

Prendo l'es. 5 e aggiungo javascript.

La servlet passa i dati alla jsp ma l'html resta statica.

A noi interessano anche pagine dinamiche e per fare ciò si usano script; metto java dentro le pagine html che però non è la stessa cosa fatta nelle jsp.

Questo meccanismo si chiama applet (applicazioni dentro html).

Uso javascript che però è poco portabile da un browser all'altro (non è un linguaggio basato su classi quindi non è il massimo anche se si può fare programmazione ad oggetti).

Posso creare dei null che di per se sono uguali fra loro ma in java posso appiccicarli dei metodi per costruire oggetti senza classe.

La soluzione migliore è scrivere in java e poi generare (con un tool di google) il javascript.

In più il codice javascript generato in questo modo è molto crossbrowser (e ottimizzato).

ECLIPSE

Esercizio6

src

```
it.unipr.informatica.esercizio6
    Sessione.java
it.unipr.informatica.esercizio6.database
    DatabaseManager.java
it.unipr.informatica.esercizio6.modello
    Modello.java
    ModelloException.java
    Studente.java
it.unipr.informatica.esercizio6.web
    ListaStudentiServlet.java
    // Servlet per modifiche su DB:
    AggiornaStudenteServlet.java
    DettagliStudenteServlet.java
    InserisciStudenteServlet.java
    ModificaStudenteServlet.java
    RimuoviStudenteServlet.java
configurazione.properties
```

WebContent

```
WEB-INF
    lib
        derbyclient.jar // file
    web.xml
index.html
errore.html
lista_studenti.jsp
// Per modifiche su DB:
aggiungi_studente.jsp
modifica_studente.jsp
dettagli_studente.jsp
```

aggiungi_studente.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Aggiungi Studente</title>
// Vogliamo cambiare lo stato di abilitazione del pulsante submit in modo dinamico:
<script type="text/javascript"> // tag per caricare lo script prima che la pagina venga caricata; da mettere nel
HEAD della pagina
// Posso scrivere html, javascript e anche java (con i tag)
// Con l'src in script si può collegare un file esterno di java, ma meglio evitare.
// Bisogna tenere disabilitato il bottone anche se si inserissero solo spazi nel nome (trim() è un metodo
recente e quindi nel browser di eclipse non c'è; quindi mi posso definire io una funzione trim() )
function trim(s) {
    return s.replace(/^\s+|\s+$/gm, ""); // Metodo per rimpiazzare stringhe; fuori dagli slash dico cosa
fare cioè in questo caso sostituisco alla lettera tra slash, "")
    /a+/ // per a ripetuta
    /a*/ // anche per nessuna a&&
    /\s/ // indica tutti quei caratteri bianchi (spazi vuoti)
    /^\s+/ // almeno uno spazio in testa
    /\s+ | \s+$/ // indica gli spazi in coda e in testa
}

function verificaCampo() {
// Si ricorda che la pagina html è come un albero e a noi interessa avere un identificativo univoco per una
foglia usando l'id. Non devono esserci due id uguali.
    var submit = document.getElementById("submit"); // getElement... metodo di document
    var nodo = document.getElementById("cognome"); // Il . fa pensare che document sia una classe.
var x = document
x. // solo i compilatori moderni mostrano dopo il . la lista dei metodi di document
    var testo = nodo.value; // Value è un attributo
    var errore = document.getElementById("errore_cognome")
    var ok = true;

    if(trim(testo) == "") {
        errore.innerHTML = "Il cognome non può essere vuoto";
        submit.disabled = "disabled";
        ok = false;
    } else
        errore.innerHTML = "";

    nodo = document.getElementById("nome");
    testo = nodo.value;
    errore = document.getElementById("errore_nome");

    if(trim(testo) == "") {
        errore.innerHTML = "Il nome non può essere vuoto";
        submit.disabled = "disabled"; // Impostando un attributo che non è presente, automaticamente
si aggiunge tale attributo
```

```

        ok = false;
    } else
        errore.innerHTML = "";
    if(ok)
        submit.removeAttribute("disabled"); // Questo mi permette di modificare l'albero (permette
        anche di togliere testo)
    }
</script>
</head>
<body>
<form action="inserisci_studente" method="post">
<p>Cognome: <input id="cognome" name="cognome" type="text" value="" onchange="verificaCampo()"
/> <span id="errore_cognome"></span></p>
<p>Nome: <input id="nome" name="nome" type="text" value="" onchange="verificaCampo()" /> <span
id="errore_nome"></span></p> // onchange = quando submit cambia, applico la funzione indicata
<p><input id="submit" type="submit" value="Aggiungi Studente" disabled="disabled" /></p>
</form>
</body> </html>
// Graficamente:
// Se cancello la scritta in nome senza spostare il focus, il tasto di invio resta attivo ma se io lo premessi,
PRIMA cambierebbe il focus e il bottone si bloccherebbe, POI avverrebbe il click che però sarebbe inutile su
un bottone disattivato.
// Potrei fare il cambiamento (uso del javascript) ogni volta che premo un tasto così il pulsante di invio
cambia anche se il focus non cambia.

```

dettagli_studente.jsp

pagina (vista) che gestisce le operazioni sul DB.

```

<%@ page import="it.unipr.informatica.esercizio6.Sessione"%>
<%@ page import="it.unipr.informatica.esercizio6.modelo.Studente"%>
<%@ page import="java.util.List"%>
<%@ page import="it.unipr.informatica.esercizio6.database.DatabaseManager"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Dettagli Studente</title>
</head>
<body>
<%
    Studente studente = (Studente)session.getAttribute("studente");
    session.removeAttribute("studente");
%>
<p>Matricola: <%=studente.getMatricola() %></p>
<p>Cognome: <%=studente.getCognome() %></p>
<p>Nome: <%=studente.getNome() %></p>
<p>Torna a <a href="lista_studenti">lista studenti</a></p>
</body>
</html>

```

modifica_studente.jsp

```
<%@ page import="it.unipr.informatica.esercizio6.modello.Studente"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Modifica Studente</title>
</head>
<body>
<%
    Studente studente = (Studente)session.getAttribute("studente");
%>
<form action="aggiorna_studente" method="post">
<p>Matricola: <input name="matricola" type="text" value="<%=studente.getMatricola() %>"
readonly="readonly" /></p>
<p>Cognome: <input name="cognome" type="text" value="<%=studente.getCognome() %>" /></p>
<p>Nome: <input name="nome" type="text" value="<%=studente.getNome() %>" /></p>
<p><input type="submit" value="Modifica Studente" /></p>
</form>
</body>
</html>
```

lista_studenti.jsp

```
<%@ page import="it.unipr.informatica.esercizio6.Sessione"%>
<%@ page import="it.unipr.informatica.esercizio6.modello.Studente"%>
<%@ page import="java.util.List"%>
<%@ page import="it.unipr.informatica.esercizio6.database.DatabaseManager"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Lista degli Studenti</title>
<script type="text/javascript">
function rimuovi(matricola) {
    var sicuro = confirm("Sei sicuro di voler cancellare lo studente con matricola: " + matricola);
    - alert(matricola); // Mostrerà una finestra di allerta con un numero
    - sicuro = confirm( "Sei sicuro..." + matricola ); // Altro metodo che richiede conferma
    if(sicuro == true)
        location.href = "rimuovi_studente?matricola="+matricola; // Carica la pagina indicata
}
</script>
</head>
<body>
```

```

<%
    List<Studente> studenti = (List<Studente>)session.getAttribute("lista_studenti");
    session.removeAttribute("lista_studenti");
    String messaggio = (String)session.getAttribute("messaggio");

    if(messaggio != null) {
        session.removeAttribute("messaggio");
    %>
    <p><%=messaggio %></p>
    <%    }    %>
    <p><a href="aggiungi_studente.jsp">Aggiungi studente</a></p>
    <table>
    <tr><td></td><td></td><td></td><td>Matricola</td><td>Cognome</td><td>Nome</td></tr>
    <%
        for(Studente studente : studenti) {
            %>
            <tr>
            <td><a href="dettagli_studente?matricola=<%=studente.getMatricola() %>">Dettagli</a></td>
            <td><a href="modifica_studente?matricola=<%=studente.getMatricola() %>">Modifica</a></td>
            <td><a href="javascript:rimuovi(<%=studente.getMatricola() %>)">Rimuovi</a></td>
            <td><a href="dettagli_studente?matricola=<%=studente.getMatricola() %>"><%=studente.getMatricola()
            %></a></td>
            <td><a href="dettagli_studente?matricola=<%=studente.getMatricola() %>"><%=studente.getCognome()
            %></a></td>
            <td><a href="dettagli_studente?matricola=<%=studente.getMatricola() %>"><%=studente.getNome()
            %></a></td>
            </tr>
            <%    }    %>
        </table>
    </body> </html>

```

InserisciStudenteServlet.java

```

package it.unipr.informatica.esercizio6.web;
import it.unipr.informatica.esercizio6.Sessione;
...

@SuppressWarnings("serial")
public class InserisciStudenteServlet 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");

DatabaseManager databaseManager = sessione.getDatabaseManager();
databaseManager.aggiungiStudiante(cognome, nome);
request.getRequestDispatcher("lista_studenti").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);
}
}

```

AggiornaStudianteServlet.java

```

package it.unipr.informatica.esercizio6.web;
import it.unipr.informatica.esercizio6.Sessione;
...

@SuppressWarnings("serial")
public class AggiornaStudianteServlet 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 matricola = request.getParameter("matricola");
            int m = Integer.parseInt(matricola);

            String cognome = request.getParameter("cognome");
            String nome = request.getParameter("nome");

            DatabaseManager databaseManager = sessione.getDatabaseManager();
            Studente studente = new Studente(m, cognome, nome);
            databaseManager.aggiornaStudiante(studente);
            request.getRequestDispatcher("lista_studenti").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);
    }
}

```

DettagliStudenteServlet.java

```

package it.unipr.informatica.esercizio6.web;
import it.unipr.informatica.esercizio6.Sessione;
...

```

```

@SuppressWarnings("serial")
public class DettagliStudenteServlet 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 matricola = request.getParameter("matricola");
            int m = Integer.parseInt(matricola);

            DatabaseManager databaseManager = sessione.getDatabaseManager();
            Studente studente = databaseManager.leggiStudente(m);
            if(studente == null) {
                session.setAttribute("messaggio", "Lo studente " + matricola + " non è più nel database");
                request.getRequestDispatcher("lista_studenti").forward(request, response);
            } else {
                session.setAttribute("studente", studente);
                request.getRequestDispatcher("dettagli_studente.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 it.unipr.informatica.esercizio6.web;
import it.unipr.informatica.esercizio6.Sessione;
...

@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);
        }

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

        DatabaseManager databaseManager = sessione.getDatabaseManager();
        Studente studente = databaseManager.leggiStudente(m);
        if(studente == null) {
            session.setAttribute("messaggio", "Lo studente " + matricola + " non è più nel database");
            request.getRequestDispatcher("lista_studenti").forward(request, response);
        } else {
            session.setAttribute("studente", studente);
            request.getRequestDispatcher("modifica_studente.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 it.unipr.informatica.esercizio6.web;
import it.unipr.informatica.esercizio6.Sessione;
...

@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);
    }

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

    DatabaseManager databaseManager = sessione.getDatabaseManager();
    databaseManager.rimuoviStudente(m);
    request.getRequestDispatcher("lista_studenti").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);
}
}

```