

Tecnologie Web T
18 Gennaio 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>):

Sessione.zip file zip contenente il sorgente java/class e pagine Web per punto 1
Takeaway.zip file zip contenente il sorgente java/class, file xml/xsd e txt per punto 2
Ricetta.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 uno strumento di gestione per sessioni di una applicazione Web qualsiasi, basato principalmente su tecnologia Java servlet, JSP e Javascript.

L'applicazione Web dovrà prevedere sia utenti connessi dopo autenticazione (tramite username e password) sia senza autenticazione. L'implementazione della logica di business dell'applicazione Web può essere lasciata vuota; semplicemente deve essere possibile per un utente aprire una sessione di interazione (con autenticazione o meno) e terminarla (terminazione automatica di default dopo 10 minuti).

Lo strumento di gestione deve invece consentire all'amministratore dell'applicazione (username=admin; password=admin), tramite una pagina JSP, di visualizzare le statistiche relative alle sessioni correnti (ovvero ancora non terminate) e recenti (terminate non oltre 6 ore). Le statistiche devono includere: ora di inizio e di fine sessione; username associato (in caso di autenticazione); numero totale di sessioni correnti e recenti per ogni username e globalmente per non autenticati.

Inoltre, lo strumento di gestione deve permettere all'amministratore, tramite una servlet, di copiare tutto lo stato di sessione di una sessione corrente verso un'altra sessione corrente e di visualizzarlo a schermo (come sequenza di nomi di attributo, tralasciandone i valori); il trasferimento di tali dati verso il cliente Web deve essere effettuato tramite JSON. L'amministratore potrà ovviamente scegliere sessione_sorgente e sessione_destinazione fra tutte le sessioni correnti.

Inoltre, si risponda, come commento al codice sorgente della servlet:

- se uno stesso utente (sia autenticato sia no) cerca di aprire più sessioni correnti, che cosa succede? Problemi di concorrenza e corse critiche?
- se più amministratori cercano di effettuare concorrentemente copie che impattano sulle stesse sessioni correnti, che cosa succede? Problemi di concorrenza e corse critiche?

Tecnologie Web T
18 Gennaio 2019 – Compito

ESERCIZIO 2 (11 punti)

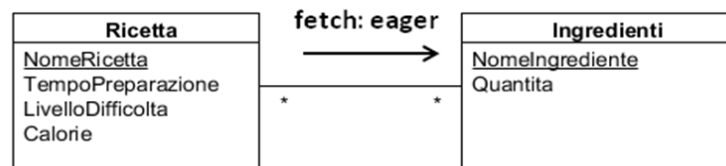
Si progetti una grammatica **XML Schema**, e un suo **documento XML** di esempio, per la modellazione delle informazioni relative al servizio on-line “**Ristoranti Takeaway**”, nel rispetto delle seguenti specifiche:

- Il servizio comprende i migliori ristoranti di una data località presso i quali l’utente può effettuare un’ordinazione di portate pronte al consumo.
- Ogni ristorante è caratterizzato da un nome (obbligatorio), da una località (obbligatoria), da un indirizzo completo (obbligatorio) - composto da via, numero civico e CAP - da un numero di telefono (obbligatorio) - costituito da prefisso e numero - e da un link a sito web dedicato se presente (opzionale).
- Inoltre, ogni ristorante espone le proprie offerte dividendo i piatti in cinque diverse categorie; nello specifico, antipasti, primi, secondi, contorni e dessert. Ogni categoria riporta l’elenco dei nomi dei piatti offerti, con associata la lista degli ingredienti che lo compongono e il relativo prezzo.

Si realizzi quindi l’applicazione Java “**Takeaway**” che, facendo uso del **parser SAX** e del **documento XML di esempio** realizzato al punto precedente, esponga il metodo `getNumeroPortateOfferte()`, unitamente a suo un `main` di prova in grado di calcolare il **numero di portate offerte**. Stampare il risultato prodotto dal metodo `getNumeroPortateOfferte` sul file **Takeaway.txt**.

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 Ricetta e Ingredienti 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), **implementato i JavaBean e realizzato le classi** relative al **Pattern DAO** per l’accesso **CRUD** alle tabelle, si richiede l’implementazione di un metodo in grado di restituire i **nomi delle ricette più “ricche”/più “povere”**, ovvero composte dal maggior/minor numero di ingredienti.

Si crei poi un **main di prova** in grado di:

- inserire due o più tuple nelle tabelle di interesse;
- fare uso corretto del metodo realizzato al punto precedente al fine di produrre la stampa del risultato sul file **Ricetta.txt**.

N.B. L’implementazione del **Pattern DAO** deve limitarsi al solo **DBMS DB2**. La soluzione deve sfruttare il **mapping N-M** specificato nell’UML e **propendere per il caricamento indicato nello stesso**. Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.