

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 la gestione dei voli effettuati da un aeromobile. Ogni aeromobile è rappresentato da codice, numero passeggeri, tipologia, data di ultima manutenzione. Ad esempio, un possibile aeromobile è {"B12n", "Boeing 747", 660 passeggeri, 26 giugno 2022}. Ogni aeromobile può effettuare vari voli. Ogni volo è descritto dalla data e ora di partenza, l'aeroporto di partenza, l'aeroporto di destinazione, e la durata in minuti. Un possibile volo è 27 luglio 2022 10:30, partenza "Fiumicino", arrivo "John F. Kennedy International Airport", 540 minuti.

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 aeromobili"

- l'utente sceglie un numero minimo di passeggeri, e un criterio di ordinamento, che può essere "passeggeri crescente", "passeggeri decrescenti" o "tipologia crescente". *NOTA: per la scelta del criterio, utilizzare JRadioButton*
- il sistema mostra in una tabella la lista degli aeromobili che hanno almeno il numero di passeggeri richiesti, ordinati secondo il criterio scelto. Nella tabella bisogna mostrare codice, tipologia, passeggeri e data di ultima manutenzione.

"Utente inserisce nuovo volo"

- l'utente seleziona un aeromobile dai risultati
- il sistema mostra, in un nuovo pannello, i dati dell'aeromobile e i viaggi attualmente registrate. Per ogni viaggio bisogna mostrare partenza, destinazione, data e durata
- l'utente può decidere di inserire i dati di un nuovo viaggio, specificando tutti i dati.
- Il sistema aggiunge il volo all'aeromobile ed aggiorna i dati

Scenario alternativo

- dati scorretti: il sistema mostra un messaggio di errore all'utente
- data ultima manutenzione: se l'ultima manutenzione è stata effettuata più di un anno prima della data di partenza del nuovo volo, il sistema mostra un messaggio di errore e il caso d'uso termina. Se invece è stata effettuata più di 6 mesi prima, il sistema mostra solo un messaggio di avvertimento, e l'inserimento viene effettuato ugualmente.

"Utente verifica archivio"

- selezionando una voce di menu, il sistema cerca tutti gli aeromobili che hanno almeno 2 viaggi, e che sono partiti sempre dallo stesso aeroporto. Tra questi calcola l'aeromobile che la media della durata dei viaggi più alta
- il sistema mostra un messaggio con il risultato calcolato al passo precedente

"Utente ricerca aeromobili" – Per la prova di livello intermedio

- il sistema calcola, per ogni tipologia di aeromobile, in quanti diversi aeroporti è stato (considerando sia gli aeroporti di partenza che di arrivo), ed inoltre il giorno della settimana in cui è atterrato più volte
- i dati saranno mostrati in una tabella ordinata per tipologia crescente. Il giorno di atterraggio dev'essere riportato in italiano. Un esempio di tabella è la seguente

Nota: è necessario sviluppare i relativi test di regressione

Tipologia	# Aeroporti diversi	Giorno di atterraggio più frequente
Boeing 747	3	Lunedì

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
- come sistema di costruzione del codice usare **NetBeans standard**

E' possibile trovare il materiale necessario nelle seguenti cartelle

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