

Tecnologie Web T
20 Settembre 2018 – Compito

Tempo a disposizione: 3 ore

La soluzione comprende la **consegna elettronica** dei seguenti file mediante l'apposito applicativo Web **esamix** (<http://esamix.labx>):

Matrice.zip file zip contenente il sorgente java/class e pagine Web per punto 1
Ludoteca.zip file zip contenente il sorgente java/class, file xml/xsd e txt per punto 2
ABC.zip file zip contenente il sorgente java/class e txt per 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, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto

N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), equamente distribuiti sui tre esercizi, ovvero almeno 6 punti sul primo esercizio, 6 punti sul secondo esercizio e 6 punti sul terzo esercizio

Studenti in debito di Tecnologie Web L-A

Viene richiesto lo svolgimento dei soli esercizi 1 (17 punti) e 2 (16 punti). Tempo a disposizione: 2 ore.

I 18 punti necessari per l'ammissione all'orale sono così distribuiti: almeno 10 punti sul primo esercizio e almeno 8 punti sul secondo

ESERCIZIO 1 (11 punti)

Si realizzi una applicazione Web per il calcolo “altamente concorrente” di somma e differenza di due matrici fissate A e B, basata principalmente su tecnologia Java servlet, Javascript, AJAX e JSON.

In particolare, l'applicazione Web deve permettere all'utente (NON necessaria autenticazione) di configurare un intervallo di *freshness* valido per la sessione (numero naturale, espresso in secondi) e di comandare le operazioni di somma e di differenza fra due matrici. Mentre si sta ancora calcolando il risultato di una operazione, ad esempio la somma, deve essere possibile per il medesimo utente comandare già un'altra operazione di calcolo, ad esempio differenza. Le due matrici A e B (quadrato, 10*10) si suppongono note server-side: in particolare, all'avvio, l'applicazione dovrà caricarne i valori da un apposito file server-side (nome definito a livello di configurazione dell'applicazione); per semplicità tali valori saranno considerati validi e immutati per tutta la durata dell'applicazione.

Inoltre, al fine di rendere più efficiente l'esecuzione dell'applicazione:

- se un utente chiede una operazione il cui risultato è stato già calcolato, per lo stesso utente o per altri, meno di *freshness* sec precedenti, l'applicazione NON deve effettuare veramente il calcolo, ma semplicemente restituire il risultato precedentemente calcolato (*cached*). In ogni caso il risultato viene restituito in formato JSON;
- se un utente chiede una operazione il cui risultato non è già calcolato o non è sufficientemente *fresh*, allora il calcolo deve essere effettuato concorrentemente da due *servlet*, ciascuna delle quali opera solo su metà delle righe delle due matrici (prima *servlet* sulle righe 0-4 di A e B, seconda *servlet* sulle righe 5-9).

Si commenti inoltre nel codice sorgente quale modello di esecuzione concorrente per le *servlet* si è deciso di utilizzare e con quali vantaggi rispetto all'altro modello possibile.

Tecnologie Web T

20 Settembre 2018 – Compito

ESERCIZIO 2 (11 punti)

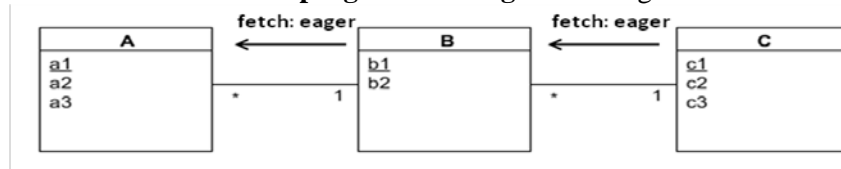
Si progetti una grammatica **XML Schema** e alcuni **documenti XML** di esempio per la modellazione delle informazioni della **Ludoteca “Chiocciola”**, nel rispetto delle seguenti specifiche:

- I giocattoli si dividono in due tipi: giochi in formato digitale, quali audiovisivi (“AV”) e cd musicali (“CDM”), e giochi tradizionali (“GT”), basati sull’uso e il tatto, quali, ad esempio, peluche, cavallo a dondolo, costruzioni lego, puzzle, giochi di società, ecc.
- Al fine di ricerca nel database, ogni giocattolo è caratterizzato da un codice univoco.
- Più nel dettaglio:
 - Ogni audiovisivo e cd musicale ha inoltre associato informazioni quali il titolo, l’autore, la casa produttrice, l’anno di produzione, la fascia di età (es. 0- 3, 4-6, ecc.), la lingua (es., “italiano”, “inglese”, ecc.) e la durata in minuti.
 - Ogni gioco tradizionale ha invece associato informazioni quali il nome, l’autore, la marca, la fascia di età (es. 0+, 1-2, 3-5, 6-8, ecc.) e la modalità di utilizzo, ovvero gioco individuale (“GI”) oppure gioco di gruppo (“GG”).

Si realizzi quindi l’applicazione **Java “Ludoteca”** che, facendo uso del **parser SAX**, esponga i tre metodi `getCDMusicali_Fascia0-3()`, `getAudiovisivi_Italiano()` e `getGiocoTradizionale_Fascia6-8_GI()`, unitamente a un `main` di prova, in grado di restituire risultati, e relative cardinalità, dei metodi stessi quando applicati ai file XML di esempio. Si produca infine la stampa opportunamente formattata e completa di quanto richiesto sul file **Ludoteca.txt**.

ESERCIZIO 3 (11 punti)

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



Nel dettaglio, dopo aver creato da applicazione Java lo schema delle tabelle nel proprio schema nel database **TW_STUD** di **DB2** (esplicitando tutti i vincoli derivati dal diagramma UML) e implementato **JavaBean e metodi necessari per la realizzazione delle operazioni CRUD**, si richiede la realizzazione di una **classe di prova** in grado di:

- Inserire due o più tuple nelle tabelle al punto sopra.
- Restituire:
 - per ogni tupla di **C**, elenco e numero di tuple di **B** ad essa associata; per ognuna di queste ultime, elenco e numero di tuple di **A** corrispondenti;
 - tuple **b2** di **B** con associato il minor numero di tuple **a1** di **A**.
- Produrre una stampa opportunamente formattata e completa dei risultati delle richieste al punto precedente sul file **ABC.txt**.

N.B. La soluzione deve sfruttare i mapping specificati nel diagramma UML. Ogni ulteriore scelta fatta dallo studente deve essere opportunamente giustificata mediante commenti nel codice.