

Esercitazione

Programmazione Orientata agli Oggetti

23/11/2017 – tempo a disposizione 2h

La ditta OverOriginalPartners gestisce una serie di prodotti da vendere. Ciascun prodotto è fornito da un solo fornitore. I clienti della ditta possono effettuare degli ordini che consistono in una serie di prodotti con le relative quantità.

Le informazioni da memorizzare nell'applicazione sono le seguenti.

- *Prodotto*: con codice, descrizione, fornitore, prezzo di vendita, numero di pezzi disponibili;
- *Ordine*: con numero progressivo, cliente e lista di prodotti con le relative quantità ordinate (*);
- *Fornitore*: con nome, cognome, indirizzo e Partita IVA;
- *Cliente*: con nome, cognome, indirizzo e codice fiscale;

Vincolo da rispettare: ciascun ordine non deve contenere elementi diversi che fanno riferimento allo stesso prodotto. Nel caso in cui si tenti di inserire un prodotto con una certa quantità in un ordine in cui tale prodotto è già esistente, la quantità indicata dovrà essere sommata a quella già presente nell'ordine.

I fornitori ed i clienti devono poter essere inseriti nel sistema dall'utente tramite un'interfaccia grafica strutturata come in figura.

The screenshot shows a Java Swing window titled "Aggiungi Cliente/Fornitore". It is divided into two main sections. The left section is for adding a client, with labels for "Nome Cliente", "Cognome Cliente", "Indirizzo Cliente", and "Codice Fiscale Cliente", each followed by a text input field. Below these fields is a button labeled "Aggiungi Cliente". The right section is for adding a supplier, with labels for "Nome Fornitore", "Cognome Fornitore", "Indirizzo Fornitore", and "Partita Iva Fornitore", each followed by a text input field. Below these fields is a button labeled "Aggiungi Fornitore". At the bottom center of the window is a button labeled "Aggiunta Terminata".

Alla pressione del tasto "Aggiungi Cliente" e "Aggiungi Fornitore" devono essere effettuati i seguenti controlli:

- I campi Nome, Cognome, Indirizzo, Codice Fiscale (o Partita Iva) devono essere valorizzati
- Se Fornitore, il campo Partita Iva deve essere composto da 11 caratteri numerici
- Se Cliente il campo Codice Fiscale deve essere composto da 16 caratteri alfanumerici di cui: 6 caratteri alfabetici + 2 caratteri numerici + 1 carattere alfabetico + 2 caratteri numerici + 1 carattere alfabetico + 3 caratteri numerici + 1 carattere alfabetico

Se i controlli non vanno tutti a buon fine, l'inserimento deve essere bloccato, e l'utente informato da un popup, altrimenti l'inserimento ha successo e i campi devono essere resettati per un eventuale nuovo inserimento. Quando l'inserimento di istanze è terminato, l'utente deve premere il pulsante "Aggiunta Terminata", ed il programma proseguirà la sua esecuzione.

(Suggerimento: la classe *GUIPersone* che conterrà la GUI dovrà implementare l'interfaccia *Lockable.java* fornita. I due metodi *isLocked()* e *unlock()* andranno ad interagire con una variabile booleana contenuta in *GUIPersone* che verrà opportunamente settata dal pulsante di "Aggiunta Terminata").

Sulla base delle informazioni memorizzate secondo le specifiche indicate precedentemente, è necessario effettuare le seguenti operazioni.

1. Dato un ordine, calcolare il prezzo totale
2. Dato un ordine e la lista dei prodotti della ditta, restituire la lista di prodotti presenti nell'ordine per i quali non c'è disponibilità (ovvero prodotti presenti nell'ordine in cui il numero di pezzi ordinati è maggiore del numero di pezzi disponibili). Nel caso in cui tutti i prodotti dell'ordine siano disponibili bisogna restituire una lista vuota.

Parte 1. Realizzare le classi Java necessarie a rappresentare il dominio di interesse descritto in precedenza, adottando i principi fondamentali della programmazione orientata agli oggetti e indipendentemente dalla parte successiva.

Parte 2. Realizzare l'interfaccia grafica e le sue funzionalità descritte in precedenza.

Parte 3. Realizzare una classe Operazioni contenente i seguenti due metodi statici che implementano le operazioni descritte in precedenza

```
public static double totaleOrdine(Ordine ord)
```

```
public static List<Prodotto> prodottiNonDisponibili (Ordine ord, List<Prodotto> lp)
```

() Suggerimento:*

- Per memorizzare le informazioni sui prodotti e le quantità in un ordine, si definisca una classe CoppiaProdottoQuantita contenente una coppia di valori <prodotto, quantità>.

Note:

- 1) Ridefinire opportunamente il metodo toString() in ogni classe.
- 2) Si tenga conto del programma allegato nel file **Main.java** per ulteriori specifiche sui nomi di classi e metodi.