

# **Progetto per l'esame del corso di Programmazioen ad Oggetti, a.a. 2025/2026**

Il progetto va svolto in gruppo di 2 persone (o singolarmente).

Il progetto deve essere consegnato 7 giorni prima dell'esame scritto. Metterò un contenitore sul sito per la consegna (la scadenza sarà la stessa della prenotazione all'esame).

## **Requisiti**

Lo svolgimento del progetto deve tenere in considerazione i seguenti requisiti:

- le classi del dominio del sistema devono essere corredate da classi di test;
- la documentazione deve essere prodotta con **javadoc** (trovate una sintetica descrizione di come annotare il codice sul sito del corso);
- ogni progetto deve essere accompagnato da una relazione in cui vengono illustrate le scelte adottate: le classi che sono state realizzate con il loro scopo e una guida a come fare a far girare il codice da riga di comando e da interfaccia grafica.
- inoltre per quanto riguarda l'interfaccia utente (che deve essere separata dalle classi che realizzano la logica del sistema) deve essere possibile eseguire tutte le operazioni catturando le eventuali eccezioni sollevate che vengono gestite facendo ripetere di nuovo l'operazione. Dovete avere sia
  - una interfaccia da riga di comando, che deve avere un menu in cui si presentano le operazioni che è possibile eseguire e il risultato della loro esecuzione;
  - una semplice interfaccia grafica, fatta usando le librerie swing che permetta di fare le stesse operazioni dell'interfaccia da riga di comando.

Per l'interfaccia utente che deve essere rigorosamente separata dall'interfaccia utente dovete usare il MVC che vedremo nelle prossime lezioni. Per cui per il momento fate solo le classi di test.

Nello svolgere il progetto usare i principi visti nel corso. In particolare, buona strutturazione delle classi. Ogni classe ha un compito preciso (e fa solo quello), ha campi privati, uso di eccezioni e non "stampe" per segnalazioni di malfunzionamenti, metodi di dimensione limitata, non ripetizioni di codice, ecc...  
Mantenete separate le classi del dominio dall'interfaccia grafica/testuale

## **Valutazione**

La discussione del progetto con i singoli studenti verterà sulle modalità di implementazione adottate e sulla padronanza di alcuni dei concetti necessari per realizzare il progetto e/o spiegati a lezione. La valutazione del progetto sarà fatta in base alla: conformità dell'implementazione scelta per risolvere il problema con il paradigma di programmazione a oggetti e alla conformità del codice presentato alle regole di buona programmazione.

## Progetto: ListaDiArticoli

Realizzare un programma per mantenere varie **liste di articoli** (ad esempio liste della spesa per supermercato o del negozio sotto casa, degli articoli da comprare da Ikea, ...), che hanno un nome e un contenitore degli **articoli** che devono essere comprati. Gli articoli sono classificati secondo **categorie** definite dall' utente e hanno associato un **prezzo** e una **nota** (una stringa facoltativa che li descrive). Le categorie sono condivise da tutte le liste come pure gli articoli.

L'associazione fra un articolo e una categoria può essere *modificata*. Quando *inserite* un articolo se per questo non è specificata la categoria o il prezzo o la nota potete usare dei default (ad esempio "Non categorizzato", prezzo 0 e nota ""). Categoria, prezzo e nota di un articolo possono essere modificate. La *rimozione* di un articolo passa l'articolo in una **lista dei cancellati** (ogni lista di articoli ha i suoi articoli cancellati). Potete *selezionare* un articolo dalla lista dei cancellati e riportarlo nella lista della spesa. La lista dei cancellati di può essere svuotata. Oltre a inserire e cancellare da una lista deve essere possibile data una stringa trovare un articolo che ha la stringa come prefisso, cercando sia nella lista della spesa che in quella dei cancellati (non deve poter essere in entrambe) e calcolare il prezzo totale degli articoli nella lista.

La lista della spesa deve essere iterabile (iterando sia sugli articoli della lista che su quelli cancellati) per permettere ulteriori elaborazioni da parte del codice cliente.

Ci deve and essere una classe GestioneListe che contiene: un campo statico *listeArticoli* con l'associazione fra il nome di una lista e un riferimento a un oggetto di tipo ListaDiArticoli e un campo statico *categorie* che lista tutte le categorie e infine un campo *articoli* con l'ovvio contenuto e i metodi per creare una lista, cancellare una lista e per inserire e cancellare una categoria e inserire a cancellare un articolo.

Se volete potete aggiungere altre funzionalità, ma questo non aumenta il voto.