

Matricola: _____
Cognome: _____
Nome: _____
Insegnamento: LAB Basi di dati e MM
Basi di dati e MM 4 CFU

Laboratorio di Basi di dati e MM

Basi di dati e MM (4 crediti)

Prova scritta del 13 luglio 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. (4) Illustrare i metodi principali della classe HttpServletRequest (classe dell'oggetto request parametro dei metodi doGet/dopost/service di una servlet).
2. (4) Illustrare le caratteristiche della classe PreparedStatement della libreria JDBC.

Data la base di dati "StabilimentiBalneari" su SQLServer, contenente le seguenti tabelle:

STABILIMENTO(Nome, Indirizzo, Città, Responsabile, Telefono, DataApertura, DataChiusura, Ristorante:{si,no}, OrariRistorante*)

POSTISPIAGGIA(Numero, Stabilimento, Fila, Libero:{si,no})

PERIODO(Nome, DataInizio, DataFine)

PREZZO(Stabilimento, Fila, Periodo, Prezzo)

Vincoli di integrità referenziale:

POSTISPIAGGIA.Stabilimento → STABILIMENTO,

PREZZO.Stabilimento → STABILIMENTO,

PREZZO.Periodo → PERIODO

Per questa parte rispondere sul testo e usare eventualmente il foglio protocollo come brutta copia

3. (5) Scrivere il comando SQL per la creazione della tabella PREZZO supponendo già creata la tabella STABILIMENTO. Si richiede di precisare la chiave primaria e i vincoli di integrità referenziale specificando la politica "cascade" in caso di aggiornamento delle tabelle "master".

Si ipotizzi di dover implementare secondo il modello MVC-2 servlet centric un'applicazione web contenente tra le altre le seguenti pagine:

- a. Nella **homePage** si presenti l'elenco degli stabilimenti aperti alla data corrente riportando: il nome, la città, e il numero di telefono (il nome è un link verso la pagina **stabilimentoPage**).
 - b. Nella pagina **stabilimentoPage** si presentano le seguenti informazioni di uno specifico stabilimento (l'identificatore dello stabilimento verrà passato come parametro nella richiesta HTTP). Tale pagina contiene:
 - Il nome, l'indirizzo, la città, il responsabile, il numero di telefono, la data di apertura e chiusura. Per gli stabilimenti con ristorante si riportano anche gli orari di apertura del medesimo. Si riporta infine il numero totale di posti spiaggia dello stabilimento.
 - L'elenco dei prezzi dei posti spiaggia dello stabilimento: mostrare per ogni periodo il prezzo per ogni fila.
 - L'elenco dei posti spiaggia attualmente liberi: riportando fila e numero.
4. (8) 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 la pagina **stabilimentoPage**. 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 i parametri aggiuntivi (che si ipotizzano presenti nella richiesta HTTP) necessari per le interrogazioni che alimentano la pagina **stabilimentoPage**.
5. (12) Si completi la classe JAVA DBMS contenente i metodi:
- i. `public StabilimentoBean getStabilimento(String nome),`
 - ii. `public Vector getPrezzi(String stabilimento),`
 - iii. `public Vector getPostiLiberi(String stabilimento),`

Sono disponibili e non vanno implementati:

- I Java Data Beans: *StabilimentoBean* (proprietà: *nome, indirizzo, città, responsabile, telefono, dataApertura, data Chiusura, Ristorante, orariRistorante*), *PostoSpiaggiaBean* (proprietà: *numero, stabilimento, fila*) e *PrezzoBean* (proprietà: *periodo, fila, stabilimento, prezzo*).
- La JSP che presenta le informazioni come richiesto dalla specifica della pagina corrispondente. Si richiede solo di scrivere il codice necessario per il recupero dei dati passati alla JSP dalla servlet Main.

DOMANDA OBBLIGATORIA SOLO PER BASI DI DATI E MULTIMEDIA (4 CREDITI)

6. (4) Lo studente illustri la tecnica di compressione delle immagini adottata dalla codifica JPEG.

Cognome e nome: _____

Matricola: _____

main.java

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import beanStabilimentoBalneare.*;

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 stabilimentoPage.jsp

        } catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

Cognome e nome: _____

Matricola: _____

cinemaPage.jsp

```
<%@page import="java.io.*"%>
<%@page import="java.util.*"%>
<%@page import="beanStabilimentoBalneare.*"%>
<html>
<%
//Dichiaro un bean che conterrà i dati dello stabilimento
StabilimentoBean stabilimento;
//Recupero il bean passato come attributo contenente i dati dello stabilimento
stabilimento =

//Recupero il vector passato come attributo contenente la lista dei prezzi
Vector listaPrezzi =

//Recupero il vector passato come attributo contenente la lista dei posti liberi
Vector listaPostiLiberi =

%>

<!--Inizio la pagina HTML-->
<head>
<title>Pagina che descrive un cinema e i suoi film di oggi</title>
</head>
<body>

NON IMPLEMENTARE!

</body>
</html>
```

```
-----

public class StabilimentoBean {
...
} NON IMPLEMENTARE!

public class PrezzoBean {
...
} NON IMPLEMENTARE!

public class PostoSpiaggiaBean {
...
} NON IMPLEMENTARE!
```

Cognome e nome: _____

Matricola: _____

DBMS.java

```
import java.sql.*;
import java.util.*;
import beanStabilimentoBalneare.*

public class DBMS {
    // Dati di identificazione dell'utente

    // URL per la connessione alla base di dati

    // Driver da utilizzare per la connessione JDBC

    // Definizione delle interrogazioni SQL necessarie per i tre metodi:
    // getPrezzi, getStabilimento e getPostiLiberi

    //Costruttore della classe DBMS

    //Metodi per la creazione di un bean a partire dal record attuale
    //del ResultSet dato come parametro
    private PrezzoBean makePrezzoBean (

}

private StabilimentoBean makeStabilimentoBean (...) {...} NON IMPLEMENTARE!
private PostoSpiaggiaBean makePostoSpiaggiaBean (...) {...} NON IMPLEMENTARE!
```

Cognome e nome: _____

Matricola: _____

```
public Vector getPrezzi (
    // Dichiarazione delle variabili necessarie
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    Vector result = new Vector();
    try {
```

```
}
```

```
private StabilimentoBean getStabilimento (
    // Dichiarazione delle variabili necessarie
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    try {
```

```
public Vector getPostiLiberi (...) {...} NON IMPLEMENTARE
```

```
} // classe DBMS.java
```