

**Tecnologie Web T**  
**Prova d'Esame di Programmazione – 9 Giugno 2020 – Versione D**

**Tempo a disposizione: 90 minuti**

---

La soluzione comprende la consegna elettronica dei seguenti file:

<b>Conteggio2.zip</b>	file zip contenente il sorgente java/class e pagine Web per punto 1
<b>Bam.zip</b>	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 di caratteri** su un input testuale inserito dall'utente, basandosi principalmente su tecnologie **Javascript/Ajax** e **Java Servlet**.

In particolare, l'applicazione Web deve partire da una pagina che consenta all'utente, autenticato tramite username e password, di inserire testo libero fino alla digitazione del carattere speciale '\$'. Non appena viene digitato il carattere speciale '\$', il testo inserito deve essere automaticamente inviato server-side per il processamento, senza pressione esplicita di pulsanti da parte dell'utente.

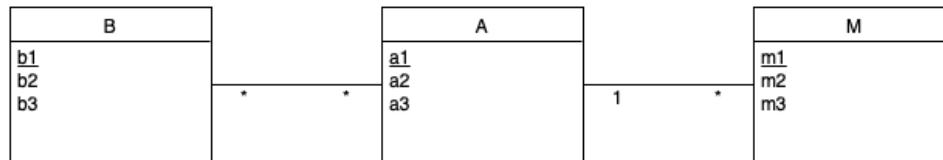
Il processamento server-side deve avvenire **in modo concorrente da parte di tre servlet S1, S2 e S3**. S1 deve ricevere l'intero testo di input e contare le occorrenze di caratteri minuscoli; analogamente S2 deve ricevere l'intero testo di input e contare le occorrenze di caratteri non alfabetici; infine S3 deve contare le occorrenze di caratteri numerici. S1, S2 e S3 devono lavorare concorrentemente e inviare il risultato del loro conteggio al cliente.

Inoltre, l'applicazione deve tenere traccia di quante operazioni di conteggio sono state effettuate da ciascun cliente e complessivamente da tutta l'applicazione.

**Tecnologie Web T**  
**Prova d'Esame di Programmazione – 9 Giugno 2020 – Versione D**

**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 **Hibernate** in grado di “mappare” efficientemente e con uso di ID surrogati il modello di dominio rappresentato dai **JavaBean** “B”, “A” e “M” 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:

- istanziare alcuni JavaBean “B”, “A” e “M”, rendendoli persistenti rispetto alla base di dati associata al diagramma UML;
- restituire: (i) per un prefissato valore dell'attributo chiave *m1*, restituire i valori degli attributi *a2* e *a3* dell'istanza di *A* a esso associata, assieme alla lista di valori per l'attributo *b3* delle istanze di *B*; (ii) per ogni valore di chiave *b1*, l'insieme delle istanze di *A* a esso associate per cui valga la condizione  $a2 \neq b2$ ; **producendo una stampa opportunamente formattata del risultato** sul file **Bam.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.