Βάσεις Δεδομένων και Web-based Εφαρμογές

Ενότητα: Εισαγωγή στις δυναμικές JSP σελίδες με παράδειγμα - Calculator

Χ. Σκουρλάς

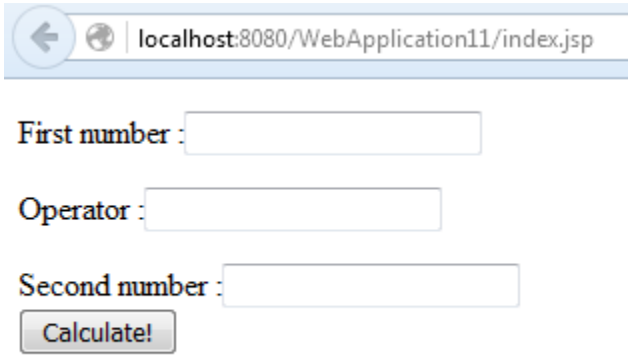
Τμήμα Μηχανικών Πληροφορικής ΤΕ

Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

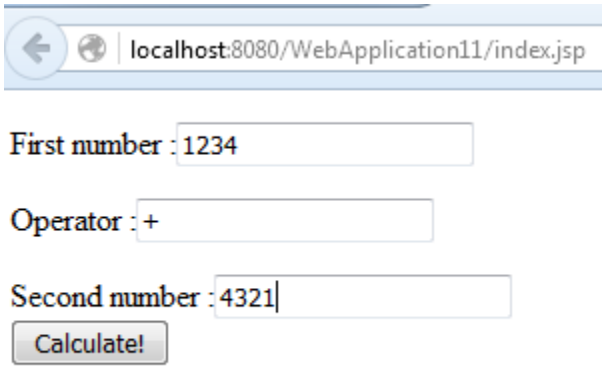
Εισαγωγή στις δυναμικές JSP σελίδες με παράδειγμα - Calculator

Στο κεφάλαιο αυτό θα κατασκευάσουμε έναν απλοποιημένο Calculator. Θα ξεκινήσουμε με την πρόσθεση (addition) δύο ακεραίων.

Στην αρχή θα κατασκευάσουμε τη σελίδα index.html / index.jsp που εμφανίζει μία ηλεκτρονική φόρμα.



Στη φόρμα αυτή μπορούμε να πληκτρολογήσουμε τους δύο αριθμούς και την πράξη.



Η σελίδα index.html / index.jsp περιλαμβάνει στον κώδικά της μία φόρμα. Να η πρώτη γραμμή της φόρμας:

<form name="Formname" method="post" action="calc.jsp" >

Το όνομα της φόρμας είναι Formname. Η μέθοδος είναι "post" και η ενέργειά της (action) είναι η κλήση της σελίδας "calc.jsp"

Στη φόρμα υπάρχουν τρία κουτιά εισαγωγής κειμένου (text boxes) και ένα κουμπί.

Να τι κάνει η ακόλουθη γραμμή κώδικα:

First number :<input type="text" name="nu1" >

Εμφανίζει την οδηγία/λεκτικό (caption) “First number :” και δίπλα το κουτί κειμένου (input type="text"). Ο αριθμός που πληκτρολογεί ο χρήστης στο κουτί αποθηκεύεται στη μεταβλητή nu1.

Να τι κάνει η ακόλουθη γραμμή κώδικα:

Operator :<input type="text" name="op" maxlength="1">

Εμφανίζει την οδηγία/λεκτικό (caption) “ Operator :” και δίπλα το κουτί κειμένου (input type="text") που έχει μήκος 1. Το σύμβολο που πληκτρολογεί ο χρήστης στο κουτί αποθηκεύεται στη μεταβλητή op.

Να τι κάνει η ακόλουθη γραμμή κώδικα:

Second number :<input type="text" name="nu2">

Εμφανίζει την οδηγία/λεκτικό (caption) “ Second number :” και δίπλα το κουτί κειμένου (input type="text"). Ο αριθμός που πληκτρολογεί ο χρήστης στο κουτί αποθηκεύεται στη μεταβλητή nu2.

Να τι κάνει η ακόλουθη γραμμή κώδικα:

<input type="submit" name="name" value="Calculate!">

Εμφανίζει το κουμπί που γράφει “Calculate!”. Όταν πατηθεί το κουμπί που γράφει “Calculate!” η σελίδα μας καλεί τη σελίδα calc.jsp.

Η μέθοδος της φόρμας είναι post, και η φόρμα όταν πατηθεί το κουμπί που γράφει “Calculate!” καλεί τη σελίδα calc.jsp. Η μέθοδος post «περνάει» στη φόρμα calc.jsp τις τιμές που πληκτολογήσαμε στα κουτιά κειμένου. Οι μεταβλητές nu1, nu2, op, που στέλνει η σελίδα μας, έχουν τις τιμές που πληκτρολογήσαμε, και οι τιμές αυτές είναι είναι συμβολοσειρές (string).

Ακολουθεί ο κώδικας της φόρμας Formname:

```
<form name="Formname" method="post" action="calc.jsp" >
    <p>   First number :<input type="text" name="nu1" >
    <p>   Operator :<input type="text" name="op"  maxlength="1">
    <p>   Second number :<input type="text" name="nu2">
    <br><input type="submit" name="name" value="Calculate!">
  </form>
```

Στη συνέχεια, παρατίθεται ο κώδικας όλης της σελίδας index.html / index.jsp.

index.html

```
<!DOCTYPE html>
<!--
Calculator
-->
<html>
  <head>
    <title>Simple calculator</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form name="Formname" method="post" action="calc.jsp" >
        <p>  First number :<input type="text" name="nu1" >
        <p>  Operator :<input type="text" name="op"  maxlength="1">
        <p>  Second number :<input type="text" name="nu2">
        <br><input type="submit" name="name" value="Calculate!">
    </form>
  </body>
</html>
```

Η jsp σελίδα calc.jsp είναι μία html σελίδα που περιλαμβάνει scriptlets (κώδικα java) ανάμεσα σε tag:

<% %>

Στο επόμενο scriptlet ορίζουμε μεταβλητές:

```
<%
  int result;
  int n1=Integer.parseInt(request.getParameter("nu1"));
  String op=request.getParameter("op");
  int n2=Integer.parseInt(request.getParameter("nu2"));
%>
```

Χρησιμοποιούμε τις ακέραιες μεταβλητές result για το αποτέλεσμα της πράξης, n1 για τον πρώτο αριθμό, n2 για τον δεύτερο αριθμό. Χρησιμοποιούμε τη μεταβλητή συμβολοσειρά (String) op για την πράξη.

Για να αποθηκεύσουμε τις τιμές που έστειλε η σελίδα index.html/index.jsp χρησιμοποιούμε τη μέθοδο getParameter της κλάσης request. Επειδή οι τιμές είναι συμβολοσειρές (String) αν θέλουμε να τις μετατρέψουμε σε ακέραιους αριθμούς τότε χρησιμοποιούμε τη μέθοδο parseInt της κλάσης Integer.

Αν θέλουμε να τις μετατρέψουμε σε πραγματικούς αριθμούς τότε χρησιμοποιούμε τη μέθοδο parseFloat της κλάσης Float.

Στο επόμενο scriptlet κάνουμε την πράξη και εμφανίζουμε το αποτέλεσμα.

```
<%
    if(op.equals("+"))
     { result=n1+n2;
       out.println(result); }
 %>
```

Προσοχή! Δε χρησιμοποιούμε τη System.out.println γιατί τότε το αποτέλεσμα είναι στον server. Παρατηρήστε, επιπλέον, ότι δε γινεται κανένας έλεγχος για την πράξη που πληκτρολογεί ο χρήστης κ.λπ. Στη συνέχεια, παρατίθεται ο κώδικας όλης της σελίδας calc.jsp.

calc.jsp (-- simple addition)

```
<%--
    Document   : calc
    Created on : 20 Νοε 2017, 8:23:54 πμ
    Author     : Christos
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="java.util.*" %>
<%@ page import="java.*" %>
<%
    int result;
    int n1=Integer.parseInt(request.getParameter("nu1"));
    String op=request.getParameter("op");
    int n2=Integer.parseInt(request.getParameter("nu2"));
%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Simplified calculator - additin</title>
    </head>
    <body>
        <h1>Results</h1>
 <%
```

```
      if(op.equals("+")) { result=n1+n2;
          out.println(result); }
                        %>
</body>
</html>
```

Αλλάζουμε τη σελίδα calc.jsp έτσι ώστε όταν πληκτρολογούμε λάθος πράξη να απαντά με μήνυμα "wrong operation".

```
<%
      String opname="none";
      if(op.equals("+"))
           {result=n1+n2; opname="     addition";
            out.println(result); }
      else {out.println("wrong operation");}
 %>
```

Το scriptlet αυτό εύκολα ξαναγράφεται ώστε η σελίδα μας να υποστηρίζει τις τέσσερις πράξεις.

Στη συνέχεια, παρατίθεται ο κώδικας όλης της νέας σελίδας calc.jsp.

calc.jsp (-- addition and wrong operator)

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="java.util.*" %>
<%@ page import="java.*" %>
<%
   int result;
   int n1=Integer.parseInt(request.getParameter("nu1"));
```

```
  String op=(String)request.getParameter("op");
  int n2=Integer.parseInt(request.getParameter("nu2"));
%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Simple calculator</title>
   </head>
   <body>
      <h1>Results</h1>
<%
      String opname="none";
      if(op.equals("+"))
            {result=n1+n2; opname="     addition";
             out.println(result); }
      else {out.println("wrong operation");}
                  %>
</body>
</html>
```

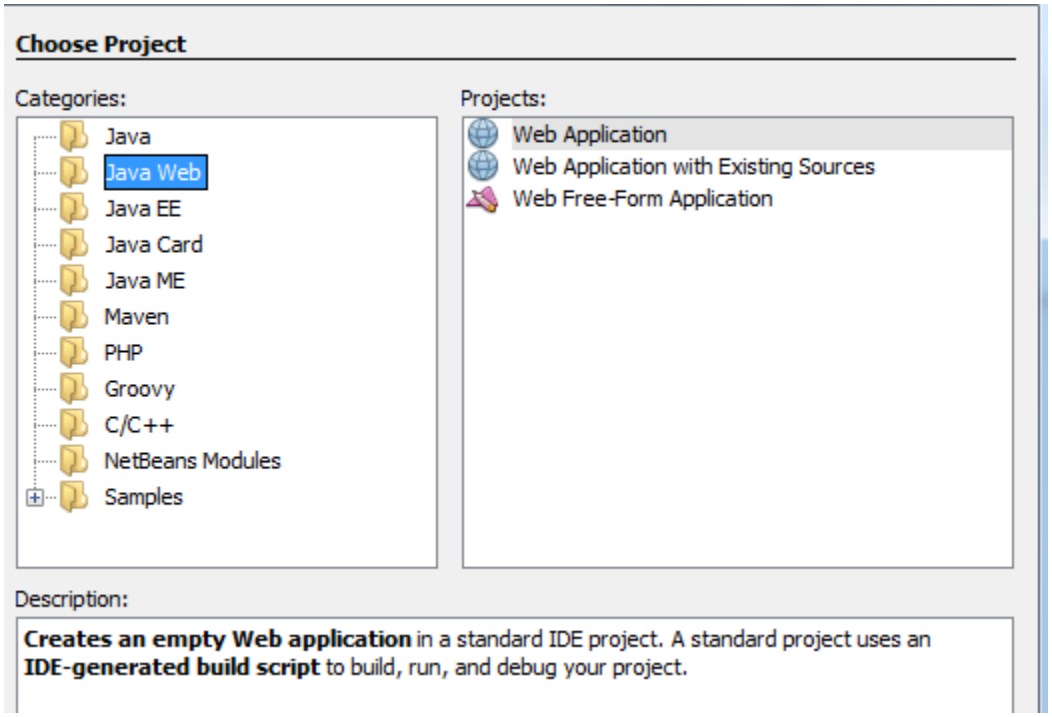

Στη συνέχεια, παρατίθεται ο κώδικας της αρχικής σελίδας calc.jsp με μικροτροποιήσεις που επιτρέπουν πρόσθεση πραγματικών αριθμών.

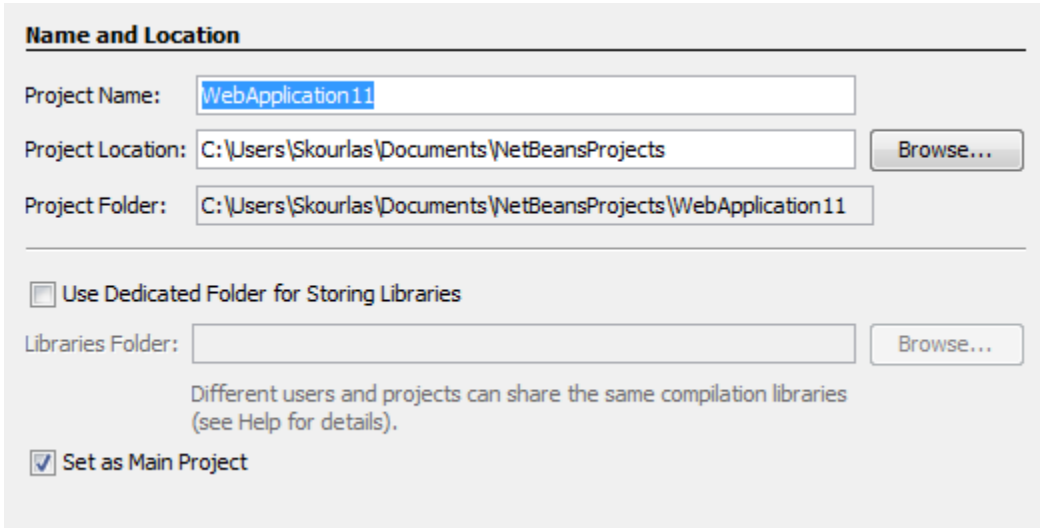
calc.jsp (-- addition – Float – parseFloat)

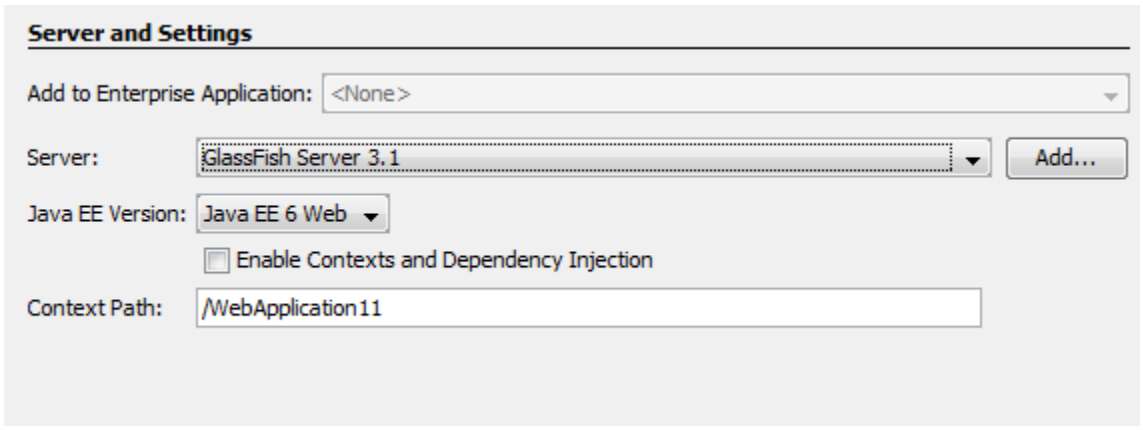
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="java.util.*" %>
<%@ page import="java.*" %>
<%
  float result;
  float n1=Float.parseFloat(request.getParameter("nu1"));
  String op=(String)request.getParameter("op");
  float n2=Float.parseFloat(request.getParameter("nu2"));
%>
<!DOCTYPE html>
<html>
   <head>
       <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
       <title>Simple calculator</title>
   </head>
   <body>
       <h1>Results</h1>
<%
       String opname="none";
       if(op.equals("+"))
              {result=n1+n2; opname="     addition";
               out.println(result); }
       else {out.println("wrong operation");}
                    %>
```

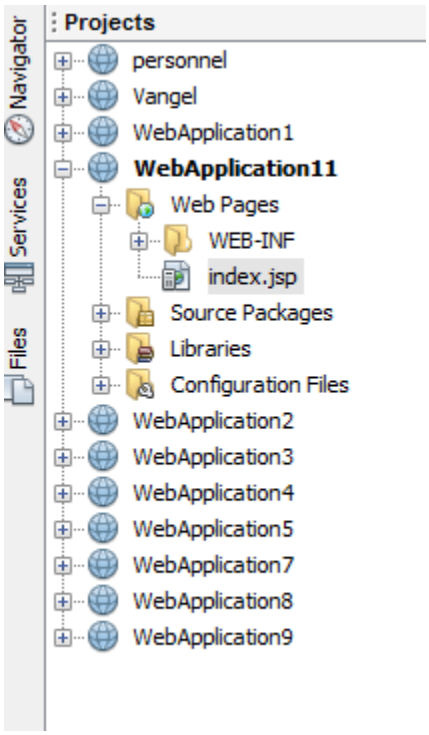
```
</body>

</html>
```

Πως θα κατασκευάσουμε τις σελίδες στο προϊόν Netbeans.

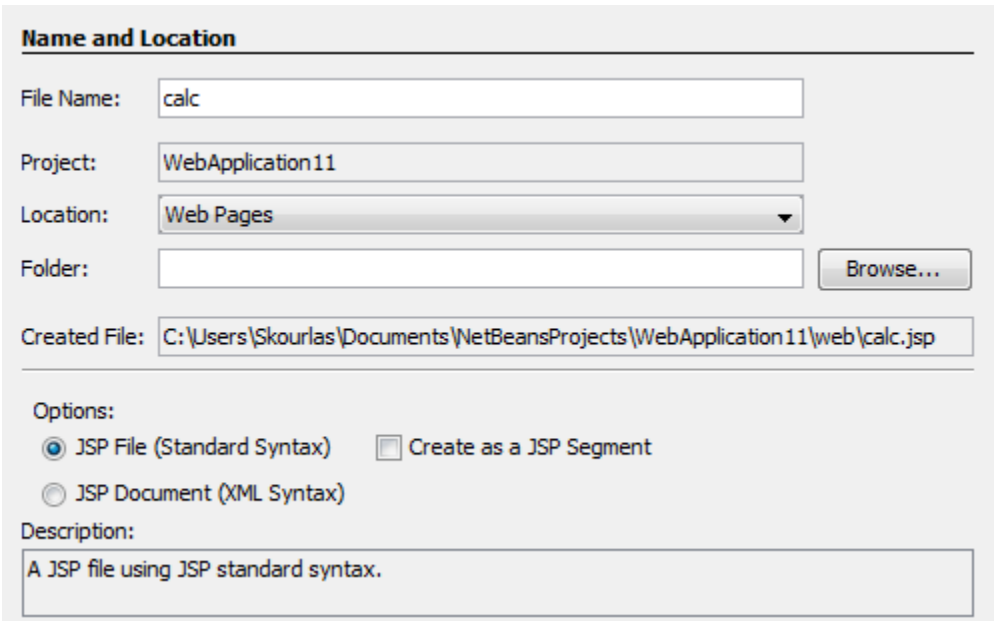


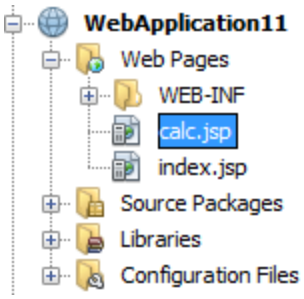



Server and Settings
Add to Enterprise Application:
<None>
Server:
GlassFish Server 3.1
Add...
Java EE Version:
Java EE 6 Web
Enable Contexts and Dependency Injection
Context Path:
/WebApplication11


Navigator
Services
Files
Projects
personnel
Vangel
WebApplication1
WebApplication11
Web Pages
WEB-INF
index.jsp
Source Packages
Libraries
Configuration Files
WebApplication2
WebApplication3
WebApplication4
WebApplication5
WebApplication7
WebApplication8
WebApplication9

```
<!DOCTYPE html>
<!--
Calculator
-->
<html>
    <head>
        <title>Simple calculator</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
          <form name="Formname" method="post" action="calc.jsp" >
              <p>   First number :<input type="text" name="nu1" >
              <p>   Operator :<input type="text" name="op"  maxlength="1">
              <p>   Second number :<input type="text" name="nu2">
              <br><input type="submit" name="name" value="Calculate!">
          </form>
    </body>
</html>
```





Έλεγχος ορθότητας των δεδομένων που πληκτρολογεί ο χρήστης – Χρήση try/catch Block, και κλάσης Exception

Ακολουθεί παράδειγμα χρήσης try/catch Block.

```
<%--
    Document   : try_catch
    Created on : 2 Δεκ 2017, 7:45:44 πμ
    Author     : Christos
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<HTML>
    <HEAD>
        <TITLE>Using a try/catch Block</TITLE>
    </HEAD>

    <BODY>
        <H1>Using a try/catch Block</H1>
    <%
```

```
    try{
        int interest = 0;
        int balance = 1000;
        int interest_rate = 0;
        interest = (balance * interest_rate)/100;
        interest = interest / interest_rate;
        out.println("The answer is " + interest);
    }
    catch (Exception ex){
        out.println("An exception occurred: " + ex.getMessage());
    }
  %>
  </BODY>
</HTML>
```


Το μπλοκ try προστατεύει τον κώδικα που περιλαμβάνει τη μέθοδο που μπορεί να προκαλέσει εξαίρεση (exception). Το μπλοκ catch διαχειρίζεται την εξαίρεση. Να πως θα «μεταφράζαμε» τον κώδικα:

Δοκίμασε (try) τον κώδικα

```
        int interest = 0;
        int balance = 1000;
        int interest_rate = 0;
        interest = (balance * interest_rate)/100;
        interest = interest / interest_rate;
        out.println("The answer is " + interest);
```

Αν εκτελείται σωστά συνέχισε το πρόγραμμά σου.

Αλλιώς, «παγίδευσε»/«άδραξε» (catch) την εξαίρεση και διαχειρίσου την.

```
catch (Exception ex)
    {
        out.println("An exception occurred: " + ex.getMessage());
      }
```

Exception είναι το όνομα της κλάσης που θα «παγιδευτεί» και ex το όνομα της μεταβλητής.

Αν στη σελίδα calc,jsp ήθελα να δημιουργήσω έναν ακέραιο n1 από την τιμή String της μεταβλητής nu1 που μου στέλνει η σελίδα index.jsp:

(JSP page) try_catch.jsp

```
<%--
    Document   : try_catch
    Created on : 2 Δεκ 2017, 7:45:44 πμ
    Author     : Christos
--%>
```



```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<HTML>
    <HEAD>
        <TITLE>Using a try/catch Block</TITLE>
    </HEAD>

    <BODY>
        <H1>Using a try/catch Block</H1>
    <%
        try{
```

```
        String nu1="wrong value";
        float n1=Float.parseFloat(nu1);
        out.println("The answer is " + n1);
        }
        catch (Exception ex){
            out.println("An exception occurred: " + ex.getMessage());
        }
    %>
    </BODY>
</HTML>
```

Έλεγχος ορθότητας των δεδομένων που πληκτρολογεί ο χρήστης – Χρήση try/catch Block, και κλάσης NumberFormatException

Στο επόμενο παράδειγμα η μέθοδος parseFloat() θα προκαλέσει εξαίρεση της κλάσης NumberFormatException.

```
try {
        String nu1="wrong value";
        float n1=Float.parseFloat(nu1);
        out.println("The answer is " + n1);
        }
```

Το μπλοκ catch όταν λαμβάνει οποιαδήποτε αντικείμενα NumberFormatException για τις εξαιρέσεις που προκαλούνται στο μπλοκ try μπορεί να χρησιμοποιήσει μία μεταβλητή για να αναφέρεται στο αντικείμενο της εξαίρεσης.

```
        catch (NumberFormatException nfe)
       {
            out.println("An exception occurred: " + nfe.getMessage());
        }
```

Μπορεί να καλέσει τη μέθοδο getMessage() που υπάρχει σε όλες τις εξαιρέσεις (Exceptions). Η μέθοδος getMessage() εμφανίζει ένα λεπτομερές μήνυμα σφάλματος:

An exception occurred: For input string: "wrong value"

Ακολουθεί το πρόγραμμα.

(JSP page) try_catch.jsp

```
<%--
    Document   : try_catch
    Created on : 2 Δεκ 2017, 7:45:44 πμ
    Author     : Christos
--%>
<%@page import="java.sql.*" %>
<%@page import="java.util.*" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<HTML>
    <HEAD>
        <TITLE>Using a try/catch Block</TITLE>
    </HEAD>

    <BODY>
        <H1>Using a try/catch Block</H1>
    <%
        try{
        String nu1="wrong value";
        float n1=Float.parseFloat(nu1);
```

```
        out.println("The answer is " + n1);
        }
        catch (NumberFormatException nfe){
            out.println("An exception occurred: " + nfe.getMessage());
        }
    %>
    </BODY>
</HTML>
```