

**SCRITTO DI**  
**IUM, IUM+TWEB, Ist. di Tecnologie Web, SWAM, Laboratorio di**  
**Servizi Web, Servizi Web**  
**11 febbraio 2016 - PROFF. ARDISSONO-SACCO**

COGNOME E NOME	N.MATRICOLA
-------------------	-------------

Tempo: 1:40 hr. IUM+TWEB; 40 min. per IUM; 1 hr per SWAM, Ist. Tec. Web, Lab. Servizi Web e Servizi Web  
Non sono ammessi libri di testo e appunti. Utilizzare per le risposte solo i fogli che vi verranno consegnati.

**Si prega di SVILUPPARE GLI ESERCIZI DELLA PARTE 1 E DELLA PARTE 2 IN FOGLI PROTOCOLLO SEPARATI per permettere ai docenti di fare la correzione in parallelo.**

**1-IUM SACCO**

1. (6 punti) Il modello di Shneiderman: definizione, implicazioni e limiti
2. (6 punti) Linee guida: le scorciatoie
3. (3 punti) Che cos'è il daltonismo e quali sono le sue implicazioni per IUM?

Si disegni inoltre la struttura del DOM specificata da questa grammatica.

- **Applicazione Web in ambiente java – parte teorica (4,5 punti):** Si descriva in dettaglio il pattern architetturale MVC per il web, eventualmente con l'aiuto di un disegno, e se ne discutano vantaggi e svantaggi.

**2-TWEB ARDISSONO**

- **XML (5,5 punti):** si definisca un XMLSchema che specifichi documenti XML la cui radice è di tipo **VoloAereo**, caratterizzato dai seguenti tipi di informazione:

- codiceVolo <!-- codice del volo – stringa di caratteri -->
- posti <!-- elenco di coppie <codice posto, passeggero> che specificano i dati dei passeggeri assegnati ai posti a sedere. Se un posto è vacante l'elemento passeggero viene omissis. I passeggeri sono rappresentati da elementi di tipo *PasseggeroType* -->
- partenza <!-- aeroporto di partenza – stringa di caratteri -->
- destinazione <!-- aeroporto di destinazione – stringa di caratteri -->

**PasseggeroType** è caratterizzato dagli elementi:

- nome <!-- nome del passeggero – stringa di caratteri -->
- cognome <!-- cognome del passeggero – stringa di caratteri -->
- frequentFlyer <!-- elemento opzionale che riporta il codice numerico frequent flyer del passeggero, se disponibile -->

- **Applicazione Web in ambiente java – esercizio (5 punti):** Scrivere il codice di una Servlet Java che gestisca richieste POST `http://localhost:8080/info?action=a` come segue:

- Se l'azione richiesta specificata nel parametro "action" è "meteo", invia il controllo alla Servlet "Meteo" che gestisce la richiesta;
- Se l'azione è "exit", visualizza a video la scritta "Arrivederci *utente*" (prendendo il valore di *utente* da sessione utente) e invalida la sessione;
- Per ogni altro valore della richiesta invia il controllo ad una pagina di errore "error.jsp" specificando che il messaggio da visualizzare sul browser utente è "Richiesta non valida".

NB: Non si implementino né Meteo né error.jsp.