

Tecnologie Web T
04 Febbraio 2015 – Compito

Tempo a disposizione: 3 ore

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

Questionario.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
Prodotto.zip	file zip contenente le pagine Web (html, css e js) per punto 2
Maratona.zip	file zip contenente il sorgente java/class 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** che permetta a utenti autenticati di rispondere a domande a risposta multipla caricate da un file di configurazione server-side, ricevendo al termine della loro sessione la valutazione delle risposte date. L'applicazione deve essere basata su **tecnologie Java servlet, Javascript e JSON**.

In particolare, l'applicazione dovrà avere una pagina iniziale **start** che richieda all'utente nome e cognome (considerati identificatori univoci) e che gli permetta di aprire una sessione di interazione. Dopo l'apertura della sessione verranno visualizzate, in una o più pagine generate dinamicamente, le domande a risposta multipla a cui l'utente dovrà rispondere. Il testo di domande e risposte multiple viene estratto da un file domande.txt, presente sul file system server-side, formattato nel seguente modo:

```
domanda1
rispostala; rispostalb; rispostalc; rispostald;
a|b|c|d
domanda2
...
```

Al termine della lista delle domande deve essere presente un pulsante di consegna, che serve a inviare le risposte e chiedere la loro valutazione (+1 se risposta corretta, -0.3 se risposta sbagliata, 0 se risposta non selezionata); i dati relativi alle risposte selezionate e al risultato finale devono essere scambiati tramite formato JSON.

Inoltre, si preveda che, nel caso in cui l'utente non prema il tasto di consegna entro 30 minuti dall'inizio della sessione, la sessione sia comunque terminata, portando comunque alla valutazione delle eventuali risposte fornite fino a quell'istante. alla cartella.

Tecnologie Web T
04 Febbraio 2015 – Compito

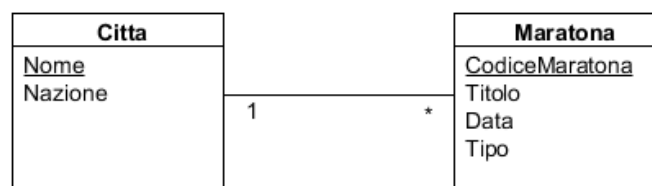
ESERCIZIO 2 (11 punti)

Si realizzi la **pagina HTML dinamica MioProdotto.html** basata su tecnologia **Javascript** che rispetti le seguenti specifiche:

- layout grafico pagina (confinato in opportuno file **MioProdotto.css**). La pagina è divisa orizzontalmente in tre parti di diversa altezza: i) la parte alta (20% dell'altezza totale) è dedicata al titolo della pagina, ha sfondo grigio e testo (*Comic, Bold, Italic, 16pt*) fucsia. Nel dettaglio, sul lato sinistro riporta il titolo "**MioProdotto**" mentre sul lato destro l'immagine del logo del sito (rappresentato dall'immagine "**logo.jpeg**") assieme a un piccolo form per l'autenticazione dell'utente; cliccando sul logo è possibile accedere al sito personale del progettista che lo ha ideato ("<http://www.iodesigner.it>"); ii) la parte centrale occupa il 10% dell'altezza totale e ospita un menu orizzontale con la classificazione dei prodotti, ovvero *categoria A, categoria B, categoria C, categoria D* e *categoria E*, (ogni categoria riporta 3 sottocategorie selezionabili dall'utente, ovvero *sottocategoria X1, sottocategoria X2* e *sottocategoria X3*); lo sfondo è ancora grigio e il testo *Arial, Bold, 11pt*; iii) la parte restante (pari al 70% dell'altezza totale) ospita la visualizzazione sequenziale su sfondo bianco delle immagini dei prodotti selezionati dall'utente. A titolo di esempio, per i prodotti di sottocategoria *sottocategoria X1*, verranno visualizzate le immagini *sottocategoriaX1-1.jpeg, sottocategoriaX1-2.jpeg, ... sottocategoriaX1-N.jpeg*; per semplicità si suppone che *N* sia costante;
- contenuto form: **login e password utente** (caselle di testo editabili lunghe 12 e 8 caratteri, rispettivamente); **bottone invia**;
- controllo eventi: **A) mentre l'utente edita i campi del form** i) subito dopo che l'utente ha editato il campo login, deve verificare che tale campo non contenga numeri; ii) subito dopo che l'utente ha editato il campo password, deve verificare che tale campo contenga almeno 2 numeri, 1 lettera maiuscola e un carattere speciale tra *!, \$ e ?*; **B) al momento dell'invio del contenuto del form** deve verificare che i campi login e password non siano vuoti.

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 il modello di dominio rappresentato dai **JavaBean Citta** e **Maratona** con le corrispondenti **tabelle relazionali derivate dalla progettazione logica del diagramma** dato.



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 la **realizzazione di un metodo** che permetta di ottenere:

- il numero totale di maratone di tipo "mezza maratona" tenutesi a New York.

Infine, si crei un semplice *main* di prova che:

- inserisca due o più maratone di tipo diverso tenutesi in una stessa città e in città diverse;
- faccia uso corretto del metodo realizzato al punto precedente al fine produrre una stampa (opportunamente formattata) del risultato sul file **Maratona.txt**.

N.B. L'implementazione del **Pattern DAO** deve limitarsi al solo **DBMS DB2**.