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

Tempo a disposizione: 90 minuti

La soluzione comprende la consegna elettronica dei seguenti file:

Composizione.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
Pio.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.

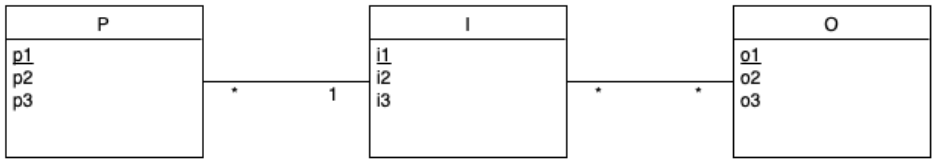
N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), ben distribuiti sui 2 esercizi, ovvero in ciascuno dei due esercizi si deve raggiungere un punteggio di almeno 9 punti.

ESERCIZIO 1 (16,5 punti)

Si realizzi una applicazione Web per il processamento di testo di input tramite **composizione di filtri server-side**, basandosi principalmente sulle tecnologie **Java Servlet** e **JSP**.
In particolare, l'applicazione Web deve partire da una pagina Web che consenta all'utente di inserire testo libero fino a 100 caratteri. Digitato il 100esimo carattere, 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 avviene in questo modo: una prima servlet S1 deve ricevere il testo di input ed eliminare tutte le eventuali cifre numeriche contenute. L'output di S1 deve essere inviato in ingresso a una pagina JSP S2 che trasforma i caratteri maiuscoli ricevuti nei corrispondenti minuscoli. Infine, l'output di S2 viene dato in ingresso a una servlet S3 che deve eliminare i caratteri non alfabetici presenti e girare la risposta finale al cliente. La risposta finale al cliente deve anche contenere la frase "Risposta alla **i-esima** richiesta di questo utente", dove **i** conta il numero di richieste dello stesso utente all'interno della sessione di interazione.
Lo scambio dati fra S2 e S3 e lo scambio dati fra S3 e il cliente devono avvenire in formato JSON.

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

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** "**P**", "**I**" e "**O**" **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 "P", "I" e "O", rendendoli persistenti rispetto alla base di dati associata al diagramma UML;
- restituire: (i) per un prefissato valore dell'attributo chiave *p1*, restituire i valori degli attributi *i2* e *i3* dell'istanza di *I* a esso associata, assieme alla lista di valori per l'attributo *o3* delle istanze di *O*; (ii) per ogni valore di chiave *o1*, l'insieme delle istanze di *I* a esso associate per cui valga la condizione $i2 \neq o2$; **producendo una stampa opportunamente formattata del risultato** sul file **Pio.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.