

Tecnologie Web T
3 Luglio 2017 – Compito

Tempo a disposizione: 3 ore

La soluzione comprende la **consegna elettronica** dei seguenti file mediante l'apposito applicativo Web **esamix** (<http://esamix.labx>):

Split_Counting.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
UCL.zip	file zip contenente il sorgente java/class e pagine Web per punto 2
Pallacanestro.zip	file zip contenente il sorgente java/class e il file 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 un'applicazione Web, principalmente basata su tecnologie **JSP** e **servlet**, per il **conteggio concorrente di caratteri maiuscoli o non alfabetici** in file selezionati dall'utente e mantenuti server-side.

L'applicazione deve avere una **home.jsp** tramite la quale viene mostrata una lista, dinamicamente determinata server-side, dei nomi di file correntemente presenti nel direttorio *Split_Counting*; tale pagina deve dare la possibilità all'utente di selezionare un file dalla lista e di scegliere se andare a contare i caratteri maiuscoli (opzione 1) oppure i caratteri non alfabetici (opzione 2).

Una volta effettuata la selezione, senza pressione di alcun pulsante, il conteggio deve cominciare ad essere eseguito in modo concorrente da due servlet: la prima servlet deve effettuare il conteggio selezionato sulla prima metà del file; la seconda servlet deve occuparsi del conteggio sulla seconda metà dello stesso file in modo concorrente. Si giustifichi nel codice sorgente tramite commento la scelta effettuata in termini di modello concorrente di esecuzione (ad esempio, quanti thread? Quanti oggetti? Possibili problemi di accesso concorrente a che cosa?). Ovviamente deve essere restituito al cliente il risultato complessivo del conteggio, ottenuto come somma dei risultati parziali calcolati da ciascuna delle due servlet.

Infine, ci si assicuri che ciascun utente non possa effettuare più di 3 operazioni di conteggio all'interno di una singola sessione, al fine di non sovraccaricare troppo le risorse computazionali lato server.

Tecnologie Web T

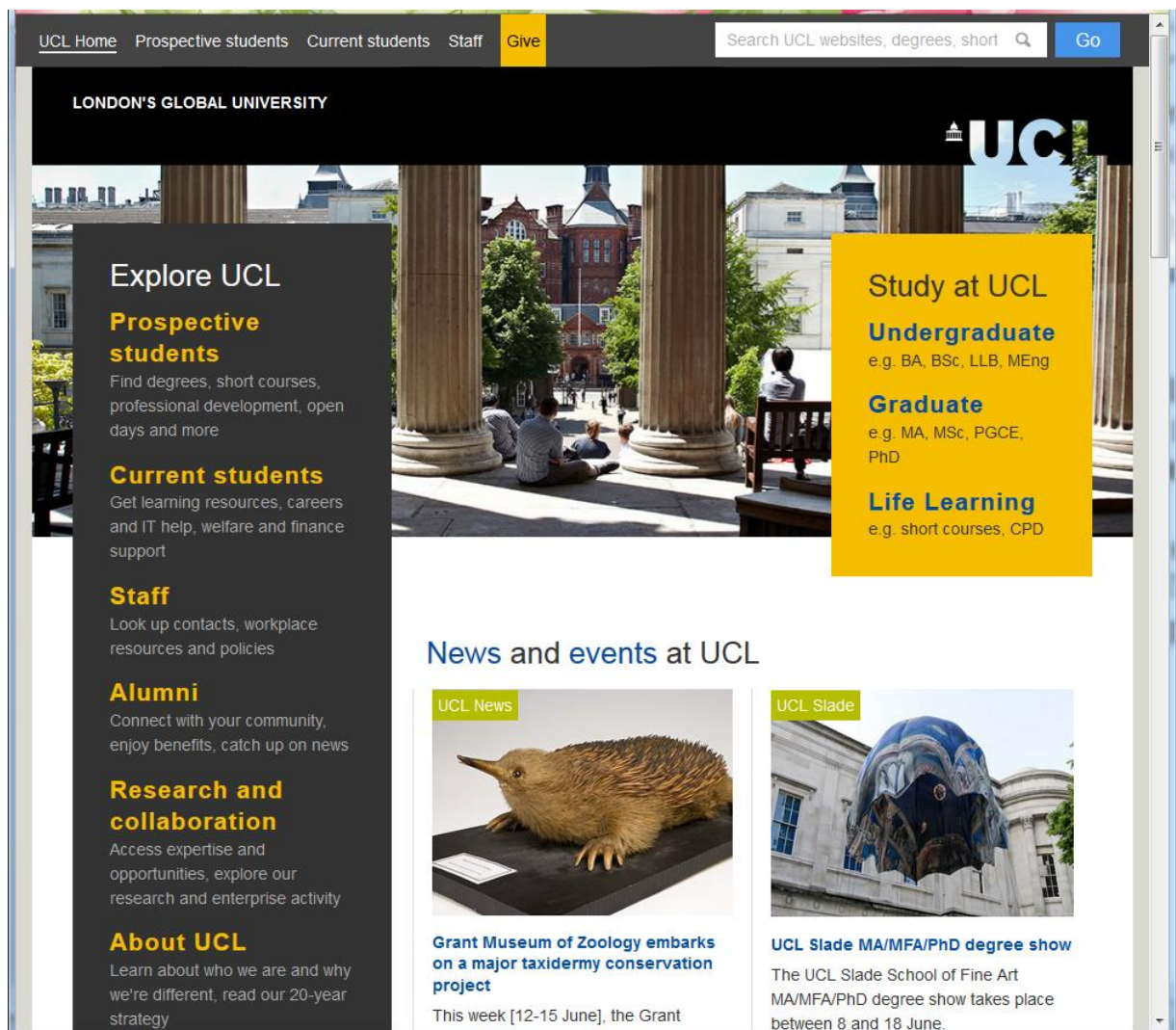
3 Luglio 2017 – Compito

ESERCIZIO 2 (11 punti)

Si realizzino le **pagine Web dinamiche** (basate su **tecnologia HTML, CSS, e Javascript**) in grado di **riprodurre il contenuto e il layout grafico** dello “snapshot” del sito Web di seguito riportato. Nello specifico, la pagina visualizza il contenuto relativo alla voce “UCL Home” del Menù generale “orizzontale” in alto a sinistra. Si preveda inoltre che il campo di ricerca in alto a destra possa ammettere solo stringhe alfabetiche maiuscole composte da un massimo di tre parole.

Per i due menù verticali a sfondo grigio e giallo rispettivamente, si considerino **link** le soli parti testuali scritte in “giallo” (per il menù a sfondo grigio posto a destra) e in “azzurro” (per il menù a sfondo giallo posto a sinistra). Per la parte in basso a destra relativa alle new e agli eventi presso UCL, rappresentano **link** le soli parti testuali in “azzurro” e quelle in “bianco” contornate dai piccoli riquadri verdi inseriti nelle immagini in alto a sinistra.

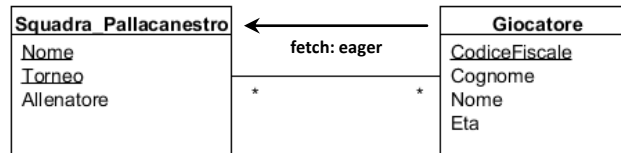
N.B. La soluzione NON deve far uso del costrutto HTML `frame`. Per rappresentare le immagini riportate nello snapshot, si utilizzino figure di esempio a piacere.



Tecnologie Web T
3 Luglio 2017 – Compito

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* del diagramma UML con le corrispondenti *tabelle relazionali* derivate dalla progettazione logica del diagramma stesso. Nel dettaglio, la gestione degli ID surrogati è demandata a un'apposita classe `IdBroker()` la cui logica si deve basare sull'uso di campi “auto increment” (ovvero, `identity column` in DB2).



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 della seguente operazione* (a livello di logica di business):

- “Per ogni giocatore, i nomi degli allenatori delle squadre in cui ha giocato”.

Si crei un *main di prova* che:

- (i) inserisca due o più tuple nelle tabelle di interesse;
- (ii) faccia uso corretto dell'operazione realizzata al punto precedente al fine di produrre una stampa opportunamente formattata del risultato sul file **Pallacanestro.txt**.

Lo studente giustifichi opportunamente tramite commento nel codice la scelta della propria modellazione di JavaBean e tabelle.

N.B. L'implementazione del pattern DAO deve limitarsi al solo DBMS DB2. La *soluzione deve sfruttare direttamente i mapping M-N specificati nell'UML e propendere per il caricamento dei dati indicato nello stesso*.