**2022-2023-2 Web应用开发技术JSP-补考2**

3抽一：

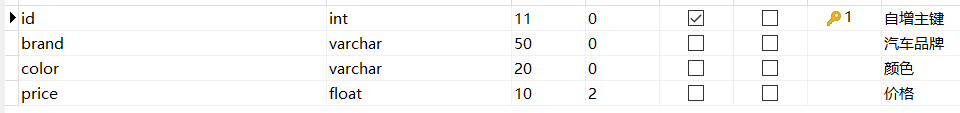
1. 假定一个Web项目，其项目的Context Root（根路径）为：localhost:8080/javaweb/ 按下列要求在该项目中定义一个Servlet

1. Servlet的类名自定义，假定可以以下url访问该Servlet，“http://host:8080/javaweb/Hello?name=xxx” ，请用注解方式配置该Servlet的url。
2. 该Servlet处理get方式请求；
3. 该Servlet返回HTML格式的内容，并支持中文输出；  
   访问该Servlet将根据参数name的不同会输出相应的：“你好！xxx!”。  
   比如name=张三，将输出“你好！张三！”，name=Tomcat则会输出“你好！Tomcat！”

@WebServlet("/Hello")  
public class HelloServlet extends HttpServlet {  
  
 private static final long *serialVersionUID* = 1L;  
 @Override  
 protected void doGet(HttpServletRequest *request*, HttpServletResponse *response*) throws ServletException, IOException {  
  
 *response*.setContentType("text/html;charset=UTF-8");  
  
 String *name* = *request*.getParameter("name");  
  
 PrintWriter *out* = *response*.getWriter();  
  
 *out*.print("<!DOCTYPE html>");  
 *out*.print("<html>");  
 *out*.print("<head>");  
 *out*.print("<title>Hello</title>");  
 *out*.print("</head>");  
 *out*.print("<body>");  
 *out*.println("<h2>你好！"+ *name* + "!</h2>");  
 *out*.print("</body>");  
 *out*.print("</html>");  
 }  
}

2. 假定：本地服务器（127.0.0.1）上有一个数据库系统（数据库自行设选择），  
用户名为：root  
密码为：root#123  
数据库名为：webexam

* 也可以是一个SQLite数据库，SQLite数据库文件是webexam.db
* 该数据库中有一个名为car的表，存放有汽车数据  
  car表的设计如下：



现要求：

1. 根据上表定义一个实体类Car，该类的成员变量参考上面的数据库表car；

public class Car {  
 private Integer id;  
  
 private String brand;  
 private String color;  
 private Double price;  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer *id*) {  
 this.id = *id*;  
 }  
  
 public String getBrand() {  
 return brand;  
 }  
  
 public void setBrand(String *brand*) {  
 this.brand = *brand*;  
 }  
  
 public String getColor() {  
 return color;  
 }  
  
 public void setColor(String *color*) {  
 this.color = *color*;  
 }  
  
 public Double getPrice() {  
 return price;  
 }  
  
 public void setPrice(Double *price*) {  
 this.price = *price*;  
 }  
}

1. 在下面的CarDao类中定义一个方法，要求该方法实现查询car表中的所有汽车数据，返回并返回查询到的数据。

//定义实体类Car

/\*\*

\* 自定义一个方法，该方法要求实现从car表中查询所有的汽车数据，并封装为Car对象集合返回

\* @return 查询到数据则返回一个List<Car>，如果没有数据则返回一个空的List<Car>对象

\*/

public class CarDao {  
   
 private String jdbcUri = "jdbc:mysql://127.0.0.1/webexam";  
 private String username = "root";  
 private String password = "root#123";  
  
 public CarDao(){  
 try {  
 Class.*forName*("com.mysql.jdbc.Driver");  
 } catch (ClassNotFoundException *e*) {  
 throw new RuntimeException(*e*);  
 }  
 }  
 // 查询所有汽车数据  
 public List<Car> findAllCars(){  
  
 List<Car> *list* = new ArrayList<Car>();  
  
 Connection *connection* = null;  
 Statement *statement* = null;  
  
  
 try {  
  
 *connection* = DriverManager.*getConnection*(jdbcUri, username, password);  
 *statement* = *connection*.createStatement();  
 ResultSet *result* = *statement*.executeQuery("SELECT \* FROM car");  
 Car *car* = new Car();  
 if(*result*.next()){  
 *car*.setId(*result*.getInt("id"));  
 *car*.setBrand(*result*.getString("brand"));   
 *car*.setColor(*result*.getString("color"));   
 *car*.setPrice(*result*.getDouble("price"));   
 *list*.add(*car*);  
 }  
  
 return *list*;  
 } catch (SQLException *e*) {  
 throw new RuntimeException(*e*);  
 }finally {  
 // 关流  
 try {  
 *connection*.close();  
 *statement*.close();  
 } catch (SQLException *e*) {  
 throw new RuntimeException(*e*);  
 }  
  
 }  
  
 }  
}

3. 在JavaWeb中，JSP页面主要用于展示数据，通常是展示从数据库中查询到的数据。

* 现假定有一个展示一条汽车数据的JSP页面，其展示一条汽车信息的HTML代码如下：
* <h1>汽车信息：</h1>
* <p>品牌：</p>
* <p>颜色：</p>

<p>价格：</p>

* 要求1：定义一个显示一条汽车数据的JSP页面，在该页面中使用上面的HTML代码来显示汽车数据，该数据来自于数据库，配置page指令，能正常显示中文；
* 假定在某个Servlet中调用CarDao类查询到一条汽车数据，并将查询结果的Car对象存入到request对象中request.setAttribute("car",car)，并跳转该JSP页面。现要求在该JSP页面取出request对象中的Car对象，并将  
  该条汽车数据显示在该JSP页面；如果查询结果为null，则输出“未找到对应的汽车数据”。
* 在该JSP页面可使用JSTL和EL，也可使用JSP内置对象与脚本元素实现上述功能。

JSTL ：

<%@ **page** language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  
<%@ **taglib** prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>  
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>汽车信息</title>  
</head>  
<body>  
<h1>汽车信息：</h1>  
 <**c:choose**>  
 <**c:when** test="${car.id!=null}">  
 <p>品牌：${car.brand} </p>  
 <p>颜色：${car.color} </p>  
 <p>价格：${car.price} </p>  
 </**c:when**>  
 <**c:otherwise**>  
 <p>未找到对应的汽车数据</p>  
 </**c:otherwise**>  
 </**c:choose**>  
</body>  
</html>

也可以用JSP:

<%@ **page** language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  
<%@ **page** import="org.example.POJO.Car" %>  
<%  
Car car = (Car)*request*.getAttribute("car");  
%>  
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>展示汽车信息</title>  
</head>  
<body>  
<h1>汽车信息：</h1>  
<p>品牌：<%= *car*.getBrand() %></p>  
<p>颜色：<%= *car*.getColor() %></p>  
<p>价格：<%= *car*.getPrice() %></p>  
<% if (*car* == null) { %>  
<p>未找到对应的汽车数据</p>  
<%  
}  
%>  
</body>  
</html>

**2021-2022-2 Web应用开发技术（JSP） 期末考试-第一场**

* [1.](https://pintia.cn/problem-sets/1524911823390928896/overview) 有一个登录页面（login.html），其登录表单的HTML代码如下：
* <form method="post" action="login">
* 用户名：<input name="name" type="text">
* 密码： <input name="passwd" type="password">
* <br>
* <input value="登录" type="submit">

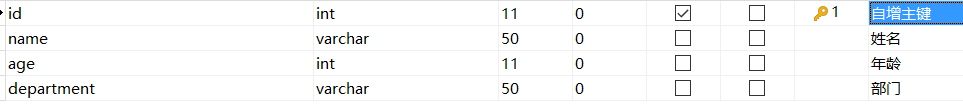
</form>

要求：定义一个Servlet，响应上述的登录页面，在该Servlet中认证用户名和密码，认证通过后设置一个Cookie，并通过响应对象将Cookie发送到客户端，该Servlet具体要求如下：  
1、用注解方式定义Servlet的UrlPatterns，其地址见login.html页面；  
2、获取表单中的用户名密码；  
3、验证登录用户提交的用户名是否为Zhangsan，密码为123456，如果验证通过，则将用户名添加到Cookie中，再将请求重定向到另一个Servlet（名称为UserServlet，地址为：/home.view，该Servlet无需定义）中；  
3、如果用户名和密码不正确，则将请求重定向到login.html；  
4、提交该Servlet的完整代码，Servlet的类名自定义。

@WebServlet("/Hello")  
public class doLogin extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {  
 String userName = req.getParameter("name");  
 String password = req.getParameter("password");  
 if("Zhangsan".equals(userName)&& "123456".equals(password)){  
 Cookie cookie = new Cookie("userName",userName);  
 resp.addCookie(cookie);  
 resp.sendRedirect("/home.view");  
 }  
 else{  
 resp.sendRedirect("login.html");  
 }  
 }  
}

2. 服务器（IP为：192.168.1.1）上有一个数据库系统（数据库自行设选择），  
用户名为：root  
密码为：jspexam  
数据库名为：jweb

* 也可以是一个SQLite数据库，SQLite数据库文件是jweb.db
* 该数据库中有一个名为employee的表，存放有职员数据  
  employee表的设计如下：



现要求：

1. 根据上表定义一个实体类Employee，该类的成员变量参考上面的数据库表employee；

public class Employee{  
 private int id;  
 private String name;  
 private int age;  
 private String department;  
 public void setId(int id){  
 this.id = id;  
 }  
 public int getId(){  
 return id;  
 }  
 public void setName(String name){  
 this.name = name;  
 }  
 public String getName(){  
 return name;  
 }  
 public void setAge(int age){  
 this.age = age;  
 }  
 public int getAge(){  
 return age;  
 }  
 public void setDepartment(String department){  
 this.department = department;  
 }  
 public String getDepartment(){  
 return department;  
 }  
}

1. 在下面的EmployeeDao类中定义一个方法，要求该方法实现向employee表中插入一条职员数据，返回插入的结果。

public class EmployeeDao {  
  
 private String jdbcURi = "jdbc:mysql://192.168.1.1/jweb";  
 private String username = "root";  
 private String password = "jspexam";  
  
 public EmployeeDao(){  
 try {  
 Class.*forName*("com.mysql.jdbc.driver");  
 } catch (ClassNotFoundException *e*) {  
 throw new RuntimeException(*e*);  
 }  
 }  
  
 public int insert(Employee *employee*){  
  
 int *result* = 0;  
 Connection *connection*=null;  
 PreparedStatement *pstmt*=null;  
 String *sql*= "INSERT INTO employee (name, age, department) VALUES (?, ?, ?)";  
  
 try {  
 *connection* = DriverManager.*getConnection*(jdbcURi, username, password);  
 *pstmt* = *connection*.prepareStatement(*sql*);  
   
 *pstmt*.setString(1, *employee*.getName());  
 *pstmt*.setInt(2, *employee*.getAge());  
 *pstmt*.setString(3, *employee*.getDepartment());  
  
 *result* = *pstmt*.executeUpdate();  
  
 return *result*;  
 } catch (SQLException *e*) {  
 throw new RuntimeException(*e*);  
 }finally {  
 try {  
 *pstmt*.close();  
 *connection*.close();  
 } catch (SQLException *e*) {  
 throw new RuntimeException(*e*);  
 }  
 }  
  
 }  
}

1.定义一个Servlet，该Servlet中读取客户端的Cookie，并验证是否存在一个存储有用户名的cookie，具体要求如下：  
1、该Servlet的类名和URL都自行定义，采用注解方式定义；  
2、在该Servlet中判断是否存在cookie，如果有，就遍历所有cookie；  
3、判断cookie中是否包含一个名为"user"和值为"Alice"的cookie；  
4、如果有，就将"Alice"存入到session中，存储的属性名同样为"user"，再forward到UserViewServlet(路径为"user.view")；  
5、如果没有任何cookie或cookie中没满足3中的cookie时，重定向到login.html。

提交该Servlet的完整代码。

@WebServlet("/check-cookie")  
public class CookieServlet extends HttpServlet {  
  
 private static final long *serialVersionUID* = 1L;  
 @Override  
 protected void doGet(HttpServletRequest *request*, HttpServletResponse *response*) throws ServletException, IOException {  
  
 Cookie[] *cookies* = *request*.getCookies();  
  
 if(*cookies*!=null){  
 for (Cookie *cookie* : *cookies*) {  
 if(*cookie*.getName().equals("user")&& *cookie*.getValue().equals("Alice")){  
 HttpSession *session* = *request*.getSession();  
 *session*.setAttribute("user","Alice");  
 *request*.getRequestDispatcher("/user.view").forward(*request*,*response*);  
 }  
 }  
 }  
  
 *response*.sendRedirect("login.html");  
 }  
}

2020-2021-1 Web应用开发技术JSP-期末考试-第一场

1.有一个sayHello页面中有一个表单HTML代码如下：

<h2>打招呼</h2>

<form action="sayHello.view" method="post">

<input type="text" name="userName"><br>

<input type="submit" value="打招呼">

</form>

该页面显示可参考如下所示：

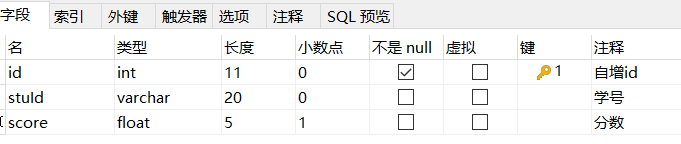


现按下面的要求定义一个Servlet：

* 要求1：定义一个Servlet，sayHello.html的表单数据提交到这个Servlet，根据sayHi页面表单中的设置来配置本Servlet的访问路径，要求采用注解方式配置访问路径；
* 要求2：本Servlet中获取sayHello.html表单提交的用户名参数，并产生HTML格式的响应，输出内容为：“您好：***”，其中***就是sayHello页面提交的用户名；例如：在sayHello页面表单填写"张三"并提交，则输出“您好，张三”，本Servlet能正确显示中文用户名；
* 要求3：在输出“您好：\*\*\*”的信息同时，还要输出当前的时间，并且在5秒钟后自动跳转到sayHello.html页面

@WebServlet("/sayHello")  
public class sayHello extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 String userName = request.getParameter("userName");  
 PrintWriter out = response.getWriter();  
 out.print("<!DOCTYPE html>");  
 out.print("<head>");  
 out.print("<meta charset=\"UTF-8\">");  
 out.print("<title>satHello</title>");  
 out.print("</head>");  
 out.print("<body>");  
 out.print("你好："+userName);  
 out.print("</body>");  
 out.print("</html>");  
 resp.setHeader("refresh","5;url=sayHello.html");  
 }  
}

2. 本地服务器（127.0.0.1）上有一个数据库系统（数据库自行设置，但需要匹配驱动），  
用户名为：root  
密码为：root  
数据库名为：jspexam  
该数据库中有一个名为examscore的表，存放有考试成绩数据  
examsocre表的设计如下：



现要求：

1. 根据上表定义一个实体类ExamScore，该类的成员变量参考上面的数据库表examscore；

public class ExamScore{  
 private int id;  
 private String stuId;  
 private float score;  
 public void setId(int id){  
 this.id = id;  
 }  
 public int getId(){  
 return id;  
 }  
 public void setStuId(String stuId){  
 this.stuId = stuId;  
 }  
 public String getStuId(){  
 return stuId;  
 }  
 public void setScore(float score){  
 this.score = score;  
 }  
 public float getScore(){  
 return score;  
 }  
}

1. 在下面的ExamScoreDao类中定义一个方法，要求该方法实现根据学号查询到某个学生的成绩数据

public class ExamScoreDao{  
 private String jdbcUri = "jdbc:mysql://localhost:3306/jspexam";  
 private String userName = "root";  
 private String password = "root";  
 public ExamScoreDao(){  
 try{  
 Class.*forName*("com.mysql.jdbc.Driver");  
 }  
 catch(Exception e){  
 throw new RuntimeException(e);  
 }  
 }  
 public ExamScore findByStuId(stuId){  
 Connection conn = null;  
 Statement statement = null;  
 try{  
 conn = DriverManager.*getConnection*(jdbcUri,userName,password);  
 statement = conn.createStatement();  
 ResultSet result = statement.executeQuery("SELECT \* FROM examscore WHERE stuId = "+stuId+";");  
 ExamScore examScore = new ExamScore();  
 if(result.next()){  
 examScore.setId(result.getInt("id"));  
 examScore.setStuId(result.getString("stuId"));  
 examScore.setScore(result.getFloat("score"));  
 return examScore;  
 }  
 else {  
 return null;  
 }  
 }  
 catch(Exception e){  
 throw new RuntimeException(e);  
 }  
 finally{  
 try {  
 conn.close();  
 statement.close();  
 }  
 catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
}

1. public class ExamScoreDao {
2. /\*\*
3. \* 自定义一个方法，该方法要求实现根据学号从examscore表中查询出学生成绩，并封装为ExamScore对象返回
4. \* @return 查询到数据则返回一个ExamScore对象，如果没查询到则返回null
5. \*/

}

3. 参考题目ExamScoreDao类，假定有一个页面用于显示成绩查询的结果，显示成绩信息的HTML代码如下：

<h1>成绩查询结果：</h1>

<p>学号：</p>

<p>分数：</p>

* 要求1、定义一个显示成绩查询结果的JSP页面，在该页面中使用上面的HTML代码来显示成绩查询结果，配置page指令，能正常显示中文；
* 要求2、假定在某个Servlet中进行了成绩查询，并将查询结果的ExamScore对象存入到request对象中request.setAttribute("score",score)，并跳转该JSP页面。现要求在该JSP页面取出request对象中的ExamScore对象，并将学号和成绩显示在指定的位置；如果查询结果为null，则输出“未找到对应的成绩”
* <%@ **page** contentType = "text/html;charset = UTF-8" language = "java"%>  
  <%@ **taglib** prefix = "c" uri = "http://java.sun.com/jsp/jstl/core"%>  
  <!DOCTYPE html>  
  <html>  
  <head>  
   <meta charset = "UTF-8">  
   <title>成绩查询</title>   
  </head>  
  <body>  
  <h1>成绩查询结果：</h1>  
  <**c:choose**>  
   <**c:when** test="${score.getStuId != null}">  
   <p>学号：${score.getStuId()}</p>  
   <p>分数：${score.getScore()}</p>  
   </**c:when**>  
   <**c:otherwise**>  
   <p>未找到对应的成绩</p>  
   </**c:otherwise**>  
  </**c:choose**>  
  </body>  
  </html>