

## 样题

### 一、概念综合题（本题满分 30 分）

已知 register.html 运行效果如图，请据此回答网页开发技术相关知识。

**用户注册**

---

用户名:	<input type="text"/>
密码:	<input type="password"/>
密码确认:	<input type="password"/>
昵称:	<input type="text"/>

- (1) 该前台静态网页开发需采用哪三种技术，说明各自含义。
- (2) 本页面需通过表单向后台服务器提交数据，表单对应哪个 HTML 标记？本页面需要用到什么表单元素？写出对应的 HTML 标记。
- (3) 假设服务器处理本表单的页面为 register.jsp，给出对应表单元素的核心代码。
- (4) 提交的注册数据若需存入数据库，需要采用何种技术？简要说明其含义。
- (5) JavaWeb 开发技术中的两种开发模型是什么？本页面适合采取哪种模型？

参考答案：

- (1) 静态网页开发技术需采用的三种技术：HTML+JavaScript+CSS。

HTML 是英文 Hyper Text Markup Language 的缩写，中文译为“超文本标记语言”，其主要作用是通过 HTML 标记对网页中的文本、图片、声音等内容进行描述。

JavaScript 是 Web 中一种功能强大的脚本语言，被设计为向 HTML 页面增加交互性，常用来为网页添加各式各样的动态功能，它不需要进行编译，直接嵌入在 HTML 页面中，就可以把静态的页面转变成支持用户交互并响应事件的动态页面。

CSS 是 Cascading Style Sheet 的缩写，译作“层叠样式表单”，是用于(增强)控

制网页样式并允许将样式信息与网页内容分离的一种标记性语言。

## (2) 表单标记<form></form>

本页面的表单元素：单行文本框，密码框、密码框、单行文本框、提交按钮、重置按钮。

```
<input type="text" name="username">  
<input type="password" name="password">  
<input type="password" name="password1">  
<input type="text" name="nickname">  
<input type="submit" value="注册" name="b1">  
<input type="reset" value="重置" name="b2">
```

## (3) <form action="register.jsp" method="post">...</form>

(4) 页面数据若需存入数据库，需要采用 JDBC 技术。JDBC 的全称是 Java 数据库连接 (Java Database Connectivity)，它是一套用于执行 SQL 语句的 Java API。应用程序可通过这套 API 连接到关系数据库，并使用 SQL 语句来完成对数据库中数据的查询、更新和删除等处理。

(5) JavaWeb 开发技术中的两种开发模型分别是 Model1 和 Model2。本页面适合采取 Model1，因为比较简单。

Model1 采用 JSP+JavaBean 技术开发 Web 应用。其中，JSP 实现页面显示、业务逻辑和流程控制，数据处理由 JavaBean 完成。适合快速和小规模的应用开发。

## 二、代码分析题（本题满分 30 分）

1. (10 分) 以下为 test 项目下 basicjsp.jsp 文件的 JSP 代码：

```
<%@ page contentType="text/html;charset=GBK" import="java.util.*"%>  
<html>  
<head><title>JSP基本语法</title></head>  
<body>  
<%! long i=0;%>  
<%! String name="王红";%>  
<%! public String sayHello(String who){  
    if(Calendar.getInstance().get(Calendar.AM_PM) == Calendar.AM){  
        return who + "上午好！"  
    }  
}
```

```

}else{
    return who + "下午好！ "
}
%>
<%    i++;  %>
<%-- 表达式的使用--%>
<h2 align="center">
    <%=sayHello(name)%><br/>
    您是本站的第<%=i%>位访客。
</h2>
</body>
</html>

```

请回答以下问题：

- (1) 请分别指出代码中的 JSP 代码段、JSP 声明及 JSP 表达式。
- (2) <%! long i=0;%>代码中的 i 是页面内的局部变量还是全局变量吗？
- (3) 请指出代码中的指令元素，并说明对应属性的作用。
- (4) 在 tomcat 中运行该 jsp 请求，需在浏览器地址栏输入什么请求？

参考答案：

- (1) 代码中的 JSP 代码段：

```
<%    i++;  %>
```

代码中的 JSP 声明：

```

<%! long i=0;%>
<%! String name="王红";%>
<%! public String sayHello(String who){
    if(Calendar.getInstance().get(Calendar.AM_PM) == Calendar.AM){
        return who + "上午好！ "
    }else{
        return who + "下午好！ "
    }
}

```

代码中的表达式：

```

<%=sayHello(name)%><br/>中的<%=sayHello(name)%>
您是本站的第<%=i%>位访客。中的<%=i%>

```

- (2) <%! long i=0;%>代码中的 i 是页面内的全局变量。
- (3) 代码中的指令元素及对应属性的作用：

```
<%@ page contentType="text/html; charset=GBK" import="java.util.*"%>
中的 page 指令, contengType 属性用于指定 JSP 页面输出内容的类型和字符编码方式, import 属性用于导入所需的 Java 类。
```

(5) 在 tomcat 中运行该 jsp 请求, 浏览器地址栏应为:

http://localhost:8080/test/basicjsp.jsp

2. (5 分) 请利用 response 对象设置页面自动加载的代码, 要求实现浏览器 5 秒后自动加载新页面 "http://www qlu.edu.cn/"。

```
<% response.setHeader("Refresh", "5;url=http://www qlu.edu.cn/"); %>
```

3. (5 分) 请利用 response 对象实现页面重定向功能, 将页面重定向到 register.jsp 页面, 同时传递参数值 id=rose。

```
<% response.sendRedirect("register.jsp?id=rose"); %>
```

4. (10 分) 假设已有描述个人信息的 JavaBean beans.Book, 现在要使用 JavaBean 的 3 个标准动作元素来访问 JavaBean, 请据此补全 book.jsp 的代码。

```
<%@ page pageEncoding="GBK" %>
<html>
<head>
    <title>使用动作元素访问JavaBean</title>
</head>
<body>
    <%--代码1 在网页中创建beans.Pook实例, 作用范围为page--%>
    <%--代码2 一次性利用传入的输入参数为JavaBean的同名属性赋值--%>
```

根据传入参数更改的属性值为: <br/>

书名: <%--代码3 读取并显示JavaBean的bookName属性值--%>

作者: <%--代码4 读取并显示JavaBean的author属性值--%>

价格: <%--代码5 读取并显示JavaBean的price属性值--%>

出版社: <%--代码6 读取并显示JavaBean的press属性值--%>

```
</body>
</html>
```

参考答案:

```
<%--代码1 在网页中创建beans.Pook实例, 作用范围为page--%>
```

```

<jsp:useBean id="book" class="beans.Book" scope="page" />
<%--代码2 一次性利用传入的输入参数为JavaBean的同名属性赋值--%>
<jsp:setProperty name="book" property="*" />
根据传入参数更改的属性值为: <br/>
书名: <%--代码3 读取并显示JavaBean的bookName属性值--%><jsp:getProperty
name="book" property="bookName" /><br/>
作者: <%--代码4 读取并显示JavaBean的author属性值--%><jsp:getProperty
name="book" property="author" />
价格: <%--代码5 读取并显示JavaBean的price属性值--%><jsp:getProperty name="book"
property="price" />
出版社: <%--代码6 读取并显示JavaBean的press属性值--%><jsp:getProperty
name="book" property="press" />

```

### 三、程序设计题（本题满分 40 分）

1. 请利用 JDBC 实现图书管理系统的图书添加功能。静态页面如下：

**添加图书信息**

书名:	<input type="text"/>
作者:	<input type="text"/>
出版社:	<input type="text"/>
价格:	<input type="text"/>
<input type="button" value="添加"/> <input type="button" value="重置"/>	

要求：

- (1) 编写上面的静态 HTML 页面 add.html (不要求写 css)。(5 分)
  - (2) 给出数据库中对应表 bookinfo 的 SQL 创建语句。(5 分)
  - (3) 使用 JDBC 编写 JSP 页面 add.jsp, 将图书数据存入数据库中。(10 分)
- (1) 添加图书页面 add.html

```

<html>
<head>
<title>添加图书信息</title>
</head>
<body>
<h2 align="center">添加图书信息</h2>
<form name="form1" action="add.jsp" method="post">
<table align="center" width="30%" border="1">
<tr><th width="30%">书名: </th>
<td><input type="text" name="bookname"></td></tr>
<tr><th>作者: </th>
<td><input type="text" name="author"></td></tr>

```

```

<tr><th>出版社: </th>
<td><input type="text" name="press"></td></tr>
<tr><th>价格: </th>
<td><input type="text" name="price"></td></tr>
<tr><th colspan="2">
<input type="submit" value="添加">
<input type="reset" value="重置"></th></tr>
</table>
</form>
</body>
</html>

```

## (2) 所设计表结构对应的 sql 语句

```

create table bookinfo(
    id int(10) unsigned not null auto_increment,
    bookname varchar(100) not null default '',
    author varchar(20) not null default '',
    press varchar(50) not null default '',
    price float not null default 0,
    primary key(id)
)engine=InnoDB default charset=gbk;

```

## (3) add.jsp

```

<%@ page pageEncoding="GBK" import="java.sql.*"%>
<%
    request.setCharacterEncoding("GBK");
    String bookname = request.getParameter("bookname");
    String author = request.getParameter("author");
    String press = request.getParameter("press");
    String price = request.getParameter("price");

    Class.forName("com.mysql.cj.jdbc.Driver"); //需将驱动jar包拷到Web项目中
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/book", "root",
    "root");

    String sql = "insert into bookinfo values(null,?, ?, ?, ?)";
    PreparedStatement pstmt = con.prepareStatement(sql);
    pstmt.setString(1, bookname);
    pstmt.setString(2, author);
    pstmt.setString(3, press);
    pstmt.setFloat(4, Float.parseFloat(price));
    int result = pstmt.executeUpdate();
    String msg = "添加失败, 点击确定跳转到图书列表页!";
    if (result == 1) {
        msg = "添加成功, 点击确定跳转到图书列表页!";
    }
    pstmt.close();
    con.close();
%>
<script>window.alert('<%=msg%>
');
</script>

```

2.对于图书管理系统，若利用 JavaBean 技术，可使系统更易于维护和开发。

请给出针对图书信息的 JavaBean 代码（假设：（1）连接数据库的操作已封装

到 JavaBean util.DBUtil 中，如上课所讲；（2）所有的 JavaBean 都在 beans 包中）。(20 分)

```
package beans;
import java.util.List;
import java.util.Map;
import util.DBUtil;
public class Book {
    private String id;
    private String bookname;
    private String author;
    private String press;
    private String price;
    private DBUtil db;
    public Book() { db = new DBUtil(); }
    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 getPress() { return press; }
    public void setPress(String press) { this.press = press; }
    public String getPrice() { return price; }
    public void setPrice(String price) { this.price = price; }
    //读取所有图书信息
    public List getAllBooks(){
        List books = null;
        String sql = "select * from bookinfo";
        books = db.getList(sql, null);
        return books;
    }
    //添加图书信息
    public int addBook(){
        int result = 0;
        String sql = "insert into bookinfo values(null,?,?,?,?)";
        String[] params = {bookname,author,press,price};
        result = db.update(sql, params);
        return result;
    }
    //通过图书 id 读取图书信息
    public Map getBook(){
        Map book = null;
        String sql = "select * from bookinfo where id=?";
        String[] params={id};
```

```
book = db.getMap(sql, params);
return book;
}
//修改图书信息
public int updateBook(){
    int result = 0;
    String sql = "update bookinfo set bookname=?,author=?,press=?,price=? where id=?";
    String[] params = {bookname,author,press,price,id};
    result = db.update(sql, params);
    return result;
}
//删除图书信息
public int delBook(){
    int result = 0;
    String sql = "delete from bookinfo where id=?";
    String[] params = {id};
    result = db.update(sql, params);
    return result;
}
}
```