

ATTENZIONE: nello svolgimento voglio fare un'applicazione web con qualche funzione in più rispetto a quella della consegna (vedi cartella "Preview").

Procedimento:

("Perspective" di Eclipse: Java EE)

- Creo il DB con il terminale.
- Creo la classe "Studente" che mi rappresenta l'oggetto con cui si dovrà lavorare (Matricola, Cognome, Nome, DataNascita).
- Dato che dovrò comunicare con il DB creo l'interfaccia "Modello" implementata dalla classe "DatabaseManager".
- Mi tornerà comoda anche una classe "Sessione" (anche se in questo caso è semplice e si potrebbe farne a meno).
- Essendo un sito, la parte grafica verrà gestita da una pagina "sistema.jsp".

Scrittura codice:

- Faccio la parte grafica senza funzioni.
- Scelgo una delle funzioni che devo fare (inserisci, aggiorna, rimuovi, aggiorna tabella, carica dati, cambia lingua); ad es, scelgo "inserisci".
- Aggiungo il metodo "inserisci" all'interfaccia "Modello" e poi lo implemento in "DatabaseManager".
- Poi creo la servlet "InserisciStudenteServlet".
- Se serve, aggiungo il codice java alla "sistema.jsp".
- Provo e sistemo la funzione dell'applicazione web finché non va.
- Ripeto questi passaggi per tutte le funzioni che ho da implementare.

Gestione lingua (potrebbe non essere gestita nel modo migliore):

- Imposto come attributo della sessione la lingua (se non è stata scelta, metto "Italiano").
- All'inizio del codice della "sistema.jsp", accedo al file "configurazione.properties" dal quale (in base alla lingua) estraggo le stringhe e le uso dove servono.

In questo modo (se gestisco bene l'utilizzo del java in "sistema.jsp") mi basta aggiungere una sola riga:

listaLingue.add("nuova lingua da aggiungere");

E inserire le traduzioni in "configurazione.properties".

P.S.

Non l'ho fatto, ma con procedimento simile a quello usato in "sistema.jsp", si può fare la traduzione anche dei messaggi di errore scritti nelle varie Servlet.

ECLIPSE:

src

```
esame4.database
    DatabaseManager.java
esame4.modello
    Modello.java
    Studente.java
esame4.sessione
    Sessione.java
esame4.web
    AggiungiStudenteServlet.java
    CaricaDatiServlet.java
    LinguaSitoServlet.java
    ListaStudentiServlet.java
    ModificaStudenteServlet.java
    RimuoviStudenteServlet.java
configurazione.properties
WebContent
    WEB-INF
        lib
            derbyclient.jar
        web.xml
    index.html
    errore.html
    sistema.jsp
```

Studente.java

```
package esame4.modello;
```

```
public class Studente {
    public int matricola;
    public String nome;
    public String cognome;
    public String dataNascita;

    public Studente(int matricola, String nome, String cognome, String dataNascita) {
        this.matricola = matricola;
        this.nome = nome;
        this.cognome = cognome;
        this.dataNascita = dataNascita;
    }

    public int getMatricola() {
        return matricola;
    }
    public void setMatricola(int matricola) {
        this.matricola = matricola;
    }
}
```

```

    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getCognome() {
        return cognome;
    }
    public void setCognome(String cognome) {
        this.cognome = cognome;
    }
    public String getDataNascita() {
        return dataNascita;
    }
    public void setDataNascita(String dataNascitaString) {
        this.dataNascita = dataNascitaString;
    }
}

```

Modello.java

```

package esame4.modello;
import java.util.List;

public interface Modello {
    public List<Studente> ricaricaStudenti();
    public boolean modificaStudente(Studente studente);
    public boolean aggiungiStudente(Studente studente);
    public boolean rimuoviStudente(int matricola);
    public Studente caricaDati(int matricola);
}

```

DatabaseManager.java

```

package esame4.database;
import java.sql.Connection;
...
public class DatabaseManager implements Modello{
    protected String url;

    public DatabaseManager() {
        try {
            ResourceBundle resourceBundle = ResourceBundle.getBundle("configurazione");
            String clazz = resourceBundle.getString("database.class");
            url = resourceBundle.getString("database.url");
            Class.forName(clazz);
        } catch (Throwable throwable) {
            throwable.printStackTrace();
        }
    }
}

```

```

@Override
public List<Studiante> ricaricaStudenti() {
    List<Studiante> studenti = new ArrayList<Studiante>();
    Connection connessione = null;
    PreparedStatement statement = null;
    ResultSet resultSet = null;

    try {
        connessione = connetti();
        statement = connessione.prepareStatement("SELECT * FROM STUDENTE");
        resultSet = statement.executeQuery();

        while(resultSet.next()) {
            Studiante studente = new Studiante(resultSet.getInt("MATRICOLA"),
resultSet.getString("NOME"), resultSet.getString("COGNOME"), resultSet.getString("NASCITA"));
            studenti.add(studente);
        }
    } catch(Throwable throwable) {
        throwable.printStackTrace();
        disconnetti(statement, connessione, resultSet);
    }
    disconnetti(statement, connessione, resultSet);
    return studenti;
}

@Override
public boolean modificaStudiante(Studiante studente) {
    Connection connessione = null;
    PreparedStatement statement = null;

    try {
        connessione = connetti();
        statement = connessione.prepareStatement("UPDATE STUDENTE SET COGNOME
= ?, NOME = ?, NASCITA = ? WHERE MATRICOLA = ?");
        statement.setString(1, studente.getCognome());
        statement.setString(2, studente.getNome());
        statement.setString(3, studente.getDataNascita());
        statement.setInt(4, studente.getMatricola());
        statement.execute();

    } catch(Throwable throwable) {
        throwable.printStackTrace();
        disconnetti(statement, connessione, null);
        return false;
    }

    disconnetti(statement, connessione, null);
    return true;
}

```

@Override

```
public boolean rimuoviStudente(int matricola) {
    Connection connessione = null;
    PreparedStatement statement = null;

    try {
        connessione = connetti();
        statement = connessione.prepareStatement("DELETE FROM STUDENTE
                                                WHERE MATRICOLA = ?");

        statement.setInt(1, matricola);
        statement.execute();

    } catch (Throwable throwable) {
        throwable.printStackTrace();
        disconnetti(statement, connessione, null);
        return false;
    }
    disconnetti(statement, connessione, null);
    return true;
}
```

@Override

```
public Studente caricaDati(int matricola) {
    Connection connessione = null;
    PreparedStatement statement = null;
    ResultSet resultSet = null;
    Studente studente = null;

    try {
        connessione = connetti();
        statement = connessione.prepareStatement("SELECT * FROM STUDENTE
                                                WHERE MATRICOLA = ?");

        statement.setInt(1, matricola);
        resultSet = statement.executeQuery();

        if(resultSet.next()) {
            studente = new Studente(matricola, resultSet.getString("NOME"),
                                    resultSet.getString("COGNOME"), resultSet.getString("NASCITA"));
            disconnetti(statement, connessione, resultSet);
            return studente;
        }
        else { // Se l'SQL non ha dato risultati:
            disconnetti(statement, connessione, resultSet);
            return null;
        }
    } catch (Throwable throwable) {
        throwable.printStackTrace();
        disconnetti(statement, connessione, resultSet);
        return null;
    }
}
```

@Override

```
public boolean aggiungiStucente(Studente studente) { //La "matricola" non viene considerata
    Connection connessione = null;
    PreparedStatement statement = null;
    ResultSet resultSet = null;

    try {
        connessione = connetti();
        statement = connessione.prepareStatement("INSERT INTO STUDENTE
(COGNOME,NOME,NASCITA) VALUES (?,?,?)", statement.RETURN_GENERATED_KEYS);
        statement.setString(1, studente.getCognome());
        statement.setString(2, studente.getNome());
        statement.setString(3, studente.getDataNascita());
        statement.execute();

    } catch (Throwable throwable) {
        throwable.printStackTrace();
        disconnetti(statement, connessione, resultSet);
        return false;
    }

    disconnetti(statement, connessione, resultSet);
    return true;
}

public Connection connetti() throws SQLException {
    return DriverManager.getConnection(url);
}

public void disconnetti(Statement statement, Connection connection, ResultSet resultSet) {
    try {
        statement.close();
    } catch (Throwable throwable) {
        //Vuoto
    }
    try {
        connection.close();
    } catch (Throwable throwable) {
        //Vuoto
    }
    try {
        resultSet.close();
    } catch (Throwable throwable) {
        //Vuoto
    }
}
}
```

Sessione.java

```
package esame4.sessione;
import esame4.database.DatabaseManager;

public class Sessione {
    protected DatabaseManager databaseManager;
    public Sessione() {
        databaseManager = new DatabaseManager();
    }
    public DatabaseManager getDatabaseManager() {
        return databaseManager;
    }
}
```

ListaStudentiServlet.java

```
package esame4.web;
import java.io.IOException;
...

@SuppressWarnings("serial")
public class ListaStudentiServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        try {
            HttpSession session = request.getSession();
            Sessione sessione = (Sessione)session.getAttribute("sessione");

            if(sessione==null) {
                sessione = new Sessione();
                session.setAttribute("sessione", sessione);
            }

            DatabaseManager databaseManager = sessione.getDatabaseManager();

            List<Studente> studenti = databaseManager.ricaricaStudenti();
            session.setAttribute("lista_studenti", studenti);

            request.getRequestDispatcher("sistema.jsp").forward(request, response);
        } catch(Throwable throwable) {
            request.getRequestDispatcher("errore.html").forward(request, response);
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        doGet(request, response);
    }
}
```

CaricaDatiServlet.java

```
package esame4.web;
import java.io.IOException;
...

@SuppressWarnings("serial")
public class CaricaDatiServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        try {
            HttpSession session = request.getSession();
            Sessione sessione = (Sessione)session.getAttribute("sessione");

            if(sessione==null){
                sessione = new Sessione();
                session.setAttribute("sessione", sessione);
            }

            DatabaseManager databaseManager = sessione.getDatabaseManager();
            int matricola = Integer.parseInt(request.getParameter("matricola"));
            Studente studente = databaseManager.caricaDati(matricola);
            String messaggio = null;
            if(studente==null)
                messaggio = "ERRORE: Lo studente non non è iscritto!";

            List<Studente> studenti = databaseManager.ricaricaStudenti();

            session.setAttribute("lista_studenti", studenti);
            session.setAttribute("messaggio", messaggio);
            session.setAttribute("dati_studente", studente);
            request.getRequestDispatcher("sistema.jsp").forward(request, response);
        } catch(Throwable throwable) {
            request.getRequestDispatcher("errore.html").forward(request, response);
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}
```


AggiungiStudenteServlet.java

```
package esame4.web;
import java.io.IOException;
...

@SuppressWarnings("serial")
public class AggiungiStudenteServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        try {
            HttpSession session = request.getSession();
            Sessione sessione = (Sessione)session.getAttribute("sessione");

            if(sessione==null) {
                sessione = new Sessione();
                session.setAttribute("sessione", sessione);
            }

            String cognome = request.getParameter("cognome");
            String nome = request.getParameter("nome");
            String dataNascita = request.getParameter("dataNascita");
            dataNascita = dataNascita.replaceAll("-", "");

            DatabaseManager databaseManager = sessione.getDatabaseManager();
            String messaggio = null;

            if(cognome.isEmpty() || nome.isEmpty() || dataNascita.length()!=8) {
                messaggio = "ERRORE: Dati inseriti non correttamente!";
            }
            else {
                dataNascita = dataNascita.substring(4, 8) + dataNascita.substring(2, 4) +
dataNascita.substring(0,2);
                Studente studente = new Studente(0, nome, cognome, dataNascita);

                if(databaseManager.aggiungiStudente(studente))
                    messaggio = "Studente aggiunto correttamente.";
                else
                    messaggio = "ERRORE: Non è stato possibile aggiungere lo studente!";
            }

            List<Studente> studenti = databaseManager.ricaricaStudenti();
            session.setAttribute("lista_studenti", studenti);
            session.setAttribute("messaggio", messaggio);
            request.getRequestDispatcher("sistema.jsp").forward(request, response);

        } catch(Throwable throwable) {
            request.getRequestDispatcher("errore.html").forward(request, response);
        }
    }
}
```

```

        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException {
            doGet(request, response);
        }
    }
}

```

ModificaStudenteServlet.java

```

package esame4.web;
import java.io.IOException;
...

@SuppressWarnings("serial")
public class ModificaStudenteServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {
        try {
            HttpSession session = request.getSession();
            Sessione sessione = (Sessione)session.getAttribute("sessione");

            if(sessione==null) {
                sessione = new Sessione();
                session.setAttribute("sessione", sessione);
            }

            int matricola = Integer.parseInt(request.getParameter("matricola"));
            String cognome = request.getParameter("cognome");
            String nome = request.getParameter("nome");

            String dataNascita = request.getParameter("dataNascita");
            dataNascita = dataNascita.replaceAll("-", "");

            DatabaseManager databaseManager = sessione.getDatabaseManager();
            String messaggio = null;

            if(cognome.isEmpty() || nome.isEmpty() || dataNascita.length()!=8) {
                messaggio = "ERRORE: Dati modificati non correttamente!";
            }
            else {
                dataNascita = dataNascita.substring(4, 8) + dataNascita.substring(2, 4) +
                                                                    dataNascita.substring(0,2);
                Studente studente = new Studente(matricola, nome, cognome, dataNascita);

                if(databaseManager.modificaStudente(studente))
                    messaggio = "Studente modificato correttamente.";
                else
                    messaggio = "ERRORE: Non è stato possibile aggiungere lo studente!";
            }
        }
    }
}

```

```

        List<Studente> studenti = databaseManager.ricaricaStudenti();
        session.setAttribute("lista_studenti", studenti);
        session.setAttribute("messaggio", messaggio);

        request.getRequestDispatcher("sistema.jsp").forward(request, response);

    } catch (Throwable throwable) {
        request.getRequestDispatcher("errore.html").forward(request, response);
    }
}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    doGet(request, response);
}
}

```

RimuoviStudenteServlet.java

```

package esame4.web;
import java.io.IOException;
...

@SuppressWarnings("serial")
public class RimuoviStudenteServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        try {
            HttpSession session = request.getSession();
            Sessione sessione = (Sessione)session.getAttribute("sessione");

            if(sessione==null) {
                sessione = new Sessione();
                session.setAttribute("sessione", sessione);
            }

            int matricola = Integer.parseInt(request.getParameter("matricola"));

            DatabaseManager databaseManager = sessione.getDatabaseManager();
            databaseManager.rimuoviStudente(matricola);

            String messaggio = "Studente rimosso.";
            List<Studente> studenti = databaseManager.ricaricaStudenti();
            session.setAttribute("lista_studenti", studenti);
            session.setAttribute("messaggio", messaggio);

            request.getRequestDispatcher("sistema.jsp").forward(request, response);

```

```

        } catch(Throwable throwable) {
            request.getRequestDispatcher("errore.html").forward(request, response);
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        doGet(request, response);
    }
}

```

LinguaSitoServlet.java

```

package esame4.web;
import java.io.IOException;
...

@SuppressWarnings("serial")
public class LinguaSitoServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        HttpSession session = request.getSession();
        Sessione sessione = (Sessione)session.getAttribute("sessione");

        if(sessione==null) {
            sessione = new Sessione();
            session.setAttribute("sessione", sessione);
        }

        String lingua = request.getParameter("lingua");
        session.setAttribute("lingua_sito", lingua);

        DatabaseManager databaseManager = sessione.getDatabaseManager();
        List<Studente> studenti = databaseManager.ricaricaStudenti();
        session.setAttribute("lista_studenti", studenti);

        request.getRequestDispatcher("sistema.jsp").forward(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        doGet(request, response);
    }
}

```

sistema.jsp

```
<%@page import="java.util.ArrayList"%>
<%@page import="esame4.modelo.Studente"%>
<%@page import="esame4.database.DatabaseManager"%>
<%@page import="esame4.session.Sessione"%>
<%@page import="java.util.List"%>
<%@page import="java.util.ResourceBundle"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Sistema</title>
</head>
<%
    // LINGUA
    ResourceBundle resourceBundle = ResourceBundle.getBundle("configurazione");

    String lingua = (String)session.getAttribute("lingua_sito");
    if(lingua==null)
        lingua = "Italiano";

    String ConfLing = "Lingua." + lingua + ".";

    List<String> listaLingue = new ArrayList();
    listaLingue.add("Italiano");
    listaLingue.add("English");
    listaLingue.add("Klingon");
    //listaLingue.add("altra nuova lingua da aggiungere");

    // TABELLA STUDENTI
    Studente datiStudente = (Studente)session.getAttribute("dati_studente");
    String dataNascita = "";

    if(datiStudente==null) {
        datiStudente = new Studente(0, "", "", "");
    }
    else {
        dataNascita = datiStudente.getDataNascita();
        dataNascita = dataNascita.substring(6, 8) + "-" + dataNascita.substring(4, 6) + "-" +
                                                                dataNascita.substring(0, 4);
    }
    session.removeAttribute("dati_studente");
%>

<body>
<h2><%=resourceBundle.getString(ConfLing + "TitoloSito") %></h2>
```

```
<form method="post" action="">
<p><span id="matricolaS"><%=resourceBundle.getString(ConfLing + "LblMatricolaTxt") %></span></p>
<p><input type="text" name="matricola" id="matricola" value="<%= datiStudiante.getMatricola() %>"
readonly/></p>
<p><span id="cognomeS"><%=resourceBundle.getString(ConfLing + "LblCognomeTxt") %></span></p>
<p><input type="text" name="cognome" id="cognome" value="<%= datiStudiante.getCognome() %>" /></p>
<p><span id="nomeS"><%=resourceBundle.getString(ConfLing + "LblNomeTxt") %></span></p>
<p><input type="text" name="nome" id="nome" value="<%= datiStudiante.getNome() %>" /></p>
<p><span id="dataNascitaS"><%=resourceBundle.getString(ConfLing + "LblNascitaTxt") %></span></p>
<p><input type="text" name="dataNascita" id="dataNascita" value="<%= dataNascita %>" /></p>
<p>
<table border="1">
<tr>
<td>
<input type="submit" value="<%=resourceBundle.getString(ConfLing + "BtnRimuovi") %>"
onclick="form.action='rimuovi_studiante'/">
<input type="submit" value="<%=resourceBundle.getString(ConfLing + "BtnModifica") %>"
onclick="form.action='modifica_studiante'/">
<input type="submit" value="<%=resourceBundle.getString(ConfLing + "BtnAggiungi") %>"
onclick="form.action='aggiungi_studiante'/">
</td>
<td>
<a href="lista_studenti"><input type="button" value="<%=resourceBundle.getString(ConfLing +
"BtnAggiornaTabella") %>" /></a>
</td>
<td>
<input type="submit" value="<%=resourceBundle.getString(ConfLing + "BtnApplicaLingua") %>"
onclick="form.action='lingua_sito'/">
<select name="lingua">
<%
// LINGUA
for(int i=0; i<listaLingue.size();i++) {
%>
<option value="<%= listaLingue.get(i)%>"
%>
if(listaLingue.get(i).equals(lingua)) {
selected
%>
}
%>
%> ><%= listaLingue.get(i)%></option>
%>
}
%>
</select>
</td>
</tr>
</table> </p>
</form>
```

```

<%
    // MESSAGGIO (di errore)
    String msg = (String)session.getAttribute("messaggio");
    session.removeAttribute("messaggio");
    if(msg==null)
        msg="";
%>
<p><font size="2"><%=msg %></font></p>
<br>
<span id="studentiInscrittiS"><%=resourceBundle.getString(ConfLing + "LblStudentiInscritti") %></span>
<table id="tabellaStudenti" border="1">
<tr>
    <td><%=resourceBundle.getString(ConfLing + "LblMatricolaT") %></td>
    <td><%=resourceBundle.getString(ConfLing + "LblCognomeT") %></td>
    <td><%=resourceBundle.getString(ConfLing + "LblNomeT") %></td>
</tr>
<%
    // TABELLA STUDENTI
    List<Studente> studenti = (List<Studente>)session.getAttribute("lista_studenti");
    session.removeAttribute("lista_studenti");

    if(studenti!=null) {
        for(Studente studente : studenti) {
%>
<tr>
<td> <a href="carica_dati?matricola=<%=studente.getMatricola()%>"><%= studente.getMatricola() %></a>
</td>
<td> <a href="carica_dati?matricola=<%=studente.getMatricola()%>"><%= studente.getCognome() %></a>
</td>
<td> <a href="carica_dati?matricola=<%=studente.getMatricola()%>"><%= studente.getNome() %></a>
</td>
</tr>
<%
        }
    }
%>
</table>
</body> </html>

```

index.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<a href="lista_studenti">Sistema di gestione studenti</a>
</body>
</html>

```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
ERRORE
</body>
</html>
```