

专业基础训练 II

设计报告

题 目： 工资管理系统

姓 名： 艾程

班 级： 软件 16-04 班

指导教师： 赵春刚

软件工程教研室
2018 年 7 月 6 日

1 总体设计

1.1 设计简介

一个完善的工资管理系统可以让管理者更方便地管理员工的应得资金，也会让员工更加清楚自己的工资。员工可以通过登陆账号查看自己的信息细则以及出勤情况，管理人员登陆后可以对员工信息进行增删改查、工资发放、更改奖惩/职位等信息的操作。登陆页面设置为消息栏，确保访问网页的员工可以及时获取到管理员发布的消息。工资管理系统可以让使用人员更好地对员工的工资进行管理及分配，让使用人员可以在各个地点各个时间对信息进行查询或者修改，以及可以查看出每个员工的出勤情况。使用网络技术将工资管理系统加入企业的日常管理中，为企业的管理改造提供可行的途径。

本系统采用 JSP 和 Servlet 技术完成系统编码。前端使用采用 html、jstl 技术，数据库使用 MySQL 保存数据。系统主要的目标是开发一个页面对用户友好，用户方便操作的工资管理系统。通过信息化的建设，使企业的管理更轻松。此外，还能够通过查询出勤情况，了解哪个员工今日未出勤。设计的实现包括：员工基本信息管理、工资发放管理、考勤管理、和奖惩、职位等级管理、消息栏、登陆等模块。

1.2 功能模块及分析

本系统业务逻辑模块分为员工基本信息管理、工资发放管理、考勤管理、和奖惩、职位等级管理等模块。

1.2.1 总体功能模块介绍

系统总体功能模块图如图 1 所示。

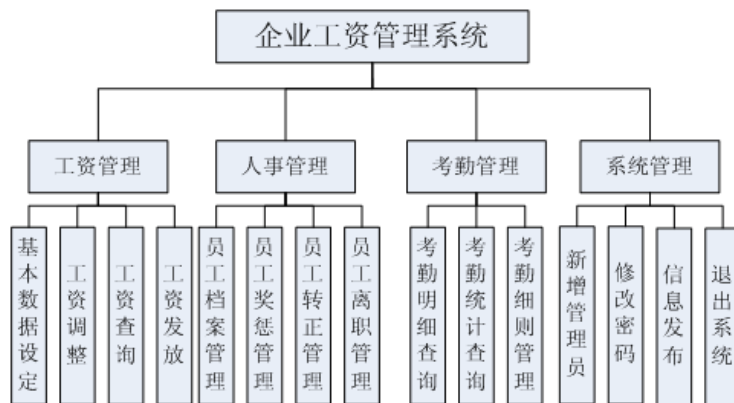


图 1 工资管理系统数据库的关系模型图

1.2.2 员工基本信息管理模块

对于员工而言：

- ① 可以访问到自己的信息，包括编号、姓名、密码、工资、提成、奖金、最后一次签到时间、最后一次得到工资的时间的信息
- ② 能修改自己的账号密码。
- ③ 进行签到操作

对于管理员而言：

- ① 可以增删改查员工的信息，包括姓名、密码、工资、提成、奖金、最后一次签到时间、最后一次得到工资的时间的信息
- ② 查询缺勤人数以及具体信息

功能模块图如图 2 所示。

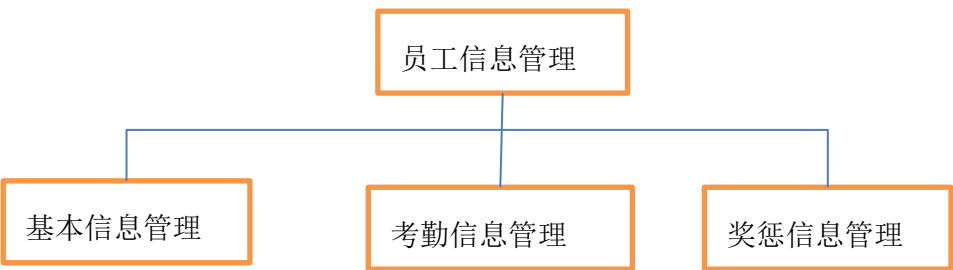


图 2 员工信息管理功能模块图

1.2.3 管理员信息管理模块

只能由管理员访问：

- ① 能增删改查所有员工、管理员的账号、密码信息
- ② 发放工资
- ③ 查看缺勤人数以及具体员工信息
- ④ 修改消息栏信息

功能模块图如图 3 所示。

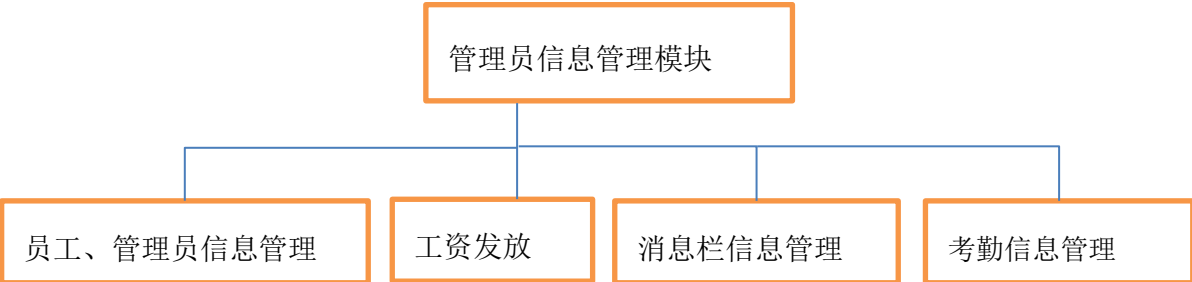


图 3 管理员信息管理模块图

1.2.4 登陆管理模块

登陆模块主要是方便区分当前用户的，从而为不同用户提供不同的操作权限访问信息。通过该模块可以访问到员工、管理员、消息栏的信息。功能模块图如图 4 所示。

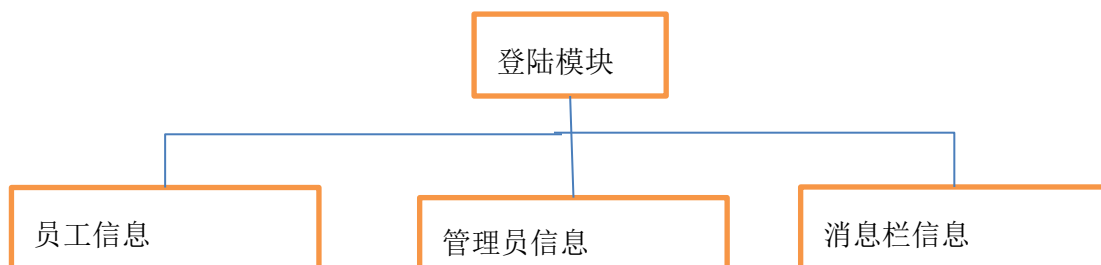


图 4 登陆功能模块图

1.2.5 其它模块

错误页面模块：当出现例如“账号或密码错误”导致登陆模块运行不成功，或者“余额”不足导致管理员信息模块发放工资失败等原因，都会跳转到这个模块。

消息栏管理模块：消息栏主要存放的是企业管理员通知员工消息的文本信息。该模块内容只能由管理员更改，员工只读。

1.3 数据库设计

1.3.1 数据库环境说明

本设计采用 MySQL 作为数据库。

MySQL 是一种开放源代码的关系型数据库管理系统，使用最常用的数据库管理语言--结构化查询语言 (SQL) 进行数据库管理。本学期我们学习了 MySQL 数据库的相关知识和操作。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了 MySQL 作为网站数据库。所以本次设计也采用 MySQL 作为网站数据库。

1.3.2 逻辑设计

根据系统功能需求和分析，可以得到系统中涉及到的实体包括员工、管理员、信息集（消息栏、可用余额）。

1. 员工实体

员工 worker 实体，包括属性有编号、姓名、密码、工资、提成、奖金、最后一次签到时间、最后一次得到工资的时间的信息。

实体关系图如图 5 所示。

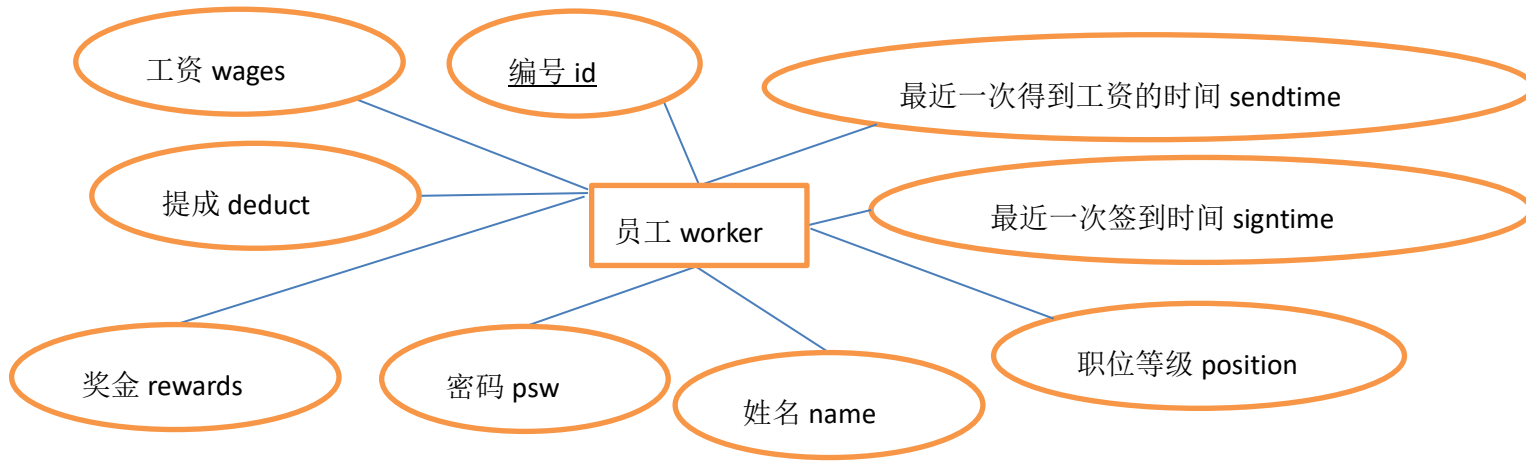


图 5 员工实体属性图

2. 管理员实体

管理员实体 manager，属性包括账号名、密码。实体关系图如图 6 所示。

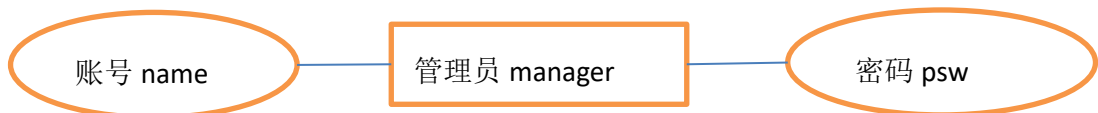


图 6 管理员实体属性图

3. 信息实体

信息 info 实体，属性包括消息栏文本内容以及可用余额。实体关系图如图 7 所示。



图 7 信息实体属性图

1.3.3 物理设计

根据实体关系图，可以得到对应的物理实现。

1. 员工表

员工表如表 1 所示。其中关键字是员工编号，是自动生成的。

表 1 员工表

列名	数据类型	可否为空	说明
id	int	NOT NULL	编号
deduct	float	NOT NULL	提成
rewards	int	NOT NULL	奖金
wages	int	NOT NULL	工资
name	varchar	NOT NULL	姓名
psw	varchar	NOT NULL	密码
signtime	datetime	NOT NULL	最后签到时间
sendtime	datetime	NOT NULL	上一次得到工资时间
position	varchar	NOT NULL	职位等级

2. 管理员表

管理员表如表 2 所示。记录管理员的账户名和密码。

表 2 管理员表

列名	数据类型	可否为空	说明
name	varchar	NOT NULL	账号名
psw	varchar	NOT NULL	密码

3. 信息表

信息表如表 3 所示。记录消息栏的信息和所剩余额。

表 3 信息表

列名	数据类型	可否为空	说明
text	varchar	NOT NULL	消息栏内容
money	varchar	NOT NULL	企业可用余额

2 详细设计

2.1 程序设计说明

2.1.1 基础类

User 类：员工类，记录员工的基本信息，如表 4 所示。

表 4 用户信息类表(User)

列名	数据类型	说明
id	int	编号
deduct	float	提成
rewards	int	奖金
wages	int	工资
name	varchar	姓名
psw	varchar	密码
signtime	datetime	最后签到时间
sendtime	datetime	上一次得到工资时间
position	varchar	职位等级

2.1.2 DAO 类

ConnDB 类:该系统的功能大部分都会和数据库进行连接并取值。为了简化代码操作，以及让代码看起来美观，于是将连接数据库并访问数据库的代码提取出来作为 ConnDB 类。如表 5 所示。

表 5 数据库连接类表(ConnDB)

列名	数据类型	说明
url	String	数据库地址
drivername	String	驱动名
username	String	用户名
pssword	String	密码
connection	Connection	连接对象

getConnection()	Connection	获得链接
-----------------	------------	------

2.1.3 BO 类

LoginServlet 类:该类与登陆管理模块息息相关，是验证用户身份是否属实，若是，则根据该用户权限的不同让它跳转到不同的界面的的类。

表 6 登陆实现类表(loginServlet)

列名	数据类型	说明
session	HttpSession	会话
sql	String	数据库操作语句
username	String	用户名
pssword	String	密码
connDB	ConnDB	数据库操作工具类
statement	Statement	数据库对象
resultSet	ResultSet	数据库结果

UserListServlet.java:该类与管理员管理模块息息相关，是给管理员管理模块界面提供员工详细信息的界面。

表 7 员工信息类表(UserListServlet)

列名	数据类型	说明
pageBean	PageBean	分页工具
sql	String	数据库操作语句
list	List<User>	用户详细信息
connDB	ConnDB	数据库操作工具类
statement	Statement	数据库对象
resultSet	ResultSet	数据库结果

UserServlet.java:该类负责有关员工信息的所有增删改查操作，记录缺勤人员编号以及负责计算企业工资余额。

表 8 管理员工信息类表(UserServlet)

列名	数据类型	说明
session	HttpSession	会话
sql	String	数据库操作语句

df	SimpleDateFormat	当前时间格式
action	String	记录当前操作
connDB	ConnDB	数据库操作工具类
statement	Statement	数据库对象
resultSet	ResultSet	数据库结果
list	List<User>	用户详细信息
sign	boolean	判别用户权限
UpdateUser()	void	更改信息操作
deleteUser()	void	删除信息操作
addUser()	void	增加信息操作
readUser()	void	查询信息操作
culculate()	void	计算余额
timeUser()	void	得到操作后的时间
abseneUser()	void	得到缺勤人员编号

2.1.4 Util 类

PageBean.java: 实现分页功能，避免管理员查询员工信息时，员工信息太多导致页面不美观。

表 9 分页功能类表 (PageBean)

列名	数据类型	说明
pageNum	int	当前页数
totalCount	int	员工总数
totalPage	int	总页数
pageSize	int	每页大小
statement	Statement	数据库对象
resultSet	ResultSet	数据库结果
sql	String	数据库操作语句
connDB	ConnDB	数据库操作工具类
culPageNum()	void	计算总页数
pageUser()	void	得到员工总数

2.2 程序运行效果

启动服务器后，打开浏览器，地址栏输入 `http://localhost:8080/ClassDesign/login.jsp`，进入系统登录页面，如图 7 所示。



登陆界面

ClassDesign/login.jsp

在此搜索

用户名

密 码

权 限

登陆

信息通知： 今天下午召开全体员工大会！望周知！

图 7 系统登录页面 (login.jsp)

用户输入用户名和密码和验证码，系统判断用户名和密码是否匹配成功，判断用户属于员工还是管理员。若密码验证通过则进入系统主页面，如图 8 所示。

(1) 员工:



图 8 员工系统主页面(workerView.jsp)

点击“签到”，提示“签到成功”（图 9）后，3 秒钟自动跳回图 8。

签到成功！3秒后返回页面！

图 9 签到成功提示页(PageError.jsp)

点击“更改密码”，则会弹出弹窗询问是否修改，然后进入图 10 界面进行操作。



图 10 更改员工密码页(adduser.jsp)

(2) 管理员：



图 11 管理员系统主页 (ManagerView.jsp)

点击“提交”后，可以通过员工编号/员工姓名查询员工的信息，得到图 12 所示图。



图 12 管理员视角的员工信息页面 (workerView.jsp)

点击“新增员工”后，会出现图 13，点击 **adduser** 提交按钮后又会回到图 11，此时点击“尾页”，就能查看到刚刚增加的人确实添加进去了：

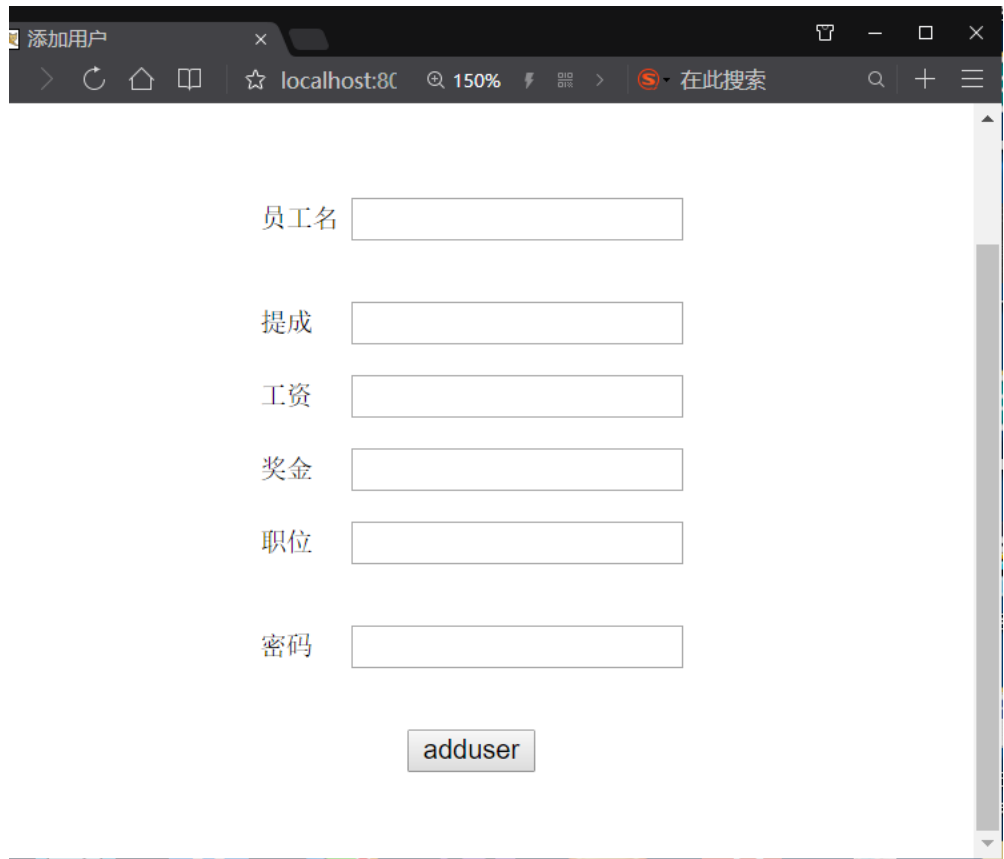


图 13 新增员工页面 (adduser.jsp)

点击员工表的“修改”，则会出现结构与图 13 一样，但是里面有原本该被修改员工信息的数值如下图，同样是点击按钮后会跳转回图 11，可看到操作成功：

65	连续	500	0	0.0	A	2018-07-05 20:07:00.0	2018-07-05 23:58:23.0	删除	修改
----	----	-----	---	-----	---	-----------------------	-----------------------	--------------------	--------------------

图 14 修改员工信息成功的部分页面 (ManagerView.jsp)

同样点击“删除”后，回弹出窗口询问是否删除，确认后会自动刷新当前页面，该员工记录消失。

点击“一键发送员工工资”后，如果发送成功，则直接刷新图 11 页面，不过“最后一次发工资时间”更改成了刚刚操作的时间，以及“当前可用资金”这一项会相应地减少。如果发送失败，则会弹出图 15 的页面。

余额不足!!!!发工资失败！总共需要15000元，当前需充值6400元！

图 15 发送工资失败的页面 (Nomoney. jsp)

点击“考勤详情”，则会出现图 16，一一列出详细的缺勤员工信息。



编号	用户名	工资	奖金	提成	职位	最近一次签到	最后一次发工资时间
68	公孙達	500	0	0.0	A	null	2018-07-05 23:58:23.0
69	别瑾	500	0	0.0	A	null	2018-07-05 23:58:23.0
70	屈强	500	0	0.0	A	null	2018-07-05 23:58:23.0
71	艾滢浯	500	0	0.0	A	null	2018-07-05 23:58:23.0
72	闾丘咎	500	0	0.0	A	null	2018-07-05 23:58:23.0
73	宦弄拨	500	0	0.0	A	null	2018-07-05 23:58:23.0
74	季雫澔	500	0	0.0	A	null	2018-07-05 23:58:23.0
75	闵婷	400	0	0.0	B	null	2018-07-05 23:58:23.0
76	成磔吵	400	0	0.0	B	null	2018-07-05 23:58:23.0
77	伏遨涸	400	0	0.0	B	null	2018-07-05 23:58:23.0
78	桂悔	400	0	0.0	B	null	2018-07-05 23:58:23.0

图 16 缺勤人员详情的页面 (absdetail. jsp)

3 结论

本次项目的编写涉及到的技术并不困难，大体框架容易写，只是更多的时间花费在了细枝末节的小 bug 上，解决了这些 bug，就相当于学到了一些新的东西：

① 知道了 `getAttribute()`和 `getParameter()`的区别

这两者虽然同样是保存数据，但是使用的范围不一样。一个是转发页面的时候用，一个是当前页面从其他地方取值的时候用。如果从页面跳转到了 `servlet` 中并没有某个属性，但是我们使用了 `getAttribute()`方法就会得到 `null` 值，如果使用另一个才会得到正确结果。

② 更了解“转发请求”与“响应”，以及他们的关系

如果已经“转发请求”了，相当于一次“伪响应”，真正的响应是转发后的页面决定的。最好在改代码下方加上“`return`”，防止服务器继续向下走程序，走到了其它的响应代码。一个请求只能得到一个响应，如果不加上 `return`，则服务器会继续向下查询代码，如果查询到两次或以上的响应代码，则会出现“`illegalComponentStatementException`”报错。

③ 如果新增记录的某个属性（例如员工编号）不需要我们写，需要自动生成，则应将该属性设置成自增

我们都知道 `PreparedStatement` 这个类极大地方便了我们动态插入数据到数据库，有的记录的部分属性值就算我们不写，该类也会传 `null` 值或者默认值过去。但是对于必须要写的属性值(主键)，例如员工编号，我们如果不想写，想要数据库随着记录自动生成的话，就必须在数据库设置该主键为自增模式，否则会报错“`Fields 'id' doesn't have a default value`”

④ `PreparedStatement` 的索引与 `resultSet` 所以的不同

使用 `preparedStatement` 上传数据时，使用语句“`preparedStatement.setInt(索引, 值)`”
使用 `resultSet` 获取数据的时候，使用语句“`resultSet.getInt(索引)`”

前者的索引是对应你 `sql` 语句中该属性的“？”所在位置的索引，后者则是对应数据库中该属性所在列数的索引。如果搞混了，很有可能由于数据类型不匹配，导致“`SQLException`”异常。

⑤ 超链接传参时，被传递数值参数要用 `getParameter()`接受数值。以及更了解了 EL 表达式

一般我们会习惯性地使用 EL 表达式得到属性：`${属性名}`，该表达式对应的是 `getAttribute()`方法。而我们的超链接需要使用 `getParameter()`方法，因此对应的 EL 表达式应该是`${param.属性名}`。否则会出现传递过去的值是 `null` 值的问题。

⑥ 数据库执行语句，查询和更改操作的语句是不同的

除了 `preparedStatement`,我们还使用常规的 `statement` 获取数据库。进行查询操作的时候是 `"statement.executeQuery()"`，但是更改数据库记录的时候就是`"statement.executeUpdate()"`了，否则运行后虽然不报错，但是操作无效。这两个方法长得很像，也很容易弄错以及忽略。

⑦ 传递给数据库的值的数据类型要和数据库的一一匹配，但是得到数据库的值的时候，数据类型可以不一样

传递的时候，为了不出现格式错误，我们要传递与数据库一致类型的数据，否则会出现 `SQLException` 异常错误。然而得到数据的时候，都是默认得到 `string` 数据类型的，所以我们在得到数据的时候，使用 `getString()`也行，如果使用 `getInt()`得到一个整型数据，则也是服务器先将该数据变成字符串类型再转换为整型类型。

还有一些操作失误、细节错误，比如在编写 `pageBean` 类的时候出现了计算页面总数错误、使用 `jstl` 的时候忘记导包导致语句失效等的小问题。这种很容易找出问题所在并且下次不容易再犯的错误，这里就不一一记录。