

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

Si consideri il caso di studio 2, Grande distribuzione, e in particolare il caso d'uso **ModificaDeiPrezzi**, tenendo conto che la politica della catena è che le modifiche dei prezzi possono essere fatte una sola volta nelle 24 ore e non possono superare il 20 per cento.

Caso d'uso: **ModificaDeiPrezzi**

Breve descrizione: *Permette di modificare i prezzi di vendita di alcuni prodotti e la loro visualizzazione sul display allo scaffale.*

Attore principale: *ResponsabilePuntoVendita*

Attore secondario: *Nessuno*

PreCondizioni: *Punto vendita chiuso*

PostCondizioni: *Prezzi aggiornati*

Sequenza principale degli eventi:

- 1 *per (ogni prezzo da aggiornare)*
  - 1.1 *il Responsabile richiama il prodotto*
  - 1.2 *il Responsabile indica il nuovo prezzo*
  - 1.3 *il Sistema aggiorna la scheda del prodotto*
  - 1.4 *il Sistema aggiorna il display*

Sequenza alternativa degli eventi: **Variazione di prezzo superiore al 20% per almeno un articolo.**

Breve descrizione: *Gli articoli in questione vengono modificati in più o meno 20%.*

Sequenza alternativa degli eventi: **Prezzo di almeno un articolo già modificato nelle 24 ore.**

Breve descrizione: *Il prezzo degli articoli in questione non viene modificato e il Responsabile viene avvertito per ognuno.*

\*\*\*\*\*

**Domanda 1.** (Architettura). Si consideri la parte di architettura del sistema TradingSystem che permette l'aggiornamento dei prezzi. Fornire la vista C&C considerando la seguente descrizione per le componenti, e mostrare l'interfaccia che GestionePuntoVendita deve offrire per realizzare il caso d'uso.

<b>Componente</b>	<b>Responsabilità</b>
DBPuntoVendita	Memorizza i dati sui prodotti in vendita.
GestionePuntoVendita	Gestisce l'aggiornamento dei prezzi del punto vendita.
ControlloPuntoVendita	Permette al ResponsabilePuntoVendita di accedere al sistema per aggiornare i prezzi del punto vendita.
DisplayPrezzo	Gestisce la visualizzazione dei prezzi sugli scaffali.

**Domanda 2.** (Architettura). Si consideri nuovamente la parte di architettura del sistema TradingSystem che permette l'aggiornamento dei prezzi. Fornire la vista di dislocazione. Si tenga conto che, oltre ai nodi hardware descritti nel testo, è previsto uno switch per inoltrare i messaggi ai Display, connesso via rete locale al Server e via cavo ai singoli Display. Il software dello switch è reso disponibile dal produttore, e non interessa in questa vista.

**Domanda 3.** (Progettazione di dettaglio) Fornire il diagramma di struttura composta della componente GestionePuntoVendita. Si considerino esplicitamente anche le interazioni con l'orologio, necessarie per evitare doppie modifiche di prezzo nelle 24 ore.

**Domanda 4.** (Progettazione di dettaglio) Fornire un diagramma di sequenza che mostri come le parti della componente GestionePuntoVendita cooperino per realizzare il caso d'uso Modifica-DeiPrezzi, incluse le sequenze alternative degli eventi.

\*\*\*\*\*

Il seguente metodo è di fantasia e non realizza esattamente il caso d'uso. E' stato scritto per controllare la lista dei nuovi prezzi rispetto al vincolo di mantenere le variazioni entro il 20 per cento e aggiornare 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)

Dare un diagramma di attività con il grafo di flusso del metodo. Si usi la numerazione indicata per non ricopiare i comandi. Inoltre, per effettuare i test, si assuma che lo stub per getPrezzo sia definito in modo da restituire un valore pari a 10 volte il suo argomento. Dare un (solo!) caso di test che soddisfi il criterio (a scatola aperta) delle decisioni. Gli argomenti dei metodi aggiornato e setPrezzo costituiscono l'output del metodo. Si utilizzi una tabella come la seguente per la risposta, con il numero di righe necessarie.

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