

Tecnologie Web T
22 Giugno 2012 – Compito

Tempo a disposizione: 3 ore

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

Scommesse.zip	file zip contenente il sorgente java dell'applicazione al punto 1
Bambino.zip	file zip contenente il sorgente java dell'applicazione al punto 2
Studente.zip	file zip contenente il sorgente java dell'applicazione 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, 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 9 punti sul primo esercizio e almeno 9 punti sul secondo

ESERCIZIO 1 (11 punti)

Si realizzi un'**applicazione Web** per la gestione delle scommesse online relative alle partite dei campionati europei di calcio, basata principalmente su tecnologie **servlet** e **JSP**. L'obiettivo primario è quello di ricevere/gestire richieste di scommesse e di consentire piccole statistiche su di queste, con memorizzazione non persistente delle scommesse.

A tal fine si realizzi una pagina **scommesse.jsp** che consenta all'utente di inserire le due squadre impegnate nella partita come stringhe "vincente" e "perdente", e l'importo che si intende scommettere. La pagina (o una servlet di supporto) dovrà consentire all'utente di chiedere quale è la vincita correlata possibile (fattore moltiplicativo * importo; per semplicità, fattore moltiplicativo=1.5 nel caso di vincente=Italia, 1.2 in tutti gli altri casi) e di decidere di confermare la scommessa. Fino a che le scommesse non sono confermate, queste dovranno essere mantenute solo per la durata della richiesta HTTP. Inoltre, la pagina dovrà mano a mano mostrare le scommesse confermate dall'utente nella sessione corrente.

Inoltre, la pagina **scommesseGlobali.jsp** deve permettere di effettuare analisi sulle scommesse confermate da parte di tutti gli utenti. Lo scopo è quello di analizzare se si stanno verificando flussi anomali di scommesse su una particolare partita. A tal fine, l'amministratore del sito di scommesse, previa autenticazione (username="admin", password="admin"), può specificare le squadre della partita per richiedere il calcolo della somma di denaro puntata su quella partita, oppure nulla per indicare la volontà di visualizzare le somme per tutte le partite per cui esiste almeno una scommessa. La richiesta deve essere diretta verso un'opportuna servlet che ha il compito di effettuare il calcolo e mostrare il risultato all'utente tramite la pagina **scommesseGlobali.jsp**. Inoltre, il risultato dell'ultima ricerca effettuata dall'amministratore deve essere visualizzata anche al successivo accesso dell'amministratore alla pagina JSP, anche dopo la terminazione della sua sessione utente e senza scatenare il richiamo della servlet.

Tecnologie Web T

22 Giugno 2012 – Compito

ESERCIZIO 2 (11 punti)

Realizzare una **pagina HTML dinamica** basata su tecnologia **Javascript** in grado di fornire funzionalità di analisi statistica sull'età dei bambini che frequentano la scuola materna. Tali funzionalità devono essere contenute nel file **Bambino.js**, opportunamente caricato dall'omonima pagina **Bambino.html**.

Nel dettaglio, la pagina **Bambino.html** si presenta all'utente con sfondo blu e testo rosso (Arial, Bold, 14pt); il caricamento della stessa mostra sullo schermo il titolo “**Dettaglio dell'età dei bambini della scuola SuperBimbo**” (centrato). Quando la pagina è stata caricata completamente, si richiede l'esecuzione di una funzione *main()* che deve:

- avviare una finestra di prompt per l'inserimento delle età dei bambini e_1, e_2, \dots, e_n (in numero non noto a priori):
 - ogni età e_i deve essere separata da un numero arbitrario di spazi;
 - per essere considerata valida, ogni età e_i deve essere un numero di una cifra appartenente al range [2-5]
- terminare, in caso di violazione di uno dei vincoli al punto sopra; produrre l'analisi statistica dei dati immessi sotto descritta, altrimenti.

L'analisi statistica richiede il calcolo e la visualizzazione a video del valore massimo, minimo e medio dell'età dei bambini; si definiscano inoltre le fasce di età “*Junior*” [dai 2 ai 3 anni compresi] e “*Senior*” [dai 4 ai 5 anni compresi], e si calcoli, per ognuna di esse, il numero di bambini che vi appartengono, formattando la visualizzazione mediante opportuna tabella.

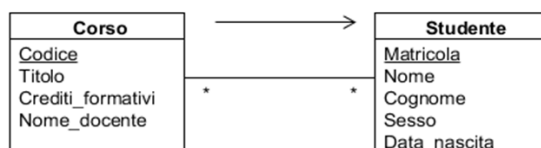
N.B.1 La soluzione deve definire l'oggetto *Bambino* dalle proprietà e dai metodi in grado di modellare la realtà sopra descritta.

N.B.2 Modellare il layout grafico della pagina **Bambino.html** nel file dedicato **Bambino.css**.

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 il modello di dominio relativo.

In particolare, la soluzione richiesta deve propendere per una modellazione del problema basata su **due JavaBean**, ovvero **Corso** e **Studente**, lato Java, e sulle corrispondenti **tre tabelle** **Corso**, **Corso_Studente**, e **Studente** contenute nel database **TW_STUD** di DB2.



Nel dettaglio, sfruttando l'efficienza degli **ID surrogati**, dopo aver creato da applicazione le tabelle all'interno del proprio schema nel database **TW_STUD** (esplicitando gli opportuni **vincoli di PK e FK**), implementato i JavaBean e definito i file XML di mapping Hibernate necessari, si richiede la realizzazione di una classe di prova facente uso delle API Hibernate in grado di:

- inserire due corsi e tre o più studenti per ognuno di essi;
 - restituire, partendo dal Codice di un corso, il numero di studenti donna (Sesso='F') iscritti a quel corso;
 - stampare sul file **Studentesse.txt** le statistiche al punto precedente riportando, per ogni studentessa, anche il suo **Nome**, **Cognome** e **Data di nascita**
- il tutto, mediante opportuna **gestione delle transazioni**.

N.B.1 Partendo dalle specifiche sopra, la soluzione deve **minimizzare il numero di riferimenti** mantenuti esplicitamente da applicazione.

N.B.2 La relazione **M-N** tra **Corso** e **Studente** deve essere specificata esplicitamente nel file XML di mapping.