Secondo 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 ricerca di corsi di fitness di una palestra. Ogni corso ha un nome, l'istruttore, il costo mensile. Ad esempio, un possibile corso è {"Crossfit 101", "Giuseppe Forte", 45€ mensili}. Per ogni corso bisogna tener traccia delle lezioni in programma. Ogni lezione ha la data e ora, la difficoltà (un valore da 1 a 5), la durata in minuti, se si svolge all'aperto o al chiuso. Ad esempio, una possibile lezione è {01/09/2024 16:30, difficoltà 1, 60 minuti, al chiuso}.

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 corsi"**

- l'utente inserisce
  - una data (giorno, mese e anno) *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.*
  - un livello di difficoltà massimo
  - un criterio di ordinamento, che può essere "costo crescente" o "costo decrescente".
- il sistema mostra in una tabella la lista dei corsi che hanno almeno una lezione con le seguenti caratteristiche:
  - la lezione è programmata in una data successiva alla data inserita;
  - la lezione è di livello minore o uguale del livello di difficoltà massimo inserito.
- la tabella che mostra la lista dei corsi deve essere ordinata secondo il criterio scelto dall'utente ("costo crescente" o "costo decrescente").

"**Utente inserisce nuova lezione"**

- l'utente seleziona un corso dai risultati
- il sistema mostra, in un nuovo pannello, i dati del corso
- il sistema calcola e mostra inoltre, per il corso selezionato e per ogni livello di difficoltà da 1 a 5:
  - il numero di lezioni del corso di quel livello di difficoltà
  - la durata media delle lezioni di quel livello di difficoltà.

I dati saranno mostrati in una tabella ordinata per difficoltà crescente. Un esempio di tabella è la seguente

Difficoltà	Numero di lezioni	Durata media
1	1	90 minuti
2	3	45,5 minuti

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

Scenario alternativo

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

"**Utente verifica archivio"**

- selezionando una voce di menu, l'utente conta quanti corsi hanno lezioni sovrapposte. Una lezione è sovrapposta se termina dopo l'inizio della lezione successiva*. Nota: per effettuare la verifica si consiglia di ordinare le lezioni per orario di inizio*
- il sistema mostra un messaggio con il risultato calcolato al passo precedente. Ad esempio "Non ci sono corsi con lezioni che si sovrappongono" oppure "Ci sono 3 corsi con lezioni che si sovrappongono"

"**Utente elimina mesi frequenti" – Per la prova di livello intermedio**

- il sistema calcola il periodo più lungo nel quale non c'è alcuna lezione in archivio
- il sistema mostra il messaggio con il risultato calcolato. Ad esempio: "Il periodo più lungo senza lezioni va da 1 agosto 2024 al 15 settembre 2024"
- ***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**