

Ingegneria del Software

23. Esercizi riassuntivi

Dipartimento di Informatica
Università di Pisa
A.A. 2014/15

un esercizio cumulativo

LA GRANDE DISTRIBUZIONE

Descrizione del problema.

L'esempio descrive un sistema per il commercio, chiamato TradingSystem, come lo si può osservare in un supermercato. Comprende sia i processi alla cassa, tipo la scansione dei prodotti usando un lettore di codice a barre e il pagamento con carta di credito o contanti, sia compiti di gestione quali l'ordine di prodotti quando si stanno esaurendo le scorte o la generazione di resoconti.

La cassa è il posto dove il cassiere scandisce i prodotti che il cliente vuole comprare e dove avviene il pagamento, in contanti, con bancomat, o con carta di credito. Una cassa può cambiare modalità da normale a rapida, e accettare solo clienti con un limitato numero di prodotti, e pagamento in contanti. Per gestire i processi alla cassa sono necessari alcuni dispositivi hardware: un PC, un registratore di cassa, un display, una stampante (per le ricevute di bancomat e della carta di credito), un lettore di carte con tastierino e un lettore di codici a barre.

Il registratore di cassa permette di iniziare e finire un acquisto, e di gestire un pagamento in contanti. Per i pagamenti con carta di credito o bancomat si usa il lettore di schede. Il cassiere identifica tutti gli articoli che il cliente intende comprare con il lettore di codici a barre. Alla fine del pagamento, si produce lo scontrino con la stampante. Ogni cassa ha un display che segnala se la cassa sta operando in modo "espresso" o normale. Il cassiere può tornare al modo normale da quello espresso premendo un apposito pulsante.

L'unità centrale di ogni cassa è il PC di cassa, che connette tutti gli altri componenti. Anche il software responsabile della gestione del processo di vendita, ivi comprese le comunicazioni con la banca, gira su questa macchina.

Un punto vendita contiene diverse casse, collettivamente chiamate barriera casse. La barriera casse è collegata al ServerDelPuntoVendita, a sua volta collegato al ClientDelPuntoVendita. Il ServerDelPuntoVendita mantiene l'inventario del corrispondente punto vendita. Il ClientDelPuntoVendita viene usato dal responsabile del punto vendita per modificare i prezzi di vendita (a negozio chiuso) e per fare gli ordini. La modifica di un prezzo si riflette automaticamente nella modifica del prezzo esposto su un piccolo display posto sugli scaffali dove sono esposti i prodotti. Quando dei prodotti arrivano al punto vendita, il responsabile controlla la corrispondenza rispetto all'ordine fatto prima di inserirli nell'inventario. Inoltre, il TradingSystem permette al responsabile del punto vendita di generare resoconti sullo stato del magazzino.

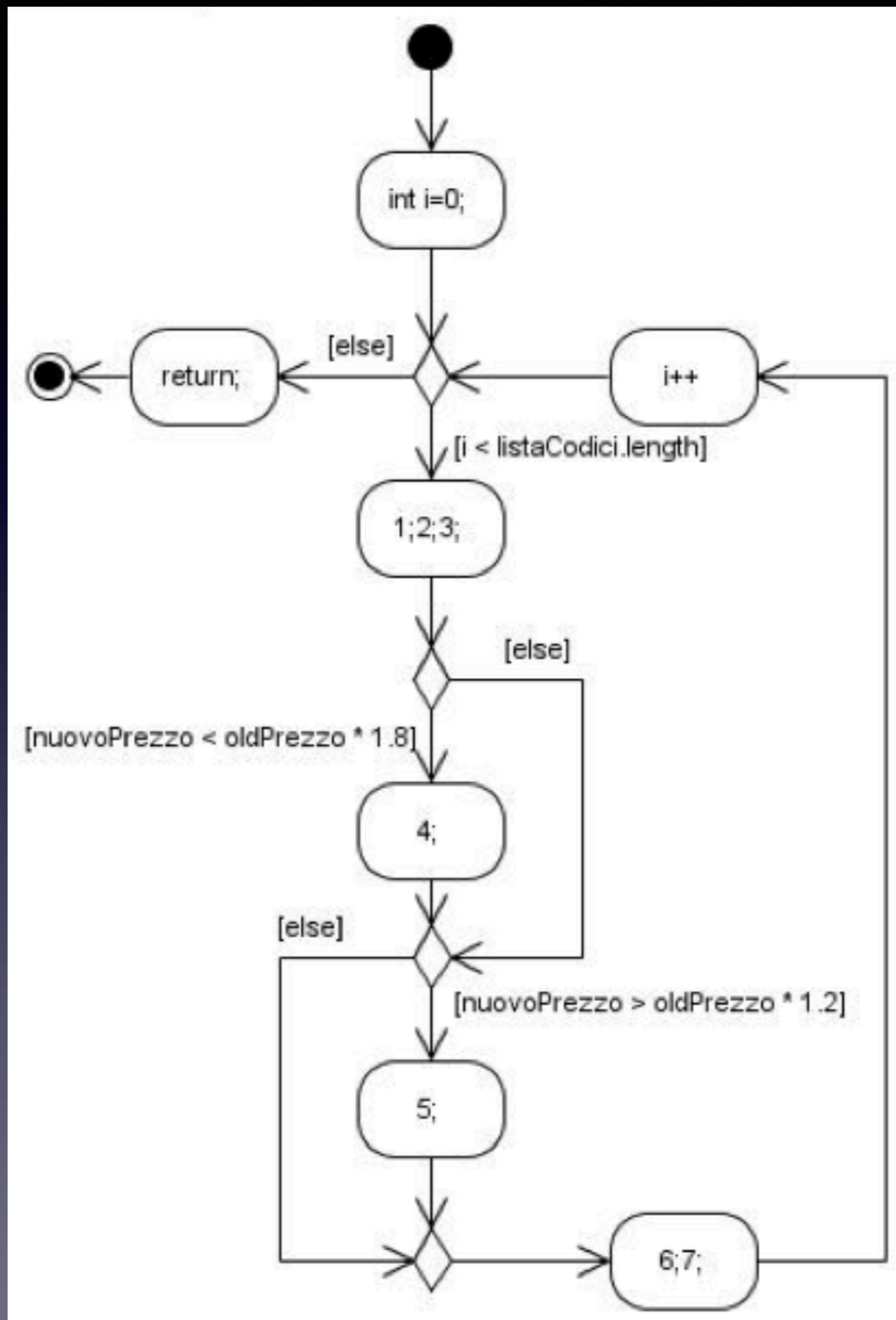
Un insieme di punti vendita è organizzato in una catena di grande distribuzione: un ServerDellaCatena è collegato a tutti i punti vendita. Usando un ClientDellaCatena, il responsabile della catena può generare diversi tipi di resoconti. Ad esempio, TradingSystem permette di calcolare i tempi medi di consegna da un fornitore ai punti vendita.

Se un negozio finisce le scorte di un prodotto, viene mandata una query per sapere se il prodotto è disponibile in qualche altro punto vendita della catena. A questo punto, il ServerDellaCatena deve sincronizzarsi con i ServerDiPuntoVendita per avere dati globali aggiornati. Dopo che una query ha avuto successo, i prodotti richiesti sono inviati da un punto vendita ad un altro. Deve però essere valutato, con l'uso di euristiche, se il trasporto è utile. Per esempio, se il prodotto scarseggia in tutti i punti vendita e si presume finirà prima della successiva fornitura, non ha senso effettuare il trasporto. Una volta che il trasporto è stato deciso, la quantità scambiata è inserita come "in arrivo" al punto vendita ricevente. Quando la merce arriva al punto vendita, viene rimossa l'etichetta "in arrivo".

grafo di flusso...

- Si fornisca un diagramma di attività (mediante grafo di flusso) del seguente metodo per il controllo della correttezza dei cambi di prezzo

```
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;  
}
```



...e (tabella dei) test

- Si assuma che lo stub per `getPrezzo` sia definito in modo da restituire un valore pari a 10 volte il suo argomento, e che gli argomenti dei metodi `aggiornato` e `setPrezzo` costituiscano l'output del metodo

input			output	
i	lista codici[i]		codice	nuovoPrezzo
	codici[0]	codici[1]		

le domande!

1. Si fornisca un (solo!) caso di test che soddisfi il criterio (a scatola aperta) delle decisioni
2. Si fornisca un insieme minimale di casi di test soddisfi il criterio (a scatola aperta) di copertura dei comandi
3. Si fornisca un insieme minimale di casi di test soddisfi il criterio (a scatola aperta) di copertura dei cammini, nel caso di 1-test dei cicli

prima domanda

input			output	
i	lista codici[i]			
	codici[0]	codici[1]	codice	nuovoPrezzo
0	1	11	1	11
1	2	15	2	16
2	3	40	3	36

un possibile caso di test

seconda domanda

input			output	
i	lista codici[i]			
	codici[0]	codici[1]	codice	nuovoPrezzo
1	2	15	2	16
2	3	40	3	36

basta una lista con due prezzi, fuori dall'intervallo permesso, dalle due parti

terza domanda

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