

**Tecnologie Web T**  
**7 Febbraio 2020 – Compito**

**Tempo a disposizione: 3 ore**

---

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

<b>Tennis.zip</b>	file zip contenente il sorgente java/class e pagine Web per punto 1
<b>MINIVeg.zip</b>	file zip contenente il sorgente java/class, xsd, xml e txt per punto 2
<b>BIGProgetto.zip</b>	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 un'applicazione Web, principalmente basata su tecnologie Java servlet, JSP e Javascript, per la gestione delle prenotazioni di **campi da tennis**.

L'applicazione Web deve consentire a un nuovo utente di registrarsi, indicando il proprio nome utente, la propria password e il proprio livello di abilità (da A a E, con livello di abilità decrescente). Utenti già registrati possono accedere al servizio di prenotazione dopo avere effettuato il login, senza ri-inserire il loro livello di abilità.

Dopo l'autenticazione l'utente ha la possibilità di prenotare un campo (campi con identificatore da 1 a 4, prenotazione di un'ora) o di aggiungersi a un campo prenotato se il livello di abilità del prenotante è uguale al suo e se c'è ancora posto (per semplicità, si suppongano solo partite di singolare, non di doppio). Tutti i dati inseriti in input devono essere verificati nel loro formato e range localmente al browser.

Inoltre, se la prenotazione è stata effettuata da più di 72 ore e ancora il gruppo di due giocatori non si è completato, il campo deve ritornare allo stato di libero e il "prenotante" deve essere notificato, al suo successivo login, della cancellazione della prenotazione.

Infine, deve essere data la possibilità all'amministratore (previa autenticazione, pagina **admin.jsp**) di visualizzare lo stato complessivo dei campi in una data specificata, con l'indicazione per ogni prenotazione "incompleta" di quanti minuti residui rimangono prima della cancellazione automatica della prenotazione.

## Tecnologie Web T

### 7 Febbraio 2020 – Compito

#### **ESERCIZIO 2 (11 punti)**

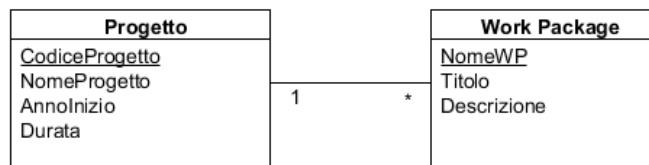
Si progetti una grammatica **XML Schema**, e un suo **documento XML** istanza, per la modellazione delle informazioni del negozio “**MINIVeg**”, nel rispetto delle seguenti specifiche:

- I prodotti disponibili sono caratterizzati da un codice prodotto, da un nome, da una categoria di appartenenza e da un prezzo al Kg; il prezzo è espresso in *Euro*, rappresentato da una parte intera e da una parte decimale composta esattamente da due cifre decimali. Parte intera e parte decimale sono separate dal “.”.
- Le categorie sono due e rispettivamente verdura (“V”) e frutta (“F”).
- La categoria verdura comprende i soli prodotti: *carota, zucchina, finocchio, pomodoro e sedano*; la categoria frutta include i soli prodotti: *mela, pera, arancia e mandarino*.

Si realizzi quindi l'**applicazione Java “MINIVeg”** che, facendo uso sia del parser SAX che del parser DOM, esponga i metodi `getFruttaMaxPrezzo()` e `getVerduraMinPrezzo()`, unitamente a un `main` di prova, in grado di testare i metodi sul documento XML istanza definito e restituire il nome di frutta che costa di più e di nome di verdura che costa di meno, rispettivamente. Si produca infine la stampa opportunamente formattata di quanto richiesto sul file **MINVeg.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 tecnologia **Hibernate** in grado di “mappare” efficientemente e con uso di ID surrogate il modello di dominio rappresentato dai **JavaBean Progetto e Work Package 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 “Progetto”** e relativi **JavaBean “Work Package”**, rendendoli persistenti rispetto alla base di dati associata al diagramma UML;
  - selezionare codice e nome dei progetti con maggior numero di **work package** associati, producendo una stampa opportunamente formattata del risultato sul file **BIGProgetto.txt**;
  - eliminare i progetti restituiti al punto precedente;
- il tutto, **mediante opportuna gestione delle transazioni**.

**N.B.** L'implementazione **deve limitarsi** al solo **DBMS DB2**. La soluzione Java **deve sfruttare esplicitamente il mapping specificato nell'UML**. Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.