

Tecnologie Web T
21 Giungo 2019 – Compito

Tempo a disposizione: 3 ore

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

Chat.zip file zip contenente il sorgente java/class e pagine Web per punto 1
Matrice.zip file zip contenente il sorgente java/class e pagine Web per punto 2
Progetto.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 JSP, Java servlet e Javascript per la realizzazione di una **chat con moderatore automatico**.

L'applicazione deve consentire a utenti NON autenticati (non richiesta fase di autenticazione tramite username e password) di accedere a una pagina home.jsp che permetterà di inviare e visualizzare messaggi di testo. Una volta terminata la scrittura di un messaggio (automaticamente determinata dalla pressione del carattere '@'), il browser cliente deve trasformare in minuscoli eventuali caratteri maiuscoli contenuti nel messaggio e inviare il messaggio trasformato al server.

Il server deve controllare che il messaggio non contenga alcune parole "banned" indicate in una black list definita come parametro di inizializzazione della applicazione Web tramite file XML di deployment. Inoltre, il server deve determinare quale sottoinsieme di clienti dovrà essere destinatario e poter visualizzare i messaggi ricevuti secondo questa regola: i messaggi vengono sequenzializzati lato server; messaggi di posizione pari vengono inviati ai soli clienti con sessioni attive e identificatori di sessione pari; messaggi di posizione dispari ai clienti con identificatore di sessione dispari. Per rendere più veloce il controllo e lo smistamento dei messaggi, queste operazioni devono essere svolte **in concorrenza** da due "moderatori" server-side. Ci possono essere operazioni concorrenti che possono portare a inconsistenze? In quali casi? Nel caso di risposta positiva, descrivere quali sono esattamente i rischi di inconsistenza e in quali condizioni si possono verificare, come commento nel file sorgente.

ESERCIZIO 2 (11 punti)

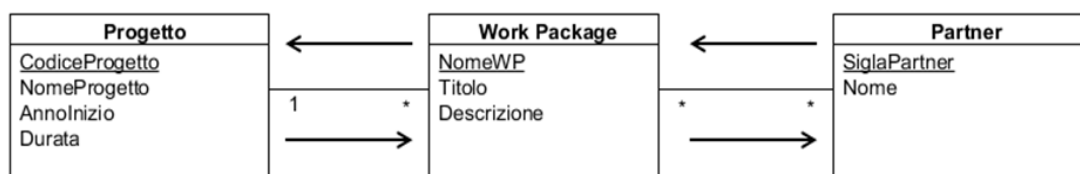
Si realizzi un'applicazione Web, basata su tecnologie Javascript, AJAX, JSON e Java servlet per la realizzazione di un servizio altamente concorrente di **calcolo matriciale** su matrici quadrate 4*4.

L'applicazione deve consentire a soli utenti autenticati (come al solito, tramite username e password) di accedere a una pagina home.html in cui è possibile specificare gli elementi di una riga delle matrici A e B; tale inserimento deve essere ripetuto 4 volte per le 4 righe delle matrici. Terminato l'inserimento, deve essere possibile invocare concorrentemente due servlet che si occuperanno **in parallelo** di calcolare la somma delle matrici A+B, ciascuna lavorando su due colonne diverse (prima servlet sulle colonne di posizione 0 e 1, seconda servlet sulle altre due colonne). Sia gli elementi delle matrici passati alle servlet sia il risultato parziale di calcolo matriciale restituito come risposta dalle servlet devono essere trasferiti in formato JSON.

Infine, si consideri che, per non sovraccaricare il sistema, ogni utente abbia la possibilità di invocare NON più di 5 calcoli matriciali all'interno di una sessione di interazione; la sessione di interazione avrà durata massima di 10 minuti e dovrà esserne forzata la terminazione dopo 2 minuti di inattività utente.

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 Progetto, Work Package e Partner** 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 di PK e FK), **implementato i JavaBean** e **realizzato le classi** relative al **Pattern DAO** per l'**accesso CRUD** alle tabelle, si richiede la realizzazione di **un metodo che restituisca i codici dei progetti per cui esiste almeno un partner che partecipa a tutti i suoi work package**. Si crei poi un **main di prova** in grado di: (i) inserire due o più tuple nelle tabelle di interesse; (ii) fare uso corretto del metodo realizzato al fine di produrre la stampa del risultato sul file **Progetto.txt**. Si risponda, infine, come commento al codice sorgente, alla domanda: “Per **implementare il metodo richiesto nel modo più efficiente possibile**, a quale tipo di **caricamento dei dati** occorre fare riferimento nella soluzione proposta e perché?”

N.B. L'implementazione del **Pattern DAO** deve **limitarsi** al solo **DBMS DB2**. La soluzione deve **sfruttare esplicitamente i mapping 1-N e N-M specificati nell'UML** e **propendere per il tipo di caricamento dei dati in grado di garantire l'implementazione più efficiente del metodo richiesto**. Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.