Tecnologie Web T
15 Settembre 2017 – Compito

Tempo a disposizione: 3 ore

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

ClientiLettori.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
Biblioteca.zip	file zip contenente il sorgente XSD, XML, java/class e txt per punto 2
Acquisto.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 un'applicazione Web, principalmente basata su tecnologie **Java servlet, Javascript e JSON**, per la **lettura e l'analisi collaborativa** di un documento da parte di **lettori multipli**.
L'applicazione deve consentire a ogni utente di autenticarsi tramite username e password; come ipotesi semplificativa, non è possibile per uno stesso utente avere più sessioni correntemente attive, ad esempio da terminali differenti. Deve essere necessariamente utilizzato un **cookie** per mantenere l'informazione booleana di utente autenticato.
L'utente autenticato dovrà ricevere, tramite **JSON**, la sua porzione personale (un quarto della lunghezza totale del file, ovviamente di porzione non precedentemente assegnata) del contenuto di un documento **lettura.txt** già presente sul server: il messaggio JSON ricevuto sarà costituito da una informazione di lunghezza seguita da una stringa con numero di caratteri uguale a tale lunghezza. Se il numero di clienti concorrenti è superiore a 4, le porzioni del file possono essere ri-assegnate, per garantire maggiore affidabilità all'applicazione Web complessiva grazie a lettori multipli.
Ricevuto il messaggio, ogni cliente dovrà analizzarlo **lato cliente** per contare quante volte la parola "esame" è presente. Alla fine del conteggio, il risultato parziale di ogni cliente deve essere inviato **lato servitore**, dove sarà quindi determinato il conteggio totale.
Inoltre, deve essere data la possibilità all'amministratore del sistema (username=admin; password=admin) di intervenire per bloccare un conteggio in corso (perché ad esempio il risultato del conteggio parziale di qualche cliente non sta arrivando entro un intervallo di tempo ragionevole); in tal caso, il conteggio deve essere completato forzatamente lato servitore.

Tecnologie Web T
15 Settembre 2017 – Compito

ESERCIZIO 2 (11 punti)

Si progetti una grammatica **XML Schema** e un **documento XML** di esempio in grado di modellare la realtà di una **Biblioteca Musicale** nel rispetto delle seguenti specifiche:

- ciascun documento XML si riferisce a una biblioteca;
- una biblioteca si compone di zero o più musicisti;
- un musicista è descritto mediante i campi *nome*, *cognome*, *nome d'arte*, *genere musicale*, uno o più *album prodotti* (tutti i campi sono obbligato tranne *nome d'arte* che è opzionale);
- ogni album è modellato dai campi *titolo album* e *anno pubblicazione* (obbligatori) e da una *lista brani* (opzionale);
- ciascun brano della lista è rappresentato dai campi *titolo brano* e *lunghezza brano* (espressa nella forma *minuti:secondi*).

Si realizzi poi un'**applicazione Java** facente uso del **parser DOM** in grado di esibire i seguenti metodi:

`Set<String> getCognomi(Document doc, String genere, int totAlbum);`

dati in input un genere musicale e un numero intero rappresentante il numero totale di album prodotti da un artista (`totAlbum`), restituisce il cognome degli artisti associati al genere musicale in input che hanno pubblicato meno di `totAlbum` album (per semplicità, si assume che non possono esistere due artisti con lo stesso cognome);

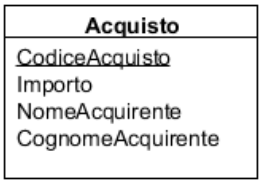
`boolean addAlbum(Document doc, String nome, String cognome, String genere, String titoloAlbum, int annoPubblicazione);`

aggiunge alla lista di album prodotti dall'artista in input un nuovo album con titolo e anno specificati come parametri. In particolare, il metodo `addAlbum` restituisce `false` se non esiste un artista con nome, cognome e genere specificati in input, oppure esiste ma possiede già un album con quel titolo, `true` altrimenti.

L'applicazione deve produrre la stampa (sul file **Biblioteca.txt**) dei risultati ottenuti testando i metodi `getCognomi` e `addAlbum` sul documento XML di esempio**.**

ESERCIZIO 3 (11 punti)

Partendo dalla realtà illustrata nel **diagramma UML** di seguito riportato, si fornisca una soluzione alla gestione della persistenza basata su metodologia **Forza Bruta** in grado di "mappare" efficientemente con uso di ID surrogati il modello di dominio rappresentato dal **JavaBean** del diagramma UML con la corrispondente **tabella relazionale derivata dalla progettazione logica** del diagramma stesso.



Nel dettaglio, dopo aver creato da applicazione Java lo schema della tabella 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 definizione del metodo `Set<String> AcqustoSoglia(double Soglia)` in grado di ottenere l'insieme degli acquisti di importo superiore alla soglia specificata in input. Si richiede quindi di realizzare una classe di prova in grado di:

- inserire tre acquisti;
- modificare il costo di uno dei tre acquisti inseriti;
- utilizzare correttamente il metodo `AcqustoSoglia` per ottenere l'insieme di degli acquisti (`CodiceAcquisto`) con importo inferiore a 200 euro;
- eliminare gli acquisti ottenuti al punto precedente;
- produrre una stampa, opportunamente formattata, dei risultati delle operazioni ai punti precedenti sul file **Acquisto.txt**.