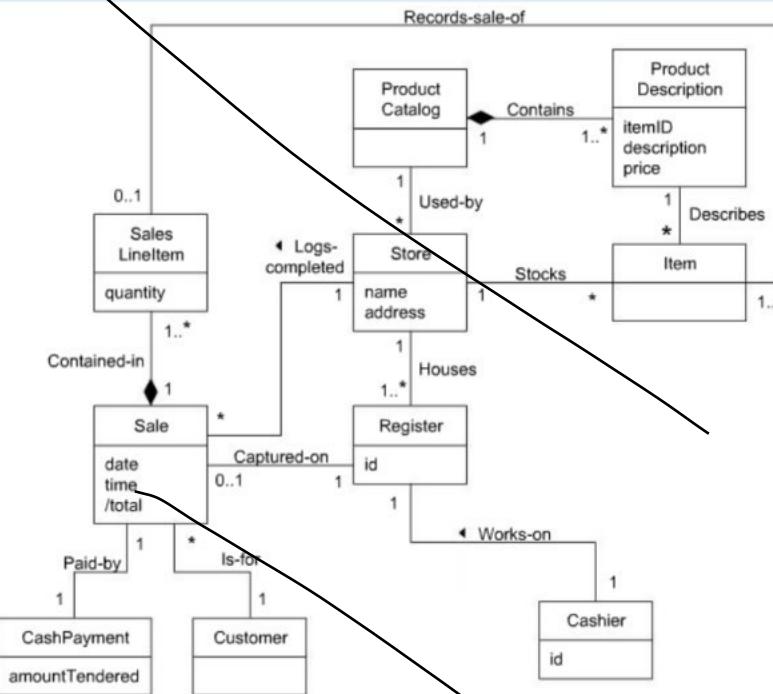
Domanda  
4

Parzialmente corretta

Punteggio ottenuto  
1,60 su  
2,00

Contrassegna domanda

Si consideri il seguente Modello di Domnio, dire se le seguenti affermazioni sono vere o false. Si supponga:  
 $SalesLineItem = \{s1, s2\}$   
 $Item = \{i1, i2, i3, i4\}$



Records-sale-of può essere:  $\{(s1, i1), (s2, i1)\}$

Falso ✓

Records-sale-of può essere:  $\{(s1, i1), (s1, i2)\}$

Vero ✗

Records-sale-of può essere:  $\{(s1, i1), (s1, i2), (s2, i1)\}$

Falso ✓

Records-sale-of può essere:  $\{(s1, i1), (s2, i2)\}$

Vero ✓

Records-sale-of può essere:  $\{(s1, i1), (s1, i2), (s2, i3), (s2, i4)\}$

Vero ✓

La risposta corretta è: Records-sale-of può essere:  $\{(s1, i1), (s2, i1)\}$  - Falso, Records-sale-of può essere:  $\{(s1, i1), (s1, i2)\}$  - Falso, Records-sale-of può essere:  $\{(s1, i1), (s1, i2), (s2, i1)\}$  - Falso, Records-sale-of può essere:  $\{(s1, i1), (s2, i2)\}$  - Vero, Records-sale-of può essere:  $\{(s1, i1), (s1, i2), (s2, i3), (s2, i4)\}$  - Vero