

Tecnologie Web T
16 Settembre 2016 – Compito

Tempo a disposizione: 3 ore

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

Doc_multipli.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
Cineteca.zip	file zip contenente il sorgente java/class e file xml/xsd per punto 2
Conta.zip	file zip contenente il sorgente java/class e file XML 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 **JSP**, **Java servlet** e **Javascript**, per la scrittura collaborativa di un documento da parte di autori multipli.

L'applicazione deve consentire ad ogni utente di autenticarsi tramite username e password; come ipotesi semplificativa, non è possibile per uno stesso utente avere più sessioni contemporaneamente attive, ad esempio da terminali differenti. Superata l'autenticazione, l'utente deve ricevere, via **JSON**, il contenuto corrente del documento: tale contenuto sarà costituito da blocchi strutturati in sequenze "nome_autore" + "testo_di quell'autore". L'intero documento dovrà essere visualizzato nel browser Web, dando anche la possibilità all'utente di aggiungere un nuovo elemento della sequenza in fondo al documento corrente. Le modifiche saranno salvate e rese visibili agli altri utenti solo dopo la pressione esplicita di un tasto di "Append".

In caso di append concorrenti da parte di utenti differenti, ad esempio se utente1 ha ricevuto una versione precedente del documento rispetto a quella modificata lato servitore da utente2, la richiesta di "Append" degli utenti con versione "non aggiornata" deve fallire; a tali utenti deve essere inviata la nuova versione del documento JSON in background tramite **AJAX**.

Inoltre, deve essere data la possibilità all'amministratore del sistema (username=admin; password=admin) di intervenire in scrittura su qualsiasi parte del documento, ovvero di modificare sia autori che contenuti in qualunque parte del documento, ovviamente in modo tale da evitare criticità.

Tecnologie Web T
16 Settembre 2016 – Compito

ESERCIZIO 2 (11 punti)

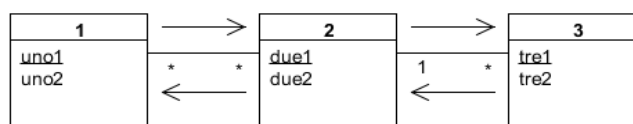
Si progetti una grammatica **XML Schema**, e un suo **documento XML** di esempio, per la modellazione delle informazioni **dell'archivio film della Cineteca "BO"**, nel rispetto delle seguenti specifiche:

- L'archivio film si divide in due tipologie di contenuti: l'archivio audiovisivo costituito da 31.000 filmati e l'archivio pellicola (contenente 60.000 pellicole).
- A fini di ricerca nel database, ogni audiovisivo è caratterizzato dalla coppia [*Cognome, Nome*] del regista e da un titolo [*Titolo italiano*].
- Ogni audiovisivo ha inoltre associato informazioni quali la *nazione* (es., "IT"), la *lingua* (es., "ITA"), il *tipo di colore*, ovvero {*Colore, BN*}, la *durata* in minuti, la *tipologia* (es., "Documentario"), il *formato* specifico (nel dettaglio, {*Blu-ray, DVD, Divx, VHS*}), l'*anno* di produzione, la *disponibilità* per la consultazione (*si/no*) e la *collocazione* fisica all'interno della cineteca. Nel dettaglio, la collocazione è rappresentata da un codice complesso formato da 5 componenti separate dal carattere "."; nel dettaglio, le prime tre parti del codice sono formate da due cifre intere, la quarta è formata da 3 cifre intere, mentre l'ultima è formata da una stringa di 3 caratteri (a titolo di esempio, "05.02.20.053.BIB").
- Per quanto riguarda le pellicole, le stesse possono essere richieste solo in forma "manuale", ovvero mediante lettera scritta, indicando "regia" e "titolo" del film di interesse; le pellicole possono essere ammesse al prestito se, previa revisione, e formulazione di un esito (*si/no*), risultino essere in buono stato di conservazione. Le pellicole possono essere di due tipologie (pellicole in 35 e 16 mm).

Si realizzi quindi l'**applicazione Java "Cineteca"** che, facendo uso del **parser DOM**, esponga i metodi `getAudiovisiviBN()` e `getPellicole16()`, unitamente a suo un `main` di prova, in grado di restituire quanti audiovisivi in bianco e nero sono presenti nella cineteca e quante pellicole in 16 mm sono in buono stato di conservazione presso la cineteca.

ESERCIZIO 3 (11 punti)

Partendo dalla realtà illustrata nel **diagramma UML** di seguito riportato, si fornisca una soluzione alla gestione della persistenza basata su **Hibernate** in grado di "mappare" efficientemente e con uso di ID surrogati 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 gli schemi delle tabelle all'interno del proprio schema nel database **TW_STUD** di DB2 (esplicitando tutti i vincoli derivati dal diagramma UML), implementato i **JavaBean**, definiti i **file XML di mapping** e il **file XML di properties**, si richiede la realizzazione di una classe di prova facente uso delle **API Hibernate** in grado di:

- inserire due o più tuple nelle tabelle di interesse;
- restituire i) per ogni tupla di **1**, il numero di tuple di **2** ad essa associata e, per ognuna di queste ultime, il numero di tuple di **3** corrispondenti ii) per ogni tupla di **3**, la corrispondente coppia (*due1, due2*) di **2**, congiuntamente al numero di tuple di **1** associate alla coppia;
- produrre una stampa opportunamente formattata e completa dei risultati delle query del punto precedente sul file **Conta.txt**;

il tutto, mediante opportuna gestione delle **transazioni**.

N.B. La soluzione deve sfruttare i mapping M-N e 1-N specificati nel diagramma UML. Ogni ulteriore scelta fatta dallo studente deve essere opportunamente giustificata mediante commenti nel codice.