

Tecnologie Web T
Prova d'Esame di Programmazione – 29 Giugno 2020 – Versione C

Tempo a disposizione: 90 minuti

La soluzione comprende la consegna elettronica dei seguenti file:

Seriale.zip file zip contenente il sorgente java/class e pagine Web per punto 1
Topolino.zip file zip contenente il sorgente java/class e file XML per punto 2

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), ben distribuiti sui 2 esercizi, ovvero in ciascuno dei due esercizi si deve raggiungere un punteggio di almeno 9 punti.

ESERCIZIO 1 (16,5 punti)

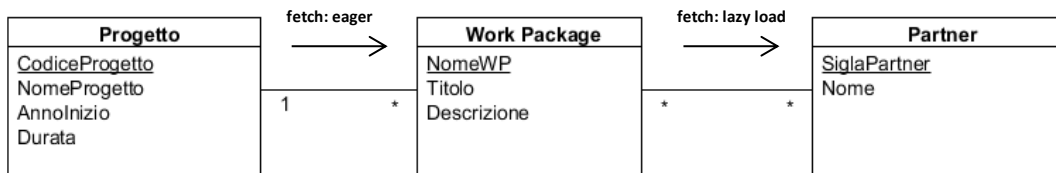
Si realizzi una applicazione Web per il **conteggio lato servitore di caratteri maiuscoli e minuscoli**, basandosi principalmente sulle tecnologie **Java Servlet e Javascript/AJAX**.

In particolare, l'applicazione Web deve chiedere all'utente di inserire la modalità desiderata ("seriale" o "parallelo") per il conteggio e un testo di 100 caratteri. Una volta terminata la digitazione del 100-esimo carattere del testo di ingresso, automaticamente senza pressione di tasti, la parte cliente dell'applicazione deve invocare il servizio lato servitore: in caso di modalità prescelta "**seriale**", due servlet S1 e S2 dovranno effettuare il conteggio una dopo l'altra, con S1 che trasferirà il testo di input a S2 dopo avere terminato il suo conteggio; nel caso "**parallelo**", invece, le due servlet S1 e S2 devono essere invocate ed eseguire concorrentemente. Il risultato del conteggio deve essere restituito al cliente in formato JSON.

Tecnologie Web T
Prova d'Esame di Programmazione – 29 Giugno 2020 – Versione C

ESERCIZIO 2 (16,5 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 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** e realizzato le **classi relative al Pattern DAO** per l'accesso **CRUD** alle tabelle, si richiede la realizzazione di un **metodo** che a partire dal progetto di nome “Topolino”, restituisca i titoli dei work package ad esso associati e, per ognuno di questi, l'elenco delle sigle dei partner che vi lavorano.

Si crei poi un main di prova in grado di:

- (a) inserire due o più tuple nelle tabelle di interesse;
- (b) fare uso corretto dei metodi realizzati ai punti precedenti al fine di produrre la stampa dei risultati sul file **Topolino.txt**;

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