

Primo Appello - Prova di Livello Base-Intermedio

Cognome e Nome: _____ Matricola: _____

Tempo a disposizione: 3 ore

È necessario scrivere un'applicazione che simula il frammento di un sistema informativo di compravendita di auto. Ogni auto ha una targa, modello, anno di prima immatricolazione. Ad esempio, una possibile auto è {“FG093XF”, “Audi X8”, 2023}. Per ogni auto bisogna tener traccia delle operazioni di compravendita associate all'auto. Ogni operazione ha la data, il numero di km che aveva l'auto in quella data, l'importo di vendita, il nome del nuovo proprietario. Ad esempio, all'auto “FG093XF” sono associate due operazioni di compravendita:

1. 01/01/2023 - 0km - 38500 euro - Mario Rossi
2. 04/03/2024 - 15400 km - 25000 euro – Rocco Verdi

L'applicazione deve consentire di svolgere i seguenti casi d'uso:

“Utente carica un archivio”

- l'utente carica l'archivio *NOTA: per velocizzare le operazioni, si suggerisce di utilizzare un "mock object" per simulare il funzionamento del DAO che carica i dati dal disco*

“Utente cerca automobili”

- l'utente inserisce una data e un criterio di ordinamento, che può essere “nome crescente” o “nome decrescente”. *NOTA: per la data utilizzare tre campi di testo, per il giorno, mese e anno. Il campo dell'anno dev'essere pre-popolato con tutti i valori degli anni dal 1900 fino all'anno corrente.*
- il sistema mostra in una tabella la lista delle auto che non hanno nessuna operazione di compravendita successiva alla data scelta, ordinato per criterio scelto dall'utente

“Utente inserisce nuova operazione”

- l'utente seleziona un'automobile dai risultati
- il sistema mostra, in un nuovo pannello, i dati dell'automobile
- il sistema calcola e mostra inoltre, per ogni anno in cui c'è stata una operazione di compravendita, il numero di operazioni effettuate in quell'anno e l'importo medio delle operazioni. I dati saranno mostrati in una tabella ordinata per anno crescente. Un esempio di tabella è la seguente

Anno	Numero di operazioni	Importo medio
2021	1	45000
2023	3	25000

- l'utente può decidere di inserire i dati di una nuova operazione, specificando tutti i dati. Dopo aver effettuato l'inserimento, il sistema aggiorna i dati della tabella sulle statistiche degli anni

Scenario alternativo

- dati scorretti: il sistema mostra un messaggio di errore all'utente

“Utente verifica archivio”

- selezionando una voce di menu, l'utente verifica quante automobili hanno movimenti di compravendita sospetti. Un'auto ha operazioni sospette se ci sono operazioni effettuate in due mesi successivi (es un'operazione a dicembre 2023 e un'altra a gennaio 2024)
- il sistema mostra un messaggio con il risultato calcolato al passo precedente. Ad esempio “Non ci sono automobili con operazioni sospette” oppure “Ci sono 4 automobili con operazioni sospette”

“Utente elimina mesi frequenti” – Per la prova di livello intermedio

- il sistema calcola qual è il mese più frequente nel quale sono state effettuate le operazioni
- Il sistema elimina dall'archivio tutte le operazioni effettuate in quel mese
- Il sistema mostra un messaggio con il numero di operazioni eliminate. Ad esempio “Il mese più frequente è novembre. Sono state eliminate 10 operazioni”
- **Nota:** è necessario sviluppare i relativi test di regressione

“Utente carica un archivio JSON” – Per la prova di livello intermedio

- Modificare il caso d'uso “Utente carica un archivio” per permettere all'utente di caricare l'archivio da un file json, facendolo selezionare dal disco
- **Nota:** per inizializzare il file si consiglia di scrivere un test per salvare il contenuto dell'archivio mock in un file json

Sviluppare l'applicazione che implementa i casi d'uso elencati, seguendo il processo di sviluppo descritto a lezione, e in particolare le seguenti operazioni:

- costruire e documentare il modello concettuale dell'applicazione

- sviluppare il frammento di applicazione che implementa i casi d'uso descritti
- sviluppare l'interfaccia grafica in **Java Swing** utilizzando l'**architettura MVC** presentata a lezione, rispettando i requisiti per l'accessibilità e l'usabilità.
- sviluppare i test di regressione utilizzando la libreria JUnit
- per effettuare operazioni di logging utilizzare le librerie SLF4J e LogBack
- utilizzare **Gradle** come sistema di costruzione del codice

Per le librerie, se necessario, utilizzare le seguenti dipendenze

- org.slf4j:slf4j-api:2.0.1
- ch.qos.logback:logback-classic:1.3.1
- ch.qos.logback:logback-core:1.3.1
- org.junit.jupiter:junit-jupiter:5.8.2

Per utilizzare Lombok, applicare il seguente plugin gradle

- id "io.freefair.lombok" version "6.5.1"

E' possibile trovare il materiale necessario nelle seguenti cartelle

- materiale didattico: **c:\documentazione\POO1 e POO2**
- file di esempio di logback.xml: **c:\documentazione\POO1 - materiale software aggiuntivo\java**