Tecnologie Web T
26 Giugno 2015 – Compito

Tempo a disposizione: 3 ore

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

Convers_Asta.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
Multimedia.zip	file zip contenente il sorgente java/class e file xml/dtd per punto 2
Ospedale.zip	file zip contenente il sorgente java/class 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** che permetta a utenti autenticati di accedere a una conversazione Web per l'esecuzione e l'eventuale aggiudicazione di un'asta; l'applicazione deve essere basata principalmente su **tecnologie Java servlet, JSP e JSON**.
In particolare, l'applicazione dovrà essere costituita da una servlet **accesso** che autentichi un utente sulla base di nome e cognome (considerati identificatori univoci) e che gli permetta di aprire una sessione di partecipazione all'asta se gli utenti già partecipanti sono in numero inferiore a 100. Inoltre, la servlet **accesso** visualizzerà le informazioni relative all'asta in corso (inclusa l'offerta corrente più alta) e condurrà la conversazione verso una pagina JSP **offerta** che permetterà di fare una nuova offerta (ovviamente più alta della corrente). Entrambi i flussi di informazione (dati sull'asta in corso e nuova offerta) devono essere trasferiti tramite serializzazione **JSON**.
Inoltre, si preveda che dopo 30 minuti dalla partenza dell'asta (ovvero della prima offerta su di essa), l'asta venga automaticamente chiusa e si consideri vincitore l'utente con offerta corrente più alta; a questo punto l'utente vincitore dell'asta, se richiede un refresh della sua pagina **offerta**, deve essere condotto nella conversazione alla pagina JSP per il pagamento tramite carta di credito.
Utenti non autenticati (e utenti autenticati quando non vi sono sessioni d'asta attive in corso) devono potere comunque accedere all'informazione sull'ultima asta terminata e in particolare sull'importo finale di aggiudicazione della stessa.

Tecnologie Web T
26 Giugno 2015 – Compito

ESERCIZIO 2 (11 punti)

Si progettino **una grammatica XML Schema e una grammatica DTD**, assieme a un loro **documento XML di esempio**, per la **descrizione delle informazioni di un'applicazione Web per la formulazione d'interrogazioni Multimediali**. Nello specifico, data un'immagine in input (detta *query*) e un insieme di parametri, si vogliono restituire all'utente le *K* immagini di una collezione di riferimento che sono più "simili" alla query dal punto di vista visuale.

In particolare, le informazioni da modellare per la formulazione di un'interrogazione rispettano le seguenti specifiche:

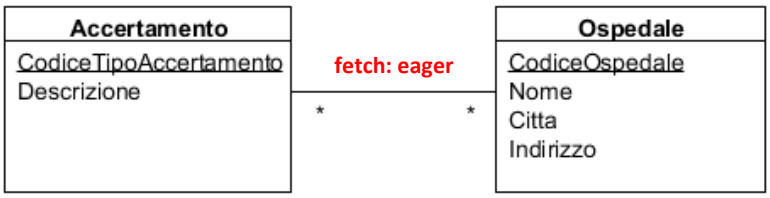
- il documento XML modella l'applicazione in termini di query e risultato corrispondente ed esplicita: ID dell'immagine query in input (di tipo alfanumerico), modalità di confronto tra immagini da utilizzare durante l'elaborazione dell'interrogazione (rientrante nell'insieme {"A", "B", "C"} e mutuamente esclusiva), cardinalità del risultato *K* e lista delle *K* immagini (ID) che costituiscono il risultato della query, una volta che la stessa verrà eseguita.

Si realizzi poi l'**applicazione Java "Multimedia"** che, facendo uso del **parser DOM e SAX**, esponga il metodo `getQuery()`, unitamente a suo un `main` di prova, al fine di restituire *l'insieme degli ID delle query richieste dagli utenti raggruppate per modalità di confronto* (si stampi il risultato sul file **multimedia.txt**).

N.B. Tutti i campi sono obbligatori tranne la lista del risultato che si popolerà solo dopo l'esecuzione della query.

ESERCIZIO 3 (11 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 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** opportuni), **implementato** i **JavaBean** e **realizzato le classi** relative al **Pattern DAO** per **l'accesso CRUD** alle tabelle**,** si richiede l'implementazione di **opportuni metodi per il supporto delle seguenti operazioni**:

- per ogni ospedale di "Bologna", si richiede il numero di accertamenti offerti, assieme all'elenco delle relative descrizioni; per l'accertamento "TAC", si richiede il numero di ospedali di "Bologna" in cui è possibile eseguirlo, unitamente all'elenco dei nomi degli ospedali rilevanti.

Si crei poi un **main di prova** che: (i) inserisca due o piu' tuple nelle tabelle di interesse; (ii) faccia uso corretto dei metodi realizzati al punto precedente al fine di produrre una stampa del risultato sul file **ospedale.txt**.

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