



成绩

中国农业大学

课程论文

(2018 -2019 学年 夏 季学期)

论文题目: 中国农业大学本科课程论文

课程名称: 网络程序设计

任课教师: 雷宏洲

班 级: 计算机 182

学 号: 2018308250122

姓 名: 贾云嵩

一、 基础概念题：

1. 说明 Java 事件的处理机制以及如何响应和处理事件？请举例说明。

(1) 定义方法：通过调用事件监听器的方法，并在定义所产生事件类型的事件子类的实例中，将事件从事件源传播到事件监听器对象，实现对应的事件处理器的功能。

(2) 响应处理：通过特定动作产生特定事件，特定事件传播到对应的事件监听器，触发对应的事件处理器。即 Pub/Sub 模式。

(3) 例子 - AWT 窗体（环境：Eclipse - 4.7.1a / JavaSE - 1.8）：

➤ 定义窗体：

```
JFrame w=new JFrame();
```

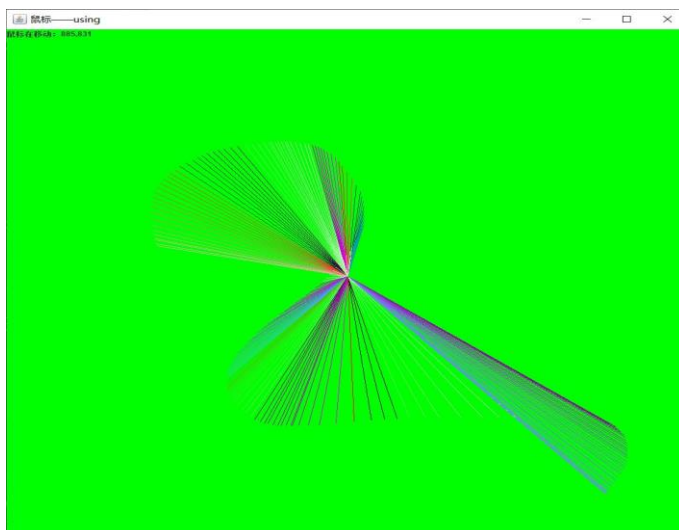
➤ 实现事件处理器：

```
MouseMotionListener m=new MouseMotionListener() {  
    public void mouseMoved(MouseEvent e) {  
        String str=String.format("鼠标在移动: %d,%d", e.getX(),e.getY());  
        info.setText(str);  
    }  
    public void mouseDragged(MouseEvent e) {  
        String str=String.format("鼠标在拖动: %d,%d", e.getX(),e.getY());  
        info.setText(str);  
        Graphics g=w.getGraphics();  
        g.setColor(new Color(e.getX()%255,  
            e.getY()%255,(e.getX()+e.getY())%255));  
        g.drawLine(w.getWidth()/2, w.getHeight()/2, e.getX(), e.getY());  
    }  
};
```

➤ 将事件处理器添加到窗体的事件监听器：

```
w.addMouseMotionListener(m);
```

➤ 运行程序，拖动鼠标，效果如下：



2. 详细说明 JSP/Servlet 的工作原理？

(1) 当客户端浏览器向服务器请求一个 Servlet 时，服务器收到该请求后：

- 首先到容器中检索与请求匹配的 Servlet 实例是否已经存在。若不存在，则 Servlet 容器负责加载并实例化出该类 Servlet 的一个实例对象，接着容器框架负责调用该实例的 `init()` 方法来对实例做一些初始化工作。然后 Servlet 容器运行该实例的 `service()` 方法。
- 若 Servlet 实例已经存在，则容器框架直接调用该实例的 `service()` 方法。`service()` 方法在运行时，自动派遣运行与用户请求相对应的 `doXX()` 方法来响应用户发起的请求。通常，每个 Servlet 类在容器中只存在一个实例，每当请求到来时，则分配一条线程来处理该请求。

(2) 当客户端浏览器向服务器请求一个 JSP 页面时，服务器收到该请求后：

- 首先检查所请求的这个 JSP 文件内容（代码）是否已经被更新，或者是否是 JSP 文件创建后的第一次被访问，如果是，那么，这个 JSP 文件就会在服务器端的 JSP 引擎作用下转化为一个 Servlet 类的 Java 源代码文件。紧接着，这个 Servlet 类会在 Java 编译器的作用下被编译成一个字节码文件，并装载到 jvm 解释执行。剩下的就等同于 Servlet 的处理过程了。
- 如果被请求的 JSP 文件内容（代码）没有被修改，那么它的处理过程也等同于一个 Servlet 的处理过程。即直接由服务器检索出与之对应的 Servlet 实例来处理。

3. 解释 Socket 模型的通讯方式和过程？举例说明如何使用 Java Socket 编写简单的 C/S 模式程序？

(1) Socket 模型的通讯方式和过程：

- 通讯方式：TCP/IP 协议
- 通讯过程：
 - 服务器分配一个端口号。如果客户请求一个连接，服务器使用 `accept()` 方法打开 socket 连接。
 - 客户在 host 的 port 端口建立连接。
 - 服务器和客户使用 `InputStream` 和 `OutputStream` 进行通信。
 - 当通信结束，可以关闭客户机或者服务器。

(2) 例子（环境：Eclipse - 4.7.1a / JavaSE - 1.8）：

- 实现客户机：

```
package socketTest;

import java.io.*;
import java.net.*;

public class JTimeClient {
```

```

public static void main(String[] args) {
    Socket s=null;
    InputStreamReader in=null;
    OutputStreamWriter out=null;
    try {
        //客户机在host的port端口建立连接
        s=new Socket("localhost",8080);
        System.out.print("localhost:8080 connected.....");

        //使用InputStream和OutputStream进行通信
        out=new OutputStreamWriter(s.getOutputStream());
        int id=(int)(Math.random()*10);
        String request="Client"+id;
        out.append(request);
        out.flush();
        System.out.println("向服务器发送的是: "+request);
        in =new InputStreamReader(s.getInputStream());
        char[] buf=new char[100];
        in.read(buf, 0, buf.length);
        String response=new String(buf);
        System.out.print("接收的服务响应是: "+response);
        System.out.println("其编码是: "+in.getEncoding());

    }catch(IOException e) {}
    finally {

        //关闭通信及客户机
        try {
            if(in!=null) in.close();
            if(out!=null) out.close();
            if(s!=null) s.close();
        }catch(IOException e) {}

    }
}
}

```

➤ 实现时间服务器:

```

package socketTest;

import java.io.*;
import java.net.*;
import java.time.*;

```

```

public class JServerSocketTest {
    static public int i=0;

    public static void main(String[] args) {
        try {
            //服务器分配一个端口号
            ServerSocket ss=new ServerSocket(8000);
            System.out.println("HelloServer started at 8000.....");
            while(i==0) {

                //使用accept()方法打开socket连接
                Socket s=ss.accept();
                System.out.println(s.getInetAddress()+":"+s.getPort());

                //timeService(s);
                TimeService a=new JServerSocketTest().new TimeService(s);
                Thread t=new Thread(a);
                t.start();

                s.close();
            }
            ss.close();
        }catch(IOException e) {}
    }

    /**
     * 多线程
     * @author 贾云嵩
     *
     */
    public class TimeService implements Runnable{
        private Socket s=null;
        TimeService(Socket user){s=user;}
        public void run() {
            i++;
            InputStreamReader in=null;
            OutputStreamWriter out=null;
            try {

                //使用InputStream和OutputStream进行通信
                in=new InputStreamReader(s.getInputStream());
                char[] buf=new char[7];
                in.read(buf,0,buf.length);
                String request=new String(buf);
                System.out.print("客户端的请求是: "+request);
                System.out.print("其编码是: "+in.getEncoding());
            }
        }
    }
}

```

```

        out=new OutputStreamWriter(s.getOutputStream());
        LocalDateTime t=LocalDateTime.now();
        String response=String.format("Hello,%s!%s", request,t);
        out.write(response,0,response.length());
        out.flush();
    }catch(IOException e) {}
    finally {
        try {
            if(in!=null) in.close();
            if(out!=null) out.close();
            if(s!=null) s.close();
        }catch(IOException e) {}
    }
}
}
}

```

```

/**
 * 单线程
 * @author 贾云嵩
 *
 */
public static void timeService(Socket s) {
    InputStreamReader in=null;
    OutputStreamWriter out=null;
    try {
        i++;

        //使用InputStream和OutputStream进行通信
        in=new InputStreamReader(s.getInputStream());
        char[] buf=new char[7];
        in.read(buf,0,buf.length);
        String request=new String(buf);
        System.out.print("客户端的请求是: "+request);
        System.out.print("其编码是: "+in.getEncoding());
        out=new OutputStreamWriter(s.getOutputStream());
        LocalDateTime t=LocalDateTime.now();
        String response=String.format("Hello,%s!%s", request,t);
        out.write(response,0,response.length());
        out.flush();
    }catch(IOException e) {}
    finally {
        try {
            if(in!=null) in.close();

```

```

        if(out!=null) out.close();
        if(s!=null) s.close();
    }catch(IOException e) {}
}
}
}

```

➤ 运行服务器，再运行客户机：

■ 服务器（在 NetBeans 运行）：



■ 客户机（在 Eclipse 运行）：



4. 什么是 HTTP 协议？请说明它的内容，特点，数据结构以及相关命令。

(1) 内容：

HTTP 是 Hyper Text Transfer Protocol（超文本传输协议）的缩写，是基于请求与响应模式的、无状态的、应用层的协议，是用于从 WWW 服务器传输超文本到本地浏览器的传送协议。

(2) 特点：

- 支持客户/服务器模式。
- 简单快速：客户向服务器请求服务时，只需传送请求方法和路径。请求方法 GET、HEAD、POST 等。每种方法规定了客户与服务器联系的类型不同。
- 灵活：允许传输任意类型的数据对象。正在传输的类型由 Content-Type 加以标记。
- 无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。

- 无状态：HTTP 协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。

(3) 数据结构：

请求消息：HTTP 请求到服务器的请求消息包括以下格式：请求行（request line）、请求头部（header）、空行和请求数据 4 个部分组成，下图给出了请求报文的一般格式。

- 请求行：

由三个标记组成：请求方法、请求 URI 和 HTTP 版本，它们用空格分隔。

- 消息头（请求头部）：

由域名/值对组成，每行一对，域名和值之间用紧跟的英文冒号（“:”），单空格（SP）分开。消息头用于通知服务器有关于客户端的功能和标识。

- 消息体（请求数据）：

消息体（如果存在）用于携带与请求相关联的数据，例如可以存一些请求需要的参数等。由消息头中的 Content-Length 或 Transfer-Encoding 来指示。消息头里的 Content-Type 说明了数据的传输类型。

响应消息：HTTP 响应也由四个部分组成，分别是：状态行、消息报头、空行、响应正文。

- 状态行：

由三个标记组成，即 HTTP 版本、响应代码和响应描述：

- HTTP 版本：向客户端指明其可理解的最高版本。
- 响应代码：3 位的数字代码，指出请求的成功或失败，如果失败则指出原因。
- 响应描述：为响应代码的可读性解释。

- 响应头标（消息报头）：

像请求头标一样，它们指出服务器的功能，标识出响应数据的细节。

- 响应数据（响应正文）：

HTML 文档和图像等，也就是 HTML 本身。

(4) 相关命令：

- HTTP 请求方法

序号	方法	描述
1	GET	请求指定的页面信息，并返回实体主体。
2	HEAD	类似于get请求，只不过返回的响应中没有具体的内容，用于获取报头
3	POST	向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST请求可能会导致新的资源的建立和/或已有资源的修改。
4	PUT	从客户端向服务器传送的数据取代指定的文档的内容。
5	DELETE	请求服务器删除指定的页面。
6	CONNECT	HTTP/1.1协议中预留给能够将连接改为管道方式的代理服务器。
7	OPTIONS	允许客户端查看服务器的性能。
8	TRACE	回显服务器收到的请求，主要用于测试或诊断。

二、 实践题：

(1). 项目描述： 题目名称及背景概况。

➤ **题目名称：**搭建我国各民族人口性别构成网络查询系统

➤ **背景概况：**

性别结构描述了不同性别人口在总人口中的构成状况，反映了人口群体的自然特征。合理的性别结构不仅是民族人口兴旺发达的重要前提，还是一个民族繁荣发展的必然要求。而性别结构的失衡不仅不利于民族人口自身的和谐发展，还影响着民族经济的繁荣和多民族社会的稳定。

研究各民族性别构成可以间接了解到各民族的有关社会、生活、观念等方面的差异，对促进民族人口自身发展，守护民族经济的繁荣，维持平等、团结、互助、和谐的民族关系的巩固和多民族社会的稳定具有积极影响。

通过现今互联网技术，使得查询与更新信息更为便捷，依托 Java Web 平台与数据库技术，从技术上可以实现我国各民族人口性别构成网络查询系统的构建，实现人口信息的更新与共享，为对此方面数据需求的人员提供更方便的数据查询与修改服务，让数据实现更为广泛与高效的同步与运用，达到方便人口研究的目的。

(2). 项目分析： 项目的目的、用途或需要解决的问题，以及需求。

➤ **目的：**

实现我国各民族人口性别构成有关信息的实时网络共享，同时提供更便捷的数据修改与保护机制。

➤ **用途：**

可供政府机构、数据分析机构及社会调查团体实时提供有关我国各民族人口性别构成的有效信息，可以给特定人员以修改权限来进行数据维护与更新。

➤ **需要解决的问题：**

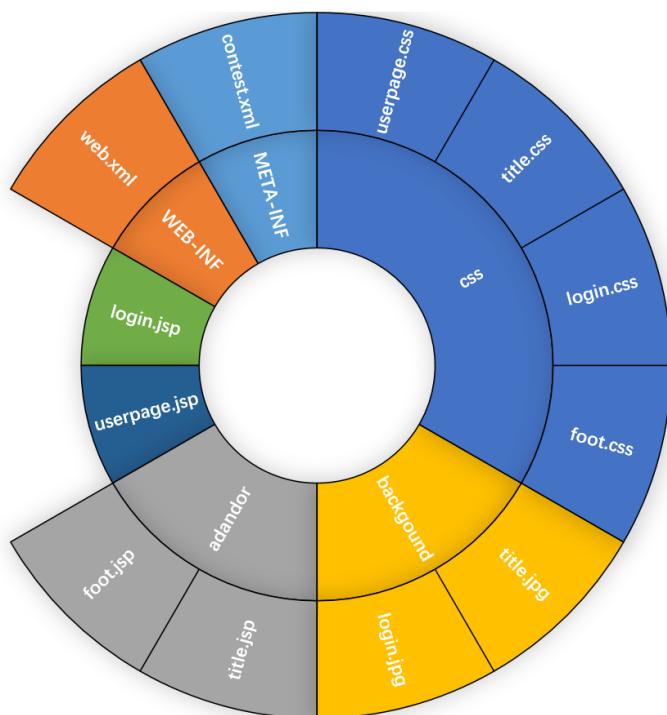
■ 页面要实时更新内容，不是静态页面

■ 数据需要专人维护

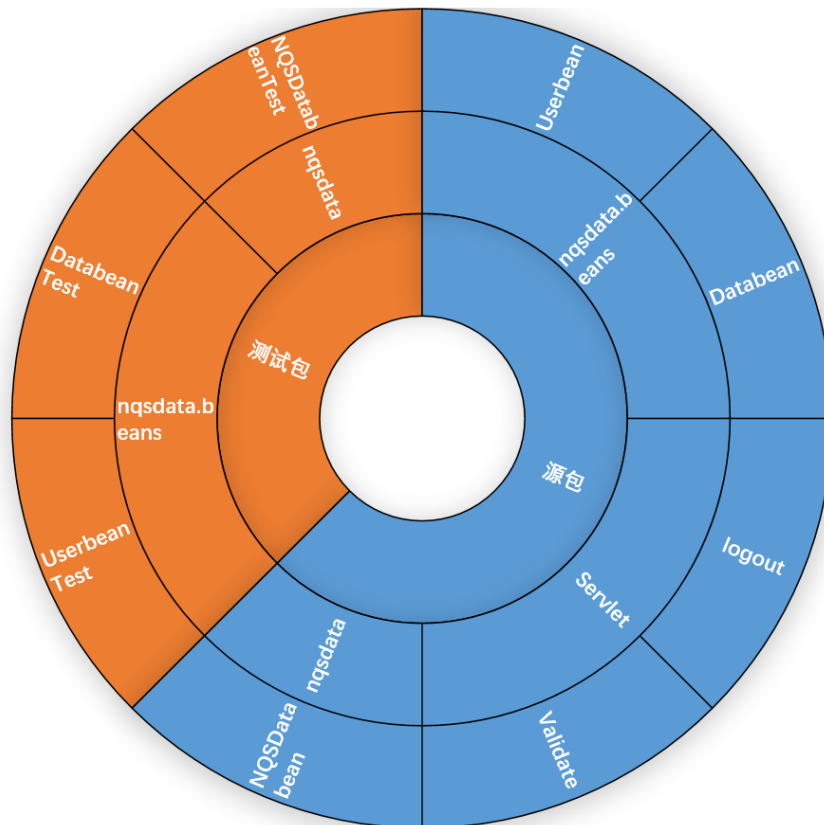
➤ **需求：**

■ 当服务器开启时可以实时查询

■ 用户信息和人口数据需要保存至非即时存储媒介上，即本地数据库



✓ Java packet 结构:



✓ 库结构:



✓ 设计时 web 应用的根目录:

E:\1. 课内资料\1-大一下课程\主-JAVA 小学期\03-实验作业\网络程序设计课程考核\我国各民族人口性别构成网络查询系统\

➤ 各 Jsp 界面的界面及功能设计:

■ login.jsp

- ✓ 实现用户名与密码的输入, 并将用户信息传递给 validate.java(Servlet)
- ✓ 可以接收来自 validate.java(Servlet)的错误提示并告知用户

■ `userpage.jsp`

- ✓ 为数据使用界面，判断登录用户类型提供相应的服务
- ✓ （因为项目没有要求，所以没有制作相应的界面）可以通过 `userbean.java` 添加，以用户名为索引查询、更改和删除用户信息
- ✓ 可以通过 `databean.java` 添加，以民族、男性人口、女性人口为索引查询，以民族为索引添加、更改、删除我国各民族人口性别构成的有关信息
- ✓ 可以登出，跳转至 `logout.java` (Servlet)

■ `title.jsp`

- ✓ 为 `userpage.jsp` 界面提供退出功能及标题展示

■ `foot.jsp`

- ✓ 为所有可见页面提供网站制作信息

➤ 各 Servlet 功能设计：

■ `validate.java`

- ✓ 读取 `login.jsp` 传递的用户信息
- ✓ 通过 `userbean.java` 判断用户信息的真实性与用户类别
- ✓ 如果用户名或密码错误，则跳转回 `login.jsp` 并传递登录错误提示
- ✓ 如果数据库未能连接，则跳转回 `login.jsp` 并传递数据库错误提示
- ✓ 如果用户登录成功，则根据用户类型跳转至 `userpage.jsp`，并传递有关用户名与账户类型的信息

■ `logout.java`

- ✓ 清除用户会话数据
- ✓ 跳转至 `login.jsp`

➤ 各界面之间传输的信息的设计：

■ `<session> String UserName`

- ✓ 作用：用户登录后为平台提供用户名，用作欢迎信息
- ✓ 建立：`validate.java` (Servlet) 登录成功时
- ✓ 销毁：`logout.java` (Servlet) 用户登出时

■ `<session> String UserType`

- ✓ 作用：用户登录后为平台提供用户类型，用作功能界定的信息
- ✓ 类型：`[“admin” , “ordin”]`
- ✓ 建立：`validate.java` (Servlet) 登录成功时
- ✓ 销毁：`logout.java` (Servlet) 用户登出时

■ **<session> String fail**

- ✓ 作用：为 login.jsp 传递登录失败的有关信息
- ✓ 类型：[“loss”，“fail”]
- ✓ 建立：validate.java(Servlet) 登录失败时
- ✓ 销毁：login.jsp fail 信息接收时

■ **<param> String modeValue**

- ✓ 为 userpage.jsp 传递选择的主服务类型
- ✓ 类型：[“查询服务”，“更改服务”]

■ **<param> String stateValue**

- ✓ 为 userpage.jsp 传递选择的查询服务类型
- ✓ 类型：[“以民族名称为索引查询”，“以男性人数为索引查询”，“以女性人数为索引查询”]

■ **<param> String DataMess**

- ✓ 为 userpage.jsp 传递选择的查询服务的填入信息

■ **<param> String alterValue**

- ✓ 为 userpage.jsp 传递选择的更改服务类型
- ✓ 类型：[“添加数据”，“更改数据”，“删除数据”]

■ **<param> String mes**

- ✓ 为 userpage.jsp 传递选择的更改服务表单是否提交的信息
- ✓ 类型：[“putting”]

■ **<param> String NationName, MenPop, WomPop**

- ✓ 为 userpage.jsp 传递选择的更改服务的填入信息

➤ **各 Java Class 功能设计：**

■ **NQSDatabean.java**

- ✓ 可以建立，还原，连接 NQSDatabean.java DB
- ✓ 提供 JDBC 驱动程序类名与数据库路径
- ✓ 包装 SQL 语句，便于使用

■ **Userbean.java (extends NQSDatabean)**

- ✓ 连接 NQSDatabean.java DB
- ✓ 提供添加用户信息的服务
- ✓ (已经实现，但未加入 JSP 界面中)提供以用户名为索引查询、更改和删除用户信息的服务

■ Databean.java (extends NQSDatabean)

- ✓ 连接 NQSDData.Java DB
- ✓ 提供以民族、男性人口、女性人口为索引查询我国各民族人口性别构成的有关信息的服务
- ✓ 提供以民族为索引更改、删除我国各民族人口性别构成的有关信息的服务

➤ Java DB 数据库数据结构:

■ NQSDData

NQSDData					
Cust			Data		
UserName	PassWord	UserType	NationName	MenPop	WomPop

- ✓ UserName(用户名) - String - 唯一
- ✓ PassWord(用户密码) - String
- ✓ UserType(用户类型) - String - [“admin”, “ordin”]
- ✓ NationName(民族) - String - 唯一
- ✓ MenPop(男性人数) - String
- ✓ WomPop(女性人数) - String

(4). 主要功能或算法描述及流程。

➤ 搭建环境相关信息:

- Windows 10
- JavaSE 1.8
- NetBeans IDE 8.1
- Apache Derby 10.11.1.1
- Apache Tomcat 8.0.27.0

➤ web.xml 配置:

■ 响应时长设置:

```
<session-config>
  <session-timeout>
    30
  </session-timeout>
</session-config>
```

■ 欢迎界面设置：

```
<welcome-file-list>
  <welcome-file>login.jsp</welcome-file>
</welcome-file-list>
```

➤ 各 Jsp 界面的界面、功能及传输信息的实现：

■ login.jsp

✓ 进入界面设计：



✓ 登录用户信息错误：



✓ 数据库连接错误：



✓ 源代码:

[illegible]


```

        用户名或密码错误，请重新输入！
    </font>
    <%}else if("loss".equals(fail)){%>
    <font color="#ff6666" face="黑体" size="5">
        数据库连接错误，请检查数据库配置！
    </font>
    <%}
    }%>
</div>
</div>
</div>
</header>
<!-- 嵌入结尾界面，提供界面信息 -->
<jsp:include page="adandor/foot.jsp"/>
</body>
</html>

```

✓ login.css 源代码:

```

header {
    background-image: url(../background/login.jpg);
    color: black;
    width: 1600px;
    height: 990px;
    text-align: center;
    margin: 0 auto;
}
#title {
    display: inline-block;
    font-size: 64px;
    font-family: 黑体;
    padding-top: 200px;
    width: 960px;
}
#titlebottom {
    display: inline-block;
    box-sizing: border-box;
    width: 960px;
    text-align: left;
    font-size: 24px;
    padding: 50px 130px 70px 110px;
}
#login {
    display: inline-block;
    border: 1px solid white;
    width: 960px;
    height: 240px;
}

```

```
#login div:first-child {
    color: #ffffff;
    font-family: 黑体;
    font-size: 40px;
    padding-top: 60px;
    padding-bottom: 15px;
}
#login div input:first-child {
    font-size: 20px;
    width: 300px;
    height: 50px;
    border: none;
}
#login div input:nth-child(2) {
    font-size: 20px;
    width: 300px;
    height: 50px;
    border: none;
}
#login div input:nth-child(3) {
    font-size: 20px;
    width: 120px;
    height: 50px;
    background-color: #000000;
    cursor: pointer;
    color: #ffffff;
    border: none;
}
```

■ userpage.jsp（内部循环更新）

✓ 管理员进入页/查询服务界面：

欢迎您, CIEE!

退出登录



我国各民族人口性别构成网络查询系统

Network Query System for Gender Composition of Population of All Ethnic Groups in China

查询服务

更改服务

索引:

查询

☒ 以民族名称为索引查询
 ☐ 以男性人数为索引查询
 ☐ 以女性人数为索引查询

✓ 普通用户进入页/查询服务界面：

欢迎您，RSMX!

退出登录



我国各民族人口性别构成网络查询系统
Network Query System for Gender Composition of Population of All Ethnic Groups in China

查询服务

索引:

查询

☒ 以民族名称为索引查询 ☐ 以男性人数为索引查询 ☐ 以女性人数为索引查询

✓ 查询服务 - 查询汉族（数据库已有）：

索引:

查询

☒ 以民族名称为索引查询 ☐ 以男性人数为索引查询 ☐ 以女性人数为索引查询

民族名称	男性人口	男性比重	女性人口	女性比重	性别比例
汉族	625032848	0.511967%	595811672	0.488032%	104.9044%

✓ 查询服务 - 查询X族（数据库未有）：

索引:

查询

☒ 以民族名称为索引查询 ☐ 以男性人数为索引查询 ☐ 以女性人数为索引查询

对不起，未找到该民族信息！

✓ 进入更改服务页：


查询服务

更改服务

✓ 更改服务页/添加数据界面：

欢迎您, CIEE!

退出登录



我国各民族人口性别构成网络查询系统

Network Query System for Gender Composition of Population of All Ethnic Groups in China

[< 返回上一级](#)

[添加数据](#)

[更改数据](#)

[删除数据](#)

民族名称：

男性人口：

女性人口：

添加数据

✓ 添加数据 - 添加成功：

民族名称：

男性人口：

女性人口：

添加数据

添加结果：success!

✓ 添加数据 - 民族存在：

民族名称：

男性人口：

女性人口：

添加数据

添加结果：existing!

✓ 添加数据 - 输入有误，（更改数据、删除数据雷同）：

民族名称：


男性人口：

女性人口：

输入有误！

✓ 更改数据界面：

欢迎您，CIEE! 退出登录

 **我国各民族人口性别构成网络查询系统**
Network Query System for Gender Composition of Population of All Ethnic Groups in China

[< 返回上一级](#)

[添加数据](#)

[更改数据](#)

[删除数据](#)

民族名称：

男性人口：

女性人口：

✓ 更改数据 - 更改成功：

民族名称：

男性人口：

女性人口：

更改结果：success！

✓ 更改数据 - 民族不存在，（删除数据雷同）：

民族名称：

男性人口：

女性人口：

更改结果：non-existent!

✓ 删除数据界面：

欢迎您, CIEE! 退出登录

 **我国各民族人口性别构成网络查询系统**
Network Query System for Gender Composition of Population of All Ethnic Groups in China

< 返回上一级

添加数据

更改数据

删除数据

民族名称：

✓ 删除数据 - 删除成功：

民族名称：

删除结果：success!

✓ 源代码：

```
<%--
Document    : userpage
Created on  : 2019-8-11, 13:20:14
Author     : 贾云嵩
--%>

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

```

<jsp:useBean id="Databean" scope="session" class="nqsddata.beans.Databean"/>
<%
    //接收传递的参数
    request.setCharacterEncoding("utf-8");
    //用户类型
    String UserType=(String) session.getAttribute("UserType");
    //服务类型
    String modeValue=request.getParameter("modeValue");
    //查询类型
    String stateValue=request.getParameter("stateValue");
    //更改类型
    String alterValue=request.getParameter("alterValue");
    String alterType[]={"添加数据","更改数据","删除数据"};
    //查询服务参数
    String DataMess=request.getParameter("DataMess");
%>
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" type="text/css" media="screen" href="css/title.css" />
        <link rel="stylesheet" type="text/css" media="screen"
            href="css/userpage.css" />
        <link rel="stylesheet" type="text/css" media="screen" href="css/foot.css" />
        <title>我国各民族人口性别构成网络查询系统</title>
    </head>
    <body>
        <!-- 嵌入标题界面，提供退出服务 -->
        <jsp:include page="adandor/title.jsp"/>
        <div id="all">
            <%//进入界面，或跳转至查询服务
            if("查询服务".equals(modeValue) || modeValue==null){
                设置主页面菜单%>
                <div id="mode">
                    <form action="userpage.jsp" method="post">
                        <input type="button" class="input_out"
                            onmousemove="this.className='input_move'"
                            onmouseout="this.className='input_out'"
                            onclick="this.form.submit()" value="查询服务"/>
                        <input type="hidden" name="modeValue" value="查询服务">
                    </form>
                    <%//判断用户类型，如果是管理员，则开启修改权限
                    if("admin".equals(UserType)) {%>
                        <form action="userpage.jsp" method="post">
                            <input type="button" class="input_out"
                                onmousemove="this.className='input_move'"
                                onmouseout="this.className='input_out'"

```

[illegible]


```

        que=false;
    else{
        que=true;
        NationName=Databean.queWomPop(DataMess)[0];
        MenPop=Databean.queWomPop(DataMess)[1];
        WomPop=Databean.queWomPop(DataMess)[2];
    }
    //查询服务结果显示
    if(que){%>
        <table class="gridtable">
            <tr>
                <td>民族名称</td>
                <td>男性人口</td>
                <td>男性比重</td>
                <td>女性人口</td>
                <td>女性比重</td>
                <td>性别比例</td>
            </tr>
            <tr>
                <%
                    //保留小数位数
                    String MenPor=String.valueOf
(Integer.valueOf(MenPop)/(Integer.valueOf(MenPop)+Integer.valueOf(WomPop)+0.0));
                    if(MenPor.length()>9)
                        MenPor=MenPor.substring(0, 8);
                    String WomPor=String.valueOf
(Integer.valueOf(WomPop)/(Integer.valueOf(MenPop)+Integer.valueOf(WomPop)+0.0));
                    if(WomPor.length()>9)
                        WomPor=WomPor.substring(0, 8);
                    String M_WPor=String.valueOf
(Integer.valueOf(MenPop)/(Integer.valueOf(WomPop)+0.0)*100);
                    if(M_WPor.length()>9)
                        M_WPor=M_WPor.substring(0, 8);
                %>
                <td><%=NationName%></td>
                <td><%=MenPop%></td>
                <td><%=MenPor%></td>
                <td><%=WomPop%></td>
                <td><%=WomPor%></td>
                <td><%=M_WPor%></td>
            </tr>
        </table>
        <%>else{%>
        <font style="margin-left: 20px;
            color: #ff3333;" face="黑体" size="6">
            对不起，未找到该民族信息！
    
```

```

        </font>

        <%}

    }%>
</div>

</div>
<%//进入更改服务界面
}else if("更改服务".equals(modeValue)){
//设置更改服务菜单%>
<div id="alter">
    <form action="userpage.jsp" method="post">
        <input type="button" class="input_out"
            onmousemove="this.className='input_move'"
            onmouseout="this.className='input_out'"
            onclick="this.form.submit()" value="返回上一级"/>
        <input type="hidden" name="modeValue" value="查询服务">
    </form>
    <%for(String alertype:alterType){%>
    <form action="userpage.jsp" method="post">
        <input type="button" class="input_out"
            onmousemove="this.className='input_move'"
            onmouseout="this.className='input_out'"
            onclick="this.form.submit()" value="<%=alertype%>"/>
        <input type="hidden" name="modeValue" value="<%=modeValue%>">
        <input type="hidden" name="alterValue" value="<%=alertype%>">
    </form>
    <%}%>
</div>
<%//处理更改服务数据
String mes=request.getParameter("mes");
if(mes!=null){
    String NationName=request.getParameter("NationName");
    String MenPop=request.getParameter("MenPop");
    String WomPop=request.getParameter("WomPop");
    if("添加数据".equals(alterValue))
        if(NationName!=null && NationName!="" && MenPop!=null
            && MenPop!="" && WomPop!=null && WomPop!="")
            mes="添加结果: "+Databean.addData
                (NationName, MenPop, WomPop)+"!";
        else
            mes="输入有误!";
    else if("更改数据".equals(alterValue))
        if(NationName!=null && NationName!="" && MenPop!=null
            && MenPop!="" && WomPop!=null && WomPop!="")
            mes="更改结果: "+Databean.revData
                (NationName, MenPop, WomPop)+"!";
        else

```

```

        mes="输入有误!";
    else if("删除数据".equals(alterValue))
        if(NationName!=null && NationName!="")
            mes="删除结果: "+Databean.delData(NationName)+"!";
        else
            mes="输入有误!";
    }
    //更改服务子界面%>
<div id="alter-data">
    <div></div>
    <form action="userpage.jsp" method="post">
        <div id="alter-dataput">
            <div>民族名称:</div>
            <input type="text" name="NationName"/>
        </div>
        <%if(!"删除数据".equals(alterValue)){%>
            <div id="alter-dataput">
                <div>男性人口: </div>
                <input type="text" name="MenPop"/>
            </div>
            <div id="alter-dataput">
                <div>女性人口: </div>
                <input type="text" name="WomPop"/>
            </div>
            <%}if(alterValue==null) alterValue="添加数据";%>
            <input type="submit" id="alterbottum" value="<%=alterValue%"/>
            <input type="hidden" name="modeValue" value="<%=modeValue%"/>
            <input type="hidden" name="alterValue" value="<%=alterValue%"/>
            <input type="hidden" name="mes" value="putting">
        </form>
        <%//更改服务结果显示
        if(mes!=null){%>
            <div><%=mes%></div>
        <%}%>
    </div>
    <%}%>
</div>
<!-- 嵌入结尾界面, 提供界面信息 --%>
<jsp:include page="adandor/foot.jsp"/>
</body>
</html>

```

✓ userpage.css 源代码:

```

#all {
    width: 1600px;
    height: 990px;
    margin: 0 auto;
}

```

```
    font-family: 黑体;
    margin-top: 0px;
}
.input_out {
    font-size: 20px;
    width: 300px;
    height: 50px;
    background-color: #99ccff;
    cursor: pointer;
    color: #000000;
    border: none;
    border-bottom: 2px solid white;
}
.input_move {
    font-size: 20px;
    width: 300px;
    height: 50px;
    background-color: #ccffff;
    cursor: pointer;
    color: #000000;
    border: none;
    border-bottom: 2px solid white;
}
#mode {
    float: left;
    width: 300px;
    height: 990px;
    background-color: #6aa5ee;
    margin-top: 0px;
}
#state {
    float: left;
    width: 1300px;
    height: 990px;
    background-color: #deecfa;
    margin-top: 0px;
}
#state div {
    float: left;
    font-size: 30px;
    margin-left: 100px;
    margin-top: 22px;
}
#state input:nth-child(2) {
    float: left;
    font-size: 25px;
```

```
margin-left: 20px;
margin-top: 20px;
width: 500px;
height: 30px;
}
#state input:nth-child(3) {
    float: left;
    font-size: 25px;
    text-align: center;
    margin: 20px 450px 10px 40px;
    width: 90px;
    height: 35px;
}
#state input:nth-child(5) {
    margin-left: 100px;
}
table.gridtable {
    font-family: 隶书;
    font-size: 25px;
    color: #333333;
    border-width: 1px;
    border-color: #666666;
    border-collapse: collapse;
}
table.gridtable th {
    border-width: 1px;
    padding: 10px;
    border-style: solid;
    border-color: #666666;
    background-color: #dedede;
}
table.gridtable td {
    border-width: 1px;
    padding: 10px;
    border-style: solid;
    border-color: #666666;
    background-color: #ffffff;
}
#alter {
    float: left;
    width: 300px;
    height: 990px;
    background-color: #6aa5ee;
    margin-top: 0px;
}
#alter-data {
```

```
float: left;
width: 1300px;
height: 990px;
background-color: #deecfa;
margin-top: 0px;
}
#alter-data div:nth-child(1) {
width: 1300px;
height: 30px;
}
#alter-dataput {
width: 1300px;
height: 60px;
}
#alter-dataput div:nth-child(1) {
float: left;
font-size: 30px;
margin-left: 100px;
margin-top: 10px;
width: 200px;
height: 40px;
}
#alter-dataput input {
float: left;
font-size: 25px;
margin: 10px 200px 5px 0px;
width: 500px;
height: 30px;
}
#alterbottum {
margin: 30px 500px 40px 400px;
font-size: 25px;
width: 200px;
height: 35px;
}
#alter-data div:last-child {
font-size: 30px;
margin-left: 100px;
color: #ff3333;
}
```

■ title.jsp

✓ 显示登录用户名，提供退出按钮：



✓ 源代码：

```
<%--
    Document    : title
    Created on  : 2019-8-11, 21:00:31
    Author      : 贾云嵩
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%-- 下一行提供因 include 界面不在同一路径时的调试用 css 路径 --%>
<!--<link rel="stylesheet" type="text/css" media="screen"
      href="../css/title.css" />-->

<header>
    <%-- 用户欢迎信息 --%>
    <div>
        欢迎您, <%=session.getAttribute("UserName")%>!
    </div>
    <%-- 退出登录, 跳转至 logout.java --%>
    <div>
        <form action="logout" method="post">
            <input type="submit" value="退出登录"/>
        </form>
    </div>
</header>

<%-- 大标题 --%>
<section></section>
```

✓ Title.css 源代码：

```
header {
    background-color: #6aa5ee;
    border-bottom: 2px solid white;
    width: 1600px;
    height: 50px;
    text-align: center;
    margin: 0 auto;
}
header div {
```

```

    font-family: 黑体;
}
header div:nth-child(1) {
    color: #000000;
    font-size: 22px;
    margin-top: 14px;
    margin-left: 15px;
    float: left;
}
header div:nth-child(2) {
    margin-top: 10px;
    margin-right: 15px;
    float: right;
}
header div:nth-child(2) input {
    font-size: 20px;
    width: 120px;
    height: 30px;
    background-color: #b0d2e5;
    cursor: pointer;
    color: #000000;
    border: none;
}

section {
    background-image: url(../background/title.jpg);
    width: 1600px;
    height: 200px;
    text-align: center;
    margin: 0 auto;
    border-bottom: 2px solid white;
}

```

■ foot.jsp

- ✓ 提供网站制作信息:

Copyright. 贾云嵩, 海淀, 2019. 08. 14

- ✓ 源代码:

```

<%--
    Document    : foot
    Created on  : 2019-8-14, 14:28:40
    Author      : 贾云嵩
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

```



```

<!-- 下一行提供因 include 界面不在同一路径时的调试用 css 路径 -->
<!--<link rel="stylesheet" type="text/css" media="screen"
      href="../../css/foot.css" />-->

<div id="foot">
    <!-- 网站制作信息 -->
    <div>
        Copyright.贾云嵩.海淀.2019.08.14
    </div>
</div>

```

✓ foot.css 源代码:

```

#foot {
    background-color: black;
    width: 1600px;
    height: 30px;
    text-align: center;
    font-family: 黑体;
    font-size: 15px;
    margin: 0 auto;
}
#foot div:first-child {
    float: left;
    margin-top: 7px;
    color: white;
    width: 1600px;
    height: 16px;
}

```

➤ 各 Servlet 功能及传输信息的实现:

■ validate.java

✓ 源代码:

```

package Servlet;

//引入相关包
import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

//引入 bean
import nqsddata.beans.*;

/**
 * 用于采集用户登入信息进行比对，跳转至指定页面
 * @author 贾云嵩

```

```

*/
//设置响应路径
@WebServlet(name = "Validate", urlPatterns = {"/Validate"})
public class Validate extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        //接收参数
        String UserName=request.getParameter("UserName");
        String Password=request.getParameter("PassWord");

        //判断登入信息的准确性及用户类型
        Userbean ub=new Userbean();
        String mes=ub.login(UserName, Password);

        //登入信息有误时
        if(mes.equals("fail")){
            request.setAttribute("fail", mes);
            request.getRequestDispatcher("/login.jsp").forward(request, response);
        }

        //数据库连接错误时
        else if(mes.equals("loss")){
            request.setAttribute("fail", mes);
            request.getRequestDispatcher("/login.jsp").forward(request, response);
        }

        //登入用户为"admin" || "ordin"
        else{
            request.getSession().setAttribute("UserName", UserName);
            request.getSession().setAttribute("UserType", mes);
            response.sendRedirect("userpage.jsp");
        }

    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request,

```

```

        HttpServletResponse response) throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "登入验证";
}
}

```

■ logout.java

✓ 源代码:

```

package Servlet;

//引入相关包
import java.io.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

/**
 * 用于清理用户登录数据并重建 session
 * @author 贾云嵩
 */
//设置响应路径
@WebServlet(name = "logout", urlPatterns = {"/logout"})
public class logout extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        //移除 session 中的信息
        request.getSession().removeAttribute("UserName");
        request.getSession().removeAttribute("UserType");
        request.getSession().invalidate();
        response.sendRedirect("login.jsp");
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request,

```

```

        HttpServletResponse response) throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "退出清理";
}
}
}

```

➤ 各 Java Class 功能的实现:

■ NQSDatabean.java

✓ 源代码:

```

package nqsdata;

import java.sql.*;

/**
 * NQSDatabean
 * -整合了 SQL 语言和 Databean 与 Userbean 的交集部分
 * -可以重新生成 NQSDData
 * @author 贾云嵩
 */
public class NQSDatabean {
    /**
     * Java DB 的 JDBC 驱动程序类名
     * @author 贾云嵩
     */
    protected String dbDriver="org.apache.derby.jdbc.EmbeddedDriver";
    /**
     * NQSDData 的路径
     * @author 贾云嵩
     */
    protected String URL="E:\\1.课内资料\\1-大一下课程\\主-JAVA 小学期\\03-实验作业\\网络
程序设计课程考核\\我国各民族人口性别构成网络查询系统";

    protected Connection con;

    protected String sqlCreateTable;    //字符串-建立表用
    // String sqlInsert;        //字符串-插入数据用
    // String sqlSelect;        //字符串-查询数据用
    // String sqlUpdate;        //字符串-更新数据用
    // String sqlDelete;        //字符串-删除数据用

    /**

```

```

* sqlInsert
* @author 贾云嵩
* @param s — Statement
* @param table — 表名
* @param data — 数据, 格式为'data1','data2'...
* @throws SQLException
*/
protected void sqlInsert(Statement s,String table, String data)
    throws SQLException {
    String sqlInsert="INSERT INTO "+table+" VALUES("+data+")";
    s.executeUpdate(sqlInsert);
}

/**
* sqlSelete:查询表中所有信息
* @author 贾云嵩
* @param s — Statement
* @param table — 表名
* @return ResultSet
* @throws SQLException
*/
protected ResultSet sqlSelect(Statement s,String table) throws SQLException {
    return s.executeQuery("SELECT * FROM "+table);
}

/**
* sqlSelete:查询表中指定信息
* @author 贾云嵩
* @param s — Statement
* @param table — 表名
* @param key — 关键词, 格式为: column='value'
* @param data — 数据, 格式为: column1,column2...
* @return ResultSet
* @throws SQLException
*/
protected ResultSet sqlSelect(Statement s,String table, String key, String data)
    throws SQLException {
    String sqlSelect="SELECT "+data+" FROM "+table+" WHERE "+key;
    return s.executeQuery(sqlSelect);
}

/**
* sqlUpdata
* @author 贾云嵩
* @param s — Statement
* @param table — 表名

```

```

    * @param key — 关键词, 格式为: column='value'
    * @param data — 数据, 格式为: column1='value1',column2='value2'...
    * @throws SQLException
    */
protected void sqlUpdate(Statement s,String table, String key, String data)
    throws SQLException {
    String sqlUpdate="UPDATE "+table+" SET "+data+" WHERE "+key;
    s.executeUpdate(sqlUpdate);
}

/**
 * sqlDelete
 * @author 贾云嵩
 * @param s — Statement
 * @param table — 表名
 * @param key — 关键词, 格式为: column='value'
 * @throws SQLException
 */
protected void sqlDelete(Statement s,String table, String key)
    throws SQLException {
    String sqlDelete="DELETE FROM "+table+" WHERE "+key;
    s.executeUpdate(sqlDelete);
}

/**
 * 连接数据库, 更新 con
 * @author 贾云嵩
 */
public NQSDatabean() {
    try {
        //加载 Java DB 驱动
        Class.forName(dbDriver);
        System.out.println(dbDriver+" loaded.");

        //建立数据库连接, 如果不存在, 则新建数据库
        con=DriverManager.getConnection("jdbc:derby:"+URL+"/NQSData;create=true");
        System.out.println("NQSData connected.");

    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

/**

```

```

* 还原数据库信息至原始数据
* @author 贾云嵩
*/
final public void setNQSDData() {
    if(con==null)
        return;
    try {
        Statement s=con.createStatement();

        //建立 User Table 及内部结构
        sqlCreateTable=
            "CREATE TABLE Cust("+
                "UserName CHAR(20) PRIMARY KEY,"+
                "PassWord CHAR(20)," +
                "UserType CHAR(20))";
        s.executeUpdate(sqlCreateTable);
        System.out.println("TABLE Cust created.");
        {
            //默认管理员
            sqlInsert(s, "Cust", "'CIEE','123456','admin'");

            //默认普通用户
            sqlInsert(s, "Cust", "'RSMX','666666','ordin'");
        }

    }

    /**
    * 建立 Data Table 及内部结构
    */
    sqlCreateTable=
        "CREATE TABLE Data("+
            "NationName CHAR(20) PRIMARY KEY,"+
            "MenPop CHAR(20)," +
            "WomPop CHAR(20))";
    s.executeUpdate(sqlCreateTable);
    System.out.println("TABLE Data created.");
    {
        //默认数据
        sqlInsert(s, "Data", "'汉族','625032848','650481765'");
        sqlInsert(s, "Data", "'壮族','8689488','8236893'");
        sqlInsert(s, "Data", "'满族','5401812','4986146'");
        sqlInsert(s, "Data", "'回族','5373741','5212346'");
        sqlInsert(s, "Data", "'维吾尔族','5097594','4971752'");
        sqlInsert(s, "Data", "'苗族','4870328','4555679'");
        sqlInsert(s, "Data", "'彝族','4456382','4258011'");
        sqlInsert(s, "Data", "'土家族','4307260','4046652'");
        sqlInsert(s, "Data", "'藏族','3155625','3126562'");
    }
}

```

```

        sqlInsert(s, "Data", "'蒙古族','2999520','2982320'");
        sqlInsert(s, "Data", "'侗族','1511959','1368015'");
    }

    System.out.print("NQSDData Java DB created.");
    s.close();

} catch (SQLException e) {
    e.printStackTrace();
}
}
}

```

■ Userbean.java

✓ 源代码:

```

package nqsdata.beans;

import java.sql.*;
import nqsdata.NQSDatabean;

public class Userbean extends NQSDatabean {

    /**
     * 连接数据库, 更新 con
     * @author 贾云嵩
     */
    public Userbean() {
        super();
    }

    //-----function-validate-----
    /**
     * 检验登入用户信息:validate.jsp
     * @author 贾云嵩
     * @param UserName
     * @param PassWord
     * @return "fail"|"admin"|"ordin"
     */
    public String login(String UserName,String PassWord){
        if(con==null)
            return "loss";
        try {
            Statement s=con.createStatement();

            //与数据库比对信息

```



```

        ResultSet rs=sqlSelect(s, "Cust", "UserName='"+UserName+"'",
            "PassWord,UserType");
        if(rs.next())
            if(rs.getString("PassWord").replaceAll("\\s*", "").equals(PassWord))
                return rs.getString("UserType").replaceAll("\\s*", "");

    } catch (SQLException e) {
        return "loss";
    }
    return "fail";
}

//-----function-admin-----
/**
 * 添加用户:admin.jsp
 * @author 贾云嵩
 * @param UserName
 * @param PassWord
 * @param UserType
 * @return "success"||"existing"
 */
public String addUser(String UserName,String PassWord,String UserType){
    try {
        Statement s=con.createStatement();

        //与数据库比对信息
        if(sqlSelect(s, "Cust", "UserName='"+UserName+"'",
            "PassWord,UserType").next())
            return "existing";

        //添加用户
        sqlInsert(s, "Cust", "'"+UserName+"','"+PassWord+"','"+UserType+"'");

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return "success";
}

/**
 * 删除用户:admin.jsp
 * @author 贾云嵩
 * @param UserName
 * @return "success"||"non-existent"
 */
public String delUser(String UserName){

```

```

    try {
        Statement s=con.createStatement();

        //与数据库比对信息
        if(sqlSelect(s, "Cust", "UserName='"+UserName+"'",
            "PassWord,UserType").next()) {
            sqlDelete(s, "Cust", "UserName='"+UserName+"'");
            return "success";
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return "non-existent";
}

/**
 * 查询用户密码:admin.jsp
 * @author 贾云嵩
 * @param UserName
 * @return PassWord||"non-existent"
 */
public String getPassWord(String UserName){
    try {
        Statement s=con.createStatement();

        //与数据库比对信息
        ResultSet rs=sqlSelect(s, "Cust", "UserName='"+UserName+"'",
            "PassWord,UserType");
        if(rs.next())
            return rs.getString("PassWord").replaceAll("\\s*", "");

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return "non-existent";
}

/**
 * 查询用户类型:admin.jsp
 * @author 贾云嵩
 * @param UserName
 * @return UserType||"non-existent"
 */
public String getUserType(String UserName){
    try {

```

```

        Statement s=con.createStatement();

        //与数据库比对信息
        ResultSet rs=sqlSelect(s, "Cust", "UserName='"+UserName+"'",
            "PassWord,UserType");
        if(rs.next())
            return rs.getString("UserType").replaceAll("\\s*", "");

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return "non-existent";
}

/**
 * 修改用户密码:admin.jsp
 * @author 贾云嵩
 * @param UserName
 * @return "success"||"non-existent"
 */
public String revPassWord(String UserName,String PassWord){
    try {
        Statement s=con.createStatement();

        //与数据库比对信息
        ResultSet rs=sqlSelect(s, "Cust", "UserName='"+UserName+"'",
            "PassWord,UserType");
        if(rs.next()) {
            sqlUpdate(s, "Cust", "UserName='"+UserName+"'",
                "PassWord='"+PassWord+"'");
            return "success";
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return "non-existent";
}

/**
 * 修改用户类型:admin.jsp
 * @author 贾云嵩
 * @param UserName
 * @return "success"||"non-existent"
 */
public String revUserType(String UserName,String UserType){

```

```

try {
    Statement s=con.createStatement();

    //与数据库比对信息
    ResultSet rs=sqlSelect(s, "Cust", "UserName='"+UserName+"'",
        "PassWord,UserType");
    if(rs.next()){
        sqlUpdate(s, "Cust", "UserName='"+UserName+"'",
            "UserType='"+UserType+"'");
        return "success";
    }

} catch (SQLException e) {
    e.printStackTrace();
}
return "non-existent";
}
}

```

■ Databean.java

✓ 源代码:

```

package nqsdata.beans;

import java.sql.*;
import nqsdata.NQSDatabean;

public class Databean extends NQSDatabean {

    /**
     * 连接数据库, 更新 con
     * @author 贾云嵩
     */
    public Databean() {
        super();
    }

    //-----function-all-----
    /**
     * 按"民族"查询:admin.jsp & ordin.jsp
     * @author 贾云嵩
     * @param NationName
     * @return { , , } || null
     */
    public String[] queNationName(String NationName){
        try {

```

```

Statement s=con.createStatement();

//与数据库比对信息
ResultSet rs=sqlSelect(s, "Data", "NationName='"+NationName+"'",
    "NationName,MenPop,WomPop");
if(rs.next()) {
    String[] data= {"NationName","MenPop","WomPop"};
    data[0]=rs.getString("NationName").replaceAll("\\s*", "");
    data[1]=rs.getString("MenPop").replaceAll("\\s*", "");
    data[2]=rs.getString("WomPop").replaceAll("\\s*", "");
    return data;
}

} catch (SQLException e) {
    e.printStackTrace();
}
return null;
}

/**
 * 按"男性人口"查询:admin.jsp & ordin.jsp
 * @author 贾云嵩
 * @param MenPop
 * @return { , , } || null
 */
public String[] queMenPop(String MenPop){
    try {
        Statement s=con.createStatement();

        //与数据库比对信息
        ResultSet rs=sqlSelect(s, "Data", "MenPop='"+MenPop+"'",
            "NationName,MenPop,WomPop");
        if(rs.next()) {
            String[] data= {"NationName","MenPop","WomPop"};
            data[0]=rs.getString("NationName").replaceAll("\\s*", "");
            data[1]=rs.getString("MenPop").replaceAll("\\s*", "");
            data[2]=rs.getString("WomPop").replaceAll("\\s*", "");
            return data;
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return null;
}

```

```

/**
 * 按"女性人口"查询:admin.jsp & ordin.jsp
 * @author 贾云嵩
 * @param WomPop
 * @return { , , } || null
 */
public String[] queWomPop(String WomPop){
    try {
        Statement s=con.createStatement();

        //与数据库比对信息
        ResultSet rs=sqlSelect(s, "Data", "WomPop='"+WomPop+"'",
            "NationName,MenPop,WomPop");
        if(rs.next()) {
            String[] data= {"NationName","MenPop","WomPop"};
            data[0]=rs.getString("NationName").replaceAll("\\s*", "");
            data[1]=rs.getString("MenPop").replaceAll("\\s*", "");
            data[2]=rs.getString("WomPop").replaceAll("\\s*", "");
            return data;
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return null;
}

//-----function-admin-----
/**
 * 添加数据:admin.jsp
 * @author 贾云嵩
 * @param NationName
 * @param MenPop
 * @param WomPop
 * @return "success"||"existing"
 */
public String addData(String NationName,String MenPop,String WomPop){
    try {
        Statement s=con.createStatement();

        //与数据库比对信息
        if(sqlSelect(s, "Data", "NationName='"+NationName+"'",
            "MenPop,WomPop").next())
            return "existing";

        //添加用户

```

```

        sqlInsert(s, "Data", ""+NationName+"', '"+MenPop+"', '"+WomPop+"'");

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return "success";
}

/**
 * 通过民族名删除数据:admin.jsp
 * @author 贾云嵩
 * @param NationName
 * @return "success"||"non-exist"
 */
public String delData(String NationName){
    try {
        Statement s=con.createStatement();

        //与数据库比对信息
        if(sqlSelect(s, "Data", "NationName='"+NationName+"'",
            "MenPop,WomPop").next()) {
            sqlDelete(s, "Data", "NationName='"+NationName+"'");
            return "success";
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return "non-existent";
}

/**
 * 以民族名为 key 修改数据:admin.jsp
 * @author 贾云嵩
 * @param NationName
 * @param MenPop
 * @param WomPop
 * @return "success"||"non-exist"
 */
public String revData(String NationName,String MenPop,String WomPop){
    try {
        Statement s=con.createStatement();

        //与数据库比对信息
        ResultSet rs=sqlSelect(s, "Data", "NationName='"+NationName+"'",
            "MenPop,WomPop");

```

```

        if(rs.next()){
            sqlUpdate(s, "Data", "NationName='"+NationName+"'",
                "MenPop='"+MenPop+"',WomPop='"+WomPop+"'");
            return "success";
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return "non-existent";
}
}

```

➤ 各 Java Test Class 功能的实现(检验 class 的正确性):

■ NQSDatabeanTest.java

✓ 源代码:

```

package nqsdata;

public class NQSDatabeanTest {

    public static void main(String[] args) {
        new NQSDatabean();
    }

}

```

■ UserbeanTest.java

✓ 源代码:

```

package nqsdata.beans;

import nqsdata.beans.Userbean;

public class UserbeanTest {

    public static void main(String[] args) {
        Userbean ub=new Userbean();
        System.out.println(ub.login("RSMX", "666666"));
        System.out.println(ub.getPassWord("RSMX"));
        System.out.println(ub.getUserType("RSMX"));
    }

}

```


■ DatabeanTest.java

✓ 源代码:

```
package nqsdata.beans;

public class DatabeanTest {

    public static void main(String[] args) {
        Databean db=new Databean();
        System.out.println(db.addData("X 族", "666", "666"));
        System.out.println(db.addData("X 族", "666", "666"));
        System.out.println(db.queNationName("X 族")[0]);
        System.out.println(db.queNationName("X 族")[1]);
        System.out.println(db.revData("X 族", "555", "555"));
        System.out.println(db.queNationName("X 族")[2]);
        System.out.println(db.delData("X 族"));
        System.out.println(db.queNationName("X 族"));
        System.out.println(db.delData("X 族"));
    }
}
```

➤ Java DB 数据库的实现:


■ 建立过程:

- ✓ 导入 derby JAR 包
- ✓ 运行 NQSDatabean.java 的 setNQSdata() 函数建立数据库

■ 内部默认数据:

✓ Cust table

- ✧ 管理员: CIEE 密码: 123456
- ✧ 普通用户: RSMX 密码: 666666

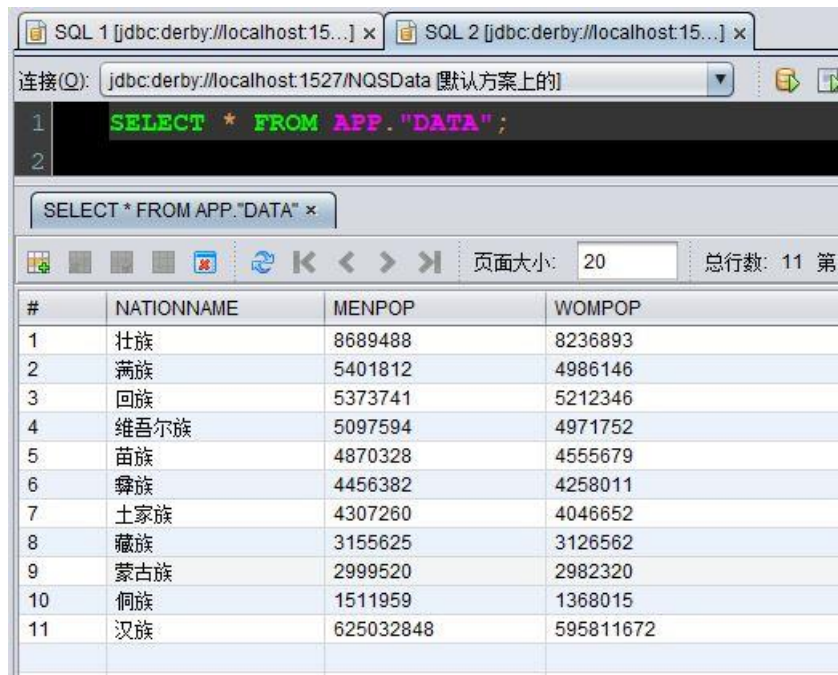


The screenshot shows a SQL client window with two tabs: 'SQL 1 [jdbc:derby://localhost:1527/...] x' and 'SQL 2 [jdbc:derby://localhost:1527/...] x'. The active tab is 'SQL 2'. The connection string is 'jdbc:derby://localhost:1527/NQSDData 默认方案上的'. The query entered is 'SELECT * FROM APP.CUST;'. The result is displayed in a table with 4 columns: '#', 'USERNAME', 'PASSWORD', and 'USERTYPE'. There are 2 rows of data.

#	USERNAME	PASSWORD	USERTYPE
1	CIEE	123456	admin
2	RSMX	666666	ordin

✓ Data table

✧ 来自题目的原始民族数据



The screenshot shows a SQL IDE interface with two tabs: 'SQL 1 [jdbc:derby://localhost:1527/NQSDData [默认方案上的]]' and 'SQL 2 [jdbc:derby://localhost:1527/NQSDData [默认方案上的]]'. The active tab is SQL 1, which contains the query: `SELECT * FROM APP."DATA" ;`. Below the query editor, a toolbar shows various icons for execution and navigation. The result pane displays a table with 11 rows and 4 columns: #, NATIONNAME, MENPOP, and WOMPOP. The data is as follows:

#	NATIONNAME	MENPOP	WOMPOP
1	壮族	8689488	8236893
2	满族	5401812	4986146
3	回族	5373741	5212346
4	维吾尔族	5097594	4971752
5	苗族	4870328	4555679
6	彝族	4456382	4258011
7	土家族	4307260	4046652
8	藏族	3155625	3126562
9	蒙古族	2999520	2982320
10	侗族	1511959	1368015
11	汉族	625032848	595811672

(5). 文献资料

[1] 东北少数民族人口的性别结构. 卢守亭, 侯胜利. 河南洛阳. 1. 7031 (2007) 04-0112-04