

# 浙江工业大学

## Web 应用开发上机作业

2022 级



实验题目： 处理请求和发送响应的应用练习

学生姓名： 李飞飞

学生学号： 2022

任课教师： 赵 小 敏

专业班级： 2022 软件工程（移动应用开发 1）

学 院： 计算机科学与技术学院

2024 年 3 月 17 日

# 目 录

1、算三角形面积.....	6
1.1 题目要求: .....	6
1.2 需求分析 .....	6
1.2.1 功能描述.....	6
1.2.2 业务处理流程及要求.....	6
1.2.3 输入信息.....	6
1.2.4 输出信息.....	6
1.3 开发步骤.....	6
1.4 运行结果.....	6
1.5 问题及解决方案.....	9
1.6 实验收获.....	9
1.7 源代码.....	9
2. Excel 导出工具 .....	12
2.1 题目要求 .....	12
2.2 需求分析 .....	12
3.2.1 功能描述.....	12
3.2.2 业务处理流程及要求.....	12
3.2.3 输入信息.....	12
3.2.4 输出信息.....	12
2.3 开发步骤 .....	12
2.4 运行结果 .....	12
2.5 问题及解决方案 .....	14
2.6 实验收获 .....	14
2.7 源代码 .....	15
3. 简单的企业会员注册和登录功能.....	20
3.1 题目要求 .....	20
3.2 需求分析 .....	20

2.2.1 功能描述.....	20
2.2.2 业务处理流程及要求.....	21
2.2.3 输入信息.....	21
2.2.4 输出信息.....	21
3.3 开发步骤.....	21
3.4 运行结果.....	22
3.5 问题及解决方案.....	26
3.6 实验收获.....	26
3.7 源代码.....	27

注释：三道题写在一个 **Maven** 工程里，启动后在首页根据链接跳转不同页面。



第二题用了 **Vue.js** 和 **element Ui**，所以第一次加载网络不好的话需要等他加载一下，并且写法和要求不一样，但是已经提前和老师说过了，老师说可以的。第二第三题分别在 **Pom.xml** 里引了不同的依赖。代码包括程序流程图打包放在单独传了。以下是文件结构，不同颜色代表不同题目：



# 1、算三角形面积

## 1.1 题目要求:

用 html 和 servlet 编程实现输入三角形的三个边长并计算和输出三角形的面积

(1) 编写一个 input.html 页面，页面中包括输入三个边长和提交按钮的表单。

(2) 编写一个文件名为 TriangleServlet.java 的 Servlet，其 URL 为 /computeTriangleArea.do，响应来自 input.html 的请求。如果能够构成三角形，则将计算得到的面积构成字符串“三角形面积=XXX”并在页面输出，要求面积保留 2 位小数，否则输出“三条边长无法构成三角形”，如果输入的边长为非数字型或不是正数，则在页面输出“输入的边长有误!”

## 1.2 需求分析

根据输入给出面积，同时判断非法情况，使用 servlet 技术

### 1.2.1 功能描述

算面积小工具

### 1.2.2 业务处理流程及要求

输入——确认——给出结果/报错非法

### 1.2.3 输入信息

三边长度

### 1.2.4 输出信息

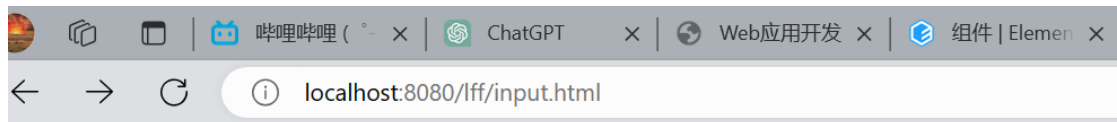
面积/非法提示

## 1.3 开发步骤

写页码再写 servlet

## 1.4 运行结果

1. 一般情况



这是实验一，请输入三角形三边

三角形判断

边a: 3

边b: 4

边c: 5

确认

确认

取消



三角形面积=6.00



这是实验一，请输入三角形三边

三角形判断

边a: 3.4

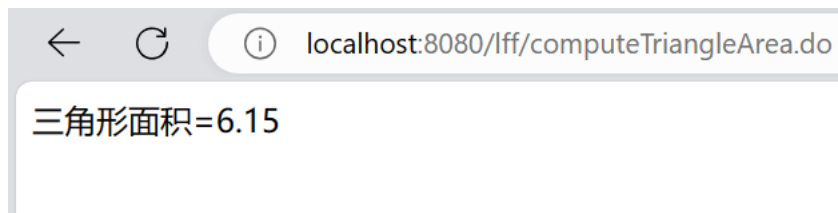
边b: 4.1

边c: 5

确认

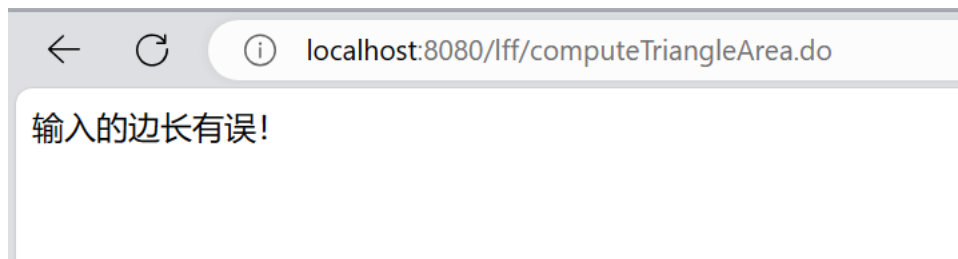
确认

取消



## 2. 错误情况

A screenshot of a web form titled "这是实验一，请输入三角形三边" (This is Experiment 1, please enter the three sides of a triangle). Below the title is a section labeled "三角形判断" (Triangle Judgment). The form contains three input fields for the sides: "边a:" (Side a) with the value "-3.4", "边b:" (Side b) with the value "4.1", and "边c:" (Side c) with the value "5". At the bottom of the form are three buttons: "确认" (Confirm), "确认" (Confirm), and "取消" (Cancel).





← → ↻ ⓘ localhost:8080/lff/input.html

这是实验一，请输入三角形三边

三角形判断

边a:

边b:

边c:

← ↻ ⓘ localhost:8080/lff/computeTriangleArea.do

输入的边长有误!

## 1.5 问题及解决方案

1. 算三角形面积：海伦公式

## 1.6 实验收获

没收获

## 1.7 源代码

1. Input.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <a>这是实验一，请输入三角形三边</a>
  <form action="computeTriangleArea.do" method="post">
    <fieldset>
      <legend>三角形判断</legend>
```

```

        <p>
            <label>边 a: <input type="text" name="a"
/></label>
        </p>
        <p>
            <label>边 b: <input type="text" name="b"
/></label>
        </p>
        <p>
            <label>边 c: <input type="text" name="c"
/></label>
        </p>
        <p>
            <label>
                <input type="submit" value="确认" />
                <input type="reset" value="取消" />
            </label>
        </p>
    </fieldset>
</>
</body>
</html>

```

## 2. inputServlet

```

@WebServlet(name = "computeTriangleArea", value =
"/computeTriangleArea.do")
public class inputServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
        IOException {

    }

    @Override
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
        IOException {
        String a=request.getParameter("a");
        String b=request.getParameter("b");
        String c=request.getParameter("c");
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
    }
}

```

```

//html 格式
out.println("<!DOCTYPE html>");
out.println("<html><body>");
//判断
Number aa,bb,cc;
NumberFormat format=NumberFormat.getInstance();
    try {
        aa= format.parse(a);
        bb=format.parse(b);
        cc=format.parse(c);

if(aa.doubleValue()<0||bb.doubleValue()<0||cc.doubleValue()<0)
{
            out.println("输入的边长有误!");
            out.println("</body></html>");
            return;
        }
        double
temp=aa.doubleValue()+bb.doubleValue()+cc.doubleValue();
        double
ma=Math.max(aa.doubleValue(),Math.max(bb.doubleValue(),cc.doubleValue()));
        double mi2=temp-ma;
        if(ma>mi2){
            out.println("三条边长无法构成三角形");
            out.println("</body></html>");
            return;
        }
        double s=temp/2;
        DecimalFormat df = new DecimalFormat("#.00");
        out.println("三角形面积="+
            df.format(Math.sqrt(s*(s-
aa.doubleValue())*(s-bb.doubleValue())*(s-
cc.doubleValue()))));
            out.println("</body></html>");
        } catch (ParseException e) {
            out.println("输入的边长有误!");
            out.println("</body></html>");
            throw new RuntimeException(e);
        }
    }
}

```

## 2.Excel 导出工具

### 2.1 题目要求

假设某成绩系统导出 Java 程序设计的的成绩为 JSON 格式,保存为 c:/temp/javascore.json, 请你编写文件解析的 web 应用程序, 将该文件通过 uploadfile.html 上传, 由映射地址为 parsejson.do 的 ParseJsonServlet 解析并以表格方式在页面进行显示, 表格下方有超链接“导出 excel”可将解析的数据导出为 excel 文件

### 2.2 需求分析

用户上传成绩 json 文件, 实现可视化选择并导出

#### 3.2.1 功能描述

用户上传成绩 json 文件, 实现可视化选择并导出

#### 3.2.2 业务处理流程及要求

用户上传成绩 json 文件——选择 t 条目——导出 Excel

#### 3.2.3 输入信息

Json 文件

#### 3.2.4 输出信息

Excel

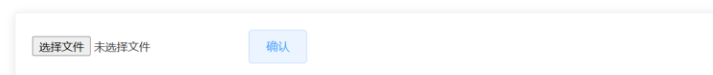
### 2.3 开发步骤

写 html,添加解析 json 功能。写 servlet 导出 Excel,servlet 用 fastjson 解析 html 传的 json,解析到 Student 数组中

### 2.4 运行结果

注: 因为动态引入了 vue 和 elementui 所以刚打开要等他加载一下

#### 1. 导入页面



## 2. 导入后

选择文件 javascore.json

确认

## 3. 列表页面与勾选

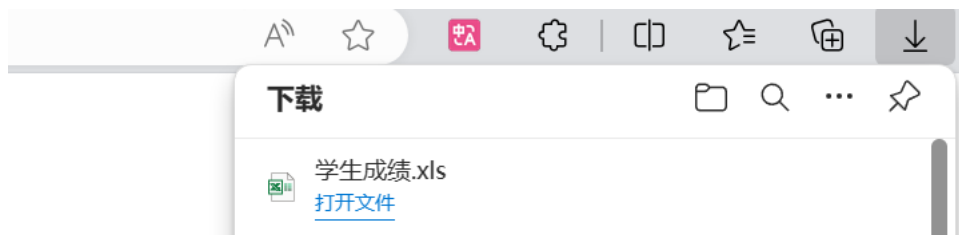
取消选择可以一次性全部取消勾选

localhost:8080/fff/uploadFile.html

<input type="checkbox"/>	学号	姓名	课程名字	分数
<input type="checkbox"/>	201826630601	王明	java程序设计	90
<input type="checkbox"/>	201826630602	张三	java程序设计	80
<input type="checkbox"/>	201826630604	王五	java程序设计	85

<input checked="" type="checkbox"/>	学号	姓名	课程名字	分数
<input checked="" type="checkbox"/>	201826630601	王明	java程序设计	90
<input type="checkbox"/>	201826630602	张三	java程序设计	80
<input checked="" type="checkbox"/>	201826630604	王五	java程序设计	85

## 4. 导出 Excel 结果



学号折叠是为 Excel 会自动改为科学计数法

A1	学号				
	A	B	C	D	E
1	学号	姓名	课程	分数	
2	2.02E+11	王明	java程序设	90	
3	2.02E+11	王五	java程序设	85	
4					
5					

## 2.5 问题及解决方案

没问题

## 2.6 实验收获

- 1、学会了用原生 servlet 导出 excel

## 2.7 源代码

### 1. UploadFiles.html

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <style>
    .card{
      margin-top: 10%;
      margin-left: 25%;
      width: 50%;
    }
  </style>
</head>

<body>
<div id="app">
  <el-card v-if="showing" class="card">
    <input type="file" @change="handleFileChange">
    <el-button type="primary" @click="parseFile" plain>确认
  </el-button>
  </el-card>
  <el-card v-else class="card">
    <el-table
      ref="multipleTable"
      :data="jsonData"
      style="width: 100%"
      @selection-change="handleSelectionChange">
      <!-- 选择 -->
      <el-table-column
        type="selection"
        width="55">
      </el-table-column>
      <el-table-column
        prop="stuid"
        label="学号"
        width="180">
      </el-table-column>
      <el-table-column
        prop="name"
        label="姓名"
        width="180">
```

```

        </el-table-column>
        <el-table-column
            prop="courseName"
            label="课程名字">
        </el-table-column>
        <el-table-column
            prop="score"
            label="分数">
        </el-table-column>
    </el-table>

    <el-divider></el-divider>
    <el-button @click="toggleSelection()">取消选择</el-button>
    <el-button type="primary" @click="exportData">确认</el-button>

    <form id="exportForm" action="export.do" method="post"
style="display: none;">
        <input type="hidden" name="jsonData">
    </form>
</el-card>
</div>
</body>
<!-- import Vue before Element -->
<script
src="https://cdn.jsdelivr.net/npm/vue@2.6.14/dist/vue.js"></script>
<!-- 引入样式 -->
<link rel="stylesheet"
href="https://unpkg.com/element-ui/lib/theme-chalk/index.css">
<!-- 引入组件库 -->
<script src="https://unpkg.com/element-ui/lib/index.js"></script>
<script>
    new Vue({
        el: '#app',
        data: function () {
            return {
                showing:true,
                jsonData:null,
                multipleSelection: [],
            }
        },
        methods: {
            //选中文件
            handleFileChange(event) {
                this.selectedFile = event.target.files[0];
            }
        }
    })

```



```

    },
    //解析文件
    parseFile() {
        if (this.selectedFile) {
            const reader = new FileReader();
            reader.onload = () => {
                try {
                    this.jsonData = JSON.parse(reader.result);
                } catch (error) {
                    console.error('解析 JSON 文件出错:',
error);
                }
            };
            reader.readAsText(this.selectedFile);
            this.showing=false;
        } else {
            console.error('请选择一个文件');
        }
    },
    //改变选中状态
    handleSelectionChange(val) {
        this.multipleSelection = val;
    },
    //取消选择
    toggleSelection(rows) {
        this.$refs.multipleTable.clearSelection();
    },
    //导出数据
    exportData() {
        if (this.multipleSelection.length > 0) {
            const jsonDataToSend =
JSON.stringify(this.multipleSelection);
            // 将数据设置到隐藏的表单元素中
            document.getElementById('exportForm').querySelector('input[name="jsonData"]').value = jsonDataToSend;
            // 提交表单
            document.getElementById('exportForm').submit();
        } else {
            console.error('请选择要导出的数据!');
        }
    }
}
})

```

```
</script>
```

```
</html>
```

## 2. ExportScoreServlet.java

用了 fastjson

```
<!--      json-->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>2.0.25</version>
</dependency>
```

```
@WebServlet(name = "ExportScoreServlet", value = "/export.do")
public class ExportScoreServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
    }

    @Override
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        //指定编码
        response.setHeader("Content-Encoding", "gb2312");
        //指示浏览器以附件的形式下载响应体内容 并指定文件编码
        response.setHeader("Content-Disposition",
            "attachment; filename=" + java.net.URLEncoder.encode("
学生成绩.xls", StandardCharsets.UTF_8));
        //设置响应体格式格式

        response.setContentType("application/vnd.ms-excel;charset=gb2312");

        //获取数据
        String data=request.getParameter("jsonData");
        Student[] students = JSON.parseObject(data, Student[].class);
        PrintWriter out = response.getWriter();

        out.println("学号\t姓名\t课程\t分数\t");
    }
}
```

```

        for (Student student:students) {
            out.print(student.getStuid()+"\t");
            out.print(student.getName()+"\t");
            out.print(student.getCourseName()+"\t");
            out.print(student.getScore()+"\t");
            out.println("");
        }
        System.out.println(data);
    }
}

```

### 3. Student 类

用了 Lombok

```

39
40      <!--      Lombok-->
41      <dependency>
42          <groupId>org.projectlombok</groupId>
43          <artifactId>lombok</artifactId>
44          <version>1.18.24</version>
45          <scope>provided</scope>
46      </dependency>
47

```

```

package zjut.edu.homework2.utils;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

/**
 * @ Author      : Li Feifei
 * @ Date        : Created in 3:02 2024/3/16
 * @ Description: 学生类
 */
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Student {
    private String stuid;
    private String name;
    private String courseName;
    private Integer score;
}

```

## 3.简单的企业会员注册和登录功能

### 3.1 题目要求

(1) 注册页面 `register.html` 包括登录名、密码、姓名、性别、邮箱、手机号等信息，要求对所有字段进行非空判断，密码为数字、字母和特殊字符的组合（不少于 8 位），对邮箱和手机号进行有效性判断，否则弹出对话框进行提示。注册信息处理由映射地址为 `register.do` 的 `RegisterServlet.java` 进行处理，将所有用户注册的登录名、密码、姓名、性别、邮箱、手机号等信息按行保存到 `userinfo.txt` 中，每个字段用“|”分隔，要求将密码以国密 SM3 加密（网上查找或者参考学习通上的资料代码）后写入文件

如果当前注册的用户名不与已注册的用户名冲突，则提示“注册成功”，5 秒后自动跳转或有超链接至登录页面。

(2) 登录页面 `login.html`，页面风格如下

用户登录页面需判断用户名或密码是否为空，登录成功与否由映射地址为 `login.do` 的 `LoginServlet.java` 处理，该 `Servlet` 从 `userinfo.txt` 查询判断用户名与密码是否一致，如果一致则跳转至欢迎页面 `welcome.html` 提示“登录成功！”，否则跳转至失败页面 `failed.html` 提示“用户名或密码错误，请重新登录”，其中“重新登录”为超链接，可以跳转至登录页面 `login.html`

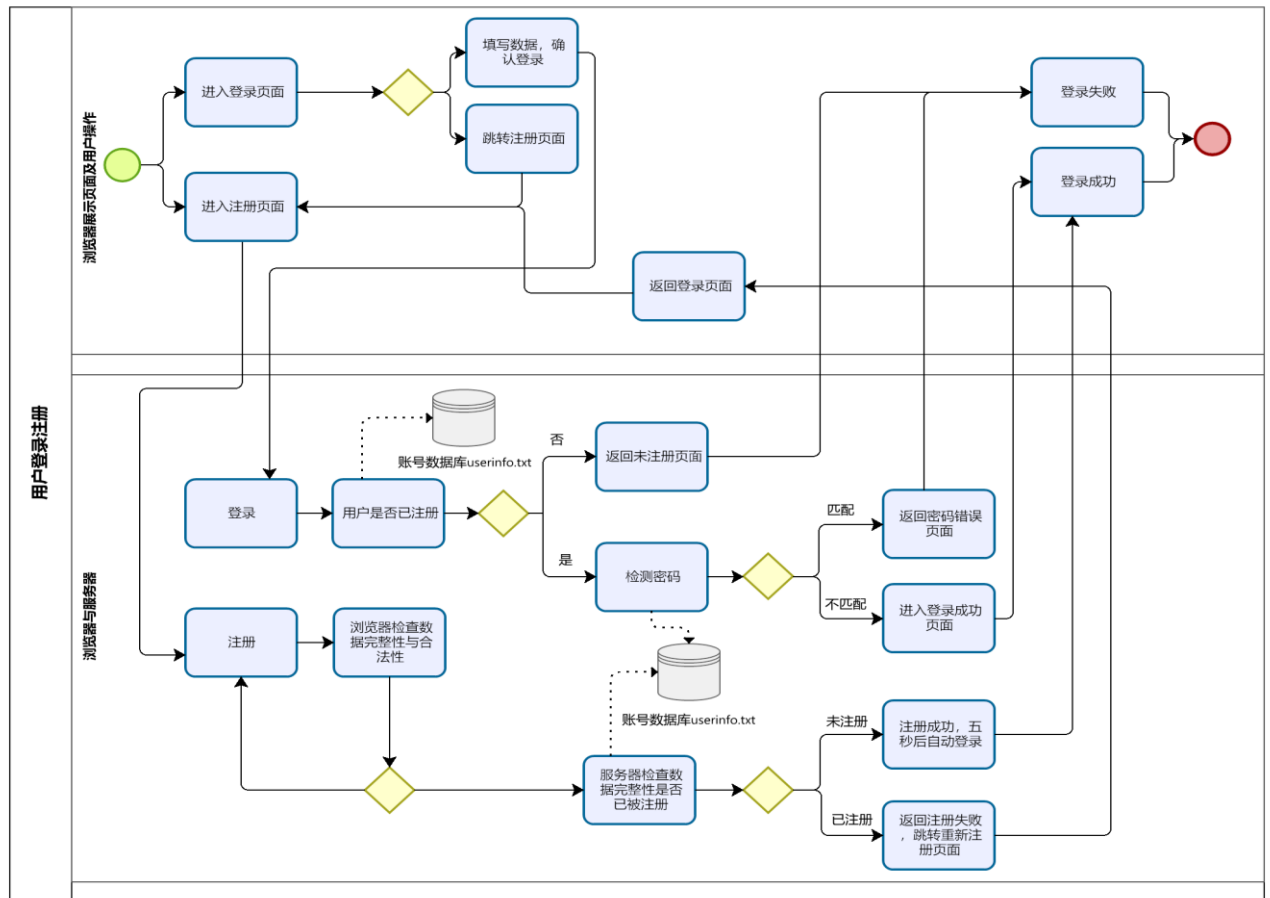
### 3.2 需求分析

#### 2.2.1 功能描述

写录注册页面，并持久化用户数据，不成功时给出相应错误

## 2.2.2 业务处理流程及要求

### 程序流程图



## 2.2.3 输入信息

账号信息

## 2.2.4 输出信息

登录注册结果

## 3.3 开发步骤

写 html,再写 servlet,写 User 类再写 User 工具类用于保存和读取数据就做完了

### 3.4 运行结果

- 登录页面

### 企业会员登录

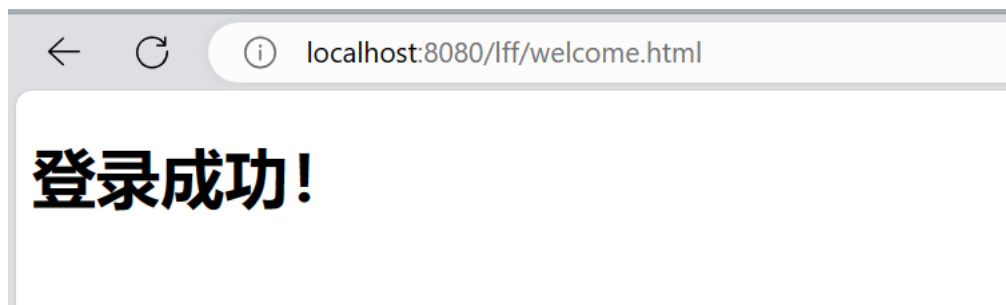
用 户 名

密 码

[立即登录](#)

没有帐号? [免费注册](#)

- 登录成功



- 登录失败



- 注册页面

localhost:8080/f/f/register.html

用户注册

用户名

1122

密码

111

姓名

飞飞

性别

☒男☐女

邮箱

11@

手机号

12345678910

立即注册

- 注册信息有误

f/register.html

localhost:8080 显示

姓名不能为空!

确定

用户注册

用户名

1122

密码

111

姓名

请输入姓名

性别

☒男☐女

邮箱

11@

手机号

12345678910

立即注册

localhost:8080 显示

请输入正确的邮箱!

确定

## 用户注册

用 户 名 1122

密 码 111

姓 名 飞飞

性 别 ☒ 男 ☐ 女

邮 箱 11

手 机 号 12345678910

立即注册



localhost:8080 显示

手机号要求11位!

确定

## 用户注册

用 户 名 1122

密 码 111

姓 名 飞飞

性 别 ☒ 男 ☐ 女

邮 箱 11@

手 机 号 12345

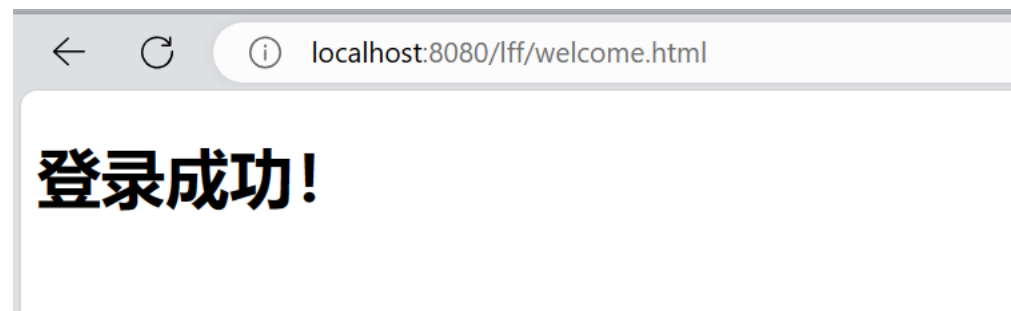
立即注册

- 昵称已占用

← ↻ ⓘ localhost:8080/lff/registerServlet.do

用户名已存在，请重新输入。 [点击返回](#)

- 注册成功，等待五秒跳转登录



- 注册后数据

### 3.5 问题及解决方案

没啥问题

### 3.6 实验收获

1. 了解了一直加密技术
2. 了解了 JavaWeb 文件操作与文件结构

### 3.7 源代码

部分代码和前面作用很像就不放了，放在附件里，这里只放重要代码  
集成了依赖于加密

```
47
48      <!--      加密-->
49      <dependency>
50      <groupId>org.bouncycastle</groupId>
51      <artifactId>bcprov-jdk15on</artifactId>
52      <version>1.68</version>
53      </dependency>
54
```

- 登录 login.html

```
75      </table>
76  </form>
77  <script language="JavaScript" type="text/javascript">
78      function custCheck() {
79          var userName = document.getElementById("userName");
80          var password = document.getElementById("password");
81          if (userName.value === "" || userName.value === "请输入用户名") {
82              alert("用户名不能为空!");
83              return false;
84          } else if (password.value === "" || password.value === "请输入密码") {
85              alert("密码不能为空!");
86              return false;
87          }
88          // var reg = new RegExp("[\\u4E00-\\u9FFF]+", "g");
89          // if (!reg.test(memberName.value)) {
90          //     alert("会员名必须为中文");
91          //     return false;
92          // }
93          // 成功
94          return true;
95      }
96  </script>
```

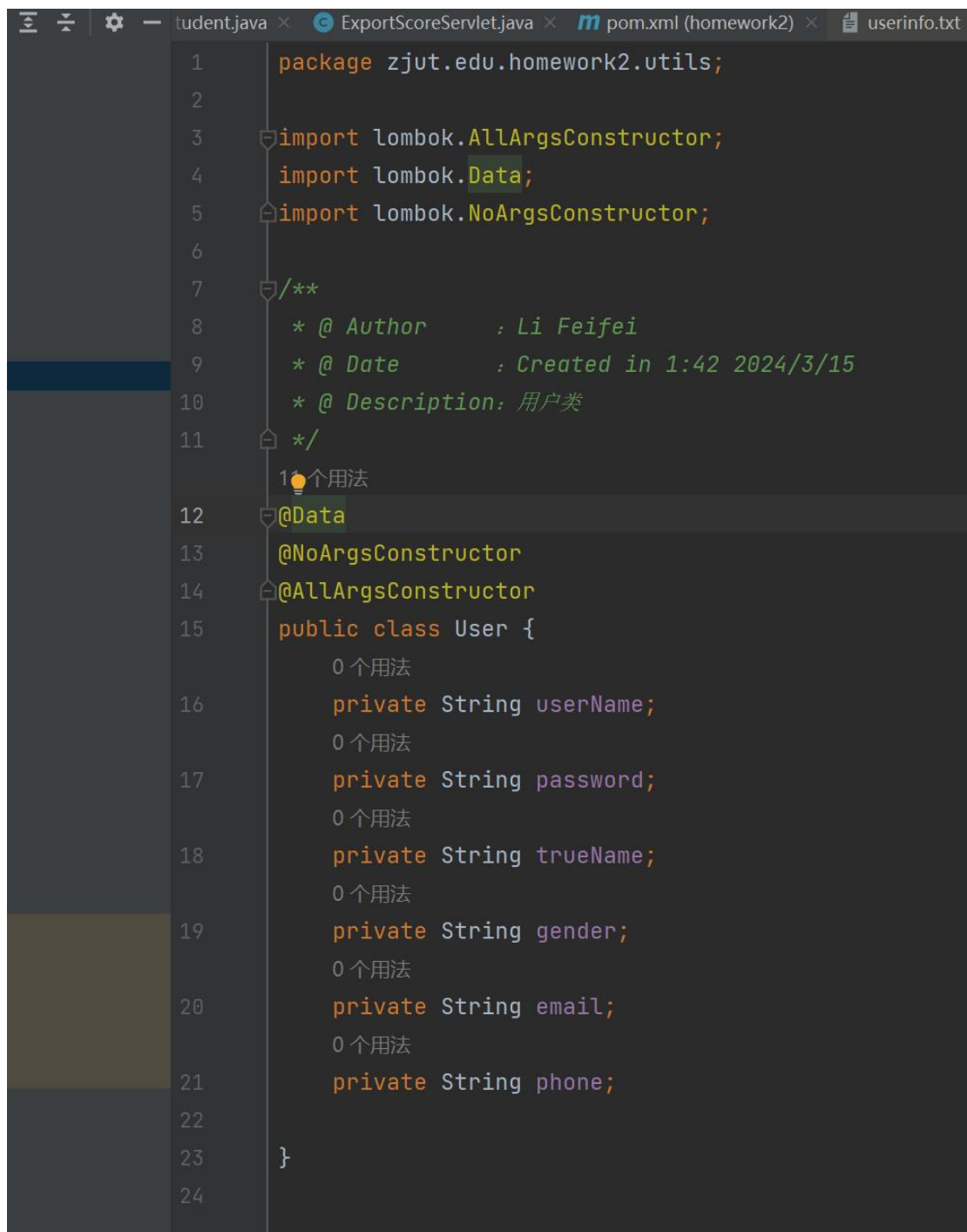
- 注册 register.html

```

115 <script language="JavaScript" type="text/javascript">
116     function custCheck() {
117         var userName = document.getElementById("userName");
118         var password = document.getElementById("password");
119         var trueName = document.getElementById("trueName");
120         var gender = document.querySelector('input[name="gender"]:checked');
121         var email = document.getElementById("email");
122         var phone = document.getElementById("phone");
123
124         // console.log(gender)
125         // console.log(gender.value)
126         //存在性校验
127         if (userName.value === "" || userName.value === "请输入用户名") {
128             alert("用户名不能为空!");
129             return false;
130         } else if (password.value === "" || password.value === "请输入密码") {
131             alert("密码不能为空!");
132             return false;
133         }
134         else if (trueName.value === "" || trueName.value === "请输入姓名") {
135             alert("姓名不能为空!");
136             return false;
137         } else if (!gender) {
138             alert("请选择性别!");
139             return false;
140         }
141         else if (email.value === "" || email.value === "请输入邮箱") {
142             alert("邮箱不能为空!");
143             return false;
144         } else if (phone.value === "" || phone.value === "请输入手机号") {
145             alert("手机号不能为空!");
146             return false;
147         }
148
149         //合理性校验
150         if (email.value.indexOf('@') === -1) {
151             alert("请输入正确的邮箱!");
152             return false;
153         } else if (phone.value.length !== 11) {
154             alert("手机号要求11位!");
155             return false;
156         }
157         return true;
158     }
159 </script>

```

- 用户类 User.java



```
1 package zjut.edu.homework2.utils;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Data;
5 import lombok.NoArgsConstructor;
6
7 /**
8  * @ Author      : Li Feifei
9  * @ Date        : Created in 1:42 2024/3/15
10 * @ Description: 用户类
11 */
12 @Data
13 @NoArgsConstructor
14 @AllArgsConstructor
15 public class User {
16     private String userName;
17     private String password;
18     private String trueName;
19     private String gender;
20     private String email;
21     private String phone;
22
23 }
24
```

- 用户工具类 UserUtils.java

```
10 import java.util.Set;
11
12 /**
13  * @ Author      : Li Feifei
14  * @ Date        : Created in 1:42 2024/3/15
15  * @ Description: 用户工具类
16  */
17 // 8个用法
18 public class UsersUtil {
19     // 2个用法
20     // 获取所有userinfo已注册用户
21     @ public static Set<User> getAllUsers(){
22         Set<User> users = new HashSet<>();
23
24         // 尝试读取userinfo.txt文件, 不存在创建
25         File file = new File( pathname: "userinfo.txt");
26         if (!file.exists()) {
27             try {
28                 file.createNewFile(); // 如果文件不存在, 则创建文件
29             } catch (IOException e) {
30                 e.printStackTrace();
31             }
32         }
33
34         try (InputStream inputStream = UsersUtil.class.getResourceAsStream( name: "/userinfo.txt");
35             BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream))) {
36             String line;
37             while ((line = reader.readLine()) != null) {
38                 // 按 "|" 分隔行数据
39                 String[] fields = line.split( regex: "\\|");
40                 if (fields.length == 6) // 确保每行数据都包含了所有字段
41                     users.add(
42                         new User(fields[0], fields[1], fields[2], fields[3], fields[4], fields[5])
43                     );
44             }
45         } catch (IOException e) {
46             e.printStackTrace();
47         }
48
49         return users;
50     }
51 }
```

```
49 }
50
51 // 记录一个用户(已经保证合法)
52 // 1个用法
53 @ public static Boolean saveUser(User user){
54     try (BufferedWriter writer =
55         new BufferedWriter(
56             new FileWriter(UsersUtil.class.getClassLoader().getResource( name: "userinfo.txt").getPath(),
57                 append: true))) {
58         // 使用 | 分隔字段, 并加密密码
59         String encryptedPassword = passwordEncode(user.getPassword());
60         String userInfo = user.getUserName() + "|" + encryptedPassword + "|" +
61             user.getTrueName() + "|" + user.getGender() + "|" +
62             user.getEmail() + "|" + user.getPhone();
63
64         // 将用户信息写入文件
65         writer.write(userInfo);
66         writer.newLine(); // 换行
67         writer.close();
68         return true;
69     } catch (IOException e) {
70         e.printStackTrace();
71         return false;
72     }
73 }
74
75 // 获取加密后字符串
76 // 2个用法
77 @ public static String passwordEncode(String password){
78     MessageDigest digest = new SM3.Digest();
79     byte[] hashBytes = digest.digest(password.getBytes());
80     return Hex.toHexString(hashBytes).toUpperCase();
81 }
```

- 登录 Servlet loginServlet.Java

```

24
25 0个用法
26 @Override
27 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28     request.setCharacterEncoding("UTF-8");
29     response.setContentType("text/html;charset=UTF-8");
30     String userName=request.getParameter( s: "userName");
31     String password=request.getParameter( s: "password");
32     //检测是否存在
33     Set<User> set= UsersUtil.getAllUsers();
34     for (User user : set) {
35         if (user.getUserName().equals(userName)) {
36             //存在 判断密码
37             if (UsersUtil.passwordEncode(password)
38                 .equalsIgnoreCase(user.getPassword())) {
39                 //成功直接跳转
40                 response.sendRedirect( s: "welcome.html");
41                 return;
42             }
43             //密码不匹配, 跳转失败
44             request.setAttribute("message", "您的密码错误!");
45             response.sendRedirect("/failed.html");
46             String errorMessage = "您的密码错误! ";
47             String redirectUrl = "failed.html?error=" +
48                 URLEncoder.encode(errorMessage, StandardCharsets.UTF_8);
49             response.sendRedirect(redirectUrl);
50             return;
51         }
52     }
53     //用户不存在 跳转失败
54     // 设置请求属性
55     String errorMessage = "用户名不存在! ";
56     String redirectUrl = "failed.html?error=" +
57         URLEncoder.encode(errorMessage, StandardCharsets.UTF_8);
58     // 重定向到 HTML 页面
59     response.sendRedirect(redirectUrl);
60 }
61 }
62 }
63

```

- 注册 Servlet registerServlet.java

```

23
24 0个用法
25 @Override
26 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
27     request.setCharacterEncoding("UTF-8");
28     response.setContentType("text/html;charset=UTF-8");
29     String userName=request.getParameter( s: "userName");
30     String password=request.getParameter( s: "password");
31     String trueName=request.getParameter( s: "trueName");
32     String gender=request.getParameter( s: "gender");
33     String email=request.getParameter( s: "email");
34     String phone=request.getParameter( s: "phone");
35
36     Set<User> set= UsersUtil.getAllUsers();
37     for (User user : set) {
38         if (user.getUserName().equals(userName)) {
39             //用户重复 返回
40             String htmlResponse = "<html><body><h2>用户名已存在, 请重新输入。" +
41                 "<a href='./register.html'>点击返回</a>" +
42                 "</h2></body></html>";
43             response.getOutputStream().write(htmlResponse.getBytes(StandardCharsets.UTF_8));
44             return;
45         }
46     }
47     Boolean success=UsersUtil.saveUser(new User(userName,password
48         ,trueName,gender,email,phone));
49     if(Boolean.TRUE.equals(success))
50         response.setHeader( s: "Refresh", s: "5;URL='./welcome.html'");
51 }
52 }
53

```