**Laboratorio di Basi di dati per Bioinformatica (2009/2010)**

Prova scritta del 22 giugno 2010

**Avvertenze:** e’ severamente vietato consultare libri e appunti; chiunque verrà trovato in possesso di materiale attinente al corso vedrà annullata la propria prova

Durata 1h 40m

# Cognome e nome:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Matricola:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (3) Illustrare le differenze principali tra la tecnologia CGI e la tecnologia servlet.
2. (3) Illustrare le caratteristiche della classe HttpServletRequest (classe dell’oggetto request parametro dei metodi doGet/doPost/Service di una servlet.
3. Estendere il seguente frammento XML, aggiungendo, attraverso opportuni attributi, la specifica che tutti i conti correnti descritti sono del primo cliente. Dare anche il DTD per questa rappresentazione.

<banca>

<conto>

<numero\_conto>101</numero\_conto>

<agenzia> citta’</agenzia>

<saldo> 500 </saldo>

</conto>

<conto>

<numero\_conto>102</numero\_conto>

<agenzia> roncadelle</agenzia>

<saldo> 900 </saldo>

</conto>

<conto>

<numero\_conto>201</numero\_conto>

<agenzia> negrar</agenzia>

<saldo> 1500 </saldo>

</conto>

<cliente>

<nome\_cliente>Bianchi</nome\_cliente>

<citta\_cliente> Verona </citta\_cliente>

</cliente>

<cliente>

<nome\_cliente>Verdi</nome\_cliente>

<citta\_cliente> Garda </citta\_cliente>

<via\_cliente> Garibaldi</via\_cliente>

</cliente>

</banca>

1. Data la base di dati “Multisala” su SQLServer, contenente le seguenti tabelle:

**FILM(Codice, Titolo, Regista, AttoreProtagonista, Anno, Descrizione)**

**PROIEZIONE(CodFilm, Cinema, Sala, Giorno, Ora, PostiDisponibili)**

**SALA(Cinema, Numero, Posti, Edificio, Piano)**

**CINEMA(Nome, Indirizzo, Città)**

Vincoli di integrità referenziale:

PROIEZIONE.CodFilm → FILM,

PROIEZIONE.(Cinema,Sala) → SALA,

SALA.Cinema → CINEMA

* 1. (4) Scrivere il comando SQL per la creazione della tabella PROIEZIONE supponendo già create le tabelle FILM, SALA e CINEMA. Si richiede di precisare la chiave primaria e i vincoli di integrità referenziale specificando la politica cascade in caso di aggiornamento delle tabelle “master”.
  2. (8) Si ipotizzi di dover implementare secondo il modello MVC-2 servlet centric un’applicazione web contenente tra le altre le seguenti pagine:
     1. una **homePage** contenente l’elenco dei cinema che hanno almeno una proiezione in una delle proprie sale nei prossimi 7 giorni. L’elenco va riportato nell’ordine alfabetico del nome includendo: il nome del cinema, la città e il numero di proiezioni di oggi. Il nome del cinema è un link verso lo schema di pagina **cinemaPage**.
     2. nello schema di pagina **cinemaPage** si presentano tutti i dati di un cinema: nome, indirizzo e città. Si elencano inoltre i film in programmazione oggi presso le sale del cinema (vale a dire per i quali è prevista una proiezione oggi) indicando: codice, titolo, regista, attore protagonista del film. Inoltre si mostra la sala, l’ora e i posti disponibili.

Partendo dai requisiti sopra descritti, si completi la servlet Main.java mostrata nei fogli successivi con il codice necessario all’attivazione della JSP che implementa lo schema di pagina **cinemaPage**. Si supponga presente un parametro “ps” che indica la JSP da attivare. Se il parametro è assente viene invocata la JSP che implementa la **homePage**. Vanno scelti e inclusi nella richiesta HTTP i parametri aggiuntivi necessari per le interrogazioni che alimentano gli schemi di pagina.

* 1. (8) Si completi la JSP cinemaPage.jsp che implementa lo schema di pagina **cinemaPage** nel seguente modo:
     1. Indicare come titolo di primo livello il nome del cinema.
     2. seguire su righe separate indicare: l’indirizzo e la città dove in cinema è ubicato.
     3. Riportare poi in una tabella l’elenco dei film in programmazione oggi presso il cinema mostrando nelle colonne della tabella tutti gli attributi indicati nello schema di pagina **cinemaPage** per l’elenco dei film di oggi.

Sono disponibili e non vanno implementati:

1. I Java Data Beans: *FilmBean* (proprietà: *codice*, *titolo*, *registra* *attoreProtagonista*, *sala*, *data*, *ora*) e *CinemaBean* (proprietà: *nome*, *indirizzo*, *città*).
2. La classe DBMS contenente i metodi
   * 1. *public CinemaBean* *getCinema(String nome)* e
     2. *public Vector getFilmOggi(String cinema)*

# Cognome e nome:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Matricola:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**main.java**

import java.io.\*;

import java.util.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import beanMultiSala.\*;

public class main extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws IOException, ServletException {

//Definizione e recupero dell'eventuale parametro ps della servlet

String ps = "";

RequestDispatcher rd = null;

if (request.getParameter("ps") != null) {

**ps =**

}

try {

**// Oggetto per l'interazione con il Database**

if (ps.equals("")) {

**// Parametro ps assente o vuoto, viene attivata la**

**// home page del sito. Non implementare**

} else if

**// Implementare quanto è necessario per l’attivazione**

**// della jsp cinemaPage.jsp**

} catch(Exception e) {

e.printStackTrace();

}

}

}

# Cognome e nome:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Matricola:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**cinemaPage.jsp**

<%@page import="java.io.\*"%>

<%@page import="java.util.\*"%>

<%@page import="beanMultiSala.\*"%>

<html>

<%

//Dichiaro un bean che conterrà i dati del cinema

CinemaBean cinema;

//Dichiaro un bean che conterrà i dati del film

FilmBean film;

**//Recupero il bean passato come attributo contenente i dati del cinema**

**//Recupero il vector passato come attributo contenente la lista dei film di oggi**

**Vector lista =**

%>

<!--Inizio la pagina HTML-->

<head>

<title>Pagina che descrive un cinema e i suoi film di oggi</title>

</head>

<body>

</body>

</html>

# Cognome e nome:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Matricola:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**public class CinemaBean** {

…

} NON IMPLEMENTARE

**public class FilmBean** {

…

} NON IMPLEMENTARE

-------------------------------------------------------------------------------

# DBMS.java

import java.sql.\*;

import java.util.\*;

import beanMultiSala.\*

public class DBMS {

…

**private FilmBean makeFilmBean** (

} NON IMPLEMENTARE

**private CinemaBean makeCinemaBean** (

} NON IMPLEMENTARE

**public Vector getFilmOggi**(…) {…} NON IMPLEMENTARE

**public CinemaBean getCinema** (…) {…} NON IMPLEMENTARE

} // fine classe DBMS.java