

Progettazione di Applicazioni Web T
Prova d'Esame – 24 gennaio 2023

Tempo a disposizione: 2 ore e 30 minuti

La soluzione comprende la consegna elettronica dei seguenti file:

Shopping.zip	file zip contenente il sorgente java/class, file XML/XSD e txt per il punto 1
Partite.zip	file zip contenente il sorgente java/class e file txt per il punto 2
WebRatio.zip	file zip contenente il testo di risposta (txt/doc/pdf) al 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, codice java e relativi .class, descrittori XML, file txt/doc/pdf, ecc.) e NON dell'intero progetto.

N.B. La prova si intende superata se il punteggio finale non è inferiore a 18 punti (su un totale di 33). In particolare, è necessario (ma non sufficiente) ottenere la sufficienza in tutti gli esercizi.

ESERCIZIO 1 - 14 punti (sufficienza: 7 punti)

Si progetti una **grammatica XML Schema**, assieme a un suo **documento XML** di esempio, in grado di modellare le informazioni di una **applicazione Web per Shopping On-line** di abbigliamento femminile, nel rispetto delle seguenti specifiche:

- Ciascun documento XML modella una sessione di “visita” di un utente registrato al sito Web. Nel dettaglio, si tiene traccia di (i) informazioni utente quali e-mail (di tipo email e obbligatorio) e password (di tipo password e obbligatorio) (ii) scelte dell’utente dal menù di navigazione; in particolare, le scelte rientrano nell’insieme {“abiti”, “camice”, “giacche”, “gonne”, “pantaloni”, “ultimi arrivi”} e sono non esclusive. Si noti che è obbligatoria almeno una scelta di visita.
- Il tipo password è una stringa di 8 caratteri che deve contenere almeno un numero e un carattere speciale tra {?!*}\$}; il tipo e-mail è una stringa formata da due sottostringhe separate dal carattere speciale “@” e in cui la seconda sottostringa contiene un “.”
- Ogni scelta di visita, è modellata a sua volta come un insieme di “oggetti” complessi (da 1 a N), ognuno formato dalla tripla: fotografia capo abbigliamento (rappresentata dal nome file jpeg), descrizione capo (di tipo testuale) e prezzo di vendita (espresso in Euro). Tutti e tre i campi sono obbligatori. Durante la navigazione, l’utente può poi selezionare da 0 a N capi d’abbigliamento in base al suo interesse.

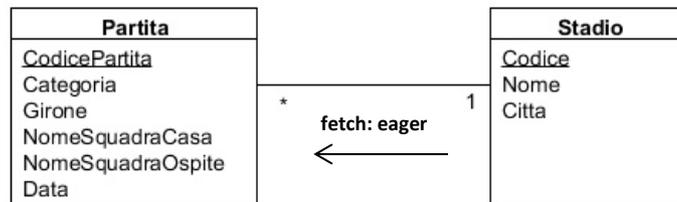
Si realizzi quindi l’**applicazione Java “Shopping”** che, facendo uso del **parser SAX** e del **documento XML di esempio** realizzato al punto precedente, esponga il metodo `getScelte()`, unitamente a suo un `main` di prova, in grado di restituire la lista di scelte effettuate dall’utente durante la sessione di navigazione, ognuna delle quali caratterizzata dagli oggetti selezionati dall’utente. Il tutto, a fine d’indagine di mercato e analisi di preferenze utente. Si stampi il risultato prodotto opportunamente formattato sul file **Shopping.txt**.

Progettazione di Applicazioni Web T

Prova d'Esame – 24 gennaio 2023

ESERCIZIO 2 - 14 punti (sufficienza: 7 punti)

Partendo dalla realtà illustrata nel **diagramma UML** di seguito riportato, si fornisca una soluzione alla gestione della persistenza basata su **metodologia DAO** in grado di “mappare” efficientemente, e con uso di ID surrogati, il modello di dominio rappresentato dai **JavaBean Partita** e **Stadio** del **diagramma UML** con le corrispondenti **tabelle relazionali derivate dalla progettazione logica** del diagramma stesso.



Nel dettaglio, dopo aver creato da applicazione Java gli **scemi 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 *“Dato uno stadio, il numero totale di partite in esso giocate raggruppate per categoria”*.

Si richiede quindi di realizzare una **classe di prova** in grado di:

- inserire due o più partite per ogni stadio coinvolto nel campionato;
- fare uso corretto del metodo realizzato al punto precedente al fine produrre una stampa (opportunamente formattata) del risultato sul file **Partite.txt**.

N.B. L'implementazione **deve limitarsi** al solo **DBMS DB2**. La soluzione **deve sfruttare i mapping specificati nello schema UML**. Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.

ESERCIZIO 3 - 5 punti (sufficienza: 3 punti)

Considerando l'ipertesto progettato nello **schema IFML di WebRatio** di seguito riportato, il candidato descrive dettagliatamente le relazioni esistenti tra le componenti pagina, link e condizioni espresse sulle componenti della *site view SV*.

SV

