Tecnologie Web T
19 Luglio 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**):

Calcol.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
Function.zip	file zip contenente il sorgente java/class e pagine Web per punto 2
WP.zip	file zip contenente il sorgente java/class, file XML 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, JSP e cookie per la realizzazione di una **calcolatrice di gruppo**.
L'applicazione calcolatrice deve consentire a utenti autenticati (tramite username e password in pagina **login.html**; in fase di autenticazione all'utente sarà richiesto anche un *id* intero di gruppo di appartenenza) di scegliere un'operazione aritmetica (addizione, moltiplicazione, …), di inserire i due operandi (controllare che siano numeri reali, localmente al Web browser), di salvare il risultato ottenuto all'interazione precedente e di usare il risultato salvato descritto precedentemente come uno degli operandi (al posto dell'inserimento di uno di essi). Il mantenimento del risultato salvato deve essere ottenuto tramite cookie ed essere visibile a tutto il gruppo.
Infine, deve essere data la possibilità all'amministratore (pagina **admin.jsp**) di cancellare i risultati salvati, sia di uno specifico gruppo che di tutti i gruppi insieme; tutti i risultati salvati si cancellano comunque automaticamente una volta al giorno.

ESERCIZIO 2 (11 punti)

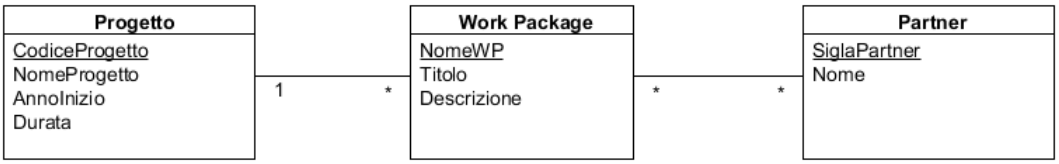
Si realizzi un'applicazione Web, principalmente basata su tecnologie Javascript, Ajax e JSON per il **calcolo della funzione f(x) invocata su n dati di ingresso in modo concorrente**.

L'applicazione deve consentire a utenti non autenticati di specificare il numero n di dati di ingresso da inserire e di specificare tali n valori; si controlli che n sia un numero naturale > 0. Una volta inserito l'ultimo valore di ingresso, il cliente deve richiedere concorrentemente l'esecuzione del calcolo di f(xi) al servitore, dove sarà ospitata una servlet/JSP per il calcolo della funzione f(x) dato il valore della sua variabile indipendente. Si definisca f(x) a piacere, ad esempio f(x)=x*x-7. Sia il trasferimento di xi dal cliente al servitore che il trasferimento del risultato f(xi) dal servitore al cliente devono avvenire in formato JSON.

Per evitare sovraccarichi, ogni cliente potrà effettuare al max N richieste totali per giorno. N ed n devono essere parametri che vengono definiti all'interno del file web.xml di configurazione dell'applicazione. Infine, deve essere data la possibilità all'amministratore (pagina **admin.jsp**) di visualizzare le statistiche (numero di richieste già invocate all'interno della giornata corrente) per tutti i clienti.

ESERCIZIO 3 (11 punti)

Partendo dalla realtà illustrata nel **diagramma UML** di seguito riportato, si fornisca una soluzione alla gestione della persistenza basata su tecnologia **Hibernate** in grado di "mappare" efficientemente e con uso di ID surrogati il modello di dominio rappresentato dai **JavaBean Partner, Work Package e Progetto del diagramma UML** con le corrispondenti **tabelle relazionali derivate dalla progettazione logica del diagramma** stesso.



Nel dettaglio, dopo aver creato da applicazione Java le tabelle all'interno del proprio **schema** nel database **TW_STUD** di **DB2** (esplicitando tutti i **vincoli** opportuni di PK e FK), 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 il progetto di nome "ATLAS", l'elenco dei nomi dei partner che vi partecipano*; ii) *per i partner il cui nome inizia per "U", l'insieme dei WP a cui partecipano suddivisi per nome progetto*;
- produrre una stampa opportunamente formattata dei risultati delle query al punto precedente sul file **WP.txt**;

il tutto, mediante opportuna gestione delle transazioni.

N.B. L'implementazione **deve limitarsi** al solo **DBMS DB2**. La soluzione Java **deve sfruttare esplicitamente i mapping 1-N e N-M specificati nell'UML**. Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.