

Programmazione Web (Proff. Costagliola - Fuccella)**11 Luglio 2016****Studente: 0**

È vietato utilizzare appunti o libri. Il vostro banco deve contenere solo i fogli utilizzati per sviluppare il compito e la traccia pena l'esclusione. Tutto il rimanente (borse, telefonini, smartwatch ed altro) deve essere posto ai piedi della cattedra o poggiato ad una parete laterale. Il materiale di esame consiste in:

1. un documento con le 26 domande di cui una extra (le risposte non vanno marcate qui). **Verificare che la stampa della traccia contenga tutte le domande.**
2. 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).

Le domande con il simbolo ♣ prevedono una o più risposte da marcare. Le altre domande prevedono una sola risposta da marcare.

Per le domande a risposta multipla ogni risposta corretta vale 1 punto, senza risposta vale 0 punti, mentre quelle sbagliate valgono -0.5. VANNO CONSEGNATI SOLO QUESTI DUE DOCUMENTI.

Domanda 1 [Q001] Quando si modifica una jsp su Tomcat, il modo più semplice 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 2 [Q002] Una servlet è:

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

Domanda 3 [Q003] Quale tra questi tag HTML sono indicati per implementare la selezione di un set di proprietà "non mutuamente esclusive"?

- ☐ A <select>
- ☒ B <input type="checkbox">
- ☐ C <input type="radio">
- ☐ D <select type="range">

Domanda 4 [Q004] 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 POST
- ☒ B GET
- ☐ C PUT
- ☐ D Nessuno dei tre

Domanda 5 [Q005] Dove sono memorizzate le informazioni dei cookie?

- ☐ A Nell'hard-disk del computer remoto.
- ☒ B Nell'hard-disk del computer locale.
- ☐ C Dipende dal tipo di cookie.
- ☐ D Nella Random Access Memory del computer locale

Domanda 6 [Q006] In una tabella HTML quale attributo del tag <td> consente di "espandere" una cella sulle successive celle adiacenti? (per cella si intende un elemento <td /> "table data")

- ☐ A width
- ☐ B maxlength
- ☒ C colspan
- ☐ D rowspan

Domanda 7 [Q007] Dato il codice

```
<head>
<style>
  p { background-color: yellow; color: red;    }
  .foo { color: yellow;}
  p.bar { color: yellow; background-color: transparent; }
  .foo2 { background-color: red; }
</style>
</head>

<body class="foo2">
  1. <p>BRAVO CHI LEGGE</p>
  2. <p class="foo">BRAVO CHI LEGGE</p>
  3. <p class="bar">BRAVO CHI LEGGE</p>
</body>
```

Quale statement rende non leggibile a schermo il testo BRAVO CHI LEGGE?

- ☐ A Linea 1.
- ☒ B Linea 2.
- ☐ C Linea 3.
- ☐ D Nessuna delle tre

Domanda 8 [Q008] Dato il codice

```
<!DOCTYPE html>
<html>
<script>
  function init(obj){
    document.getElementById("t3") =
      document.getElementById("t1") +
      String.toUpperCase(document.getElementById("t2")) +
      document.getElementById("t3") + obj
  }
</script>

<body onload="init('ver1.5');">
  <input type="hidden" id="t1" name="t1" value="Questa è una prova" />
  <input type="hidden" id="t2" name="t2" value="di abilità" />
  <p id="t3">che utilizza Javascript</p>
</body>

</html>
```

indicare quale verrà visualizzata fra le seguenti opzioni (attenzione ad eventuali errori nel codice):

- ☒ che utilizza Javascript
- ☐ CHE UTILIZZA JAVASCRIPT ver1.5
- ☐ Questa è una prova CHE UTILIZZA JAVASCRIPT ver1.5
- ☐ Questa è una prova DI ABILITA' che utilizza Javascript ver1.5

Domanda 9 [Q009] HTML5 rispetta le proprietà dei linguaggi XML.

- ☐ Vero
- ☒ Falso

Domanda 10 [Q010] (Extra) 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 "Will you see the source?"
- ☒ B "Will you see the target?"
- ☐ C Viene lanciata una IllegalStateException.
- ☐ D Non appare niente; il thread si arresta.
- ☐ E Appaiono sia "Will you see the source?" che "Will you see the target?".

Domanda 11 [Q011] ♣ 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 “IIIappello15-16”:

```
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?

- ☒ http://localhost:8080/IIIappello15-16/ser_chiamante
- ☐ http://localhost:8080/IIIappello15-16/servlet/ser_chiamante
- ☐ http://localhost/IIIappello15-16/ser_chiamante
- ☐ http://localhost:8080/IIIappello15-16/Chiamante
- ☐ Nessuna risposta è giusta.

Domanda 12 [Q012] Qual è la sequenza che descrive l’ordine in cui sono invocati i metodi JSP?

- ☐ jspInit(), do_Get(), jspDestroy()
- ☐ jspService(), jspInvoke(), _jspDestroy()
- ☐ jspinit(), jspCreate(), _jspService()
- ☒ jspInit(), _jspService(), jspDestroy()

Domanda 13 [Q013] I seguenti sono i passi necessari per l’esecuzione di una query in JDBC. 1) Caricamento del driver (non sempre necessario) 2) Apertura di una connessione con i parametri opportuni 3) Creazione di un oggetto di tipo PreparedStatement che prende come parametro una query 4) Esecuzione della query (eventualmente memorizzazione delle tuple di ritorno in un oggetto di tipo ResultSet) 5) Chiusura della connessione

- ☒ Vero
- ☐ Falso

Domanda 14 [Q014] ♣ Indicare quali delle seguenti affermazioni sono vere:

- ☒ se ci sono più utenti su macchine diverse che accedono alla stessa servlet, lo scope di applicazione permette loro di condividere lo stesso dato
- ☒ una variabile di istanza di una servlet viene condivisa da tutti gli utenti di quella servlet
- ☐ non c'è differenza di visibilità fra un dato mantenuto in una variabile di istanza di una servlet ed un dato mantenuto in una variabile con scoping di sessione.
- ☐ lo scoping di sessione permette di far condividere a più sessioni lo stesso dato
- ☒ lo scope di request fa condividere uno stesso dato a più servlet o jsp a patto che queste stiano eseguendo all'interno della stessa request.
- ☐ Nessuna risposta è giusta.

Domanda 15 [Q015] ♣ La sessione tra più richieste di un client può essere mantenuta:

- ☒ Utilizzando HttpSession
- ☒ Utilizzando i cookies
- ☒ Utilizzando un identificativo di sessione che viene spedito ogni volta dal client come parametro
- ☐ Nessuna risposta è giusta.

Domanda 16 [Q016] Quale sarà l'output risultante dall'esecuzione del seguente pezzo di codice JSP:

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

- ☒ To be
- ☐ or not to be
- ☐ To be, or not to be
- ☐ Il codice non verrà compilato.

Domanda 17 [Q017] In quale directory sono messi i file JAR?

- ☐ Direttamente nella directory dell'applicazione
- ☐ Direttamente all'interno della directory /WEB-INF
- ☐ Direttamente all'interno della directory /WEB-INF/classes
- ☒ Direttamente all'interno della directory /WEB-INF/lib

Domanda 18 [Q018] 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 installata) prima di Giovanni e dopo Michele quale valore di x vedranno i tre amici? I tre amici vivono in città diverse.

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

Domanda 19 [Q019] Come è classificabile il seguente documento XML

```
<?xml version="1.0"?>
  <!DOCTYPE note [
    <!ELEMENT note (to,from,heading,body)>
    <!ELEMENT to (#PCDATA)>
    <!ELEMENT from (#PCDATA)>
    <!ELEMENT heading (#PCDATA)>
    <!ELEMENT body (#PCDATA)>
  ]>
  <note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend</body>
  </note>
```

- ☐ A ben formato ma non valido
- ☐ B mal formato ma valido
- ☒ C valido
- ☐ D non valido
- ☐ E mal formato e non valido.

Domanda 20 [Q020] Quali sono i livelli di “scope” accessibili da una JSP ?

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

Domanda 21 [Q021] Quale metodo si utilizza in una servlet per leggere il valore di un parametro [name] inviato via metodo POST?

- ☐ A request.getAttribute(name)
- ☐ B request.getInitParameter(name,true)
- ☐ C response.getPostParameter(name)
- ☒ request.getParameter(name)

Domanda 22 [Q022] (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.

(Soluzione a fine documento)

Domanda 23 [Q023] (punteggio: da 0 a 4) Produrre una DTD che descriva la struttura di un documento XML che rappresenti una lista non vuota di prodotti. Ciascun prodotto è descritto dai seguenti dati: marca, modello, prezzo, url_immagine, nome e cognome del fornitore. Tali dati devono rispettare i seguenti vincoli: 1) Il prezzo deve essere sempre presente e deve essere definito come attributo; 2) il fornitore deve essere un elemento; 3) nome e cognome devono essere elementi figli del fornitore.

```
<!DOCTYPE listaprodotti [  
  <!ELEMENT listaprodotti (prodotto)+>  
  <!ELEMENT prodotto (marca,modello,url_immagine,fornitore)>  
  <!ELEMENT fornitore (nome,cognome)>  
  <!ELEMENT marca (#PCDATA)>  
  <!ELEMENT modello (#PCDATA)>  
  <!ELEMENT url_immagine (#PCDATA)>  
  <!ELEMENT nome (#PCDATA)>  
  <!ELEMENT cognome (#PCDATA)>  
  <!ATTLIST prodotto prezzo cdata #REQUIRED>  

```


Domanda 24 [Q024] (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. (Suggerimento: svolgere dapprima l'esercizio visualizzando solo l'ultimo indirizzo e poi completarlo)

```
<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 25 [Q025] (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;
}
```

Domanda 26 [Q026] (punteggio: da 0 a 4) Effettuare (su questo foglio) le opportune correzioni nel metodo `processRequest` della seguente servlet:

```
package esame;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.http.HttpSession;

@WebServlet("/Messaggio")
public class Messaggio extends HttpServlet {
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }
    public void destroy() {
    }
    protected void processRequest(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        HttpSession session = request.getSession();
        String test = (String)requestsession.getAttribute("test");
        // output in HTML
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Sei proprio tu</title>");
        out.println("</head>");
        out.println("<body>");
        if (test==null) {
            out.println("<h1>Felice di conoscerti</h1>");
            requestsession.setAttribute("test", "s");
        } else out.println("<h1>Bentornati</h1>");
        out.println("</head>");
        out.println("<body>");
        out.println("</body>");
        out.println("</html>");
        out.close();
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
    public String getServletInfo() {
        return "Short description";
    }
}
```

Soluzione domanda 22:

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 *****
```

CATALOGO

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>
```

CATALOGO

CATALOGO

Foglio risposte

Studente:

0

Domanda 1: ☐ A ☐ B ☐ C ☒Domanda 2: ☒ ☐ B ☐ C ☐ DDomanda 3: ☐ A ☒ ☐ C ☐ DDomanda 4: ☐ A ☒ ☐ C ☐ DDomanda 5: ☐ A ☒ ☐ C ☐ DDomanda 6: ☐ A ☐ B ☒ ☐ DDomanda 7: ☐ A ☒ ☐ C ☐ DDomanda 8: ☒ ☐ B ☐ C ☐ DDomanda 9: ☐ A ☒Domanda 10: ☐ A ☒ ☐ C ☐ D ☐ EDomanda 11: ☒ ☐ B ☐ C ☐ D ☐ EDomanda 12: ☐ A ☐ B ☐ C ☒Domanda 13: ☒ ☐ BDomanda 14: ☒ ☒ ☐ C ☐ D ☒ ☐ FDomanda 15: ☒ ☒ ☒ ☐ DDomanda 16: ☒ ☐ B ☐ C ☐ DDomanda 17: ☐ A ☐ B ☐ C ☒Domanda 18: ☐ A ☐ B ☒ ☐ DDomanda 19: ☐ A ☐ B ☒ ☐ D ☐ EDomanda 20: ☒ ☐ B ☐ C ☐ DDomanda 21: ☐ A ☐ B ☐ C ☒Domanda 22: [MVC] ☐ 0 ☐ 05 ☐ 1 ☐ 15 ☐ 2 ☐ 25 ☐ 3 ☐ 35 ☒ Non marcare questi boxDomanda 23: [XML prodotti] ☐ 0 ☐ 05 ☐ 1 ☐ 15 ☐ 2 ☐ 25 ☐ 3 ☐ 35 ☒ Non marcare questi boxDomanda 24: [JSP indirizzo posta] ☐ 0 ☐ 05 ☐ 1 ☐ 15 ☐ 2 ☐ 25 ☐ 3 ☐ 35 ☒ Non marcare questi boxDomanda 25: [JS verifica testo] ☐ 0 ☐ 05 ☐ 1 ☐ 15 ☐ 2 ☐ 25 ☐ 3 ☐ 35 ☒ Non marcare questi boxDomanda 26: [JS lettura servlet] ☐ 0 ☐ 05 ☐ 1 ☐ 15 ☐ 2 ☐ 25 ☐ 3 ☐ 35 ☒ Non marcare questi box

CATALOGO