

MyTableWithForm.java

```
1 package mypackage;
2
3 import java.io.*;
4 import javax.servlet.*;
5 import javax.servlet.annotation.*;
6 import javax.servlet.http.*;
7
8 /**
9  * Η ακόλουθη δήλωση (@WebServlet)
10 * εξασφαλίζει ότι το servlet θα κληθεί
11 * καθορίζοντας το εγγραφο mytableform στο URL προσπέλασης
12 * (π.χ. http://localhost/myproject/mytableform
13 * )
14 @WebServlet("/mytableform")
15 public class MyTableWithForm extends HttpServlet {
16     @Override
17     public void doGet(HttpServletRequest request, HttpServletResponse response)
18         throws ServletException, IOException {
19
20         /* Δηλώσεις τοπικών μεταβλητών */
21         String rowCountParam;
22         String columnCountParam;
23         int rowCount = 0;
24         int columnCount = 0;
25
26         /* Οι μεταβλητές τύπου boolean μπορούν να έχουν τιμές true ή false */
27         boolean buttonPressed;
28         boolean badValues = false;
29
30         /*
31         * Η μεταβλητή buttonPressed θα γίνει true αν ανιχνεύσουμε ότι πατήθηκε
32         * το κουμπί "Make Table". Χρησιμοποιούμε την τιμή της ιδιότητας name
33         * του κουμπιού.
34         */
35         buttonPressed = (request.getParameter("makeTable") != null);
36         rowCountParam = request.getParameter("rowCount");
37         columnCountParam = request.getParameter("columnCount");
38         /*
39         * Αν πατήθηκε το κουμπί επεξεργαζόμαστε τις τιμές στα δύο πεδία. Τις
40         * μετατρέπουμε από String σε int χρησιμοποιώντας τη βοηθητική κλάση
41         * Integer. Αν η μετατροπή δεν είναι εφικτή λόγω μη εισαγωγής ακεραίων
42         * θα δημιουργηθεί η εξαίρεση NumberFormatException την οποία την
43         * συλλαμβάνουμε και θέτουμε μεταβλητή ελέγχου badValues σε true.
44         */
45         if (buttonPressed) {
46
```

MyTableWithForm.java

```
47         try {
48             rowCount = Integer.parseInt(rowCountParam);
49             columnCount = Integer.parseInt(columnCountParam);
50         } catch (NumberFormatException e) {
51             badValues = true;
52         }
53     }
54
55     /* Ξεκινούμε το χτίσιμο της απόκρισης */
56     response.setContentType("text/html");
57     response.setCharacterEncoding("UTF-8");
58     PrintWriter out = response.getWriter();
59     out.println("<!DOCTYPE html>\n" + "<html>\n"
60         + "<head><title>A Table Generating Example 2</title></head>\n"
61         + "<body bgcolor=\"#fdf5e6\">\n" + "<h1>A Table Generating
Example</h1>\n");
62
63     /*
64     * Εισάγουμε την html σύνταξη που θα δημιουργήσει την φόρμα. Δεν
ορίζουμε
65     * την ιδιότητα ACTION της φόρμας και αυτό θα έχει σαν αποτέλεσμα να
66     * κληθεί το ίδιο URL που εμφανισε το έγγραφο και μετά το πάτημα του
κουμπιού.
67     *
68     * Η εκφραση (rowCountParam==null?"":rowCountParam) της Java
69     * χρησιμοποιείται για το καθορισμό αρχικών τιμών στα πεδία κειμένου.
70     * Έχει την μορφή <συνθήκη>τιμη1:τιμη2. Αν η συνθήκη είναι true τότε
71     * επιστρέφεται η τιμη1 διαφορετικά επιστρέφεται η τιμη2
72     */
73     out.println("<form>\n"
74         + "Row Count <input type=\"text\" name=\"rowCount\" value=\""
75         + (rowCountParam == null ? "" : rowCountParam)
76         + "\" /><br>\n"
77         + "Column Count <input type=\"text\" name=\"columnCount\"
value=\""
78         + (columnCountParam == null ? "" : columnCountParam)
79         + "\" /><br>\n"
80         + "<input type=\"submit\" name=\"makeTable\" value=\"Make
Table\" /><br>\n"
81         + "</form>");
82
83     /*
84     * Μετά την δημιουργία της φόρμας ελεγχουμε αν βρισκόμαστε στην
85     * περίπτωση που δημιουργούμε το έγγραφο επειδή ο χρήστης απλα
86     * χρησιμοποίησε το URL ή πάτησε το κουμπί "Make Table"
87     */
```

MyTableWithForm.java

```
88     if (buttonPressed) {
89         /*
90          * Αν πάτησε το κουμπί ελέγχουμε αν οι τιμές στα πεδία κειμένου
91          * μετατράπηκαν σε ακραίους χωρίς πρόβλημα
92          */
93         if (badValues) {
94             /*
95              * Αν υπήρξε πρόβλημα στην μετατροπή εμφανίζουμε ένα μήνυμα
96              * λάθους
97              */
98             out.println("<strong style=\"color:red\">Πρέπει να δώσετε
ακέριες τιμές στις παραμέτρους</strong>");
99         } else {
100             /*
101              * Διαφορετικά τυπώνουμε τον πίνακα καλώντας την συνάρτηση
102              * printTable
103              */
104             printTable(out, rowCount, columnCount);
105         }
106     }
107
108     /*
109     * Και στις δύο περιπτώσεις εισάγουμε τις HTML ετικέτες που θα
110     * τερματίσουν το έγγραφο.
111     */
112     out.println("</body></html>");
113 }
114
115 /*
116 * Η τοπική συνάρτηση printTable θα τυπώσει τον HTML κώδικα για την
117 * δημιουργία του πίνακα στη ροή εξόδου
118 */
119 private void printTable(PrintWriter out, int rowCount, int columnCount) {
120     out.println("<table border=\"1\">");
121     // Print the table header
122     out.println("<tr>");
123     for (int i = 1; i <= columnCount; i++) {
124         out.println("<th> Header " + i + "</th>");
125     }
126     out.println("</tr>");
127     // Print the table rows
128     for (int i = 1; i <= rowCount; i++) {
129         out.println("<tr>");
130         for (int j = 1; j <= columnCount; j++) {
131             out.println("<td>row " + i + ", col " + j + "</td>");
132         }
133     }
134 }
```

MyTableWithForm.java

```
133         out.println("</tr>");
134     }
135     out.println("</table>");
136 }
137 }
138
```