



Programmazione ad Oggetti

ItemTracker

1. Obiettivo del Progetto

Il sistema **ItemTracker** permette la gestione di liste multiple. Il programma consente di creare categorie personalizzate, definire un'anagrafica di articoli e organizzare tali articoli in liste distinte. Il sistema garantisce la persistenza dei dati.

2. Architettura del Progetto

Le classi del dominio non hanno alcuna dipendenza da librerie grafiche:

- **Articolo:** Rappresenta l'entità base con nome, prezzo, nota e categoria. Gestisce i valori di default.
- **Categoria:** Definisce la classificazione degli articoli.
- **ListaDiArticoli:** Rappresenta l'oggetto che gestisce la lista vera e propria suddivisa in `articoliAttivi` e `articoliRimossi`. Implementa l'interfaccia `Iterable` per consentire il ciclo su tutti gli articoli e il metodo per la ricerca per prefisso.
- **GestioneListe:** La classe "registra" che mantiene l'associazione tra nomi e liste tramite una `Map`, oltre che alle liste contenenti tutti gli articoli e tutte le categorie.

Le interfacce disponibili per utilizzare l'applicazione sono 2:

- **Interfaccia Grafica (Swing):** Composta da `MainGui` e dai pannelli specializzati (`PannelloArticoli`, `PannelloCategorie`, `PannelloListe`).
 - **Interfaccia Testuale:** Un controller da console che offre un menu creato tramite case per eseguire le medesime operazioni della GUI.
-

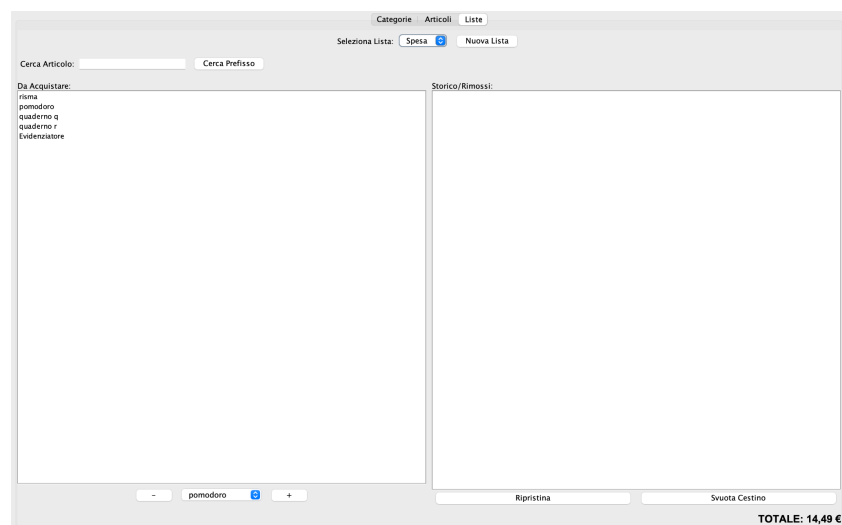
3. Scelte Implementative e Design Pattern

- **Persistenza:** È stata utilizzata la **Serializzazione Java** (`ObjectOutputStream` / `ObjectInputStream`) per salvare lo stato dell'intero sistema su un file binario (`.ser`). Questa scelta permette di mantenere le relazioni complesse tra oggetti (es. articoli legati a categorie) senza database esterni.
- **Gestione Errori:** Sono state create **Eccezioni Personalizzate** (`ElementoNonTrovatoException`, `ElementoDuplicatoException`, ecc.). Questo permette alla logica di segnalare malfunzionamenti in modo preciso senza usare `System.out.print("ERRORE");`.
- **Collezioni:**
 - `HashMap` per le liste (ricerca rapida per nome).
 - `HashSet` per le categorie (per evitare duplicati).
 - `ArrayList` per gli articoli (mantiene l'ordine di inserimento).

4. Guida all'Esecuzione

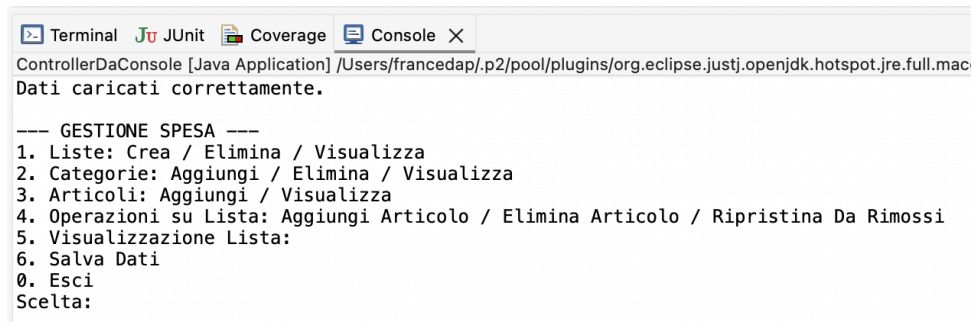
Esecuzione da Interfaccia Grafica

1. Individuare il pacchetto `controllerGui`.
2. Eseguire la classe `MainGui`, altrimenti tramite CLI selezionando il Case adibito all'apertura della GUI.
3. Utilizzare le schede in alto per navigare tra la gestione delle categorie, la creazione degli articoli e la composizione delle liste.



Esecuzione da Riga di Comando

1. Usare il Run As all'interno del package `controllerDaConsole` all'interno della classe `ControllerDaConsole.java`
2. All'interno della Console seguire le istruzioni del menu numerico per interagire con il sistema.



```
Terminal  JUnit  Coverage  Console X
ControllerDaConsole [Java Application] /Users/francedap/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64.jre/Contents/Home/bin/java
Dati caricati correttamente.

--- GESTIONE SPESA ---
1. Liste: Crea / Elimina / Visualizza
2. Categorie: Aggiungi / Elimina / Visualizza
3. Articoli: Aggiungi / Visualizza
4. Operazioni su Lista: Aggiungi Articolo / Elimina Articolo / Ripristina Da Rimossi
5. Visualizzazione Lista:
6. Salva Dati
0. Esci
Scelta:
```

5. Documentazione e Test

- **Javadoc:** Tutta la documentazione tecnica è stata generata automaticamente tramite Javadoc ed è disponibile nella cartella `/doc` allegata.
- **Test:** Il corretto funzionamento del Model è garantito dalle classi di test JUnit situate nel pacchetto `test`, che verificano i calcoli dei totali, la ricerca per prefisso e la corretta gestione dei cancellati.