

Progettazione di Applicazioni Web T
Prova d'Esame – 10 settembre 2021

Tempo a disposizione: 120 minuti

La soluzione comprende la consegna elettronica dei seguenti file:

ArchivioVolo.zip file zip contenente il sorgente java/class, file XML e txt per il punto 1
CompagniaAerea.zip file zip contenente il sorgente java/class e txt per il punto 2
WebRatio.zip file zip contenente il testo di risposta (txt/doc/pdf) al punto 3

Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, codice java e relativi .class, descrittori XML, file txt/doc/pdf, ecc.) e NON dell'intero progetto.

N.B. La prova si intende superata se il punteggio finale non è inferiore a 18 punti (su un totale di 33). In particolare, è necessario (ma non sufficiente) ottenere la sufficienza in tutti gli esercizi.

ESERCIZIO 1 - 14 punti (sufficienza: 7 punti)

Si progetti una grammatica **XML Schema**, e un suo **documento XML** di esempio, per la modellazione delle informazioni relative al servizio archivio prenotazioni voli aerei “**ArchivioVoli**”, nel rispetto delle seguenti specifiche:

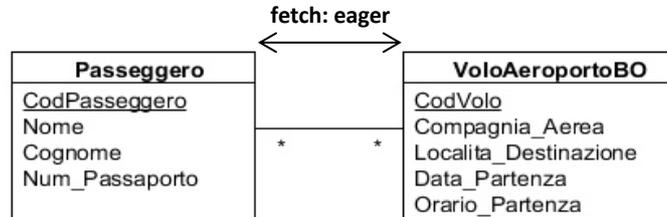
- L’archivio comprende tutti gli archivi personali relativi ai clienti che hanno sottoscritto il servizio.
- Ogni archivio personale è costituito da nome, cognome, numero di passaporto della persona associata all’archivio e da zero o più prenotazioni.
- Ogni prenotazione è rappresentata da codice di prenotazione, nome della compagnia aerea, tipo di volo, aeroporto di partenza, data e ora di partenza, aeroporto di arrivo e data e ora di arrivo (tutti i campi devono essere modellati come obbligatori).
- Il tipo di volo può assumere uno dei seguenti valori: {Andata-Ritorno”, “Solo-Andata”}.
- La data è costituita dai campi “giorno”, “mese” e “anno”, mentre l’ora di partenza da “ora” e “minuti”; tutti i campi sono di tipo numero intero.

Si realizzi quindi l’applicazione **Java “Voli”** che, facendo uso del **parser DOM** e del **documento XML di esempio** realizzato al punto precedente, esponga il metodo `Set<String> getVoli()`, unitamente a suo un `main` di prova, in grado di restituire il l’insieme dei codici di prenotazione relativi ai voli di tipo “Andata-Ritorno” operati da “Ryanair” durante i mesi estivi (ovvero il mese di volo è compreso tra “Giugno” e “Agosto”), e con partenza da “Bologna”. Stampare il risultato prodotto dal metodo `getVoli()` sul file **Volo.txt**.

Progettazione di Applicazioni Web T
Prova d'Esame – 10 settembre 2021

ESERCIZIO 2 - 14 punti (sufficienza: 7 punti)

Partendo dalla realtà illustrata nel **diagramma UML** di seguito riportato, si fornisca una soluzione alla gestione della persistenza basata su **Pattern DAO** in grado di “mappare” efficientemente e con uso di ID surrogati il modello di dominio rappresentato dai **JavaBean Passeggero e VoloAeroportoBO** del **diagramma UML** con le corrispondenti **tabelle relazionali derivate dalla progettazione logica del diagramma** stesso.



Nel dettaglio, dopo aver creato da applicazione Java gli **scemi delle tabelle** all'interno del proprio schema nel database **TW_STUD** di **DB2** (esplicitando tutti i **vincoli** opportuni), **implementato i JavaBean e realizzato le classi** relative al **Pattern DAO** per l'**accesso CRUD** alle tabelle, si richiede **l'implementazione di un metodo in grado di restituire** *“nome della compagnia aerea che ha trasportato il maggior numero di passeggeri con destinazione Monaco di Baviera”*.

Si crei poi un **main di prova** in grado di:

- inserire due o più tuple nelle tabelle di interesse al fine di produrre un risultato consistente rispetto a quanto richiesto dal metodo;
- fare uso corretto del metodo realizzato al punto precedente al fine di produrre la stampa del risultato, opportunamente formattata, sul file **CompagniaAerea.txt**.

N.B. L'implementazione del **Pattern DAO** deve limitarsi al solo **DBMS DB2**. La soluzione deve sfruttare il **mapping N-M** specificato nello schema **UML** e **propendere per il caricamento dei dati indicato**. Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.

ESERCIZIO 3 - 5 punti (sufficienza: 3 punti)

Considerando l'ipertesto progettato nello **schema IFML di WebRatio** di seguito riportato, il candidato descriva le relazioni esistenti tra i componenti pagina **pA**, **pB**, **pC**, **pD**, l'area **aA** e il view component simple list **IA** presenti all'interno della site view **SV**.

