

## RELAZIONE PROGETTO IN JAVA

Il progetto implementa una bacheca di annunci in cui gli utenti possono interagire. Il codice è scritto in Java e segue i principi della programmazione ad oggetti. La struttura del progetto è organizzata nei seguenti pacchetti:

- **main:** contiene la classe Main, permette la scelta tra interfaccia grafica e interfaccia da riga di comando.
- **modello:** include le classi fondamentali per la gestione della bacheca:
  - Annuncio, Utente e Bacheca.
  - `modello.exception`: contiene le eccezioni personalizzate utilizzate nel sistema.
  - `modello.test`: include le classi di test per verificare il corretto funzionamento delle classi principali.
- **interfaccia:** suddiviso in due sotto-pacchetti:
  - `interfaccia.rigaDiComando`: gestisce l'interazione con l'utente tramite linea di comando.
  - `interfaccia.grafica`: contiene la gestione della GUI ed è suddiviso in:
    - `interfaccia.grafica.vista`: gestisce la parte grafica dell'applicazione (finestre, bottoni, ecc...).
    - `interfaccia.grafica.controllo`: implementa il controller che gestisce la logica dell'interfaccia utente.

### Classe Annuncio

La classe Annuncio rappresenta un annuncio pubblicato sulla bacheca. Ogni annuncio è caratterizzato da:

- Autore: Istanza della classe Utente che ha creato l'annuncio.
- Articolo: Nome o descrizione dell'oggetto in vendita o da acquistare.
- Prezzo: Valore economico dell'articolo.
- Parole chiave: Insieme di stringhe associate all'annuncio per facilitarne la ricerca.
- Data di scadenza (solo per annunci di vendita): Utilizzata per determinare se l'annuncio è ancora valido. Implementata con `LocalDate` della libreria `java.time`.
- ID: Identificatore univoco dell'annuncio, generato automaticamente alla creazione. Se l'annuncio viene caricato da file, l'ID viene mantenuto. Per questo motivo, la classe presenta due costruttori: uno con il campo ID e uno senza.

La classe include i seguenti metodi:

- `generalId`: Genera un ID casuale per l'annuncio.
- `validaArticolo`: Verifica la validità del nome dell'articolo che si vuole istanziare.
- `validaPrezzo`: Verifica che il prezzo sia strettamente positivo.
- `validaTipologia`: Verifica che la tipologia sia o "acquisto" o "vendita".
- `validaParoleChiave`: Verifica che le parole chiavi inserite rispettino la propria espressione regolare.
- `validaDataScadenza`: Verifica che per gli annunci di tipo vendita, la data di scadenza non sia null e che rispetti la propria espressione regolare. Per gli annunci di tipo acquisto ritorna null.
- `isScaduto`: Verifica se la data di scadenza degli annunci di tipo vendita sia già passata rispetto al giorno di oggi.
- Override dei metodi `toString` ed `equals`; getters dei vari campi.

## Classe Utente

La classe Utente rappresenta un utente che può inserire e rimuovere annunci. Gli utenti sono identificati da:

- Nome
- Email

La classe include i seguenti metodi:

- isEmailValida: verifica che la email inserita rispetti la sua espressione regolare
- isNomeValido: verifica che il nome inserito rispetti la sua espressione regolare
- Override dei metodi toString ed equals; Getters dei vari campi

## Classe Bacheca

La classe Bacheca gestisce l'insieme degli annunci pubblicati. La bacheca è composta da:

- ArrayList<annuncio> annunci (lista degli annunci)
- ArrayList<Integer> poolId (lista degli id presenti in bacheca per gestire la unicità di essi)

La classe include i seguenti metodi:

- aggiungiAnnuncio: Aggiunge un nuovo annuncio alla bacheca, se non è già presente. Se l'annuncio è di tipo "acquisto", cerca nella bacheca articoli in vendita con parole chiave in comune.
- rimuoviAnnuncio: Rimuove un annuncio dalla bacheca, se l'utente è l'autore dell'annuncio.
- cercaPerParolaChiave: Cerca gli annunci che contengono almeno una delle parole chiave specificate.
- pulisciBacheca: Pulisce la bacheca rimuovendo gli annunci scaduti.
- aggiungiNuovaParolaChiave: Aggiunge una nuova parola chiave a un annuncio esistente, se l'utente è l'autore.
- salvaAnnunciSuFile: Salva gli annunci presenti nella bacheca in un file.
- caricaAnnunciDafile: Carica gli annunci da un file alla bacheca.
- generaAutore: usato dal metodo caricaAnnunciDaFile per istanziare un utente da una stringa
- controlloIdPresente: Controlla se l'id passato come argomento è presente nella pool degli id
- Override del metodo toString; Getters dei vari campi

La classe Bacheca implementa inoltre l'interfaccia Iterable<Annuncio>, consentendo di iterare sugli annunci senza modificarli, perciò è stato fatto l'override del costruttore di iterator e dei metodi hasNext e next.

## Classi per le Eccezioni

Sono state realizzate le seguenti classi che estendono la classe Exception, per gestire le eccezioni:

- AnnuncioException: viene lanciata quando avvengono errori nella creazione di un tipo Annuncio.
- UtenteException: viene lanciata quando avvengono errori nella creazione di un tipo Utente.
- BachecaException: viene lanciata quando avvengono errori nella creazione di un tipo Bacheca.
- AutoreNonAutorizzatoException: viene lanciata quando un Utente cerca di rimuovere o modifica un annuncio di cui non è lui stesso l'autore.

## Interfaccia Grafica

Per la gestione dell'interfaccia grafica, è stata adottata l'architettura MVC (Model-View-Controller). Sono state utilizzate le seguenti librerie: javax.swing e java.awt.