

**Programmazione Web (Proff. Costagliola - Fuccella) Esercitazione del 31/05/2017**

È vietato utilizzare appunti, libri e dispositivi elettronici. Il vostro banco deve contenere solo i fogli utilizzati per sviluppare il compito e la traccia pena l'esclusione. Tutto il rimanente (borse con telefonini, smartwatch ed altro) deve essere posto ai piedi della cattedra o poggiato ad una parete laterale. **Non è ammesso comunicare con gli altri studenti in nessun caso, pena l'esclusione dalla prova.** Il materiale di esame consiste in:

- un documento con le 20 domande (le risposte non vanno marcate qui). **Verificare che la stampa della traccia contenga tutte le domande.**

- il foglio delle risposte per le domande a risposta multipla su cui marcare le proprie risposte **annerendo accuratamente con una penna nera o blu** i quadratini relativi. Poiché la correzione sarà automatica, evitare correzioni su questo foglio. Non è ammesso l'uso di bianchetti (contattare un esaminatore per effettuare una correzione).

Per le domande a risposta multipla ogni risposta corretta vale 1 punto, senza risposta vale 0 punti, mentre quelle sbagliate valgono -0.25. **VANNO CONSEGNATI SOLO QUESTI DUE DOCUMENTI.**

---

**Domanda 1** Quale dei seguenti tipi consente di accedere ai dati recuperati da un database dopo l'esecuzione di una query SQL?

- ☐ A Driver
- ☒ B ResultSet
- ☐ C PreparedStatement
- ☐ D Nessuna delle altre opzioni

**Domanda 2** Quale dei seguenti tipi consente di effettuare una query parametrica?

- ☒ A PreparedStatement
- ☐ B ResultSet
- ☐ C Driver
- ☐ D Nessuna delle altre opzioni

**Domanda 3** La directory WEB-INF (e le sue eventuali sottodirectory) deve contenere:

- ☐ A Nessuna delle altre opzioni
- ☒ B Le librerie utilizzate dall'applicazione web
- ☐ C Il codice sorgente delle servlet
- ☐ D Tutte le JSP

**Domanda 4** Le scriptlet in JSP:

- ☒ A Includono codice java
- ☐ B Eseguono delle azioni specifiche
- ☐ C Includono delle direttive specifiche
- ☐ D Eseguono codice javascript

**Domanda 5** Quando si modifica una jsp su Tomcat, il modo più rapido per ottenere l'aggiornamento della corrispondente servlet è:

- ☐ A Effettuare il redeploy della jsp
- ☐ B Riavviare il server
- ☐ C Compilare la jsp e quindi la servlet
- ☒ D Effettuare una nuova chiamata alla jsp modificata

**Domanda 6** Una servlet è:

- ☐ A Una libreria java di un web server
- ☒ B una classe java che si interfaccia con HTTP
- ☐ C Un qualsiasi programma javascript
- ☐ D Un programma javascript che si interfaccia con HTTP

**Domanda 7** Dati i due seguenti pezzi di codice, qual è l'output atteso?

Servlet chiamante:

```
..  
public void doGet(HttpServletRequest req, HttpServletResponse res)  
    throws ServletException, IOException {  
    res.setContentType("text/html");  
    PrintWriter out = res.getWriter();  
    out.println("<HTML>");  
    out.println("<BODY>Will you see the source?");  
    out.println("</BODY></HTML>");  
    RequestDispatcher disp=req.getRequestDispatcher("/Test");  
    disp.forward(req, res);  
    out.close();  
}  
..
```

Servlet chiamata ("/Test"):

```
..  
protected void doGet(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
    response.setContentType("text/html");  
    PrintWriter out = response.getWriter();  
    out.println("<HTML><TITLE>The Test</TITLE>");  
    out.println("<BODY>Will you see the target?");  
    out.println("</BODY></HTML>");  
    out.close();  
}  
..
```

- ☐ A Non appare niente; il thread si arresta.
- ☐ B Appaiono sia "Will you see the source?" che "Will you see the target?".
- ☒ C "Will you see the target?"
- ☐ D "Will you see the source?"
- ☐ E Viene lanciata una IllegalStateException.

**Domanda 8** Secondo la specifica semantica del tag HTML <form> quale tra i seguenti metodi prevede l'accodamento dei campi e dei valori del form all'URL di destinazione?

- ☐ A Nessuno dei tre
- ☐ B PUT
- ☐ C POST
- ☒ D GET

**Domanda 9** Data la seguente servlet il cui deploy è stato fatto su un servlet container su “localhost” con porta “8080” e che è compresa nel package “servlet” dell’applicazione web “esercitazione2017”:

```
package servlet;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

@WebServlet("/ser_chiamante")
public class Chiamante extends HttpServlet {
    protected void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        out.println("bye");
        RequestDispatcher disp=req.getRequestDispatcher("/Test");
        disp.forward(req, res);
        out.close();
    }
}
```

qual è l’URL da digitare per poterla invocare tramite client browser?

- ☒ **A** http://localhost:8080/esercitazione2017/ser\_chiamante
- ☐ **B** http://localhost:8080/esercitazione2017/Chiamante
- ☐ **C** http://localhost:8080/esercitazione2017/servlet/ser\_chiamante
- ☐ **D** http://localhost/esercitazione2017/ser\_chiamante

**Domanda 10** Quale sarà l’output risultante dall’esecuzione del seguente pezzo di codice JSP:

```
<% if(true) { %>
To be
<% } else { %>
or not to be
<% } %> ?
```

- ☐ **A** or not to be
- ☐ **B** Il codice non verrà compilato.
- ☐ **C** To be, or not to be
- ☒ **D** To be

**Domanda 11** Quali sono i livelli di “scope” accessibili da una JSP ?

- ☐ **A** application, page, cookie, context
- ☒ **B** page, request, session, application
- ☐ **C** page, response, servletContext, session
- ☐ **D** page, cookie, request, session

**Domanda 12** In quale directory sono messi i file JAR?

- ☐ **A** Direttamente nella directory dell’applicazione
- ☐ **B** Direttamente all’interno della directory /WEB-INF/classes
- ☐ **C** Direttamente all’interno della directory /WEB-INF
- ☒ **D** Direttamente all’interno della directory /WEB-INF/lib

**Domanda 13** Dato il seguente codice:

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <TITLE>Esame</TITLE>
  </HEAD>
  <BODY>
    <H1>Esame</H1>
    <%! private int x = 0; %>
    <H2>Risultato: <%= ++x %> </H2>
  </BODY>
</HTML>
```

se Franco si collega alla risultante servlet (appena istallata) prima di Giovanni e dopo Michele quale valore di x vedranno i tre amici?

- ☐ A Franco vede 3, Giovanni vede 0, Michele vede 1
- ☐ B Franco vede 0, Giovanni vede 1, Michele vede 2
- ☒ C Franco vede 2, Giovanni vede 3, Michele vede 1
- ☐ D Franco vede 1, Giovanni vede 2, Michele vede 0

**Domanda 14** Tramite codice JSP è possibile popolare automaticamente le proprietà di un bean con i parametri input del client

- ☒ A Vero
- ☐ B Falso

## CORREZIONE

**Domanda 15** (punteggio: da 0 a 4) Seguendo l'architettura MVC, scrivere un'applicazione web che contenga una servlet che generi un numero casuale, usando `Math.random()`, lo metta in un bean, e lo invii ("forward") ad una pagina jsp che lo visualizzi. Si usi la condivisione di bean basata su "request". Si scrivano le parti significative di tutti i file necessari e si indichi in quali cartelle questi sono disposti dopo il deployment dell'applicazione.

Nome applicazione web: `esercizioPW`

```
***** Servlet *****
apache-tomcat/webapps/esercizioPW/WEB-INF/classes/servlets/RandGen.class

package servlets;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import beans.RandBean;

/**
 * Servlet implementation class RandGen
 */
@WebServlet("/RandGen")
public class RandGen extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public RandGen() {
        super();
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.getWriter().append("Served at: ").append(request.getContextPath());
        double rand = Math.random();
        RandBean rb = new RandBean();
        rb.setRand(rand);
        request.setAttribute("randBean", rb);
        request.getRequestDispatcher("showrand.jsp").forward(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        doGet(request, response);
    }
}

***** Bean *****
apache-tomcat/webapps/esercizioPW/WEB-INF/classes/beans/RandBean.class

package beans;

public class RandBean {
    double rand;

    public double getRand() {
        return rand;
    }

    public void setRand(double rand) {
        this.rand = rand;
    }
}

***** JSP *****
apache-tomcat/webapps/esercizioPW/showrand.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<jsp:useBean id="randBean" scope="request" class="beans.RandBean" />
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Random Generator</title>
</head>
<body>
    Il numero casuale &grave; <jsp:getProperty name="randBean" property="rand"/>
</body>
</html>
```

## CORREZIONE

**Domanda 16** (punteggio: da 0 a 4) Scrivere una jsp che riceva un indirizzo di posta elettronica da ciascun client che vi accede e visualizzi gli indirizzi degli ultimi due accessi incluso quello appena ricevuto.

```
<html>
<head>
<title>test.jsp</title>
</head>
<body>
<%
String uEmail = request.getParameter("email");
if(uEmail != null) {
    String aEmail = (String)application.getAttribute("LAST_EMAIL");
    application.setAttribute("LAST_EMAIL", uEmail);
}%>
    Ultima email: <%= uEmail %><br>
    Penultima email: <%= aEmail %>
<% } else { %>
    <form action="test.jsp">
        Inserisci email:<br>
        <input type="text" name="email" value=""><br>
        <input type="submit" value="Invia">
    </form>
<% } %>
</body>
</html>
```

**Domanda 17** (punteggio: da 0 a 4) Scrivere una funzione Javascript per verificare che un campo di testo con id="username" abbia lunghezza almeno 8 caratteri e contenga solo numeri.

```
function verify() {
    username = document.getElementById("username").value;
    var reg = new RegExp('^([0-9]{8}[0-9]*$)');
    if (reg.test(username))
        return true;
    else
        return false;
}
```

## CORREZIONE

**Domanda 18** (punteggio: da 0 a 4) Scrivere una servlet che controlli la correttezza di immissione di una login ed una password provenienti da un form nella pagina "login.jsp". In particolare la login non deve essere vuota e la password deve avere almeno 6 caratteri. Nel caso in cui queste condizioni siano verificate, verrà effettuato un forward alla pagina "benvenuto.jsp". In caso di inserimento scorretto verrà effettuato un forward alla stessa pagina con i dati inseriti precedentemente ed un messaggio che descrive l'errore presente nei dati.

```
package testPackage;

import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class Accedi
 */
@WebServlet("/Accedi")
public class Accedi extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Accedi() {
        super();
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        doPost(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        String login = request.getParameter("login");
        String password = request.getParameter("password");
        String errore = "";

        if (login == null)
            errore = "login nulla";
        if (login.length() == 0)
            errore = "login vuota";
        if (password.length() <= 6)
            errore = "password corta";

        if (errore.length() == 0){
            RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/benvenuto.jsp");
            dispatcher.forward(request,response);
        }else{
            RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/login.jsp?errore="+errore);
            dispatcher.forward(request,response);
        }
    }
}
```

**Domanda 19** (punteggio: da 0 a 2) Aggiungere del codice Javascript in modo che il paragrafo mostri la scritta "Hello world!" ed abbia taglia del font di 40 pixel.

```
<!DOCTYPE html>
<html>
  <body>
    <p id="demo">Hi.</p>
    <script>
      document.getElementById("demo").innerHTML = "Hello World!";
      document.getElementById("demo").style.fontSize = "40px";
    </script>
  </body>
</html>
```

**Domanda 20** (punteggio: da 0 a 2) Completare il seguente codice inserendo uno script per la validazione dell'input del campo di testo. Il testo del paragrafo demo deve essere impostato a "Input OK!" nel caso in cui sia un numero tra 1 e 10, a "Input non valido!" altrimenti:

```
<!DOCTYPE html>
<html>
  <body>
    <p>Please input a number between 1 and 10:</p>
    <input id="numb">
    <button type="button" onclick="myFunction()">Submit</button>
    <p id="demo"></p>
    <script>
      function myFunction() {
        numb = parseInt(document.getElementById("numb").value);
        if (numb>=1 && numb<=10)
          document.getElementById("demo").innerHTML = "input OK";
        else
          document.getElementById("demo").innerHTML = "input non valido";
      }
    </script>
  </body>
</html>
```



# CORREZIONE

Note

.....

Domanda 1: ☐ A ☒ B ☐ C ☐ D

Domanda 2: ☒ A ☐ B ☐ C ☐ D

Domanda 3: ☐ A ☒ B ☐ C ☐ D

Domanda 4: ☒ A ☐ B ☐ C ☐ D

Domanda 5: ☐ A ☐ B ☐ C ☒ D

Domanda 6: ☐ A ☒ B ☐ C ☐ D

Domanda 7: ☐ A ☐ B ☒ C ☐ D ☐ E

Domanda 8: ☐ A ☐ B ☐ C ☒ D

Domanda 9: ☒ A ☐ B ☐ C ☐ D

Domanda 10: ☐ A ☐ B ☐ C ☒ D

Domanda 11: ☐ A ☒ B ☐ C ☐ D

Domanda 12: ☐ A ☐ B ☐ C ☒ D

Domanda 13: ☐ A ☐ B ☒ C ☐ D

Domanda 14: ☒ A ☐ B

Domanda 15: ..... [MVC] ☐ 0 ☐ 05 ☐ 1 ☐ 15 ☐ 2 ☐ 25 ☐ 3 ☐ 35 ☒ Non marcare questi box

Domanda 16: ..... [JSP indirizzo posta] ☐ 0 ☐ 05 ☐ 1 ☐ 15 ☐ 2 ☐ 25 ☐ 3 ☐ 35 ☒ Non marcare questi box

Domanda 17: ..... [JS verifica testo] ☐ 0 ☐ 05 ☐ 1 ☐ 15 ☐ 2 ☐ 25 ☐ 3 ☐ 35 ☒ Non marcare questi box

Domanda 18: ..... [Servlet login] ☐ 0 ☐ 05 ☐ 1 ☐ 15 ☐ 2 ☐ 25 ☐ 3 ☐ 35 ☒ Non marcare questi box

Domanda 19: ..... [JS Hello world (40 pixel)] ☐ 0 ☐ 05 ☐ 1 ☐ 15 ☒ Non marcare questi box

Domanda 20: ..... [Validazione input] ☐ 0 ☐ 05 ☐ 1 ☐ 15 ☒ Non marcare questi box

CORREZIONE