

# Prova d'Esame—Tecnologie Web L-A

A.A. 2008/2009

**Tempo Disponibile: 3 ore**

*Ai candidati è consentita la consultazione di manuali e materiale didattico*

## Esercizio 1. Modellazione

Si progetti il DTD per documenti XML che rappresentano l'orario dei treni che fermano in una stazione ferroviaria. L'orario si riferisce ad una ed una sola stazione. La stazione ha un nome, un codice identificativo ed un indirizzo. L'indirizzo è caratterizzato da città, via e numero civico. L'orario è inoltre caratterizzato da una lista di treni in partenza. Ogni treno ha un nome, un codice, un orario di partenza ed una lista di fermate. Ogni fermata è caratterizzata da un nome e da un orario.

## Esercizio 2. Programmazione Client-Side

Si richiede di implementare la sola parte client-side di un applicativo Web in tecnologia AJAX per il reperimento di informazioni sui treni disponibili fra due destinazioni specificate. La pagina è costituita da una form contenente due campi di testo editabili dall'utente. I campi di testo devono consentire all'utente di inserire il nome di due stazioni ferroviarie di suo interesse. La form è dotata di un bottone che consente di inviare la request al server. Le informazioni devono essere inviate al server in modo asincrono tramite una chiamata GET e passando le stringhe che rappresentano il nome delle stazioni di partenza e di destinazione. A seguito della pressione del bottone la pagina deve:

- verificare che i nomi delle stazioni di partenza e destinazione siano costituiti da soli caratteri alfabetici. In caso contrario deve essere segnalato il problema inserendo sotto la form un messaggio che informa l'utente dell'errore;
- nel caso in cui entrambe i nomi delle stazioni siano stati inseriti correttamente viene inviata la richiesta al server. Per semplicità si assume che esistano sempre le stazioni specificate dall'utente. Si richiede che i risultati contenuti nella risposta del server vengano presentati in una tabella costituita da tre righe ed una colonna (vedi Tabella 1) che deve essere inserita sotto la form. In particolare, la prima riga riporta il nome della stazione di partenza, la seconda il nome della stazione di destinazione e, infine, la terza riga riporta la lista dei codici dei treni che collegano le stazioni ed i loro orari;

<i>Nome della Stazione di Partenza</i>
<i>Nome della Stazione di Destinazione</i>
<ul style="list-style-type: none"><li>● <i>CodiceTreno, OrarioPartenza</i></li><li>● <i>CodiceTreno, OrarioPartenza</i></li><li>● <i>CodiceTreno, OrarioPartenza</i></li><li>...</li><li>● <i>CodiceTreno, OrarioPartenza</i></li></ul>

*Tabella 1. modello della tabella che deve essere inserita sotto la form*

- nel caso in cui i nomi delle stazioni di partenza e di destinazione siano stati inseriti correttamente, ma non esistano treni che le collegano, si deve segnalare il problema inserendo sotto la form un messaggio che informa l'utente dell'errore.

Si assume che il server restituisca i dati sui treni in formato XML come segue:

```
<?xml version='1.0' encoding='UTF-16'?>
  <orario>
    <partenza> nome della stazione di partenza </partenza>
    <destinazione> nome della stazione di destinazione </destinazione>
    <listatreni>
      <treno cod="codice treno">
        orariopartenza
      </treno>
      <treno cod="codice treno">
        orariopartenza
      </treno>
      .....
      <treno cod="codice treno">
        orariopartenza
      </treno>
    </listatreni>
  </orario>
```

dove *nome della stazione di partenza* e *nome della stazione di destinazione* rappresentano rispettivamente il nome della stazione di partenza e di destinazione; *codice treno* rappresenta il codice del treno; *orariopartenza* l'orario di partenza del treno. Nel caso in cui non siano disponibili treni fra le stazioni di partenza e di destinazione specificate si assume che il server restituisca un documento XML privo delle indicazioni sui treni disponibili (vedi esempio sotto).

```
<?xml version='1.0' encoding='UTF-16'?>
  <orario>
    <partenza> nome della stazione di partenza </partenza>
    <destinazione> nome della stazione di destinazione </destinazione>
    <listatreni>
    </listatreni>
  </orario>
```

Nota bene: non si richiede di generare due diverse risposte XML da parte del server, ma solo di prevedere nel codice Javascript la gestione dei due diversi tipi di risultato.

### Esercizio 3. Programmazione Server-Side

Si richiede di implementare una sezione dell'applicazione Web per la gestione dell'orario di una stazione ferroviaria. In particolare, si richiede che il candidato implementi due distinte JSP che consentano di inserire nuovi treni (*InserisciTreno.jsp*) e di leggere l'orario dei treni in partenza verso una stazione specificata dopo un'ora specificata (*LeggiTreni.jsp*). Le JSP hanno accesso ad una comune base di dati, costituita da una sola tabella strutturata come segue :

<i>NomeTreno</i>	<i>TipoTreno</i>	<i>OrarioPartenza</i>	<i>Destinazione</i>
<i>NomeTreno</i>	<i>TipoTreno</i>	<i>OrarioPartenza</i>	<i>Destinazione</i>
<i>NomeTreno</i>	<i>TipoTreno</i>	<i>OrarioPartenza</i>	<i>Destinazione</i>

<i>NomeTreno</i>	<i>TipoTreno</i>	<i>OrarioPartenza</i>	<i>Destinazione</i>
------------------	------------------	-----------------------	---------------------

dove *NomeTreno* è una stringa che rappresenta il nome del treno, *TipoTreno* è una stringa che rappresenta il tipo di treno (es. regionale, treno interregionale, ...), *OrarioPartenza* rappresenta l'orario di partenza del treno (per semplicità si assume che sia un intero), *Destinazione* è una stringa che rappresenta il nome della stazione di destinazione.

La JSP *InserisciTreno.jsp* consente di inserire un nuovo treno nel database. La JSP richiede il passaggio di *NomeTreno*, *TipoTreno*, *OrarioPartenza*, e *Destinazione* via HTTP POST. La JSP, una volta ricevuti i parametri, verifica che non esista nel database alcun treno con *OrarioPartenza*, e *Destinazione* specificati. Nel caso in cui i controlli vadano a buon fine, la JSP inserisce il nuovo treno nella base di dati e restituisce all'utente una pagina di conferma dell'avvenuto inserimento. In caso contrario viene presentata all'utente una pagina in cui si segnala che si sono presentati errori nell'inserimento.

La JSP *LeggiTreni.jsp* consente di presentare all'utente una pagina che visualizza tutti i treni in partenza verso una certa destinazione dopo un orario specificato. A questo fine, la JSP richiede il passaggio di una richiesta HTTP POST che specifica il nome della destinazione e l'orario dopo cui l'utente intende partire. Una volta ricevuta la richiesta, la JSP presenta all'utente una pagina in cui è inclusa la lista di tutti i treni che verificano i criteri di ricerca. La lista è strutturata come segue:

- *NomeTreno, TipoTreno, OrarioPartenza*
- *NomeTreno, TipoTreno, OrarioPartenza*
- *NomeTreno, TipoTreno, OrarioPartenza*
- ...
- *NomeTreno, TipoTreno, OrarioPartenza*

Nel caso in cui non esista nel database alcun treno in partenza che soddisfa i criteri di ricerca, viene mostrata all'utente una pagina che segnala l'errore. In particolare, viene specificato all'utente se non esistono treni per la destinazione specificata oppure se esistono treni per la destinazione specificata, ma solo ad orari precedenti a quello specificato.

Si richiede che il candidato implementi l'accesso al database utilizzando il pattern DAO. Si richiede inoltre che il database sia implementato utilizzando un DBMS relazionale (es. hsqldb) oppure, in alternativa, utilizzando semplici file di testo.

Il commento del codice è tenuto in considerazione nella valutazione.