**Experiment3**

1. 编写JSP程序，实现图4-24所示的简易计算器。要求：输入“第一个参数”，选择运算类型（+，-，\*，/），输入 “第二个参数”后，按“计算”按钮，结果将显示在“结果”文本框中。

注意：程序需要对输入参数是否合法进行判断，例如参数是否为数字，除法时，除数不为0的判断。

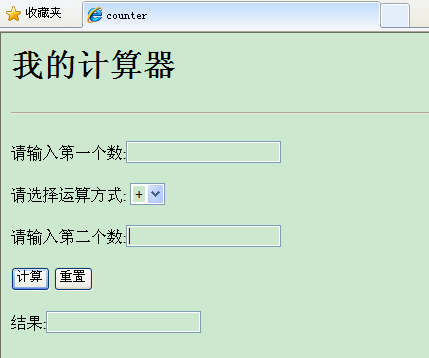
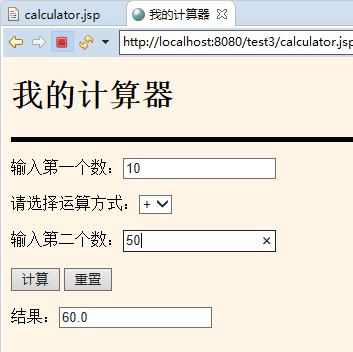


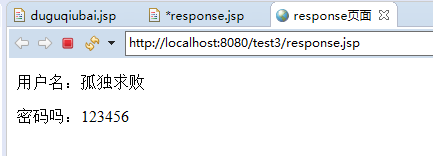
图4-24 简易计算器

**运行结果如下：（代码见calculator.jsp）**



2. 编写一个JSP页面，将用户名和密码存放到会话中(假设用户名为“孤独求败”，密码为“123456”)，再重新定向到另一个JSP页面，将会话中存放的用户名和密码显示出来。(提示：使用response对象的sendRedirect()方法进行重定向。)

**运行结果如下：（代码见duguqiubai.jsp和response.jsp）**



3. 编写一个JSP登录页面，可输入用户名和密码，提交请求到另一个JSP页面，该JSP页面获取请求的相关数据并显示出来。请求的相关数据包括用户输入的请求数据和请求本身的一些信息(比如请求使用的协议getProtocol() 、请求的URI request.getServletPath() 、请求方法request.getMethod() 、远程地址request.getRemoteAddr()等)。

**运行结果：（代码见loginsucess.jsp和registe.jsp）**

登录界面：



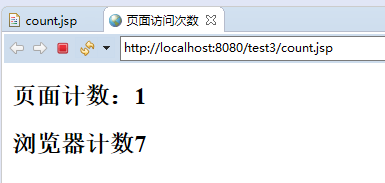
登录成功后的界面：

4. 利用隐式对象为某一网站编写一个JSP程序，统计该网站的访问次数。

一种情况是按照客户进行统计（按照浏览器进行统计，一个浏览器如果访问网站的话，就算一次访问，换句话说如果这个浏览器刷新多次网站的话，也算是一次访问）；

另一种情况是刷新一次页面，就算是一次访问。 要求用隐式对象去实现。

**运行结果：（代码见count.jsp）**



1. 页面跳转的两种方式(转发和重定向)区别及应用场景分析

<https://blog.csdn.net/liubin5620/article/details/79922692>

**分析：**

区别：**转发使用的是getRequestDispatcher()方法;重定向使用的是sendRedirect();**

1.forward方法只能将请求转发给同一个WEB应用中的组件；而sendRedirect 方法不仅可以重定向到当前应用程序中的其他资源，还可以重定向到同一个站点上的其他应用程序中的资源

2. 调用sendRedirect方法重定向的访问过程结束后，浏览器地址栏中显示的URL会发生改变成重定向的目标URL；而调用forward 方法的请求转发过程结束后，浏览器地址栏保持初始的URL地址不变。转发：浏览器URL的地址栏不变。重定向：浏览器URL的地址栏改变；

3. sendRedirect方法对浏览器的请求直接作出响应，响应的结果就是告诉浏览器去重新发出对另外一个URL的 访问请求，forward方法在服务器端内部将请求转发给另外一个资源，浏览器只知道发出了请求并得到了响应结果，并不知道在服务器程序内部发生了转发行为。转发是浏览器只做了一次访问请求。重定向是浏览器做了至少两次的访问请求；

4. forward方法的调用者与被调用者之间共享相同的request对象和response对象，它们属于同一个访问请求和响应过程；而sendRedirect方法调用者与被调用者使用各自的request对象和response对象，它们属于两个独立的访问请求和响应过程。转发2次跳转之间传输的信息不会丢失，重定向2次跳转之间传输的信息会丢失（request范围）。

5.无论是forward方法还是sendRedirect方法，在调用它们之前，都不能有内容已经被实际输出到了客户端。否则程序会运行异常。

应用场景：

转发是要比重定向快，因为重定向需要经过客户端，而转发没有。有时候，采用重定向会更好，若需要重定向到另外一个外部网站，则无法使用转发。另外，重定向还有一个应用场景：避免在用户重新加载页面时两次调用相同的动作。使用重定向不太方便的地方是，使用它无法将值轻松地传递给目标页面。而采用转发，则可以简单地将属性添加到Model,使得目标视图可以轻松访问。由于重定向经过客户端，所以Model中的一切都会在重定向时丢失。但幸运的是，在Spring3.1版本以后，我们可以通过Flash属性，解决重定向时传值丢失的问题。