
Java EE 7 Web 練習三 Solution

鄭安翔

ansel_cheng@hotmail.com

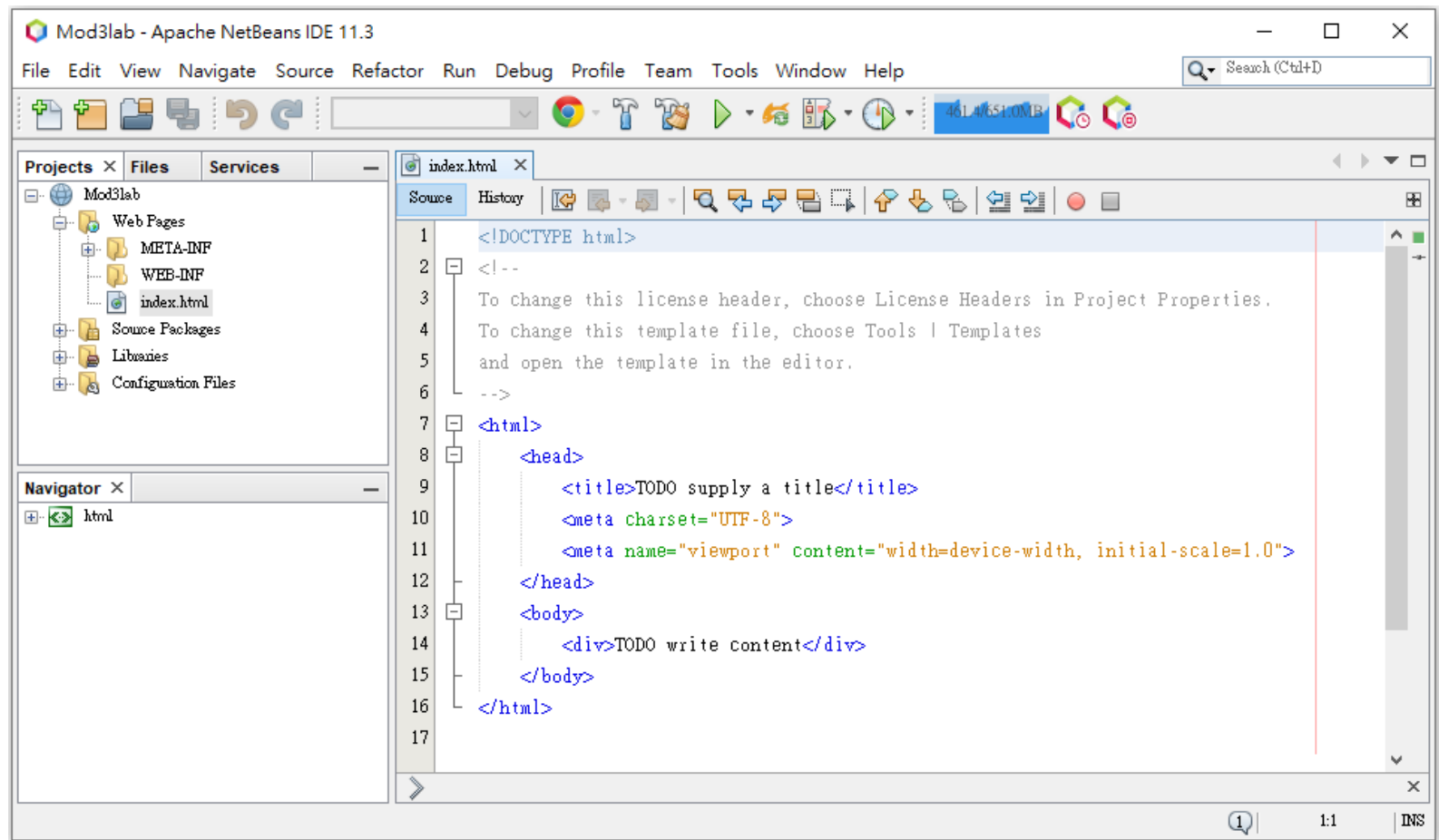
Lab - 觀察JSP解譯過程

- 使用NetBeans IDE，建立Mod3lab 網路應用專案
 - 刪除 index.html
- 撰寫一些簡單的JSP程式
 - index.jsp
- 測試、執行
- 檢視 JSP 解譯後的 Java程式碼
 - 選取JSP檔案,按滑鼠右鍵,選擇 View Servlet

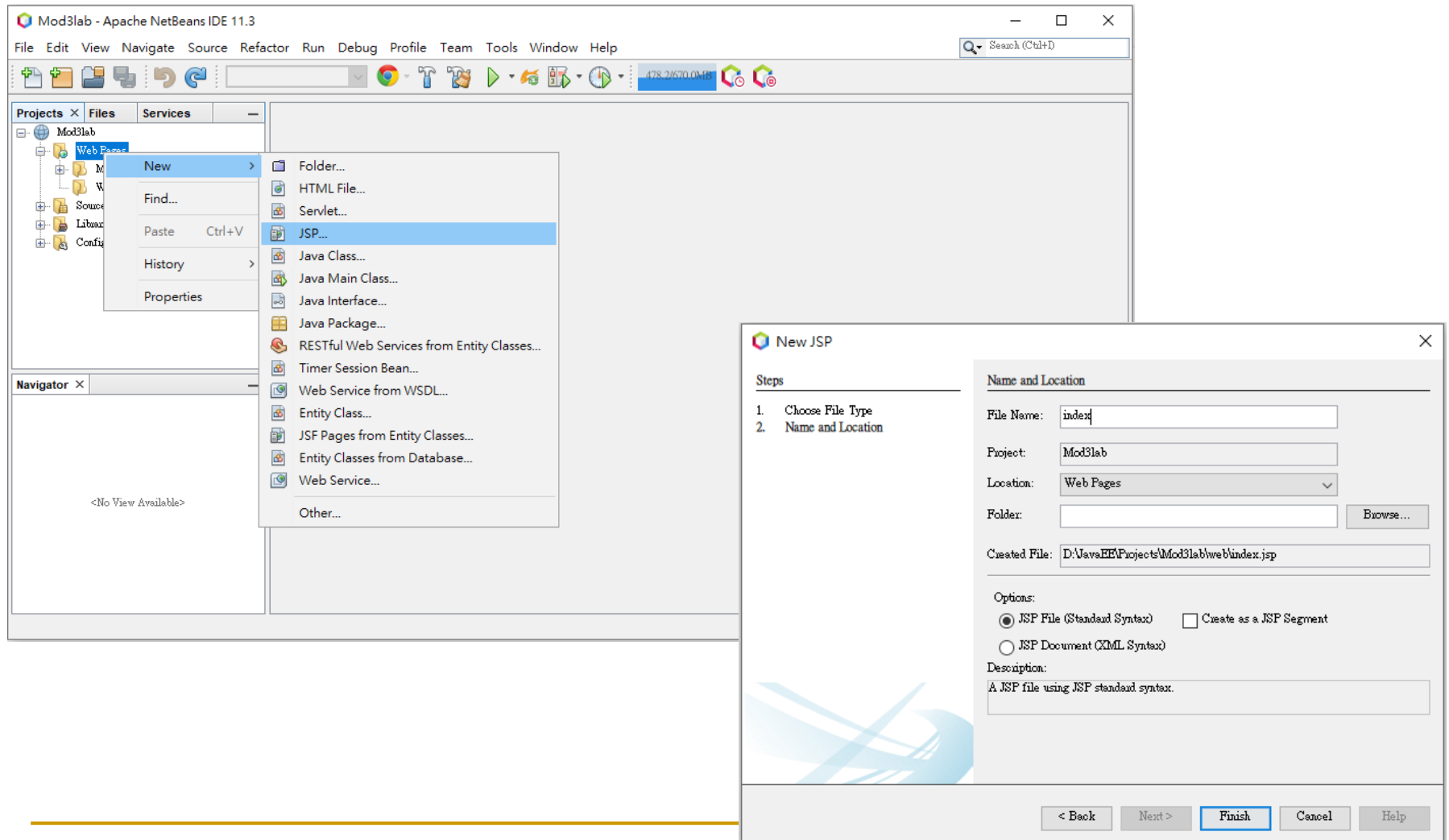
Lab - 觀察JSP解譯過程

- 對JSP作下列修改,檢視解譯後Java程式碼的變化
 - 使用註解標籤(Comment Tag)
 - 使用指令標籤(Directive Tag), import java.util.*
 - 使用宣告標籤(Declaration Tag),宣告一個private int x
 - 使用宣告標籤(Declaration Tag),覆寫 jsplnit()方法
 - 使用表示式標籤(Expression Tag),輸出 x 的值
 - 使用scriptlet標籤, 列印1到x的值
 - 試著移除 <% } %>, 使程式有誤

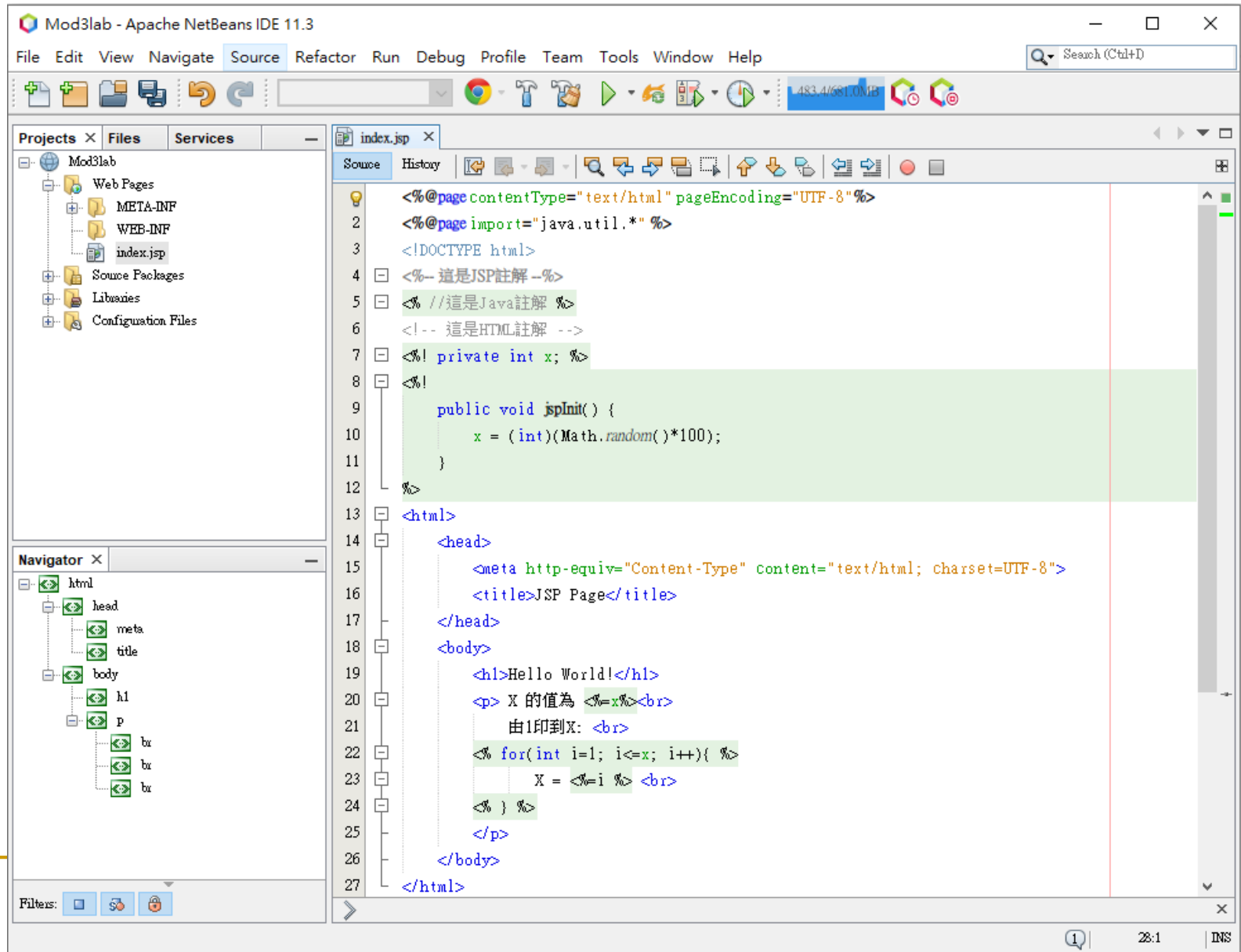
建立Mod3lab專案



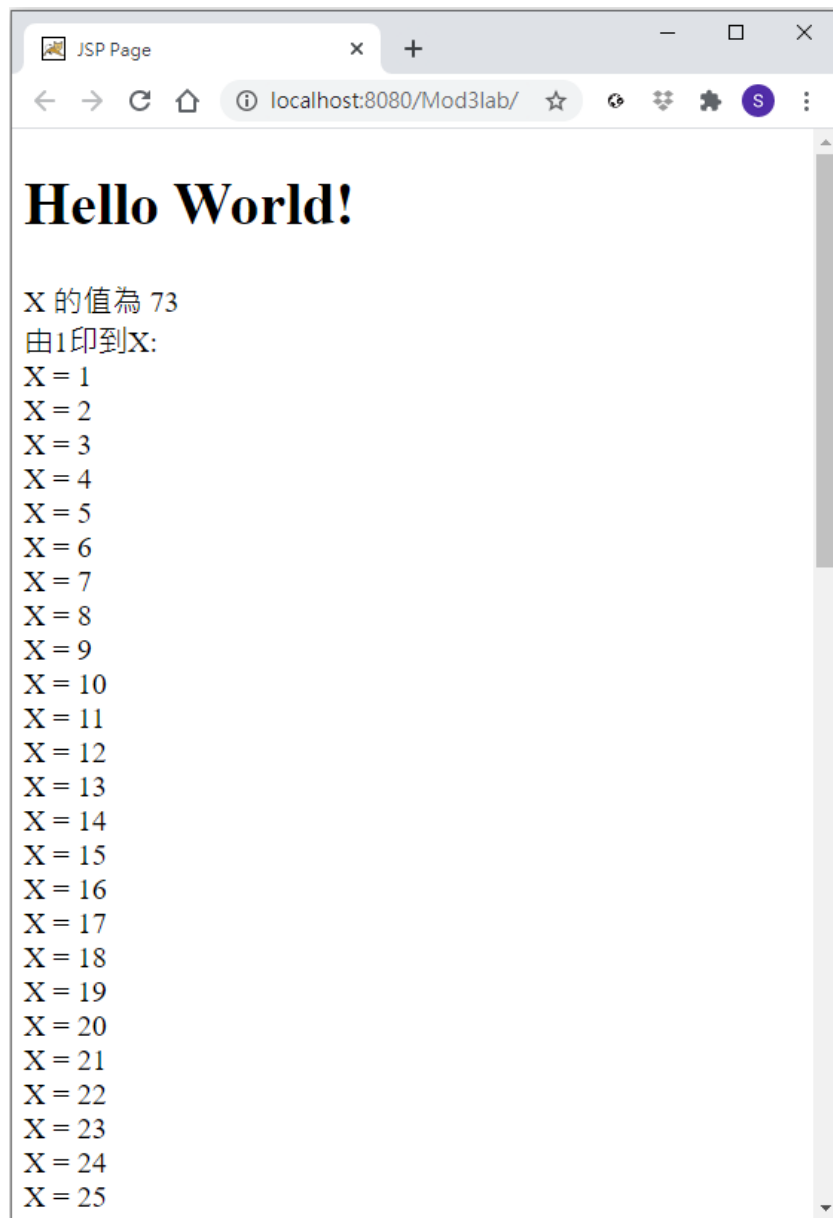
新增 index.jsp



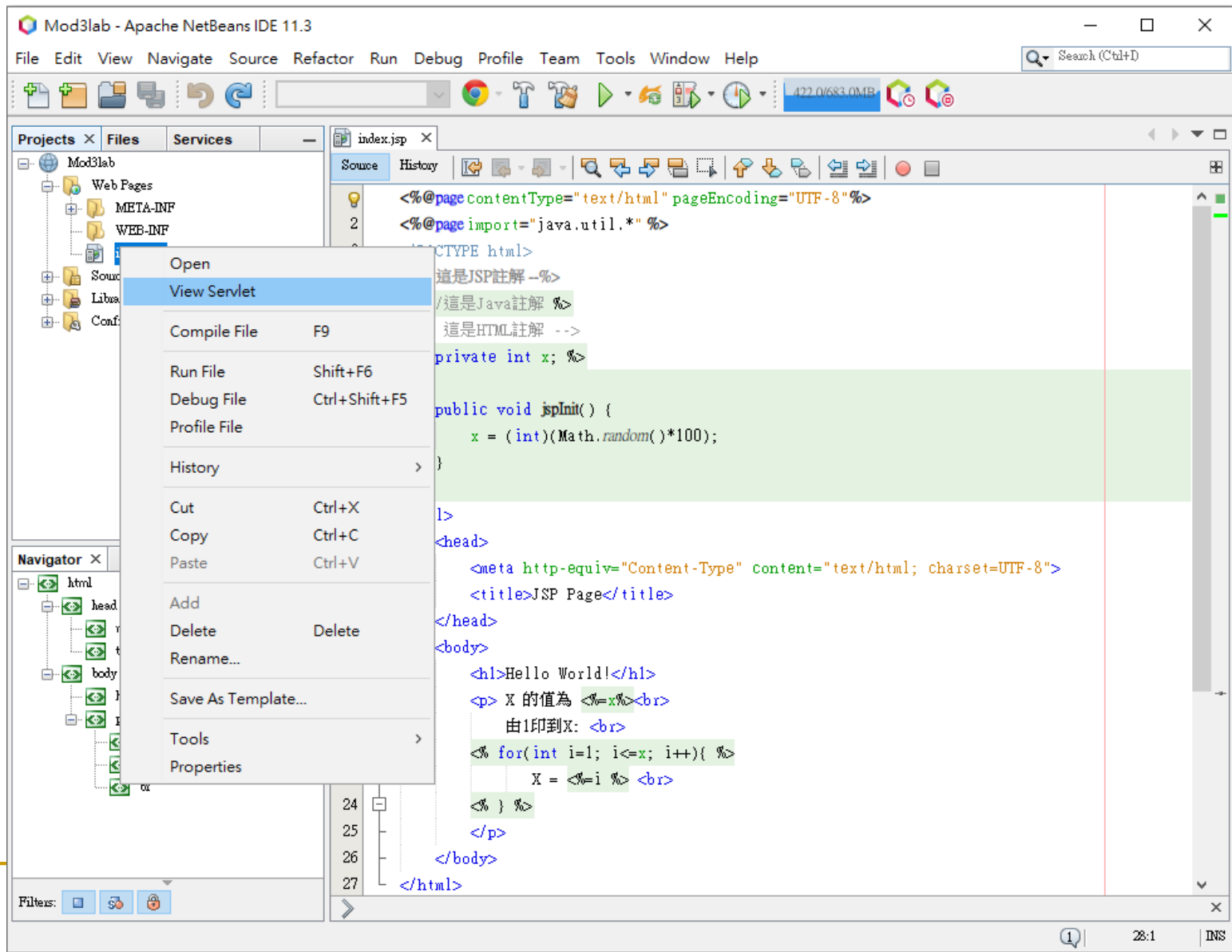
index.jsp



測試、執行



檢視 JSP 解譯後的 Java 程式碼



index_jsp.java

```
index.jsp x index_jsp.java x
package org.apache.jsp;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
import java.util.*;

public final class index_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent,
               org.apache.jasper.runtime.JspSourceImports {

    private int x;

    public void jspInit() {
        x = (int)(Math.random()*100);
    }

    private static final javax.servlet.jsp.JspFactory _jspxFactory =
        javax.servlet.jsp.JspFactory.getDefaultFactory();

    private static java.util.Map<java.lang.String,java.lang.Long> _jspx_dependants;

    private static final java.util.Set<java.lang.String> _jspx_imports_packages;

    private static final java.util.Set<java.lang.String> _jspx_imports_classes;

    static {
        _jspx_imports_packages = new java.util.HashSet<>();
        _jspx_imports_packages.add("javax.servlet");
        _jspx_imports_packages.add("java.util");
        _jspx_imports_packages.add("javax.servlet.http");
        _jspx_imports_packages.add("javax.servlet.jsp");
        _jspx_imports_classes = null;
    }
}
```

```
44 private volatile javax.el.ExpressionFactory _el_expressionfactory;
45 private volatile org.apache.tomcat.InstanceManager _jsp_instancemanager;
46
47 public java.util.Map<java.lang.String,java.lang.Long> getDependants() {
48     return _jspx_dependants;
49 }
50
51 public java.util.Set<java.lang.String> getPackageImports() {
52     return _jspx_imports_packages;
53 }
54
55 public java.util.Set<java.lang.String> getClassImports() {
56     return _jspx_imports_classes;
57 }
58
59 public javax.el.ExpressionFactory _jsp_getExpressionFactory() {
60     if (_el_expressionfactory == null) {
61         synchronized (this) {
62             if (_el_expressionfactory == null) {
63                 _el_expressionfactory = _jspxFactory.getJspApplicationContext(getServletConfig().getServletContext()).getExpressionFactory();
64             }
65         }
66     }
67     return _el_expressionfactory;
68 }
69
70 public org.apache.tomcat.InstanceManager _jsp_getInstanceManager() {
71     if (_jsp_instancemanager == null) {
72         synchronized (this) {
73             if (_jsp_instancemanager == null) {
74                 _jsp_instancemanager = org.apache.jasper.runtime.InstanceManagerFactory.getInstanceManager(getServletConfig());
75             }
76         }
77     }
78     return _jsp_instancemanager;
79 }
80
```

```

81 public void _jspInit() {
82 }
83
84 public void _jspDestroy() {
85 }
86
87 public void _jspService(final javax.servlet.http.HttpServletRequest request, final javax.servlet.http.HttpServletResponse response)
88     throws java.io.IOException, javax.servlet.ServletException {
89
90     final java.lang.String _jspx_method = request.getMethod();
91     if (!"GET".equals(_jspx_method) && !"POST".equals(_jspx_method) && !"HEAD".equals(_jspx_method) && !javax.servlet.DispatcherType.ERROR.equals(request.getDispatcherType())) {
92         response.sendError(HttpServletResponse.SC_METHOD_NOT_ALLOWED, "JSPs only permit GET POST or HEAD");
93         return;
94     }
95
96     final javax.servlet.jsp.PageContext pageContext;
97     javax.servlet.http.HttpSession session = null;
98     final javax.servlet.ServletContext application;
99     final javax.servlet.ServletConfig config;
100     javax.servlet.jsp.JspWriter out = null;
101     final java.lang.Object page = this;
102     javax.servlet.jsp.JspWriter _jspx_out = null;
103     javax.servlet.jsp.PageContext _jspx_page_context = null;
104
105
106     try {
107         response.setContentType("text/html;charset=UTF-8");
108         pageContext = _jspxFactory.getPageContext(this, request, response,
109             null, true, 8192, true);
110         _jspx_page_context = pageContext;
111         application = pageContext.getServletContext();
112         config = pageContext.getServletConfig();
113         session = pageContext.getSession();
114         out = pageContext.getOut();
115         _jspx_out = out;
116

```

```
117 out.write("\n");
118 out.write("\n");
119 out.write("\n");
120 out.write("<!DOCTYPE html>\n");
121 out.write('\n');
122 //这是Java注释
123 out.write("\n");
124 out.write("<!-- 这是HTML注释 -->\n");
125 out.write('\n');
126 out.write("\n");
127 out.write("<html>\n");
128 out.write("    <head>\n");
129 out.write("        <meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\">\n");
130 out.write("        <title>JSP Page</title>\n");
131 out.write("    </head>\n");
132 out.write("    <body>\n");
133 out.write("        <h1>Hello World!</h1>\n");
134 out.write("        <p> X 的值为 ");
135 out.print(x);
136 out.write("<br>\n");
137 out.write("            由1到X: <br>\n");
138 out.write("        ");
139 for(int i=1; i<=x; i++){
140 out.write(" \n");
141 out.write("            X = ");
142 out.print(i );
143 out.write(" <br> \n");
144 out.write("        ");
145 }
146 out.write("\n");
147 out.write("    </p>\n");
148 out.write("    </body>\n");
149 out.write("</html>\n");
150 } catch (java.lang.Throwable t) {
151 if (!(t instanceof javax.servlet.jsp.SkipPageException)){
```