

## Quarto 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 per la gestione dei noleggi di libri cartacei effettuati in una biblioteca.

Nel sistema sono registrati alcuni utenti. Un utente è descritto da un (a) nome utente univoco (b) nome (c) cognome. Un possibile utente è {"hpotter", "Harry", "Potter"}. Un utente può prendere in prestito vari libri. Ogni prestito è registrato con il nome del libro, l'autore, la data di inizio e la data di restituzione (valorizzata quando l'utente restituisce il libro). Ad esempio l'utente "hpotter" ha noleggiato:

- Il libro "Il Signore degli Anelli" di "J.R.R. Tolkien" in data 01/10/2023, e restituito il 15/10/2023
- Il libro "Norwegian Woods" di "Murakami" in data 01/10/2023, e non ancora restituito

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

#### "Operatore carica un archivio"

- l'operatore 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*

#### "Operatore cerca utenti"

- l'operatore inserisce un cognome e/o un nome. Entrambi i campi possono essere vuoti.
- il sistema mostra in una tabella la lista di tutti gli istituti che rispettano il criterio scelto. Ad esempio, se l'utente non ha inserito né nome né cognome, il sistema restituisce tutti gli utenti. Se ha inserito il nome, verranno restituiti solo gli utenti con quel nome. In maniera simile se ha inserito solo il cognome. Se ha inserito sia nome che cognome, verranno restituiti tutti gli utenti con quel nome e quel cognome.
- Per ogni utente bisogna mostrare username, nome, cognome, numero totale di libri presi in prestito, e numero di libri in prestito ancora da restituire.
- I risultati devono essere ordinati in maniera crescente per numero totale di libri presi in prestito. In caso di parità, verrà ordinato per numero di libri ancora da restituire

#### "Operatore gestisce prestiti"

- l'operatore seleziona un utente dai risultati
- il sistema mostra, in un nuovo pannello, i dati dell'utente
- il sistema calcola e mostra inoltre il numero di libri presi in prestito (totali e ancora da restituire) raggruppati per autore. I dati saranno mostrati in una tabella ordinata per numero di libri decrescente. Un esempio di tabella è la seguente

Autore	Numero totale di libri noleggiati	Numero di libri noleggiati ancora da restituire
J.R.R. Tolkien	4 libri	0 libri
Murakami	1 libro	1 libro

- l'operatore può decidere di inserire i dati di un nuovo noleggio, specificando titolo del libro, autore, data di inizio noleggio.
- Se l'utente ha attualmente più di 3 libri in prestito non ancora consegnati, il sistema marcherà come restituito, in data odierna, il libro noleggiato da più tempo tra questi
- Dopo aver effettuato l'inserimento, il sistema aggiorna i dati della tabella precedente

#### Scenario alternativo

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

#### "Operatore verifica archivio"

- selezionando una voce di menu, l'operatore richiede di fare una verifica sui libri presi in prestito da più di 6 mesi e non ancora restituiti.
- in particolare il sistema estrae questi libri, e verifica se sono tutti di autori diversi tra di loro
- il sistema mostra un messaggio con il risultato calcolato al passo precedente. Ad esempio, un possibile messaggio è "Tutti i libri presi in prestito da più di 6 mesi appartengono ad autori tutti diversi tra di loro" oppure "Tra i libri presi in prestito da più di 6 mesi ci sono più volte libri di uno stesso autore"

#### "Operatore elimina rappresentanti duplicati" – Per la prova di livello intermedio

- selezionando una voce di menu, l'operatore richiede di calcolare l'autore che ha il noleggio medio in giorni dei suoi libri più alto.
- Il sistema mostra un messaggio del tipo "L'autore con il noleggio medio più alto è J.R.R. Tolkien con una durata media di 95 giorni"
- **Nota:** è necessario sviluppare i relativi test di regressione

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

- Modificare il caso d'uso “Operatore carica un archivio” per permettere all'operatore 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**