**Java Web开发**

**实验七**

**作业报告**

|  |  |
| --- | --- |
| 学 院 | 计算机 |
| 所在系 班 级 | 2017软件工程系15班 |
| 学 号 姓 名 | 04171427詹明 |
| 指 导 教 师 | 罗永升 |
| 完 成 时 间 | 2019年4月- |

# 实验题目

1、页面输入收入，得到相应的缴纳税收

要求：Jsp提交收入，显示结果（可一个页面也可两个页面）。编写一个JavaBean计算税收，并在JSP页面输出结果

计算方法：全月应纳税所得额＝月收入－1600元

级数 全月应纳税所得额 税率% 速算扣除法(元)   
1 dr <=500 5 0

2 500 <dr<=3000 10 25

3 3000 < dr <=5000 15 125

4 5000 < dr <=20000 20 375

5 20,000 < dr <=40000 25 1375

6 40,000 < dr <=60,000 30 3375

7 60,000 < dr <=80,000 35 6375   
8 80,000 < dr <=100,000 40 10375   
9 dr >100,000元 45 15375

如：月收入3000，则（3000-1600）\*10%-25=115，因为3000-1600大于500小于2000，所以按10%计算后再扣除速算的部分。

# 分析与设计

Jsp提交收入，显示结果（可一个页面也可两个页面）。编写一个JavaBean计算税收，并在JSP页面输出结果

# 关键源码（部分）

Input.jsp:

<%@ page language=*"java"* import=*"java.util.\*"* pageEncoding=*"UTF-8"*%>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<body bgcolor = *pink*><font size = *5*>

<form action = *"compute.view"* method = *"POST"* name = *"form"*>

<h1><center>税收计算</center></h1><br>

<h2><center>请输入月收入<input type = *"text"* name = *"number"*></center></h2>

<center><input type = *"submit"* value = *"提交"*></center>

</font></form></body>

</html>

Result.jsp:

<%@ page language=*"java"* import=*"java.util.\*"* pageEncoding=*"UTF-8"*%>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<body bgcolor = *pink*><font size = *5*>

<% String after = (String)request.getAttribute("after"); %>

<jsp:useBean id = *"Bean"* class = *"cc.openhome.Computebean"* scope = *"page"*/>

<jsp:setProperty property = *"money"* name = *"Bean"* value =*"*<%=after %>*"* />

<h2><center>个人应缴纳税收:<%=Bean.getMoney()%>元(结果精确到两位小数)</center></h2></h2>

<h2><center><a href = *"input.jsp"*>返回重新计算</a></center></h2>

</font></form></body>

</html>

Compute.java:

**package** cc.openhome;

**import** java.io.IOException;

**import** javax.servlet.RequestDispatcher;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

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

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

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

**import** javax.servlet.http.HttpSession;

/\*\*

\* Servlet implementation class compute

\*/

@WebServlet("/compute.view")

**public** **class** Compute **extends** HttpServlet {

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

// **TODO** Auto-generated method stub

doPost(request, response);

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// **TODO** Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

**double** number = Double.*parseDouble*(request.getParameter("number")) - 1600;

**double** after = 0;

**if**(number <= 0){

after = 0;

}

**else** **if**(number <= 500){

after = number \* 0.05;

}

**else** **if**(number <= 3000){

after = number \* 0.1 - 25;

}

**else** **if**(number <= 5000){

after = number \* 0.15 - 125;

}

**else** **if**(number <= 20000){

after = number \* 0.2 - 375;

}

**else** **if**(number <= 40000){

after = number \* 0.25 - 1375;

}

**else** **if**(number <= 60000){

after = number \* 0.3 - 3375;

}

**else** **if**(number <= 80000){

after = number \* 0.35 - 6375;

}

**else** **if**(number <= 100000){

after = number \* 0.4 - 10375;

}

**else**{

after = number \* 0.45 -15375;

}

String after1=String.*format*("%.2f", after);

request.setAttribute("after", after1);

RequestDispatcher requestDispatcher = request.getRequestDispatcher("/result.jsp");

requestDispatcher.forward(request, response);

}

}

ComputeBran:

**package** cc.openhome;

**public** **class** Computebean {

**private** String money;

**public** Computebean() {

}

**public** String getMoney() {

**return** money;

}

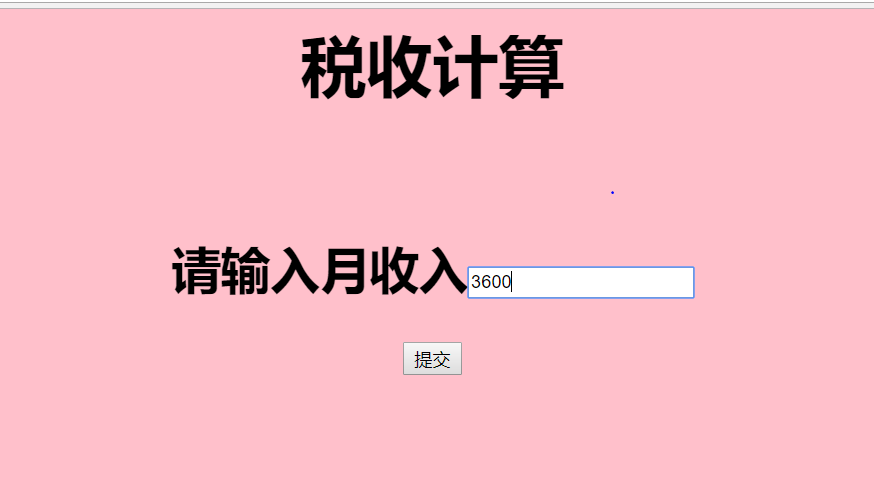
**public** **void** setMoney(String money) {

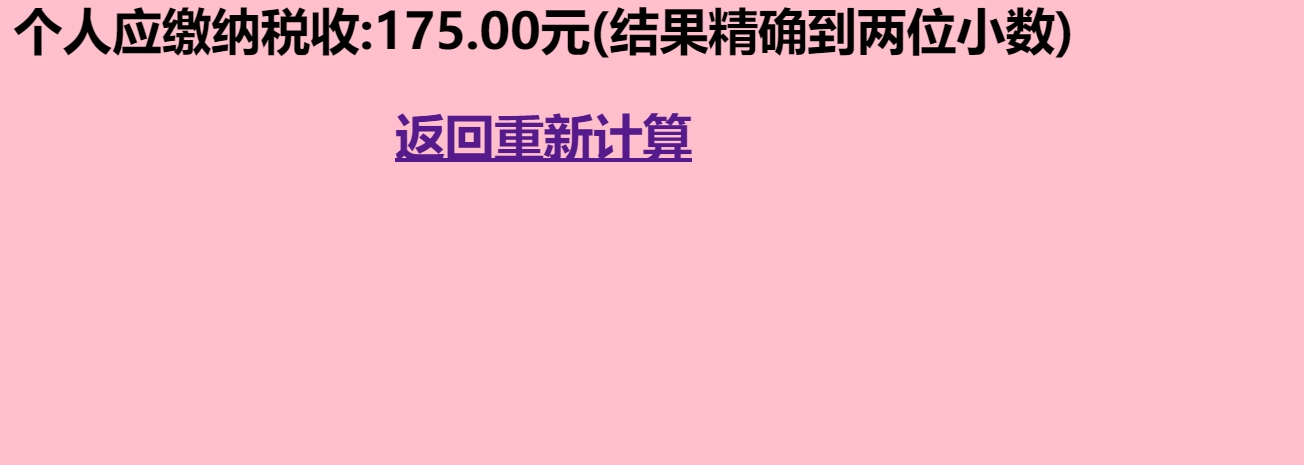
**this**.money = money;

}

}

# 运行效果图





# 收获与问题

Jsp+javabean 确实能够使思路更加清晰，之后会更加利用