

<p align="center">Corso di INGEGNERIA DEL SOFTWARE Corsi di Laurea in INGEGNERIA INFORMATICA e in INGEGNERIA DELL'AUTOMAZIONE A.A. 2014-2015</p> <p align="center">ESAME DEL 7 LUGLIO 2015 DURATA DELLA PROVA: 2,5 ORE</p>	<p align="center"><u>STUDENTE</u></p> <p>Cognome e Nome:</p> <p>Matricola:</p>
--	--

Prima Parte della Prova

1. Si consideri il seguente documento di descrizione informale dei requisiti:

Si vuole realizzare un'applicazione per la gestione di una boutique di fiori.

Il proprietario vuole tener traccia delle specie di piante e di fiori (recisi) disponibili in vendita (es. pianta *ficus benjamina* oppure fiore *rosa rossa*). Di tutti i prodotti (piante e fiori) occorre memorizzare il nome, l'origine geografica, il prezzo unitario e la quantità disponibile in negozio. Inoltre, delle piante si specifica la loro tipologia (es. se sono da esterno, o da interno), mentre per i fiori si specifica il loro colore.

Il proprietario può aggiungere una specie di pianta o fiori al sistema, e può modificare la quantità disponibile in negozio per la vendita. Il sistema non consente di aggiungere una specie di pianta o fiore se coincide per nome con un'altra specie già memorizzata.

Interagendo col sistema, il proprietario può inoltre effettuare ordini di piante e fiori, i quali sono inviati elettronicamente al vivaio. Ciascun ordine specifica i prodotti richiesti (ovvero una o più specie di piante e fiori) e, per ciascuna specie, le quantità necessarie. Di un ordine il sistema registra la data di creazione e il suo stato (in attesa, consegnato). Quando il proprietario riceve fisicamente la consegna in negozio, egli notifica il sistema della ricezione dell'ordine. Alla notifica di ricezione di un ordine, il sistema aggiorna lo stato dell'ordine (che da *in attesa*, evolve in *consegnato*) e procede all'aggiornamento della disponibilità dei prodotti in negozio: per ogni specie di pianta o fiore segnata sull'ordine, il sistema somma la quantità richiesta nell'ordine alla quantità al momento disponibile in negozio.

Anche i dipendenti del negozio interagiscono col sistema. Quando uno o più prodotti sono venduti, i dipendenti decrementano le quantità disponibili dei prodotti appena usciti dal negozio. Inoltre, i dipendenti aggiungono al sistema le composizioni floreali da loro create: una composizione floreale è formata da più specie di fiori, unite ognuna in un preciso numero, e disposte in una forma artistica. Di una composizione floreale, il sistema deve memorizzare le specie e il numero di fiori usati per formarla, il nome della composizione floreale e il prezzo di vendita. I dipendenti possono inoltre modificare o rimuovere composizioni floreali esistenti.

Per tale sistema, si realizzino in linguaggio UML:

- il diagramma dei casi d'uso

- il diagramma delle classi raffinato, che mostri anche attributi e responsabilità attribuite alle classi;

Suggerimento: si progetti applicando le buone norme dell'ingegneria del software, utilizzando in maniera opportuna le classi «boundary», «control».

- il diagramma di sequenza raffinato della funzionalità di decremento della disponibilità di un prodotto in negozio.

Seconda Parte della Prova

Lo studente produca autonomamente i seguenti artefatti, da presentare sia in forma di relazione stampata sia in formato elettronico per la discussione alla successiva prova orale.

2. Si implementi in linguaggio Java la parte del sistema necessaria a realizzare la funzionalità di **decremento della disponibilità di un prodotto in negozio**.

- Si realizzi con JDBC una classe DAO per una delle entità persistenti del sistema.

A corredo della relazione, lo studente dovrà consegnare per la discussione alla prova orale:

- I file del progetto (comprese le eventuali librerie – è consigliato l'uso di Eclipse per la gestione dei file del progetto). Come nome del progetto si indichi il proprio cognome seguito dal numero di matricola completo (ad esempio: ROSSI_N46000000)
- Una base di dati che permetta di esercitare la classe DAO implementata. È consigliato scegliere come DBMS di supporto H2 (<http://www.h2database.com/>) o Microsoft Access.

3. Si progetti ed esegua un insieme di casi di test black box per la funzionalità **di decremento della disponibilità di un prodotto in negozio**, tenendo conto anche delle diverse pre-condizioni per tale funzionalità.

Si usi lo schema seguente per definire i casi di test.

T C	Descrizione Test Case	Classi di Equivalenza coperte	Precondizioni	Input	Output Attesi	Post- Condizioni Attese	Output Ottenuti	Post- Condizioni Ottenute	Esito

Lo studente dovrà consegnare:

- Un file di testo (formato .DOC, .RTF o .XLS) contenente la tabella dei casi di test progettati;
- Un Progetto di Test in JUnit contenente i casi di test necessari a testare la funzionalità richiesta;
- Esito dei test eseguiti con JUnit e opzionalmente i livelli di copertura del codice raggiunti.
- il *Class diagram* di progettazione del sistema; si può trascurare in questo diagramma la gestione della persistenza (perché si suppone sia realizzata da un framework esterno), l'interfacciamento con gli utenti e le politiche di security (autenticazione con gli utenti);
- il *Sequence diagram* di progettazione per la realizzazione della funzionalità **di decremento della disponibilità di un prodotto in negozio**.