课程名称：JavaEE 技术	年级：xxxx	实验日期：
姓名：　xxx	学号：xxxxx	班级：　xxxxx
实验名称：JSP 综合编程	实验序号：实验三	成员人数：1

一、 实验目的及要求

实验目的：

1、掌握 JSP 内置对象的含义；
2、理解客户端发出请求的类型和特点；
3、熟练 JSP 处理客户端请求的方法；
4、熟悉各种内置对象的常用方法。

实验原理：

1、掌握使用 HashMap 编写购物车技术；
2、掌握 JSP 界面嵌入 Java 代码片段技术。

二、实验环境(仪器与材料)

Windows 7/10，MySQL/MariaDB，Eclipse,Tomcat

三、实验内容及完成情况

1、创建后台支撑数据库

```
create database Shopping
on
( name='Shopping_data',
   filename='d:\Shopping_data.mdf',
   size=10mb
)
log on
( name='Shopping_log',
   filename='d:\Shopping_log.ldf',
   size=10mb
)
go
use Shopping
go
create table users
(
   id int identity(1,1) primary key, --标签编号  为自增长字段，增长步长为 1，主码
   username nvarchar(20) not null,     --用户姓名  可变字符串类型，长度 20
   [password] nvarchar(20) not null, --密码         可变字符串类型，长度 20
   email nvarchar(50) not null,
   tel     nvarchar(20) not null,
   grade int default 1 not null
)

insert into users values('wl','123','wl@126.com','1303030333',2)
insert into users values('admin','123','admin@126.com','346789',2)
```

```
create table book
(
   id int identity(1,1) primary key, --标签编号  为自增长字段，增长步长为 1，主码
   bookname nvarchar(50) not null,     --书名  可变字符串类型，长度 20
   author nvarchar(50) not null,
   publishHouse nvarchar(200) not null,
   price numeric(10,3) not null,
   nums numeric(10,3) default 1000
)
insert into book values('jsp 应用开发详解','刘晓华','电子工业出版社',59.80,500)
insert into book values('Java 项目开发案例全程实录(第 2 版)','李钟尉,陈丹丹','清华大学出版社',69.80,200)
insert into book values('ASP.NET 项目开发案例全程实录(第 2 版)','郑齐心,房大伟,刘云峰','清华大学出版社',74.00,456)
insert into book values('jquery 技术内幕：深入解析 jquery 架构设计与实现原理','高云','机械工业出版社',99.00,200)
……
```

2、编写 JSP+JavaBean 程序，验证用户是否为合法的用户，合法用户，则登录到主页面，并显示当前登录用户名。可以查看数据库中所有信息，并可以插入信息用户信息。

步骤如下：

（1）在 com.wl.domain 包下面创建 Users 和 Book 类

具体代码如下：

```
package com.wl.domain;
//这是一个javabean，和数据库中的book表是相对应的
public class Book {
     private String id;
     private String bookname;
     private String author;
     private String publishHouse;
     private String price;
     private String nums;  //该书的库存量
     private int buyNums=1;//已经购买的数量
     public int getBuyNums() {
          return buyNums;
     }
     public void setBuyNums(int buyNums) {
          this.buyNums = buyNums;
     }
     public String getId() {
          return id;
     }
     public void setId(String id) {
          this.id = id;
     }
     public String getBookname() {
          return bookname;
     }
     public void setBookname(String bookname) {
          this.bookname = bookname;
     }
```

```
	public String getAuthor() {
		return author;
	}
	public void setAuthor(String author) {
		this.author = author;
	}
	public String getPublishHouse() {
		return publishHouse;
	}
	public void setPublishHouse(String publishHouse) {
		this.publishHouse = publishHouse;
	}
	public String getPrice() {
		return price;
	}
	public void setPrice(String price) {
		this.price = price;
	}
	public String getNums() {
		return nums;
	}
	public void setNums(String nums) {
		this.nums = nums;
	}
}
//这是一个javabean，和数据库中的users表是相对应的
package com.wl.domain;
public class Users {
	private String id;
	private String username;
	private String password;
	private String email;
	private String tel;
	private int grade;
	public String getId() {
		return id;
	}
	public void setId(String id) {
		this.id = id;
	}
	public String getUsername() {
		return username;
	}
	public void setUsername(String username) {
		this.username = username;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}
	public String getEmail() {
```

```
            return email;
      }
      public void setEmail(String email) {
            this.email = email;
      }
      public String getTel() {
            return tel;
      }
      public void setTel(String tel) {
            this.tel = tel;
      }
      public int getGrade() {
            return grade;
      }
      public void setGrade(int grade) {
            this.grade = grade;
      }
}
```

（2）编写 UserService，使用 JavaBean 体现出面向对象，有助于程序员关注对象编程

```
package com.wl.service;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import com.wl.domain.Users;
import com.wl.utils.SQLHelper;
//专门用于处理业务逻辑的类
//处理和users表相关的业务逻辑
public class UserService {
      public boolean checkUsers(Users users) {
            String sql = "select * from users where username=?
                        and password=?";
            String[] parameters = { users.getUsername(),
                                    users.getPassword() };
            ArrayList al = new SQLHelper().executeQuery2(
                              sql, parameters);
            if (al.size() == 0) {
                  return false;
            } else {
                  Object[] obj = (Object[]) al.get(0);
                  // 把对象数组装入users对象
                  users.setUsername((String) obj[1]);
                  users.setEmail((String) obj[3]);
            users.setGrade(Integer.parseInt(obj[5].toString()));
                  return true;
            }
      }
}
```

测试修改后代码，能否正常登陆。

（3）编写 BookService

```
package com.wl.service;
import java.sql.ResultSet;
import java.util.ArrayList;
```

```
import com.wl.domain.Book;
import com.wl.utils.SQLHelper;
//专门用于处理业务逻辑的类,处理和book表相关的业务逻辑
public class BookService {
public ArrayList getAllBooks() {
          String sql = "select * from book where 1=?";
          String[] parameters = { "1" };
          ArrayList al = new SQLHelper().executeQuery2(sql,
parameters);
          ArrayList<Book> newal = new ArrayList<Book>();
          // 需要进行二次业务封装，因为，al中得到的是 对象数组，需要转换成
             bookBean(一个Bean就是一个java类).
          for (int i = 0; i < al.size(); i++) {
               Object[] obj = (Object[]) al.get(i);
               Book book = new Book();
               book.setId(obj[0].toString());
               book.setBookname(obj[1].toString());
               book.setAuthor(obj[2].toString());
               book.setPublishHouse(obj[3].toString());
               book.setPrice(obj[4].toString());
               book.setNums(obj[5].toString());
               newal.add(book);
          }
          return newal;
}
     //根据Book id , 返回一个bookBean
     public Book getBookByID(String id){
     String sql="select * from book where id=?";
     String []parameters={id};
     ArrayList al=new SQLHelper().executeQuery2(sql, parameters);
          //为了安全起见，使用if
          Book book=new Book();
          if(al.size()==1){
               Object[] obj = (Object[]) al.get(0);
               book.setId(obj[0].toString());
               book.setBookname(obj[1].toString());
               book.setAuthor(obj[2].toString());
               book.setPublishHouse(obj[3].toString());
               book.setPrice(obj[4].toString());
               book.setNums(obj[5].toString());
          }
          return book;
     }
}
```

（4）修改 index.jsp 代码

```
<body>
  <jsp:forward page="/WEB-INF/login.jsp" ></jsp:forward>
</body>
```

（5）编写登录界面 JSP 代码

```
<body>
    <h1>欢迎登陆购物系统--登录页面</h1>
    <form action="/myshopping/GoHallUI" method="post">
```

```
    <table border='1'>
      <tr><td>用户名：</td><td><input type="text"
name="username"/></td></tr>
      <tr><td>密　码：</td><td><input type="password"
name="password"/></td></tr>
      <tr><td><input type="submit" value="登陆"/>  </td><td><input
type="reset" value="清空"/></td></tr>
    </table>
    </form>
    <%
    String errinfo=(String)request.getAttribute("errinfo");
    if(errinfo!=null){
      out.println(errinfo);
    }
    %>
  </body>
```

运行效果如图 3-1 所示。

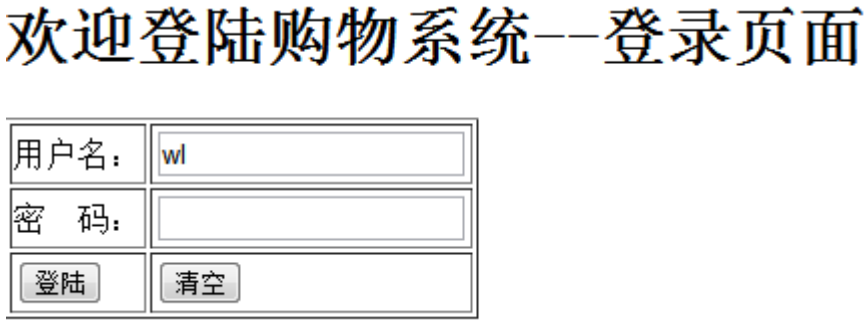

图 3-1 登录界面

（6）编写控制层GoLogin. java(用servlet编写)

```
package com.wl.controller;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class GoLogin extends HttpServlet {
     public void doGet(HttpServletRequest request, HttpServletResponse
response)throws ServletException,IOException
    {
          response.setContentType("text/html;charset=utf-8");
          PrintWriter out = response.getWriter();
     if(request.getSession().getAttribute("userinfo")!=null){
          request.getSession().setAttribute("userinfo", null);
          request.getRequestDispatcher("/WEB-INF/login.jsp").
forward(request, response);
          }
     }
     public void doPost(HttpServletRequest request, HttpServletResponse
response)throws ServletException, IOException
    {
          this.doGet(request, response);
```

```
    }
}
```

（7）编写编写控制层 GoHall.java(用 servlet 编写)

```
package com.wl.controller;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.util.ArrayList;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.wl.domain.Users;
import com.wl.service.BookService;
import com.wl.service.MyCart;
import com.wl.service.UserService;
import com.wl.utils.SQLHelper;
public class GoHallUI extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse
response)throws ServletException, IOException {
          response.setContentType("text/html;charset=utf-8");
          PrintWriter out = response.getWriter();
          // 取出从登陆页面数据的用户名和密码。只要是页面传递的 参数，都可以使用
request.getParameter()方法
          String username = request.getParameter("username");
          String password = request.getParameter("password");
          // 创建一个登陆对象
          Users users = new Users();
          users.setUsername(username);
          users.setPassword(password);
          //先判断用户是否已经成功登录过,如果登陆过，则直接跳转到购物大厅。无需再次
分配一个新的购物车。
     if(request.getSession().getAttribute("userinfo")!=null){
       BookService bookService = new BookService();
       ArrayList al = bookService.getAllBooks();
       request.setAttribute("books", al);
       request.getRequestDispatcher("/WEB-INF/hall.jsp").
       forward(request,response);
       return ;//这句话必须加上，禁止程序继续向下执行
    }
          // 用对应的业务逻辑类UserService 来完成验证用户的工作。首先要新建一个业
务逻辑类的对象,在调用这个对象对应的方法即可
          UserService userService = new UserService();
          if (userService.checkUsers(users)) {
               request.getSession().setAttribute("userinfo", users);
               // 说明是合法的，可以跳转到购物大厅
               // 在跳转之前，得到所有的书信息，放入request的Attribute域中，以
便带入到下一个页面，一定不能放入session中，
               //用户登录成功时，就给这个用户分配一个空的购物车
          MyCart myCart=new MyCart();
          request.getSession().setAttribute("myCart", myCart);
```

```
            BookService bookService = new BookService();
            ArrayList al = bookService.getAllBooks();
            request.setAttribute("books", al);
           request.getRequestDispatcher("/WEB-INF/hall.jsp").
          forward(request,response);
            } else {
            request.setAttribute("errinfo", "错误的用户名或密码！");
            request.getRequestDispatcher("/WEB-INF/login.jsp").
          forward(request,response);
            }
      }
      public void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
            this.doGet(request, response);
      }
}
```

（8）编写购物大厅界面 JSP 代码

```
<%@ page language="java"
import="java.util.* ,java.sql.*,com.wl.domain.*" pageEncoding="utf-8"%>
<body>
    <h1>欢迎购物</h1>
    <form action='/myshopping/ShoppingServlet?type=showMyCart'
method='post'>
    <table border='1' cellspacing='0'>
     <tr><th>书名</th><th>作者</th><th>出版社</th><th>价格</th><th>剩余数量
</th><th>购买</th></tr>
    <%
    //取出GoHallUI这个servlet带过来的books信息
    //即：取出request中的ArrayList
    ArrayList al=(ArrayList)request.getAttribute("books");
    for(int i=0;i<al.size();i++){
      Book book=(Book)al.get(i);
      %>
      <tr>
            <td><%=book.getBookname()%></td>
            <td> <%=book.getAuthor()%></td>
            <td><%=book.getPublishHouse()%></td>
            <td><%=book.getPrice()%></td>
            <td><%=book.getNums()%></td>
            <td><a href='/myshopping/ShoppingServlet?type=addBook&id=
<%=book.getId() %>'>购买</a></td>
      </tr>
    <%
    }
    %>
    <tr><td colspan='6'><input type='submit' value=
    '查看购物车'/></td></tr>
    </table>
    </form>
    <a href="/myshopping/GoLogin">返回重新登录</a>
  </body>
```

运行效果如图 3-2 所示。

欢迎购物

书名	作者	出版社	价格	剩余数量	购买
jsp应用开发详解	刘晓华	电子工业出版社	59.800	500.000	购买
Java项目开发案例全程实录(第2版)	李钟尉,陈丹丹	清华大学出版社	69.800	200.000	购买
ASP.NET项目开发案例全程实录(第2版)	郑齐心,房大伟,刘云峰	清华大学出版社	74.000	456.000	购买
jquery技术内幕：深入解析jquery架构设计与实现原理	高云	机械工业出版社	99.000	200.000	购买
轻量级java ee企业应用实战（第3版）	李刚	电子工业出版社	99.000	100.000	购买
Struts+Spring+Hibernate框架及应用开发（Java EE工程师零起点培训系列）	王建国,王建英	清华大学出版社	69.000	2000.000	购买
Java编程思想(第4版)	Bruce Eckel	机械工业出版社	108.000	3000.000	购买
Java从入门到精通(第3版)	明日科技	清华大学出版社	59.800	100.000	购买
java核心技术(卷i)基础知识(原书第9版)	Cay S.Horstmann;Gary Cornell	机械工业出版社	119.000	300.000	购买
C#从入门到精通(第3版)	明日科技	清华大学出版社	69.800	200.000	购买

查看购物车

返回重新登录

图 3-2 购物大厅界面

（9）编写购物车代码，放置在 com.wl.service 包中

```
package com.wl.service;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import com.wl.domain.Book;
//表示我的购物车
public class MyCart {
	HashMap<String, Book> hm = new HashMap<String, Book>();
	// 添加书(购买数)的第二个版本
	public void addBook2(String id){
		if(hm.containsKey(id)){
			//hm已经有了这本书
			Book book=hm.get(id);
			int buyNums=book.getBuyNums();
			book.setBuyNums(buyNums+1);
		}else{
			hm.put(id, new BookService().getBookByID(id));
		}
	}
	// 添加书(购买数)
	public void addBook(String id, Book book) {
		if (hm.containsValue(id)) {//如果已经购买过这本书，再次购买，则需要把buyNums++即可
			book=hm.get(id);
			int buyNums= book.getBuyNums();
			buyNums++; //增加1之后，再次放进去即可
			book.setBuyNums(buyNums);
		} else {//第一次购买
			hm.put(id, book);
		}
	}
	// 删除书
	public void deleteBook(String id) {
		hm.remove(id);
	}
```

```
	// 写一个显示该购物车里所有商品（书）的函数
	public ArrayList showMyCart() {
		ArrayList<Book> al = new ArrayList<Book>();
		Book book = new Book();
		// 遍历HashMap;
		Iterator it = hm.keySet().iterator();
		while (it.hasNext()) {
			// 首先取出key
			String id = (String) it.next();
			// 然后取出bean
			book = hm.get(id);
			al.add(book);
		}
		return al;
	}
	// 更新书(其实是更新购买书的数量)
	public void updateBook(String id, int num) {
		//取出id对应的那本书
		Book book=hm.get(id);
		book.setBuyNums(num);
	}
	// 清空书,清空购物车
	public void clearBook() {
		hm.clear();
	}
	//返回购物车中物品的总价格
	public float getTotalPrice(){
		float  totalPrice=0.0f;
		ArrayList<Book> al=new ArrayList<Book>();
		//得到总价格
		Iterator iterator=hm.keySet().iterator();
			while(iterator.hasNext()){
				//取出书的编号
				String bookId=(String)iterator.next();
				Book book=hm.get(bookId);
				totalPrice+=
book.getBuyNums()*Float.parseFloat( book.getPrice());
			}
		return totalPrice;
    }
}
```

（10）编写购买代码，放置在 com.wl.controller 包中

```
package com.wl.controller;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.wl.domain.Book;
```

```
import com.wl.service.BookService;
import com.wl.service.MyCart;
//相应用户购买商品的请求
public class ShoppingServlet extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse
response)throws ServletException, IOException {
          response.setContentType("text/html;charset=utf-8");
          PrintWriter out = response.getWriter();
          //接收type值，以区分用户所想执行的动作
          String type=request.getParameter("type");
          MyCart myCart=new MyCart();
          if("addBook".equals(type)){
               //接收用户想购买的书号
               String id=request.getParameter("id");
myCart=(MyCart)request.getSession().getAttribute("myCart");
               //购买书的方法2
          myCart.addBook2(id);
          BookService bookService = new BookService();
          ArrayList al = bookService.getAllBooks();
          request.setAttribute("books", al);
          request.getRequestDispatcher("/WEB-INF/hall.jsp").
forward(request, response);
          return;
          }else if("deleteBook".equals(type)){
               //接收用户想删除的书号
               String id=request.getParameter("id");
               //取出购物车，并把这本书从这个购物车里删除
               myCart =(MyCart)request.getSession().
getAttribute("myCart");
               myCart.deleteBook(id);
          }else if("updateBook".equals(type)){
               myCart =(MyCart)request.getSession().
getAttribute("myCart");  //得到用户希望更新的书号和数量
     String []id=request.getParameterValues("id");
     String []num=request.getParameterValues("booknum");
                    //拿到了要更新的书号和对应的数量，更新整个购物车
     for(int i=0;i<id.length;i++){
          myCart.updateBook(id[i], Integer.parseInt(num[i]));    }

          }else if("showMyCart".equals(type)){
myCart =(MyCart)request.getSession().getAttribute("myCart");
               //把要显示的总价格，放入到request域中，准备显示
     }else if("clearBook".equals(type))
          {
myCart =(MyCart)request.getSession().getAttribute("myCart");
          myCart.clearBook();
          }
          //把要显示的总价格，放入到request域中，准备显示
          request.setAttribute("totalPrice", myCart.getTotalPrice());
          request.setAttribute("bookList", myCart.showMyCart());
          //跳转到"显示我的购物车"页面去
```

```
            request.getRequestDispatcher("/WEB-
INF/showMyCart.jsp").forward(request, response);
        }
public void doPost(HttpServletRequest request, HttpServletResponse
response)
                throws ServletException, IOException {
            this.doGet(request, response);
        }
}
```

（11）在购物页面点击购买若干书籍，查看购物车，运行效果如图 3-3 所示。

我的购物车

BookID	书名	价格	出版社	数量	删除
3	ASP.NET项目开发案例全程实录(第2版)	74.000	清华大学出版社	1 本	删除
2	Java项目开发案例全程实录(第2版)	69.800	清华大学出版社	1 本	删除
1	jsp应用开发详解	59.800	电子工业出版社	1 本	删除
更新购物车					
总价格：203.6元					

清空购物车提交订单

图 3-3　查看“我的购物车”界面

（12）可以删除购物车中的商品，也可以清空购物车。

四、 出现的问题及解决方案

1、HashMap 编程时注意取出 hm 中的键值，可以得到一个 Set 集合，既然是 Set 集合，那么就可以使用 Collection 接口的 iterator()方法得到该集合的迭代器。

2、index.jsp 页面的跳转问题，使用 forword 转向。

3、改写 SQLHelper 封装类中的查询方法。