

Esercitazione per il capitolo 10 – interfacce Java

Apportare delle modifiche alla libreria *negozioprodotti.jar*, realizzata per l'esercitazione del 31 ottobre 2023, al fine di consentire ad ogni Negozio di:

1. Tenere traccia di ogni fattura da pagare mediante un'operazione *addFattura(Fattura)* che permetta di aggiungere una fattura per volta nel negozio. A tale scopo sarà necessario modellare una nuova classe, denominata *Fattura*, in modo che:
 - ogni fattura sia caratterizzata da una quantità di articoli (intero) e da un prezzo unitario (double).
 - per ogni fattura sia possibile sapere l'importo da pagare mediante l'operazione denominata *calcolaImporto()* che, nella sua versione iniziale, restituirà come risultato il prodotto del prezzo unitario per la quantità di articoli.
2. Ottenere, mediante l'operazione denominata *getAllPagamenti()*, l'elenco dei pagamenti da effettuare, fattura o prodotti che siano.
3. Ottenere, mediante un'operazione denominata *getAllProdotti()*, l'elenco dei prodotti inseriti nel negozio rispetto ai quali sarà possibile ottenere soltanto le informazioni relative a *descrizione* e *prezzo*.
 - L'operazione *getCodice()* non deve dunque essere disponibile.
 - Il prezzo è quello ottenuto mediante l'operazione *calcolaPrezzo()*.

NOTA. Finalità dell'esercitazione

1. mostrare come creare un'interfaccia Java;
2. mostrare come usare (implementare) un'interfaccia;
3. mostrare come l'uso delle interfacce permetta a classi che non siano in relazione di ereditarietà di esporre operazioni comuni pur non condividendone le implementazioni.
 - Le interfacce consentono a classi diverse di implementare operazioni comuni senza la necessità di una relazione diretta di ereditarietà.
 - **Benefici:** le classi possono implementare un'interfaccia in modo indipendente, consentendo una progettazione più flessibile e promuovendo la separazione delle responsabilità.
4. mostrare come oggetti che implementino la stessa interfaccia possa essere trattati polimorficamente se acceduti mediante un riferimento a tale interfaccia.
 - Il polimorfismo consente agli oggetti di essere trattati in modo generico, indipendentemente dalle specifiche implementazioni delle classi.
 - **Benefici:** quando gli oggetti implementano la stessa interfaccia, possono essere acceduti uniformemente tramite un riferimento a quell'interfaccia. Questo promuove la flessibilità nel codice e la possibilità di sostituire oggetti concreti senza influire sulla logica del programma.
5. Mostrare come una classe possa implementare molteplici interfacce
 - Java supporta l'ereditarietà singola delle classi (concrete o astratte) e l'ereditarietà multipla delle interfacce.