

La prova si svolge a libri chiusi (non è permessa la consultazione di materiale didattico).
Traccia delle soluzioni.

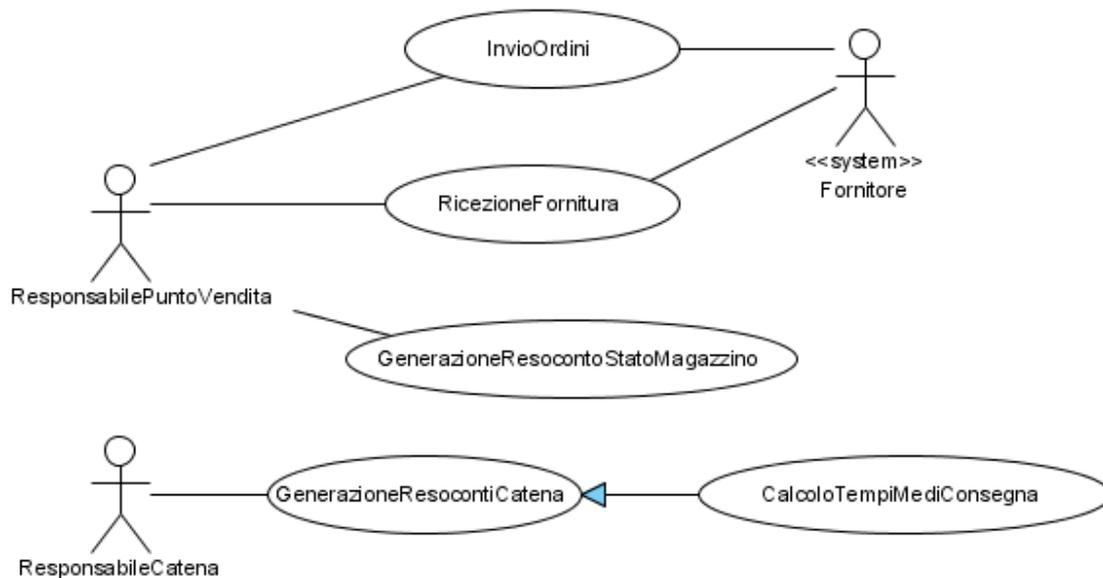
Si consideri il caso di studio 2, Grande distribuzione, e in particolare i casi d'uso: InvioOrdini, RicezioneFornitura, GenerazioneResocontoStatoMagazzino, e CalcoloTempiMediConsegna.

Domanda 1. (Analisi dei Requisiti)

- Dare un diagramma dei casi d'uso che comprenda i casi d'uso sopra elencati.
- Dare la narrativa del caso d'uso InvioOrdini, assumendo che un ordine venga fatto solo se non è possibile il trasferimento interno

Risposta:

a.



b.

Caso d'uso: InvioOrdini

Breve descrizione: *Permette di inviare un ordine di alcuni prodotti a dei fornitori*

Attore principale: *ResponsabilePuntoVendita*

Attore secondario: *Fornitore*

PreCondizioni: *Prodotti sotto scorta, impossibile il trasferimento interno*

PostCondizioni: *Prodotti ordinati*

Sequenza principale degli eventi:

1 per (ogni fornitore)

1.1 il ResponsabilePuntoVendita chiede al TradingSystem di creare un nuovo ordine per il fornitore

- 1.2 per (ogni prodotto da ordinare a quel fornitore)
 - 1.2.1 il ResponsabilePuntoVendita chiede al TradingSystem di aggiungere il prodotto all'ordine
- 1.3 il ResponsabilePuntoVendita chiede a TradingSystem l'invio dell'ordine

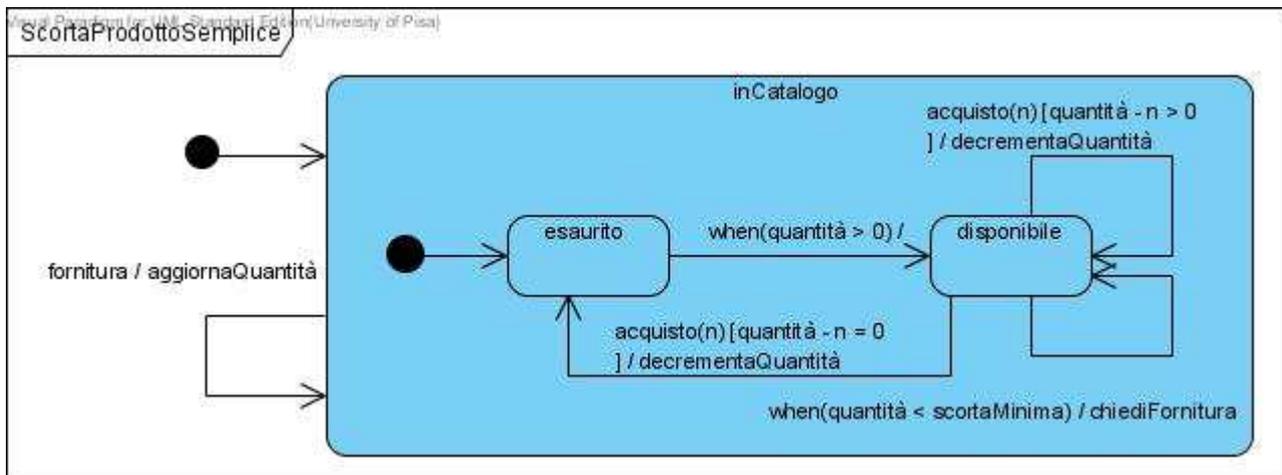
Sequenza alternativa degli eventi:

Fornitore non più convenzionato con la catena

Prodotto non più in commercio.

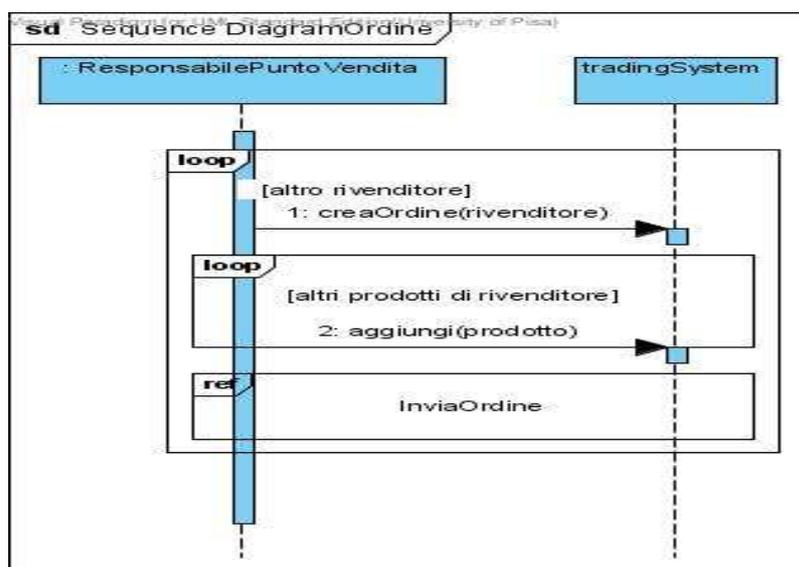
Domanda 2. (Analisi del dominio) Dare un diagramma di macchina a stati che descriva l'evoluzione della scorta di un prodotto, ignorando la possibilità di trasferire articoli da un punto vendita all'altro.

Risposta:

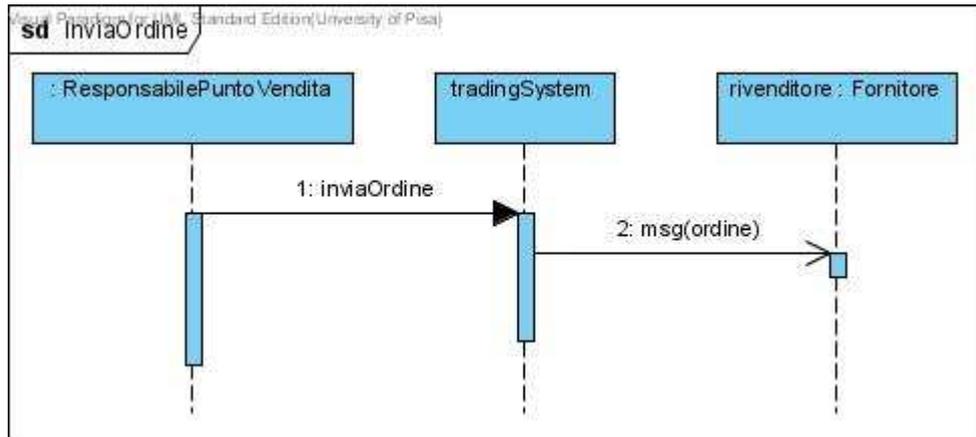


Domanda 3. (Analisi dei Requisiti) Dare un diagramma di sequenza che descriva la sequenza principale degli eventi del caso d'uso InviaOrdini.

Risposta: Una possibile soluzione è

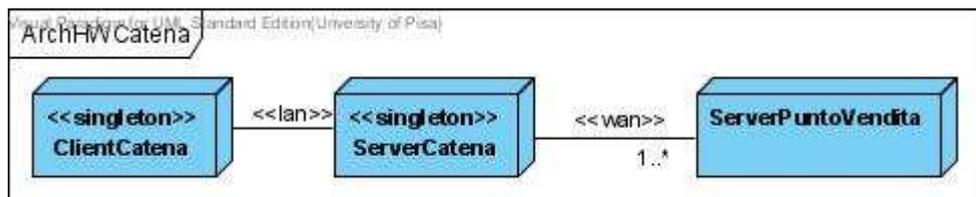
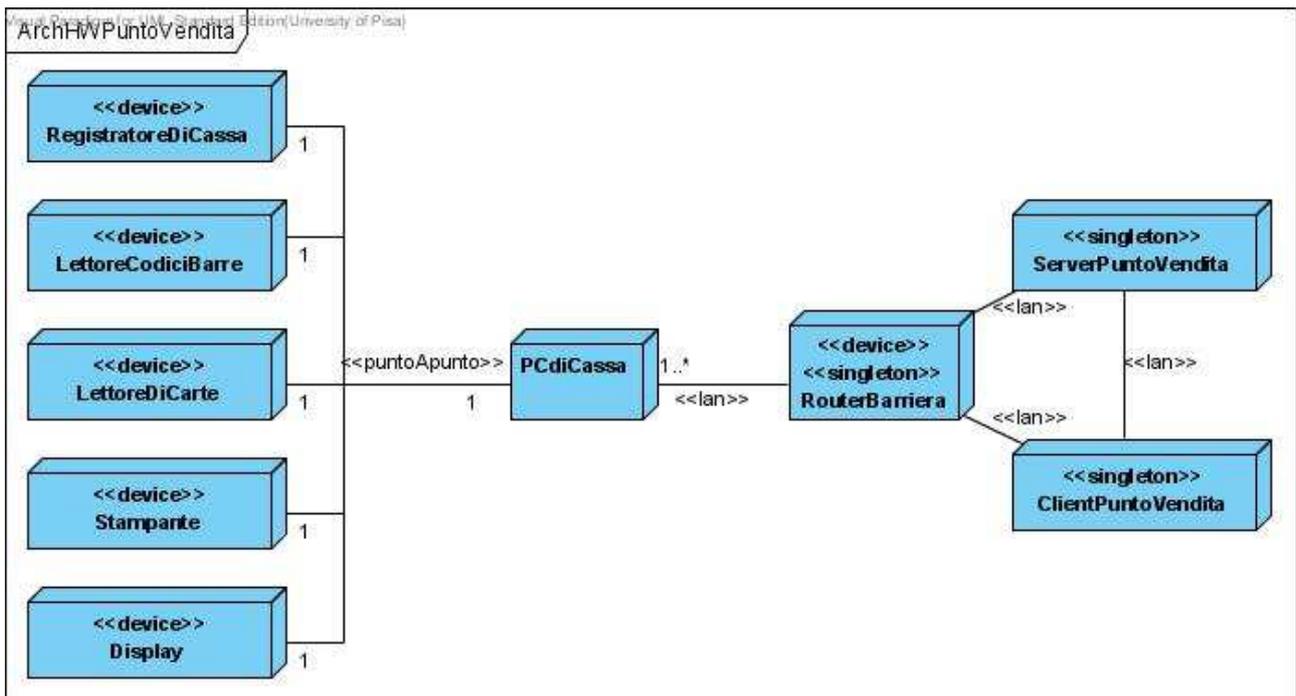


dove il frammento combinato InviaOrdine considera il rivenditore corrente nel ciclo esterno:



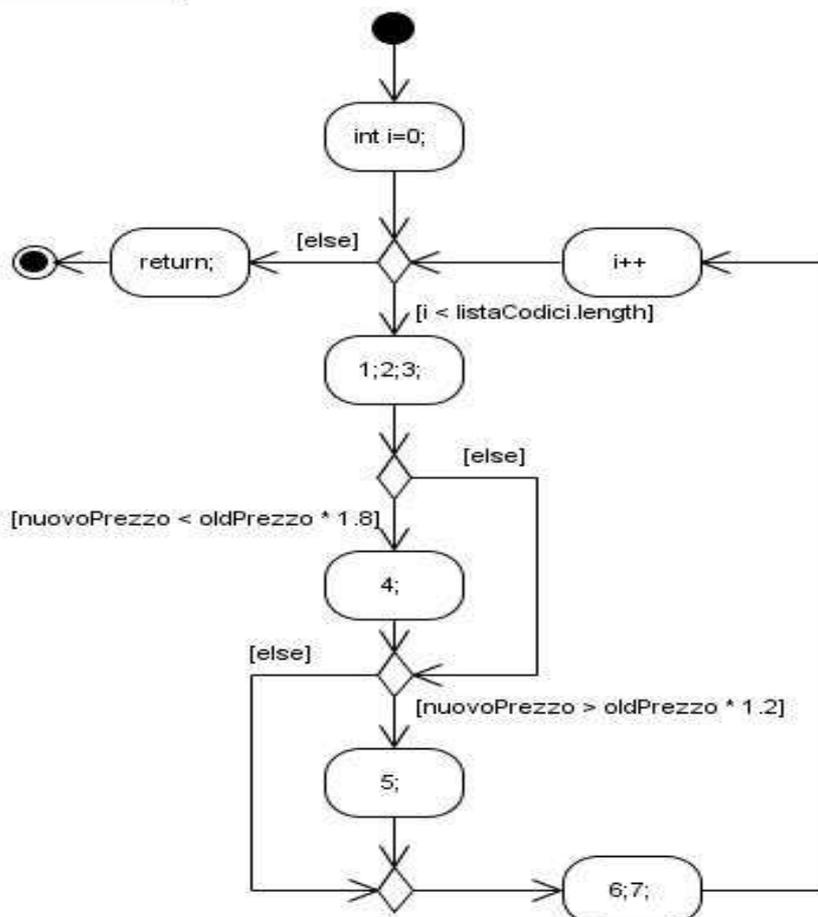
Domanda 4. (Architettura) Dare un diagramma di deployment parziale, che consideri solo la struttura hardware di TradingSystem. Si assuma l'esistenza di un router che funge da gateway per il sotto-sistema barriera.

Risposta: Si vedano i prossimi due diagrammi, che danno il punto di vista del punto vendita e della catena, rispettivamente.



Si consideri il caso d'uso **ModificaDeiPrezzi**. Il seguente metodo, con relativo grafo di flusso, realizza parte del caso d'uso: controlla la lista dei nuovi prezzi rispetto al vincolo di mantenere le variazioni entro il 20 per cento e aggiorna il data base. Le variabili cl e db denotano gli oggetti che gestiscono rispettivamente le interazioni col cliente e col data base.

```
public void controllaPrezzi(int [][] listaCodici) {  
    for(int i=0; i < listaCodici.length; i++) {  
        int codice = listaCodici[i][0];           //1  
        int oldPrezzo = db.getPrezzo(codice);     //2  
        int nuovoPrezzo = listaCodici[i][1];     //3  
        if (nuovoPrezzo < oldPrezzo * 0.8){  
            nuovoPrezzo = (int) (oldPrezzo * 0.8); //4  
        }  
        if (nuovoPrezzo > oldPrezzo * 1.2) {  
            nuovoPrezzo = (int) (oldPrezzo * 1.2); //5  
        }  
        cl.aggiornato(codice, nuovoPrezzo);      //6  
        db.setPrezzo(codice, nuovoPrezzo);      //7  
    }  
    return;  
}
```



Domanda 5. (Controllo del software)

Si assuma che lo stub per `getPrezzo` sia definito in modo da restituire un valore pari a 10 volte il suo argomento.

- a. Dare un insieme minimale di casi di test soddisfi il criterio (a scatola aperta) di copertura dei comandi.
- b. Dare un insieme minimale di casi di test soddisfi il criterio (a scatola aperta) di copertura dei cammini, nel caso di 1-test dei cicli.

Gli argomenti dei metodi `aggiornato` e `setPrezzo` costituiscono l'output del metodo. Per la risposta si utilizzino tante tabelle come la seguente quante necessarie, ciascuna con il numero di righe necessarie.

Input			Output	
i	listaCodici[i]		codice	nuovoPrezzo
	[0]	[1]		

Risposta:

a) Basta una lista con due prezzi, fuori dall'intervallo permesso, dalle due parti:

Input			Output	
i	listaCodici[i]		codice	nuovoPrezzo
	[0]	[1]		
1	2	15	2	16
2	3	40	3	36

b) Servono tre liste di un solo elemento, per fare tre cicli di un solo passo, in ciascuno dei quali si percorre uno dei tre cammini possibili nel corpo del ciclo:

Input			Output	
i	listaCodici[i]		codice	nuovoPrezzo
	[0]	[1]		
0	1	11	1	11

Input			Output	
i	listaCodici[i]		codice	nuovoPrezzo
	[0]	[1]		
0	2	15	2	16

Input			Output	
i	listaCodici[i]		codice	nuovoPrezzo
	[0]	[1]		
0	3	40	3	36