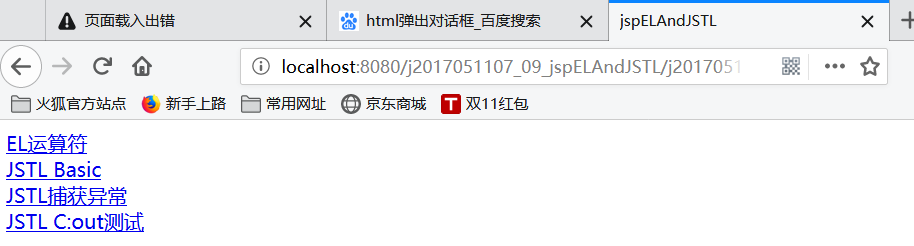
一、项目描述：

1. j2017051107\_09\_jspELAndJSTL：该项目主要实验EL和JSTL的基本用法和JSP页面之间值的传递

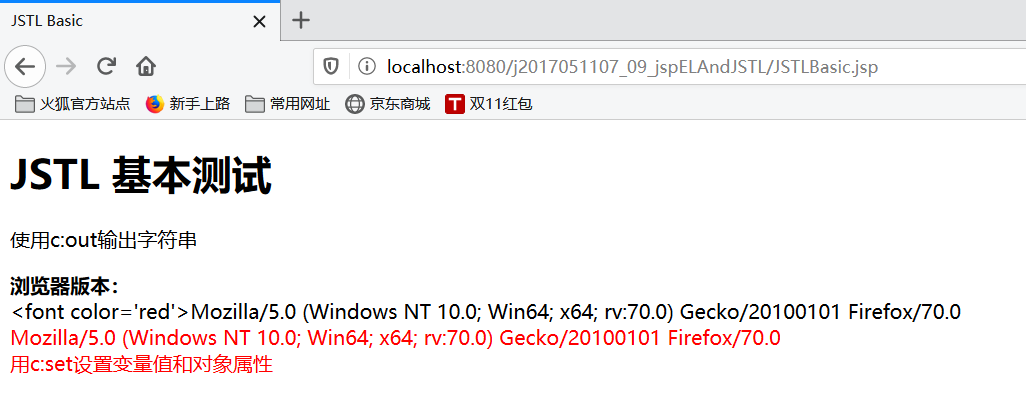
项目的导航页面：



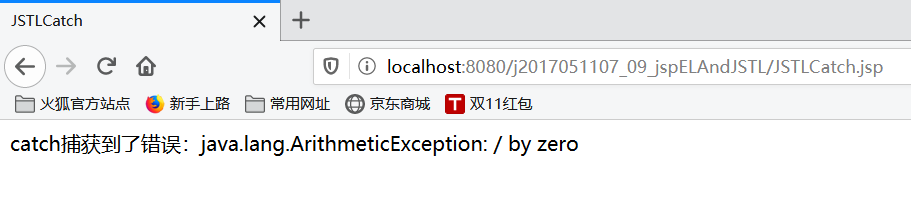
1. EL运算符通过对EL运算符的简单实验，展示EL的基本用法



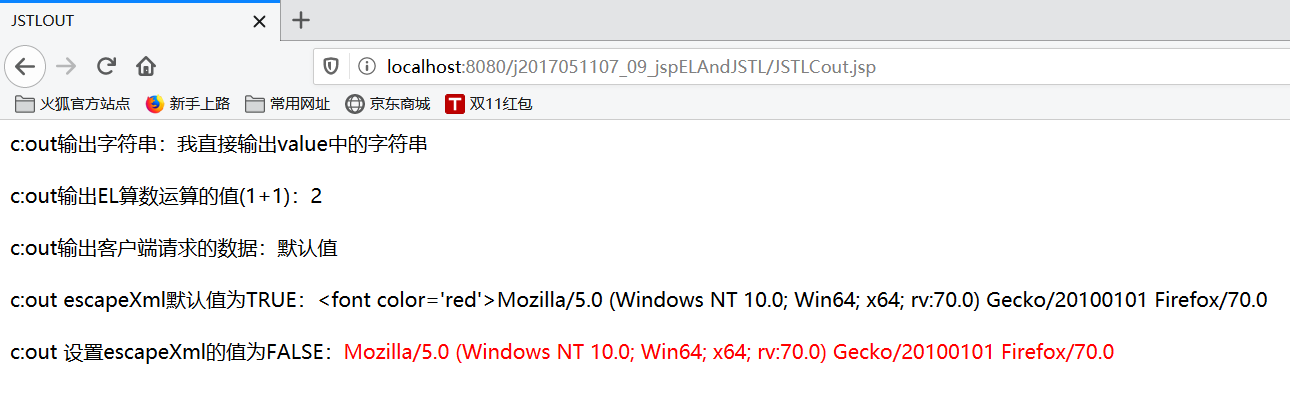
1. JSTL Basci通过c:out展示JSTL基本用法



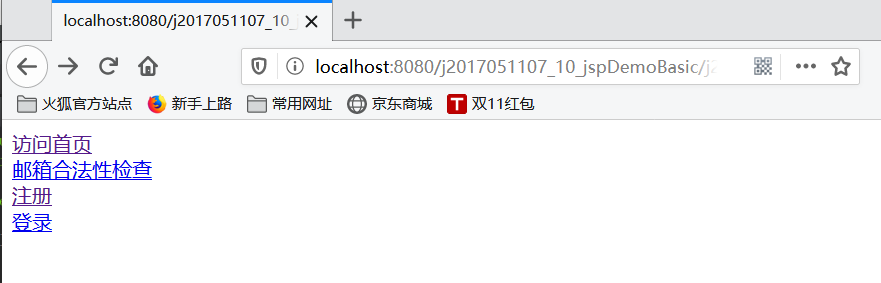
1. JSTL利用catch语句捕获异常



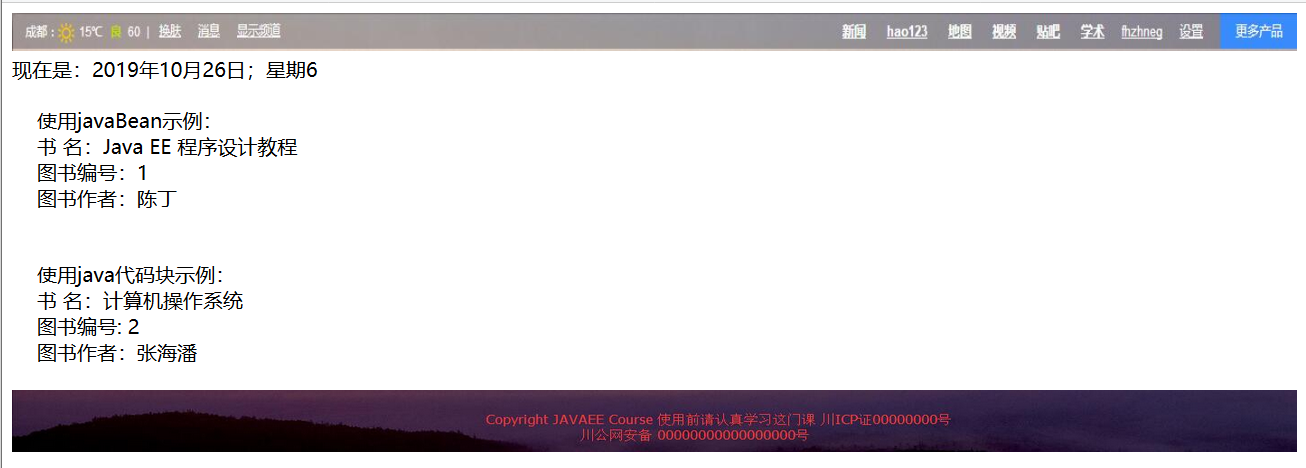
1. 使用JSTL的c:out输出各类型的值



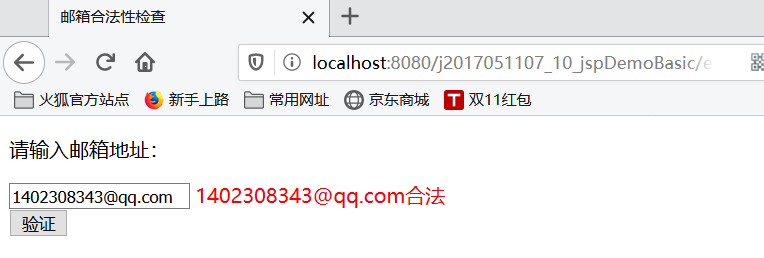
1. j2017051107\_10\_jspDemoBasic主要完成JSP页面模拟用户登录的实验，同时还伴有其他功能。
2. 项目导航页面

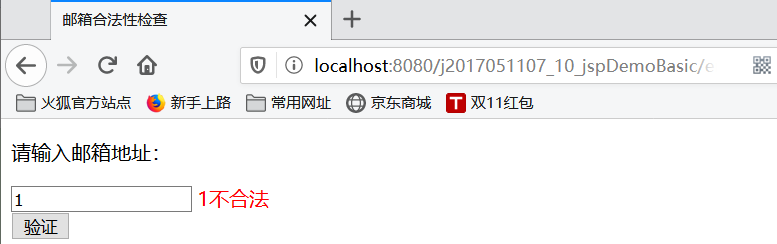


1. 首页：用户点击访问首页链接可以跳转到首页，首页展示使用javaBean的示例和直接使用java代码块的示例



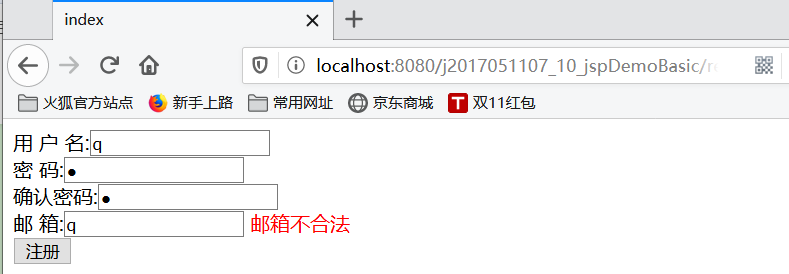
1. 邮箱合法性检查：点击邮箱合法性检查链接可以进入到检查页面。系统将会根据用户输入的邮箱来判断是否合法。如果邮箱合法，将会在输入框的后面提示邮箱合法。否则，将会提示邮箱不合法。



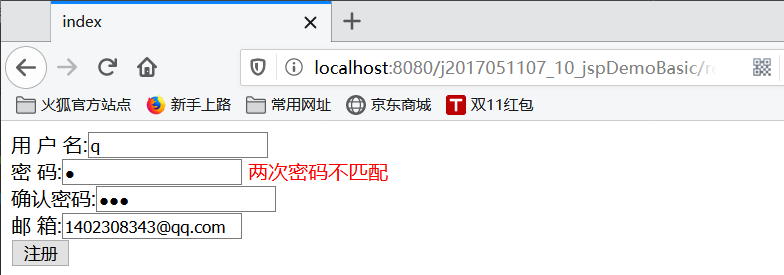


1. 注册：用户在注册页面进行注册。只有当用户的两次密码都正确，并且邮箱也合法的时候才能够注册成功，并且自动跳转到登录界面。如果当两次输入的密码不一致的时候，或者邮箱不合法的时候，系统将会提示错误，直到用户进行纠正所有错误后方可注册成功。

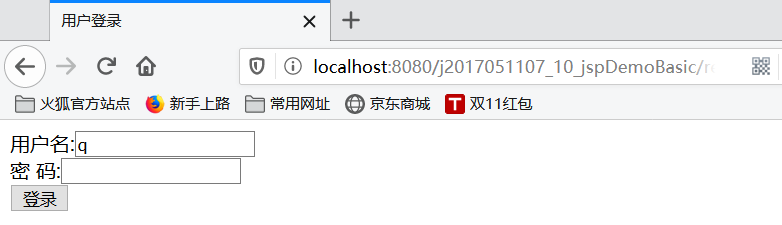
邮箱不合法：



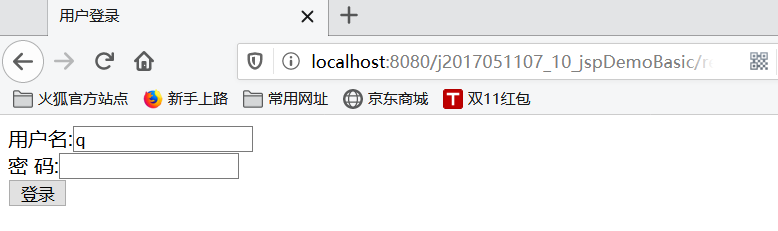
两次密码不一致：

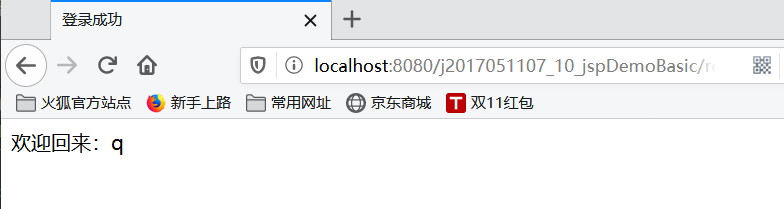


注册成功，跳转到登录页面：

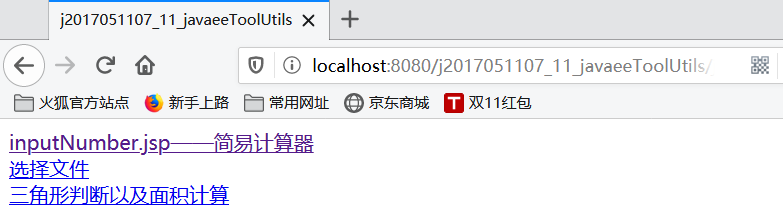


1. 登录：用户可以直接访问登录界面，也可以注册成功后自动跳转到登录界面。登录成功后，将会跳转到欢迎界面。

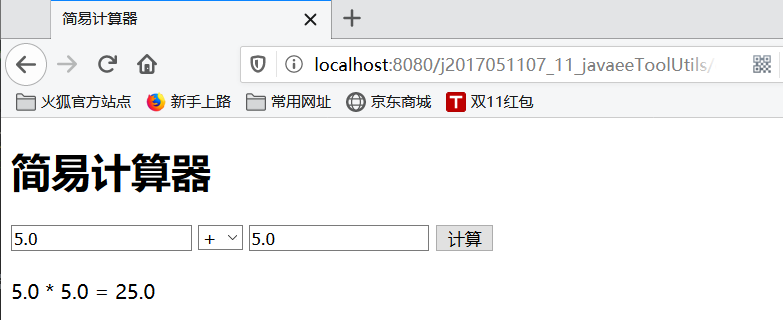




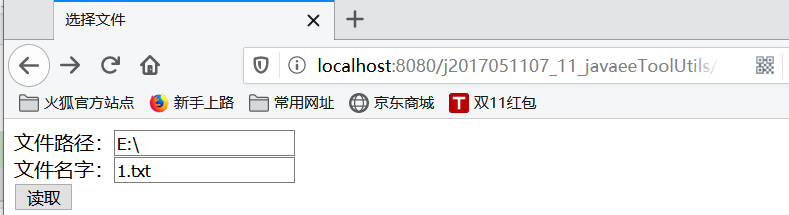
1. j2017051107\_11\_javaeeToolUtils：该项目主要是完成Bean实验，用Bean的方式，完成各种功能，然后在JSP中应用。项目中有三种Bean，分别是Computer、MyFile和Triangle。这三种Bean分别用于进行简单的四则运算、文件操作和判断给定的边能否构成三角形以及面积的计算。
2. 项目导航页面：

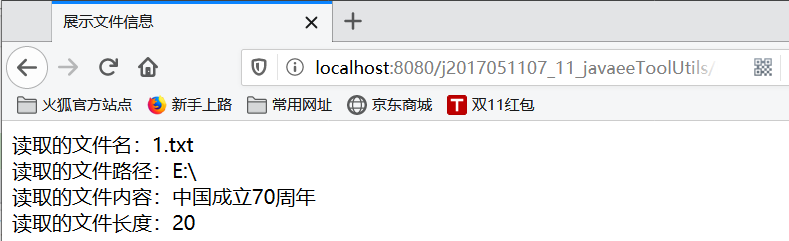


1. 简易计算器



1. 文件操作：用户在输入框中输入文件的路径和名字，然后点击读取按钮即可跳转到文件信息界面。在文件信息界面显示了读取的文件的名字、文件的路径、文件的内容以及文件内容的长度。

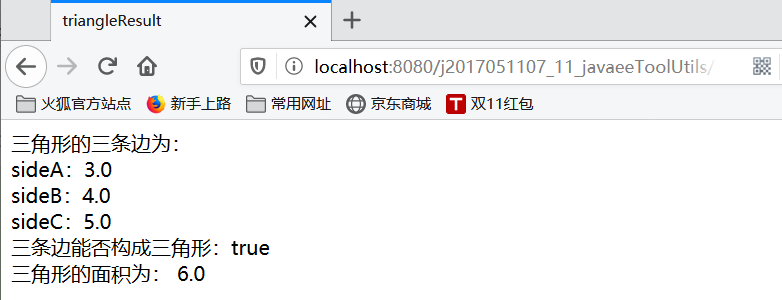




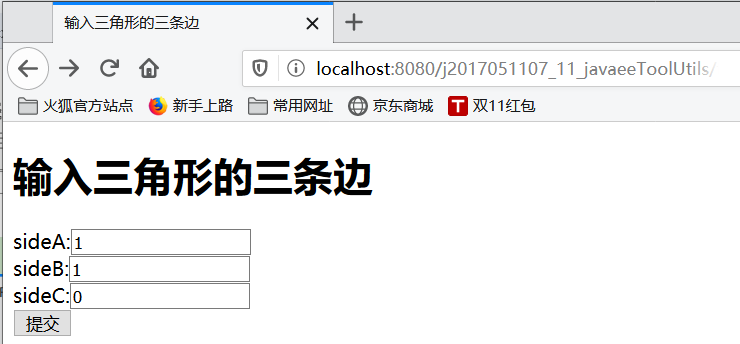
1. 三角形判断以及面积计算

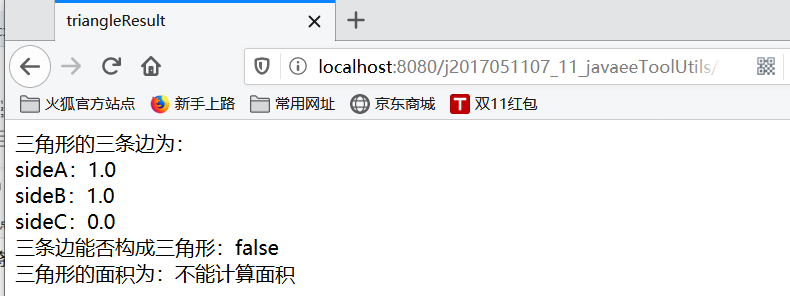
能构成三角形：





不能构成三角形：





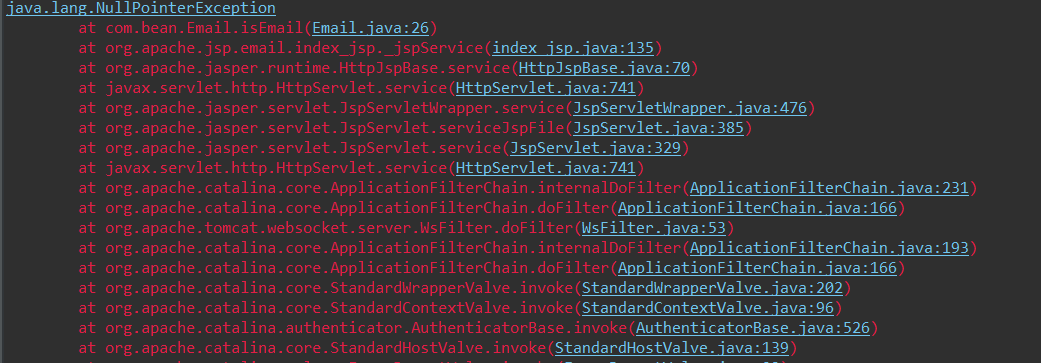
错误1：出现404



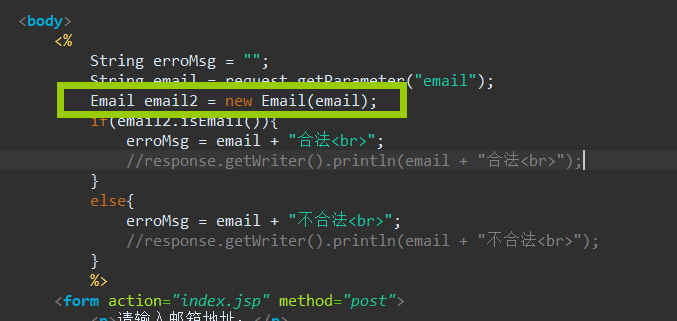
错误原因：在进行javaeeToolUtils中的计算器的实验时，在ComputerServlet中我想将处理的结果返回到inputNumber.jsp直接进行显示，并可继续进行其他计算。而不是按照老师的方法重新写个界面进行显示。在ComputerServlet处理完后，我刚开始也采取转发的方法，第一次结果是能正常显示，但是继续进行计算的时候就会报404的错误。原因就是采用转发的方法URL是不变的，所以我再次进行计算的时候，其实当前URL是指向ComputerServlet的，而不是inputNumber.jsp。

解决方案：采用重定向的方法，在ComputerServlet处理完后重定向到inputNumber.jsp，便既可以显示结果，又可以多次进行计算。

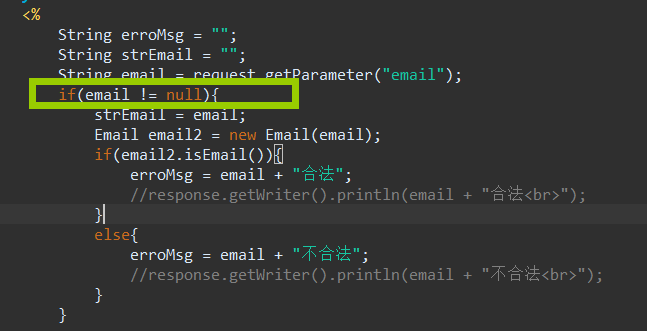
错误2：空指针错误



错误原因：我将检查和检查结果放在同一个界面，在第一次进入进行邮箱检查的时候，邮箱其实是null，这时候创建对象当然会出错。



解决方案：添加判断邮箱不为空的条件语句，只有当邮箱不为空的时候，才创建对象。如图所示：



错误3：年份和月份显示错误

错误原因：

1.date.getYear()返回的是从1900年到当前年份的整数，所以获取当前年份还需要加上1900

2.Date.getMonth()返回的月份是从0~11的，所以获取到以后还要加1

解决方案：

1.获取当前年份的时候使用：date.getYear()+1900还需要加上1900

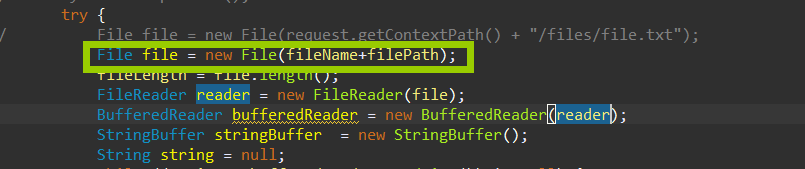
2.获取月份的时候使用：Date.getMonth()+1

错误4：读取文件时路径出错



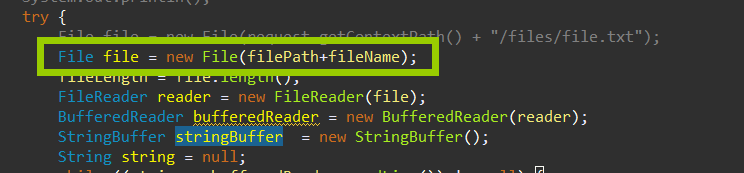
错误原因：

在创建路径的时候，将文件的目录和名称放反了

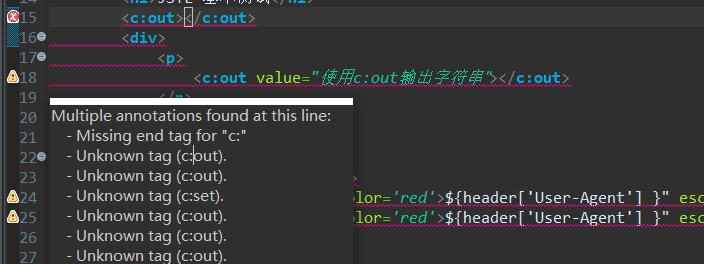


解决方案：

改正名称和目录的顺序



错误5：使用JSTL核心标签库的时候系统不能识别



错误原因：没有在项目的WEB-INF目录下的lib目录中导入两个包，jstl.jar和standerd.jar以及在需要用到JSTL的JSP页面中做声明：<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core%>

解决方案：导入两个包到相应的位置，然后在需要用到JSTL的JSP页面中做声明

心得体会：

1. 自己有些粗心大意，然后在逻辑周密性方面也不够严谨。就比如说错误2和错误4。对于错误2出现的空指针错误，空指针其实是会经常出现的，这就需要在可能出现空指针的地方多留心，然后做相应的判断。还有就是错误4，居然会把路径和文件名反着拼凑，我真不知道当时脑子里面是怎么想的，居然还会犯这么低级的错误。希望以后自己能及时改正，不然这些不必要的错误将会花费很多不必要的时间去调试和检查错误。
2. EL基础知识：
   1. 语法：${EL表达式}
   2. “.”与 [ ] 运算符：EL支持这两种方式获取数据
   3. EL运算符：四种运算符
      1. 算数运算符
      2. 关系运算符
         1. 格式："${x 运算符 y }"
         2. 运算符可以为符号，也可以为英文运算符
         3. 返回值为true或false
      3. 逻辑运算符
      4. Empty运算符
3. 格式：${empty 字符串 }"
4. 返回值为true或false

Empty运算符

1. JSTL基础知识：
2. 使用前提：
3. 要使用JSTL，必须要在WEB-INF目录下的lib目录中导入两个包，jstl.jar和standerd.jar
4. 导入两个包以后，还需要在需要用到JSTL的JSP页面中做声明：

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core%>

1. 五大类标签函数库
2. 核心标签库
3. I18N格式标签库
4. SQL标签库
5. XML标签库
6. 函数标签库
7. 核心标签库小结

在本次的实验中，我只是用到了核心标签库，所以在此处仅针对核心标签库做一个小结。其他标签库暂时没有用到，但是相信再后来的项目中会遇到的，到时候遇到了再有针对性地进行系统总结。

1. <c:out>标签：
2. 作用：在页面进行输出
3. 语法：<c:out value=”param” default=”default” escapeXml=”boolean”>
4. value属性：在页面输出的内容即为属性value的值，value的值可以是单纯的基本数据对象，也可以嵌入EL表达式
5. default属性：他的值为value的默认值
6. escapeXml属性：意思是是否过滤为xml文档。

如果为escapeXml="false"，则将value中的html、xml解析出来

如果为escapeXml="true"，则将value中的内容以纯字符串的形式输出

1. <c:set>标签：
   1. 作用：将变量储存至JSP范围中或是JavaBean的属性中
   2. 语法：<c:set var="varName" value="value" [scope="{ page|request|session|application }"]/>
2. var属性：定义一个变量，变量的名字就是varName
3. value属性：变量的值
4. scope属性：变量的范围，为四个范围之一，默认为page
5. <c:if>标签：
   1. 作用：逻辑判断，同if
   2. 语法：<c:if test="testCondition" [var="varName"] [scope="{page|request|session|application}"]> </c:if>
6. test属性：他的值为表达式的结果，如果结果为true，则执行，否则不执行
7. var属性：用来存储test运算后的boolean值
8. scope属性：变量的范围，为四个范围之一，默认为page
9. <c:catch>标签：
   1. 作用：主要用来处理产生错误的异常状况，并且将错误信息储存起来
   2. 语法：<c:catch var="<string>"></c:catch>
10. var属性：用来存储错误信息
11. javaBean基础知识:
12. javaBean规范：
    1. 实现可序列化接口
    2. public类型的无参构造函数
    3. 属性应为private类型，并为属性设置public类型的set和get方法
13. javaBean使用：
14. <jsp:useBean>
    1. 功能：创建或者获得一个对象。
    2. 语法：<jsp:useBean id="" class="" scope=""></jsp:useBean>
15. id属性：创建JavaBean对象名
16. class属性：JavaBean的完全类名
17. scope属性：变量的范围，为四个范围之一，默认为page
18. <jsp:setProperty>
19. 功能：设置JavaBean中属性的属性值
20. 语法1：<jsp:setProperty name=“” property=“” value=“”/>
21. name属性：JavaBean名称
22. property属性：属性名称
23. Value属性：设置属性的值
24. 语法2：<jsp:setProperty name=“” property=“\*”/>

所有名字和name属性匹配的请求参数都将被传递给相应的属性set方法

1. <jsp:getProperty>
   1. 作用：将javaBean的属性的值输出到页面
   2. 语法：<jsp:getProperty name=“” property=“”/>
2. name属性：指定javaBean的名称
3. property属性：指定要获取的属性的名称