

# Progetto di Paradigmi: Programmazione ad Oggetti 2024/2025

Il progetto va svolto o singolarmente o in gruppo di al più 2 persone. Per avere un gruppo da 3 si deve avere una buona motivazione e la mia autorizzazione.

Il progetto deve essere consegnato una settimana prima dell'esame scritto (ci sarà un contenitore sul sito del corso per la consegna) insieme ai contenitori per le prenotazioni.

## 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 e le loro relazioni e metodi pubblici, a cosa servono, le eccezioni che sollevano e perché le sollevano
- inoltre per quanto riguarda l'interfaccia utente (che deve essere separata dalla 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 semplice interfaccia grafica, che
  - una interfaccia da riga di comando.

Nello svolgere il progetto usare i principi visti nel corso. In particolare, buona strutturazione delle classi, uso di classi astratte e interfacce se necessario. 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.

## Realizzare un'applicazione per gestire una Bacheche di Annunci

Progettare ed implementare in Java un sistema per la gestione di una **bacheca** di **annunci**. In ogni annuncio un **utente** (una persona identificata da **email** e che ha un **nome**) **cerca** o **vende** un oggetto, specificandone il **prezzo**, alcune **parole chiave** e nel caso di annuncio di vendita una **data di scadenza**. (Siete liberi di aggiungere altre informazioni agli utenti e agli annunci.) Deve essere fornito un metodo, **pulisciBacheca**, che quando chiamato rimuova dalla bacheca gli annunci di vendita scaduti. Una bacheca deve poter essere memorizzata e letta da file ed inoltre è obbligatorio che provveda un iteratore in modo tale che si possano scorrere i suoi annunci (ma non cancellarli).

Un utente può:

- Inserire una nuova parola chiave.
- Inserire un annuncio nella bacheca. Per inserire un annuncio deve specificare
  - se si vuole acquistare o vendere
  - nome dell'articolo
  - prezzo
  - zero o più parole chiave (stringa con parole chiave separate da un separatore)
  - nel caso di vendita la data di scadenza

In caso l'articolo sia un acquisto oltre ad aggiungere l'annuncio nella bacheca, viene ritornata una lista degli articoli che hanno parole chiave che intersecano quelle specificate dell'utente.

- Rimuovere annuncio. Questo può essere fatto solo dall'utente che ha inserito l'annuncio. La rimozione specifica (Agli annunci viene assegnato un identificativo, quando vengono creati e inseriti nella bacheca.)
- Cercare articoli per parola chiave. Si devono poter specificare più parole chiave e viene ritornata la lista degli articoli che hanno parole chiave che intersecano quelle specificate dell'utente. (Come per l'inserimento!)
- Chiedere di ripulire la bacheca dagli annunci scaduti

(Se volete potete implementare altre operazioni.)