**实验二 内置对象的应用**

# 实验目的

编写具有Request、response、session对象的程序实验。

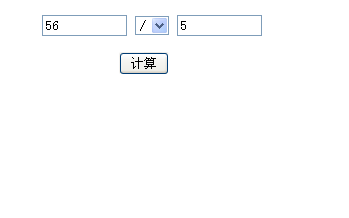
掌握怎样在JSP中使用内置的Request对象、怎样使用Response对象动态响应用户的请求、怎样使用Session对象存储和用户有关的数据。

# 实验要求

编写一个JSP页面，该页面提供一个表单，用户通过表单输入两个数和四则运算符号提交给该页面；编写两个JSP页面，一个页面提交数字给另一个页面，另一个页面使用response对象作出动态响应。

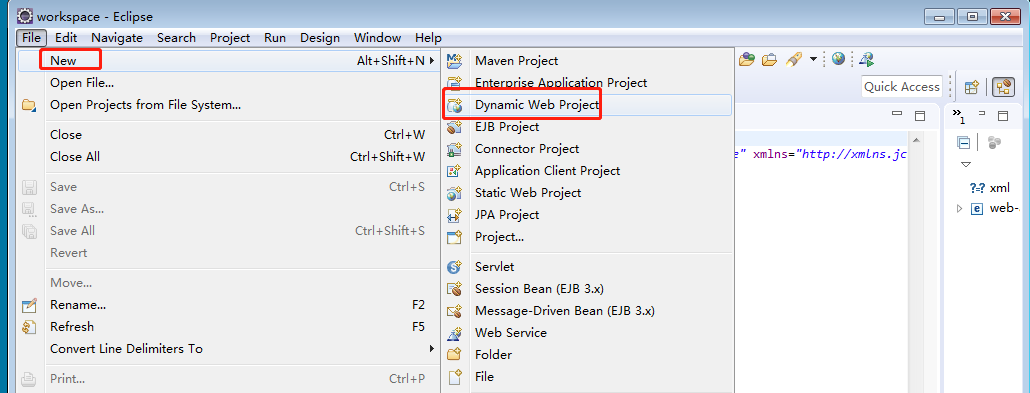
页面的大致效果如下图所示，具体安排可以自由发挥

提交页面：

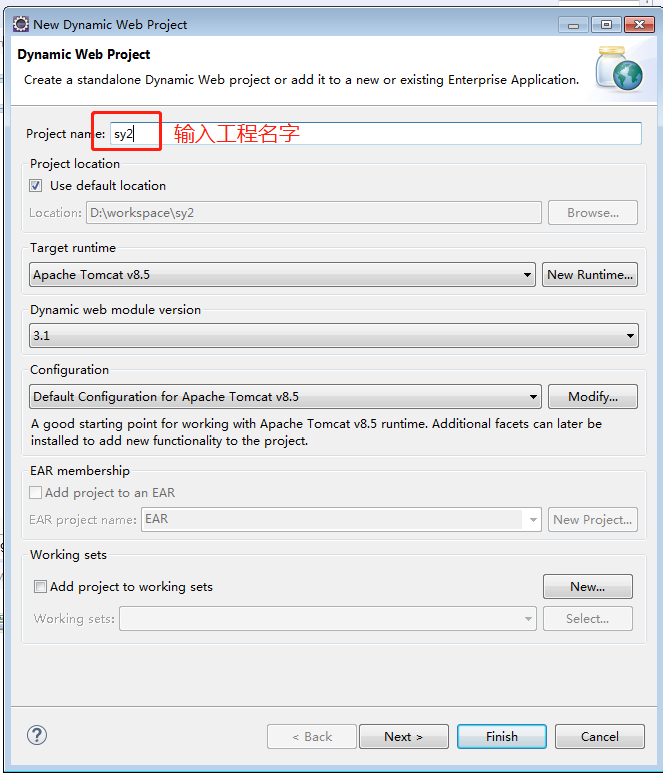


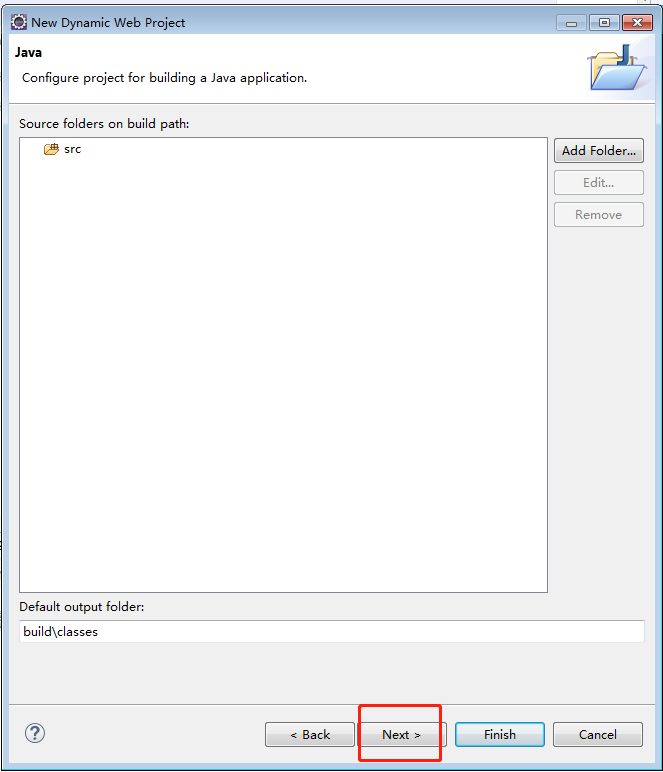
# 新建工程

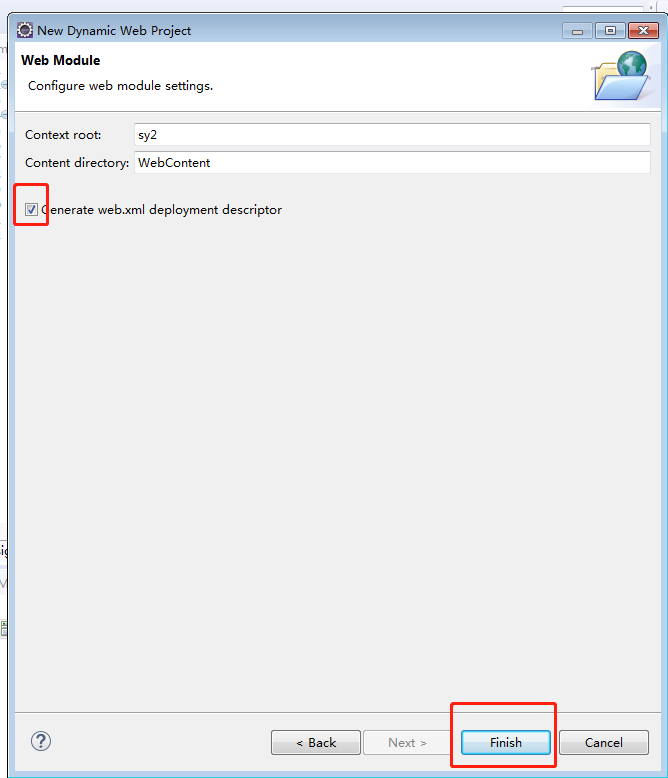
菜单【file】-【new】，选择 dynamic web project



输入工程名字

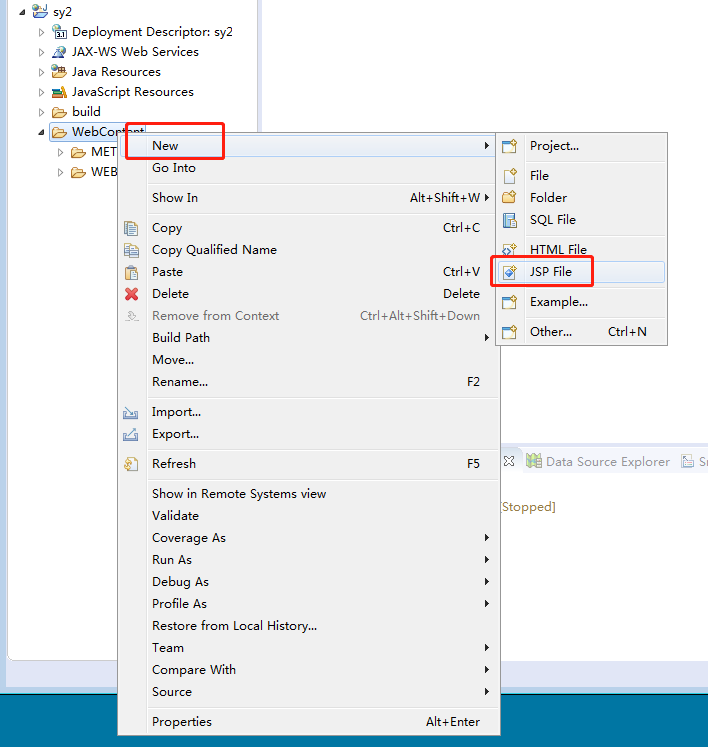


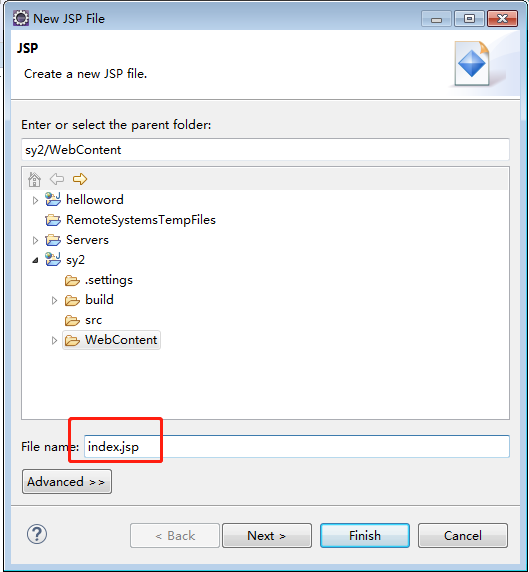




# 两个jsp页面之间数据传递

1. 新建index.jsp文件，如下图：





1. 打开index.jsp文件，字符集改为utf-8
2. 将下图所示的内容用如下代码替换

<br>

Request示例<br>

<br>

<form action="result.jsp" method="post" onsubmit="return check()">

<input id="numA" type="text" name="numA"/>

<select id="operC" name="operC">

<option value = "+">+</option>

<option value = "-">-</option>

<option value = "\*">\*</option>

<option value = "/">/</option>

</select>

<input id="numB" type="text" name="numB">

<br>

<br>

<input type="submit" name="submit" value="计算一">

</form>

1. 在index.jsp最后增加如下函数

<script language="JavaScript">

function check(){

var numA = document.getElementById("numA").value;

if (!numA){

alert("第一个数值不能为空");

document.getElementById("numA").focus();

return false;

}

if(isNaN(numA)){

alert("第一个请输入数值数据!");

document.getElementById("numA").value = "";

document.getElementById("numA").focus();

return false;

}

var numB = document.getElementById("numB").value;

if (!numB){

alert("第二个数值不能为空");

document.getElementById("numB").focus();

return false;

}

if(isNaN(numB)){

alert("第二个请输入数值数据!");

document.getElementById("numB").value = "";

document.getElementById("numB").focus();

return false;

}

}

</script>

1. index.jsp增加后的代码图示如下：



1. 在WebContent下新建result.jsp文件

1）修改字符集为“utf-8”，将<body> </body>的内容替换为如下

<body onload="setData()">

计算结果：

<input id="result" type="text" name="result" value=""/>

<br>

</body>

2）在result.jsp最后增加如下代码

<script language="JavaScript">

function setData(){

var numA = "<%=request.getParameter("numA")%>";

var numB = "<%=request.getParameter("numB")%>";

var oper = "<%=request.getParameter("operC")%>";

var result=0;

if (oper=="+"){

result = parseFloat(numA) + parseFloat(numB);

}

if (oper=="-"){

result = parseFloat(numA) - parseFloat(numB);

}

if (oper=="\*"){

result = parseFloat(numA) \* parseFloat(numB);

}

if (oper=="/"){

result = parseFloat(numA) / parseFloat(numB);

}

document.getElementById("result").value = result;

}

</script>

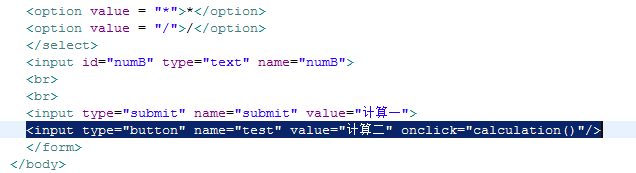
1. 查看运行结果

**另：可使用HTML+CSS+Javascript的方式实现功能**

**jsp页面与java文件数据传递（选做,不做要求）**

1. 在index.jsp文件下增加下图的代码：

<input type="button" name="test" value="计算二" onclick="calculation()"/>



1. 在index.jsp文件后面增加函数:

function calculation(){

var numA = document.getElementById("numA").value;

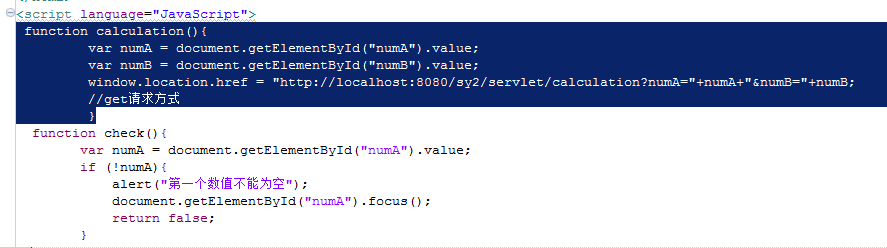
var numB = document.getElementById("numB").value;

window.location.href = "http://localhost:8080/sy2/servlet/calculation?numA="+numA+"&numB="+numB;

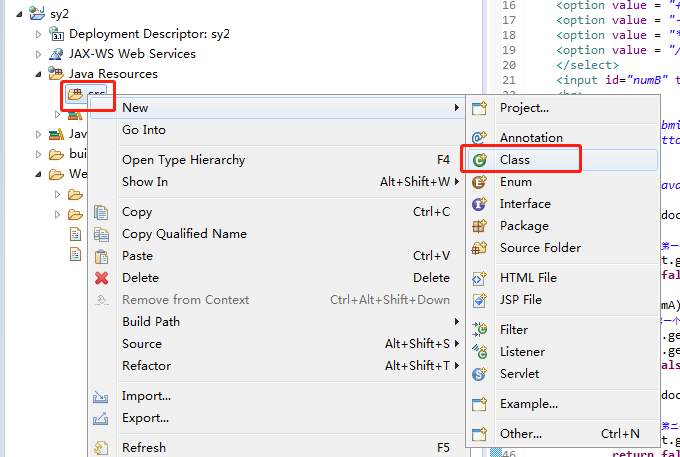
//get请求方式

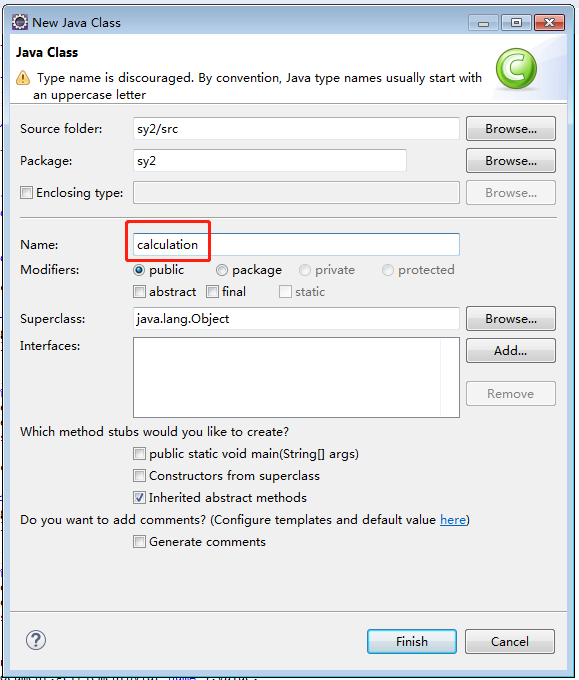
}

如下图：



1. 新建java文件，如下图：





按照如下代码替换整个文件

**package** sy2;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**public** **class** calculation **extends** HttpServlet{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** **void** doGet(HttpServletRequest request,HttpServletResponse response) **throws** IOException{

String numA = request.getParameter("numA");//接受前台请求的参数

String numB = request.getParameter("numB");

response.setContentType("text/html; charset=UTF-8");//构造前台相应输出内容

PrintWriter out = response.getWriter();

out.println("<html>");

out.println("<head><title>填写参数</title></head>");

out.println("<body>");

out.println("<center>");

//显示变量

out.println("<p>numA = " + numA + "</p>");

out.println("<p>numB= " + numB + "</p>");

out.println("</center>");

out.println("</body>");

out.println("</html>");

}

}

1. 修改WebContentt\WEB-INF\下的web.xml文件，增加如下代码

<servlet>

<servlet-name>calculation</servlet-name>

<servlet-class>sy2.calculation</servlet-class>

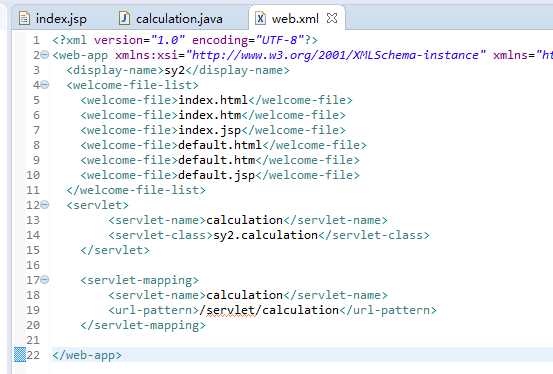
</servlet>

<servlet-mapping>

<servlet-name>calculation</servlet-name>

<url-pattern>/servlet/calculation</url-pattern>

</servlet-mapping>



1. 点击按钮“计算二”看输出结果